ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ, РОБОТОТЕХНИЧЕСКИЕ СИСТЕМЫ, МАШИННОЕ ОБУЧЕНИЕ

УДК 004.08

АВТОМАТИЗИРОВАННОЕ ВЫЯВЛЕНИЕ РАКА МОЛОЧНОЙ ЖЕЛЕЗЫ С ПОМОЩЬЮ НЕЙРОННОЙ СЕТИ

Халиль Аббуд

Воронежский государственный университет

Аннотация. Статья посвящена разработке алгоритма обработки медицинских изображений для диагностики раковых опухолей. Рассматривается роль цифровых изображений в медицине, их разнообразие по характеристикам контрастности и четкости, а также необходимость автоматизации диагностики. Основное внимание уделено созданию алгоритма на MATLAB для анализа МРТ молочной железы, выявления опухолей, их изоляции и расчета площади. Используются методы морфологической обработки, такие как операции вскрытия и закрытия, которые улучшают выделение опухолей. Результаты алгоритма способствуют автоматизации диагностики и повышению точности прогнозирования состояния, заданного пользователем.

Ключевые слова: обработка изображений, морфологические операции, расширение, эрозия, нейронные сети.

Введение

С быстрым развитием науки о цифровой обработке изображений и её широким применением в медицине, диагностика заболеваний с использованием медицинских изображений становится всё более эффективной. Сегодня исследователи сосредоточены на разработке методов, позволяющих быстро, точно и легко извлекать необходимую информацию из визуальных данных, улучшая качество диагностики.

Компьютерная диагностика определяется как процесс анализа медицинских изображений с использованием компьютерных технологий. Она предоставляет врачу вторичное заключение, помогая выявлять патологические изменения и уточнять диагноз. Однако окончательное решение всегда остается за врачом. Компьютерная диагностика выступает в роли вспомогательного инструмента, который используется как «второй считыватель» и повышает точность и скорость работы специалистов.

Система компьютерной диагностики опухолей делится на три этапа:

Этап выявления: определение подозрительных областей, представляющих интерес (например, аномалий в ткани молочной железы).

Извлечение признаков: анализ параметров, используемых для классификации.

Классификация: разделение случаев на здоровые и патологические.

Рак молочной железы развивается, когда клетки ткани молочной железы начинают бесконтрольно делиться, образуя опухоль. Большинство опухолей доброкачественные, однако некоторые клетки могут стать злокачественными. Злокачественные опухоли обладают способностью проникать в соседние ткани и распространяться через лимфатические сосуды и кровоток, поражая удалённые органы. Тяжесть заболевания зависит от типа рака:

Локализованный рак: поражение ограничено областью молочной железы и не выходит за её пределы. На ранних стадиях этот тип поддается успешному лечению, но при отсутствии диагностики и лечения он может прогрессировать.

Метастатический рак: характеризуется проникновением опухоли через стенки протоков молочных желез в жировую ткань груди. Через кровеносные и лимфатические сосуды этот рак распространяется на другие части тела, значительно усложняя лечение. Эффективность раннего выявления и правильной диагностики является ключевым фактором в борьбе с этим заболеванием.

1. Обработка изображений

Это одна из отраслей компьютерных наук, которая заинтересована в выполнении операций с изображениями с целью их улучшения в соответствии с определенными критериями или извлечения из них некоторой информации

Система обработки изображений выполняет четыре основные задачи:

1) получение изображения, т. е. преобразование оптического изображения в цифровое в памяти компьютера;

2) сохранение изображения;

3) обработка;

4) изображение.

Обработка изображений направлена на:

1. Улучшение фотографической информации для ее интерпретации человеком, например, улучшение контрастности, устранение шумов, обработка ложных цветов и восстановление изображений путем извлечения из них некоторой недостающей информации.

2. Обработка данных изображений для понимания и восприятия компьютером.

Покажем две морфологические операции, расширение и эрозия, которые являются двумя основными операциями, от которых зависят многие алгоритмы обработки изображений.

Расширение: Расширение выполняется оперативно путем перемещения центра формирующего элемента по всему изображению, начиная с верхнего левого угла (0,0), и в каждом элементе изображения, если формирующий элемент пересекается с каким-либо элементом (значение 1), мы создаем элемент изображения, расположенный под центром формирующего элемента. элемент — элемент изображения, содержащий результат расширения (ему присваивается значение 1).

Таким образом, процесс расширение расширяет объекты на изображении, удлиняет их конечности и сужает отверстия.

Эрозия: Процесс эрозии осуществляется аналогично процессу растяжения путем перемещения центра формирующего элемента по всему изображению, начиная с левого верхнего угла (0,0) и у каждого элемента изображения Z2. Если элемент формирования равен единице, принимается значение результирующего элемента изображения, соответствующего центру элемента формирования. Исходная фотография равна единице.

Поэтому процесс эрозии сжимает объекты на изображении, укорачивает их края и расширяет отверстия.

Выполнение операций расширения и удаления изображения в определенной последовательности приводит к двум новым морфологическим операциям, открытия и закрытия, которые определяются следующим образом:

открытие: это операция удаления, за которой следует операция расширения.

закрытие: это процедура расширения, за которой следует операция удаления.

На рис. 1 показаны в иллюстративной форме процессы открытия и закрытия, где (*a*) — необязательная форма, содержащая выступы и впадины, (*c*) — результат эрозии *a* с помощью круглого формующего элемента, (*e*) — результат раскрытия *a*, (*g*) — результат значения расширения *a* и (*i*) являются результатом закрытия a, в то время как цифры (*h*, *f*, *d*, *b*) являются иллюстрацией эффекта процесса, в результате которого была создана фигура, следующая за каждой из них.

2. Нейронные сети

Искусственная нейронная сеть состоит из ряда обрабатывающих элементов, называемых нейронами, узлами или единицами, каждая ячейка имеет ряд входных сигналов с периферии

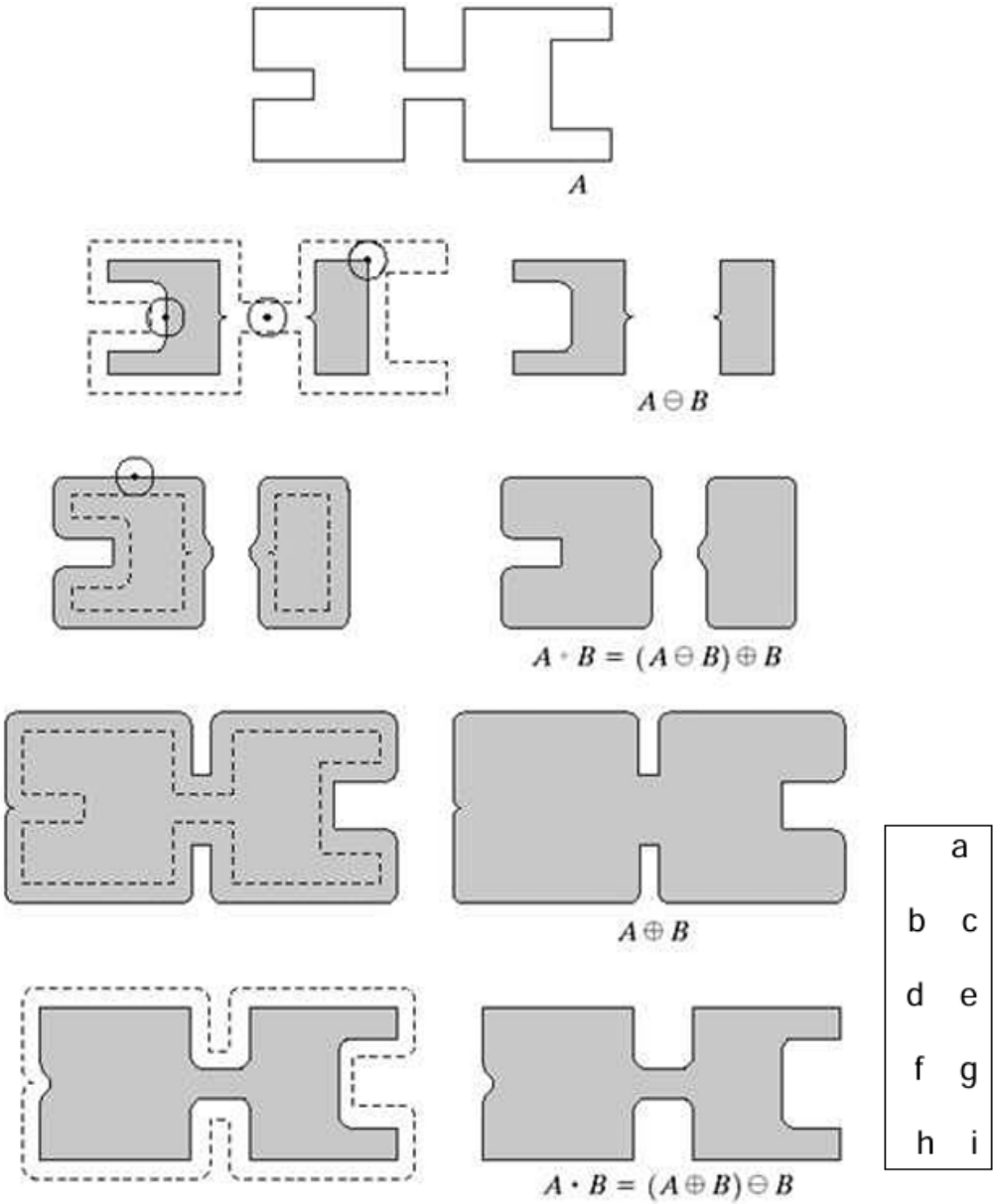


Рис. 1. Операции открытия и закрытия с использованием круглого формующего элемента

или от других нейронов и имеет только один выходной сигнал, распределенный между другими нейронами или выходящий на периферию. Нейронам удалось решить многие задачи с высокой эффективностью, например, в распознавании образов, обработке изображений, распознавании звуков и в области управления.

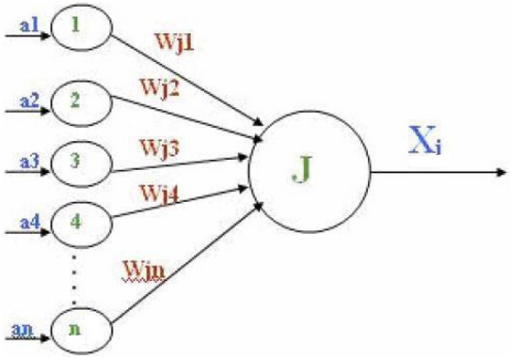


Рис. 2. Искусственная нервная клетка

Из рисунка видно, что нейрон состоит из:

1 — **входных сигналов** (a1,a2,a3,..... an)

2 — **силы тяжести** (веса: w1, w2, w3, win), где вес выражает интенсивность корреляции между элементом до и элементом после.

3 — **обрабатывающий элемент** (Processing Element), и этот элемент разделен на две части:

– Коллектор (Adder) для сбора сигналов на входе

– Передаточная функция или функция активации. Эта функция ограничивает выходной сигнал нейрона, поэтому она называется функцией сжатия, поскольку она помещает выходной сигнал в диапазон [0,1] или в диапазон [–1,1].

4 — **вывод** Xj (Output).

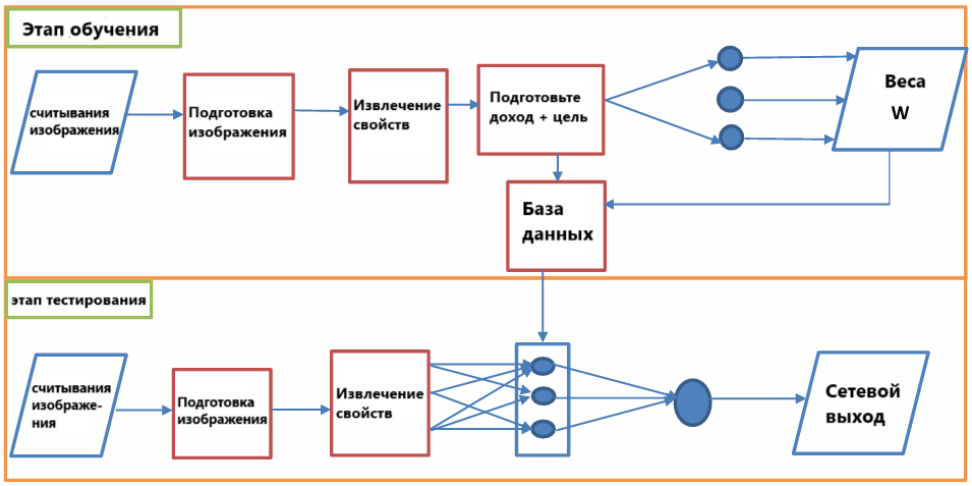


Рис. 3. Алгоритм, использованный при проектировании модели

На рис. 3 показаны два этапа обучения и тестирования, где на этапе обучения считывается изображение, а затем обрабатывается с помощью некоторых морфологических процессов, после чего из изображения извлекаются интересующие характеристики, чтобы обучить на них нейронную сеть для диагностики текущего состояния, и на основе выходных данных нейронной сети получается значения веса и сохраняется в файле базы данных. На этапе тестирования считывается изображения и обрабатывается, как на этапе обучения, а затем извлекаются характеристики изображения, такие как входные данные нейронной сети, с весами, полученными на этапе обучения, чтобы получить диагностический результат на выходе сети.

3. Практическая часть проекта

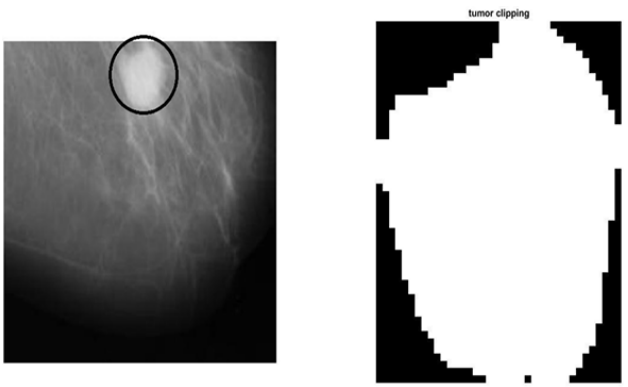


Рис. 4. Отделение опухоли от остальной части изображения

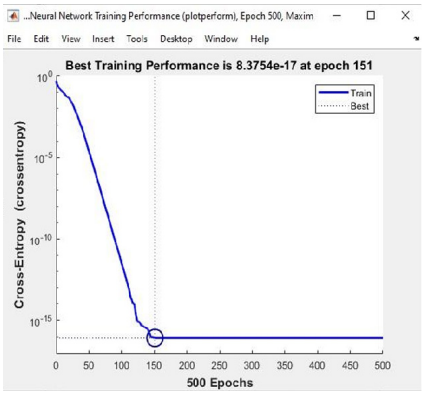


Рис. 5. Тренировочные роли для достижения наилучшего результата

Рис. 5 показывает снижение ошибки (Cross-Entropy) при обучении нейронной сети. Синяя линия отражает процесс оптимизации, резкое снижение в начале свидетельствует об успешном обучении. Последующая стабилизация указывает на достижение оптимального уровня без переобучения.

Результаты

Протестировано в системе 30 изображений, включая 15 изображений, на которых сеть была предварительно обучена, разделенных на 5 изображений здоровых людей, 5 изображений злокачественных раковых образований и 5 изображений доброкачественных раковых образований. Уровень точности результатов системы составил 100 %. Затем протестировалась система на 15 изображениях, на которых сеть не была обучена. Заранее было 5 изображений здорового образования, 5 изображений злокачественного образования и 5 изображений доброкачественного образования. Точность системы составила 90 %, так как среди из 5 нормальных изображений система распознает 5, из 5 изображений доброкачественного образования 4, а из 5 изображений злокачественного образования 4.

Заключение

Разработанная программа продемонстрировала свою эффективность как вспомогательный и поддерживающий диагностический инструмент для принятия обоснованных медицинских решений относительно выбора метода лечения. Она позволяет врачам заранее определить площадь опухоли в квадратных сантиметрах, что способствует оценке целесообразности лечения, включая возможность проведения операции по удалению опухоли. Данная функциональность помогает медикам учитывать площадь и стадию опухоли при выборе стратегии лечения, что в конечном итоге повышает точность и персонализированность медицинской помощи.

Перспективы дальнейшего развития программы включают:

Сегментацию опухоли в трех измерениях, что позволит повысить точность анализа и предоставить врачам более детальную информацию о ее форме и размере.

Развитие алгоритмов на основе глубокого обучения, чтобы нейронная сеть могла различать типы опухолей, что является важным шагом для персонализированной диагностики и лечения.

Добавление функций измерения расстояния от краев опухоли до ближайших чувствительных областей, что позволит врачу определить максимально допустимый размер удаляемой ткани, минимизируя риск повреждения жизненно важных структур.

Реализация этих перспективных направлений существенно расширит функциональные возможности программы, повысит ее клиническую ценность и интеграцию в процесс принятия медицинских решений. Это создаст основу для дальнейшего внедрения современных технологий в медицину, улучшая качество лечения и прогнозы для пациентов.

Литература

1. *Маан Амма* Системы отображения и обработки медицинских изображений. – Публикации Дамасского университета, 2007–2008 г.

2. *Марьям Сай* Нейронные сети. – Публикации Университета Аль-Баас, 2017–2018 г.

3. *Майк Диксон* Рак молочной железы. Публикации Аль-Муалеф. 2013 г.

УДК 004.85 : 004.946

ПРИМЕНЕНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ В ТЕХНОЛОГИЯХ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

Т. В. Азарнова, Д. А. Поздняков

Воронежский государственный университет

Аннотация. В данной статье рассматривается влияние методов машинного обучения (ML) на развитие технологий виртуальной реальности (VR). Благодаря интеграции ML, VR-технологии становятся более доступными и многогранными, открывая новые возможности в таких областях, как образование, медицина, развлечения и производство. Машинное обучение способно улучшить взаимодействие пользователей с виртуальными средами, адаптируя контент под индивидуальные потребности и эмоции. Также в работе представлены примеры применения ML в образовательных и медицинских VR-системах, описаны перспективы и вызовы, связанные с внедрением ML в VR, например, необходимость в больших объемах данных и вычислительных мощностей, а также возможные направления будущих исследований.

Ключевые слова: виртуальная реальность, VR, машинное обучение, ML, адаптивные алгоритмы, образование, медицина, реабилитация, эмоциональное состояние, 3D-моделирование, обратная связь, системы распознавания жестов, персонализация, вычислительные ресурсы.

Введение

В последние годы технологии виртуальной реальности (VR) претерпели значительные изменения благодаря внедрению методов машинного обучения (ML). Эти технологии стали более доступными и многофункциональными, что открывает новые горизонты для их применения в различных областях, таких как образование, медицина, развлечения и производство. В данной статье рассматривается влияние методов машинного обучения на развитие технологий виртуальной реальности, преимущества и перспективы их внедрения.

1. Виртуальная реальность и машинное обучение

Виртуальная реальность — это иммерсивная технология, которая создает искусственную среду, позволяя пользователям оперировать с 3D-объектами в трехмерном пространстве. Основные технологии, задействованные в VR, включают шлемы виртуальной реальности, трекеры движений и контроллеры [1].

Машинное обучение является важнейшим направлением искусственного интеллекта, позволяющим обрабатывать большие объемы информации, находить скрытые закономерности и использовать данные закономерности в задачах управления и принятия решений.

Комбинация технологий VR и ML может быть направлена на развитие и улучшение уже существующих VR технологий. Эти технологии могут включать в себя приложения в сферах образования, медицины, развлекательной индустрии и производства.

Машинное обучение может предоставить возможность для развития более интерактивных и адаптивных пользовательских интерфейсов. Данные алгоритм помогают улучшить восприятие пользователями виртуальной среды, что делает взаимодействие более естественным и продуктивным. Например, алгоритмы могут анализировать поведение пользователей, чтобы в реальном времени адаптировать содержание и сложность взаимодействия, что значительно улучшает пользовательский опыт.

1.1. Применение ML для улучшения взаимодействия в VR

Одним из ключевых аспектов использования методов машинного обучения в виртуальной реальности является улучшение взаимодействия между пользователями и технологиями виртуальной реальности.

Алгоритмы ML могут анализировать данные о взаимодействиях, распознавать паттерны и адаптировать сценарии взаимодействия на основе поведения и эмоционального состояния пользователя. В работе Oyebola O. [2] для предсказания успешности взаимодействия используется логистическая регрессия, описывающая зависимость вероятности успешного взаимодействия от различных факторов взаимодействия. Если алгоритм обнаруживает, что вероятностная модель логистической регрессии указывает на наличие трудностей (1), то применяются методы кластеризации (2) для группировки пользователей с похожими проблемами:

$$P(X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n)}}, \tag{1}$$

где $P(X)$ — вероятность того, что пользователь испытывает трудности при управлении объектом,

$$J = \sum_{i=1}^{k} \sum_{j=1}^{n} \left\| x_j^{(i)} - c_i \right\|^2, \tag{2}$$

где J — функция стоимости, $x_j^{(i)}$ — точки данных в кластере i, c_i — центроид кластера.

В результате, если пользователь испытывает трудности с управлением объектом, ML-алгоритмы могут своевременно предложить подсказки или изменить параметры взаимодействия для облегчения процесса, используется графическая визуализация данных для лучшего понимания паттернов поведения. Таким образом, если пользователь испытывает трудности с управлением объектом, ML-алгоритмы могут обнаружить это и предлагать подсказки или осуществлять изменение параметров взаимодействия для облегчения процесса.

Алгоритмы ML в сочетании с технологиями компьютерного зрения могут распознавать эмоции пользователей по выражению лица и жестам [3, 4]. Это открывает возможность для адаптации виртуальной среды в зависимости от эмоционального состояния пользователя. Так, в обучающих VR приложениях, если система определяет, что студент испытывает стресс, она может снизить уровень сложности задач, обеспечивая комфортную атмосферу для обучения [4].

Алгоритмы машинного обучения могут автоматически генерировать 3D-объекты и анимацию, что значительно облегчает и ускоряет процесс разработки [5]. Это позволяет разработчикам сосредоточиться на более важных аспектах, таких как пользовательский опыт и содержание приложения. На рис. 1. приведен пример генерации 3D моделей с помощью машинного обучения.

Кроме того, системы распознавания жестов и голосового управления, основанные на ML, становятся важными элементами управления в VR. Эти технологии позволяют пользователям

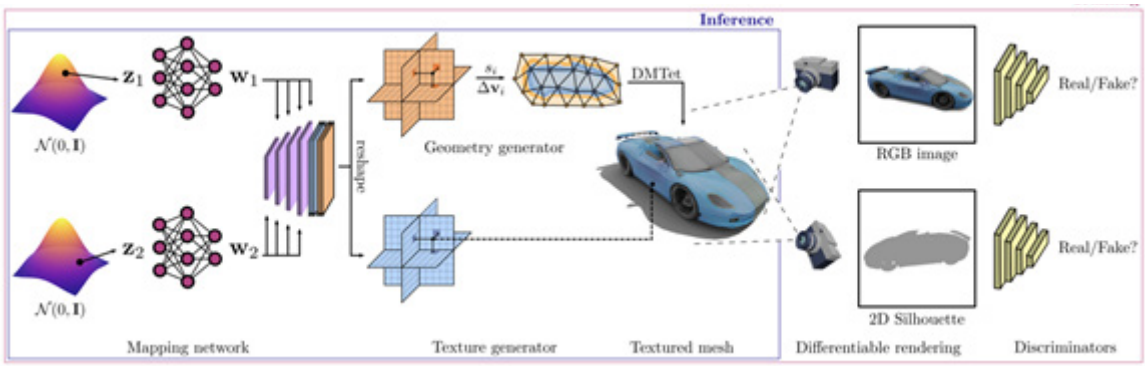


Рис. 1. Генерация 3D моделей с помощью машинного обучения

взаимодействовать с виртуальными объектами без необходимости использования традиционных контроллеров. Они обеспечивают большую свободу движений и делают интерфейсы более интуитивно понятными. Например, такие компании, как Leap Motion, разрабатывают решения, основанные на машинном обучении, для отслеживания движений рук, что позволяет пользователям манипулировать виртуальными объектами только с помощью жестов [6].

1.2. Применение ML в VR технологиях обучения и образования

Технологии виртуальной реальности уже продемонстрировали свою эффективность применительно к сфере образования, позволяя создавать симуляции, которые помогают учащимся лучше понимать сложные концепции. Машинное обучение добавляет еще один уровень адаптивности в образовательные VR-программы. С помощью анализа данных об успеваемости и взаимодействии студентов, алгоритмы могут подстраивать учебный процесс под индивидуальные потребности каждого ученика. Это включает в себя предоставление персонализированных заданий, рекомендаций по материалам и динамическую настройку сложности задач [7].

Технологии VR и ML в образовании существенно используются в медицинских учебных заведениях, где студенты могут проводить виртуальные операции или исследования. Применение машинного обучения позволяет данные компоненты учебного процесса адаптировать к уровню квалификации учащихся, предоставляя обратную связь и рекомендации, что в конечном итоге улучшает качество обучения. Такой подход не только повышает вовлеченность, но и способствует лучшему усвоению материала, позволяя студентам учиться на своих ошибках в безопасной и контролируемой среде.

В медицине виртуальная реальность уже зарекомендовала себя как инструмент для тренировки хирургов, лечения фобий и управления болевыми синдромами. Использование методов машинного обучения в этой области открывает новые возможности для повышения точности и эффективности лечения. Например, ML-алгоритмы могут анализировать данные о состоянии пациента и предсказывать, какая терапия будет наиболее эффективной. Это может включать не только традиционное лечение, но и VR-терапию, которая может быть адаптирована в зависимости от индивидуальных предпочтений и чувствительности пациента [8].

1.3. Применение ML в VR технологиях психотерапии и реабилитации

В последние годы VR применяется в психотерапии, особенно в лечении посттравматического стрессового расстройства (ПТСР). Методы машинного обучения могут быть использованы для анализа поведения клиентов, а также для адаптации сценариев в зависимости от их реакций. Например, алгоритмы могут отслеживать уровень тревожности пациентов и изменять интенсивность терапии в реальном времени [9].

Виртуальные симуляторы также используются в процессе реабилитации после травм [10]. Они позволяют пациентам выполнять упражнения в игровой форме, одновременно собирая данные о их прогрессе. МЛ может анализировать собранные данные для определения наиболее эффективных реабилитационных программ для каждого конкретного пациента [11].

1.4. Применение ML в VR технологиях развлекательной индустрии

Развлекательная индустрия, особенно игровая, также активно исследует возможности, которые предоставляет сочетание VR и ML. Игры, использующие виртуальную реальность, могут стать более персонализированными и динамичными благодаря алгоритмам машинного обучения [12].

Современные игровые компании, такие как Valve и Epic Games, уже изучают возможности применения машинного обучения для создания более реалистичных и интерактивных игровых миров. Это может включать в себя улучшение графики, более сложные сценарии взаимодействия и реалистичное поведение окружающей среды. Такие инновации не только увеличивают привлекательность игр, но и поднимают планку ожиданий пользователей, что неизменно требует от разработчиков новых решений и технологий.

Виртуальные туры, использующие VR, также могут расширить свои возможности за счет применения методов машинного обучения. Например, анализируя пользовательские предпочтения и их поведение, системы могут генерировать персонализированные маршруты и рекомендации, улучшая общее впечатление от виртуального пейзажа [13].

2. Проблемы и сложности внедрения ML в VR технологиях

Несмотря на множество преимуществ, использование машинного обучения в виртуальной реальности сталкивается с рядом проблем и вызовов. Во-первых, для успешного применения ML требуется значительное количество данных для обучения моделей. Сбор и обработка таких данные может быть трудоемким и затратным процессом. Кроме того, качество данных напрямую влияет на результат — плохая или неполная информация может привести к неточным предсказаниям и негативному пользовательскому опыту [14].

Другой значимой проблемой является необходимость в мощных вычислительных ресурсах. Алгоритмы машинного обучения, особенно глубокие нейронные сети, требуют значительных объемов вычислительных мощностей, которые не всегда доступны в условиях VR. Для решения этой проблемы необходимо разрабатывать более эффективные алгоритмы и методы оптимизации, которые смогут работать в реальном времени с ограниченными ресурсами.

3. Перспективы применения машинного обучения в технологиях VR

Проведенный анализ показывает мощные перспективы применения методов машинного обучения в технологиях виртуальной реальности. Ожидается, что дальнейшее развитие методов машинного обучения будет способствовать созданию более сложных и адаптивных VR-приложений [15]. В частности, можно ожидать улучшения в области естественного языка и распознавания эмоций, что позволит создать более глубокое и эмоционально насыщенное взаимодействия между пользователями и виртуальными мирами.

С повышением уровня аппаратного обеспечения технологий VR откроются новые возможности для внедрения ML. Более мощные и доступные устройства позволят разработчикам реализовывать сложные алгоритмы, которые в настоящий момент могут быть недоступны. В результате можно ожидать появления новых форматов контента и способов взаимодействия, которые в корне изменят наше восприятие виртуальных пространств.

Заключение

Методы машинного обучения открывают новые возможности для технологий виртуальной реальности, значительно улучшают взаимодействие пользователей с виртуальными объектами и обогащая пользовательский опыт. Во всех сферах ML и VR работают в тандеме, предоставляя уникальные решения и формируя будущее этих технологий. Однако для эффективного внедрения требуется преодолеть объективные сложности, что создает новые направления инновационных исследований.

Литература

1. *Vince J.* Introduction to Virtual Reality / J. Vince. – National Centre for Computing Animation Bournemouth University, UK, 2004. – 163 p.

2. *Oyebola O.* AI in education: A review of personalized learning and educational technology / O. Oyebola et al // GSC Advanced Research and Review. – 2024. – URL: https://gsconlinepress.com/journals/gscarr/sites/default/files/GSCARR-2024-0062.pdf (дата обращения: 07.09.2024)

3. *Fathma F.* Virtual reality and machine learning for predicting visual attention in a daylit exhibition space: A proof of concept / F. Fathma et al // Ain Shams Engineering Journal. – V. 14, Iss. 6. – 2023. – URL: https://www.sciencedirect.com/science/article/pii/S2090447922004099 (дата обращения: 09.09.2024)

4. *Abinaya M.* Enhancing the Potential of Machine Learning for Immersive Emotion Recognition in Virtual Environment / M. Abinaya, G. Vadivu // EAI Endorsed Transactions on Scalable Information Systems. – 2024. – URL: https://publications.eai.eu/index.php/sis/article/view/5036/2871 (дата обращения: 11.09.2024)

5. *Song B.* Progress and Prospects in 3D Generative AI: A Technical Overview including 3D human / B. Song, L. Jie // 2024. – URL: https://arxiv.org/pdf/2401.02620 (дата обращения: 15.09.2024)

6. *Frank W.* Analysis of the Accuracy and Robustness of the Leap Motion Controller / W. Frank, B. Daniel, R. Bartholomäus, F. Denis // Physical Sensors. – 2013. – URL: https://arxiv.org/pdf/2401.02620 (дата обращения: 15.09.2024)

7. *Mariia H.* Virtual Reality Platform Using ML for Teaching Children with Special Needs / H. Mariia, A. Vasyl, C. Lyubomyr // 2020. – URL: https://ceur-ws.org/Vol-2631/paper16.pdf (дата обращения: 17.09.2024)

8. *Mahnaz S.* The Applications of Virtual Reality Technology in Medical Groups Teaching / S. Mahnaz et al // National Library of Medicine. – 2018. – URL: https://pmc.ncbi.nlm.nih.gov/articles/PM6039818/ (дата обращения: 18.09.2024)

9. *Jessica L.* The use of virtual reality technology in the treatment of anxiety and other psychiatric disorders / L. Jessica // Harv Rev Psychiatry. – 2017. – URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC5421394/ (дата обращения: 21.09.2024)

10. *Pinar T.* Virtual Reality in the Rehabilitation of Patients with Injuries and Diseases of Upper Extremities / T. Pinar et al // Healthcare (Basel). – 2022. – URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC9222955/ (дата обращения: 22.09.2024)

11. *Hamid B.* Use of Virtual Reality in Physical Therapy as an Intervention and Diagnostic Tool / B. Hamid et al // Rehabil Res Pract. – 2024. – URL: https://pubmed.ncbi.nlm.nih.gov/38304610/ (дата обращения: 22.09.2024)

12. *Wang X.* Research on Application of Artificial Intelligence in VR Games / X. Wang // Frontiers in Artificial Intelligence and Applications. – Volume 331: Fuzzy Systems and Data Mining VI. – 2024. – URL: https://www.researchgate.net/publication/346805454_Research_on_Application_of_Artificial_Intelligence_in_VR_Games (дата обращения: 22.09.2024)

13. *Chourouk O.* The Impact of Virtual Reality (VR) Tour Experience on Tourists' Intention to Visit / O. Chourouk // Digital Economy and Management. – 2023. – URL: https://www.mdpi.com/2078-2489/14/10/546 (дата обращения: 22.09.2024)

14. *Tayebeh B.* Challenges and Practical Considerations in Applying Virtual Reality in Medical Education and Treatment / B. Tayebeh, M. Seyed, M. Niloofar // Oman Med J. – 2020. – URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC7232669/ (дата обращения: 23.09.2024)

15. *I Gede P. S.* Systematic Literature Review of Machine Learning in Virtual Reality and Augmented Reality / Partha Sindu I Gede [et al] // Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI). – 2023. – URL: https://www.researchgate.net/publication/372094412_Systematic_Literature_Review_of_Machine_Learning_in_Virtual_Reality_and_Augmented_Reality (дата обращения: 24.09.2024)

УДК 004.942

МУЛЬТИМОДЕЛЬНЫЙ ПОДХОД К РАЗРАБОТКЕ СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ КЛАССИФИКАЦИИ СНИМКОВ ОПТИЧЕСКОЙ КОГЕРЕНТНОЙ ТОМОГРАФИИ

А. А. Арзамасцев[1,2], О. Л. Фабрикантов[2], Н. А. Зенкова[3], Е. В. Кулагина[2]

[1]Воронежский государственный университет
[2]Тамбовский филиал Федерального государственного автономного учреждения Национальный медицинский исследовательский центр «Межотраслевой научно-технический комплекс «Микрохирургия глаза» имени академика С. Н. Федорова
[3]Тамбовский государственный университет имени Г.Р. Державина

Аннотация. Оптическая когерентная томография (ОКТ) — современный метод выявления патологии сетчатки глаза и преретинальных слоев стекловидного тела. Однако описание и интерпретация результатов исследования требуют высокой квалификации и специальной подготовки врача-офтальмолога, значительных временных затрат врача и пациента. Целью данной работы является разработка мультимодельного подхода классификации снимков ОКТ, имитирующего в общих чертах последовательность действий врача-офтальмолога. Предложен общий алгоритм метода, базируемый на последовательном глубоком обучении и использовании девяти различных моделей сверточных нейронных сетей (ИНС-моделей), обученных на уникальных датасетах. Осуществлен выбор оптимальных архитектур ИНС-моделей) и их гиперпараметров. Проведено сравнение результатов вычислительных экспериментов, проведенных с использованием средств Python в Google Colaboratory при одномодельном и нескольких этапов многомодельного подхода. Получены оценки точности классификации.
Ключевые слова: глубокое машинное обучение, классификация, оптическая когерентная томография, модели искусственного интеллекта, сверточные нейронные сети, мультимодельный подход.

Введение

Оптическая когерентная томография (ОКТ) является современным высокотехнологичным и информативным методом выявления патологии сетчатки глаза и преретинальных слоев стекловидного тела [1] в офтальмологии. Однако описание и интерпретация результатов исследования требуют высокой квалификации и специальной подготовки врача, значительных временных затрат офтальмолога и пациента. По этой причине решение задач, связанных с автоматизацией процесса классификации снимков ОКТ, является актуальным.

Вместе с этим, в настоящее время наблюдается быстрое развитие компьютерного инструментария и технологий, позволяющих создавать системы искусственного интеллекта на основе нейронных сетей различной архитектуры и назначения [2–4].

В наших работах [5, 6], опубликованных в материалах предыдущих конференций, мы описали компьютерные методы анализа стекловидного тела, выделения и аппроксимации границы сетчатки, определения кривизны границы, расчетов средней толщины сетчатки и др., выполненных в том числе и с использованием искусственных нейронных сетей (ИНС). В работе [7] авторами опубликованы результаты классификации с использованием одномодельной схемы, а также впервые сформулированы основные этапы мультимодельного подхода классификации снимков ОКТ, имитирующего в общих чертах последовательность действий врача-офтальмолога. В данном докладе, являющимся логическим продолжением работ [5–7], приведены результаты, полученные при классификации снимков ОКТ с использованием мультимодель-

ного подхода, базируемого на последовательном применении девяти различных моделей сверточных нейронных сетей (ИНС-моделей), обученных на уникальных датасетах.

Цели данной работы. Разработать оптимальные архитектуры ИНС-моделей на основе глубокого обучения сверточных искусственных нейронных сетей, также значения гиперпараметров для классификации снимков ОКТ с использованием библиотек Python Keras и Tensorflow в Google Colaboratory. Сравнить результаты классификации снимков при использовании одномодельного и мультимодельного подходов.

Материалы и методы

Предлагаемый способ классификации должен различать следующие 14 видов снимков ОКТ: норма, кистозный макулярный отек (КМО), диффузный макулярный отек (ДМО), отслойка нейроэпителия (ОНЭ), отслойка пигментного эпителия (ОПЭ), твердые экссудаты (ТЭ), эпиретинальный фиброз (ЭРФ), эпиретинальная мембрана (ЭРМ), друзы пигментного эпителия (ДПЭ), ламеллярный макулярный разрыв (ЛМР), сквозной макулярный разрыв (СМР), задняя отслойка стекловидного тела (ЗОСТ), витреомакулярная тракция (ВМТ), витреомакулярная адгезия (ВМА). Первоначальный датасет для обучения ИНС-моделей для первых трех этапов технологии представлял собой обезличенные снимки ОКТ реальных пациентов и включал 2801 изображения, полученных непосредственно с томографа DRI OCT Triton 3D optical coherence tomography в разрешении 1920 × 969 × 24 BPP в виде файлов формата .JPG. В ходе предварительной обработки из всех снимков автоматизировано были выделены целевые фрагменты в формате 1024 × 512 × 24 BPP. Весь датасет был разделен опытными врачами — офтальмологами на части следующим образом. Для обучения ИНС-моделей первого этапа 1405 изображений, для валидации — 200 изображений. Для обучения ИНС-моделей второго этапа 599 изображений, для валидации — 100 изображений. Для обучения ИНС-моделей третьего этапа 447 изображений, для валидации — 50 изображений.

Количество изображений каждого класса соответствовало частоте встречаемости патологии у пациентов.

При обучении ИНС-моделей использовали следующие параметры компиляции: оптимизатор Adam — один из самых эффективных алгоритмов оптимизации, использующий при адаптации скорости обучения параметров первый и второй моменты градиента; функция потерь categorical_crossentropy — категориальная перекрестная энтропия, метрика accuracy — доля правильных ответов алгоритма. Все технологические процессы с моделями проводили с использованием средств языка Python в Google Colaboratory.

Полученные результаты и их обсуждение

На рис. 1 представлена общая технология классификации снимков ОКТ в соответствии с мультимодельным подходом. Она, по сути, имитирует процесс идентификации снимков опытным врачом-офтальмологом. Технология включает девять этапов, на каждом из которых работает специальная ИНС-модель, предназначенная для решения одной из узких проблем. Технический результат достигается тем, что с помощью глубоких сверточных нейронных сетей с использованием сканов ОКТ, содержащих изображения различных патологических структур, последовательно дифференцируют наличие, по меньшей мере, 13 основных классов патологий сетчатки и витреомакулярного интерфейса, представленных в табл. 1. Для проведения обучения ИНС-моделей используют исходный набор сканов ОКТ нормы и патологий, приведенный в табл. 1. Из них формируют датасеты для каждой модели.

Первый датасет предназначен для обучения ИНС-модели 1, позволяющей классифицировать норму и патологию, состоит из двух массивов сканов: сканов нормы (класс 1) и сканов, указанных в табл. 1 тринадцати патологий (классы 2–14).

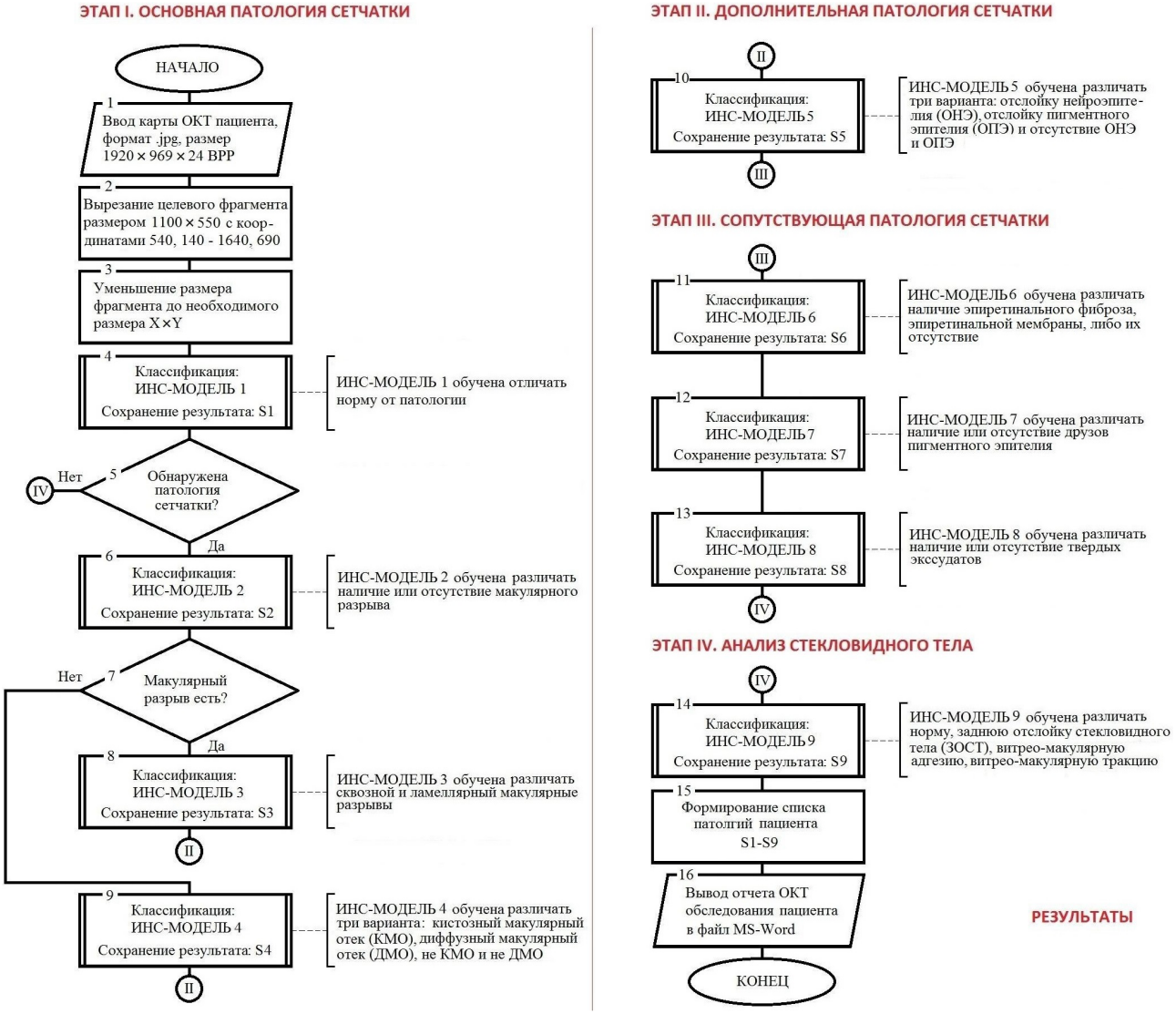


Рис. 1. Общая блок-схема реализации мультимодельного подхода классификации снимков ОКТ

Таблица 1

Классы нормы и патологий для формирования датасетов

Номер класса	Название класса
1	Норма
2	Кистозный макулярный отек (КМО)
3	Диффузный макулярный отек (ДМО)
4	Отслойка нейроэпителия (ОНЭ)
5	Отслойка пигментного эпителия (ОПЭ)
6	Твердые экссудаты (ТЭ)
7	Эпиретинальный фиброз (ЭРФ)
8	Эпиретинальная мембрана (ЭРМ)
9	Друзы пигментного эпителия (ДПЭ)
10	Ламеллярный макулярный разрыв (ЛМР)
11	Сквозной макулярный разрыв (СМР)
12	Задняя отслойка стекловидного тела (ЗОСТ)
13	Витреомакулярная тракция (ВМТ)
14	Витреомакулярная адгезия (ВМА)

Второй датасет предназначен для обучения ИНС-модели, позволяющей выявить наличие или отсутствие МР, состоит из двух массивов сканов: первый массив — сканы с СМР и ЛМР (классы 10,11) и второй массив — сканы указанных в таблице 1 патологий (классы 1–9 и 12–14).

Третий датасет предназначен для обучения ИНС-модели, которая при наличии МР позволяет определить, имеет ли место СМР или ЛМР и состоит из двух массивов сканов: первый массив — сканы с СМР (класс 11), второй массив — сканы с ЛМР (класс 10).

Четвертый датасет предназначен для обучения ИНС-модели, позволяющей определить наличие, либо отсутствие КМО либо ДМО, состоит из трех массивов сканов: первый массив — сканы с КМО (класс 2), второй массив — сканы с ДМО (класс 3) и третий массив — сканы указанных в таблице 1 патологий классов 4–9 и 12–14).

Пятый датасет предназначен для обучения ИНС-модели, способной классифицировать три варианта — ОНЭ, ОПЭ, либо отсутствие ОНЭ и ОПЭ, состоит из трех массивов сканов: первый массив — сканы с ОНЭ (класс 4), второй массив — сканы с ОПЭ (класс 5) и третий массив — сканы указанных в таблице 1 патологий (классы 2, 3 и 6–14).

Шестой датасет предназначен для обучения ИНС-модели, позволяющей выявить наличие или отсутствие ЭРФ либо ЭРМ, состоит из трех массивов сканов: первый массив — сканы ЭРФ (класс 7), второй массив — сканы с ЭРМ (класс 8) и третий массив — сканы указанных в таблице 1 патологий (классы 2–6 и 9–14).

Седьмой датасет предназначен для обучения ИНС-модели, способной классифицировать наличие, либо отсутствие ДПЭ, состоит из двух массивов сканов: первый массив — сканы с ДПЭ (класс 9) и второй массив — сканы указанных в таблице 1 патологий (классы 2–8 и 10–14).

Восьмой датасет предназначен для обучения ИНС-модели, способной классифицировать наличие, либо отсутствие ТЭ, состоит из двух массивов сканов: первый массив — сканы с ТЭ (класс 6) и второй массив — сканы указанных в таблице 1 патологий (классы 2–5 и 7–14).

Девятый датасет предназначен для обучения ИНС-модели, способной классифицировать норму, ЗОСТ, ВМА и ВМТ, состоит из четырех массивов сканов: первый массив —сканы нормы (класс 1), второй массив — сканы с ЗОСТ (класс 12), третий массив — сканы с ВМА (класс 14) и четвертый массив — сканы с ВМТ (класс 13).

В настоящее время осуществлена реализация трех первых этапов схемы, показанной на рис. 1. Все технологические процессы с моделями проводили с использованием средств языка Python в Google Colaboratory. Одновременно с обучением ИНС-моделей проводили практическую оптимизацию гиперпараметров сверточных нейронных сетей, важнейшими из которых являлось число сверточных слоев, количество и размер фильтров на каждом слое, определяющие общее число оптимизируемых параметров. На рис. 2 и 3 представлены примеры трехслойной сверточной сети, которая принималась в качестве стартовой при оптимизации.

На рис. 4 a), b) показан процесс обучения одной из ИНС-моделей. Видно, что принятая архитектура и число оптимизируемых параметров хорошо соответствуют задаче, так что функция потерь быстро убывает практически до нулевых значений, в то время как точность на обучении и валидации достигает в конце процесса 100 %.

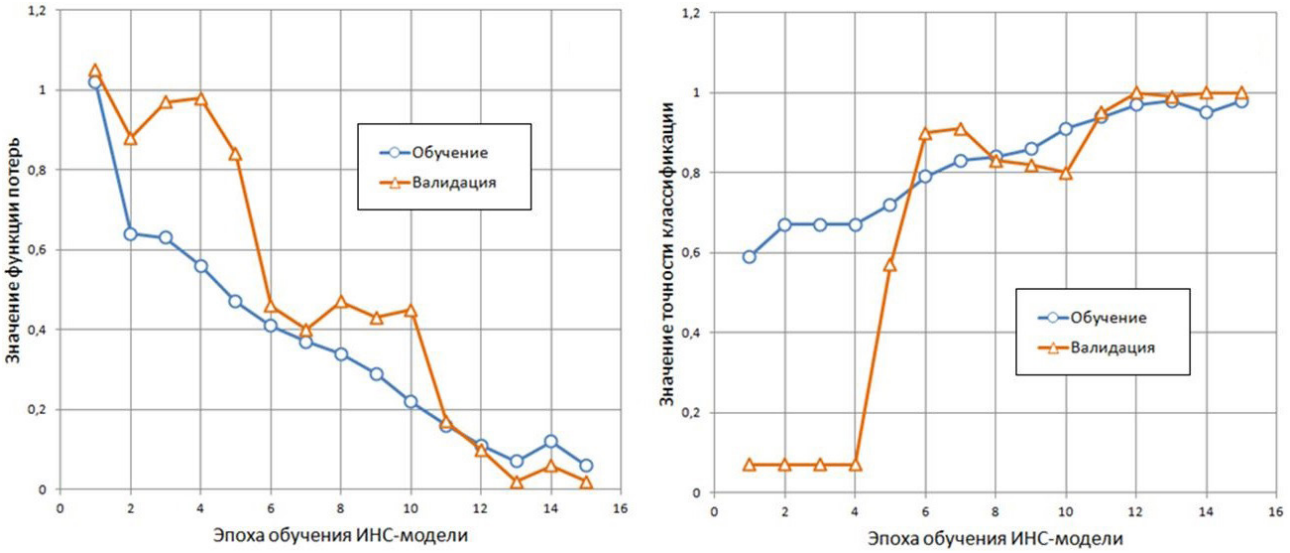
В ходе исследования проверяли модели с тремя–семью сверточными слоями для всех трех этапов технологии. При этом число эпох обучения из-за значительных затрат времени было ограничено 25 эпохами. Результаты показаны в табл. 2, из анализа которой можно сделать общее заключение о том, что архитектура сети и значения гиперпараметров, некоторые из которых показаны на рис. 2, хорошо соответствуют поставленной задаче и датасетам снимков патологий и нормы, так как точность обучения и валидации во всех случаях была равна 100 % или близка к этому значению. Однако на тестовой выборке, состоящей из новых снимков, которые в процессе обучения моделей не участвовали (test2), точность заметно ниже. Этот параметр позволяет определить оптимальное число сверточных слоев для ИНС-моделей всех этапов. Этот параметр зависит от специфичности снимков датасета.

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(4, (9,9), activation='relu', input_shape=(IMG_SHAPE, IMG_SHAPE, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),

    tf.keras.layers.Conv2D(8, (5, 5), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),

    tf.keras.layers.Conv2D(16, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),

    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(225, activation='relu'),
    tf.keras.layers.Dense(2, activation='softmax')
])
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

Рис. 2. Архитектура и параметры модели искусственной нейронной сети с тремя свёрточными слоями и параметры ее компиляции. Первая цифра в Conv2D — число используемых фильтров в слое свёртки, две следующие цифры — размер фильтра в пикселях. Активационные функции нейронов сети — relu, в выходном слое классификации — softmax. Первая цифра в полносвязном слое Dense — число нейронов

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 504, 504, 4)       976

 max_pooling2d (MaxPooling2  (None, 252, 252, 4)       0
 D)

 conv2d_1 (Conv2D)           (None, 248, 248, 8)       808

 max_pooling2d_1 (MaxPoolin  (None, 124, 124, 8)       0
 g2D)

 conv2d_2 (Conv2D)           (None, 122, 122, 16)      1168

 max_pooling2d_2 (MaxPoolin  (None, 61, 61, 16)        0
 g2D)

 flatten (Flatten)           (None, 59536)             0

 dense (Dense)               (None, 225)               13395825

 dense_1 (Dense)             (None, 2)                 452

=================================================================
Total params: 13399229 (51.11 MB)
Trainable params: 13399229 (51.11 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Рис. 3. Параметры ИНС-модели, представленной на рис. 2



а) *б)*

Рис. 4. Пример обучения и валидации ИНС-модели с тремя сверточными слоями для первого этапа: a) график уменьшения функции потерь, b) график увеличения точности

Так, на этапе 1 оптимальной является ИНС-модель, содержащая 5 сверточных слоев, обеспечивающая 87 % верных ответов на новой выборке. Видно, что зависимость точности от числа сверточных слоев имеет максимум, приходящийся на пять. Сети с меньшим числом сверточных слоев имеют большее число оптимизируемых параметров, показывают хорошие результаты при обучении и валидации, но их результаты на новых данных существенно ниже.

Таблица 2

Результаты обучения, валидации и проверки на дополнительных тестовых выборках ИНС-моделей первого — третьего этапов

Этап	Число сверточных слоев	Эпох обучения	Точность обучения, %	Точность валидации, %	Верных ответов (test1), %	Верных ответов (test2), %	Число оптимизируемых параметров
Этап 1	3	18	100	100	100	77	13399229
	4	19	100	100	100	77	6063469
	5	25	98	99	97	87	1234317
	6	25	96	97	98	76	396013
	7	25	95	97	51	39	87341
Этап 2	3	11	100	100	100	64	13399229
	4	9	100	100	100	68	6063469
	5	15	100	100	100	86	1234317
	6	14	100	100	100	85	396013
	7	13	100	100	100	84	87341
Этап 3	3	7	100	100	100	100	13399229
	4	13	100	100	100	100	6063469
	5	9	100	100	100	100	1234317
	6	11	100	100	100	100	396013
	7	7	100	100	100	100	87341

Можно сделать вывод о необходимости увеличения размера обучающей выборки и добавления в неё большего разнообразия снимков, являющихся нормой и тринадцати вариантов других патологий. Также видно, что при увеличении числа сверточных слоев количество оптимизируемых параметров уменьшается, и низкие результаты в этом случае можно связать с этим фактором.

Аналогичный оптимум для ИНС-моделей этапа 2 также существует — это также нейронная сеть с пятью сверточными слоями.

Для третьего этапа все модели показали практически одинаковые стопроцентные результаты. Данное обстоятельство может быть связано, как с малым количеством снимков в датасете, так и с их «простотой», заключающейся в близком соответствии изображений патологий.

Отметим, что полученные модели и их исследование (табл. 2) несколько улучшают результаты использования в аналогичной ситуации одномодельного подхода, где наилучшая точность классификации достигала 85 % [7].

Заключение

Таким образом, в докладе приведены результаты реализации первых этапов многомодельного подхода классификации снимков ОКТ в офтальмологии. Определены гиперпараметры ИНС-моделей: число сверточных слоев, размеры и количество фильтров. Вычислительные эксперименты показали удовлетворительную точность классификации, что позволяет надеяться на реализуемость подхода и его использовании в системах поддержки принятия решений в области офтальмологии.

Благодарности

Работа выполнена в соответствии с договором о научно-техническом сотрудничестве Воронежского государственного университета и Федерального государственного автономного учреждения «Национальный медицинский исследовательский центр «Межотраслевой научно-технический комплекс «Микрохирургия глаза» имени академика С.Н. Федорова», Тамбовского филиала от 28.11.2022.

Литература

1. Оптическая когерентная томография сетчатки / Под ред. Дж. С. Дакера, Н. К. Вэхид, Д. Р. Голдмана; пер. с англ. под ред. А. Н. Амирова. – 3-е изд. – М. : МЕДпресс-информ, 2021. – 192 с.

2. *Будума Н.* Основы глубокого обучения. Создание алгоритмов для искусственного интеллекта следующего поколения / Нихиль Будума, Николас Локашо ; пер. с англ. А. Коробейникова; [науч. ред. А. Созыкин]. – М. : Манн, Иванов и Фербер, 2020. – 304 с.

3. *Фостер Д.* Генеративное глубокое обучение. Творческий потенциал нейронных сетей / Д. Фостер – СПб. : Питер, 2020. – 336 с.

4. *Постолит А. В.* Основы искусственного интеллекта в примерах на Python. – СПб. : БХВ-Петербург, 2021. – 448 с.

5. *Арзамасцев А. А.* Автоматизированный анализ протоколов оптической когерентной томографии сетчатки глаза с использованием системы искусственного интеллекта / А. А. Арзамасцев, О. Л. Фабрикантов, Н. А. Зенкова, Е. В. Кулагина // Актуальные проблемы прикладной математики, информатики и механики : сборник трудов Международной научной конференции, Воронеж, 13–15 декабря 2021 г. – Воронеж : Издательство «Научно-исследовательские публикации», 2022. – С. 823–827.

6. *Арзамасцев А. А.* Алгоритмы глубокого машинного обучения в решении задач автоматизированной обработки данных в офтальмологии / А. А. Арзамасцев, О. Л. Фабрикантов, Е. В. Кулагина, С. В. Беликов, Н. А. Зенкова // Актуальные проблемы прикладной математики, информатики и механики : сборник трудов Международной научной конференции, Воронеж, 12–14 декабря 2022 г. – Воронеж : Издательство «Научно-исследовательские публикации», 2023. – С. 285–293.

7. *Арзамасцев А. А., Фабрикантов О. Л., Кулагина Е. В., Зенкова Н. А.* Классификация снимков оптической когерентной томографии с использованием методов глубокого машинного обучения // Digital Diagnostics. – 2024. – Т. 5, № 1. – С. 5–16. DOI: https://doi.org/10.17816/DD623801

УДК 004.942

МОДЕЛИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ ПРЕДОПЕРАЦИОННОГО РАСЧЕТА ИНТРАОКУЛЯРНЫХ ЛИНЗ

А. А. Арзамасцев[1,2], О. Л. Фабрикантов[2], Н. А. Зенкова[3], А. А. Чикина[2]

[1]Воронежский государственный университет
[2]Тамбовский филиал Национального медицинского исследовательского центра «Межотраслевой научно-технический комплекс «Микрохирургия глаза» имени академика С. Н. Федорова
[3]Тамбовский государственный университет имени Г. Р. Державина

Аннотация. Для предоперационного расчета интраокулярных линз (ИОЛ) в офтальмологии используются формулы первого–третьего поколений, такие как SRK II, SRK/T, Hoffer-Q, Holladay II, Haigis, Barrett. С развитием систем искусственного интеллекта связано появление современных формул четвертого поколения, таких как Barrett Universal II, Hill-RBF, Kane и Pearl DGS. Главным недостатком первой группы формул является их низкая точность (средняя относительная погрешность на значительных массивах эмпирических данных составляет 11–15 %), что снижает процент попаданий расчетных значений в необходимый для офтальмологов диапазон ±0,5 дптр до 13–14 %. Формулы четвертого поколения имеют существенно меньшую среднюю относительную погрешность (2,7–3,2 %), что позволяет получить больший процент попаданий в диапазон ±0,5 дптр (55–67%). Данные формулы представлены в сети в виде специализированных калькуляторов, их структура и коэффициенты скрыты от пользователей, что исключает возможность анализа влияния и взаимовлияния входных показателей на оптическую силу ИОЛ, а также их использование в системах поддержки принятия решений в области офтальмологии. В данной работе авторы продолжили серию исследований, посвященных разработке систем искусственного интеллекта для расчета оптической силы интраокулярных линз на основе моделей искусственных нейронных сетей (ИНС-моделей). В докладе приведены: авторские разработки систем искусственного интеллекта на основе ИНС-моделей для расчета ИОЛ, внедренные в офтальмологическую практику, а также их сравнение с моделями, полученными в результате аппроксимации данных, полученных с калькуляторов формул Barrett Universal II, Hill-RBF, Kane и Pearl DGS. На значительных по объему локальных датасетах разработанные модели незначительно превосходят известные формулы, однако, их открытость и возможность переучивания по мере накопления новых данных открывают значительную перспективу использования в системах поддержки принятия решений в области офтальмологии.

Ключевые слова: модели искусственного интеллекта, искусственные нейронные сети, глубокое машинное обучение, оптическая сила интраокулярной линзы формулы расчета ИОЛ, формулы четвертого поколения.

Введение

Имплантация современных интраокулярных линз (ИОЛ) позволяют офтальмологам успешно решать задачи по хирургическому лечению катаракты. При этом улучшение зрительных функций пациента напрямую связано с точностью предоперационного расчета оптической силы ИОЛ. Повышение точности таких расчетов представляет собой актуальную проблему, для решения которой в последнее время используют системы искусственного интеллекта на основе искусственных нейронных сетей (ИНС-модели).

Наиболее точными методиками расчета ИОЛ, используемыми в настоящее время в мировой офтальмологической практике являются формулы Barrett Universal II, Hill-RBF, Kane и Pearl DGS (формулы четвертого поколения), полученные, по утверждению авторов, с использованием систем искусственного интеллекта [1–4]. Они представлены в сети в виде специа-

лизированных калькуляторов, их структура и коэффициенты скрыты от пользователей, что исключает возможность анализа влияния и взаимовлияния входных показателей на оптическую силу ИОЛ, а также их использование в системах поддержки принятия решений в области офтальмологии.

Имеется несколько современных работ, посвященных анализу точности этих формул и некоторых систем искусственного интеллекта.

Так работа [5] посвящена исследованию использования машинного обучения для прогнозирования возникновения послеоперационной рефракции после операции по удалению катаракты и сравнение точности этого метода с традиционными формулами расчета оптической силы интраокулярной линзы. При этом коэффициенты формул расчета оптической силы ИОЛ и модели машинного обучения были оптимизированы с использованием обучающих выборок, т.е. адаптированы к локальным данным. Были оценены результаты, полученные по формулам SRK/T, Haigis, Holladay 1, Hoffer Q и Barrett Universal II. Показано, что среди традиционных формул Barrett Universal II имела самую низкую среднюю и медианную абсолютную ошибку прогноза. Поэтому авторы сравнили точность моделей машинного обучения с точностью Barrett Universal II. Абсолютные ошибки некоторых методов машинного обучения были ниже, чем у Barrett Universal II. Однако статистически значимой разницы не наблюдалось. Авторы сделали вывод, что точность методов машинного обучения не уступала точности формулы Barrett Universal II.

В обзоре [6] оценивали точность формул для расчета оптической силы ИОЛ, основанных на искусственном интеллекте на основе анализа статей, опубликованных с 2017 по июль 2023 года. Всего было рассмотрено 25 рецензируемых статей на английском языке с максимальной выборкой и наибольшим количеством сравниваемых формул. Для оценки точности формул использовались оценки средней абсолютной ошибки и процента пациентов, для которых рефракционная ошибка расчета попадала в диапазоны ±0,5 дптр и ±1,0 дптр. В большинстве случаев формула Kane позволила получить наименьшую среднюю абсолютную ошибку и самый высокий процент пациентов в указанных диапазонах. Второе место обычно занимала формула Pearl DGS.

В наших предыдущих работах [7–14] были получены следующие результаты в области разработки методов предоперационного расчета оптической силы ИОЛ с использованием ИНС-моделей:

– показана генерализуемость эмпирических данных, полученных от большого числа пациентов офтальмологической клиники в рамках одной ИНС-модели, сделан вывод об оптимальной архитектуре такой модели, полученной на основе разложения функции многих переменных в ряд Тейлора и реализованной с использованием машинного обучения искусственных нейронных сетей средствами Python в Google Colaboratory;

– проведена оценка значимости входных параметров;

– предложена концепция и ее реализация для динамически развивающейся системы искусственного интеллекта, переобучаемой по мере поступления новых эмпирических данных о пациентах;

– реализован способ расчета оптической силы ИОЛ на основе системы искусственного интеллекта, внедренный в офтальмологическую практику.

Однако, проблема повышения точности ИНС-моделей и повышения процента попаданий расчетных значений оптической силы ИОЛ в диапазон ±0,5 дптр по-прежнему остается.

Целью данной работы является дальнейшее совершенствование ИНС-моделей для системы искусственного интеллекта расчета оптической силы ИОЛ, их сравнение с калькуляторами четвертого поколения и изучение влияния на целевой показатель главных входных факторов.

Материалы и методы

При разработке ИНС-модели использовали датасеты, состоящие из 904 (890) обезличенных записей пациентов Тамбовского филиала МНТК «Микрохирургия глаза» им. академика С. Н. Федорова. Для получения аппроксимационных моделей формул Barrett Universal II, Hill-RBF, Kane, Pearl DGS использовали по 400 записей, полученных с online-калькуляторов [1–4] для реальных входных параметров. Для доказательства стационарности полученных результатов все расчеты были сделаны сначала на датасете в 150 записей, а затем для всех 400 записей. Использование такой технологии было необходимо для подтверждения репрезентативности выборки ввиду небольшого количества записей. Сравнение результатов показало их полную идентичность. Для обучения ИНС-моделей использовали градиентные методы спуска, а также безградиентные — покоординатного спуска и Монте-Карло, реализованные средствами Python в Google Colaboratory. Оценку точности ИНС-моделей проводили по средней относительной погрешности и проценту попадания расчетных значений оптической силы ИОЛ в диапазон ±0,5 дптр.

Полученные результаты и их обсуждение

Оптимальная архитектура ИНС-моделей была получена в наших предыдущих работах по генерализуемости данных на основе разложения функции многих переменных в ряд Тейлора [7, 8, 11]. Эта архитектура показана на рис. 1 и соответствует формуле:

$$Y = w_{19}\sum_{i=1}^{6} w_i x_i + \left(w_{20}\sum_{i=1}^{6} w_{i+6} x_i \right)^2 + \left(w_{21}\sum_{i=1}^{6} w_{i+12} x_i \right)^3 + w_{22},$$

где символом w обозначены весовые коэффициенты, значения которых определяются в результате решения оптимизационной задачи на основе датасетов эмпиричес ких данных.

Получены следующие показатели точности ИНС-модели и аппроксимационных моделей известных офтальмологических формул (табл. 1). Таким образом, полученная на локальных данных ИНС-модель имеет меньшие значения средней относительной и максимальной относительной погрешностей по сравнению с аппроксимационными моделями известных формул четвертого поколения. Процент попадания расчетных значений оптической силы ИОЛ в диапазон ±0,5 дптр у ИНС-модели лучше, чем у формул Hill-RBF, Kane, Pearl DGS и сопоставим с формулой Barrett Universal II. Данная ИНС-модель оформлена патентом РФ на способ расчета оптической силы ИОЛ [14] и внедрена в офтальмологическую практику в виде специализированного калькулятора системы поддержки принятия решений врачей-офтальмологов.

После получения адекватной ИНС-модели оценивали влияние входных факторов — константы А, глубины передней камеры, длины глаза, рефракции сильного и слабого меридианов, толщины линзы, на величину оптической силы ИОЛ по различным моделям. Вычислительный эксперимент проводили таким образом, чтобы получить однопараметрические зависимости от каждого из входных факторов при средних значениях других факторов. Такие зависимости полезны для врачей-офтальмологов и позволяют понять направление и силу влияния каждого входного фактора на целевой показатель (рис. 2–4).

Результаты данных вычислительных экспериментов, обсуждаемые в докладе, свидетельствуют о близости оценок оптической силы ИОЛ, полученным по различным моделям в диапазонах их применимости.

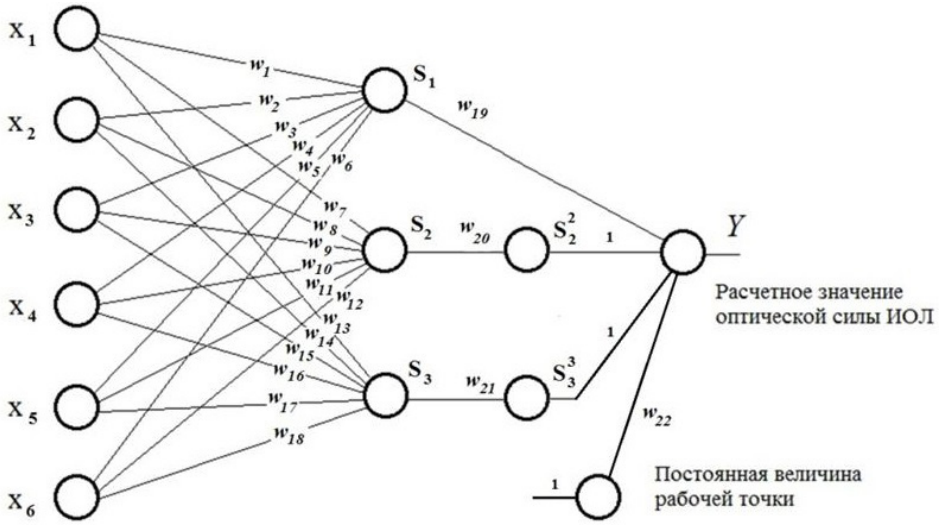


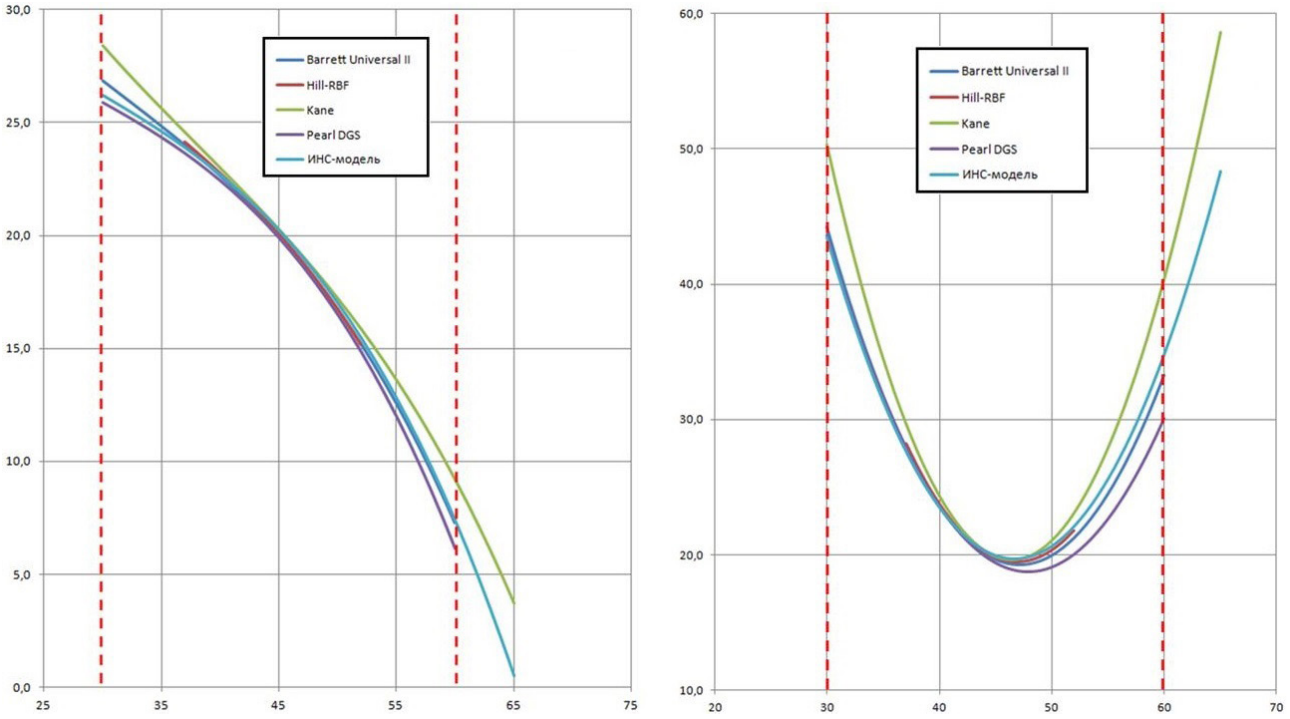
Рис. 1. Схема ИНС-модели

Таблица 1

Результаты расчетов оптической силы ИОЛ для ИНС-модели и аппроксимационных моделей формул четвертого поколения на локальных данных

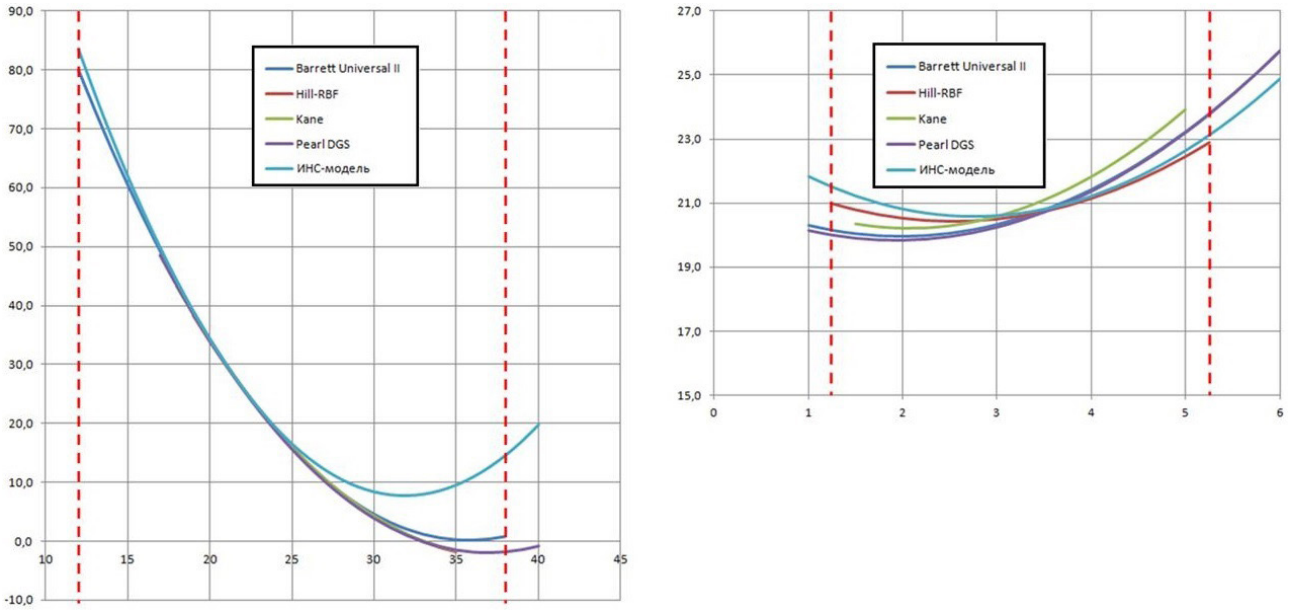
ИНС-модель или формула, год опубликования	Входные параметры	Средняя относительная погрешность расчетов, %	Попаданий расчетной величины в диапазон ±0,5 дптр, %	Максимальная относительная погрешность расчетов, %	Примечание
ИНС-модель, 2024	A, ACD, AL, K1, K2, LT	2,33	67,0	44	Результаты получены на локальных данных (890 записей)
Barrett Universal II, 2019, [1]	A, ACD, AL, K1, K2, LT	2,67	67,8	64,3	Результаты получены на локальных данных (400 записей)
Hill-RBF, 2020, [2]	A, ACD, AL, K1, K2, LT	2,75	63,0	71,4	Результаты получены на локальных данных (400 записей)
Kane, 2017, [3]	A, ACD, AL, K1, K2, LT	2,78	55,3	71,4	Результаты получены на локальных данных (400 записей)
Pearl DGS, 2021, [4]	A, ACD, AL, K1, K2, LT	3,21	64,5	71,4	Результаты получены на локальных данных (400 записей)

A — константа, модель ИОЛ; ACD — глубина передней камеры, мм; AL — длина глаза, мм; K1 — рефракция слабого меридиана, дптр; K2 — рефракция сильного меридиана, дптр; LT — толщина линзы, мм.



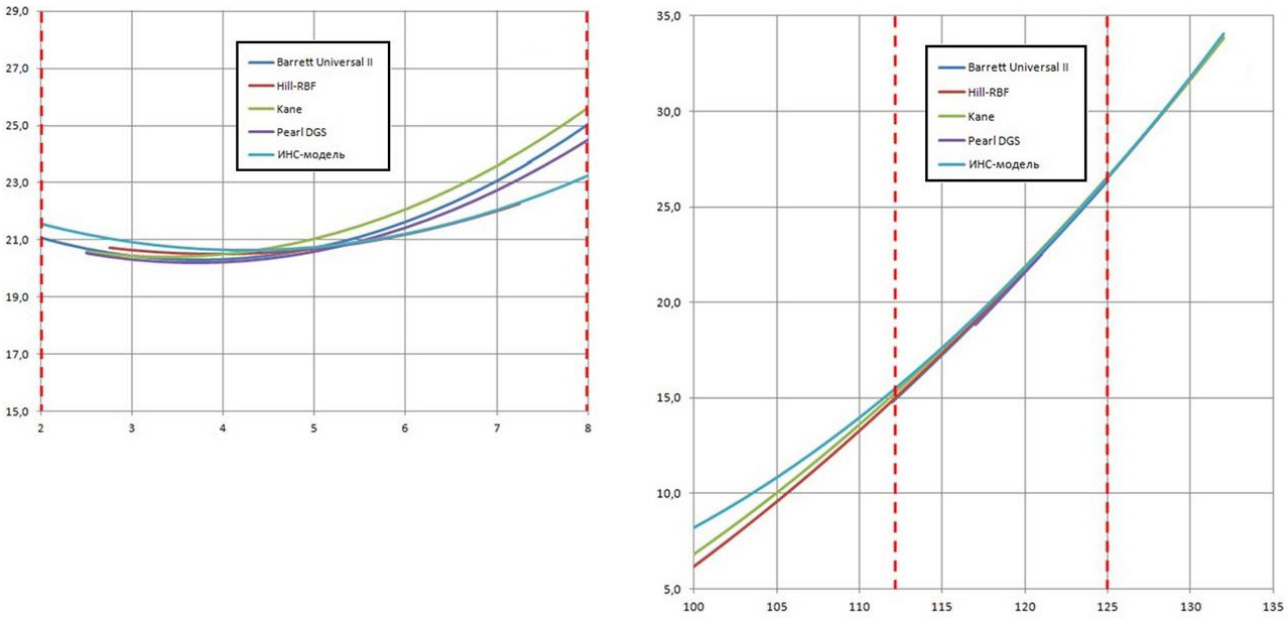
а) *б)*

Рис. 2. Зависимость оптической силы ИОЛ от рефракции сильного меридиана (dптр) — a) и рефракции слабого меридиана (dптр) — b) по различным формулам и ИНС-модели. Красная пунктирная линия — ограничения по формуле Barrett Universal II



а) *б)*

Рис. 3. Зависимость оптической силы ИОЛ от оптической длины глаза (мм) — a) и от глубины передней камеры (мм) — b) по различным формулам и ИНС-модели. Красная пунктирная линия — ограничения по формуле Barrett Universal II



а) *б)*

Рис. 4. Зависимость оптической силы ИОЛ от толщины хрусталика (мм) — a) и от модели ИОЛ (коэффициент A) — b) по различным формулам и ИНС-модели. Красная пунктирная линия — ограничения по формуле Barrett Universal II

Заключение

Таким образом, в данной работе показано, что способность моделей на основе искусственных нейронных сетей к хорошей генерализации офтальмологических данных открывает возможности создания интеллектуальной экспертной системы с динамическим поступлением новых данных и многократным глубоким машинным обучением интеллектуального ядра. Основной особенностью такой системы по сравнению с использованием традиционных формул расчета является ее адаптивность, позволяющая решать проблемы нестационарности объекта и локализации вследствие наличия в ней обратной связи. В случае, если для пациента известны не все данные, возможно также построить ИНС-модель и с использованием обучающей выборки меньшего размера или заменить недостающие данные средними или прогнозируемыми.

Благодарности

Работа выполнена в соответствии с договором о научно-техническом сотрудничестве Воронежского государственного университета и Федерального государственного автономного учреждения «Национальный медицинский исследовательский центр «Межотраслевой научно-технический комплекс «Микрохирургия глаза» имени академика С. Н. Федорова», Тамбовского филиала от 28.11.2022.

Литература

1. Barrett Universal II Formula. [Last cited on 2019 Feb 22]. Available from: – URL: https://www.apacrs.org/barrett_universal2/ (дата обращения: 04.07.2024).

2. Hill-RBF calculator version 2.0. – URL: https://rbfcalculator.com/ online/index.html (дата обращения: 04.07.2024).

3. Kane formula. – URL: https://www.iolformula.com/about/ (дата обращения: 04.07.2024).

4. *Debellemaniere G.* The PEARL-DGS formula: the development of an open-source machine learning-based thick IOL calculation formula / G. Debellemaniere, M. Dubois, M. Gauvin, A. Wallerstein, F. Brenner Luis, R. Rampat, A. Saad, D. Gatinel // American Journal of Ophthalmology. – 2021. – May 13: S0002-9394(21)00276-2. doi: 10.1016/j.ajo.2021.05.004.

5. *Yamauchi T.* Use of a machine learning method in predicting refraction after cataract surgery / T Yamauchi., T Tabuchi., K. Takase, H Masumoto // Journal of Clinical Medicine. – 2021. – 10, 1103. doi: 10.3390/jcm10051103

6. *Stopyra W.* Review of intraocular lens power calculation formulas based on artificial intelligence / W. Stopyra, D. L. Cooke, A. A. Grzybowski // Journal of Clinical Medicine. – 2024. – 13, 498. https://doi.org/10.3390/jcm13020498

7. *Арзамасцев А. А.* Генерализация данных при расчете интраокулярных линз с использованием ИНС-моделей / А. А. Арзамасцев, О. Л. Фабрикантов, Н. А. Зенкова, С. В. Беликов // Вестник ВГУ: серия «Системный анализ и информационные технологии». – 2023. – № 1. – С. 80–95. DOI: https://doi.org/10.17308/sait/1995-5499/2023/1/80-95

8. *Арзамасцев А. А.* Расчет интраокулярных линз (ИОЛ) в офтальмологии с использованием моделей искусственного интеллекта / А. А. Арзамасцев, О. Л. Фабрикантов, Н. А. Зенкова, С.В. Беликов // Актуальные проблемы прикладной математики, информатики и механики : сборник трудов Международной научной конференции, Воронеж, 13–15 декабря 2021 г. – Воронеж : Издательство «Научно-исследовательские публикации», 2022. – с. 291–296.

9. *Arzamastsev A. A.* Algorithms and methods for extracting knowledge about objects defined by arrays of empirical data using ANN models / A. A. Arzamastsev, N. A. Zenkova, N. A. Kazakov // Journal of Physics: Conference Series 1902 (2021) 012097. doi:10.1088/1742-6596/1902/1/012097.

10. *Арзамасцев А. А.* Алгоритмы глубокого машинного обучения в решении задач автоматизированной обработки данных в офтальмологии / А. А. Арзамасцев, О. Л. Фабрикантов, Е. В. Кулагина, С. В. Беликов, Н. А. Зенкова // Актуальные проблемы прикладной математики, информатики и механики : сборник трудов Международной научной конференции, Воронеж, 12–14 декабря 2022 г. – Воронеж : Издательство «Научно-исследовательские публикации», 2023. – с. 285–293.

11. *Арзамасцев А. А., Фабрикантов О. Л., Зенкова Н. А., Беликов С. В.* Применение технологии машинного обучения для прогнозирования оптической силы интраокулярных линз: генерализация диагностических данных / А. А. Арзамасцев, О. Л. Фабрикантов, Н. А. Зенкова, С. В. Беликов // Digital Diagnostics. – 2024. – Т. 5, № 1. – С. 53–63. DOI: https://doi.org/10.17816/DD623995

12. *Arzamastsev A. A, Fabrikantov O. L., Zenkova N. A., Belikov S. V.* Predicting the optical power of intraocular lenses by means of dynamically developing artificial intelligence system / A. A. Arzamastsev, O. L. Fabrikantov, N. A. Zenkova, S. V. Belikov // Applied Mathematics, Computational Science and Mechanics: Current Problems (AMCSM). – 2023. – IEEE | DOI: 10.1109/AMCSM59829.2023.10525908

13. *Арзамасцев А. А., Фабрикантов О. Л., Беликов С. В., Зенкова Н. А.* Прогнозирование оптической силы интраокулярных линз динамически развивающейся системой искусственного интеллекта / А. А. Арзамасцев, О. Л. Фабрикантов, С. В. Беликов, Н. А. Зенкова // Актуальные проблемы прикладной математики, информатики и механики : сборник трудов Международной научной конференции, Воронеж, 4–6 декабря 2023 г. – Воронеж, Издательство «Научно-исследовательские публикации», 2024. – С. 150–154.

14. Патент РФ № 2021133515/22, заявл. 25.02.22, опубл. 06.03.23 Арзамасцев А. А., Фабрикантов О. Л., Сырых К. К., Беликов С. В. Способ определения оптической силы интраокулярной линзы с использованием искусственной нейронной сети // Патент России № 2791204 : МПК A61F 2/14, G16H 10/60, G06N 3/02. 2023, Бюл. № 7.

УДК 004.08

АНАЛИЗ ТОНАЛЬНОСТИ ДЛЯ МОДЕЛИРОВАНИЯ НАРРАТИВА: РУССКОЯЗЫЧНЫЙ ТЕКСТ И SENTIART

Е. В. Бирюкова, И. Е. Воронина

Воронежский государственный университет

Аннотация. Представлено сравнение различных подходов к определению тональности художественного текста в рамках задачи моделирования нарратива, а также анализ проблем, возникающих при применении стандартных алгоритмов для художественной литературы. Представлен экспериментальный подход, основанный на использовании модели SentiArt, который позволяет определять валентность текста без необходимости обучения на больших объемах данных.
Ключевые слова: нарратив, нарратология, моделирование нарратива, анализ тональности текста, машинное обучение, словарь тональности, LSA.

Введение

Нарратив представляет собой последовательное описание взаимосвязанных событий, представленное аудитории в виде вербального или визуального ряда. Компьютерная нарратология фокусируется на анализе существующих нарративов, преимущественно текстовых, с применением вычислительных методов.

Исследование [1] детально формулирует задачу анализа нарративов и предлагает алгоритм ее решения, учитывая специфику работы с русским языком. Работа [2] анализирует проблемы, возникающие при использовании стандартных алгоритмов для определения тональности художественной литературы. Эксперименты, проведенные в рамках этого исследования, демонстрируют неприменимость стандартных методов в контексте моделирования нарратива. Отмечено, что наиболее сложная архитектурная модель показала наилучшие результаты. Необходимо провести более глубокое исследование относительно узкого диапазона выраженности сантимента, выявленного в ходе анализа.

Проведённые в работе [3] эксперименты демонстрируют, что замена русского языка на английский не приводит к существенному повышению точности. Полученные результаты указывают на превалирующее влияние особенностей жанра на итоговые показатели, что обуславливает необходимость дальнейших исследований, направленных на адаптацию к стилистическим свойствам текста. Ввиду отсутствия соответствующих аннотированных корпусов, применение методов машинного обучения, требующих значительного объема данных для обучения модели, представляется нецелесообразным. Необходимо изучить возможность адаптации к русскому языку таких решений, как SentiArt [4] и SÉANCE [5].

Рассмотрим эксперименты с использованием более простых моделей, обучение которых не требует большого объема обучающих данных, где также применяется предварительное кодирование корпуса с помощью специального инструмента на основе модели векторного пространства.

1. Описание эксперимента

Для определения валентности текстового сегмента с помощью SentiArt используется следующий алгоритм:

Шаг 1. Создается обучающий корпус, общий или специализированный для конкретной задачи, который объединяет все тексты в один составной документ. Затем рассчитывается век-

торное пространство слов (VSM) с использованием простого в использовании инструмента fasttext [6].

Шаг 2. Рассчитываются значения семантической связи между каждым словом в VSM и метками из списка меток. Например, вычисление косинусного сходства между целевым словом «мучение» («AGONY») и тремя первыми положительными и отрицательными метками из набора меток Ekman99 («счастье», «удовольствие», «гордость»), эмпирически подтвержденных в предыдущих исследованиях английских и немецких текстов [4] даст среднее значение 0,33 для положительных меток и 0,51 для отрицательных. Вычитание одного из другого дает валентность -0,18, что указывает на отрицательную валентность слова «мучение» («AGONY»). Эта процедура применяется ко всем словам в VSM, вычисляя значения как валентности, так и интенсивности.

Шаг 3. Данная процедура создает двумерное Пространство Эмоционального Потенциала (валентность, умноженная на интенсивность). Это пространство может служить справочной точкой для многих будущих исследований анализа тональности. Таким образом, каждое слово из заданного тестового текста может быть размещено в этом пространстве получая стандартизированные (относительные) значения валентности и интенсивности. На третьем шаге рассчитываются средние баллы для интересующих текстовых сегментов (например, те, которые оценены как страшные или радостные), которые затем используются в качестве входных данных для классификаторов.

Необходимо вычислить показатели анализа настроений, используя словари: SentiWordNet [7] Rusentilex_2017 [8]. После вычисления трех показателей анализа настроений для каждого из 360 текстовых сегментов, эти показатели были стандартизированы и использованы как входные данные для пяти классификаторов, реализованных в Skitlearn [9], чтобы проверить точность прогнозирования категории настроения тестового набора (после обучения на обучающем наборе из 70% из 360 сегментов). Случайная выборка была стратифицированной (то есть сбалансированной по трем текстовым категориям) и повторялась 100 раз с различными обучающими и тестовыми наборами, чтобы получить стабильные результаты. В качестве контрольного условия использовалась LSA [10], реализованная в Gensim [11]. Эта система не является системой анализа настроений, что позволило оценить ее способность классифицировать текстовые сегменты без использования специфичных показателей настроения.

2. Результаты экспериментов

Результаты, обобщенные в табл. 1, представляют собой показатели классификации для каждого из трех методов анализа тональности (SAT) и LSA. Текущее сравнение классификаторов демонстрирует оптимальную производительность для признака валентности SentiArt (с использованием логистической регрессии), а также более низкие показатели для признака словаря Rusentilex_2017 с использованием нейронной сети) и признака Rusentilex_2017 (с использованием логистической регрессии). Точность контрольного метода (LSA), хотя и уступает другим, свидетельствует о том, что абстрактные семантические признаки, вычисленные LSA, все же улавливают аффективные аспекты, позволяющие классифицировать (в определенной степени) тексты в категории тональности.

Таблица 1

Сравнительные покаказатели классификации

Классификатор	SentiWordNet	Rusentilex_2017	SentiArt	LSA
Neural network	0.593	0.751	0.783	0.547
Logistic regression	0.614	0.739	0.816	0.531
AdaBoost	0.551	0.645	0.791	0.558
kNN	0.562	0.658	0.791	0.523
Naïve Bayes	0.578	0.745	0.746	0.539

Заключение

Анализ тональности текста — ключевой элемент построения модели текстового русскоязычного нарратива. Несмотря на трудности, вызванные спецификой жанров и особенностями русского языка, эксперименты показали, что использование английского языка не приводит к существенному повышению точности анализа. Это говорит о том, что проблемы жанровой принадлежности оказывают наибольшее влияние на результаты. Следовательно, дальнейшие исследования должны быть направлены на адаптацию моделей к стилистическим особенностям текста. Ввиду отсутствия необходимых аннотированных корпусов для обучения моделей машинного обучения, необходимо обратить внимание на существующие решения, адаптированные к русскому языку, такие как SentiArt и SÉANCE.

Использование инструмента SentiArt позволяет достичь точности выше 75 % при отсутствии необходимости обучать классификаторы на больших корпусах, что критично для решения задачи. Таким образом, при демонстрации такого же результата на других текстах, можно применять его при анализе тональности любого художественного текста с целью построения его нарратива.

Следует подчеркнуть, что моделирование нарративов представляет собой сложную задачу, не имеющую простого решения. Поэтому поиск и оценка различных подходов к решению этой задачи имеет первостепенное значение. Каждый положительный результат в этой области обладает значительной новизной, поскольку открывает новые горизонты для исследования и понимания механизмов построения нарративов.

Литература

1. *Бирюкова Е. В.* Методы моделирования и анализа нарративов русскоязычных текстов / Е. В. Бирюкова, И. Е. Воронина // Актуальные проблемы прикладной математики, информатики и механики: сборник трудов Международной научной конференции (Воронеж, 12–14 декабря 2022 г.). – Воронеж, 2022. – С. 176–180.

2. *Бирюкова Е. В.* Анализ тональности текста как метод моделирования русскоязычного нарратива / Е. В. Бирюкова, И. Е. Воронина // Международная научно-практическая конференция им. Э.К. Алгазинова «Информатика: проблемы, методы, технологии» (IPMT) (Воронеж, 15–17 февраля 2023 г.) – Воронеж, 2022. – С. 934–938.

3. *Бирюкова Е. В.* Сравнение анализа тональности англоязычных и русскоязычных художественных текстов / Е. В. Бирюкова, И. Е. Воронина // Актуальные проблемы прикладной математики, информатики и механики : сборник трудов Международной научной конференции, Воронеж, 4-6 декабря 2023 г. – Воронеж, 2024. – С. 1079–1085.

4. *Jacobs A. M.* Sentiment analysis for words and fiction characters from the perspective of computational (neuro-) poetics / A. Jacobs // Frontiers in Robotics and AI. – 2019. – Т. 6. – С. 53.

5. *Crossley S.* A Sentiment Analysis and Social Cognition Engine (SEANCE): An automatic tool for sentiment, social cognition, and social-order analysis / S. Crossley, K. Kyle, D. McNamara D // Behavior research methods. – 2017. – Т. 49. – С. 803–821.

6. *Mikolov T.* Advances in Pre-Training Distributed Word Representations / T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, A. Joulin // ArXiv – 2017. – Т. 1712. – С. 9405–9415.

7. *Baccianella S.* Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining / S. Baccianella, A Esuli, F. Sebastiani // Lrec. – 2010. – Т. 10. – № 2010. – С. 2200–2204.

8. *Loukachevitch N.* Creating a General Russian Sentiment Lexicon / N. Loukachevitch, A. Levchik // Lrec. – 2016. – Т. 16. – №. 2016. – С. 1171–1176.

9. Skitlearn – URL: https://scikit-learn.org

10. *Deerwester S.* Indexing by Latent Semantic Analysis / S. Deerwester, S. Dumais, G. Furnas, T. Landauer // Journal of the American Society for Information Science. – 1990. – Т. 41. – С. 391–397.

11. Gensim – URL: https://scikit-learn.org

УДК 004.048

ОБЗОР МЕТОДОВ ПРИМЕНЕНИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В ТРАЕКТОРНОЙ ОБРАБОТКЕ РАДИОЛОКАЦИОННОЙ ИНФОРМАЦИИ

Д. А. Василевский, И. Л. Каширина

МИРЭА – Российский технологический университет

Аннотация. В статье поднимается проблема классических методов траекторной обработки. На сегодняшний день эти методы постепенно теряют свою актуальность, поскольку имеют ряд существенных недостатков, ограничивающий их применимость в определённых условиях. Это приводит к снижению точности выдаваемой информации, увеличению длительности вычислений, невозможности работы в режиме реального времени и т. д. В данной работе рассмотрены способы применения технологии искусственного интеллекта для решения задач траекторной обработки с целью устранения недостатков классического подхода.

Ключевые слова: траекторная обработка, Multi-Target Tracking, распознавание, нелинейная экстраполяция, рекуррентные нейросети, несвёрточные нейроны.

Введение

В настоящее время радиолокация используется повсеместно не только в военных, но и в гражданских целях. При этом требования, предъявляемые к современным радиолокаторам, постоянно растут. Главным образом эти требования заключаются в повышении точности оценки параметров, улучшении вероятностных характеристик обнаружения и, конечно же, увеличению темпа выдачи радиолокационной информации.

Ещё в 70-ых годах прошлого века радиолокационная обработка стала выполняться автоматически — без участия человека принимается решение об обнаружении цели, завязке траектории и т. д. Однако в силу сложности математического аппарата, задачи радиолокации не могут быть решены в самом общем виде, в связи с чем автоматические радиолокаторы при расчётах принимают ряд допущений, таких, например, как стационарность помех, выделение сигнала на фоне белого шума, ограниченность числа целей и т. д. В условиях динамически меняющийся внешней обстановки классические алгоритмы значительно теряют свои преимущества: повышается вероятность ложных обнаружений, увеличивается время обработки, снижается точность. В присутствии нестационарных, негауссовых помех или большого числа целей возникает ошибочное отождествление целей и траекторий, появляется огромное количество ложных отметок и трасс.

Современный подход к решению перечисленных проблем заключается во внедрении искусственного интеллекта в системы обработки радиолокационной информации. Целью данной работы является изучение и анализ подходов к применению машинного обучения для модернизации классических радиолокаторов.

1. Классические методы траекторной обработки

Траекторная обработка радиолокационной информации обеспечивает возможность сопровождения цели на протяжении нескольких периодов обзора РЛС. На этапе траекторной обработки решаются следующие задачи:

1. *Завязка траектории* — принимается решение о взятии на сопровождение новой цели по результатам нескольких её обнаружений;

2. *Экстраполяция координат* — по имеющейся информации о траектории движения цели производится предсказание её будущего положения и выставление строба, где ожидается новое обнаружение;

3. *Идентификация отметки и траектории* — необходимо установить однозначное соответствие между множеством отметок, обнаруженных на новом периоде обзора, и множеством целей, сопровождаемых в данный момент

Рассмотрим классические методы решения задач траекторной обработки, а также проблемы, возникающие при этом.

1.1. Основные принципы траекторной обработки

В настоящее время традиционным базовым принципом траекторной обработки считается *принцип фильтрации Калмана*. Фильтр Калмана — адаптивный алгоритм предсказания эволюции системы (в данном случае — движения цели) на основании взвешенной суммы результатов измерений и предыдущих предсказаний. Для предсказания будущего состояния цели используется некоторая модель её движения. Для измерения текущего состояния используются результаты обнаружения цели на каждом новом периоде обзора. Весовые коэффициенты предсказания и измерения динамически изменяются и зависят от дисперсии ошибок, соответственно, модели и измерителя координат. Суть подхода можно свести к следующему. Если имеется неточный обнаружитель, с высокой дисперсией ошибок, весовой коэффициент измерений будет малым, тем самым вклад измерителя в предсказанное состояние цели уменьшается, а фильтр будет больше доверять предсказанию модели. Если же уменьшается точность модели движения, то фильтр станет больше доверять измеренным значениям, чем предсказанным. Таким образом, калмановская фильтрация позволяет системе адаптироваться к изменениям поведения цели.

Процесс завязки траектории сводится к следующему алгоритму (рис. 1). После появления в результате очередного обзора новой обнаруженной отметки создаётся формуляр траектории, включающий в себя информацию о координатах цели и её скорости. Далее выставляется строб завязки — некоторая область вокруг полученной отметки, где ожидается новое измерение координат. На следующем периоде обзора все отметки, попавшая внутрь строба, создают новые формуляры траектории, для каждого из которых выставляется собственный строб завязки.

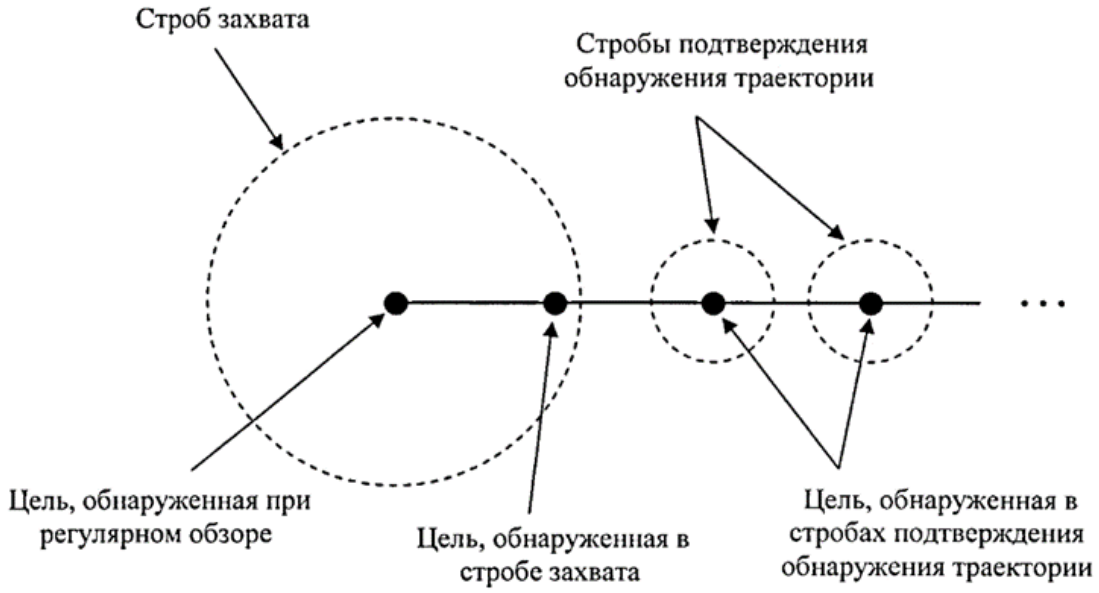


Рис. 1. Процесс завязки траектории

Решение о взятии цели на сопровождение принимается по нескольким результатам обнаружения цели в стробе. Типичным критерием может являться 3 обнаружения за 4 периода обзора. Пока этого не произошло, цель не взята на сопровождение и траектория всё ещё находится на стадии завязки. Если критерий завязки выполнен, траектория считается истинной, цель берётся на сопровождение. На основании текущих измерений с помощью экстраполяции и алгоритма калмановской фильтрации формируется предсказание следующего состояния цели. Вокруг предсказанных координат формируется строб сопровождения, размеры которого меньше строба завязки. Следующее измерение цели ожидается именно в этой области. Если же критерий завязки не выполнен (цель в стробе не обнаружена), траектория сбрасывается. Сброс сопровождаемой траектории осуществляется по новому критерию, например, 3 пропуска подряд [1].

Все перечисленные выше алгоритмы рассмотрены пока что для случая сопровождения одной цели. В случае, когда траекторий и отметок несколько, ставится задача их взаимного отождествления.

Отождествление (идентификация) производится последовательно. Сперва каждая из отметок, обнаруженных за период повторения, проверяется на возможность продолжения существующих траектории. Для каждой траектории выбирается наиболее подходящая, согласно заданному критерию оптимальности, отметка. В дальнейшей обработке эти отметки не участвуют. Для оставшихся отметок оценивается возможность продолжения траекторий, находящихся на стадии завязки. Отметки, «не пригодные» не для существующих траекторий, ни для формируемых, считаются новыми целями, для которых снова создаётся формуляр траектории, выставляется строб завязки и так далее.

Критерий оптимальности отождествления отметок формулируется на основе классической теории принятия решений: например, байесовский критерий максимального правдоподобия. Для этого следует определить совместную плотность вероятности обнаруженных отметок (измерений) и предсказаний, экстраполированных на предыдущем обзоре. Предполагая измерения координат независимыми, можно записать совместную плотность распределения в виде произведения гауссовых случайных величин:

$$w(x_1,\ldots x_n, y_1,\ldots y_n) = \exp\left(-\sum_{i=1}^{n}\sum_{j=1}^{J}\frac{(x_{ij}-y_{kj})}{\sigma_{ij}^2}\right),$$

где x_i — вектор предсказанных координат i-й траектории;

y_i — вектор измеренных координат i-й отметки;

J — количество измеряемых координат;

σ_{ij} — дисперсия отклонений j-й координаты от предсказания;

k — выбранный набор номеров идентифицируемых отметок.

Задача заключается в подборе таких номеров k, при которых функция правдоподобия будет максимальна (присвоить отметкам номера соответствующих им траекторий). Такой аппарат требует достаточно трудоёмких и объёмных вычислений. При этом количество отметок не всегда совпадает с количеством траекторий: возможны появления новых целей, ложные тревоги и отсутствие измерений.

Альтернативой этому методу может послужить построение двудольного взвешенного графа. Рёбрам графа присваиваются веса, равные сумме дисперсий отклонений измеряемых координат от предсказания σ_{ij}^2. Минимизация суммарного веса такого графа приведёт к оптимальной идентификации отметок по критерию минимума среднего квадрата ошибки. В некоторых случаях такой подход позволяет сократить объём вычислений, однако он всё ещё будет расти пропорционально 4-й степени размерности задачи.

1.2. Основные недостатки классических методов траекторной обработки

Все изложенные выше алгоритмы и методы имеют ряд существенных недостатков. Самым главным из них можно назвать *огромный объём трудоёмких вычислений*. В этом можно убедиться на примере решения задачи отождествления отметок, который при большом количестве обнаружений сводится к перебору огромного числа вариантов сочетаний отметка-траектория для оценки функции правдоподобия.

Другим недостатком является *линейность* алгоритма калмановской фильтрации. В ситуациях, когда цель активно маневрирует, фильтр Калмана накапливает ошибку предсказания из-за невозможности нелинейной экстраполяции движения, в результате чего цель может сброситься с сопровождения.

Кроме того, следует отметить, что величина строба в классических методах траекторной обработки определяется величиной ошибки измерения соответствующих координат, которая может достигать нескольких сотен метров. Эта проблема не проявляет себя при сопровождении одиночной цели, однако в условиях интенсивного налёта в стробе появляется слишком большое число отметок, что может привести к ошибочной завязке траектории.

Особенно сильно ухудшение качества сопровождения проявляется при *сопровождении групповой цели* (например, рой БПЛА). В этом случае, как правило, сопровождение целей по отдельности становится практически невозможным, а число ложных траекторий может составлять до 6 для одной группы из четырёх целей.

Для повышения точности сопровождения были предложены различные методы *кластеризации (группирования) отметок* [1]. При таком подходе цели на этапе завязки траектории не различаются в рамках кластера и сопровождаются единой целью. В дальнейшем, для каждой цели внутри группы формируются отдельные стробы сопровождения. Однако такой подход не позволяет достичь требуемой эффективности траекторной обработки, поскольку точность измерения центра кластера существенно зависит от количества целей в его пределах, их скоростей, логики расположения и построения, которые при этом не учитываются.

Таким образом, можно сделать вывод о том, что классические методы траекторной обработки обеспечивают приемлемое качество сопровождения в достаточно ограниченных условиях. При резких маневрах цели или движении целей в группах приведённые алгоритмы становятся непригодными в силу роста вычислительной сложности и ошибки сопровождения.

2. Методы применения искусственного интеллекта в траекторной обработке

Использование нейронных сетей в системах траекторной обработки может существенно сгладить недостатки классических методов. При этом появляется возможность уменьшения вычислительной сложности и распараллеливания алгоритмов обработки, что сокращает время, затрачиваемое на решение задач сопровождения. Среди таких методов можно выделить *многомерный трекинг (Multi-Target Tracking), распознавание целей и ситуаций и нелинейная экстраполяция временных рядов*.

Рассмотрим подробнее эти методы и возможности их применения.

2.1. Многомерный трекинг (Multi-Target Tracking)

Метод многомерного трекинга заключается в создании нейросетевой структуры (как правило, используются рекуррентные нейросети), способной успешно распознавать принадлежность измерений, полученных за период обзора, к траекториям, сопровождаемым в данный момент, то есть решать задачу идентификации отметка-траектория. При этом предполагается

повышение эффективности идентификации по сравнению с классическими методами в случае сопровождения групповой цели.

Для реализации данного метода может быть использована *технология DeepSORT*, применяемая для трекинга объектов изображений. Главным преимуществом архитектуры DeepSORT является возможность оценивать так называемый «внешний вид» – набор некоторых уникальных свойств объектов, запоминать его и использовать для отождествления с ним новых измерений. Таким образом, идентификация происходит путём оценки расстояния между измерением и предсказанием не только по пространственным координатам, но и по некоторому пространству признаков по формуле [2]:

$$D = k * D_m + (1-k) * D_a,$$

где D — совокупная оценка расстояния между измерением и предсказанием;

k — весовой коэффициент;

D_m — метрика Махалонобиса;

D_a — расстояние по внешней схожести (в пространстве внешних признаков)

Метрика Махалонобиса здесь представляет собой статистический аналог привычного евклидова расстояния, который устраняет учитывает ковариационные моменты между приходящими измерениями [2]:

$$D_m(x) = \sqrt{(x-\mu)^T * S^{-1} * (x-\mu)^T},$$

где x — вектор измерений;

μ — вектор математических ожиданий (средних значений) измеряемых величин;

S^{-1} — обратная ковариационная матрица измерений

Благодаря возможности анализировать схожесть отметки и сопровождаемой цели, обеспечивается не только повышение точности идентификации, но и уменьшение количества ложных траекторий. Так, например, в ситуациях, когда измерения от цели отсутствуют на протяжении нескольких периодов обзора (высокий уровень шума, помеховый сигнал, сложная фоно-целевая обстановка), траектория сбрасывается, а если с новым обзором приходит новая отметка от той же самой цели, вместо завязки новой траектории, возможно идентифицировать цель по набору внешних признаков и возобновить её сопровождение даже после сброса. Такое преимущество достижимо при объединении процессов первичной и вторичной обработки, что является перспективной областью для исследований.

Кроме технологии DeepSORT следует также упомянуть метод многомерного трекинга, рассмотренный в работе [3]. Здесь проанализирована модификация широко-используемой технологии JPDA (*joint probabilistic data association* — совместное сопоставление вероятностных данных). Основным недостатком этой технологии является необходимость решать сложную статистическую задачу максимизации функции совместного распределения вероятностей, мерность которой увеличивается многократно при увеличении числа целей.

Решить указанную проблему предлагается путём добавления в скрытый слой структуры блоков LSTM (*long short-term memory* — длительная краткосрочная память), которые позволяют лучше удерживать информацию о статистических связях измерений, пришедших за несколько периодов обзора. На рис. 2 представлена схема процесса предсказания и идентификации в такой сети. Кроме того, сопоставление измерений и траекторий при этом происходит для каждой цели по отдельности, в результате чего уменьшается вычислительная сложность. Время решения задачи также уменьшается при параллельной реализации алгоритма.

2.2. Нелинейная экстраполяция временных рядов

Данный метод позволяет устранить недостаток линейности калмановской фильтрации. Это позволяет обеспечить высокую точность сопровождения в условиях маневрирующей цели.

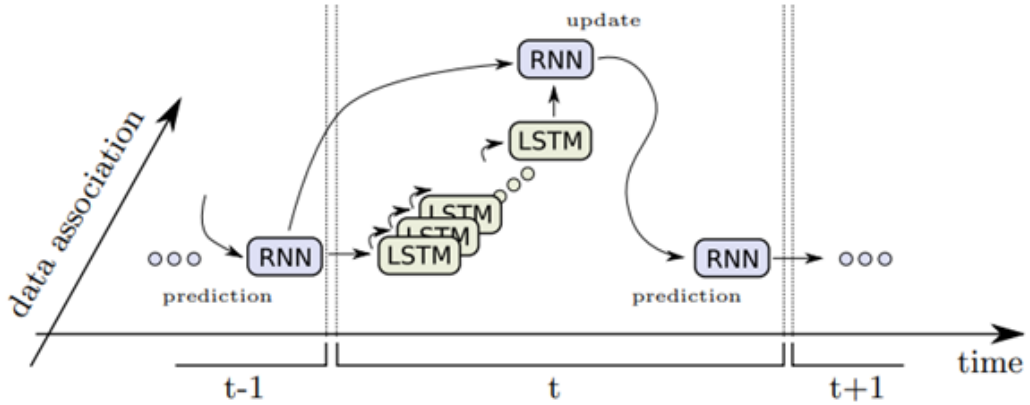


Рис. 2. Схема процесса предсказания с использованием LSTM

Основная идея использования нейронных сетей для экстраполяции координат основана на возможности *экстраполяции временных рядов*, которыми могут являться координаты сопровождаемых целей. В этом случае экстраполяция производится не по единичному измерению координат и скорости, а по совокупности всех предшествующих измерений. Следует только учесть, что данный метод должен быть обобщён для многомерного пространства (экстраполяция векторной величины).

Для решения данной задачи могут быть использованы *несвёрточные нейроны*. Их активационная функция использует семейство нелинейных функций вместо традиционной свёртки входного воздействия с вектором весовых коэффициентов. Такими нелинейными функциями могут быть, например, ряды Фурье, полиномы различных степеней и т. д.

В работе [4] было проведено исследование модели нейросети, активационная функция которых является рациональной полиномиальной дробью. Результаты моделирования показали, что рациональная функция способна повысить точность экстраполяции на 30…40 % по сравнению с полиномиальной апроксимацией. При этом для достижения такой точности достаточна сравнительно низкая степень полиномов, что означает уменьшение количества оптимизируемых коэффициентов. Кроме того, если экстраполируемая функция имеет низкий порядок, появляется возможность ещё больше понизить степень рациональной функции путём отбрасывания коэффициентов, влияние которых на экстраполяцию выражается слабо.

На практике, при решении задач траекторной обработки, заранее неизвестен характер экстраполируемой функции, поскольку цель может быть как высокоманевренной (как, например, БПЛА), так и маломаневренной (дирижабль, баллистическая ракета, боинг). В связи с этим возможно расширение возможностей нейросети адаптироваться под характер движения объекта с целью упрощения задачи экстраполяции.

Также рассматривается возможность гибридного подхода, суть которого заключается в совместном использовании классического фильтра Калмана и результатов нелинейной экстраполяции. При таком подходе на этапе завязки траектории система больше опирается на предсказания калмановского фильтра для формирования некоторой выборки отметок от цели. А уже после того, как выборка предшествующих измерений сформирована, система производит нелинейную экстраполяцию траектории.

2.3 Распознавание целей и ситуаций

Как уже было показано, информация о типе сопровождаемой цели может играть огромное значение для траекторной обработки. Во-первых, тип цели во многом определяет характеристики её движения: типичную и максимальную скорости, ускорения, способности к маневрированию и т. д. Во-вторых, при отождествлении отметок и траекторий может быть исполь-

зована информация о «внешнем виде» цели (её радиолокационный портрет). В-третьих, зная размер и скорость цели, можно выставить соответствующего размера строб завязки и сопровождения. Для малоподвижной цели не требуется выставлять большой строб, соответственно, в него может попасть значительно меньше прочих отметок.

При *распознавании целей* используются спектральные и траекторные признаки. К *спектральным признакам* относятся характерные пики на определённых частотах, наличие флуктуаций в характерных спектральных областях и пр. Кроме того, характерным признаком любой цели может являться эффективная площадь рассеяния, определяющая габариты цели и ракурс её обнаружения. К *траекторным признакам* относят типовую скорость, высоту полёта, маневр.

В классических методах возможно распознавание, как правило, двух классов целей. При увеличении числа классов растёт количество анализируемых признаков цели, из-за чего сложность задачи и объём вычислений возрастают многократно.

Технология нейронных сетей является наиболее подходящим инструментом для решения задачи распознавания объектов, поскольку изначально искусственный интеллект создавался именно для этих целей. Он способен различать гораздо большее число классов, выделять больше характерных признаков. При этом обработка по многомерному пространству этих признаков может проводиться с помощью алгоритмов нечёткой логики, особенно в задачах, когда формализованное математическое их описание затруднено. Особенно применим здесь иерархический принцип перебора признаков, например: сперва цели классифицируются по размеру, потом по максимальной скорости и так далее.

Такой же подход может быть применим и к *распознаванию ситуаций*. Для данного метода наиболее типичной задачей может являться выделение наиболее приоритетной цели, либо оценка тактики передвижения целей в группах. Структурировать эти задачи для описания строгим математическим аппаратом невозможно, однако можно использовать нейронные сети с так называемым *механизмом внимания*, рассмотренные в работах [5,6].

Модель такой нейронной сети имеет архитектуру *автоэнкодера*, модифицированного слоем рекуррентной Stacked LSTM, представленную на рис. 3. Механизм внимания расширяет информацию, поступающую от энкодера, дополняя её некоторым ситуативным контекстом. Благодаря этому контексту, декодер способен выделять некоторые скрытые состояния энкодера в качестве приоритетных и «концентрироваться» на них при формировании выходной последовательности. Stacked LTSM при этом выступает в роли некоторого накопительного буфера, информация в котором служит основой для формирования контекста.

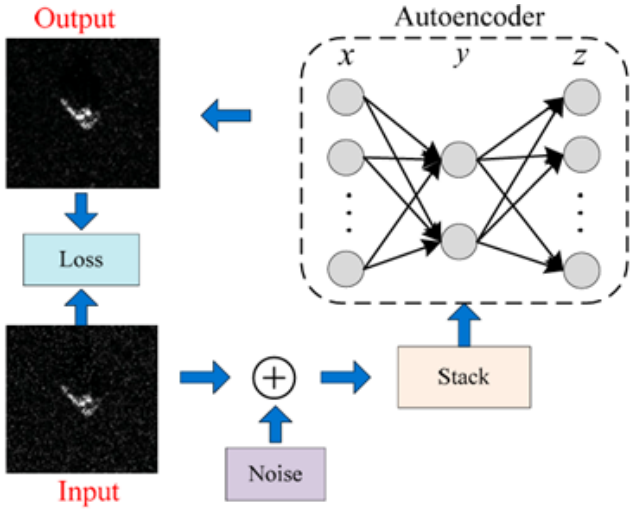


Рис. 3. Архитектура автоэнкодера с механизмом внимания

Распознавание целей и ситуаций относится скорее к третичной обработке радиолокационной информации, однако её результаты могут в значительной степени облегчить решение задач траекторной обработки. Таким образом, применение технологии нейронных сетей способствует объединению процессов и этапов радиолокационной обработки.

Заключение

В рамках данной статьи поднята проблема методов классической траекторной обработки. Были выявлены основные недостатки этих методов, заключающиеся в линейности предсказательной модели фильтра Калмана, сложности и неточности алгоритма идентификации в случае попадания нескольких отметок в строб завязки (или сопровождения), а также применение ряда допущений, которые не выполняются при сопровождении реальных целей. Все эти недостатки способствуют ухудшению результата обработки в ситуациях движения целей в группах, наличия негауссовских помех, сложной фоно-целевой обстановки и т. д. Для сглаживания этих недостатков возможно применение технологии искусственного интеллекта для решения задач траекторной обработки. В работе были рассмотрены методы применения нейронных сетей для решения обозначенных проблем. Нелинейная экстраполяция обеспечивает более высокую точность сопровождения цели в случае маневра. Для решения задачи идентификации в случае групповой цели возможно применение технологии DeepSORT, способной распознавать объекты на основании их «внешних» признаков. Алгоритм обработки может также быть упрощён, если имеется некоторая информация и типах сопровождаемых целей, которая может быть получена с помощью нейронных сетей с механизмом внимания. Все эти подходы могут использоваться совместно друг с другом и с классическими алгоритмами. В этом случае возможно значительно повысить точность сопровождения и уменьшить количество ложных траекторий. При этом также важным преимуществом является способность нейросетей к параллельному проведению вычислений, что может ускорить проведение траекторной обработки для работы системы в режиме реального времени.

Литература

1. *Татузов А. Л.* Нейронные сети в задачах радиолокации. Кн. 28. – Москва : Радиотехника, 2009. – 432 с.

2. Multi-Object Tracking with DeepSORT – URL: https://www.mathworks.com/help/fusion/ug/multi-object-tracking-with-deepsort.html (дата обращения: 14.11.2024)

3. Online Multi-target Tracking using Recurrent Neural Networks – URL: https://arxiv.org/pdf/1604. 03635v1.pdf (дата обращения: 20.11.2024)

4. *Толстых В. Н.* Нейронные сети для экстраполяции временных рядов // Наукоемкие технологии в космических исследованиях Земли. – 2023. – Т. 15, № 6. – С. 4–11.

5. *Jiang W., Wang Y., Li Y., Lin Y., Shen W.* Radar Target Characterization and Deep Learning in Radar Automatic Target Recognition: A Review. Remote Sens. 2023. – URL: https://doi.org/ 10.3390/rs15153742 (дата обращения: 16.112024)

6. Attention Mechanism in Neural Networks. – URL: https://arxiv.org/2204.13154v1 (дата обращения: 18.11.2024)

7. *Vaswani A., Shazeer N., Parmar N. [et al.]* Attention Is All You Need. – URL: https://arxiv.org/1706.03762 (дата обращения: 18.11.2024)

8. Simple Online and Real-time Tracking. – URL: https://arxiv.org/pdf/1602. 00763v2.pdf (дата обращения: 12.11.2024)

УДК 004.855.5

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ВЫЯВЛЕНИЯ ЦЕНООБРАЗУЮЩИХ ФАКТОРОВ СТОИМОСТИ НОУТБУКА

И. С. Демидова

Воронежский государственный университет

Аннотация. В работе произведен анализ машинного обучения для выявления факторов, определяющих стоимость ноутбука. Рассмотрены и использованы такие модели, как дерево решений, случайный лес, градиентный бустинг.
Ключевые слова: машинное обучение, дерево решений, случайный лес, градиентный бустинг, ноутбуки, прогнозирование цен.

Введение

В последние годы рынок ноутбуков претерпел значительные изменения, обусловленные стремительным развитием технологий, изменением потребительских предпочтений и глобальными экономическими факторами. В условиях высокой конкуренции производители стремятся оптимизировать свои предложения, а покупатели ищут наилучшее соотношение цены и качества. В этой связи возникает необходимость в глубоком анализе факторов, влияющих на стоимость ноутбуков, что позволяет не только понять динамику цен, но и предсказать их изменения в будущем.

Машинное обучение, как один из наиболее перспективных инструментов анализа данных, предоставляет уникальные возможности для выявления скрытых закономерностей и взаимосвязей между различными характеристиками ноутбуков и их ценами.

В процессе исследования будут рассмотрены различные алгоритмы машинного обучения, проведен анализ данных и оценка эффективности моделей. Результаты данного исследования могут быть полезны как для производителей при формировании ценовой политики, так и для потребителей при выборе оптимального устройства в соответствии с их потребностями и бюджетом.

1. Параметры

На цену ноутбука влияют различные факторы. В данной работе стоимость ноутбука будет оцениваться с помощью следующих параметров:

– компания, производящая ноутбук, бренд;
– тип ноутбука;
– оперативная память;
– вес ноутбука;
– наличие сенсорного экрана;
– оригинальное зарядное устройство;
– ppi, единица измерения разрешающей способности монитора;
– центральный процессор;
– жесткий диск;
– накопитель;
– графический процессор;
– операционная система.

2. Обработка данных

Исходные данные требуют предварительной обработки перед тем, как на них будут обучаться модели. Первые несколько строк загруженного набора данных представлены на рис. 1.

	Company	TypeName	Ram	Weight	Price	TouchScreen	Ips	Ppi	Cpu_brand	HDD	SSD	Gpu_brand	Os
0	Apple	Ultrabook	8	1.37	11.175755	0	1	226.983005	Intel Core i5	0	128	Intel	Mac
1	Apple	Ultrabook	8	1.34	10.776777	0	0	127.677940	Intel Core i5	0	0	Intel	Mac
2	HP	Notebook	8	1.86	10.329931	0	0	141.211998	Intel Core i5	0	256	Intel	Others
3	Apple	Ultrabook	16	1.83	11.814476	0	1	220.534624	Intel Core i7	0	512	AMD	Mac
4	Apple	Ultrabook	8	1.37	11.473101	0	1	226.983005	Intel Core i5	0	256	Intel	Mac

Рис. 1. Первые строки загруженного набора данных

Необходимо проверить, есть ли пропуски в параметрах, для этого воспользуемся функцией *info()*. На рис. 2 представлена информация о ненулевых параметрах и типах данных. Нулевых параметров нет, дополнительно проставлять значения не нужно.

```
Data columns (total 13 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Company      1273 non-null   object
 1   TypeName     1273 non-null   object
 2   Ram          1273 non-null   int64
 3   Weight       1273 non-null   float64
 4   Price        1273 non-null   float64
 5   TouchScreen  1273 non-null   int64
 6   Ips          1273 non-null   int64
 7   Ppi          1273 non-null   float64
 8   Cpu_brand    1273 non-null   object
 9   HDD          1273 non-null   int64
 10  SSD          1273 non-null   int64
 11  Gpu_brand    1273 non-null   object
 12  Os           1273 non-null   object
dtypes: float64(3), int64(5), object(5)
```

Рис. 2. Информация о параметрах

На рис. 2 видно, что пять параметров не имеют численного представления. Можно закодировать данные, не имеющие численного представления, с помощью метода dummy-кодирования, который заключается в кодировании категориального признака с *n* возможными значениями с помощью *n* бинарных признаков. Каждый бинарный признак соответствует одному из возможных значений категориального признака и является индикатором того, что на данном объекте он принимает данное значение. После кодирования всех признаков удаляются лишние исходные столбцы и получаются данные, представленные на рис. 3.

	Ram	Weight	Price	TouchScreen	Ips	Ppi	HDD	SSD	Company_Acer	Company_Apple	...	Cpu_brand_Intel Core i3	Cpu_brand_Intel Core i5	Cpu_brand_Intel Core i7	Cpu_brand_Other Intel Processor	Gpu_brand_AMD	Gpu_brand_Intel	Gpu_brand_Nvidia	Os_Mac	O
0	8	1.37	11.175755	0	1	226.983005	0	128	False	True	...	False	True	False	False	False	True	False	True	
1	8	1.34	10.776777	0	0	127.677940	0	0	False	True	...	False	True	False	False	False	True	False	True	
2	8	1.86	10.329931	0	0	141.211998	0	256	False	False	...	False	True	False	False	False	True	False	False	
3	16	1.83	11.814476	0	1	220.534624	0	512	False	True	...	False	False	True	False	True	False	False	True	
4	8	1.37	11.473101	0	1	226.983005	0	256	False	True	...	False	True	False	False	False	True	False	True	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1268	4	2.20	10.555257	0	0	100.454670	500	0	False	False	...	False	False	True	False	False	False	True	False	
1269	4	1.80	10.433899	1	1	157.350512	0	128	False	False	...	False	False	True	False	False	True	False	False	
1270	16	1.30	11.288115	1	1	276.053530	0	512	False	False	...	False	False	True	False	False	True	False	False	
1271	2	1.50	9.409283	0	0	111.935204	0	0	False	False	...	False	False	False	True	False	True	False	False	
1272	6	2.19	10.614129	0	0	100.454670	1000	0	False	False	...	False	False	True	False	True	False	False	False	

1273 rows x 44 columns

Рис. 3. Подготовленные для обучения данные

Используемые модели машинного обучения

Для анализа были выбраны следующие модели машинного обучения:

– модель дерева решений;

– модель случайного леса;

– модель градиентного бустинга.

3. Оценка качества регресии

Коэффициент детерминации R^2 — статистический показатель, отражающий объясняющую способность регрессии $f : X \to X$ и определяемый как доля дисперсии зависимой переменной, объяснённая регрессионной моделью с данным набором независимых переменных.

Модель «Дерево решений»

Дерево решений — это метод, позволяющий предсказывать значения зависимой переменной в зависимости от соответствующих значений одной или нескольких независимых переменных. У данного метода точность R^2 составила 0.84 при максимальной глубине дерева (max_depth) равной 10.

На рис. 4 представлена диаграмма важности признаков для модели «Дерево решений», которая показывает, как параметры набора данных влияют на модель. Можно сделать вывод, что признаки «Оперативная память», «Вес ноутбука», «Процессор Intel», «Тип ноутбук», «Единица измерения разрешающей способности монитора», «Процессор Intel Core i5», «Процессор Intel Core i7», «Накопитель», «Наличие сенсорного экрана», «Процессор AMD», «Операционная система», «Операционная система Windows», «Бренд Acer», «Бренд HP», «Бренд Lenovo» , «Бренд Dell» влияют на модель, а значит остальные признаки можно удалить для более быстрого обучения модели.

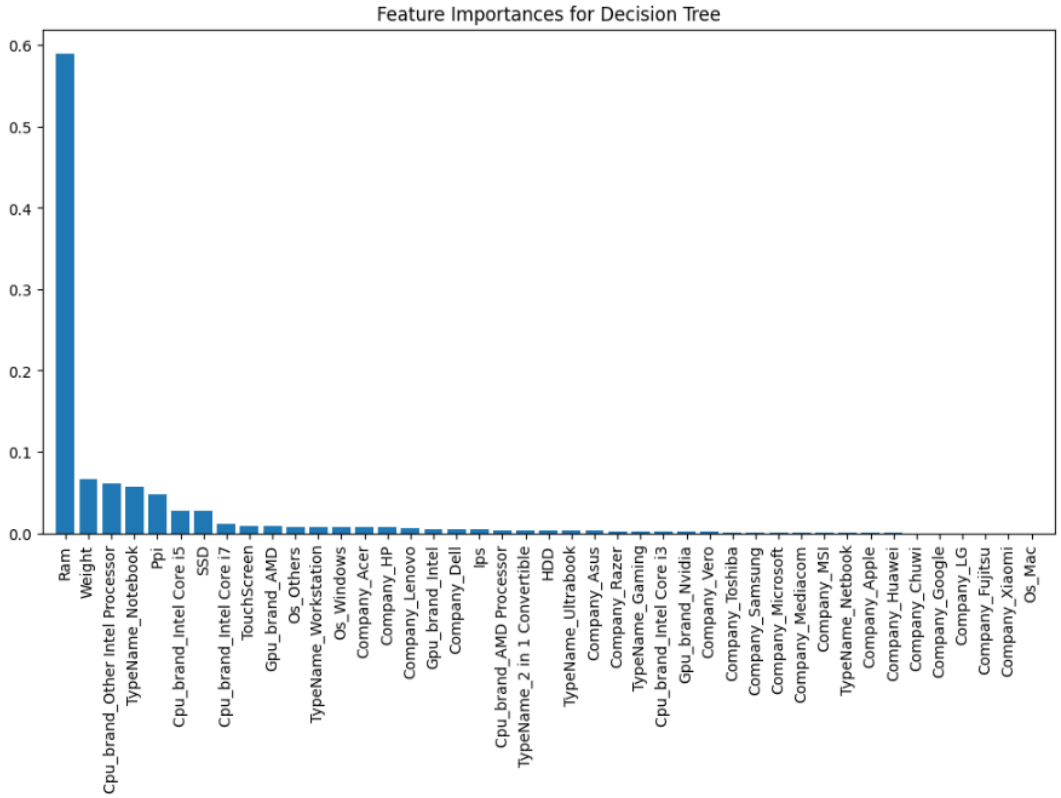


Рис. 4. Диаграмма важности признаков для модели «Дерево решений»

Модель «Случайный лес»

Модель случайного леса — алгоритм машинного обучения, заключающийся в использовании ансамбля решающих деревьев.

У данного метода точность R^2 составила 0.89 при оптимальном числе деревьев (n_estimators) равным 100 и максимальной глубине дерева (max_depth) равной 10.

На рис. 5 представлена диаграмма важности признаков для модели «Случайный лес», которая показывает, как параметры набора данных влияют на модель. Можно сделать вывод, что признаки «Оперативная память», «Вес ноутбука», «Процессор Intel», «Тип ноутбук», «Единица измерения разрешающей способности монитора», «Процессор Intel Core i5», «Накопитель», «Наличие сенсорного экрана», «Жесткий диск», «Операционная система», «Бренд Acer», «Бренд HP», «Оригинальное зарядное устройство» , «Процессор Intel Core i3» влияют на модель, а значит остальные признаки можно удалить для более быстрого обучения модели.

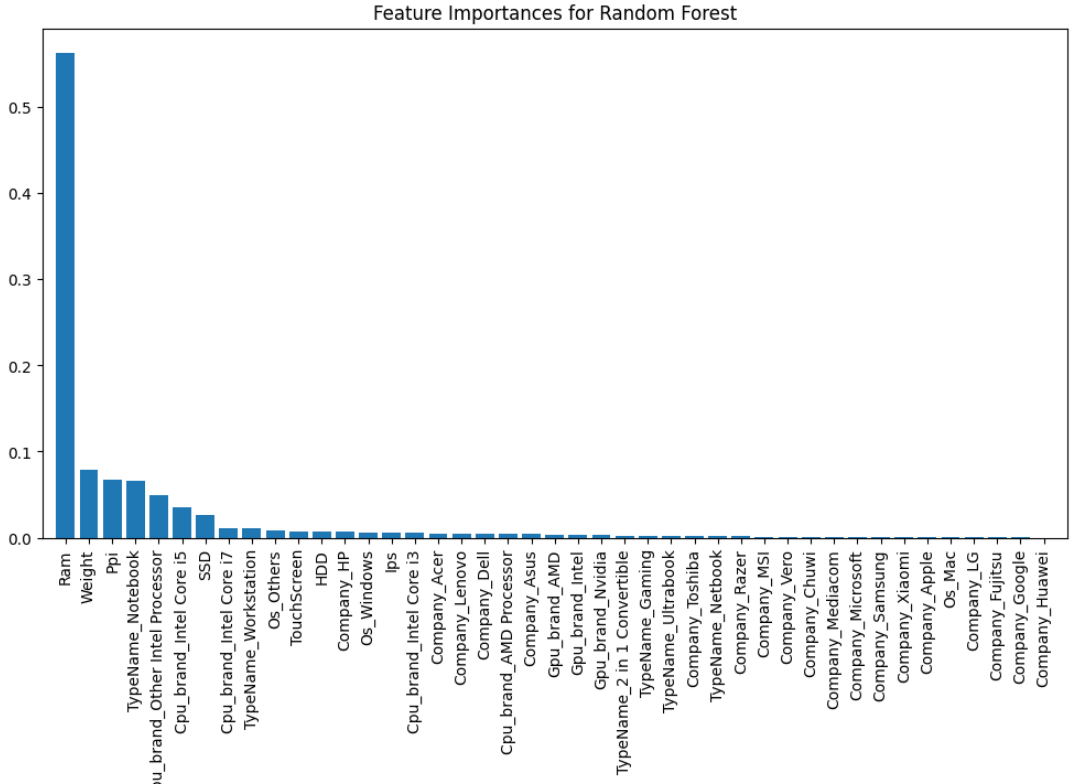


Рис. 5. Диаграмма важности признаков для модели «Случайный лес»

Модель «Градиентный бустинг»

Градиентный бустинг — это техника машинного обучения для задач классификации и регрессии, которая строит модель предсказания в форме ансамбля слабых предсказывающих моделей, обычно деревьев решений. У данного метода точность R^2 составила 0.85 при оптимальном числе деревьев (n_estimators) равным 100 и максимальной глубине дерева (max_depth) равной 4.

На рис. 6 представлена диаграмма важности признаков для модели «Градиентный бустинг», которая показывает, как параметры набора данных влияют на модель. Можно сделать вывод, что признаки «Оперативная память», «Вес ноутбука», «Процессор Intel», «Тип ноутбук», «Единица измерения разрешающей способности монитора», «Процессор Intel Core i5», «Процессор Intel Core i7», «Накопитель», «Операционная система», «Операционная система Windows», «Бренд HP», «Стационарный компьютер» влияют на модель, а значит остальные признаки можно удалить для более быстрого обучения модели.

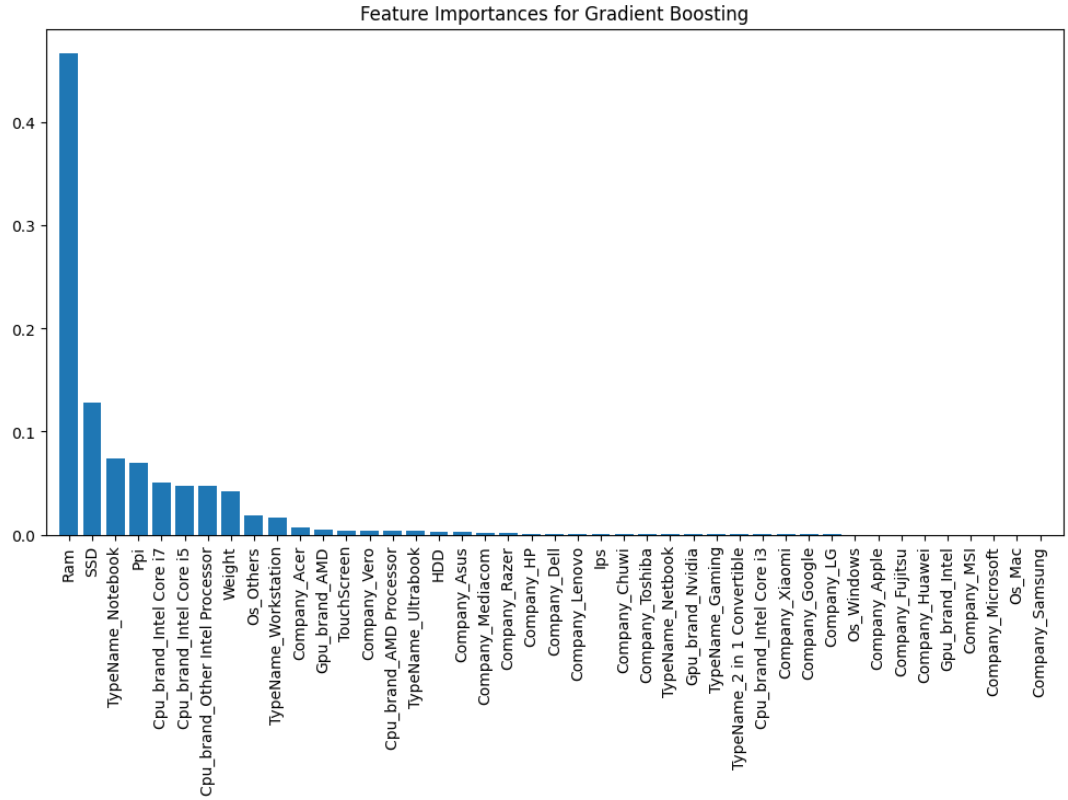


Рис. 6. Диаграмма важности признаков для модели «Градиентный бустинг»

Диаграмма рассеивания

Для оценки точности каждой из рассмотренных моделей помимо метрики R^2 была построена диаграмма рассеивания, которая показывает соотношение предсказанных и реальных значений. На рис. 7 видно, что больше совпадений в диапазоне цен от 95000 до 120000. Значительно хуже предсказываются цены на ноутбуки от 121000 рублей.

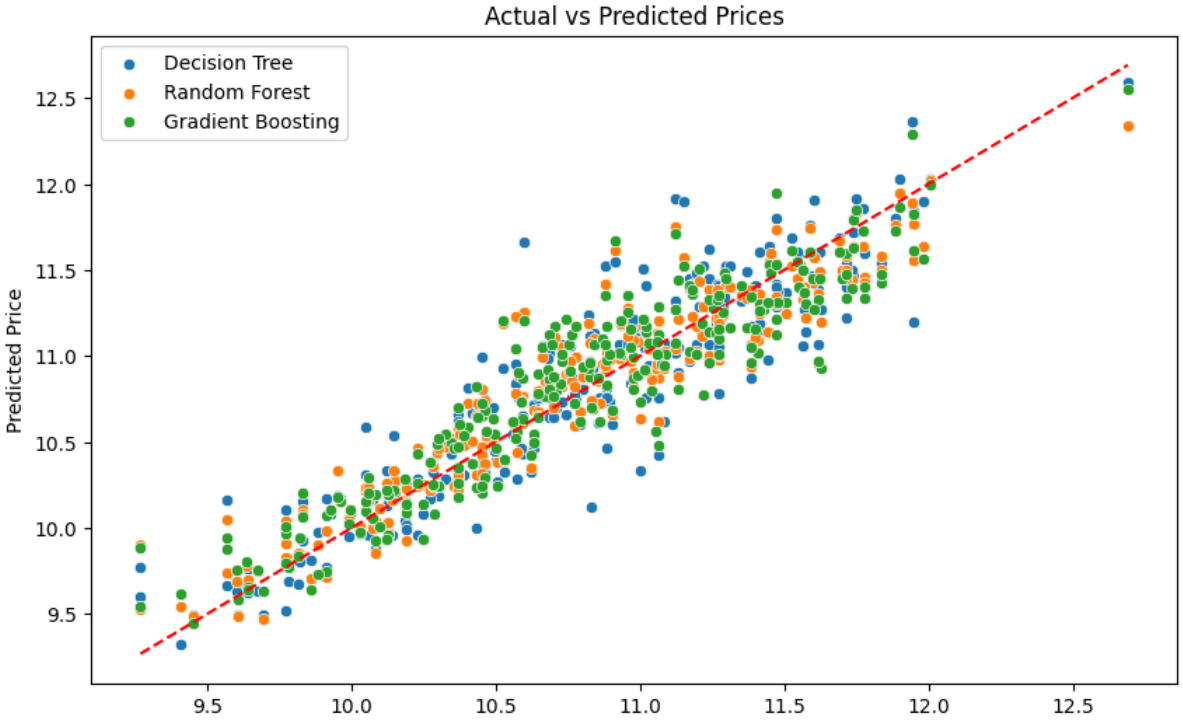


Рис. 7. Диаграмма рассеивания

Заключение

В результате сравнения наилучший коэффициент детерминации показала модель случайного леса. Среди рассмотренных моделей она лучше всего подходит для прогнозирования цен на ноутбуки.

Главным фактором, влияющим на стоимость ноутбука, у трёх моделей является оперативная память.

Литература

1. Теория и практика машинного обучения : учеб. пособие / В. В. Воронина, А. В. Михеев, Н. Г. Ярушкина, К. В. Святов. – Ульяновск : УлГТУ, 2017. – 290 с.

2. Современные ноутбуки – от чего зависит их стоимость. – URL:https://www.infoconnector.ru/devices/notebook/sovremennye-noutbuki-ot-chego-zavisit-ikh-stoimost (дата обращения 10.09.2024).

3. Метод случайного леса : основы и примеры. – URL: https://sky.pro/wiki/python/metod-sluchajnogo-lesa-osnovy-i-primery (дата обращения 20.09.2024).

4. Градиентный бустинг. – URL:https://education.yandex.ru/handbook/ml/article/gradientnyj-busting (дата обращения 17.09.2024).

5. Модели классификации: дерево решений. – URL:https://python-school.ru/blog/osnovy-ml/decisiontreeclassifier (дата обращения 23.09.2024).

УДК 004.023

ОПТИМИЗАЦИЯ КУБИЧЕСКИХ СПЛАЙНОВ ДЛЯ АКТИВАЦИИ СКРЫТОГО СЛОЯ МАШИН ЭКСТРЕМАЛЬНОГО ОБУЧЕНИЯ ПРИ РЕШЕНИИ ЗАДАЧИ КЛАССИФИКАЦИИ ТЕКСТОВ

Л. А. Демидова, В. Е. Журавлев

МИРЭА — Российский технологический университет

Аннотация. В данной работе рассматривается машина экстремального обучения (ELM), использующая кубические сплайны для активации скрытого слоя, применительно к решению задачи анализа тональности текстовых отзывов на рестораны. Координаты опорных точек для построения сплайнов подбираются с помощью представленной в данной работе островной модификации популяционного алгоритма оптимизации ETFSS, основанной на модели мягкого острова (SIM). В проведенном численном эксперименте итоговая точность классификации с использованием кубических сплайнов превосходит результаты, полученные классической моделью ELM с сигмоидальной функцией активации.
Ключевые слова: машинное обучение, классификация, обработка текстов на естественном языке (NLP), анализ тональности, векторные представления, FastText, кубический сплайн, машина экстремального обучения (ELM), оптимизация, популяционные алгоритмы, островная модель.

Введение

Классификация текстов представляет собой сложную комплексную задачу на стыке таких научных направлений, как компьютерная лингвистика, машинное обучение и математическая статистика. Любой осмысленный текст, написанный на естественном языке, всегда обладает определенной структурой и подчиняется различным правилам, например, синтаксическим или лексическим, иначе его было бы невозможно понять. Однако в реальной жизни тексты часто содержат ошибки и неточности, но не теряют при этом закладываемый в них смысл. Огромную роль играет также контекст, в зависимости от которого одна и та же фраза может иметь совершенно разные значения.

Таким образом, обработка текстовой информации прочно ассоциируется с большими объемами данных, сложной структурой и неточными условиями. Поэтому для качественного машинного анализа текстов недостаточно строгих алгоритмов с четкой логикой, требуются интеллектуальные методы, например, с использованием моделей машинного и глубокого обучения. В последнее время наблюдается рост производительности вычислительных машин, поэтому данное направление активно развивается, в результате чего регулярно появляются модели, способные воспринимать и анализировать тексты на сопоставимом с человеком уровне. Тем не менее обработка текстов по-прежнему остается вычислительно сложной задачей, поэтому наиболее актуальные исследования в данной сфере направлены не только на реализацию новых подходов и методов, но и на их оптимизацию и упрощение.

В данной работе рассматривается применение машины экстремального обучения для классификации текстов. Вместо классических функций активации для нейронов внутри модели используются кубические сплайны, значения параметров которых подбираются популяционным алгоритмом оптимизации с целью повышения точности классификации.

1. Модели и методы

1.1. Классификация

Одной из наиболее популярных моделей для решения задачи классификации, сочетающих в себе как скорость работы, так и точность производимых результатов, является машина экстремального обучения (Extreme Learning Machine, ELM), описанная в работе [1]. Модель ELM представляет собой полносвязную нейронную сеть прямого распространения, состоящую из входного, скрытого и выходного слоев, как это показано на рис. 1.

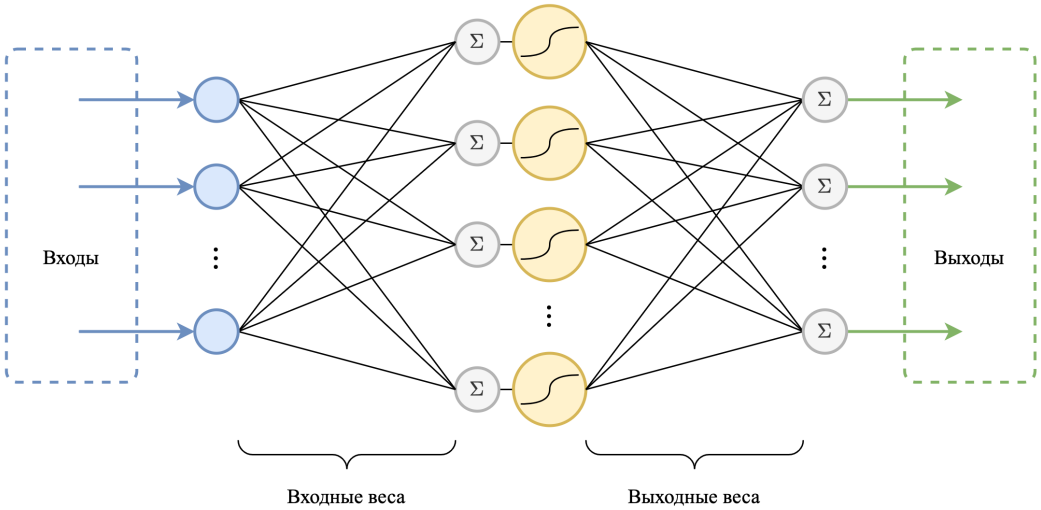


Рис. 1. Модель ELM

Ключевой особенностью данной модели является принцип ее обучения, отличающийся от классических подходов, основанных на градиентном спуске. Сначала веса связей между входным и скрытым слоями инициализируются случайными числами, которые затем не изменяются ни при обучении, ни при использовании модели. Далее для всех данных из обучающей выборки выполняется прямой проход до скрытого слоя, использующего нелинейную функцию активации. От получившегося результата вычисляется псевдообратная матрица по методу Мура — Пенроуза [2], которая умножается на матрицу правильных меток классов, чтобы получить веса связей между скрытым и выходным слоями.

Таким образом, процесс обучения модели ELM не является итеративным, что положительно сказывается на скорости ее работы. В работах [3, 4] показано, что, будучи одной из самых быстрых, данная модель также оказывается одной из самых точных в сравнении с классическими алгоритмами классификации.

Для активации нейронов скрытого слоя модели ELM может использоваться любая нелинейная функция, но чаще всего применяется так называемая сигмоида [1]. В данной работе рассматривается подход с использованием кубических сплайнов в качестве функций активации. Кубический сплайн представляет собой гладкую функцию, область определения которой состоит из конечного числа отрезков, каждый из которых соответствует некоторому кубическому многочлену. Для формирования сплайна необходимо задать набор точек, через которые будет проходить кривая, затем, составив и решив систему уравнений, найти коэффициенты для многочленов каждого из отрезков между заданными точками. Значение первых производных для многочленов в краевых точках принимается равным нулю, поскольку в данной работе поведение сплайна за их пределами моделируется как прямая, параллельная оси абсцисс. На рис. 2 представлен пример случайно сгенерированных кубических сплайнов для пяти нейронов скрытого слоя модели ELM.

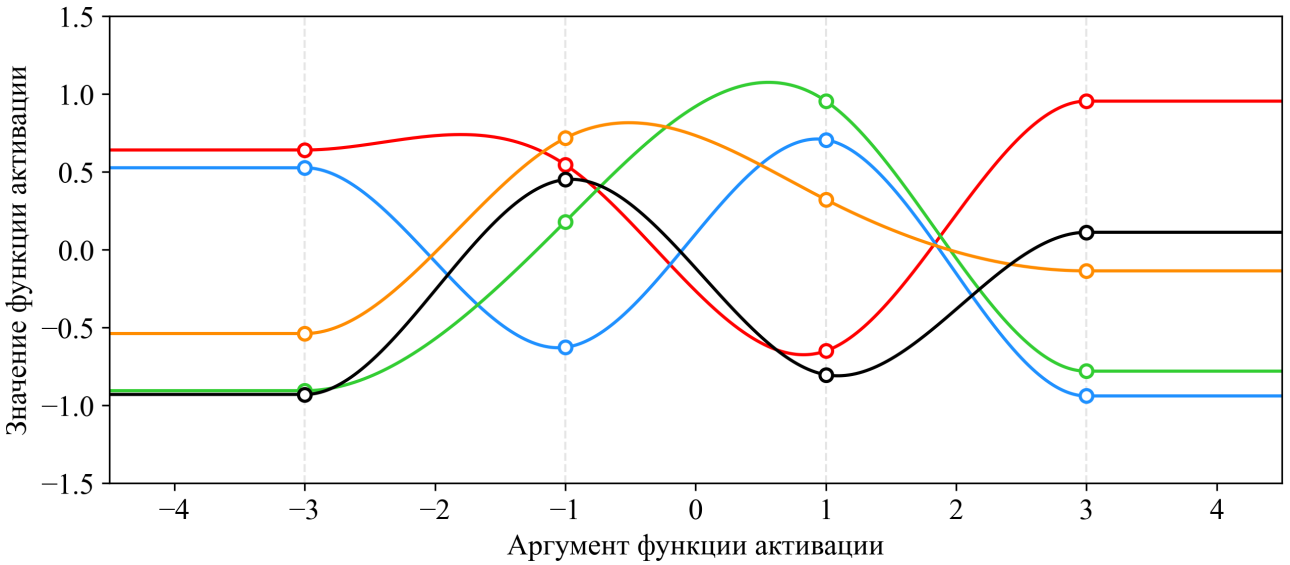


Рис. 2. Случайно сгенерированные кубические сплайны

Опорные точки для формирования сплайнов расставляются равномерно вдоль оси абсцисс на отрезке от минимального до максимального возможного значения, передаваемого каждому нейрону скрытого слоя. Координата каждой точки вдоль оси ординат выбирается из диапазона от –1 до 1.

1.2. Векторное представление текста

Ключевым шагом в реализации любого алгоритма для классификации текстов является их преобразование в числовой формат. В настоящее время известно несколько подходов для векторизации текстов, одним из которых является модель FastText [5].

Модель FastText реализует идею о дистрибутивной семантике, согласно которой семантические свойства слов или их частей определяются контекстом, в котором они появляются. Например, слова «автомобиль» и «такси» часто употребляются в схожем окружении слов, из чего можно сделать вывод об их семантической близости. Каждое слово в данной модели представляется в виде n-грамм, то есть частей, состоящих из небольшого числа символов, что позволяет учитывать морфологические свойства слов при формировании векторных представлений. Для векторизации целых предложений используется усреднение векторов всех слов, которые в него входят, с предварительным делением каждого из них на евклидову норму.

Для векторизации текстов в данной работе предлагается использовать готовые векторные представления, сформированные с помощью модели FastText [6]. Для их обучения использовались огромные наборы текстов, сформированные из статей онлайн-ресурса Wikipedia [7] и репозитория Common Crawl [8]. Каждый вектор состоит из 300 чисел, но в рамках данной работы размерность векторов была уменьшена до 50 с помощью метода главных компонент (Principal Component Analysis, PCA) [9].

1.3. Оптимизация кубических сплайнов

Для поиска наиболее подходящих координат опорных точек для кубических сплайнов, способных улучшить качество классификации, необходимо определить целевую функцию, чтобы перейти к задаче оптимизации. В данной работе для оценки координат опорных точек предлагается осуществлять 5-проходную кросс-валидацию модели ELM, использующей созданные по данным координатам сплайны в качестве функций активации, с вычислением какой-либо метрики качества решения задачи классификации и ее последующим усреднением по всем проходам.

Такая целевая функция не имеет явной и интерпретируемой аналитической формулы, поэтому ее оптимизация представляется нетривиальной задачей. Для решения подобных задач лучше всего подходят популяционные алгоритмы, принцип работы которых основан на разных эвристиках, чаще всего заимствованных из природы. Для оптимизации целевой функции в таких алгоритмах создается набор случайных решений, называемый популяцией. Ее отдельные индивиды в ходе последовательных итераций перемещаются по области поиска, обмениваясь информацией и постепенно сходясь к локальному или глобальному экстремуму. В отличие от традиционных математических подходов, популяционные алгоритмы не требуют от оптимизируемой функции непрерывности, существования производных или иных дополнительных сведений кроме возможности ее вычисления в произвольной точке. Популяционные алгоритмы часто используются для решения реальных практических задач, поскольку они способны работать в условиях неопределенности и высокой сложности имеющихся данных.

Одним из известных популяционных алгоритмов оптимизации является поиск косяком рыб (Fish School Search, FSS) [10]. Его главная идея заключается в моделировании поведения рыб, плавающих по аквариуму и взаимодействующих друг с другом в поисках как можно большего количества корма. В контексте решения задачи оптимизации косяк таких рыб представляет собой популяцию из точек в пространстве признаков, за аквариум принимается заданная область поиска, а количество корма определяется значением целевой функции. На рис. 3 представлено четыре основных этапа развития популяции в ходе одной итерации алгоритма FSS.

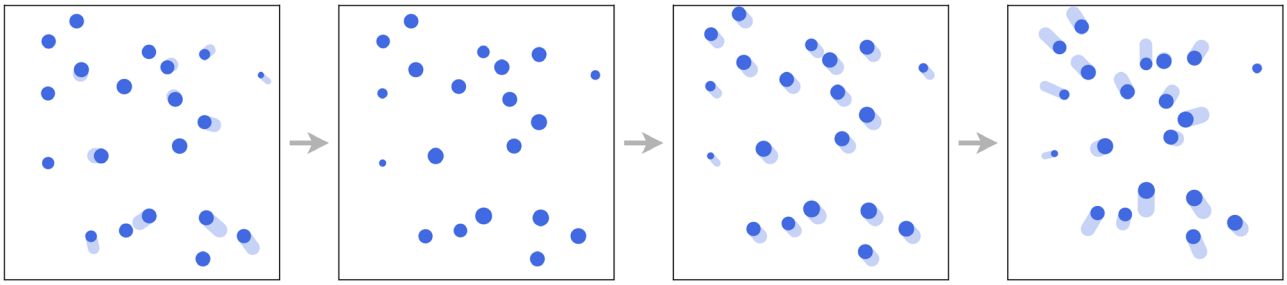


Рис. 3. Основные этапы одной итерации алгоритма FSS

На первом этапе рыбы делают индивидуальные шаги в случайных направлениях и остаются в новых позициях, если значение целевой функции в них лучше. В зависимости от достигнутых результатов на втором этапе рыбы увеличивают или уменьшают свой вес. На третьем этапе осуществляется коллективно-инстинктивное движение, в ходе которого все рыбы перемещаются в одном направлении, которое в среднем оказалось наиболее выгодным. Наконец, на четвертом этапе все рыбы подчиняются коллективно-волевому движению и перемещаются либо по направлению к общему центру масс, либо от него, в зависимости от того, улучшились ли результаты в сравнении с предыдущей итерацией.

В работе [11] была представлена модификация данного алгоритма, которая показала превосходство над оригиналом в ряде задач. Представленный авторами алгоритм ETFSS использует отображение тент в качестве генератора псевдослучайных чисел, а размеры шагов для индивидуальных и коллективно-волевых движений рыб уменьшаются по экспоненциальному закону с течением итераций.

В контексте решаемой задачи об оптимизации кубических сплайнов размерность пространства поиска напрямую зависит от количества нейронов в скрытом слое модели ELM. Если использовать всего три опорные точки для каждого сплайна, то уже с десятью нейронами размерность решаемой задачи достигнет тридцати. По этой причине имеет смысл рассмотреть возможность реализации островной модели для алгоритма оптимизации, поскольку такой подход показывает более качественные результаты на больших и сверхбольших размерностях.

Идея островной модели заключается в разбиении популяции на несколько отдельных групп, называемых островами. При этом каждый остров решает задачу независимо от других, но периодически они обмениваются друг с другом индивидами, участвуя в процессе миграции. В работе [12] была представлена модель мягкого острова (Soft Island Model, SIM). В рамках данной модели авторы предлагают подход, при котором во время миграции каждый агент с определенной вероятностью остается на своем острове и с обратной вероятностью покидает его. Данная вероятность является гиперпараметром алгоритма и не изменяется в ходе выполнения итераций. В данной работе предлагается модификация алгоритма ETFSS, основанная на SIM.

2. Проведение экспериментов

Для решения задачи классификации в данной работе рассматривается набор данных о рецензиях на рестораны [13]. Каждая рецензия в наборе данных состоит из текстового отзыва, числовой оценки по пятибалльной шкале, а также других сведений, например, дате или идентификаторе автора. Предлагается выполнить анализ тональности текстов, определив рецензии с оценкой выше трех положительными, а с оценкой ниже трех — отрицательными. Всего в наборе данных 10000 уникальных рецензий, но среди них наблюдается дисбаланс классов, а также отсутствие оценок для некоторых отзывов. По этой причине случайным образом было выбрано только 2000 положительных рецензий и столько же отрицательных. Получившийся набор данных из 4000 объектов далее был преобразован в набор векторов размерности 50 с помощью модели FastText, каждому из которых соответствует метка класса (0 для отрицательных рецензий, 1 — для положительных).

Для классификации использовалась модель ELM, имеющая 50 входов для векторизированных текстов, 100 нейронов в скрытом слое и 1 выход. Поскольку количество объектов обоих классов сбалансировано, было решено использовать долю правильных ответов модели в качестве метрики оценки точности классификации.

Все дальнейшие расчеты были выполнены с применением языка программирования Python в среде Jupyter Notebook. В ходе экспериментов использовался следующий компьютер: MacBook Pro 13 2017 A1708 (процессор: Intel(R) Core(TM) i5-7360U CPU 2.3 ГГц, 2.3 ГГц, 2 ядра; оперативная память: 8 Гб; 64-разрядная операционная система).

В качестве базовой модели для дальнейшего сравнения с предлагаемым подходом к оптимизации кубических сплайнов была реализована классическая модель ELM с сигмоидальной функцией активации. Данная модель была 1000 раз независимо обучена и протестирована 5-проходной кросс-валидацией с оценкой доли правильных ответов. В табл. 1 представлена статистика полученных результатов.

Таблица 1

Результаты оценки модели ELM с сигмоидальной функцией активации

Минимальное значение доли правильных ответов	0,8648
Максимальное значение доли правильных ответов	0,8875
Среднее значение доли правильных ответов	0,8768
Медианное значение доли правильных ответов	0,877

Затем для такой же конфигурации модели ELM, но уже с использованием кубических сплайнов в качестве функций активации, был запущен процесс их оптимизации с помощью островной модификации алгоритма ETFSS. Для каждого сплайна было использовано по 4 опорных точки, таким образом, размерность решаемой задачи составила 400. В табл. 2 представлены значения гиперпараметров алгоритма оптимизации.

Таблица 2

Значения гиперпараметров для алгоритма оптимизации

Количество итераций	50
Количество островов	3
Общий размер популяции	300 (по 100 на каждый остров при инициализации)
Вероятность агента остаться на своем острове	0,8
Максимальный размер шага при индивидуальном движении	0,25
Максимальный размер шага при коллективно-волевом движении	0,125

На рис. 4 представлена история глобального улучшения метрики качества с течением итераций алгоритма оптимизации.

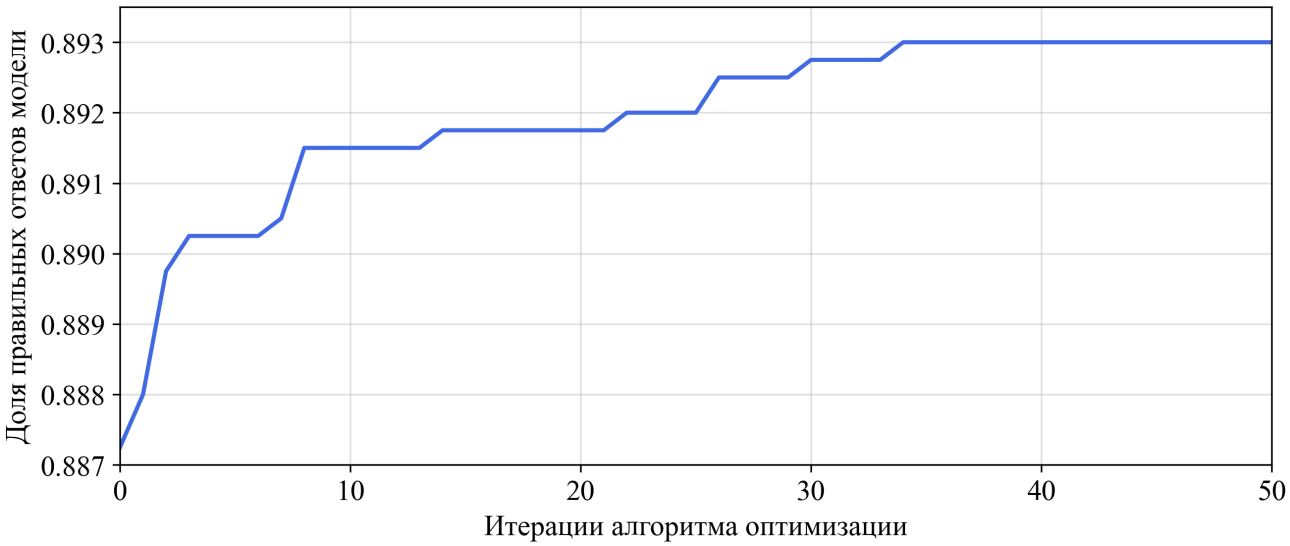


Рис. 4. Оптимизация целевой функции

По рис. 4 видно, что уже в течение первых 5 итераций алгоритма оптимизации модель ELM с кубическими сплайнами для активации скрытого слоя продемонстрировала результаты, превосходящие максимальную оценку из 1000 результатов тестирования такой же модели, но с использованием сигмоидальной функции активации. Отдельно стоит отметить и тот факт, что целевая функция уже при инициализации островного алгоритма ETFSS приняла значение, сопоставимое со средними показателями, полученными на классической модели.

Заключение

В данной работе была представлена модель ELM с кубическими сплайнами в качестве функций активации. Координаты опорных точек для построения сплайнов было предложено подбирать с помощью оптимизации целевой функции, в роли которой выступает 5-проходная кросс-валидация модели с вычислением доли правильных ответов.

Для решения задачи оптимизации была предложена и реализована островная модификация популяционного алгоритма ETFSS, основанная на SIM.

В рамках проведения экспериментов был рассмотрен набор данных о текстовых рецензиях на рестораны. После его предварительной обработки была сформулирована задача класси-

фикации текстовых отзывов по смысловому тону на положительные и отрицательные. Для числового представления текстов использовались готовые векторные представления, полученные с помощью модели FastText.

В процессе оптимизации кубических сплайнов для модели ELM была достигнута точность, соответствующая 89,3 % правильных ответов, что превышает максимальный результат, полученный с использованием классической сигмоидальной функции активации.

Дальнейшие исследования могут быть направлены на улучшение процесса подбора значений параметров для кубических сплайнов.

Литература

1. *Guang-Bin Huang*. Extreme learning machine: Theory and applications / Guang-Bin Huang, Qin-Yu Zhu, Chee-Kheong Siew // Neurocomputing. – 2006. – Vol. 70. – P. 489–501. – DOI 10.1016/j.neucom.2005.12.126.

2. *Penrose R.* On best approximate solutions of linear matrix equations / Penrose R. // Mathematical Proceedings of the Cambridge Philosophical Society. – 1956. – Vol. 52(1). – P. 17–19. – DOI 10.1017/S0305004100030929.

3. *Gupta C. A.* Machine Learning Techniques and Extreme Learning Machine for Early Breast Cancer Prediction / C. A. Gupta, N. S. Gill, P. N. Sing, Gill, V Conclusion // International Journal of Innovative Technology and Exploring Engineering. – 2020. – Vol. 9(4). – P. 163–167. – DOI 10.35940/ijitee.D1411.029420.

4. *Mercaldo F.* Extreme Learning Machine for Biomedical Image Classification: A Multi-Case Study / F. Mercaldo, L. Brunese, A. Santone, F. Martinelli, M. Cesarelli // EAI Endorsed Transactions on Pervasive Health and Technology. – 2024. – Vol. 10. – DOI 10.4108/eetpht.10.5542.

5. *Bojanowski P.* Enriching Word Vectors with Subword Information (Version 2) / P. Bojanowski, E. Grave, A. Joulin, T. Mikolov // arXiv. – 2016. – DOI 10.48550/ARXIV.1607.04606.

6. *Grave E.* Learning Word Vectors for 157 Languages (Version 2) / E. Grave, P. Bojanowski, P. Gupta, A. Joulin, T. Mikolov // arXiv. – 2018. – DOI 10.48550/ARXIV.1802.06893.

7. Wikipedia – свободная энциклопедия: официальный сайт. – URL: https://www.wikipedia.org/ (дата обращения: 20.11.2024).

8. Common Crawl repository: официальный сайт. – URL: https://commoncrawl.org/ (дата обращения: 20.11.2024).

9. *Maćkiewicz A.* Principal components analysis (PCA) / Maćkiewicz, A., & Ratajczak, W. // Computers & Geosciences. – 1993. – Vol. 19(3). – P. 303–342. – DOI 10.1016/0098-3004(93)90090-R.

10. *Filho C. J.* A novel search algorithm based on fish school behavior / C. J. Filho, F. B. Lima-Neto, A. Lins, A. I. Nascimento, M. P. Lima // 2008 IEEE International Conference on Systems, Man and Cybernetics. – 2008. – P. 2646-2651. – DOI 10.1109/ICSMC.2008.4811695.

11. *Demidova L. A.* A Study of Chaotic Maps Producing Symmetric Distributions in the Fish School Search Optimization Algorithm with Exponential Step Decay / L. A. Demidova, A. V. Gorchakov // Symmetry. – 2020. – Vol. 12(5). – P. 784. – DOI 10.3390/sym12050784.

12. *Akhmedova S.* Soft Island Model for Population-Based Optimization Algorithms / S. Akhmedova, V. Stanovov, E. Semenkin // Lecture notes in computer science. – 2018. – P. 68–77. – DOI 10.1007/978-3-319-93815-8_8.

13. 10000 Restaurant Reviews // Kaggle: [сайт]. – 2023. – URL: https://www.kaggle.com/datasets/joebeachcapital/restaurant-reviews (дата обращения 20.11.2024).

УДК 004.032.26

ЧАТ-БОТ НА ОСНОВЕ НЕЙРОННОЙ СЕТИ ДЛЯ ОПРЕДЕЛЕНИЯ ЗАБОЛЕВАНИЯ ЛЕГКИХ

М. О. Дудич, Н. Н. Максимова

Амурский государственный университет

Аннотация. В данной статье описывается процесс разработки и обучения свёрточной нейронной сети для классификации патологий лёгких на основе рентгеновских снимков из набора данных. Модель была обучена с точностью 94.97 %; кроме того, анализировались метрики precision и recall, выявившие некоторые ошибки, в частности при диагностике пневмонии. Для практического применения был создан телеграм-бот, позволяющий пользователям загружать рентгеновские снимки и получать результаты диагностики в реальном времени. Это делает технологию более доступной для использования в медицинской практике и улучшает качество оказания медицинских услуг. Исследование демонстрирует потенциал методов глубокого обучения в области медицинской диагностики.
Ключевые слова: искусственный интеллект, глубокое обучение, свёрточные нейронные сети, набор рентген-снимков, медицинская диагностика, Python, телеграм-бот.

Введение

Искусственный интеллект (ИИ) — это технология, которая стала очень быстро развиваться с появлением больших вычислительных мощностей. В современном мире он играет важнейшую роль в таких сферах как наука и бизнес. С повышением сложности нетривиальных задач и объёма необходимых данных методы машинного обучения становятся всё более востребованными. ИИ активно применяется в различных областях, включая медицину, где он помогает анализировать большие массивы данных, диагностировать заболевания и предлагать рекомендации по лечению. Благодаря возможности решать сложные задачи, для которых сложно определить чёткий алгоритм, ИИ автоматизирует процессы в прикладных и научных сферах, включая медицинскую.

Наиболее эффективным направлением в ИИ сегодня является обучение на глубоких нейронных сетях. Различные типы нейронных сетей решает конкретные прикладные задачи. Для анализа изображений в настоящее время используются, как правило, свёрточные нейронные сети.

Цель данной работы — разработка и обучение свёрточной нейронной сети для определения заболеваний лёгких по рентгеновским снимкам а также создание удобного интерфейса для её использования.

1. Описание набора данных

Набор данных — подборка рентген-снимков лёгких, разделённых на 9 классов. В наборе присутствуют изображения лёгких без патологий, а также изображения легких с 8 патологиями, такими как пневмония, обструктивные нарушения и другое. Данные взяты из открытого набора, размещённого на платформе Kaggle. В наборе представлено 6743 изображения. С информацией о каждом классе можно ознакомиться в табл. 1.

2. Архитектура нейронной сети

Свёрточные нейронные сети (СНС) считаются наиболее эффективным инструментом в задачах анализа изображений [2]. Особенность таких нейронных сетей заключается в том,

что обучаемые веса расположены в ядре свёртки, которое меньше по размеру в сравнении с исходным изображением. Это позволяет сети выделять ключевые признаки, такие как грани, контуры и края объектов.

Один свёрточный слой СНС включает:

1. одну или несколько свёрток с функциями активации;
2. слой пуллинга.

Между такими свёрточными слоями рекомендуется вставлять слои Dropout для борьбы с переобучением модели.

Таблица 1

Исходные данные

Номер набора	Количество изображений	Метка класса (норма или тип патологии)
0	1340	Здоровые лёгкие
1	1060	Пневмония
2	678	Повышенная плотность
3	629	Пониженная плотность
4	644	Обструкция лёгких
5	594	Инфекционные заболевания
6	658	Физические травмы
7	596	Измененное средостение
8	544	Нарушение развития лёгких

Как правило, сеть состоит из нескольких таких слоёв. Свёрточный слой применяет операцию свёртки к входным данным, создавая карту признаков. Далее следует слой пуллинга, который извлекает наиболее значимые параметры — обычно максимальные, либо средние значения. Результаты всех свёрточных слоёв, по сути — матриц, которые значительно меньше, чем обрабатываемое изображение, передаются на обычную полносвязную нейросеть, которая состоит из нескольких внутренних слоев и выходного слоя, который, в свою очередь, решает поставленную задачу.

Для задачи классификации изображений была создана СНС, с архитектурой которой можно ознакомиться в работе [3].

Самые первые три слоя, каждый из которых состоит из свертки, пуллинга и активации, образуют карту признаков, извлекают максимальные значения и по объёму уменьшают размер входных данных для следующих слоёв. Четвертый слой разворачивает полученную карту признаков в соответствующий вектор, который затем подаётся на последующие полносвязные слои. Последние слои с функцией активации softmax используются для классификации.

Потом начинается обучение модели. Параметры обучения включают: функция потерь — категориальная кроссэнтропия, оптимизатор — Nadam, метрика оценки — Accuracy, количество эпох — 30, размер мини-выборки — 100 изображений. Для валидации выделено 20 % данных из обучающего набора.

3. Анализ качества прогнозирования

В результате обучения модель достигла точности 94.97 % по метрике accuracy, что можно считать вполне приемлемым результатом. История обучения представлена на рис. 1.

При этом, метрика accuracy не является достаточной для качественной оценки модели. Для полноценного анализа необходимо дополнительно использовать другие метрики — например precision с recall и матрицу ошибок. Метрика precision показывает, сколько из предсказанных

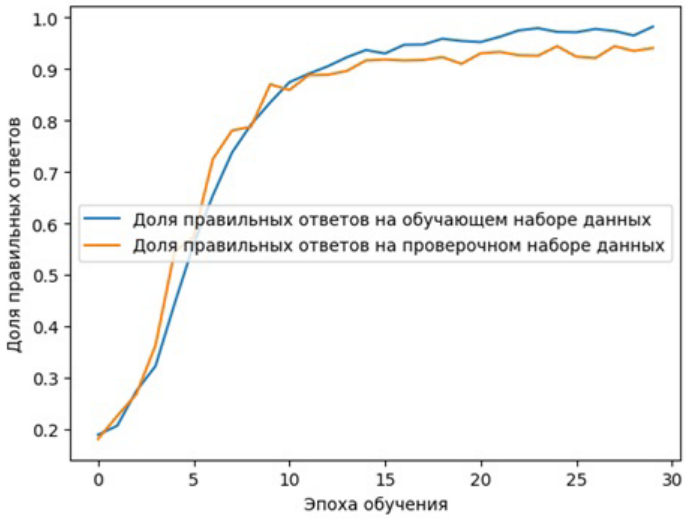


Рис. 1. История обучения нейросетевой модели

моделью объектов конкретного класса действительно принадлежат этому классу. Метрика recall отражает, какую долю от всех реальных объектов данного класса модель правильно отнесла к этому классу. Матрица ошибок (confusion matrix) демонстрирует соотношение верных и ошибочных предсказаний по всем классам. Она представлена на рис. 2, где приняты следующие обозначения: ВП — высокая плотность, НП — низкая плотность.

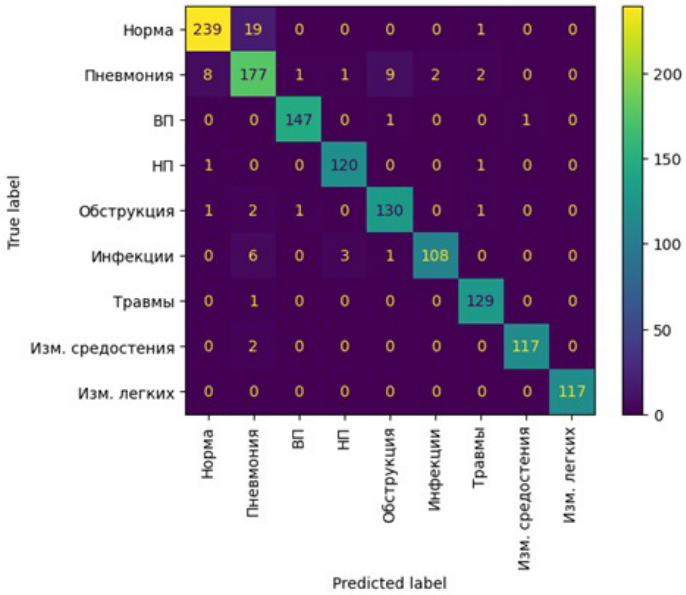


Рис. 2. Confusion matrix для обученной модели на тестовом наборе данных

Несмотря на то, что в общем результат является приемлемым, стоит отдельно обратить внимание на первый столбец матрицы ошибок. Модель ошиблась и классифицировала восемь рентгеновских снимков лёгких с пневмонией, один снимок с низкой плотностью и один с обструкцией как изображения здоровых лёгких. Хотя такие ошибки в целом считаются малыми, в медицине они являются критичными, так как приводят к пропуску патологий, которые могу усугубить здоровье пациента. Обратная ситуация — когда снимки здоровых легких классифицируются как патологии — также нежелательная, но менее опасная ситуация.

На рис. 3 показан классификационный отчёт (classification report). Видно, что изменения лёгких классифицируются с точностью 100 %. Это может быть связано с:

1. выраженным внешним отличием от других классов;

2. ограниченным количеством разнообразных изображений с изменёнными лёгкими.

По тем же причинам можно объяснить высокий уровень точности классификации «ВП», «Травмы» и «Изменения средостения». Самый низкий уровень всех метрик у модели вызывает второй класс «Пневмония». Почти 14 % всех экземпляров, помеченных как класс «Пневмония», оказываются представителями других патологий. Кроме этого, у модели не получилось правильно определить 11 % изображений, относящихся к классу «Пневмония». Эти факты показывают — настоящее качество обученной модели может быть несколько ниже, чем если судить по метрике accuracy.

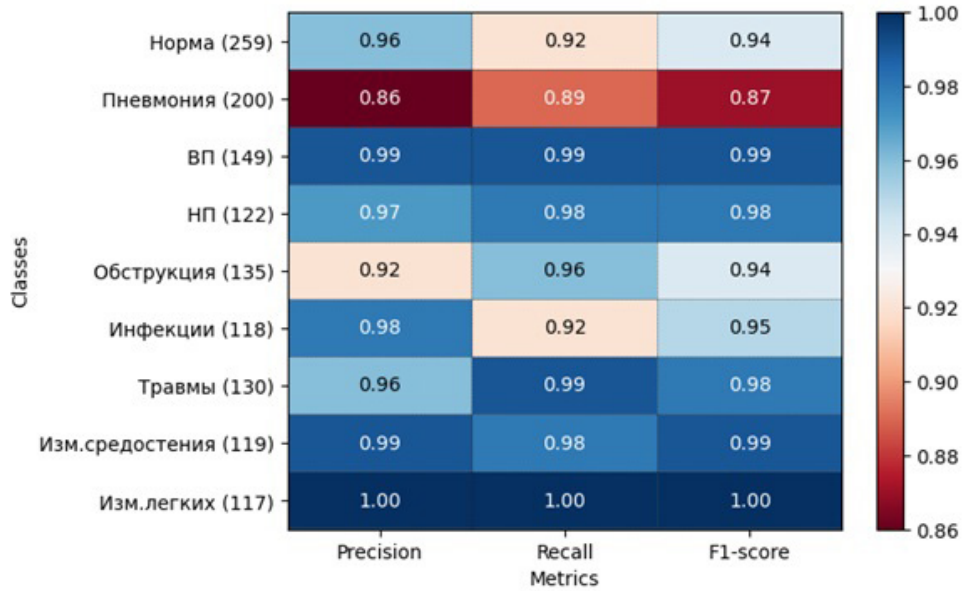


Рис. 3. Classification report для обученной модели на тестовом наборе

4. Чат-бот на основе обученной нейронной сети

Для практического использования нейросети был создан интерфейс в виде чат-бота в мессенджере Telegram (далее — телеграм-бот).

Выбор данного способа взаимодействия с нейронной сетью обуславливается несколькими причинами:

1) простота создания: для создания рабочего телеграм-бота не требуются дорогое оборудование или специальные умения — только базовые навыки программирования. Такой вариант подходит для как предварительной демонстрации работы нейронной сети, так и, при дальнейших улучшениях, для полноценного использования в медицинской сфере.

2) простота эксплуатации: в отличие от существующих решений, требующих сложных установок и специализированного ПО, телеграм-бот предлагает максимально простой и доступный интерфейс через любой смартфон с выходом в интернет. Это значительно расширяет доступ к технологии, делая её удобной и для крупных медицинских учреждений, и для небольших клиник в удалённых регионах, где может отсутствовать доступ к сложному оборудованию и квалифицированным специалистам.

Программный код написан с помощью Telegram Bot API для Python [4]. Пользователь совершает некоторые действия — формирует команду, либо составляет запрос, который передается на программное обеспечение, расположенное на серверах разработчиков. В качестве посредника используется анонимный сервер Telegram. Он отвечает за обработку шифрования и обратную связь между телеграм-ботом и пользователем.

После запуска программного кода, бот готов принимать сообщения от пользователей. Примеры использования бота для распознавания рентген-снимков легких представлен на рис. 4 и рис. 5.

Бот может работать, пока запущен программный код. При размещении на удаленном сервере, телеграм-бот может работать круглосуточно, принимая снимки от многих людей. Количество человек, обслуживаемых ботом, зависит от мощностей аппаратного обеспечения.

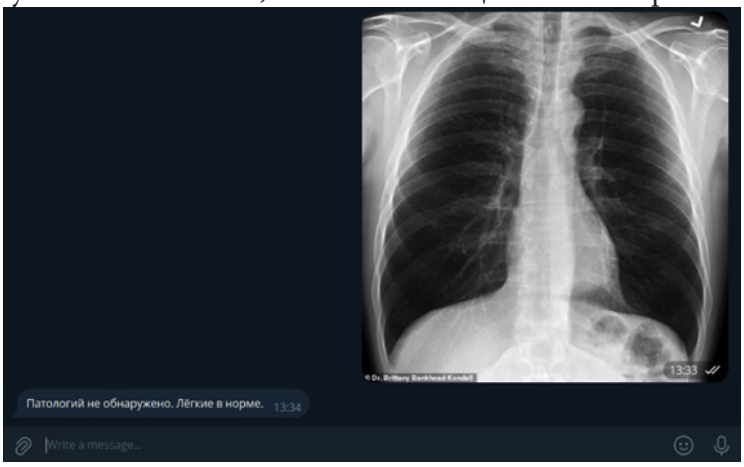


Рис. 4. Распознавание здоровых легких с помощью телеграм-бота

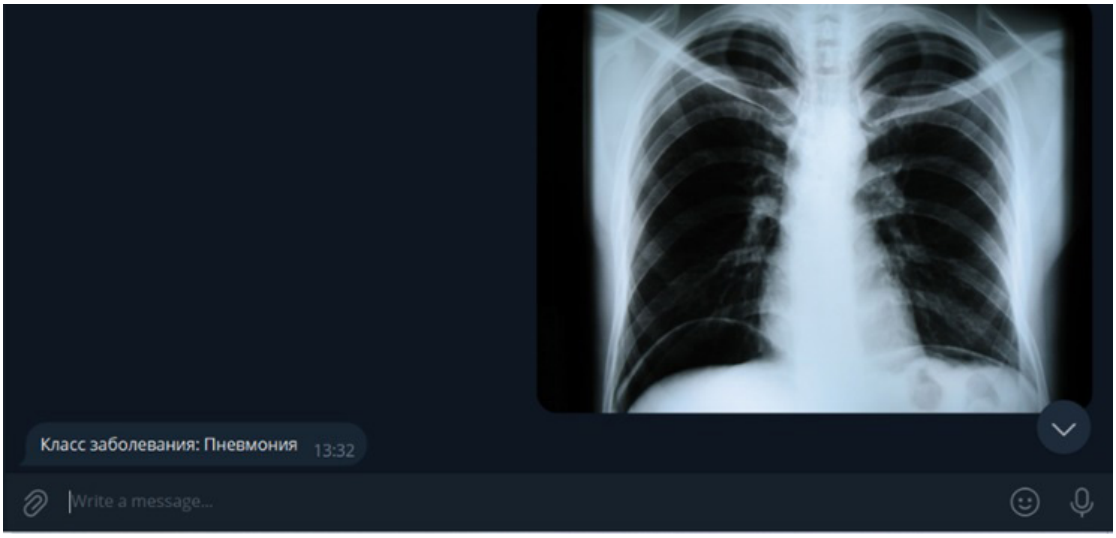


Рис. 5. Распознавание легких с пневмонией

Заключение

В рамках проведенной работы была обучена нейронная сеть по набору рентген-снимков легких, а также был создан интерфейс в виде бота мессенджера телеграм для работы с данной нейронной сетью.

Изначальный набор данных включает более 6000 изображений, на которых представлены легкие без патологий или с патологиями, относящимися к одному из восьми классов. В ходе экспериментов была выбрана архитектура и оптимизированы гиперпараметры нейронной сети, обеспечивающие максимальную точность распознавания на тестовых данных. Для разработки нейронной сети использовались библиотеки TensorFlow и Keras на языке программирования Python в среде Google Colaboratory [5].

Кроме того, был создан удобный интерфейс для взаимодействия с обученной моделью в виде телеграм-бота. Бот позволяет пользователям загружать снимки легких и получать резуль-

тат диагностики, что делает разработку более доступной для практического использования и облегчает внедрение модели в медицинскую практику. Такая интеграция способствует оперативной и удаленной оценке состояния легких, что может быть полезно для врачей и пациентов.

Благодарности

Работа выполнена при финансовой поддержке Минобрнауки РФ (проект № 122082400001-8).

Литература

1. X-ray Lung Diseases Images (9 classes) [Электронный ресурс]. URL: https://www.kaggle.com/datasets/fernando2rad/x-ray-lung-diseases-images-9-classes (дата обращения: 10.01.2024)

2. *Гудфеллоу Я.* Глубокое обучение / Я. Гудфеллоу, И. Бенджио, А. Курвилль. Перевод с английского А.А. Слинкина. – 2-е изд. – М. : ДМК Пресс, 2018. – 652 с.

3. *Дудич М. О.* Свёрточные нейронные сети в задачах диагностики заболеваний легких / М. О. Дудич, Н. Н. Максимова // Материалы V региональной научной-практической конференции «ТОГУ-СТАРТ: фундаментальные и прикладные исследования молодых» (Хабаровск, 16-20 апреля 2024 года). – Хабаровск : Издательство Тихоокеанского государственного университета, 2024. – С. 138–144.

4. Telegram Bot API [Электронный ресурс]. URL: https://core.telegram.org/bots/api (дата обращения 28.10.2024)

5. Google Colaboratory [Электронный ресурс]. URL: https://colab.google (дата обращения: 09.10.2023)

УДК 004.8

ИССЛЕДОВАНИЕ КАЧЕСТВА ВОДЫ В РАЗНЫХ РАЙОНАХ МОСКВЫ С ПРИМЕНЕНИЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

И. В. Есипов, А. Д. Личагина, А. Л. Шкерин

МИРЭА – Российский технологический университет

Аннотация. В данной работе проведен анализ качества воды с целью кластеризации водоподготовительных станций. Данные были собраны с официального сайта АО «МОСВОДОКАНАЛ» и приведены к нормализованному виду. Первоначально была выполнена кластеризация с фиксированным количеством кластеров определённым числом водоподготовительных станций Москвы, а следующим шагом была проведена кластеризация с автоматическим определением оптимального числа кластеров. Результаты показали, что оптимальное число кластеров отличается от первоначально заданного.
Ключевые слова: качество воды, машинное обучение, кластеризация, водоподготовка, хлорирование, водоподготовительные станции, метод локтя, силуэтный коэффициент.

Введение

Вода — неотъемлемая часть человеческой жизни. Благодаря воде свой функционал поддерживают предприятия, промышленность, точки общественного питания, различные объекты инфраструктуры и городские квартиры. Чтобы обеспечить население питьевой водой, необходимо очистить воду, доведя ее показатели до гигиенических нормативов, так как ее забор идет из рек и водохранилищ. Очисткой воды занимаются водоподготовительные станции, в Москве их четыре: Рублевская, Западная, Северная и Восточная [1]. По оценке экспертов, в совокупности эти станции выпускают более шести миллионов кубометров чистой воды в сутки. В данной работе рассматривается вопрос «слепой» кластеризации воды по местам ее очистки.

1. Сбор и подготовка данных

Данные о качестве воды были собраны с официального сайта АО «МОСВОДОКАНАЛ». Согласно информации с сайта были выбраны основные показатели, влияющие на качество воды (табл. 1).

Таблица 1

Основные показатели качества воды

Наименование	Обозначение
Водородный показатель, pH	Ед.
Железо общее	мг/л
Запах при 20°C	баллы
Запах при 60°C	баллы
Мутность	мг/л
Остаточный хлор	мг/л
Цветность	градус

2. Нормализация данных

Для корректного проведения кластеризации данные были нормализованы по формуле:

$$x_{норм} = \frac{x - x_{\min}}{x_{\max} - x_{\min}},$$

где x — исходное значение показателя,

$x_{\min}$ — минимальное значение показателя в выборке,

$x_{\max}$ — максимальное значения показателя в выборке соответственно.

3. Методика кластеризации

3.1. Кластеризация при k = 4

Первоначально использован алгоритм k-средних с заданным числом кластеров $k = 4$. Цель — проверить, соответствует ли эмпирически заданное число кластеров структурированной группировке данных.

Формула функции расстояния для метода k-средних:

$$J = \sum_{i=1}^{k} \sum_{j=1}^{n_i} |x_j^{(i)} - \mu_i|^2,$$

где μ_i — центр кластера i,

$x_j^{(i)}$ — j-й объект кластера i.

3.2. Автоматическое определение числа кластеров

Для определения оптимального числа кластеров применены методы:

– Метод локтя (Elbow Method)

– Средний силуэтный коэффициент

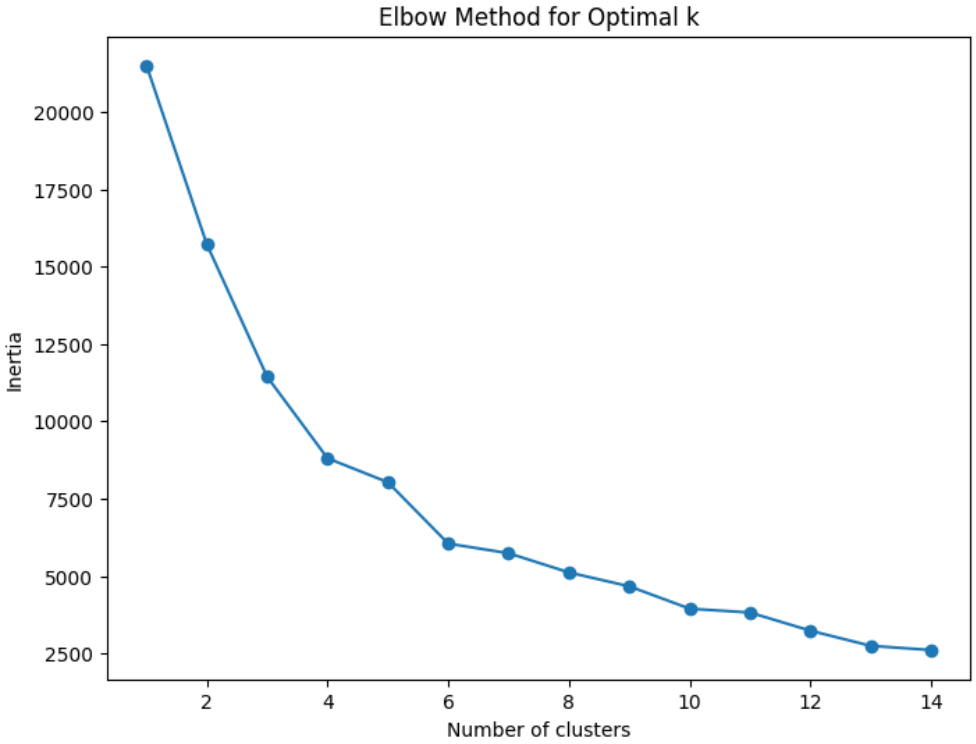


Рис. 1. График зависимости суммы квадратов ошибок от числа кластеров —Метод локтя

На графике зависимости суммы квадратов ошибок от числа кластеров наблюдается заметное снижение инерции до $k = 5$. После этой точки темп уменьшения инерции значительно

замедляется, что подтверждает выбор $k = 5$ как оптимального числа кластеров. Однако также стоит отметить, что инерция продолжает немного снижаться и после $k = 5$, что указывает на возможность дальнейшего разделения данных, если это необходимо для более детального анализа. Метод локтя чётко указывает на $k = 5$ как оптимальную точку сглаживания, где достигается баланс между компактностью кластеров и объяснением структуры данных.

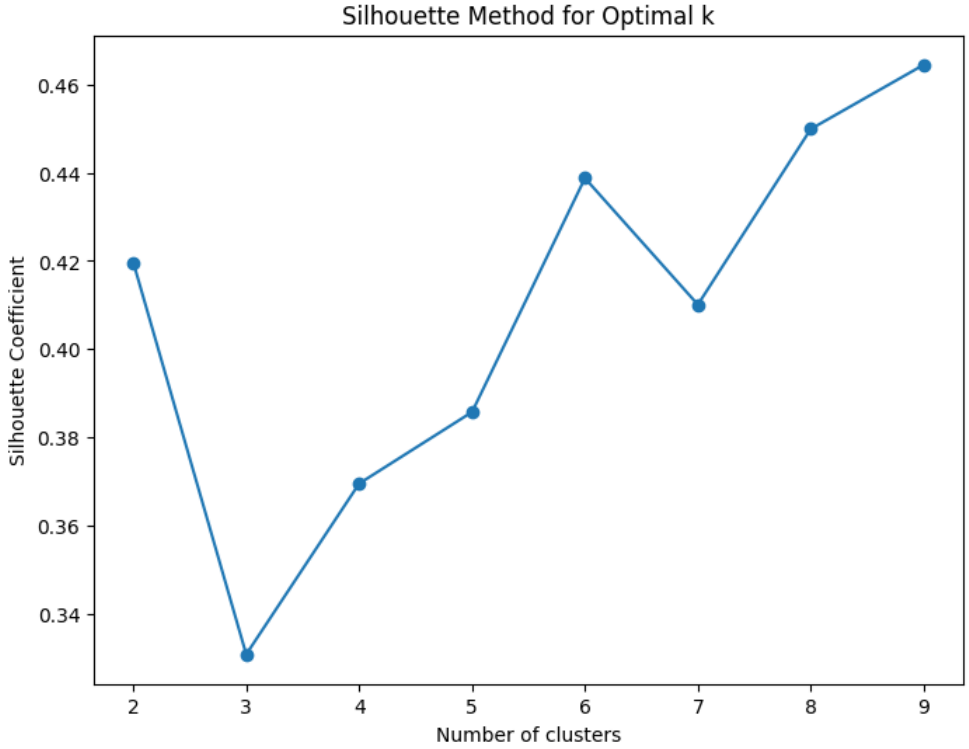


Рис. 2. График среднего силуэтного коэффициента в зависимости от числа кластеров

На графике среднего силуэтного коэффициента максимальное значение достигается при $k = 8$, а не $k = 5$, как это ранее утверждалось. Это говорит о том, что компактность и разделимость кластеров лучше при $k = 8$. Однако выбор $k = 5$ может быть оправдан при условии, что дальнейшее деление не даёт значимых улучшений для задачи анализа. Хотя коэффициент силуэта достигает пика при $k = 8$, выбор $k = 5$ может быть обоснован задачами исследования, поскольку он даёт приемлемую группировку при меньшем числе кластеров. Это снижает сложность анализа и интерпретации.

4. Результаты и обсуждение

4.1. Результаты при k = 4

Кластеризация при $k = 4$ показала следующую группировку объектов, результаты которых приведены на рис. 3.

Анализ показал неоднородность и пересечение кластеров при $k = 4$. Это может свидетельствовать о недостаточной разделимости данных при фиксированном числе кластеров. При фиксированном числе кластеров $k = 4$ наблюдаются недостатки в группировке, что указывает на необходимость уточнения параметров.

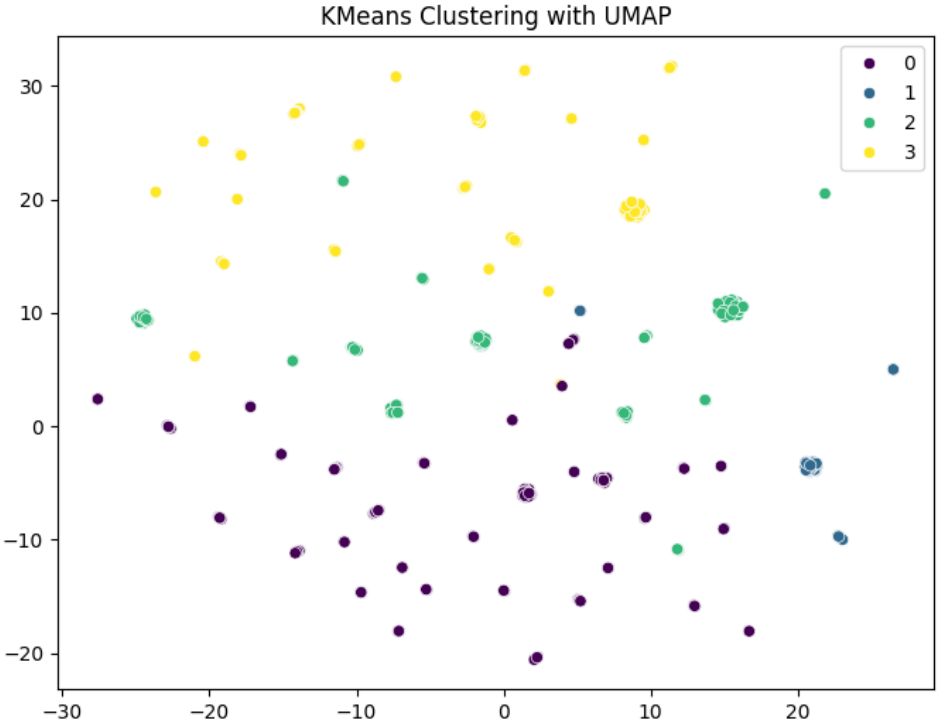


Рис 3. Распределение объектов по кластерам при k = 4

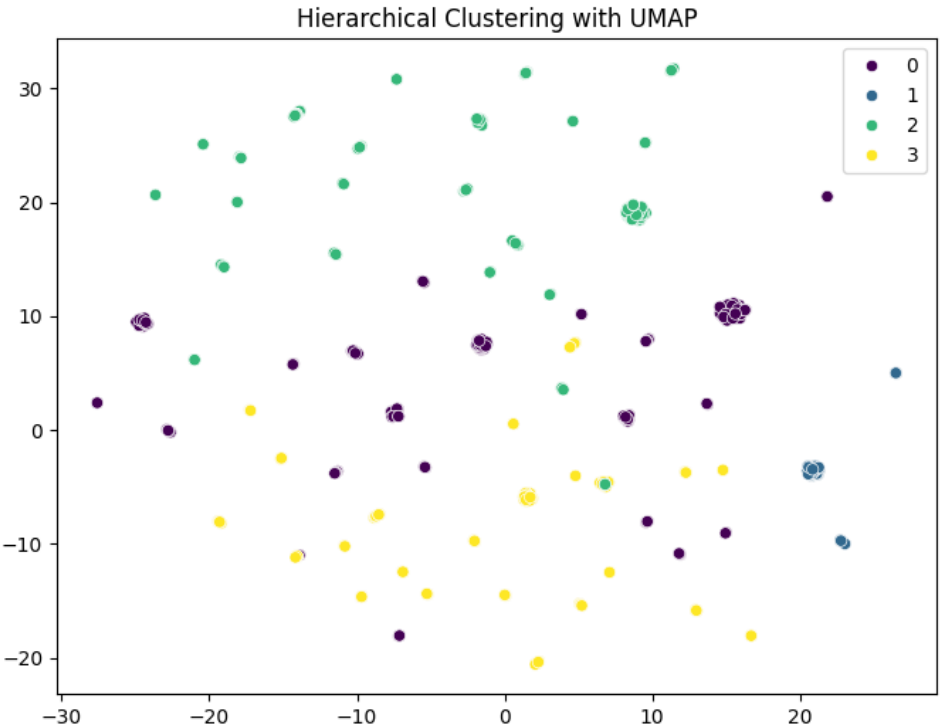


Рис 4. Распределение объектов по иерархическим кластерам при k = 4

4.2. Оптимальное число кластеров

Методы определения оптимального числа кластеров указали на то, что оптимальным является $k = 5$:

– Метод локтя показывает сглаживание кривой при $k = 5$.

– Максимальный средний силуэтный коэффициент достигается при $k = 5$.

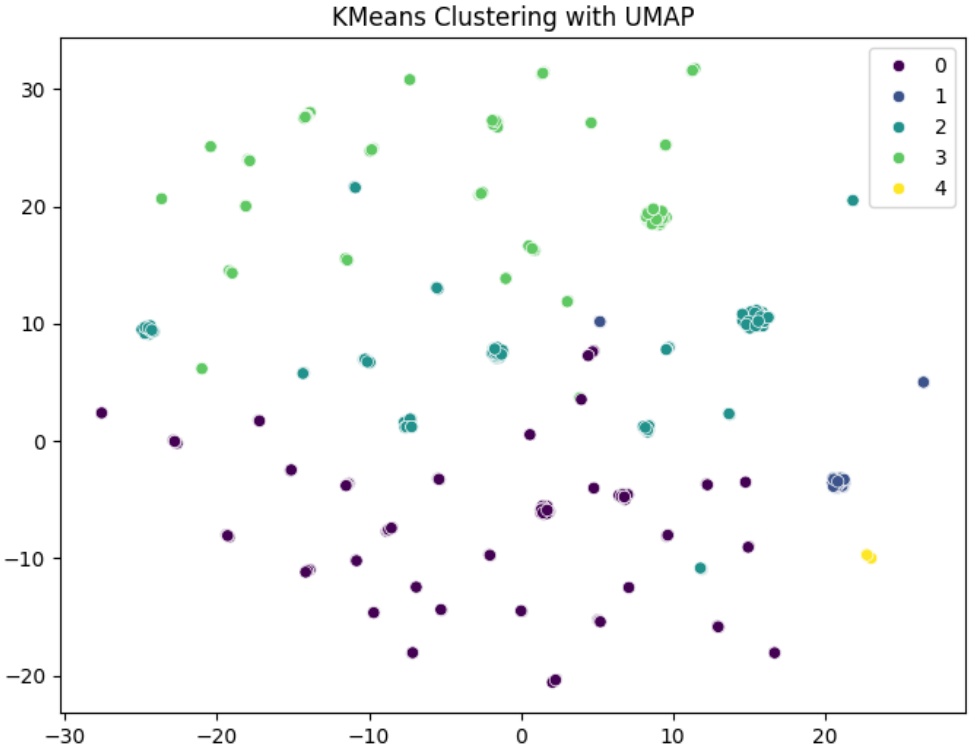


Рис. 5. Визуализация кластеров при оптимальном k = 5

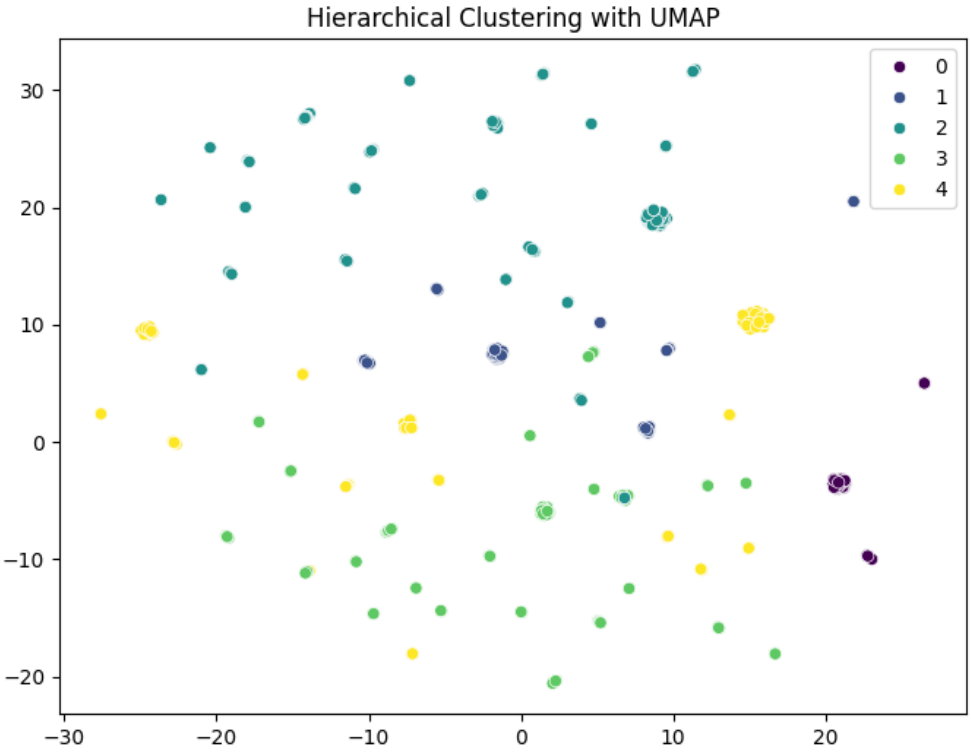


Рис. 6. Визуализация иерархических кластеров при оптимальном k = 5

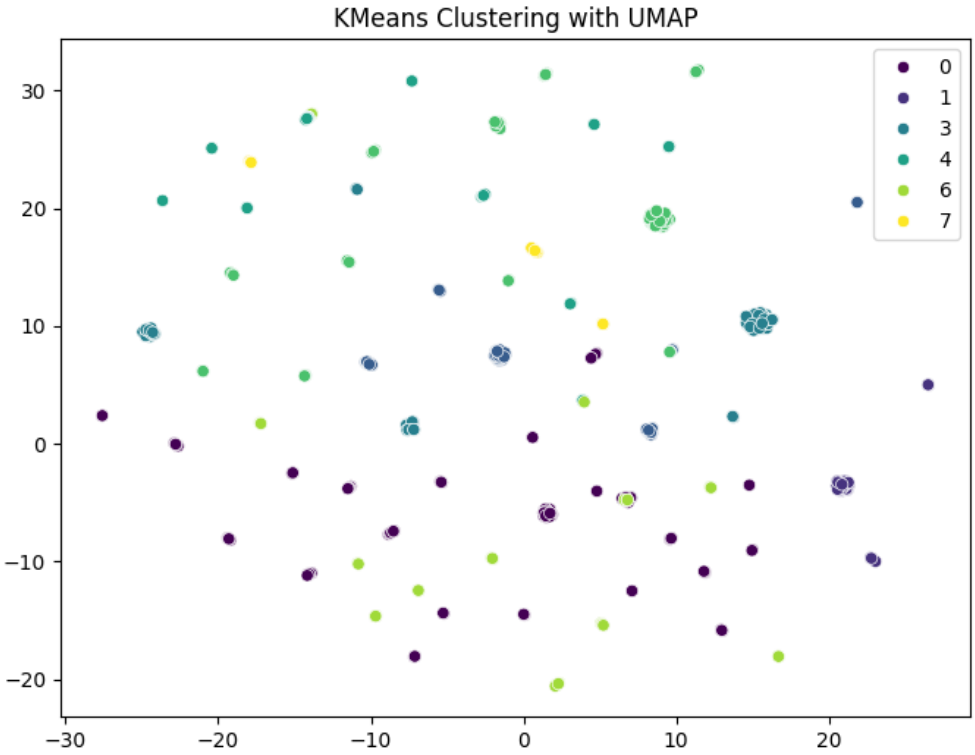


Рис. 7. Визуализация кластеров при оптимальном k = 8

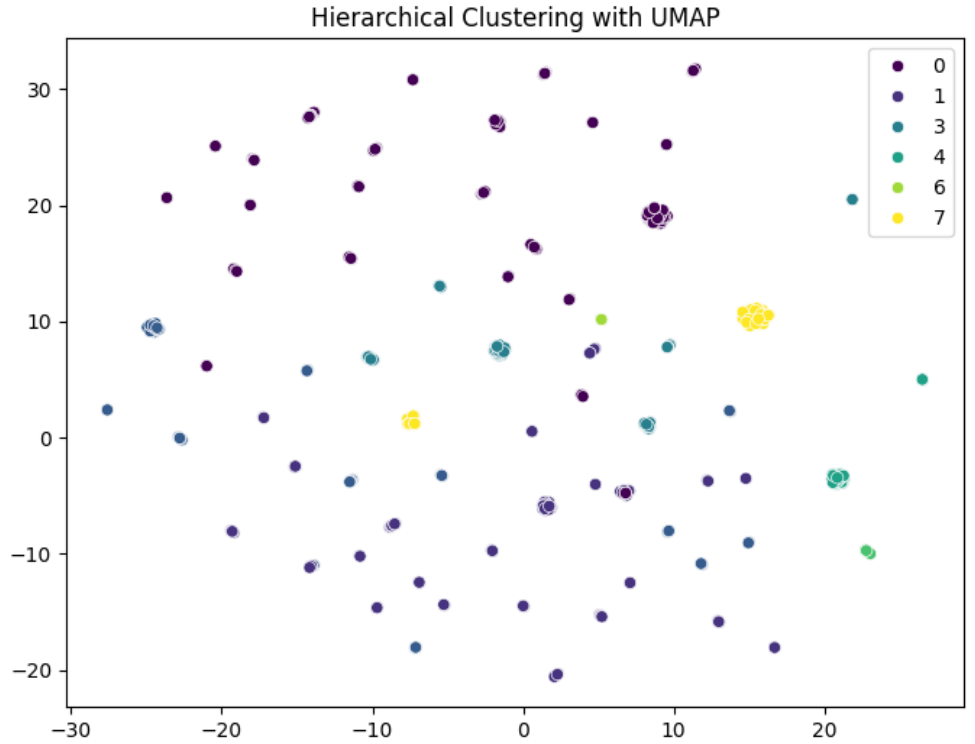


Рис. 8. Визуализация иерархических кластеров при оптимальном k = 8

4.3. Обсуждение результатов

Полученные результаты свидетельствуют о том, что очистные сооружения можно разделить на 5 групп с схожими характеристиками качества воды. Это может быть связано с различиями в технологиях очистки или географическим расположением.

Заключение

Проведенный анализ показал, что автоматическое определение числа кластеров позволяет более точно сгруппировать данные о качестве воды. Оптимальное число кластеров $k = 5$ лучше отражает структуру данных, чем первоначально заданное $k = 4$. Результаты исследования могут быть использованы для улучшения работы очистных сооружений и управления качеством воды.

Литература

1. МОСВОДОКАНАЛ. Официальный сайт. URL: https://www.mosvodokanal.ru (дата обращения: 07.11.2024).

2. *Hastie T., Tibshirani R., Friedman J.* The Elements of Statistical Learning. Springer. – 2009. – 745 p.

3. *Kaufman L., Rousseeuw P. J.* Finding Groups in Data: An Introduction to Cluster Analysis. – Wiley-Interscience, 1990. – 368 p.

4. *Ковалёв В. А., Ковалёва А. В.* Кластерный анализ в задачах экологии и охраны окружающей среды // Вестник Воронежского государственного университета. Серия: Химия. Биология. Фармация. 2018. № 2. С. 45–50.

5. *Иванов И. И., Петров П. П.* Применение методов машинного обучения для оценки качества воды // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2019. – № 3. – С. 112–118.

6. *Смирнов С. С., Кузнецова Е. Е.* Анализ качества воды с использованием методов кластеризации // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2020. – № 4. – С. 89–95.

УДК 004.89, 517.977

ДИНАМИЧЕСКАЯ ОПТИМИЗАЦИЯ СКОРОСТИ МИГРАЦИИ ДАННЫХ В МНОГОПАРАМЕТРИЧЕСКОЙ СРЕДЕ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

И. В. Замятин, К. С. Некрасов

Воронежский государственный университет

Аннотация. Рассматривается задача миграции больших объемов данных с гарантиями сохранности и отказоустойчивости. Миграция выполняется на мощностях заказчика с помощью специального программного обеспечения MaaS (Migration as a Service) и по определению включает интеграцию с внешними программно-аппаратными комплексами. Данные факторы являются источником нестабильности общего процесса миграции. Задача миграции рассматривается как задача управления по прогнозирующей модели (MPC). Постулируется существование стратегий поведения ПО MaaS, которые улучшают суммарную производительность миграции. Выбор наиболее предпочтительной стратегии поведения в текущий момент времени предполагается с учетом нечетких и заранее неизвестных характеристик внешней среды. В статье представлены подходы к адаптивной оптимизации производительности ПО MaaS на основе использования моделей и методов машинного обучения.

Ключевые слова: оптимизация, машинное обучение, миграция данных, управление по прогнозирующей модели, model predictive control, MPC, обучение с подкреплением, reinforcement learning.

Введение

В статье рассматривается задача оптимизации производительности программного продукта, реализующего сервис миграции данных — Migration as a Service (далее — ПО MaaS). ПО MaaS решает задачи переноса больших объемов медиа-данных из старых ленточных хранилищ в хранилища облачные, гарантируя отказоустойчивость и целостность данных. Типовой объем данных, подлежащих переносу, составляет от 2,5 до 22 петабайт, следовательно, миграции занимают месяцы или даже годы.

Следует отметить, что ПО MaaS является нишевым продуктом. Сама область т.н. промышленных систем хранения медиа-данных (англ. Media Asset Management system, MAM), на которую он ориентирован, довольно закрыта, и инновации в этой сфере с большой вероятностью представляют коммерческую тайну крупных игроков рынка. В этой связи проблема увеличения производительности подобных систем представляется крайне актуальной.

Особенностью ПО MaaS является то, что оно разворачивается на мощностях заказчика и работает в его сетевой инфраструктуре. Кроме того, исходным ленточным хранилищем продолжают пользоваться, поэтому ПО MaaS, при том, что оно должно работать максимально быстро, не должно при этом перегружать исходное хранилище своими запросами. Иными словами, для любой миграции помимо прочего существуют и явные ограничения сверху в виде расписания максимально допустимой нагрузки, которую может создать ПО MaaS.

Архитектура сервиса MaaS представлена на рис. 1. Как видно из рис. 1, ПО MaaS не имеет прямого контроля над ленточными считывателями. Общая статистика запросов к хранилищу недоступна так же, как и статистика ошибок чтения/записи конкретной ленты (т. е. если лента в силу возраста перестала считываться, ПО MaaS сможет узнать об этом только по косвенным признакам). При этом как состояние лент, так и нагруженность хранилища запросами других пользователей влияет на скорость выполняемой миграции данных.

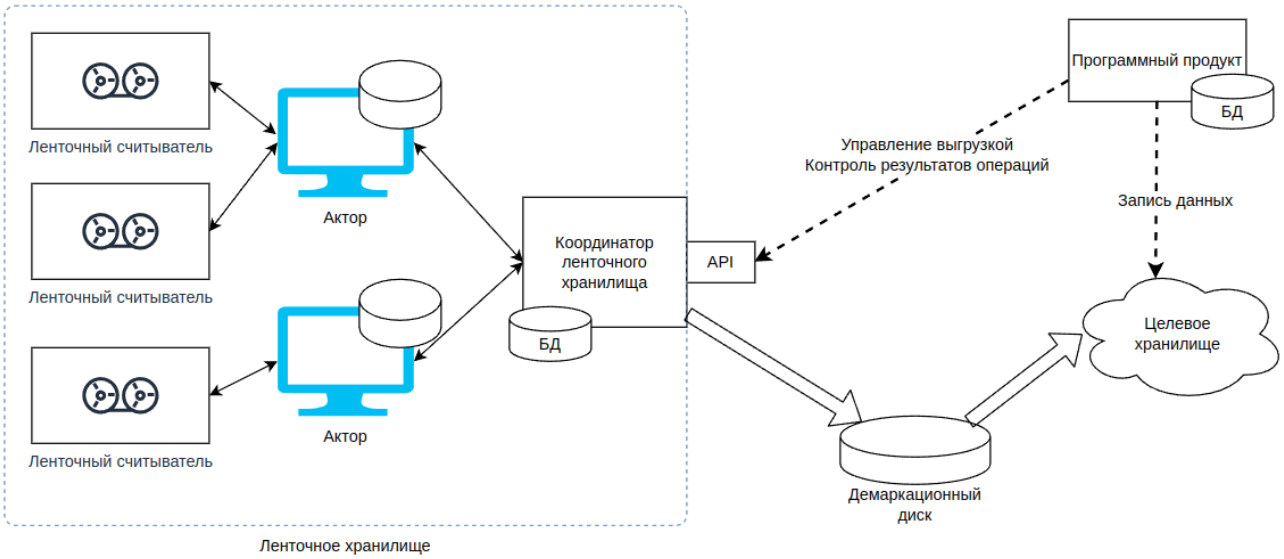


Рис. 1. Архитектура сервиса MaaS

1. Граф переходов состояний миграции

Миграция представляет собой многоступенчатый процесс. Каждый медиа-объект, подлежащий миграции, проходит через последовательность этапов обработки. Пример такой последовательности приведен на рис. 2.

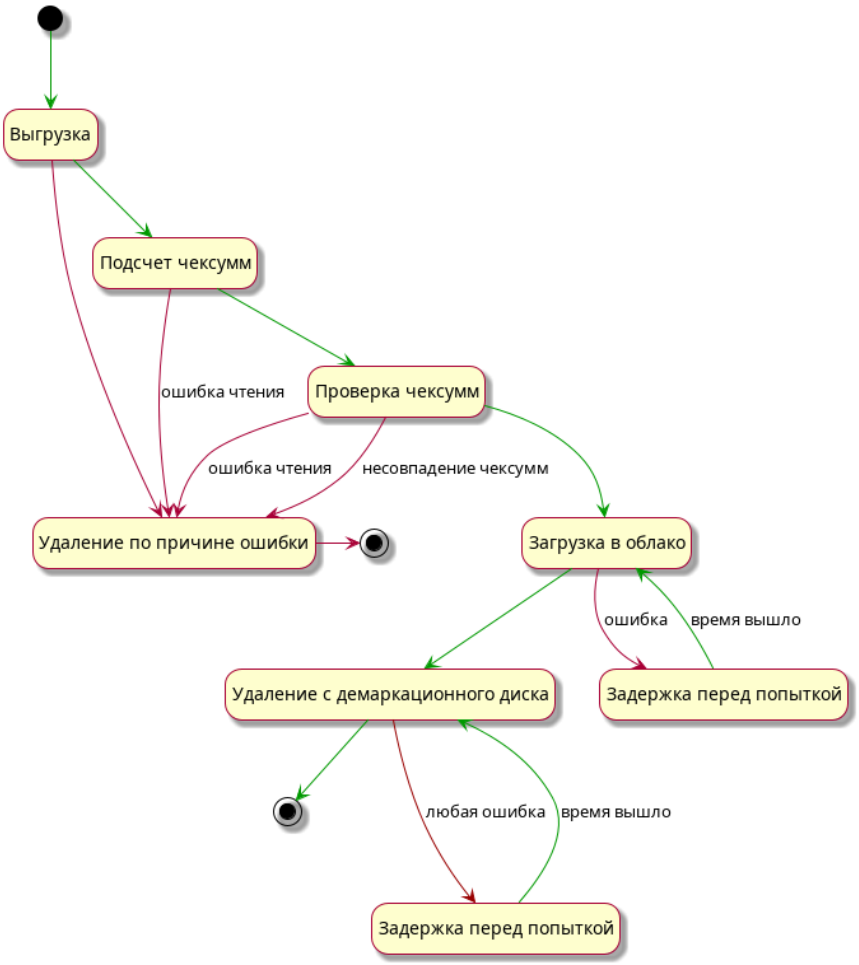


Рис. 2. Пример графа переходов состояний миграции медиа-объекта

Приведенный граф переходов описывает процедуру миграции каждого отдельного объекта из исходного хранилища. Данная процедура должна обеспечивать возможность следующих действий:

- возможность возобновления миграции после сбоя;
- возможность обработки нескольких объектов параллельно.

Заметим, что каждый из узлов представленного графа предполагает действие над объектом; это действие потребляет некоторый аппаратный ресурс (сетевой канал, вычислительный ресурс процессора, очередь ввода/вывода демаркационного диска). Таким образом, в многопоточной среде между разными потоками возникает конкуренция за общие ресурсы (resource contention), причем эта конкуренция зависит как от формы графа переходов, так и от нюансов аппаратной инфраструктуры конкретного заказчика.

Пусть (N, E) — ориентированный граф переходов состояний миграции, где $N = \{N_j\}$ — множество узлов графа, E — множество ребер (переходов) $(N_i, N_j) \in E,\ N_i \in N,\ N_j \in N$. Обозначим через N_0 начальное состояние миграции, N^ω — множество терминальных узлов.

Пусть $D = \{D_k\}$ — множество всех объектов, подлежащих миграции. Обозначим через D^{N_j} множество всех объектов, находящихся на узле N_j. Заметим, что $\bigcup_j D^{N_j} = D$.

Начальное состояние миграции выглядит следующим образом: $D^{N_0} = D,\ D^{N_j} = \varnothing,\ \forall j > 0$. Миграция полностью выполнена, когда верно $\bigcup_k D^{N_k} = D,\ \forall N_k \in N^\omega$.

Перемещением каждого медиа-объекта по графу состояний занимаются рабочие потоки приложения[1]. В каждый момент времени обязанностью рабочего потока является перевод объекта D_k по дуге $(N_i, N_j) \in E,\ N_i \notin N^\omega$, Очевидно, что для каждого нетерминального узла $N_i \notin N^\omega$ может существовать более одного результирующего узла $N_j,\ (N_i, N_j) \in E$. Неоднозначности в выборе результирующего узла для рабочего потока нет, поскольку поток лишь выполняет необходимое действие для текущего узла, а выбор дуги перехода зависит от результата этого действия. Реализация алгоритма такова, что существует только одна дуга, соответствующая успеху выполненной операции; все прочие переходы соответствуют логике обработки негативных сценариев.

2. Стратегии управления миграцией

Вся работа в рамках общей задачи миграции, возможная в данный момент времени, инкапсулирована в виде команд вида $C_i = (N_k, A, M_A)$, где A — медиа-объект, N_k — узел, на котором сейчас находится A, M_A — дополнительная информация об объекте (здесь могут быть знания об объекте, накопленные на предыдущих узлах графа миграции (N, E)). Получив команду, рабочий поток имеет всю необходимую информацию, чтобы продвинуть медиа-объект A по графу состояний на один переход вперед.

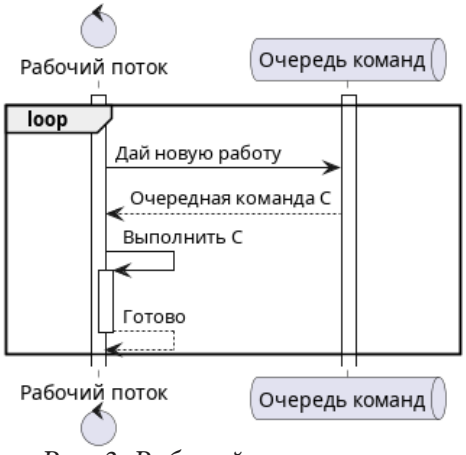


Рис. 3. Рабочий цикл потока

Рабочий цикл потока представлен на рис. 3.

Таким образом, время жизни каждой команды C_i можно представить следующим графиком (рис. 4).

Промежуток времени T_1 соответствует времени ожидания команды в очереди, T_2 — времени ее выполнения некоторым потоком. Суммарное время миграции T_Σ можно представить, как время от точки t_0 самой первой

[1]Здесь и далее под термином «поток» понимается один из подпроцессов (англ. *thread*) в выполнении программы.

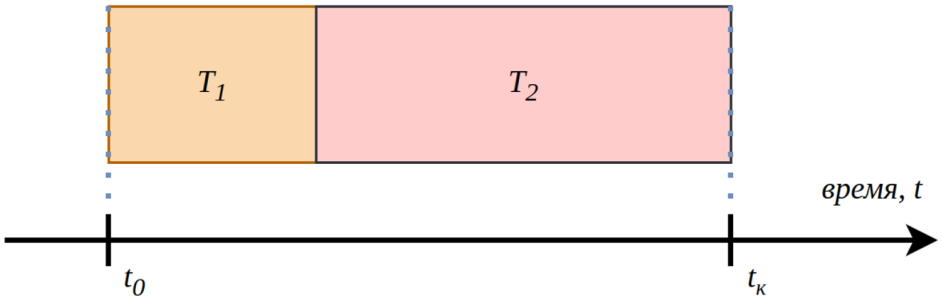


Рис. 4. Время жизни команды

команды, которая появилась в очереди, до точки $t_к^*$ самой последней команды, переводящей последний оставшийся медиа-объект в терминальное состояние. Опыт показывает, что наивное увеличение количества рабочих потоков далеко не всегда приводит к уменьшению T_Σ (а в ряде случаев к увеличению) как минимум, по следующим причинам:

• взаимная конкуренция между узлами графа (N, E) за аппаратные ресурсы, когда последовательное исполнение может оказаться быстрее, чем параллельное;

• ограниченное количество считывателей лент в исходной системе хранения.

Помимо этого, параллелизм, безусловно, ограничен, например, следующими факторами:

• ограничение на количество запросов к исходной системе хранения (как правило, ею продолжают пользоваться во время миграции, и ее доступность критична);

• объем демаркационного диска на порядки меньше, чем объем данных, которые необходимо смигрировать.

Таким образом, на скорость миграции влияют такие параметры системы, как

1. количество рабочих потоков,
2. максимальная степень параллелизма на узлах N_j графа W,
3. приоритет выполнения команд C_i.

В силу того, что процесс миграции характеризуется значительным количеством параметров (куда входят как параметры, описывающие граф переходов, так и параметры и ограничения на уровне многопоточной среды исполнения), вычленение «выигрышных» параметров в контексте данной задачи может оказаться серьезной проблемой как на этапе подготовки репрезентативного датасета, так и на этапе обучения модели. В связи с этим предлагается упрощение в виде готовых поведенческих стратегий $S = \{S_i\}$, реализованных в виде программного кода в ПО MaaS. Каждая стратегия может быть представлена как последовательность определенных команд C_i, определяющих конкретные действия с объектами миграции D_k в определенных узлах графа переходов состояний (N, E).

Изменение стратегии поведения в контексте данной задачи имеет смысл в дискретные моменты времени t_i, а промежуточную агрегированную статистику о состоянии процесса миграции стоит анализировать по временным окнам вида $[t_{i-1}, t_i]$. Предположим, что скорость миграции в момент времени t_{i+1} описывается функцией $F_{i+1} = F\left(x_1(t_i), x_2(t_i), \ldots, x_k(t_i), S, \Theta(t_i)\right)$, где $x_1(t_i), x_2(t_i), \ldots, x_k(t_i)$ — измерения, полученные на временном окне $[t_{i-1}, t_i]$, S — выбранная стратегия поведения ПО MaaS, $\Theta(t_i)$ — неизвестные внешние параметры среды (включающие, в частности, поведение других пользователей хранилища и состояние аппаратной и сетевой инфраструктуры). Тогда задачу выбора оптимальной стратегии S на следующем временном шаге t_{i+1} можно описать как задачу регрессии, построив приближенную оценку скорости миграции $F^*(x_1(t_i), x_2(t_i), \ldots, x_k(t_i), S)$.

3. Архитектура предлагаемого решения

Для решения задачи управления миграцией предлагается включить в ПО MaaS новый компонент, обозначенный на рис. 5 как Модуль анализа и принятия решений (АПР):

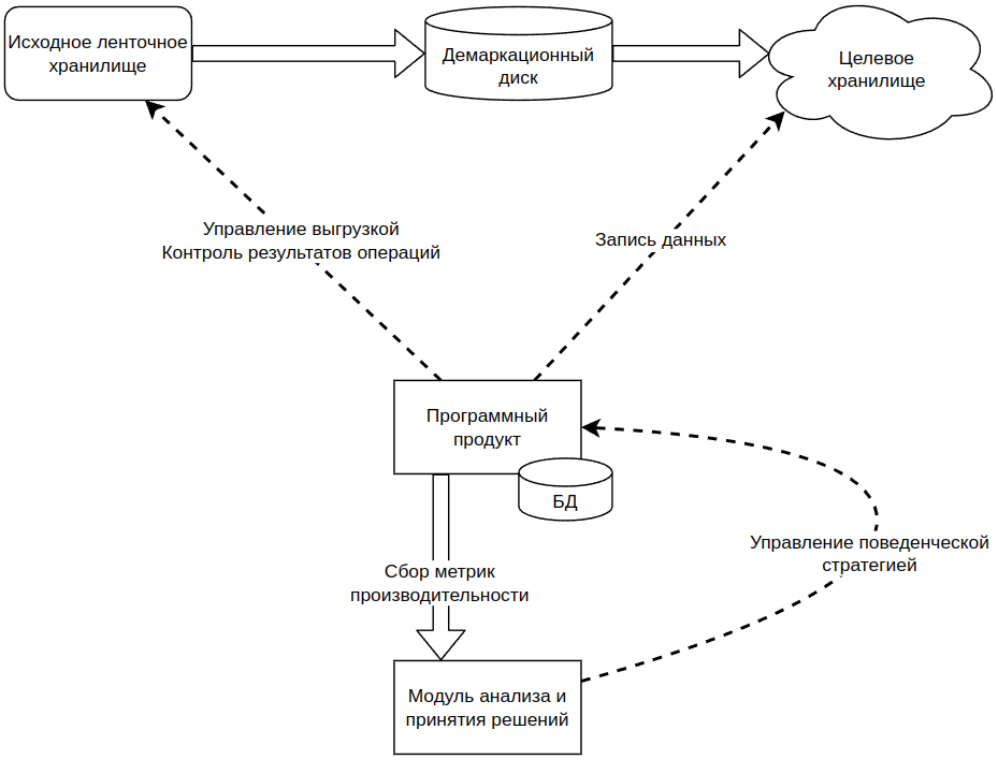


Рис. 5. Предлагаемая архитектура сервиса MaaS

Модуль АПР представляет собой программно-алгоритмический комплекс, функциями которого являются:

1. сбор и агрегирование метрик, поступающих от ПО MaaS;
2. принятие решение о смене стратегии поведения ПО MaaS;
3. отправка управляющих инструкций для ПО MaaS.

Сообщения, отправляемые ПО MaaS в Модуль АПР, по смыслу представляют собой трассировку некоторых событий. В рамках ПО MaaS реализована отправка низкоуровневых событий по шине сообщений ZeroMQ [3]. Примерами таких событий могут быть: факт добавления команды C_i в очередь исполнения, факт начала исполнения команды, факт завершения исполнения команды и т. п. Указанные события агрегируются Модулем АПР в более высокоуровневые метрики, такие как средний размер объекта, который совершил переход по дуге $(N_i, N_j) \in E$ или длительность нахождения объекта на узле N_i, а также более общие метрики, описывающие состояние миграции в целом. Агрегирование метрик предлагается выполнять с использованием колоночной базы данных ClickHouse [4].

Для решения поставленной задачи предлагаются следующие метрики:

Таблица 1

Метрики состояния миграции

Обозначение	Комментарий
x_1	Принятая стратегия поведения $S_i \in S$, где S — конечное счетное множество стратегий поведения, реализованных в Программном продукте.
x_2	Количество ошибок операций, соответствующих узлам $N_j \in N$
x_3	Медианная длительность миграции одного медиа-объекта (т. е. время прохождения маршрута по графу от начального узла до положительного терминального состояния).
x_4	Медианный размер медиа-объектов, завершивших миграцию до конца рассматриваемого временного окна.

Таблица 1 (продолжение)

x_5	Количество объектов, завершивших миграцию до конца рассматриваемого временного окна.
x_6	Суммарный размер объектов, завершивших миграцию до конца рассматриваемого временного окна.

Рассматривая ПО MaaS как совокупность параллельно выполняющихся потоков, требуется построить такое управление выбором стратегий, чтобы суммарная скорость миграции была максимальной. Такая задача, однако, имеет нечеткий характер в связи с тем, что:

• рассматриваемый программный продукт запускается на аппаратуре заказчика и в его сетевой инфраструктуре, что исключает получение информации о ее состоянии в каждый момент времени;

• у исходной системы хранения существует свой неизвестный нам график активности (включающий архивирование, использование в бизнес-процессах и т. п.).

В этой связи Модуль АПР должен стать управляющей системой, которая определяет выбор стратегии в соответствии с нечеткими зависимостями между состоянием системы, метриками состояния миграции и выполняемыми командами.

4. Адаптивная оптимизация миграции

Проблему увеличения производительности миграции логично рассматривать как задачу управления по прогнозирующей модели (MPC — Model Predictive Control) [5, 6]. Одним из наиболее эффективных способов решения данной задачи является использование моделей машинного обучения.

Исходные данные для обучения такой модели в нашем случае представляют собой временной ряд, состоящий из однотипных строк вида $\langle t, t_{DOW}, x_1, x_2, x_3, x_4, x_5, x_6 \rangle$, где t — время начала окна измерения (в пределах одного дня), t_{DOW} — номер дня недели, значения x_i, $i = \overline{1,6}$ соответствуют одноименным метрикам из табл. 1. Предполагается, что окно измерения константно и это значение является гиперпараметром. На указанных данных обучается предиктивная модель, описывающая, как прореагирует процесс миграции на изменение стратегии поведения.

Наиболее перспективными моделями, которые могут быть использованы в качестве предиктивной модели, представляются:

• рекуррентная нейронная сеть, в частности, LSTM, позволяющая учитывать последовательности состояний метрик и примененных при этом стратегий;

• модель обучения с подкреплением (Reinforcement Learning — RL).

Литература

1. *Yarens J. Cruz, Alberto Villalonga, Fernando Castaño, Marcelino Rivas, Rodolfo E. Haber* Automated machine learning methodology for optimizing production processes in small and medium-sized enterprises // Operations Research Perspectives. – 2024 ю – Vol. 12, 100308. – URL: https://doi.org/10.1016/j.orp.2024.100308 (дата обращения 24.10.24)

2. *Liang Fei, Zhang Taowen*. Application of Neural Network in Optimization of Chemical Process. – 2021. – 10.48550/arXiv.2110.04942. — URL: https://arxiv.org/abs/2110.04942 (дата обращения 24.10.24)

3. ZeroMQ : официальный сайт. — URL: https://zeromq.org/ (дата обращения 02.11.2024)

4. СУБД ClickHouse : официальный сайт. — URL: https://clickhouse.com/ (дата обращения 02.11.2024)

5. Оптимальное управление в математике // Большая российская энциклопедия. — URL: https://bigenc.ru/c/optimal-noe-upravlenie-v-matematike-5f9a4a (дата обращения 02.11.2024)
6. Basics of model predictive control // Do-MPC toolbox. — URL: https://www.do-mpc.com/en/latest/theory_mpc.html (дата обращения 18.11.2024)

УДК 004.4

ПРИМЕНЕНИЕ TENSORFLOW FRAMEWORK В СИСТЕМЕ ИНТЕЛЛЕКТУАЛЬНОГО ПОИСКА В СЕТИ ИНТЕРНЕТ С ИСПОЛЬЗОВАНИЕМ ИИ

А. И. Камышанов

Воронежский государственный университет

Аннотация. В статье рассматривается применение фреймворка TensorFlow для создания системы интеллектуального поиска в сети Интернет. Приведён анализ ключевых этапов разработки, включая обработку естественного языка, ранжирование результатов и персонализацию поиска. Особое внимание уделено архитектуре приложения, интеграции моделей машинного обучения и оптимизации процессов с использованием GPU. Представлены практические примеры реализации и схемы, демонстрирующие связь между компонентами системы. Сделан вывод о значимости TensorFlow для повышения точности и эффективности поиска.

Ключевые слова: TensorFlow, искусственный интеллект, интеллектуальный поиск, обработка естественного языка, ранжирование результатов, персонализация, архитектура приложения, машинное обучение, рекомендации, GPU-ускорение, поисковая система.

Введение

В эпоху информационного взрыва перед пользователями стоит задача быстрого и точного нахождения релевантных данных среди огромного объема информации. Традиционные поисковые системы не всегда способны удовлетворить эти требования, особенно при необходимости учитывать контекст запросов и индивидуальные предпочтения пользователей. Современные технологии искусственного интеллекта, такие как TensorFlow, предоставляют мощный инструментарий для создания интеллектуальных поисковых систем нового поколения. Эти системы способны обрабатывать запросы на естественном языке, ранжировать результаты на основе их релевантности и формировать персонализированные рекомендации. В статье подробно рассматриваются способы интеграции TensorFlow в архитектуру поискового приложения, особенности применения моделей машинного обучения для анализа данных, а также преимущества GPU-ускорения при обработке больших объемов информации. Обсуждаются ключевые подходы, которые позволяют значительно улучшить пользовательский опыт и производительность интеллектуальных систем поиска.

1. Архитектура приложения и интеграция TensorFlow

Приложение для интеллектуального поиска построено на многослойной архитектуре, обеспечивающей модульность и высокую производительность. Основу системы составляют четыре ключевых компонента. Веб-интерфейс служит точкой взаимодействия пользователя с системой, предоставляя удобный способ отправки запросов и получения результатов. Обработчики управляют запросами от веб-интерфейса, передавая их сервисам для обработки, что обеспечивает структурированное взаимодействие между слоями. Модели данных формируют надежное хранилище для всех данных, включая запросы пользователей, результаты поиска и метрики, необходимые для обучения моделей. Сервисы выполняют основную бизнес-логику, включая обработку данных и вызовы моделей TensorFlow, что позволяет реализовать интеллектуальный поиск. Архитектура приложения представлена на схеме (рис. 1).

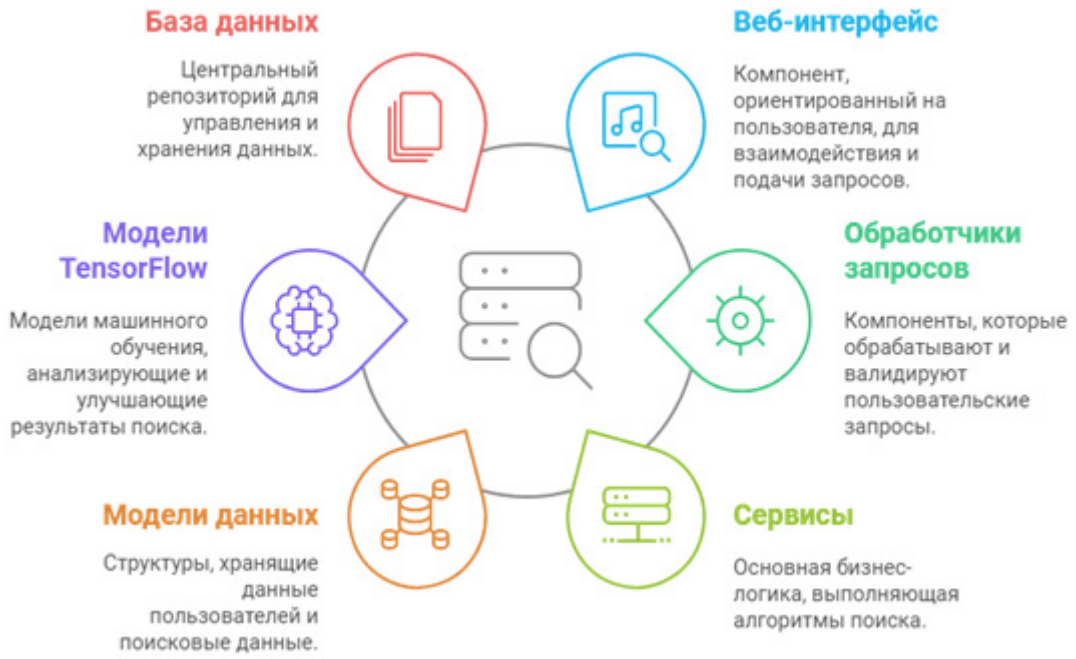


Рис. 1. Схема архитектуры приложения для интеллектуального поиска

TensorFlow играет ключевую роль в реализации интеллектуального поиска, обрабатывая сложные задачи машинного обучения на уровне сервисов. Он используется для обработки запросов пользователей с применением технологий обработки естественного языка (NLP), что позволяет системе лучше понимать намерения пользователя. Модели, разработанные с использованием TensorFlow, обеспечивают точное ранжирование результатов поиска, оценивая их релевантность запросу. Кроме того, система генерирует персонализированные рекомендации, анализируя поведение пользователей и их предпочтения. Интеграция TensorFlow в архитектуру позволяет добиться высокой эффективности и масштабируемости приложения, делая поиск более точным и удобным.

2. Использование TensorFlow для обработки естественного языка

Современные системы интеллектуального поиска требуют мощных инструментов для анализа и интерпретации текстовых данных. В рамках данного проекта для обработки запросов пользователей используются трансформерные модели, такие как BERT и GPT, которые поддерживаются TensorFlow. Эти модели позволяют анализировать семантическую структуру запросов, выявляя скрытые намерения пользователя. Это особенно важно для поиска, где корректная интерпретация запроса напрямую влияет на качество выдачи.

Благодаря поддержке TensorFlow данные модели способны автоматически исправлять ошибки в запросах, что улучшает пользовательский опыт. Также они предлагают варианты уточнений, если исходный запрос оказывается недостаточно точным. Это достигается за счет предварительного обучения моделей на больших текстовых коллекциях и последующего дообучения на данных, специфичных для задачи поиска.

Для реализации обработки запросов в реальном времени используется TensorFlow Serving, что позволяет минимизировать задержки и обеспечивать мгновенный отклик. При поступлении запроса он проходит предварительную обработку, после чего анализируется моделью трансформера. На основе предсказаний модели формируется результат, который возвращается пользователю. Схема обработки представлена на схеме (рис. 2).

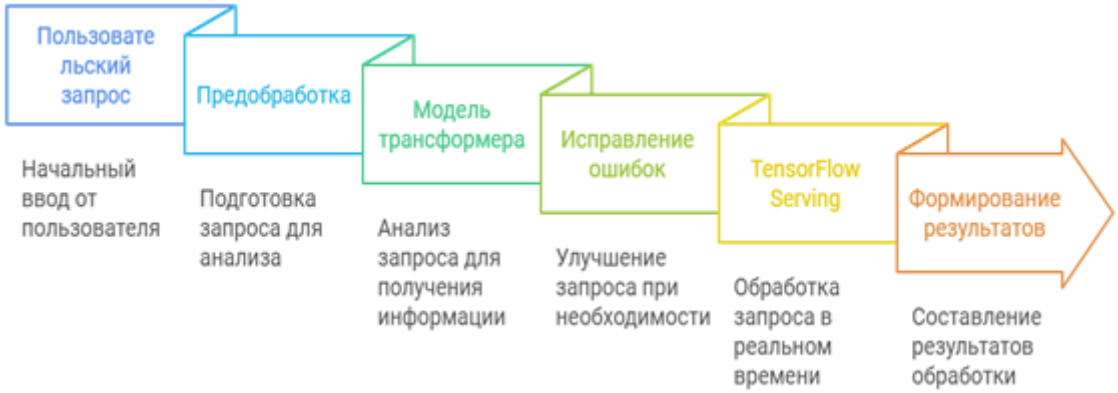


Рис. 2. Схема обработки запросов

3. Ранжирование результатов поиска

TensorFlow предоставляет мощные инструменты для создания моделей ранжирования, которые могут учитывать различные факторы, такие как частота ключевых слов, контекст запроса, а также поведение пользователей с похожими запросами. Эти алгоритмы помогают улучшить релевантность выдачи результатов поиска, ориентируясь на реальные предпочтения и действия пользователей. Для анализа работы таких моделей и мониторинга ключевых метрик, таких как Precision и Recall, можно использовать TensorBoard. Этот инструмент позволяет визуализировать данные и отслеживать качество алгоритмов ранжирования на разных этапах их обучения. В рамках конкретного примера для обучения модели ранжирования были использованы данные поисковых сессий. В них входила информация о текстах запросов, URL, кликах и времени взаимодействия с результатами. Итоговая модель способна прогнозировать вероятность удовлетворения пользователя предложенным результатом, что делает её применение эффективным для улучшения пользовательского опыта.

4. Ускорение обучения с использованием GPU

Разработка интеллектуальной поисковой системы на основе искусственного интеллекта требует обработки больших объемов данных, что делает вычислительные ресурсы ключевым фактором успеха. TensorFlow предоставляет встроенную поддержку графических процессоров (GPU), что позволяет значительно ускорить обучение моделей глубокого обучения. Благодаря технологии CUDA от NVIDIA, которая обеспечивает высокую параллельную производительность, обучение моделей, таких как рекомендательные системы или системы кластеризации данных, происходит в разы быстрее. Например, модель на базе BERT, адаптированная для анализа запросов пользователей, показала значительное снижение времени обучения: с 24 часов на CPU до 4 часов на GPU при обработке 100 000 запросов. Это ускорение позволяет быстрее развивать систему интеллектуального поиска, тестировать и внедрять новые алгоритмы, обеспечивая высокую производительность в условиях реального времени.

5. Расширение функционала с помощью TensorFlow

TensorFlow является незаменимым инструментом в создании системы интеллектуального поиска, так как предоставляет широкий набор инструментов для реализации ключевых функций. Одной из них является использование рекомендательных систем, которые анализируют историю запросов пользователей, выявляют их предпочтения и формируют персонализированные результаты. Это не только улучшает качество поиска, но и повышает уровень взаимо-

действия с пользователем, предлагая релевантные материалы, соответствующие их интересам. Другой важной функцией является кластеризация результатов поиска. TensorFlow позволяет применять алгоритмы кластеризации, чтобы группировать найденные результаты по категориям. Например, запрос на тему «искусственный интеллект» может быть разделён на такие категории, как «научные статьи», «программные библиотеки», «новости» и «курсы». Это делает интерфейс системы более удобным, а поиск — более структурированным и эффективным. В результате система интеллектуального поиска не только быстрее реагирует на запросы, но и предоставляет более понятные и организованные ответы, что особенно важно для работы в интернете с большими объемами данных.

Заключение

В ходе работы была разработана система интеллектуального поиска, использующая TensorFlow и искусственный интеллект. Приложение эффективно анализирует запросы пользователей, улучшает качество поиска благодаря рекомендательным системам и кластеризации. Использование GPU ускоряет обучение моделей, что повышает производительность и точность системы, обеспечивая быстрые и релевантные результаты для пользователей.

Литература

1. *Абрамов С.* Введение в TensorFlow: Практическое руководство / С. Абрамов. — СПб. : БХВ-Петербург, 2019. – 320 с.

2. *Хинтон Д.* Глубокое обучение / Д. Хинтон, Я. Бенгио, Ю. Лекун. – Москва : Вильямс, 2019. – 672 с.

3. *Гудвин П.* Python и машинное обучение / П. Гудвин, Д. Эммонс. — СПб. : Питер, 2018. – 448 с.

УДК 004.8:004.056

ПРИМЕНЕНИЕ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ В СИСТЕМАХ ЗАЩИТЫ ОТ СПАМА

А. А. Карташов

Воронежский государственный университет

Аннотация. В данной статье рассматриваются возможности применения алгоритмов машинного обучения в системах защиты от спама. Обсуждаются популярные алгоритмы машинного обучения, такие как наивный байесовский классификатор, метод опорных векторов, деревья решений, и другие. Приводится анализ эффективности различных моделей, оцениваются их преимущества и ограничения. Результаты работы могут быть использованы для разработки более точных и адаптивных систем фильтрации спама в электронной почте и мессенджерах.

Ключевые слова: машинное обучение, защита от спама, классификация сообщений, фильтрация, кибербезопасность, искусственный интеллект, обработка текстов, нейронные сети, наивный байесовский классификатор, метод опорных векторов, фишинг, автоматизация, анализ данных.

Введение

Современные средства коммуникации играют ключевую роль в обмене информацией. С увеличением объема передаваемой информации по всем этим каналам, проблема нежелательных сообщений, известных как спам, становится все более острой. Эффективное противодействие этому явлению требует инновационных подходов, а в современных технологиях защиты от спама ключевую роль играет искусственный интеллект и машинное обучение. Данная тема работы направлена на описание и анализ применения методов машинного обучения в технологии обнаружения и фильтрации спама.

1. Постановка задачи

На сегодняшний день существует обширный набор алгоритмов для идентификации спама. С учетом этого, разумным представляется их систематизация в соответствии с определенным критерием. Предлагается провести классификацию на основе применяемого метода или категории математического аппарата, лежащего в основе алгоритма фильтрации спама.

Для проведения анализа и обучения алгоритмов идентификации спама необходимо установить формальные параметры и определения:

Обозначим множество документов как $D = \{d_1, d_2, \ldots, d_i\}$. Зададим множество классов, представленных как $C = \{spam, not\ spam\}$. Определим множество терминов как $T = \{t_1, t_2, \ldots, t_i\}$ с количеством терминов $|T| = n$.

Обозначим обучающее множество документов как $D' \subseteq D$. Это множество состоит из документов, классы которых известны заранее.

Введем функцию классификации $class : D \to C$, которая сопоставляет каждый документ с определенным классом (spam или not spam).

Документ $d \in D$ может иметь различные формы представления. Введем понятие формы документа $R(d)$, которое зависит от конкретного алгоритма извлечения терминов из документа.

На основании этой терминологии рассмотрим существующие алгоритмы фильтрации спама, основанные на машинном обучении.

2. Вероятностные классификаторы

Введем понятие «мягкого» и «жесткого» вероятностного классификатора.

«Мягкий» вероятностный классификатор представляет собой функцию вероятности, которая оценивает вероятность того, что данный документ, представленный в форме $R(d) = x$, принадлежит к определенному классу C. Формула для «мягкого» классификатора выглядит следующим образом:

$$c_{soft}(d,c) = p\left(\text{class}(d) = cR(d) = x\right). \tag{1}$$

Этот подход позволяет проводить нечеткую классификацию, где документ может быть присвоен различным классам с разной степенью уверенности.

«Жёсткий» вероятностный классификатор также оценивает вероятность принадлежности документа к определенному классу, но с использованием порогового значения t. Если вероятность превышает пороговое значение t, то документ классифицируется как спам. Формула «жёсткого» классификатора выглядит следующим образом:

$$c_{soft}(d,c) = p\left(class(d) = cR(d) = x\right) > t. \tag{2}$$

Этот метод приводит к бинарной классификации, где документ либо относится к классу, либо нет, в зависимости от превышения порогового значения.

Приведенные выше понятия предоставляют основу для понимания алгоритмов классификации в контексте борьбы со спамом, таких как логистическая регрессия, наивный байесовский классификатор и алгоритмы, использующие марковские поля.

Алгоритм «наивной» байесовской классификации: пусть документ $d \in D$ представлен в форме вектора в пространстве терминов $x^{[d]} = \{x_1, x_2, \ldots, x_i\}$. «Наивный» байесовский классификатор определяет класс документа как наиболее вероятный из всех возможных классов, то есть с помощью оценки апостериорного максимума (MAP).

$$c_{MAP} = argmax_{c \in C} P\left(c|x^{[d]}\right). \tag{3}$$

Для идентификации спама, этот классификатор применяет следующий критерий:

$$c_{MAP} = \begin{cases} spam, & если \;\; c_{soft}(d, spam) > c_{soft}(d, not\, spam) \\ not\, spam, & иначе \end{cases} \tag{4}$$

Применяя теорему Байеса к формуле (1), получаем:

$$c_{soft}(d,c) = \frac{p(c) \times p\left(x^{[d]}|c\right)}{p\left(R(d) = x^{[d]}\right)}. \tag{5}$$

В связи с тем, что вероятности рассчитываются для одного и того же документа, вероятность $p(R(d) = x^{[d]})$ не влияет на результат сравнения при использовании критерия (4), и её можно опустить:

$$c_{soft}(d,c) = p(c) \times p\left(x^{[d]}|c\right). \tag{6}$$

Вероятность $p(x^{[d]} \mid c)$ может быть получена из обучающего множества данных, и для упрощения этого процесса используется «наивное» предположение о статистической независимости всех терминов в представлении документа. Это предположение приводит к тому, что:

$$p\left(x^{[d]}|c\right) = \prod_i p\left(x_i|c\right). \tag{7}$$

Преобразуем формулу (6) с учетом формулы (7):

$$c_{soft}(d,c) = p(c) \times \prod_i p\left(x_i|c\right). \tag{8}$$

Таким образом, «наивный» байесовский классификатор оценивает вероятности классов, основываясь на вероятностях появления каждого термина в документе. Его основными преимуществами являются малое время обучения и сравнительно высокая точность, однако он может быть уязвим к атакам с подбором «хороших» слов.

Алгоритм «не наивной» байесовской классификации: рассмотрим метод, который улучшает качество классификации в сравнении с классическим «наивным» байесовским классификатором. Основная идея заключается в аппроксимации статистической зависимости между терминами, что позволяет лучше учесть взаимосвязи между ними.

Рассмотрим следующую вероятность:

$$p\left(x^{[d]}|c\right) = p\left(x_1,\ x_2,\dots,x_i|c\right) = p\left(x_1|c\right)\times p\left(x_2,\ x_3,\dots,x_i|x_1,c\right) = \\ = p\left(x_1|c\right)\times\theta\left(x^{[d]},x_1,c\right)\times p\left(x_2,\ x_3,\dots,x_i|c\right). \tag{9}$$

При этом θ является некоторым неизвестным, но существующим распределением.

Для аппроксимации используется следующая формула вместо формулы (7):

$$p\left(x^{[d]}|c\right) = \prod_i p\left(x_i|c\right)\times\theta\left(x_i,c\right). \tag{10}$$

$\theta(x_i,c)$ изначально устанавливается в 1 для всех i (при этом формула (10) становится аналогичной формуле (6), то есть случаю классического «наивного» байесовского классификатора). При каждой ошибке в классификации происходит коррекция параметра θ с помощью так называемых коэффициентов уверенности α и β.

Такой метод проявляет высокую точность в классификации, обеспечивает быструю обработку и характеризуется относительно низким использованием памяти. Основной недостаток заключается в данный алгоритм не до конца изучен.

3. Линейные классификаторы

Линейный классификатор представляет собой вектор из n коэффициентов $\beta = (\beta_1,\beta_2,\dots,\beta_n)$, где $n = |T|$, и граничное значение t. Уравнение плоскости в пространстве терминов задается как

$$\beta\times x^{[m]} = t. \tag{11}$$

Эта гиперплоскость разделяет пространство терминов на два подпространства: одно содержит точки, классифицируемые как спам, а другое — как не спам.

Гиперплоскость считается разделяющей, если для каждого $d \in D$ выполняются условия:

$$\left(\beta\times x^{[m]} > t\right) \Leftrightarrow class\left(m\right) = spam \tag{12}$$

$$\left(\beta\times x^{[m]} \le t\right) \Leftrightarrow class\left(m\right) = not\ spam \tag{13}$$

Множество документов считается линейно разделимым, если для него существует такая разделяющая гиперплоскость.

Основной задачей линейных классификаторов является поиск оптимальной разделяющей гиперплоскости для заданного множества документов.

Персептрон: алгоритм, который итеративно находит разделяющую плоскость для заданного множества документов. Если множество документов линейно разделимо, то персептрон ищет любую разделяющую плоскость. В случае отсутствия такой плоскости алгоритм может зациклиться, если не установлено максимальное количество итераций. На каждой итерации происходит коррекция коэффициентов, соответствующих терминам точек, которые были некорректно классифицированы. Эта коррекция выполняется путем прибавления или вычитания некоторой константы.

Преимущества персептрона включают простоту, скорость и адаптивность. Однако полученная гиперплоскость не обязательно является наилучшей для данного набора документов.

Алгоритм Winnow: представляет собой вариацию персептрона с некоторыми ключевыми отличиями. Основные особенности алгоритма включают:

1. Элементы всегда положительны: в отличие от персептрона, где веса могут принимать как положительные, так и отрицательные значения, в алгоритме Winnow элементы (веса) всегда остаются положительными.

2. Мультипликативная коррекция коэффициентов: вместо аддитивной коррекции, применяемой в персептроне, алгоритм Winnow использует мультипликативную коррекцию. При ошибке первого рода (ложное срабатывание), коэффициенты β, соответствующие терминам некорректно классифицированного документа, умножаются на некоторый коэффициент $\alpha > 1$. При ошибке второго рода, коэффициенты умножаются на $0 < \gamma < 1$.

Модификация алгоритма Winnow с использованием ортогональных разреженных биграмм (которые формируют базис в пространстве терминов и позволяют наличие других слов между словами в биграмме) обеспечивает высокую точность, достигающую порядка 99 %.

Алгоритм опорных векторов (SVM): метод машинного обучения, который вычисляет разделяющую гиперплоскость, наиболее удаленную от ближайших точек обучающего множества документов. Гиперплоскость полностью определяется небольшим количеством точек, известных как опорные векторы. Линейная комбинация этих опорных векторов формирует классификатор.

Уже существуют эффективные реализации этого алгоритма для идентификации спама, так как он имеет высокую точность и подходит к нашей задаче.

4. Классификаторы на основе сходства

Рассмотрим документ $d \in D$ и множество корректно классифицированных документов $D' \subseteq D$. Пусть задана некоторая функция расстояния $distance : D \times D \to \mathbb{R}$, являющаяся метрикой на пространстве терминов T, то есть удовлетворяющая следующим требованиям:

1. $\forall x, y \in D, distance(x, y) = 0 \Leftrightarrow x = y$;

2. $\forall x, y \in D, distance(x, y) = distance(y, x)$;

3. $\forall x, y, z \in D, distance(x, z) \le distance(x, y) + distance(y, z)$.

Основная идея данного алгоритма заключается в том, что документы одного класса находятся близко друг к другу в векторном пространстве, в то время как документы разных классов расположены далеко друг от друга. Это предположение формирует основу для различных классификаторов, использующих сходство документов в пространстве терминов для определения их принадлежности к классам.

Алгоритм k-ближайших соседей: простым и интуитивным методом классификации документов является присвоение документу класса, совпадающего с классом ближайшего известного документа. Однако более эффективным подходом к этой задаче является использование алгоритма k-ближайших соседей

Этот метод учитывает классы k наиболее близких к рассматриваемому документам и принимает решение в пользу класса, которому принадлежит большинство из этих документов. Ключевой особенностью данного метода является учет не только ближайших соседей, но и их коллективного влияния на принятие решения о классификации.

Важно отметить, что несмотря на свою простоту и широкое использование в различных областях, данные методы не всегда обеспечивают высокую точность при решении задачи идентификации спама.

5. Логические классификаторы

Логические классификаторы представляют собой группу алгоритмов, которые используют аппарат логики для описания связей между терминами в документах. В данной категории выделяются два основных подхода:

Деревья принятия решений: деревья принятия решений в процессе обучения последовательно разбивают тренировочный набор данных по одному атрибуту за раз, используя определенный критерий (например, наибольшую информационную выгоду). В контексте классификации спама, обычные деревья принятия решений могут демонстрировать недостаточную эффективность. Однако, с применением модификаций, можно достичь существенного улучшения качества алгоритма.

Логический вывод на основе набора правил: алгоритмы, использующие логический вывод из заранее определенного набора правил, обладают сравнимой эффективностью по сравнению с алгоритмами на деревьях принятия решений. Тем не менее, существуют и комбинированные подходы, обеспечивающие более высокую точность классификации.

Заключение

Методы машинного обучения, такие как Наивный Байесовский Классификатор и Алгоритм опорных векторов (SVM), предоставляют эффективные инструменты для выделения паттернов и классификации текстовых данных. Классификаторы на основе сходства, такие как k-ближайших соседей, рассматривают структуру пространства терминов, учитывая близость документов.

Логические классификаторы, такие как деревья принятия решений, используют логические правила для описания связей между терминами. Вместе с тем, некоторые комбинированные подходы, такие как логический вывод на основе набора правил, могут обеспечить более высокую точность в сравнении с чистыми деревьями принятия решений.

Выбор подходящего метода зависит от конкретных требований и характеристик задачи. Эффективное решение проблемы спама требует тщательного анализа и выбора подходящего алгоритма в зависимости от особенностей текстовых данных и поставленных целей.

Литература

1. *Серано Л.* Грокаем машинное обучение / Л. Серано ; пер. с англ. А. Рыбиной. – Санкт-Петербург : Питер, 2021. – 448 с. (Серия «Грокаем») (дата обращения: 08.11.2024).

2. Применение технологий искусственного интеллекта в информационной безопасности. – URL: https://www.anti-malware.ru/analytics/Technology_Analysis/using-artificial-intelligence-technologies-in-information-security (дата обращения: 12.11.2024).

3. Наивный Байес, или о том, как математика позволяет фильтровать спам. – URL: https://habr.com/ru/articles/415963/ (дата обращения: 15.11.2024).

УДК 004.896

ВЛИЯНИЕ АРХИТЕКТУРЫ ГЛУБОКОЙ НЕЙРОННОЙ СЕТИ НА ФУНКЦИОНИРОВАНИЕ АЛГОРИТМА ГРАДИЕНТА СТРАТЕГИИ ИСПОЛНИТЕЛЬ-КРИТИК A2C

В. В. Кашко, С. А. Олейникова

Воронежский государственный технический университет

Аннотация. Объектом исследования является алгоритм Advantage Actor Critic (A2C), позволяющий обеспечить процесс обучения с подкреплением на основе взаимодействия двух аппроксиматоров: исполнителя и критика. Предметом исследования является архитектура связи нейронных сетей, которые в процессе функционирования соответственно прогнозируют стратегии и функции ценности. Для этого был проведен вычислительный эксперимент, позволяющий оценить эффективность взаимодействия исполнителя и критика и конечного результата на примере задачи перевернутого маятника путем применения различных комбинаций архитектуры.

Ключевые слова: глубокое обучение с подкреплением, метод градиента стратегии, метод исполнитель критик, A2C, агент, окружающая среда, бутстреппинг, ценность состояния, суммарный доход, подкрепление, нейронные сети, скрытый слой, архитектура, устойчивость алгоритма, сходимость.

Введение

Глубокое обучение с подкреплением открывает новые горизонты в разработке адаптивных автономных самообучающихся систем. Применение аппарата нейронных сетей позволяет решить проблему «проклятия размерности» за счёт обобщения, но при этом вносит дополнительные расходы на настройку гиперпараметров и сложность анализа функционирования агента [1–5, 7]. В отличие от стандартных приложений распознавания образов или задач классификации, которые требуют заранее подготовленных пакетов данных, глубокое обучение с подкреплением, в основном, обучается в online-режиме, непрерывно взаимодействуя со средой [1, 6–9]. Исходя из специфики соответствующих алгоритмов, следует прямая зависимость качества функционирования агента от способности аппроксиматора корректно и точно осуществлять предсказание.

В настоящее время всё больший интерес представляют гибридные методы, объединяющие в себе, как стратегические [5] — аппроксимирующие политику, так и ценностные — базирующиеся на вычислении ценности состояния (состояния-действия) [1–5]. Ярким представителем таковых является метод исполнитель-критик — Advantage Actor Critic или A2C. Особенностью стратегических алгоритмов, например REINFORCE, является отсутствие смещения значений и высокая дисперсия случайной величины, описывающей суммарный доход, которая возрастает с увеличением длины эпизода. Для устранения существующей проблемы предпринимались различные попытки. Одна из них заключалась в применении базы — базового уровня оценки дохода, что ознаменовало появление метода REINFORCE с базой [1–5]. Предложенный подход позволил уменьшить дисперсию с сохранением отсутствия смещения, но очень медленно сходится, поскольку требует больше эпизодов взаимодействия со средой. В попытке устранить данный эффект, было предложено использовать бутстреппинг — возможность оценки ценности текущего состояния на основе оценок последующих [1–5]. В результате возник метод исполнитель-критик, где в качестве исполнителя выступает стратегия, а критика — функция ценности, на основании которой производится корректировка поведения. Важное преимущество данного алгоритма заключается в его применении к неэпизодическим задачам, поскольку

он способен обучаться по неполным траекториям [3]. Подробнее с данным методом можно ознакомиться в [1–5]. В случае применения глубоких нейронных сетей для реализации алгоритма возникает вопрос организации взаимодействия между исполнителем и критиком для обеспечения корректного функционирования и сходимости. В процессе работы метода, для генерации действий и ценности состояния используются одни и те же входные данные, но корректировка весовых коэффициентов осуществляется на основании полученного дохода и предсказанной оценки. Целью данной работы является анализ воздействия архитектуры взаимодействия между глубокими сетями исполнителя и критика на функционирование алгоритма A2C.

1. Архитектура сети исполнитель-критик

Аппарат нейронных сетей в контексте алгоритма аппроксимирует две параметрические функции. Первая представляет собой стратегию, вторая функцию ценности состояния. Настройка параметров оценки оказывает прямое воздействие на эффективность прогнозирования дохода, так же как параметризация стратегии на оптимальность выбираемых действий в соответствующих состояниях. В процессе проектирования возникает вопрос о необходимости объединения сети исполнителя и критика. В связи с этим, возникает проблема организации взаимодействия аппроксиматоров для достижения наилучших показателей работы алгоритма. В случае раздельного использования сетей возрастает количество настраиваемых параметров. При этом может возникнуть ситуация, в результате которой сеть исполнителя будет осуществлять некорректное прогнозирование действий до тех пор, пока критик в достаточной степени не выполнит обучение [2]. В результате, потребуется больше данных, что приведёт к необходимости увеличения числа эпизодов. Качественная аппроксимация заключается в таком представлении пространства состояний, при котором подобные объединены в единый кластер [2, 7]. Связывание исполнителя и критика, частичное либо полное, имеет множество преимуществ, но содержит и недостатки. Поскольку стратегия и ценность предназначены для решения общей задачи, то будет целесообразно объединить обе сети в одну. Данный подход позволит использовать единый кластер признаков данных, что теоретически должно способствовать ускорению обучения. Так как при такой конфигурации сети связаны между собой, используется меньше параметров, по сравнению с разделённым вариантом, в результате чего обновление весов исполнителя происходит совместно с критиком без задержки, что очевидно должно ускорить процесс обучения алгоритма и снизить требования к количеству эпизодов. При всех преимуществах, объединение сетей предполагает наличие недостатка, связанного со снижением устойчивости алгоритма по причине обратного распространения градиентов двух компонент — исполнителя и критика [2]. На рис. 1 представлены варианты возможной архитектуры, при которых сети могут быть полностью разделены, так и частично (либо полностью) состоять из общих скрытых слоёв.

2. План проведения экспериментов

Поскольку основной задачей исследования являлось определение воздействия архитектуры нейронной сети на качество работы алгоритма исполнитель-критик A2C, были рассмотрены двухслойная и трёхслойная конфигурации аппроксиматоров. Данный выбор обусловлен наличием небольшого количества скрытых слоёв, что сопряжено с относительно малым числом вычислений, и низкой вариативностью способов конфигурации сети. В качестве решаемой агентом задачи была выбрана проблема балансировки перевёрнутого маятника.

Исходя из решаемой проблематики, в качестве среды взаимодействия использовалась OpenAI gym CartPole-v0, которая завершает эпизод в случае падения маятника (либо превы-

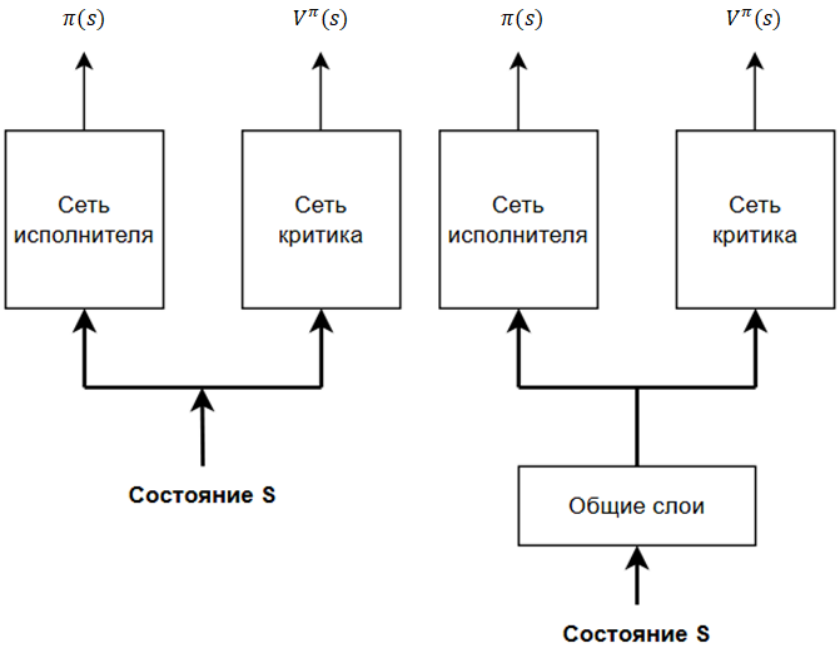


Рис. 1. Варианты архитектуры сети исполнитель-критик

шения допустимого порога отклонения), а при достижении 200 шагов, целевое состояние считается достигнутым. За каждый шаг агент вознаграждается наградой, равной 1.

Перед началом проведения экспериментов были подобраны соответствующие значения гиперпараметров алгоритма из диапазона рекомендуемых, которые не изменялись на протяжении всех опытов. В качестве коэффициента обесценивания дохода использовалось значение, равное 0.9 для каждого рассмотренного случая. В серии экспериментов с двухслойной сетью коэффициент скорости обучения — 0.003, для случая трёхслойной — 0.007. Размерности слоёв, используемых в стендах, соответственно равны 128, 64 и 32, с функцией активации relu. В качестве выхода для сети стратегии применялась операция softmax. Критик использовал необработанное активацией выходное значение.

На первом этапе рассматривался вариант двухслойных сетей. В начале, эксперимент заключался в использовании исполнителя и критика по отдельности. Далее сети были объединены первым скрытым слоем. В конце первого этапа изучалась конфигурация, объединяющая все скрытые слои. На втором этапе исследовалась трёхслойная сеть с отдельными сетями исполнителя и критика, с одним, двумя и полностью обобщёнными слоями. Результатом каждого эксперимента являлся график распределения суммарного дохода за 1000 эпизодов.

Стоит отдельно рассмотреть метрики, относительно которых рассматривалось влияние архитектуры глубокой сети на функционирование алгоритма A2C. В первую очередь главным показателем послужило непосредственное распределение суммарного вознаграждения, исходя из графика которого, можно оценить сходимость и устойчивость метода. Был использован порог суммарного дохода, который позволил определить количество удачных запусков алгоритма при соответствующей конфигурации сети, на основании которого рассчитывался показатель процента благоприятных исходов. Поскольку максимально возможное значение дохода за эпизод у используемой среды равно 200, то в качестве порогового выбрано 100. Оценки были получены по результатам 10 запусков алгоритма с соответствующей конфигурацией архитектуры нейронной сети.

3. Практические результаты

3.1. Двухслойная сеть

Вариант с разделённой конфигурацией исполнителя и критика для случая двухслойной сети продемонстрировал плохие показатели сходимости. Лишь в 30 % случаев алгоритм смог превысить выбранный порог суммарного дохода за эпизод. В очень редких случаях были получены хорошие результаты, но это связано больше с удачной генерацией начальных значений весовых коэффициентов, чем с влиянием архитектуры сети. Наличие данных примеров демонстрирует возможность обучения при разделённой конфигурации, способствующего сходимости данного алгоритма, но вероятность таких случаев достаточно низкая. После объединения первых слоёв сетей исполнителя и критика, стали заметны изменения в величине единичных всплесков дохода. Если для разделённой архитектуры количество опытов, достигших за 1000 эпизодов максимального значения, составляло всего 30% от общего числа, то для случая объединения первого слоя, оно возросло до 50 %. Количество опытов, обеспечивших полную сходимость, практически осталось без изменений. Полное слияние сетей исполнителя и критика значительно повысило качество работы алгоритма. Величина экспериментов, хотя бы раз превышающих за 1000 эпизодов пороговое значение, достигло 100 %, а случаев относительно равномерной сходимости до 70 %. Величины соотношений числа стабильных запусков и превысивших порог экспериментов для соответствующих конфигураций двухслойной нейронной сети представлены в табл. 1.

Таблица 1

Тип конфигурации	Соотношение количества опытов, превысивших порог хотя бы раз на протяжении 1000 эпизодов к общему числу экспериментов	Соотношение стабильно сходящихся опытов к общему числу экспериментов
Разделённые	~ 0.3	~ 0.2
Общий первый слой	~ 0.5	~ 0.2
Общие слои	~ 1	~ 0.7

3.2. Трёхслойная сеть

На втором этапе экспериментов был добавлен дополнительный скрытый слой, состоящий из 32 нейронов. При запуске разделённой конфигурации для трёхслойной сети, количество запусков, превысивших порог дохода, составило 80 % и продемонстрировало 40 % стабильно работающих примеров. Вариант с объединением первого скрытого слоя увеличил число достигших максимума экспериментов до 100 % при неизменном количестве стабильных случаев, которое так же, как и в предыдущем цикле опытов, составляет 40 %. Все последующие конфигурации в каждом из экспериментов демонстрировали стабильное превышение порога дохода. При объединении двух первых слоёв стало заметно увеличение количества сходящихся примеров, которое составило уже 50 %. Полное слияние сетей исполнителя и критика, так же, как и в случае двухслойной сети, продемонстрировало заметное возрастание числа стабильных результатов. Величины соотношений стабильных запусков и превысивших порог экспериментов для соответствующих конфигураций трёхслойной нейронной сети представлены в табл. 2. Стоит отметить, что на данном этапе, помимо архитектуры сети, изменилось количество скрытых слоёв. Как показывают результаты раздельного использования, расширение аппроксиматора вглубь, на данной задаче, привело к возрастанию процента пересёкших пороговое значение результатов и увеличило количество относительно равномерно сходящихся к максимуму, по сравнению с двухслойной сетью.

Таблица 2

Тип конфигурации	Соотношение количества опытов, превысивших порог хотя бы раз на протяжении 1000 эпизодов к общему числу экспериментов	Соотношение стабильно сходящихся опытов к общему числу экспериментов
Разделённые	~ 0.8	~ 0.4
Общий первый слой	~ 1	~ 0.4
Общие первые два слоя	~ 1	~ 0.5
Общие слои	~ 1	~ 0.8

Заключение

Исходя из полученных экспериментальных данных, следует, что для обеспечения увеличения вероятности стабильной работы глубокого алгоритма исполнитель-критик целесообразно использовать архитектуру нейронных сетей с общими глубокими слоями. Результаты подтвердили, описанный в [2] тезис, касательно проблем устойчивости алгоритма, связанных с использованием полностью связанных сетей.

Сеть критика оценивает текущее состояние, из которого принимается решение о предпринимаемом действии, поэтому для того, чтобы скорректировать выбор, необходимо корректно обновить веса данной сети. Как ранее было сказано, в случае раздельной архитектуры, до тех пор, пока критик не обучится в достаточной степени, исполнитель не способен корректно осуществлять выбор действия. Этим обусловлены плохие показатели данной конфигурации. Использование общих данных не гарантирует формирование одинакового пространства признаков для каждого аппроксиматора, что приводит к потере синхронизации между сетями исполнителя и критика.

Наличие связи гарантирует использование общей совокупности классов признаков. При этом в случае полного слияния сетей, распространение ошибки (сумма ошибок исполнителя и критика), в процессе обучения сразу, без задержки, оказывает влияние на весовые коэффициенты. Это в значительной степени ускоряет процесс обучения, уменьшая требования к количеству необходимых данных, но оказывает негативное воздействие на устойчивость, за счёт перекрёстного воздействия чужой ошибки на каждую из сетей. Данная проблема не являлась целью текущего исследования, поскольку рассматривалось непосредственное влияние архитектуры, а не подбор параметров для улучшения сходимости. При этом она заслуживает дальнейшего рассмотрения, поскольку уменьшение дисперсии и повышение качества алгоритма является основой проблематикой обучения с подкреплением с точки зрения приложений. Если нельзя полностью исключить влияние, то стоит рассмотреть возможность его уменьшения.

В процессе выполнения данной работы были определены эффекты от влияния архитектуры взаимодействия сетей исполнителя и критика на функционирование глубокого алгоритма обучения с подкреплением A2C. Экспериментальные данные сравнивались на основании метрик распределения суммарного дохода за эпизод, соотношения превысивших хотя бы раз за 1000 эпизодов горизонт случаев и соотношения примеров, осуществляющих равномерную сходимость к общему числу итераций производимого опыта. Сформулировано объяснение определённых в результате исследования эффектов.

Литература

1. *Саттон Р. С.* Обучение с подкреплением: Введение. 2-е изд. : Пер. с англ. / Р. Саттон, Э. Барто. – Москва : ДМК Пресс, 2020. – 552 с.

2. *Грессер Л.* Глубокое обучение с подкреплением: теория и практика на языке Python / Л. Грессер, Ван Лун Кенг. – Санкт-Петербург : Питер, 2022. – 416 с.

3. *Лонца А.* Алгоритмы обучения с подкреплением на Python: пер. с англ. А. А. Слинкина / А. Донца. – Москва : ДМК Пресс, 2020. – 286 с.

4. *Лю Ю. (Х.)* Обучение с подкреплением на PyTorch: сборник рецептов: пер. с англ. А. А. Слинкина / Ю. (Х.) Лю. – Москва : ДМК Пресс, 2020. – 282 с.

5. *Моралес М.* Грокаем глубокое обучение с подкреплением : учебное пособие / М. Моралес. – Санкт-Петербург : Питер, 2023. - 464 с.

6. *Траск Э.* Грокаем глубокое обучение : учебное пособие / Э. Траск. – Санкт-Петербург : Питер, 2021. – 352 с.

7. *Хайкин С.* Нейронные сети: полный курс, 2-е издание: Пер. с англ. / С. Хайкин. – Москва : Издательский дом «Вильямс», 2006. – 1104 с.

8. *Гафаров Ф. М.* Искусственные нейронные сети и приложения: учеб. пособие / Ф. М. Гафаров, А. Ф. Галимянов. – Казань : Изд-во Казан. ун-та, 2018. – 121 с.

9. *Сергеев А. П.* Введение в нейросетевое моделирование : учеб. пособие / А. П. Сергеев, Д. А. Тарасов ; под общ. ред. А. П. Сергеева. – Екатеринбург : Изд-во Урал. ун-та, 2017. – 128 с.

УДК 004.056

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ В ЗАДАЧЕ ОПРЕДЕЛЕНИЯ ФЕЙКОВЫХ НОВОСТЕЙ

Е. С. Кенина

Воронежский государственный университет

Аннотация. В статье анализируется эффективность применения нейросетевых моделей для решения задачи классификации фейковых новостей. Рассматриваются основные этапы работы с нейронной сетью и некоторые виды нейросетевых архитектур. Проводится сравнительный анализ полученных результатов, включающих время обучения, время предсказания и точность классификации.

Ключевые слова: классификация фейковых новостей, препроцессинг, машинное обучение, сверточная нейронная сеть (CNN), рекуррентная нейронная сеть (RNN), сеть долгой краткосрочной памяти (LSTM), рекуррентная нейронная сеть с управляемыми блоками (GRU), двунаправленная LSTM (BiLSTM), модель BERT.

Введение

В современном мире дезинформация становится довольно значимой проблемой, затрагивающей различные аспекты общества. Развитие платформ социальных сетей способствует беспрецедентному ускорению распространения дезинформации. Это приводит ко многим негативным последствиям: подрыву доверия к средствам информации, поляризации общества, препятствию адекватного реагирования на новости.

Значительным подвидом дезинформации являются фейковые новости, представляющие собой ложную или искаженную информацию, созданную с целью ввести в заблуждение или манипулировать аудиторией. Фейковые новости представляют серьёзную угрозу для общественного мнения, восприятия реальности и принятия важных решений. Чтобы минимизировать риски человеческой ошибки, для решения задачи обнаружения фейковых новостей всё чаще используются автоматизированные классификаторы, основанные на технологиях машинного обучения. Такие классификаторы способны автоматически анализировать информацию и выявлять признаки, свидетельствующие о том, что новость является фейковой, что позволяет более эффективно противостоять распространению дезинформации.

В настоящее время разработаны сложные модели для классификации фейковых новостей, позволяющие обрабатывать большие объемы данных, улавливать нюансы языка, анализировать содержание статьей и распознавать более тонкие признаки и закономерности.

В данной работе анализируются результаты применения некоторых нейросетевых архитектур для решения задачи классификации фейковых новостей. Получены показатели времени обучения, времени предсказания и точности результатов при использовании различных моделей обучения, а также сделаны выводы на основе полученных результатов с целью сравнения данных моделей.

1. Анализ задачи

Классификация фейковых новостей является сложным процессом, включающим в себя анализ источника информации, проверку фактов, оценку стиля речи и множество других факторов, что подчеркивает необходимость автоматизации данного процесса. Кроме того, существует несколько категорий, к которым можно было бы отнести ложные сведения: неверные факты,

неверная интерпретация достоверной информации, псевдонаучные тексты, компиляции, состоящие в основном из чужих цитат и другие. В [1] приводится следующая классификация:

1) сатира, относящаяся к имитационным новостным программам, в которых обычно используется юмор или преувеличение, чтобы знакомить аудиторию с обновлениями новостей; пародия, которая вместо того, чтобы прямо комментировать текущие события с помощью юмора, играет на нелепости вопросов и подчеркивает их, придумывая полностью вымышленные новые истории;

2) фальсификация, которая относится к статьям, не имеющим фактической основы, но опубликованным в стиле новостных статей для создания легитимности (производитель товара часто имеет намерение ввести в заблуждение);

3) манипуляция, чаще с реальными изображениями или видео, чтобы создать ложное повествование;

4) реклама, распространяемая под видом настоящих новостных репортажей, а также выпуски со ссылкой на прессу, публикуемые как новости;

5) пропаганда, относящаяся к новостям, которые создаются политическим субъектом для влияния на общественное восприятие (открытая цель — принести пользу общественному деятелю, организации или правительству).

Однако, несмотря на то, что фейковые новости могут быть представлены в различных формах, все они направлены на манипулирование эмоциями, вследствие чего имеют общие черты и закономерности, которые могут быть выявлены с помощью технологий машинного обучения и обработки естественного языка (NLP). Это позволяет автоматически выявлять дезинформацию на основе ключевых слов, тональности текста, структуры предложений и других признаков.

2. Этапы решения задачи

Решение задачи классификации фейковых новостей можно разделить на несколько этапов.

На первом шаге необходимо получить датасет, представляющий собой всеобъемлющую подборку новостных статей, тщательно подобранных для предоставления сбалансированного и непредвзятого набора данных. Это имеет решающее значение для получения высококачественных обучающих данных и точных результатов. Для изучения фейковых новостей доступно несколько открытых наборов данных, эти наборы данных имеют существенные ограничения по размеру, категории или предвзятости. Чтобы устранить эти ограничения, можно объединить несколько существующих наборов данных, имеющих схожую структуру.

В [2] отмечено, что:

- новостные статьи, содержащие от 450 до 550 слов, как правило, более надежны;
- общие тенденции указывают на то, что более короткие, но содержательные новости часто более правдивы;
- читаемость текста фейковых новостей хуже, чем реальных новостей;
- субъективность статей о фейковых новостях более значима, чем статей о реальных новостях;
- количество статей, представляющих реальные новости, больше, чем статей, представляющих фейковые новости.

Эти наблюдения дают ценную информацию о характеристиках фейковых и реальных новостных статей, что может сыграть важную роль в дальнейшей разработке и совершенствовании алгоритмов обнаружения фейковых новостей.

На следующем этапе происходит препроцессинг, или предварительная обработка текста новости. Здесь выполняются операции преобразования текстового заголовка новости в формат, необходимый для использования классификатора. К функциям препроцессора можно отнести:

• удаление пробелов, знаков препинания и спецсимволов с помощью регулярных выражений;

• приведение слов к нижнему регистру;

• токенизацию с целью разбиения текста на отдельные части — токены;

• удаление нерелевантных слов (стоп-слов), не играющих значительной роли;

• лемматизацию с целью приведения всех данных к нормальной словарной форме (для существительных и прилагательных это именительный падеж в единственном числе, для глаголов, причастий и деепричастий — инфинитив). Это уменьшает количество уникальных значений, что способствует улучшению результатов работы.

Поскольку алгоритмы машинного обучения не могут работать с сырым текстом, на следующем этапе полученные в результате препроцессинга данные нужно конвертировать в последовательности числовых идентификаторов каждого токена. Это называется извлечением признаков. Одним из наиболее простых и популярных способов извлечения признаков при работе с текстом является «мешок слов». Он описывает вхождения каждого слова в текст.

Чтобы использовать модель, необходимо определить словарь известных токенов и выбрать степень присутствия известных слов. При этом любая информация о порядке или структуре слов игнорируется. Иными словами, для модели важно знать только то, встречается ли знакомое слово в документе, но неважно, где конкретно. Сложность этой модели состоит в том, как определить словарь и как подсчитать вхождение слов.

Можно использовать другой способ создания словаря с помощью сгруппированных слов. Это изменит размер словаря и даст мешку слов больше деталей о документе. Такой подход называется «N-грамма». N-грамма — это последовательность каких-либо сущностей (слов, букв, чисел, цифр и т. д.). В контексте языковых корпусов, под N-граммой обычно понимают последовательность слов. Цифра N обозначает, сколько сгруппированных слов входит в N-грамму. В модель попадают не все возможные N-граммы, а только те, что фигурируют в корпусе.

На заключительном этапе для обнаружения характерных признаков каждой из категорий новостей и для формирования классификационной оценки отнесения новости к одной из категорий с определенным уровнем достоверности результата необходимо обучить нейронную сеть.

В основе классификатора новостных заголовков могут лежать следующие популярные архитектуры нейронных сетей, которые хорошо показали себя в задачах NLP и рекомендуются к использованию [3].

Сверточная нейронная сеть (Convolutional neural network, CNN) содержит один или более объединенных или соединенных сверточных слоев. CNN использует вариацию многослойного перцептрона. Сверточные слои используют операцию свертки для входных данных и передают результат в следующий слой. Эта операция позволяет сети быть глубже с меньшим количеством параметров. Они могут быть эффективными при обработке коротких текстов, таких как анализ тональности. В статье [4] автор описывает процесс и результаты задач классификации текста с помощью CNN. В работе представлена модель на основе word2vec, которая проводит эксперименты и демонстрирует хорошие результаты.

Рекуррентная нейронная сеть (RNN) — это тип нейросетей, который широко используется в NLP. Они способны учитывать последовательность слов в тексте и учиться на основе контекста. Однако у RNN есть ограничения в обработке длинных последовательностей из-за проблемы исчезающего градиента.

Сеть долгой краткосрочной памяти (Long Short-Term Memory, LSTM) и *рекуррентная нейронная сеть с управляемыми блоками* (Gated Recurrent Unit, GRU) — модификации RNN, спроектированные для решения проблемы исчезающего градиента и широко используются в NLP для анализа длинных последовательностей. LSTM создана для более точного моделирования временных последовательностей и их долгосрочных зависимостей, чем традиционная рекуррентная сеть. В работе [5] представлена модель для автоматической морфологической

разметки, которая показывает точность 97.4 %. GRU является более легкой и вычислительно эффективной по сравнению с LSTM. Ее главное преимущество перед LSTM заключается в более низкой сложности и меньшем количестве параметров, что может быть важно при работе с ограниченными вычислительными ресурсами. Однако, стоит отметить, что LSTM всё равно остается более мощным в решении некоторых сложных задач, требующих учета долгосрочных зависимостей.

Двунаправленная LSTM, или *BiLSTM* — это модель последовательности, которая содержит два уровня LSTM, один из которых предназначен для обработки входных данных в прямом направлении, а другой — для обработки в обратном направлении. Он обычно используется в задачах, связанных с NLP. Двунаправленный LSTM помогает машине лучше понимать эту взаимосвязь, чем однонаправленный LSTM. Это обеспечивает дополнительный контекст для сети и приводит к более быстрому и даже полному изучению проблемы, что важно для точной классификации фейковых новостей [6].

Для улучшения точности и сокращения времени обучения каждая из нейросетевых моделей может использовать заранее предобученную на русском корпусе слов модель BERT [7].

BERT — двунаправленная модель с transformer-архитектурой, заменившая собой последовательные по природе рекуррентные нейронные сети (LSTM и GRU), с более быстрым подходом на основе механизма внимания. Модель также предобучена на двух задачах без учителя — моделирование языковых масок и предсказание следующего предложения. Это позволяет использовать предобученную модель BERT, настраивая её под конкретные задачи, такие как определение эмоциональной окраски текста, вопросно-ответные системы и многие другие. В работе [8] приводятся достаточно высокие результаты трех обученных гибридных нейросетевых моделеи — BERT-CNN, BERT-GRU, BERT-LSTM для идентификации фейковых новостей.

Полученные векторные представления передаются в нейросетевые слои в зависимости от конкретной модели для обнаружения характерных признаков. Таким образом, модели используют семантический анализ для выявления фейковой новости. Далее для обработки полученных признаков может быть использован полносвязный нейросетевой слой для улучшения точности распознавания. Кроме того, поскольку нейросетевые модели должны относить новости к одной из многих категорий, ключевой задачей является выполнение многоклассовой классификации. Для этого в конце каждой модели можно использовать полносвязный нейросетевой слой, содержащий столько нейронов, сколько существует классов, и каждый нейрон генерирует оценку вероятности для определенного класса в диапазоне от 0 до 1.

3. Сравнительный анализ полученных результатов

Для проведения сравнительного анализа моделей классификации фейковых новостей, использующих различные методы распознавания признаков и закономерностей, были использованы следующие архитектуры нейронных сетей:

сверточная нейронная;

сеть долгой краткосрочной памяти;

рекуррентная нейронная сеть с управляемыми блоками;

двунаправленная LSTM.

Также была рассмотрена реализация модели BERT, которая может повысить эффективности представленных моделей.

Для проведения анализа необходим большой объем достаточно разнообразных данных. Был выбран датасет, состоящий из двух частей: обучающего и тестового наборов. Обучающий набор содержал около 18000 новостных статей, классифицированных как фейковые или правдивые, а тестовый набор содержал около 5000 статей, которые были использованы для проверки правильности предсказаний модели после обучения.

Следующим этапом работы являлось преобразование текстового заголовка каждой новости в оптимальный для использования классификатора формат, включающее в себя удаление пробелов, знаков препинания и спецсимволов, а также приведение слов к нижнему регистру. После этого была проведена токенизация, представляющая собой процесс разбиения текста на отдельные части — токены, что необходимо для анализа каждого слова по отдельности, а также для последующего извлечения признаков. Затем были удалены нерелевантные слова (стоп-слова), поскольку они часто встречаются в предложениях, но при этом не имеют смысловой нагрузки. Заключительным шагом препроцессинга стала лемматизация, представляющая собой приведение всех данных к нормальной словарной форме с целью уменьшения количества уникальных значений и, следовательно, повышения эффективности работы.

После препроцессинга было проведено извлечение признаков, представляющее собой конвертацию полученных данных в последовательности числовых идентификаторов каждого токена. На заключительном этапе была обучена нейронная сеть, задача которой заключалась в выявлении характерных особенностей каждой категории новостей и предоставлении предсказания, представляющего собой оценку того, какой из категорий принадлежит новость. Время обучения при использовании различных нейросетевых архитектур, лежащих в основе классификатора новостных заголовков, время предсказания и точность полученных результатов, представляющая собой долю верных результатов (истинно положительных и истинно отрицательных) относительно общего объема данных, представлены в табл. 1.

Таблица 1

Модель	Время обучения (мин)	Время предсказания (сек)	Точность (%)
CNN	40	1	83
LSTM	110	1.8	85
GRU	100	1.7	84
BiLSTM	120	2.5	88

Из представленных в таблице данных следует, что среди всех моделей самую высокую точность показывает двунаправленная LSTM. Это объясняется тем, что данная модель содержит два уровня LSTM, один из которых осуществляет обработку данных в прямом направлении, а другой — в обратном направлении, что приводит к более полному изучению задачи. Рекуррентная нейронная сеть с управляемыми блоками и сеть долгой краткосрочной памяти показывают хорошее сочетание скорости и точности, являясь более оптимальными по времени в сравнении с BiLSTM. При этом GRU является более быстрой, чем LSTM, поскольку использует меньшее количество параметров. Сверточная нейронная сеть продемонстрировала минимальное время обучения и предсказания, однако точность полученных результатов оказалась ниже, чем у рекуррентных моделей, так как CNN лучше обрабатывает локальные зависимости, но ограничена при работе с длинными последовательностями.

Использование модели BERT, предобученной для задач моделирования языковых масок и предсказания следующего предложения, существенно улучшило производительность всех упомянутых выше моделей, сократив время обучения и повысив точность предсказаний. Результаты использования BERT представлены в табл. 2.

Таблица 2

Модель	Время обучения (мин)	Время предсказания (сек)	Точность (%)
BERT-CNN	20	0.7	89
BERT-LSTM	50	0.9	90
BERT-GRU	40	0.8	89
BERT-BiLSTM	55	1.1	92

На основании полученных результатов можно сделать следующие выводы. Использование BERT значительно сократило время обучения каждой из моделей. BERT-CNN достигла высокой точности (89 %) при минимальном времени обучения и предсказания, а модели BERT-GRU и BERT-LSTM продемонстрировали хороший баланс между временем и точностью с небольшим преимуществом в скорости предсказаний у BERT-GRU. Максимальная точность (92%) была достигнута при использовании модели BERT-BiLSTM. Таким образом, интеграция BERT улучшает производительность любой используемой модели, при этом BERT-BiLSTM лучше всего подходит для задач, где важна максимальная точность, а BERT-CNN обеспечивает быстрое выполнение работы при довольно высокой, но не максимальной точности.

Заключение

В результате проведенной работы был проведен обзор решений в области идентификации фейковых новостей, рассмотрены модели глубокого обучения, используемые для задачи классификации, сделаны обоснованные выводы об эффективности рассмотренных моделей. При этом отмечается, что выбор модели зависит от поставленной задачи, поскольку каждая из них демонстрирует различное сочетание времени работы и точности результатов.

Литература

1. Fake News / C. Edson, Jr. Tandoc, Z. Lim, R. Ling. // Digital Journalism. – 2018. – Vol. 6, № 2. – P. 137–153.

2. *Padalko H.* A novel approach to fake news classification using LSTM-based deep learning models / H. Padalko, V. Chomko, D. Chumachenko // Frontiers in Big Data. – 2023. – Vol. 6. – URL: https://doi.org/10.3389/fdata.2023.1320800 (дата обращения: 15.07.2024).

3. 7 архитектур нейронных сетей для решения задач NLP. – URL: https://neurohive.io/ru/osnovy-data-science/7-arhitektur-nejronnyh-setej-nlp/ (дата обращения: 15.07.2024).

4. *Yoon K.* Convolutional Neural Networks for Sentence Classification / K. Yoon // The 2014 Conference on Empirical Methods In Natural Language Processing. – 2014. – P. 1746–1751.

5. *Wang P.* Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Recurrent Neural Network / P. Wang, Y. Qian, F. K. Soong, L. He, H. Zhao // ResearchGate. 2018. – URL: https://www.researchgate.net/publication/283118024_Part-of-Speech_Tagging_with_Bidirectional_Long_Short-Term_Memory_Recurrent_Neural_Network (дата обращения: 15.07.2024).

6. *Zeng Z.-Y.* A Review Structure Based Ensemble Model for Deceptive Review Spam / Z.-Y. Zeng, J.-J. Lin, M.-S. Chen, M.-H. Chen, Y.-Q. Lan, J.-L. Liu. // Information. – 2019. – Vol. 10, № 7. – URL: https://doi.org/10.3390/info10070243 (дата обращения: 15.07.2024).

7. BERT в двух словах: Инновационная языковая модель для NLP. – URL: https://habr.com/ru/companies/otus/articles/702838/ (дата обращения: 15.07.2024).

8. *Тумбинская М. В.* Идентификация фейк-новостей с помощью веб-ресурса на основе нейронных сетей / М. В. Тумбинская, Р. А. Галиев // Программные продукты и системы. – 2023. – № 4. – С. 590–599.

УДК 004.032.26

ИСПОЛЬЗОВАНИЕ СВЁРТОЧНОЙ НЕЙРОННОЙ СЕТИ ДЛЯ СЕГМЕНТАЦИИ СПУТНИКОВЫХ СНИМКОВ ЛЕСНЫХ МАССИВОВ

К. С. Клименко

Воронежский государственный университет инженерных технологий

Аннотация. В данной статье рассмотрена программная сегментация оптических спутниковых снимков лесных насаждений, которая имеет перспективы использования для проектировании сети лесовозных автомобильных дорог при освоении лесных хозяйств. В работе рассмотрены популярные методы и алгоритмы сегментации изображений на основе нейронных сетей, проведён анализ популярных программных библиотек для реализации нейронных сетей на языке программирования Python и описан процесс реализации сверточной нейронной сети с архитектурой U-Net.
Ключевые слова: сегментация изображений, обработка изображений, нейронные сети, Python, U-Net, Keras, спутниковые снимки, распознавание деревьев, лесные массивы, машинное обучение.

Введение

Одним из важных приоритетов лесопромышленной отрасли является развитие транспортной логистики, в которую входит проектирование и строительство лесовозных автомобильных дорог [1–3]. Таким образом возникает необходимость в анализе лесных массивов, использование которого актуально во многих сферах, включая сельское и лесное хозяйство.

Сегментации лесных насаждений на оптических спутниковых снимках высокого разрешения, при проектировании схем транспортного освоения лесных массивов, является крайне важным инструментом для анализа данных местности. Зачастую для сегментации оптических спутниковых снимков привлекают специалистов, работа которых трудозатратна. Для оптимизации процесса проектирования уместно применить программные решения, одним из которых являются технологии машинного зрения.

Целью данной работы является применение машинного распознавания для сегментации лесных насаждений на спутниковых снимках высокого разрешения. В работе рассмотрены наиболее популярные методы для сегментации изображений, программные библиотеки для разработки моделей машинного обучения, представлен алгоритм сегментации спутниковых снимков.

1. Методы и алгоритмы сегментации изображений

Для решения задачи сегментации изображений существует довольно большое множество методов. Рассмотрим самые популярные и эффективные методы для решения данной задачи. Метод случайного леса [4], является универсальным методом машинного обучения. Он использует комплексный набор деревьев на основе выборок данных с несколькими переменными для принятия решений о классификации пикселей на изображениях. Данный метод может использоваться в абсолютном большинстве задач машинного обучения, но требует слишком большого объём данных для увеличения точности результатов.

Ещё одним популярным методом, является метод основанный на графовых моделях, таких как графовые сверточные сети и марковские случайные поля [5]. Он учитывает многие факторы, включая связи между пикселями, цвета, текстуры и формы и позволяет получать высо-

кокачественную сегментацию. Однако, для обучения нейронной сети с применением данного метода требуется очень большой объём данных. Учитывая специфику задачи, данный метод не гарантирует получение результата с необходимой точностью.

Для точной сегментации изображений одними из популярных методов являются алгоритмы на основе свёрточной нейронной сети [5]. Данный метод применяет свёрточный слой для обнаружения особенностей изображения и нейронную сеть для принятия решений о классификации пикселей. В своем обучении нейронная сеть использует алгоритм обратного распространения. Происходит прямое распространение от первого слоя к последнему, затем вычисляется ошибка в выходном слое и осуществляется обратное распространение.

Метод на основе свёрточной нейронной сети, является наиболее подходящим для задач бинарной сегментации изображений, как требует относительно небольшой объём данных, а точность результатов получается довольно высокой.

2. Программная библиотека для реализации нейронной сети

Программный код нейронной сети удобнее всего реализуется на языке программирования Python, так как он имеет удобный в работе синтаксис и большие математические программные библиотеки. Для работы с нейронными сетями для данного языка программирования самыми популярными являются следующие библиотеки: PyTorch, TensorFlow и Keras.

Библиотека PyTorch имеет интуитивный синтаксис, схожий с синтаксисом языка программирования Python. В ней есть хорошая интеграция с другими инструментами и библиотеками для машинного обучения. Но её использование может быть осложнено, так-как при работе с большими моделями и данными она имеет низкую производительность в сравнении с другими библиотеками.

Библиотека TensorFlow имеет высокую производительность и широкий набор инструментов для разработки, отладки и развёртывания моделей. Она имеет поддержку большинства платформ, включая web и мобильные платформы, а модульная архитектура библиотеки позволяет удобно расширять функционал проектов. Однако, сложный синтаксис и отсутствие гибкости при работе с моделями, осложняют работу с данной библиотекой.

В данной работе, для реализации нейронной сети была выбрана библиотека Keras, так-как она имеет реализации широко применяемых блоков нейронных сетей, а также ряд заранее спроектированных и обученных моделей [4]. В неё ходят инструменты для упрощения работы с текстом и изображениями. Она проста в использовании, основана на модульной архитектуре и имеет высокую скорость развёртывания. Из недостатков данной библиотеки можно выделить проблемы на низкоуровневом API, которые возникают в специфических случаях и низкая производительность при работе с GPU в сравнении с другими библиотеками. Данные недостатки не влияют на качество и результаты при решении поставленной задачи.

3. Реализация алгоритма

Для программной сегментации оптических спутниковых снимков лесных массивов использован метод на основе свёрточной нейронной сети и архитектура U-Net. Изначально данная архитектура была разработана для использования в медицинских целях, но благодаря её результативности в решении задач бинарной сегментации изображений, со временем её начали использовать и в других сферах.

Программный код, реализующий свёрточную нейронную сеть, включающий в себя функционал для загрузки обучающих данных и модель нейронной сети, был реализован на языке программирования Python с применением модели U-Net и библиотеки Keras.

В качестве исходных данных был составлен массив данных для обучения нейронной сети, состоящий из 140 оптических спутниковых снимков высокого разрешения и соответствующих им изображений сегментированных вручную карт.

На основе массива данных нейронная сеть обучалась 28 эпох. Были проведены тесты на 20 спутниковых снимках, которых не было в составе данных для обучения. Точность результатов составила 73 %. Для улучшения точности было решено расширить массив данных для обучения нейронной сети до 250 спутниковых снимков и сегментированных вручную изображений, на основе которых удалось улучшить структура нейронной сети.

Нейронная сеть снова прошла обучение на расширенных данных, при этом точность сегментации выросла до 98 %. Один из снимков, использованных для тестирования представлен на рис. 1. Результат работы обученной нейронной сети представлен на рис. 2.

Рис. 1. Пример спутникового снимка для тестирования нейронной сети

Рис. 2. Результат сегментации спутникового снимка нейронной сетью

Заключение

Программная сегментация изображений применяется многих сферах для решения сложных задач, с которыми плохо справляются другие алгоритмы. Например, для построения сегментированных карт местности, для обнаружения объектов на фотографиях и т. д. Сегментация изображений сложный и трудозатратный процесс, требующий большой объём исходных данных для обучения. Однако, использование нейронных сетей, помогает упростить и уско-

рить этот процесс, что открывает большие перспективы для создания программного обеспечения, способного выполнять сложные, трудозатратные и специфические задачи.

Созданная и обученная в данной работе нейронная сеть может успешно использоваться для проектирования лесовозных автомобильных дорог. Также программную сегментацию изображений можно применить для своевременного обнаружения лесных пожаров, распознавания объектов на фотографиях, прогнозирования погоды по спутниковым снимкам и во многих других сферах.

Литература

1. Проектирование схем транспортного освоения лесных массивов с применением информационно-интеллектуальных систем / В. В. Никитин, А. Н. Брюховецкий, А. В. Скрыпников, И. А. Высоцкая, Р. С. Сапелкин, А. Б. Бондарев // Автоматизация. Современные технологии. – 2022. – Т. 76, № 3. – С. 130–134.
2. Экспериментальное исследование методов автоматизированного проектирования трассы лесовозной автомобильной дороги / Е. В. Чирков, А. В. Скрыпников, А. О. Боровлев, В. С. Прокопец, И. А. Высоцкая // Автоматизация. Современные технологии. – 2021. – Т. 75, № 1. – С. 29–33.
3. Информационно-интеллектуальная система проектирования лесотранспортных сетей / В. В. Никитин, И. А. Высоцкая, А. В. Скрыпников, А. Н. Брюховецкий, Р. С. Сапелкин, А. Б. Бондарев // Автоматизация. Современные технологии. – 2022. – Т. 76, № 4. – С. 185–188.
4. *Коул А.* Исскуственный интеллект и компьютерное зрение. Реальные проекты на Python, Keras и TensorFlow / А. Коул, С. Ганджу, М. Казам; РГБ. – Издательский дом «Питер», 2023. – 608 с.
5. *Постолит А. В.* Основы искусственного интеллекта в примерах на Python; БХВ – Петербург. 2022. – 448 с.
6. Обзор алгоритмов сегментации // Хабр: [сайт]. – 2024. – URL: https://habr.com/ru/companies/intel/articles/266347/ (дата обращения: 07.10.2024).
7. Библиотеки для глубокого обучения: Keras // Хабр: [сайт]. – 2024. – URL: https://habr.com/ru/companies/ods/articles/325432/ (дата обращения: 06.10.2024).

УДК 004.94

КЛАССИФИКАЦИЯ СЕРДЕЧНЫХ АРИТМИЙ С ИСПОЛЬЗОВАНИЕМ СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ

Т. А. Кувшинов, О. Ю. Марьясин

Ярославский государственный технический университет

Аннотация. В работе рассмотрена задача классификации различных сердечных аритмий на основе данных фотоплетизмографии. Описано современное состояние проблемы, методы, применяемые для классификации сердечных аритмий на основе машинного обучения и искусственных нейронных сетей, общедоступные наборы данных, которые могут быть использованы для решения подобных задач. Отмечено два основных подхода к решению задачи классификации сердечных аритмий на основе данных фотоплетизмографии. Из которых авторами был выбран подход на основе анализа сигнала фотоплетизмографии как временного ряда. Особенностью рассмотренной в работе задачи классификации, является то, что были добавлены отдельные классы для пульсовых волн разных типов для каждого вида сердечной аритмии. Это позволяет установить связь с такими болезнями как сахарный диабет, атеросклероз или гипертония. В качестве применяемой нейросетевой архитектуры авторы выбрали одномерную сверточную нейронную сеть. Значения точности полученное при решении задачи классификации с двенадцатью классами составило 99,3 %.

Ключевые слова: фотоплетизмография, сердечная аритмия, классификация, искусственная нейронная сеть, сверточная нейронная сеть.

Введение

Фотоплетизмография (ФПГ) (англ. Photoplethysmography — PPG) представляет собой оптический метод регистрации изменения объема крови в микрососудистом русле ткани. Объем крови в исследуемом участке ткани меняется по мере прохождения пульсовой волны, вызванной выбросом крови в результате сокращения сердца. Исследуемый участок ткани (например, палец руки) просвечивается светом с определенной длиной волны (такой, чтобы испускаемый свет поглощался эритроцитами в крови), и проходящий через ткани или отраженный (в зависимости от вида фотоплетизмографа) свет регистрируется фотоприемником. Объем крови в исследуемом участке ткани обратно пропорционален интенсивности регистрируемого света. Получаемая в результате фотоплетизмограмма отражает изменение объема крови в исследуемом участке ткани с течением времени. Чаще всего применяется пальцевая ФПГ, но могут использоваться и другие участки тела, например, запястье, мочка уха, лоб, области вокруг бицепса или мышц голени [1].

Пульсовые волны на ФПГ соответствуют динамике кровенаполнения за каждый сердечный цикл и синхронизированы с сокращениями сердца. По ФПГ можно измерять частоту сердечных сокращений (ЧСС), насыщение крови кислородом, кровяное давление, жесткость (ригидность) артериальной стенки, скорость распространения пульсовой волны и другие показатели. Сигнал ФПГ содержит широкий спектр информации о состоянии сердечно-сосудистой системы человека и, в том числе, о наличии сердечно-сосудистых заболеваний. Кроме того, на ФПГ присутствуют более медленные волны (волны второго порядка, дыхательные волны), синхронизированные с ритмом дыхания человека, по которым можно определять частоту дыхания.

Обычно в клинических условиях для анализа сердечной активности используют электрокардиографию (ЭКГ) — методику регистрации электрических полей, образующихся при работе сердца. Хотя ФПГ и ЭКГ являются разными технологиями, по своей сути обе технологии

предназначены для измерения активности сердца и диагностики состояния сердечно-сосудистой системы, поэтому во многих случаях ФПГ представляет собой практическую альтернативу ЭКГ.

ФПГ может уступать ЭКГ по точности и часто содержит артефакты в виде помех или шумов, но является более доступным и удобным для пациента методом оценки состояния сердечно-сосудистой системы. ФПГ может быть получена не только с помощью стационарных устройств, но и с помощью компактных портативных пульсоксиметров, предназначенных для персональной диагностики и широко представленных в продаже по доступной цене. В настоящее время большой популярностью пользуются фитнес-браслеты и смарт-часы, осуществляющие мониторинг различных показателей активности и здоровья организма, и в большинстве моделей для измерения ЧСС и других показателей используется ФПГ. Существуют приложения для смартфона, позволяющие снимать ФПГ и отображать различные рассчитываемые по ней показатели, лишь приложив палец к камере смартфона [2]. Существуют разработки в области дистанционной ФПГ — методики снятия ФПГ бесконтактным способом с помощью видеокамеры. Таким образом, снятие ФПГ, в отличие от ЭКГ, возможно не только в клинических, но и в домашних и даже в походных условиях, а использование носимых устройств позволяет обеспечить постоянный мониторинг. Сама процедура также является более простой, быстрой и комфортной для пациента — не нужно закреплять на теле электроды, достаточно закрепить на пальце «прищепку» или приложить палец к датчику, а в случае с носимыми устройствами или дистанционной ФПГ не требуется даже этого. Доступность средств автоматизированной диагностики по ФПГ и их удобство для пациента способствуют раннему обнаружению сердечно-сосудистых заболеваний [3].

1. Обзор литературных источников

В последнее время для автоматизированной диагностики по ФПГ все чаще используются методы машинного обучения (МО), в том числе, искусственные нейронные сети (ИНС). Данные технологии уже продемонстрировали свою высокую эффективность в различных задачах в медицине, и уже существует немало работ, описывающих их применение для диагностики по ФПГ.

В работе [4] были использованы классические методы МО для обработки данных ФПГ с умных часов и обнаружения: синусового ритма, фибрилляция предсердий, преждевременные сокращения желудочков или предсердий. Для каждого 30-секундного сегмента ФПГ был построен график Пуанкаре в виде растрового изображения и на его основе выделено 70 признаков с помощью различных методов обработки изображений. Для классификации были применены две модели МО: случайный лес и метод опорных векторов. При этом случайный лес продемонстрировал более высокие значения метрик качества.

В работе [5] ФПГ использовалась для классификации различных сердечных аритмий: тахикардии, брадикардии, желудочковой тахикардии и трепетания/фибрилляции желудочков. Предобработка данных включала полосовой фильтр, сглаживание с помощью фильтра скользящего среднего, устранение блуждающей нулевой линии с помощью вейвлет-преобразования, нормализацию сигнала и разбиение на 10-секундные сегменты. Был выделен 41 признак, включая морфологические и частотные признаки, и с помощью метода главных компонент были отобраны наиболее важные признаки. На полученных данных были обучены 4 модели МО: дерево решений, метод опорных векторов, метод k-ближайших соседей и ансамбль моделей. Значения метрики accuracy для полученных моделей составили 94 %, 98,4 %, 98 % и 98 % соответственно.

В работе [6] была использована глубокая сверточная ИНС для классификации по ФПГ синусового ритма или одного из 5 типов аритмии: преждевременные сокращения желудоч-

ков, преждевременные сокращения предсердий, желудочковая тахикардия, наджелудочковая тахикардия, фибрилляция предсердий. Предобработка данных включала полосовой фильтр, разбиение сигнала на 10-секундные сегменты, нормализацию сигнала и удаление сегментов с артефактами. В качестве модели использовалась глубокая сверточная нейронная сеть, являющаяся модификацией архитектуры VGGNet-16. Точность многоклассовой классификации по метрике accuracy составила 85 %. Также модель была протестирована для 4 классов (с объединением разных типов преждевременных сокращений и тахикардии) и 2 классов (синусовый ритм и аритмия) — точность составила 90,4 % и 97,8 % соответственно. В работе было проведено сравнение полученной модели с традиционными методами МО (метод k-ближайших соседей, случайный лес, метод опорных векторов) и простой полносвязной ИНС (для которых из сигнала ФПГ были выделены различные признаки). Предложенная авторами глубокая сверточная ИНС превзошла эти модели по всем метрикам качества.

В работе [7] для решения той же задачи была применена гибридная модель глубокого обучения Res-BiANet. Данная модель основана на параллельном включении улучшенной сверточной сети ResNet, извлекающей пространственные признаки, и рекуррентной сети BiLSTM с механизмом многоголового самовнимания (Multi-Head Self-Attention), извлекающей временные признаки. Затем полученные наборы признаков вместе подавались на вход двух полносвязных слоев для классификации. Метрики F1 и accuracy для многоклассовой классификации составили 86,88 % и 92,38 % соответственно. В работе было проведено сравнение модели с архитектурами AlexNet, VGG16 и ResNet18. Предложенная авторами модель продемонстрировала наилучшие метрики качества.

В работе [8] решалась задача классификации брадикардии и желудочковой тахикардии. По причине отсутствия общедоступных размеченных датасетов с эпизодами брадикардии и тахикардии, авторами было принято решение использовать искусственно сгенерированные сигналы ФПГ для тренировочной и валидационной выборки, а для тестовой выборки использовать вручную размеченные сигналы из реального датасета. Предобработка данных включала полосовой фильтр, адаптивный нормализованный фильтр наименьших среднеквадратичных значений (для устранения дрейфа), разбиение на 5-секундные сегменты, оценку качества сигнала с помощью спектрального анализа и исключение сегментов с артефактами. Для всех сегментов с помощью непрерывного вейвлет-преобразования были получены скалограммы (двумерное представление сигнала). Для классификации брадикардии и тахикардии по полученным скалограммам были использованы две отдельные 2D сверточные сети. Метрики Sensitivity/Specificity для брадикардии и тахикардии составили 98,1 %/97,9 % и 76,6 %/96,8 % соответственно.

2. Общедоступные наборы данных по ФПГ

Использование технологий МО и ИНС предполагает обучение и тестирование модели на некотором наборе данных. Существуют готовые общедоступные наборы данных (датасеты), которые позволяют упростить исследования и разработки в данной области. Рассмотрим некоторые из существующих общедоступных датасетов, подходящих для задачи выявления сердечно-сосудистых заболеваний по ФПГ.

MIMIC (Medical Information Mart for Intensive Care) [9] — это крупная общедоступная база данных, содержащая обезличенные медицинские данные пациентов отделений интенсивной терапии Beth Israel Deaconess Medical Center (г. Бостон, США). К 2016 году было выпущено 4 версии: MIMIC, MIMIC-II (включена в MIMIC-III), MIMIC-III и MIMIC-IV (частично включает MIMIC-III). База данных MIMIC включает в себя такую информацию, как демографические данные, регулярные измерения жизненно важных показателей, результаты лабораторных анализов, проводимые процедуры, назначаемые лекарства, записи о лицах, осуществляющих

уход, и данные о смертности (как в клинике, так и вне ее). За счет этого MIMIC может использоваться в широком спектре исследований и разработок. В том числе, MIMIC содержит записи физиологических сигналов (включая пальцевую ФПГ, а также ЭКГ, артериальное давление, дыхание и другие сигналы) и периодических измерений числовых показателей (таких как ЧСС и частота дыхания, насыщение крови кислородом, среднее и диастолическое кровяное давление и другие), с прикроватных медицинских мониторов. Они содержатся в отдельном модуле MIMIC Waveform Database.

На сайте Питера Чарльтона имеется большая коллекция наборов данных по ФПГ [10]. Наборы данных MIMIC PERform Training и MIMIC PERform Testing содержат 10-минутные записи ФПГ, ЭКГ и импедансной пневмографии 400 пациентов (взрослых и новорожденных). Датасет MIMIC PERform AF Dataset предназначен для выявления фибрилляции предсердий и содержит 20-минутные записи ФПГ, ЭКГ и импедансной пневмографии 35 пациентов с фибрилляцией предсердий и без нее. Датасет PPG-BP Database предназначен для исследований в области неинвазивного выявления сердечно-сосудистых и связанных с ними заболеваний, а также может быть использован для оценки кровяного давления и старения кровеносных сосудов по ФПГ. Он содержит обезличенные данные 219 пациентов возраста 20–89 лет: 3 коротких (около 3 волн) записи пальцевой ФПГ, пол, возраст, вес, индекс массы тела, систолическое и диастолическое кровяное давление, ЧСС, а также отметки о наличии заболеваний.

Соревнование The PhysioNet/Computing in Cardiology Challenge 2015 [11] было посвящено снижению количества ложных сигналов тревоги об угрожающей жизни пациента аритмии, посылаемых прикроватными мониторами в отделениях интенсивной терапии. Датасет данного соревнования содержит записи физиологических показателей пациентов перед сигналами тревоги об угрожающих жизни видах аритмии: асистолия, экстремальная брадикардия, экстремальная тахикардия, желудочковая тахикардия и трепетание/фибрилляция желудочков. В датасете содержатся записи 750 случаев как верных, так и ложных (присутствуют соответствующие отметки экспертов) сигналов тревоги с прикроватных мониторов в отделениях интенсивной терапии различных госпиталей США и Европы. Для каждого случая содержится запись ЭКГ, а также ФПГ и/или волны артериального давления за 5 минут до сигнала тревоги.

Датасет проекта DeepBeat [12], созданный в Стэнфордском университете (США), предназначен для выявления фибрилляции предсердий по ФПГ, полученной с помощью носимых устройств. Датасет содержит более 500 тысяч 25-секундных сегментов ФПГ, принадлежащих 175 пациентам: 107 с фибрилляцией предсердий и 68 без нее, полученных с помощью носимых на запястье устройств. Сигналы имеют уровень шума, соответствующий реальному качеству сигнала ФПГ, которое могут обеспечить носимые устройства.

3. Классификация сердечных аритмий на основе данных ФПГ

Анализ литературных источников показал, что существует два основных подхода к решению задачи классификации сердечных аритмий на основе данных ФПГ:

1. Подход, основанный на выделении признаков, описывающих сигнал ФПГ. Такими признаками могут быть первичные характеристики сигнала ФПГ, например, систолическая амплитуда, диастолическая амплитуда, интервал между пульсовыми волнами, интервал от пика до пика, признаки, полученные путем статистической обработки характеристик сигнала, признаки, полученные путем обработки изображений графиков или диаграмм, построенных в результате предварительной обработки сигнала и многие другие. Сформированные на основе данных признаков наборы данных используются для решения задач классификации с применением классических методов МО и ИНС.

2. Подход, основанный на анализе сигнала ФПГ как временного ряда. Сюда относятся методы, которые обрабатывают последовательности точек данных и производят классификацию

на основе вида временного ряда, например, формы пульсовой волны. Сюда также можно отнести методы, основанные на обработке изображений сигнала ФПГ.

Достоинством первого подхода является то, что для получения высокой точности классификации можно использовать относительно простые модели МО, такие как метод опорных векторов, случайный лес или метод k-ближайших соседей. К недостаткам данного подхода можно отнести необходимость применения сложных методов предобработки данных для выделения признаков, поиска и отбора наиболее важных признаков. Второй подход позволяет ограничиться более простыми методами предобработки данных такими как удаление шума, артефактов, нежелательных периодических составляющих, но для получения высокой точности классификации требует применения более сложных методов глубокого МО и сложных нейросетевых архитектур.

В данной работе для решения задачи классификации сердечных аритмий был выбран второй подход. В качестве базовой нейросетевой архитектуры, как и в работах [6–8], использовалась сверточная ИНС. Однако в отличии от ИНС, описанных в работах [6–8] авторы решили применить одномерные (1D) сверточные ИНС, которые хорошо зарекомендовали себя при обработке временных последовательностей [13].

При подготовке наборов данных для классификации сердечных аритмий авторы столкнулись с рядом проблем, часть из которых описана в приведенных ранее литературных источниках. Одной из которых является известная проблема сбалансированности классов. Для получения достоверных результатов желательно, чтобы число образцов данных для каждого класса было примерно соизмеримо и не сильно отличалось друг от друга, но это сложно обеспечить, используя реальные наборы данных, в которых записей для пациентов с одними болезнями может быть гораздо больше чем с другими. Другая проблема проявляется в том, что сигнал ФПГ получается в результате наложения сразу нескольких видов волн. Волны первого порядка — это пульсовые волны. Волны второго порядка синхронизированы с ритмом дыхания человека. Волны третьего порядка имеют период, больший, чем период дыхательных волн. Они расцениваются как отражение периодичной активности сосудодвигательного центра (волны Траубе-Геринга) [14]. Волны второго и третьего порядка могут быть слабо выражены или проявляться сильнее, например, при снятии сигнала ФПГ с носимых устройств.

Кроме проблем, связанных с подготовкой данных есть и проблемы, возникающие при обучении сверточных ИНС. Это проблемы затухания и взрыва градиента, характерные для глубоких ИНС. Появление данных проблем может быть вызвано наличием шума и артефактов в сигнале ФПГ, неудачным выбором параметров сверточной ИНС и алгоритмов ее обучения.

Описанные проблемы существенно затрудняют решение задачи классификации сердечных аритмий на основе реальных наборов данных. Поэтому авторы, стремясь снизить трудоемкость процесса подготовки и предобработки наборов данных на основе ФПГ, решили, подобно работе [8], использовать искусственно сгенерированные сигналы ФПГ. Такая стратегия позволяет значительно облегчить подготовку датасетов и обучение сверточной ИНС на начальных этапах исследования. В дальнейшем, в тренировочную и валидационную выборки планируется добавить отобранные и обработанные образцы данных из реальных датасетов. Это позволит приблизить свойства сформированных наборов данных к свойствам реальных датасетов.

Сгенерировать данные ФПГ для таких болезней как мерцательная аритмия, тахикардия и брадикардия можно с помощью приложения SIM_ARR [15]. Приложение позволяет генерировать сигналы ЭКГ и ФПГ на основе заданных значений параметров и сохранять их в файлы формата MATLAB. Для ФПГ можно задавать частоту дискретизации, длительность генерируемых сигналов, измеряемую числом R-R интервалов, медианную длительность эпизодов аритмии в интервалах R-R, процент времени, когда присутствует аритмия. Также возможно моде-

лирование пульсовых волн типа 1, 2, 3a, 3b и 4, и добавление различных типов артефактов. На рис. 1. показан фрагмент сигнала ФПГ типа 2 для случая мерцательной аритмии.

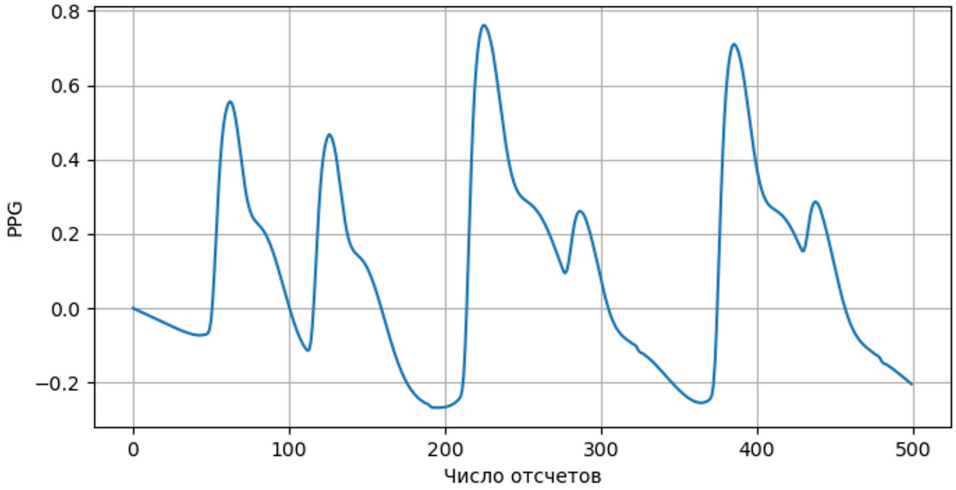


Рис. 1. Фрагмент сигнала ФПГ для случая мерцательной аритмии

Обработка данных включала загрузку данных из файлов MATLAB, разбиение на 5-секундные сегменты, разметку сегментов на классы. Кроме основных классов для синусового ритма, мерцательной аритмии, тахикардии и брадикардии было решено добавить отдельные подклассы для пульсовых волн типа 1, 2, 3a каждого вида сердечной аритмии. Пульсовые волны типов 3a, 3b и 4 имеют очень небольшие отличия, поэтому они все представлялись одним подклассом. Введение классов, связанных с типом пульсовых волн было связано с желанием получить дополнительную информацию о состоянии здоровья пациента и установить связь с такими болезнями как сахарный диабет, атеросклероз или гипертония. Это является особенностью данной работы по сравнению с другими работами, посвященными классификации сердечных аритмий, рассмотренными в обзоре литературы. Таким образом, при решении задачи классификации с использованием всех основных классов с подклассами общее количество классов достигало двенадцати.

Применяемая для классификации сверточная ИНС состояла из трех блоков, включающих 1D сверточный слой, слой батч-нормализации и слой активации “ReLU”. Выход последнего блока поступал на выравнивающий слой и выходной полносвязный слой с числом нейронов, равным числу классов в задаче классификации. Каждый 1D сверточный слой имел 64 фильтра размером 15×15. В качестве алгоритма поиска при обучении ИНС использовался алгоритм “Adam”, в качестве функции потерь использовалась категориальная кросс-энтропия (categorical cross-entropy), а в качестве метрики accuracy. При обучении ИНС размер минибатча принимался равным 128, число эпох равным 70. Такие значение гиперпараметров ИНС были определены в результате их оптимизации. На рис. 2 показаны графики точности, полученные для тренировочной и валидационной выборок в результате обучения сверточной ИНС при решении задачи классификации с 12 классами. Значения метрик F1 и accuracy для данной задачи составило 99,3 %.

Заключение

Таким образом, в работе рассмотрена задача классификации различных сердечных аритмий на основе данных ФПГ. В работе описано современное состояние проблемы, методы, применяемые для классификации сердечных аритмий на основе МО и ИНС, общедоступные наборы данных по ФПГ, которые могут быть использованы для решения подобных задач. От-

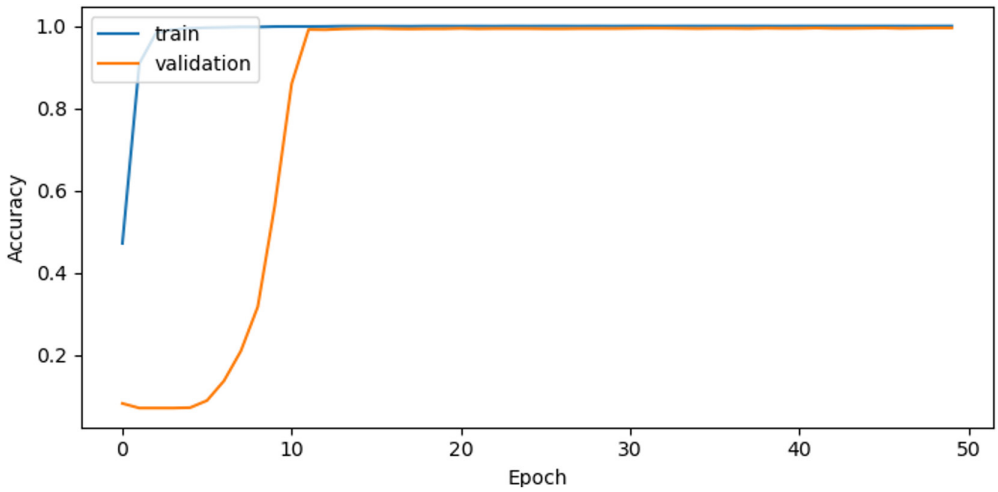


Рис. 2. Графики точности для тренировочной и валидационной выборок

мечено два основных подхода к решению задачи классификации сердечных аритмий на основе данных ФПГ. Из которых авторами был выбран подход на основе анализа сигнала ФПГ как временного ряда. Особенностью рассмотренной в работе задачи классификации, является то, что были добавлены отдельные классы для пульсовых волн разных типов для каждого вида сердечной аритмии. Это позволяет установить связь с такими болезнями как сахарный диабет, атеросклероз или гипертония. В качестве применяемой нейросетевой архитектуры авторы выбрали 1D сверточную ИНС. Значения метрики accuracy полученное при решении задачи классификации с 12 классами составило 99,3 %.

В дальнейшем, к сгенерированным данным планируется добавить отобранные и обработанные образцы данных из реальных датасетов. Это позволит выполнять классификацию на основе реальных данных. Результаты работы могут быть использованы для создания средств автоматизированной диагностики сердечных аритмий и состояния пациентов по данным ФПГ.

Литература

1. *Савостин А. А.* Метод автоматического детектирования характерных точек пульсовой волны / А. А. Савостин, Д. В. Риттер, Г. В. Савостина [и др.] // Труды университета. – 2024. – № 1(94). – С. 498–504. – DOI: 10.52209/1609-1825_2024_1_498

2. *Трофимов П. А.* Измерение вариабельности сердечного ритма человека с помощью камеры смартфона / П. А. Трофимов, К. С. Пуртов, В. С. Кубланов // Компьютерный анализ изображений: Интеллектуальные решения в промышленных сетях (CAI-2016): сборник научных трудов по материалам I Международной конференции, Екатеринбург. – Екатеринбург: ООО «Издательство УМЦ УПИ», 2016. – С. 134–137.

3. *Волков И. Ю.* Фотоплетизмографическая визуализация гемодинамики и двухмерная оксиметрия / И. Ю. Волков, А. А. Сагайдачный, А. В. Фомин // Известия Саратовского университета. Новая серия. Серия: Физика. – 2022. – Т. 22, № 1. – С. 15–45. – DOI: 10.18500/1817-3020-2022-22-1-15-45

4. *Han D.* Digital image processing features of smartwatch photoplethysmography for cardiac arrhythmia detection / D. Han, S.K. Bashar, F. Zieneddin [et al.] // 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). – 2020. – С. 4071–4074. – DOI: 10.1109/EMBC44109.2020.9176142

5. *Qananwah Q.* Cardiac arrhythmias classification using photoplethysmography database / Q. Qananwah, M. Ababneh, A. Dagamseh // Scientific Reports. – 2024. – Vol. 14. – P. 1–10. – DOI: 10.1038/s41598-024-53142-9

6. *Zengding L.* Multiclass arrhythmia detection and classification from photoplethysmography signals using a deep convolutional neural network / L. Zengding, Z. Bin, J. Zhiming [et al.] // Journal of the American Heart Association. – 2022. – Vol. 11. – P. 1–26. DOI: 10.1161/JAHA.121.023555

7. *Wu Y.* Res-BiANet: A Hybrid Deep Learning Model for Arrhythmia Detection Based on PPG Signal / Y. Wu, Q. Tang, W. Zhan [et al.] // Electronics. – 2024. – Vol. 13. – P. 1–25. DOI: 10.3390/electronics13030665

8. *Sološenko A.* Training convolutional neural networks on simulated photoplethysmography data: application to bradycardia and tachycardia detection / A. Sološenko, B. Paliakaitė, V. Marozas [et al.] // Frontiers in physiology. – 2022. – Vol. 13. – P. 1–12. – DOI: 10.3389/fphys.2022.928098

9. MIMIC (Medical Information Mart for Intensive Care). – 2024. – URL: https://mimic.mit.edu/ (дата обращения: 25.10.2024)

10. Photoplethysmography Datasets. – 2024. – URL: https://peterhcharlton.github.io/post/ppg_datasets/ (дата обращения: 25.10.2024)

11. *Clifford G. D.* The PhysioNet/Computing in cardiology challenge 2015: reducing false arrhythmia alarms in the ICU / G. D. Clifford, I. Silva, B. Moody [et al.] // Computing in Cardiology Conference (CinC). – 2015. – P. 273–276.

12. *Torres-Soto J.* Multi-task deep learning for cardiac rhythm detection in wearable devices / J. Torres-Soto, E.A. Ashley // NPJ Digital Medicine. – 2020. – Vol. 3. – P. 1–8. – DOI: 10.1038/s41746-020-00320-4

13. *Шолле Ф.* Глубокое обучение на Python / Ф. Шолле. – СПб. : Питер, 2018. – 400 с. – ISBN 978-5-4461-0770-4.

14. Плетизмография | это... Что такое Плетизмография? – 2024. – URL: https://dic.academic.ru/dic.nsf/enc_medicine/23708/Плетизмография (дата обращения: 25.10.2024)

15. Model for Simulating ECG and PPG Signals with Arrhythmia Episodes v1.3.1 – 2024. – URL: https://physionet.org/content/ecg-ppg-simulator-arrhythmia/1.3.1/zipped_binary/#files-panel (дата обращения: 25.10.2024)

УДК 004.93

ДЕТЕКЦИЯ ОПОР ЛИНИЙ ЭЛЕКТРОПЕРЕДАЧ: НЕЙРОСЕТЕВОЙ ПОДХОД

И. А. Ларин

Воронежский государственный университет

Аннотация. В данной статье рассматривается задача детекции опор линий электропередач на изображениях. В качестве метода был выбран нейросетевой подход и, в частности, архитектура свёрточных сетей последнего поколения YOLOv11. Описан процесс создания датасета для обучения и валидации нейросетевой модели. Рассмотрены особенности аннотирования такого рода изображений и проанализированы полученные результаты. Так, на валидационной выборке средняя точность (mAP) составила 85,5 %, а на тестовой — 87,3 %, что является хорошим показателем качества модели. В заключении представлены дальнейшие планы по улучшению текущей модели и возможного ее применения для задачи автоматического создания карты опор ЛЭП.
Ключевые слова: опоры ЛЭП, свёрточные нейронные сети, детекция объектов, YOLO, YOLOv11.

Введение

Обслуживание и инспекция линий электропередач (ЛЭП) является важной задачей для электрических энергокомпаний. Одним из основным трендов здесь является уменьшение доли ручного труда, что в итоге может существенно повысить эффективность и уменьшит затраты. Одной из начальных задач, которая является первым этапом для множества других, является задача детекции опор. Создание автономной системы автоматической детекции опор могло бы способствовать достижению описанных выше целей. Изображения для детекции могут быть получены с камеры, установленной на машине. В данной статье рассматриваются изображения в том числе и со штатного видеорегистратора.

Поставленную задачу можно решать различными способами. Существуют классические подходы в области компьютерного зрения, когда признаки на изображении выделяются предопределенным набором алгоритмов, и на основе этих признаков система пытается определить положение опор. Нужно признать, что эти подходы обладают рядом недостатков и показали недостаточную эффективность. Также могут быть применены технологии лазерного сканирования. Но такие устройства, как LiDAR (Laser Imaging Detection and Ranging) достаточно дороги или слишком тяжелые, чтобы применять их, например, на дронах.

В настоящее время большую популярность получили нейросетевые подходы. Они доказали свою эффективность и повсеместно используются для задач детекции объектов на изображениях. В этой работе мы создали датасет примерно из 2500 изображений и обучили на нем предобученную модель YOLOv11. На этапе проверки мы оценили эффективность обученной модели на тестовом наборе из 239 изображений. Основной целью данной работы являлось стремление показать эффективность выбранного подхода для решения задачи детекции опор ЛЭП.

1. Методы

1.1. Сбор данных и разметка (аннотирование)

Отметим, что датасет создавался итеративно, и на начальном этапе единственным источником изображений служил штатный видеорегистратор (рис. 1). Изображения формировались на основе видео, сохраняя кадры каждые 2–3 секунды в зависимости от скорости движения. В

дальнейшем для улучшения обобщающей способности нейросети были добавлены изображения, полученные в результате инспекционного обхода линий электропередач (рис. 2). На момент написания статьи последняя ревизия датасета включала 2417 изображений, из которых 1695 изображений содержится в тренировочной выборке, 483 — в валидационной и 239 — в тестовой.

Рис. 1. Изображение с видеорегистратора

Рис. 2. Изображение от обходчиков

Для подготовки датасета был выбран портал Roboflow [1]. Это популярная онлайн-платформа для подготовки и управления датасетами, предназначенными для задач компьютерного зрения, таких как детекция объектов, классификация изображений и сегментация. Она предоставляет удобные инструменты для обработки данных, аннотации и экспорта в различные форматы, которые поддерживаются популярными фреймворками машинного обучения (например, TensorFlow, PyTorch, YOLO и другими).

В процессе аннотирования возник ряд вопросов. В частности, оставалось неясным, следует ли включать в разметку мелкие объекты (рис. 3). На начальном этапе работы было принято решение добавлять в датасет объекты, которые могут быть визуально идентифицированы человеком как опоры на изображениях, предварительно приведенных к разрешению, используемому нейронной сетью. Стоит отметить, что это разрешение составляет 640 × 640 пикселей. На данный момент окончательная обоснованность данного подхода не подтверждена, и в будущем планируется провести дополнительное исследование для его оценки. Еще одним важным вопросом являлось включение скрытых частей опор в ограничивающие прямоугольники. Было принято решение включать эти элементы, что, как показала практика, позволило нейронной сети достаточно эффективно научиться предсказывать скрытые части опор, закрытые другими объектами. Так, на рис. 4 опора частично скрыта кроной дерева.

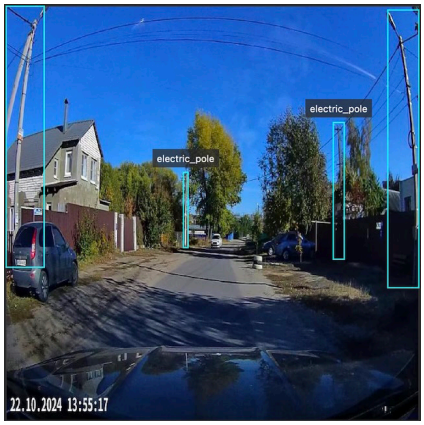


Рис. 3. Аннотирование, далеко расположенных объектов

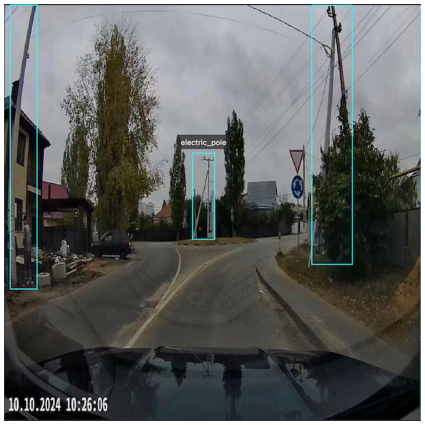


Рис. 4. Скрытые части опор

Как было сказано выше, датасет создавался итеративно, и на ранних стадиях также производилось обучение. Когда размер данных превысил 1000 изображений, нейросеть начала показывать достаточно хорошие результаты, что позволило использовать ее для аннотирования новых изображений и помогло сильно сократить время на дальнейшую подготовку данных.

1.2. Архитектура нейросети

Для нашей задачи детекции опор была выбрана архитектура YOLO (You Only Look Once), которая известна своей высокой скоростью и точностью. YOLO включает несколько версий и масштабов моделей, каждая из которых имеет предобученные веса, основанные на популярных наборах данных (например, COCO), что позволяет эффективно использовать их для задач Transfer Learning. Среди доступных вариантов мы выбрали YOLOv11n, наименьшую модель в семействе моделей YOLOv11 [2]. Она содержит 2.6 миллиона параметров, этого оказалось достаточно для нашей задачи. Данная модель была предобучена на датасете COCO val2017 [3]. Одной из причин, по которой мы выбрали данную сеть, является то, что она достаточно хорошо работает и с мелкими объектами, что для нас является немаловажным фактором.

1.3. Обучение модели

За основу была взята модель YOLOv11n. Она уже предобучена на популярном датасете COCO val2017. В качестве аппаратной платформы для обучения использовался ноутбук на базе процессора M1 pro с 16Gb объединенной памяти (процессор и графический модуль используют единое пространство).

На этапе подготовки датасета не использовались методы аугментации. Дело в том, что библиотека обучения YOLO уже включает возможности, которые позволяют использовать аугментацию сразу в процессе обучения. Не все методы аугментации применимы к нашим данным. В качестве основных были выбраны методы варьирования цветности и яркости изображения, а также весьма эффективный метод создания так называемых мозаик. Аугментации сдвигов и поворотов не применялись по причине того, что, получая изображения с видео, мы уже неявно их применили.

Остановимся подробнее на методе мозаики. Метод аугментации мозаики (Mosaic) — это техника, популяризированная в архитектуре YOLO, особенно начиная с версии YOLOv4. Она заключается в объединении четырех изображений из обучающего набора в одно, с произвольным размещением границ каждого изображения. Это создает новую композицию, где части разных изображений появляются вместе, сохраняя их аннотации. Данная техника позволяет повысить обобщающую способность сети, улучшает детекцию мелких объектов. На последних итерациях все же рекомендуется ее отключать, чтобы процесс обучения стабилизировался [4].

Так, на рис. 5 показан график метрик и значений функций потерь в процессе обучения. При его анализе можем заметить, что значения функции потерь уменьшались как на обучающей выборке, так и на валидационной, что может служить доказательством реального обучения нейросети. Однако отметим одну особенность. В тот момент как до конца обучения оставалось 30 эпох была отключена аугментация методом Mosaic.

После этого значения функции потерь на тренировочной выборке стали уменьшаться заметно быстрее, а на валидационной — немного даже расти. Это может говорить о некотором переобучении, но в то же время основные метрики на валидационной выборке стали увеличиваться, говоря о повышении качества модели. Эти данные подтверждают ранее сказанное о стабилизации процесса обучения в контексте отключения аугментации методом Mosaic на последних итерациях.

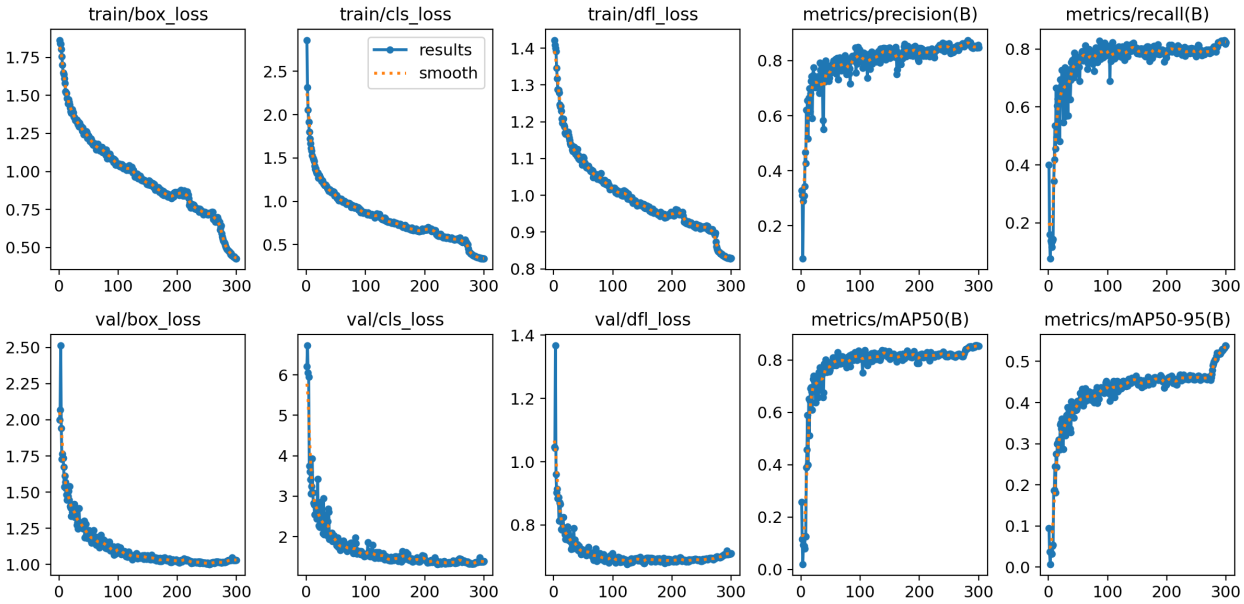


Рис. 5. Графики метрик и функции потерь

1.4. Оценка модели

Для оценки модели, в процессе обучения и после, использовались стандартные метрики mAP50 и mAP50-95. На валидационной выборке средняя точность (mAP) составила 85,5 %, а на тестовой — 87,3 %. Стоит отметить, что данные метрики могут не учитывать специфику конкретной задачи и искажать результаты в худшую сторону. В контексте задачи автоматического создания карты опор обнаружение близких объектов обладает большим приоритетом в сравнении с обнаружением дальних объектов. Для исправления потенциального искажения планируется создать специфическую метрику, которая бы, в частности, фильтровала удаленные объекты и не учитывала их при вычислении.

Также отметим, что в процессе анализа выяснилось, что на тренировочной выборке метрики значительно больше, чем на валидационной и тестовой, что может служить дополнительным поводом для увеличения размера датасета.

2. Результаты и дальнейшие планы

По результатам обучения можно сделать вывод, что нейросеть достаточно эффективно научилась детектировать опоры. Проверка модели показала ее способность определять опоры даже при достаточно сложных условиях (рис. 6–8): опоры на фоне деревьев или частично перекрытые другими объектами.

Дальнейшие исследования планируется проводить в контексте задачи автоматического создания карты опор. Для этого могут быть использованы те же данные видеорегистратора в связке с данными трека. Также есть интерес в использовании изображений сервисов Google Street View или Яндекс панорам. В частности, для использования данных Google Steet View есть ряд программ, позволяющих создавать изображения круговых панорам с информацией о геолокации и ориентации [5].

Другим направлением для дальнейшего исследования может служит задача уменьшения размера модели и увеличение ее быстродействия, используя различные техники. И конечно задача повышения точности сети все еще остается актуальной.

Рис. 6. Опора на фоне леса

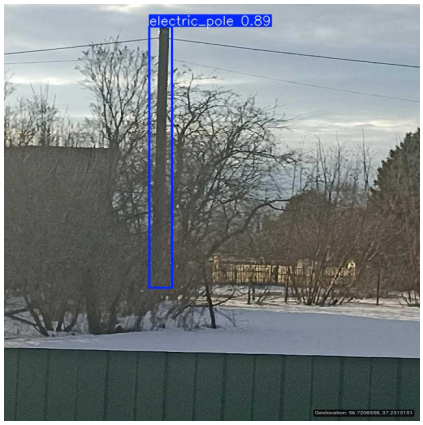


Рис. 7. Деревянная опора в дали за деревом

Рис. 8. Опора среди деревьев

Заключение

В целом поставленная цель исследования была достигнута. Создан датасет достаточных размеров для успешного обучения нейросети. Исследование показало, что предложенный подход вполне успешно может использоваться для задачи детекции опор электропередач.

Литература

1. Roboflow Docs. URL: https://docs.roboflow.com (дата обращения 12.10.2024).

2. *Rahima Khanam, Muhammad Hussain*. YOLOV11: An overview of the key architectural anhancements. arXiv:2410.17725v1 [cs.CV] 23 Oct 2024.

3. COCO val2017. URL: https://cocodataset.org/#home (дата обращения 15.11.2024)

4. Model Training with Ultralytics YOLO. URL: https://docs.ultralytics.com/modes/train (дата обращения 20.11.2024).

5. Street View Download 360. URL: https://svd360.com (дата обращения 20.11.2024).

УДК 004.8

ПРОГНОЗИРОВАНИЕ В УСЛОВИЯХ НЕПОЛНЫХ ДАННЫХ

А. В. Лепендин, Т. В. Азарнова

Воронежский государственный университет

Аннотация. Статья посвящена проблеме прогнозирования в условиях неполных данных, которая актуальна для многих областей науки и практики. Авторы рассматривают различные типы пропусков в данных (MCAR, MAR, MNAR), а также расширенные классификации, такие как неопределенные, условно зависящие и структурные пропуски. В работе анализируются традиционные и современные методы обработки пропущенных значений, включая простые импутации, множественное вменение, регрессионные модели, а также специализированные подходы, такие как скрытые марковские модели и байесовские методы. Особое внимание уделено методам подмоделей паттернов и стратегии полу-контролируемого обучения, которые демонстрируют высокую эффективность в условиях сложных механизмов пропусков. Результаты исследования подчеркивают важность выбора адекватных методов в зависимости от природы пропусков и целей анализа.
Ключевые слова: неполные данные, пропущенные значения, MCAR, MAR, MNAR, импутация, множественное вменение, прогнозирование, временные ряды, полу-контролируемое обучение, подмодели паттернов.

Введение

В современных задачах прогнозирования исследователи часто сталкиваются с проблемой неполных данных, которая возникает как в процессе сбора данных, так и при их анализе. Условие непостоянства данных приводит к необходимости разработки подходов, устойчивых к различным формам отсутствия информации и способных справляться с изменениями в данных.

Пропуски в данных могут иметь разную причину возникновения. Чаще всего пропуски образуются, когда данные по некоторым независимым переменным (предикторам) не известны, данные могут быть не зафиксированы из-за сбоев системы во время сбора данных, или из-за человеческого фактора во время предварительной обработки данных. [3] Также пропуски могут возникать в данных, представимых в виде временных рядов, когда временные интервалы между наблюдениями нерегулярны, что также возможно при технических неполадках во время записи результатов наблюдений

Пропуски в данных могут быть случайны или не случайны, игнорирование или удаления недостающих значений может привести к предвзятому или дезинформирующему анализу. [1] Для обработки пропусков существует ряд подходов, подбираемых и комбинируемых в зависимости от природы пропусков и целей анализа.

Во многих статьях с разбором задач машинного обучения и интеллектуального анализа проблеме пропусков в данных не уделяется должного внимания. [7] Для результативного прогнозирование в условиях неполных данных требуется углубленный разбор природы недостающих данных и применение современных комбинированных методов прогнозирования. Целью этой статьи является исследование проблемы недостающих данных, разбор существующих методов работы с ними, обзор статей на тему прогнозирования в условиях неполных данных.

1. Пропущенные данные

1.1. Классификация пропусков в данных

В традиционной классификации пропусков данных выделяется 3 основных класса:
Пропуски полностью случайны (MCAR)

Вероятность пропуска не зависит ни от наблюдаемых данных, ни от ненаблюдаемых. Такие пропуски считаются случайными, не влияют на анализ. Эффективными методами работы данными с такими пропусками являются удаление пропусков или простая импутация средним, медианным или наиболее часто встречающимся значением. [10]

Пропуски случайны (MAR)

Пропуски зависят только от наблюдаемых переменных. Например, если мужчины чаще пропускают ответ на конкретный вопрос, то вероятность пропуска связана с полом, но не с отсутствующим значением. Такие пропуски можно моделировать, учитывая корреляцию с другими переменными, и часто используются методы, такие как множественная импутация или подгонка модели для каждого паттерна пропуска. [10]

Пропуски не случайны (MNAR)

Вероятность пропуска связана с тем, каким было бы пропущенное значение, если бы его наблюдали. такие пропуски трудны для анализа. Например, если люди с более высоким доходом избегают вопросов о доходе, то пропуски зависят от самих отсутствующих значений. Для устранения смещения может понадобиться применение специализированных моделей или дополнительные данные о механизме пропусков. [10]

Помимо традиционной классификации пропусков на MCAR, MAR и MNAR, в последние годы также появляются и другие классификации, которые помогают точнее описывать природу отсутствующих данных и выбирать соответствующие методы их обработки:

1. Неопределенные

Если определить, является ли пропуск случайным или неслучайным трудно, его относят к неопределенным. Неопределенные пропуски могут вести себя как MCAR, MAR или MNAR в разных подвыборках данных, и стандартные подходы к их обработке часто не дают точного результата. Для обработки таких пропусков можно использовать гибридные методы, например, комбинацию множественной импутации и специфических моделей для подгрупп данных.

2. Условно зависящие пропуски

Этот тип пропусков схож с MAR, но предполагает, что отсутствие данных зависит от подмножества переменных, которые не всегда наблюдаемы в текущем анализе. Например, если в анкете о доходах пропуски чаще встречаются у людей с высоким уровнем образования, то наличие данных для анализа может зависеть от дополнительной информации о сфере занятости. Такие пропуски лучше обрабатывать методами, которые учитывают дополнительные внешние данные и корреляции.

3. Пропуски, зависящие от латентной переменной

Зависимость пропуска может проявляться и от ненаблюдаемой (латентной) переменной, которая не входит в имеющийся набор данных. Например, в медицинских данных наличие пропусков может зависеть от физического состояния пациента, которое напрямую не измерено, но может быть косвенно связано с другими переменными. Методы обработки здесь включают использование скрытых марковских моделей или факторного анализа для учёта латентных факторов.

4. Структурные пропуски

Такие пропуски связаны со структурой данных или логики их получения. Структурные пропуски лишь отражают неактуальность данных для некоторых наблюдений. Например, в некоторых вопросах опроса, собираются данные только по конкретным группам респондентов, отсутствующие значения других групп могут рассматриваться как структурные пропуски. Такие значения часто исключаются из анализа, так как не содержат значимой информации.

5. Пропуски во временных рядах с неравномерными интервалами

Временные ряды могут иметь пропуски, связанные с нерегулярными интервалами между наблюдениями. Этот вид пропусков возникает, когда события регистрируются случайно, а не по фиксированным временным интервалам. Например, стоимость товара или услуги. В таких

данных пропуски важно учитывать в контексте временной зависимости, и здесь применяются методы временной интерполяции или экстраполяции, а также специализированные модели временных рядов.

Пример линейной интерполяции недостающих значений для временного ряда с нерегулярными временными интервалами между наблюдениями, где точками обозначены значения имеющихся изначально наблюдений (рис. 1):

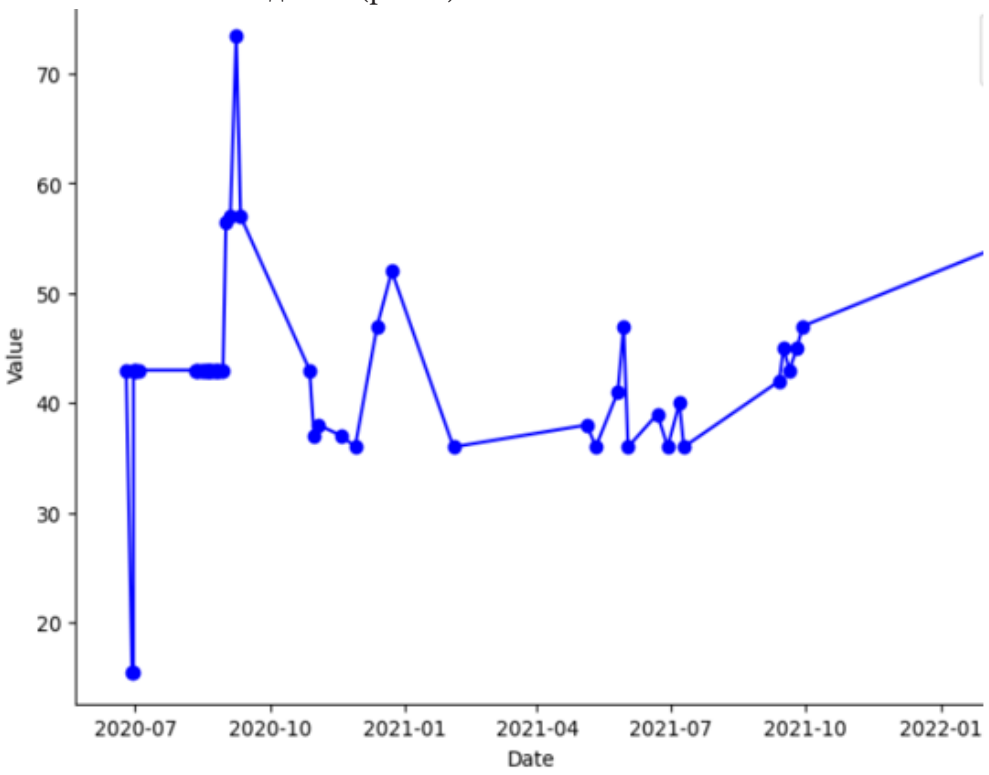


Рис. 1. Интерполяция пропущенных значений временного ряда

6. Селективные пропуски

Селективные пропуски представляют собой комбинацию пропусков, зависимых и независимых от значения. Например, данные могут отсутствовать в зависимости как от конкретной переменной, так и от действия исследователей (например, выборочных замеров или ограниченных ресурсов). Часто такие пропуски обрабатываются множественной импутацией с добавлением моделирования вероятности отсутствия данных на основе дополнительных метрик.

1.2. Методы обработки пропусков в данных

Традиционными методами обработки недостающих данных являются:

• Простое исключение наблюдений с пропущенными значениями. Метод прост, но может привести к потере значительного объема данных и к снижению статистической мощности.

• Заполнение пропусков средним или медианой по выборке, что минимизирует смещение для MCAR, но может ухудшить результаты для MAR и особенно MNAR, поскольку не учитывает структуру данных.

• Для временных рядов используется интерполяция и экстраполяция. Интерполяция подходит для небольших временных интервалов, в то время как экстраполяция может использоваться для предсказания значений за рамками имеющихся данных.

• Метод множественной импутации заполняет каждый пропуск несколькими возможными значениями, основанными на распределении наблюдаемых данных, и анализ проводится

для каждой из возможных версий данных. Этот метод хорошо работает для MAR, так как учитывает вероятностное распределение данных.

• Одним из наиболее предпочтительных статистических методов для обработки пропущенных значений является регрессионное вменение. В данном случае рассчитываются регрессионные уравнения из доступных данных, затем пропуски заменяются прогнозируемым значением на основе регрессионной модели. Эти модели могут быть достаточно точными, особенно если отсутствующие значения MAR, размер выборки достаточно велик, и данные имеют нелинейные взаимосвязи [3].

• Создание подмоделей для каждого паттерна пропуска, где каждая модель подстраивается под свой набор доступных данных. Этот метод показал высокую эффективность в ситуациях с непостоянными данными и разнообразными паттернами пропусков. [6]

Для оценки качества заполнения недостающих данных можно использовать различные способы тестирования, метрики качества. В целом, значения, которые та или иная модель предлагает на место пропусков, обычно отличаются от реальных данных, но соответствуют общим закономерностям в имеющихся данных (рис. 2):

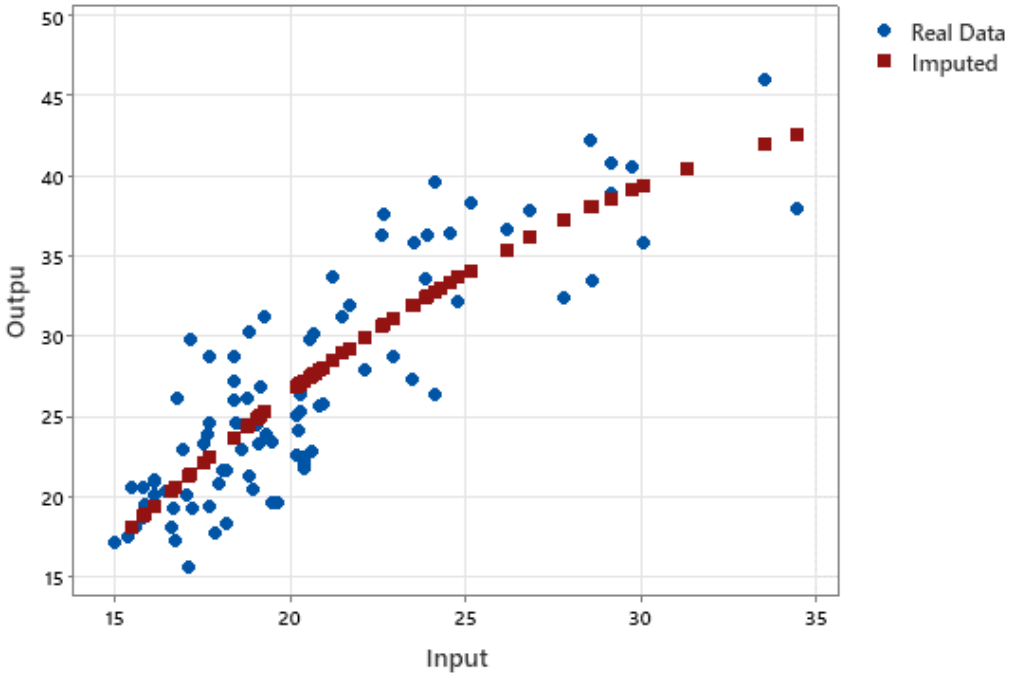


Рис. 2. Реальные значения и значения для заполнения пропусков (синим и красным цветом соответственно)

Приведу еще несколько специализированных методов обработки пропусков:

• Скрытые марковские модели используются для временных рядов с латентными переменными, где вероятности перехода между состояниями помогают учесть нерегулярные интервалы.

• Модели смешивания применяются, когда отсутствующие данные можно разделить на подмножества с различными механизмами пропусков. Они часто используются для MNAR данных, где можно предположить, что пропуски делятся на подгруппы с различными механизмами.

• Факторный анализ позволяет учитывать скрытые факторы, которые могут объяснять пропуски, используется для сложных наборов данных с множеством скрытых факторов.

• Байесовские подходы включают в себя моделирование вероятности пропуска для разных типов и комбинаций пропусков, где неопределённые или условно зависимые пропуски учитываются как вероятностные параметры модели.

Эти расширенные подходы помогают решать задачи прогнозирования и анализа, особенно когда простая классификация MCAR, MAR и MNAR не охватывает всей сложности данных.

Далее рассмотрим некоторые из методов подробнее.

2. Алгоритмы прогнозирования при неполных данных

2.1. Метод подмоделей паттернов

Традиционные методы заполнения пропусков в данных тривиальны в реализации, но часто приводят к неудовлетворительным по точности прогнозам, особенно когда неполные предикторы имеют высокую значимость в предсказании зависимой переменной. [6]

Множественное вменение обычно используется при построении модели, когда значения-заполнители выводятся из условных распределений, наблюдаемых данных для соответствия модели, а коэффициенты комбинируются с использованием правил Рубина. Этот метод повышает эффективность логических выводов за счет использования частичных данных при случайном отсутствии данных. Прогнозы, полученные на основе модели с множественным вменением, усредняются по наборам вменений. Однако применение этого подхода ограничено, если у пользователя есть только опубликованные оценки параметров, а не исходные данные. Для корректного заполнения пропусков в реальном времени требуется комбинировать новые данные с исходными и повторно запускать алгоритм множественного вменения. Этот процесс сложен, характеризуется высокими вычислительными затратами, особенно при использовании таких методов, как, например, метод k-ближайших соседей [6].

Множественное вменение с индикаторами пропусков в чем-то математически схожа с методом подмоделей паттернов (Pattern Submodels). Подход Pattern Submodels заключается в создании модели смесей паттернов и прогнозировании на основе модели, соответствующей профилю нового наблюдения с пропусками. Эффективность и гибкость подхода обеспечивается тем фактом, что подмодели строятся только на данных, относящихся к каждому конкретному паттерну, что позволяет сохранить высокую точность прогнозирования вне зависимости от механизма пропуска данных. PS снижает смещение прогноза за счет учета специфики паттерна. такие результаты достигаются благодаря минимизации специфических потерь для каждого паттерна недостающих данных. При этом возможное снижение точности параметров не оказывает влияния на качество прогноза.

Подход подмоделей паттернов эффективен с точки зрения вычислений, поскольку он обучает ряд моделей, в которых наблюдаются все необходимые данные. Таким образом, требуется выполнить обучения и перекрестную проверку подмоделей паттернов с использованием стандартных алгоритмов. Подмодели паттернов минимизируют потери для каждого конкретного паттерна пропущенных данных, что делает их эффективным инструментом в сравнении со стандартными методами, такими как нулевая или средняя импутация и анализ полных случаев.

В результате симуляции, состоящей из 1000 итераций по 6 шагов авторы статьи Missing data and prediction: the pattern submodel сравнили эффективность подмоделей паттернов с другими популярными методами работы с неполными данными. В результате предложенный ими подход показал лучшие результаты предсказания, степень улучшения в сравнении с другими методами сильно зависела от механизма отсутствия данных и от размерности отсутствующих предикторов. [6]

2.2. Стратегия полу-контролируемого обучения

Нередки случаи, когда предикторы с отсутствующими значениями содержат ценную информацию для прогнозирования зависимой переменной. В базе данных I/B/E/S, где представ-

лены ежеквартальные финансовые прогнозы по доходам публичных компаний, для небольших компаний зачастую отсутствуют данные и аналитические прогнозы. В популярной задаче классификации претендентов на получения банковского кредита, для многих людей отсутствуют данные по ключевому предиктору — кредитному рейтингу. В исследовании A Semi-Supervised Learning Approach to Handling Missing Data in Predictive Analytics авторы предлагают метод подсчета пропущенных данных, основанный на стратегии обучения под частичным наблюдением. В их подходе важным является включение механизма учета пропущенных данных в традиционный процесс подсчета с помощью программы обучения под частичным наблюдением. В отличие от обучения с учителем, в котором модель строится, используя только обучающий набор данных, в их подходе задействуется и тестовый набор данных. Авторы применяют эту стратегию для моделирования пропусков MNAR, которые затем используются для корректировки значения, полученного с помощью традиционных методов расчета. Авторы статьи подробно описывают проблему недостающих данных, а затем аналитически доказывают, что предложенный ими метод может повысить точность вычисления [4].

Стратегия обучения под частичным наблюдением направлена на изучение функции предельной плотности $f()$ от y, которая является прогнозируемой переменной, функция представима по формуле (1):

$$f(y \mid x, \vartheta), \tag{1}$$

параметр ϑ — неизвестный параметр совместного распределения, который необходимо оценить. При обучении под наблюдением параметр может быть найден при максимизации логарифмической функции правдоподобия, формула (2):

$$\vartheta = \arg\max_{\vartheta} \sum_{i=1}^{m} \ln f(x_i, y_i \mid \vartheta), \tag{2}$$

где нижний индекс i индексирует каждое наблюдение, m — это количество экземпляров в обучающем наборе.

При обучении под частичным наблюдением функция правдоподобия примет вид формулы (3):

$$\vartheta = \arg\max_{\vartheta} \sum_{i=1}^{m} \ln f(x_i, y_i \mid \vartheta) + \sum_{i=m+1}^{n} \ln f(x_i \mid \vartheta), \tag{3}$$

где n — количество экземпляров в тестовом наборе. После оценки параметра ϑ совместного распределения (ϑ, ϑ) значения ϑ в тестовом наборе могут быть выведены путем максимизации предельного распределения ϑ. Использование значений независимых переменных в тестовом наборе при обучении под частичным наблюдением позволяет получить более точную оценку ϑ в предположении о гладкости. [4]

Стоит отметить, что предлагаемый подход, хоть в общем и показал лучшие результаты прогнозирования, но не смог превзойти качество прогноза при заполнении пропусков деревом решений Авторы предполагают, что это связано с тем, что деревья решений не задают фиксированный набор параметров. Авторы рекомендуют использовать предлагаемый подход для получения более надежных прогнозов при использовании случайного леса, градиентного бустинга, нейронных сетей для заполнения пропусков.

Заключение

Таким образом, при изучении материалов для написания статьи были разобраны существующие классификации недостающих данных, рассмотрены имеющиеся методы обработки данных с пропусками, был проведен анализ эффективности различных методов в зависимости от механизма пропусков в данных. Дальнейшее исследование будет направлено на поиск, построение и анализ комбинированных моделей прогнозирования в условиях неполных данных.

Литература

1. Теория и практика машинного обучения / В. В. Воронина, А. В. Михеев, Н. Г. Ярушкина, К. В. Святов. – Ульяновск : УлГТУ, 2017. – 290 с.

2. *Хайндман Р.* Прогнозирование : принципы и практика / пер. с англ. А. Логунов / Р. Хайндман, Дж. Атанасопулос. – Москва : ДМК Пресс, 2023. – 382 с.

3. Emmanuel T. A survey on missing data in machine learning / T. Emmanuel, T. Maupong, D. Mpoeleng // Journal of Big Data. – 2021.

4. *Peng J.* A Semi-Supervised Learning Approach to Handling Missing Data in Predictive Analytics / J. Peng, F. Feng // PACIS 2024 Proceedings. 8. – 2024.

5. *Rios R.* Handling missing values in machine learning to predict patient-specific risk of adverse cardiac events: Insights from REFINE SPECT registry / R. Rios // Comput Biol Med. – 2022.

6. Fletcher Mercaldo S. Missing data and prediction: the pattern submodel / S. Fletcher Mercaldo, JD. Blume // Biostatistics – 2020. P. – 236-252.

7. *Nijman S.* Missing data is poorly handled and reported in prediction model studies using machine learning: a literature review / S. Nijman // Journal of Clinical Epidemiology – 2022. P. – 218–229.

8. *Shadbahr T.* The impact of imputation quality on machine learning classifiers for datasets with missing values / T. Shadbahr, M. Roberts, J. Stanczuk // Communications Medicine. – 2023 P. – 139–146.

9. *Saar-Tsechansky M.* Handling missing values when applying classification models / M. Saar-Tsechansky, F. Provost // Journal of Machine Learning Research – 2022. P. – 1625–1657.

10. Центр статистического анализа. – URL: https://scc.ms.unimelb.edu.au/ /resources/ preparing-your-data/missing-data/ (дата обращения: 01.11.2024).

УДК 004.855.5

ПРИМЕНЕНИЕ НЕЙРОННОЙ СЕТИ ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ИСПОЛЬЗОВАНИЯ СРЕДСТВ РАДИОЭЛЕКТРОННОЙ БОРЬБЫ: МЕТОДЫ И СПОСОБЫ ЕЕ ОБУЧЕНИЯ

К. В. Миронов

Воронежский государственный университет

Аннотация. В статье рассматривается проблема эффективности средств радиоэлектронной борьбы, возможности применения в данной сфере искусственного интеллекта, сопутствующие понятия, а также методы и способы обучения нейронной сети.
Ключевые слова: нейрон, нейросеть, БЛА, РЭБ, БАС, веса, смещения.

Введение

Начало специальной военной операции остро поставило вопрос о защите как стратегической, так и гражданской инфраструктуры от БЛА (беспилотных летательных аппаратов), применяемых недружественными странами с целью нанесения ущерба не только военным, но и гражданским объектам на территории, прилегающей к зоне СВО. Для предотвращения возможного поражения объектов, существуют различные способы выявления и подавления БЛА, однако ни один из них не дает высокой эффективности при использовании.

Способы противодействия беспилотным авиационным системам

Одним из самых распространенных способов подавления БЛА является противодроновое ружье. Оно направляется на летящий дрон, и оператор активирует систему подавления на нескольких частотах сразу, и, в случае попадания дрона в частоту противодронового ружья, дрон теряет управление и падает на землю. Ружья имеют небольшое количество диапазонов, поэтому, в случае подлета дрона на нестандартной частоте, действие ружья становится бесполезным. Также ружья имеют небольшой запас энергии батареи и работают не более 20–30 минут. Вторым распространенным способом подавления является применение стационарных комплексов радиоэлектронной борьбы (РЭБ), которые имеют большее число диапазонов подавления и могут питаться от стандартных источников (220 в) или автомобильной сети, но, несмотря на это, тоже закрывают не все диапазоны частот.

В связи с тем, что противодроновые ружья и стационарные системы не имеют функциональности распознавания дронов, эффективность этих систем противодействия зависит от случайности, попала ли частота подавления в диапазон дрона или нет. При этом, если частота дрона и системы противодействия совпадают, то дрон подавляется по этому диапазону, а другие диапазоны продолжают параллельно работать, тем самым, расходуя энергию, снижая общую продолжительность использования устройства. Эту проблему можно решить с помощью добавления в систему подавления подсистемы распознавания дронов. Ее принцип действия может быть основан на использовании нейросети, которая проведет анализ всего диапазона возможных частот для БЛА, сравнит сигнатуры и выдаст точную классификацию дрона с частотой его работы для оператора. В целевом состоянии подсистема автоматически должна не только распознавать дрон, но и сразу активировать систему подавления, включая нужный генератор шума.

Для решения задачи распознавания дрона, может использоваться обычная нейросеть, которую необходимо будет адаптировать для определения вида БЛА и частоты, на которой он работает. Для этого следует белее детально изучить некоторые понятия.

Основы работы нейронной сети

Нейронные сети бывают различных видов и вариантов, но все они используют в своей основе связанную структуру нейронов, которая может состоять из нескольких слоев. Нейрон является самой простой единицей обработки получаемых входных данных аналогично работе нервной клетки головного мозга человека. Он получает сигналы, которые «генерируют другие нейроны, расположенные ранее в цепочке обработки информации», создает свои выходные данные, передающиеся другим нейронам этой нейронной сети, находящимся на другом слое, или формируют ответ подсистемы [4].

Все нейроны в нейросети имеют свои собственные уникальные параметры, которые называются веса и смещения. Веса — это параметр обозначающий значимость каждого входного сигнала в работе отдельно взятого нейрона, а смещением называется показатель, прибавляемый к сумме входных сигналов. Он используется для достижения максимальной эластичности и получения решений в случае большого количества разнообразных данных на входе.

После получения нейроном входных данных, начинается работа с весами и смещениями. Следом нейрон применяет функцию активации, в результате работы которой он либо не активируется, либо происходит активация нейрона и дальнейшая передача сигнала по сети. Функции активации могут быть различными, их нужно подбирать в зависимости от задачи, поставленной нейросети. Например, они могут быть: сигмоидальными, ReLU и многих других видов.

Способы и методы обучения нейросетей

Для понимания принципов работы нейронной сети важно знать, что она обучается путем выставления показателей смещений и весов для получения более точного результата на тренировочных данных. Существует несколько разных способов обучения нейросети.

Процесс обучения с учителем, который также называется «обучение с подкреплением» [2], предлагает создание базы данных обучающих примеров, для того чтобы на их основе обучалась нейронная сеть. Чем больше будет эта база, тем лучше получится обучить сеть. Сами данные, как и результат, передают нейронной сети, а она, в свою очередь, начинает поиск закономерностей, устанавливает взаимосвязи, проводит сопоставление между соответствующими данными для обучения и результатами. Полученный результат сопоставляется с эталонным, и, если нейронная сеть видит, что полученное решение является неверным, то происходит корректировка параметров весов коэффициентов связи, процесс обучения автоматически перезапускается. Вероятность получения неправильных ответов заметно снижается. Все это происходит до тех пор, пока все ответы нейронной сети не совпадут с ответами в базе данных.

Отличительной чертой обучения с учителем является наличие низкого уровня ошибочных ответов сети. Он «может быть получен в результате сравнения реальных данных с прогнозируемыми нейронной сетью. При многократном повторении процесса происходит выявление стоимостной функции. Обучение с учителем подходит для решения вопросов, в которых есть возможность узнать достоверный результат, не прибегая к использованию других нейронных сетей» [1].

Существует и иной вариант обучения нейронной сети, который подразумевает «обучение без учителя, то есть наличие только вводных данных. Чтобы сеть могла из схожих по некоему признаку данных на «входе» выдать результат, обнаруживая взаимосвязи» [5], схожести и закономерности между этими данными, алгоритмы обучения нейронных сетей без учителя корректируют весовые коэффициенты. Во время обучения происходит выделение главных и характерных, для заданных моделей обучающего материала, параметров схожести и дальнейшая классификация этих моделей в группы по их признакам. Данные, поступающие на «вход», после обработки нейросетью складываются в один из нескольких вариантов ответа.

Особенностью обучения таким способом является невозможность прогнозирования конечного результата работы нейронной сети. Это возможно увидеть только тогда, когда процесс обучения подошел к концу, получены первые результаты. Происходит это потому, что нейросеть усваивает необходимые алгоритмы действий, находит закономерности и выстраивает логический процесс обработки данных, имея всего лишь переданную ей информацию. Обучение без учителя применяют для статистических моделей, кластеризации, обнаружения аномалий, языковых моделей и др. задач, где трудно или невозможно заранее узнать ответ.

Далее необходимо рассмотреть методы обучения нейронных сетей:

1. Метод обратного распространения также известен под названием Backpropagation. Он является одним из самых используемых способов обучения и основан на алгоритме вычисления градиентного спуска. Другими словами, расчет локального максимума и минимума функции происходит во время движения вдоль градиента. Поиск значений градиента происходит путем вычисления производной функции в необходимой точке. Имеющаяся точка получит случайное значение веса. В ней надо провести расчет градиента и определить направленность движения спуска. Вычисления делаются по очереди во всех точках до тех пор, пока не будет достигнута точка локального минимума функции, останавливающая дальнейший спуск. Чтобы преодолеть этот требовательный к большим вычислительным мощностям этап, нужно задать такое значение для момента, которое позволит преодолеть участок графика и оказаться в необходимой точке. В случае недостаточного значения преодолеть точку локального минимума не получится, а если значение будет слишком большим, то повысится вероятность потери глобального минимума.

Метод обучения «основан на общении нейронов с помощью синапсов, которые передают входные данные от одного нейрона другому» [3]. Передача осуществляется до тех пор, пока данные не достигнут слоя «выхода», трансформировавшись в ответ. Эта операция именуется как «передача вперед». Для данного типа алгоритмов обучения нейронных сетей необходимо использовать только дифференцируемые функции активации из-за того, что распространение в обратном направлении определяется произведением между производной функцией от входного значения и им, а также разностью между ответами. На следующем шаге необходимо будет высчитать градиент для всех исходящих связей и, с учетом полученных данных, требуется провести перерасчет весов при помощи специальной функции.

2. Метод упругого распространения называют еще Resilient propagation (сокращенно Rprop). Он был предложен как аналог предыдущего способа обучения, который требует слишком много времени и ресурсов, в следствии чего становится неудобным, если необходимо получить результат быстро. Для увеличения скорости обработки операций и уменьшения количества необходимых ресурсов было разработано большое количество второстепенных вспомогательных алгоритмов, в числе их и методика упругого распространения. Данный метод является наиболее популярным при обучении. Используется принцип одного полного прохода датасета через нейронную сеть (epoch). При подгонке весовых коэффициентов данным методом используются лишь знаки производных частного случая. При этом необходимо строго соблюдать правило определения значений коррекции коэффициента веса. Также «следует использовать значение коррекции с известными пределами, чтобы величина веса не оказалась чрезмерно маленькой или наоборот большой. В случае возникновения точки локального максимума, функция меняет свой знак с «+» на «–», что говорит о росте ошибки и необходимости уменьшении веса, а в противоположном случаи — его увеличения. В данной ситуации порядок операций будет таковым:

– определение значения коррекции;
– расчет частных производных;
– расчет новой величины коррекции весовых значений;
– корректировка весов» [1].

Цикл остановится только в одном единственном случаи, если условие остановки алгоритма будет выполнено. В случае не исполнения условия, происходит возврат к пункту два, подсчету производных, и цикл начинает работать заново.

Благодаря методу упругого распространения сходимость нейронной сети достигается в сроки, значительно меньшие, чем при других ранее упомянутых способах.

Для решения поставленной задачи, наиболее целесообразным будет использовать способ обучения с учителем через метод обратного распространения или метод упругого распространения. Оба этих метода хорошо подойдут в данном случаи, но первый метод лучше использовать, если есть большое количество обучающих данных, а второй будет более эффективен при их небольшом количестве.

Сравнение способов обучения нейронной сети

Для того, чтобы выбрать наиболее подходящий способ обучения была создана простая нейросеть [5], имеющая только два слоя. Архитектура данной сети описывается следующим образом:

1. Входной слой задается по формуле:

$$z_1 = W_1 x + b_1, \tag{1}$$

где $z_1 = R^{10}$.

2. Функция активации:

$$a_1 = \mathrm{Re}\,LU(z_1) = \max(0, z_1). \tag{2}$$

3. Выходной слой:

$$z_2 = W_2 a_1 + b_2, \tag{3}$$

где $z_2 = R^{10}$.

Итоговая формула выглядит следующим образом:

$$y = z_2 = W_2 \bullet \mathrm{Re}\,LU(W_1 x + b_1) + b_2. \tag{4}$$

Формула для векторных и матричных преобразований будет следующей:

$$y = \sum_{j=1}^{10} w_{2j} \bullet \mathrm{Re}\,LU(W_1 x + b_1) + b_2. \tag{5}$$

Программная реализация нейронной сети выглядит следующим образом:

```
class SimpleNN(nn.Module):
    def __init__(self):
        super(SimpleNN, self).__init__()
        self.fc1 = nn.Linear(1, 10)# Входной слой
        self.fc2 = nn.Linear(10, 1)# Выходной слой
    def forward(self, x):
        x = torch.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

В созданной сети следующий нейрон использован на обоих слоях, входном и выходном. Его архитектура описывается так:

1. Линейное преобразование:

$$z = (w, x) + b. \tag{6}$$

2. Формула выхода:

$$y = z = (w, x) + b. \tag{7}$$

3. Векторная и матричная форма будет выглядеть следующим образом:

$$y = \sum_{j=1}^{n} W_1 x + b. \quad (8)$$

Программная реализация нейрона выглядит следующим образом:

```
class Neuron:
    def __init__(self, input_size):
        self.weights = np.random.rand(input_size)
        self.bias = np.random.rand()
    def forward(self, inputs):
        return np.dot(inputs, self.weights) + self.bias
```

После обучения нейронной сети с нуля методом обратного распространения и методом упругого распространения, на основе имеющихся достоверных данных о дронах гражданского назначения, были подсчитаны результаты применения обоих алгоритмов после 100 эпох обучения. Датасет составлял из 30 объектов и их характеристики, по 50 % тестовых и обучающих данных. Проведенное исследование позволило проанализировать полученную информацию о взаимосвязи точности прогнозирования схожих и различных с обучающим материалом данных и времени, затраченного на одну эпоху, что нашло свое отражение в табл. 1.

Таблица 1

Сравнение результатов обучения нейросети методами обратного распространения и упругого распространения

Название метода	Время, затраченное на одну эпоху	Точность прогнозирования на основе информации схожей с обучающим материалом	Точность прогнозирования на основе сильно отличающейся от обучающего материала
Метод упругого распространения	10–25 минут	84–90 %	53–27 %
Метод обратного распространения	10–15 минут	91–95 %	33–45 %

После сравнительного анализа можно отметить, что метод упругого распространения оказался более длительным из-за регуляризации, но именно она позволила алгоритму работать значительно более точно, когда поступающая информация не похожа на обучающий материал, а это очень важно в поставленной задаче. Именно поэтому данный метод будет наиболее подходящим для обучения распознанию БЛА в данном конкретном случаи.

Заключение

Подводя итог всему выше изложенному, важно отметить, что нейронные сети требуют глубокого понимания их архитектуры и методов обучения. Правильно выбранный способ обучения и эффективное применение алгоритмов могут привести к высокоскоростной адаптации сети, что в контексте применения подобной системы против беспилотных авиационных систем (БАС), может позволить создать системы, способные обнаружить по радио сигналу вражеский летательный аппарат и классифицировать его в независимости от того, применялся ли данный аппарат до этого или нет, а после применить верный способ противодействия.

Литература

1. Алгоритмы обучения нейронной сети: наиболее распространенные варианты // GeekBrains : [сайт]. – 2022. – URL.: https://gb.ru/blog/algoritmy-obucheniya-nejronnoj-seti (дата обращения 10.11.2024).

2. Введение в обучение с подкреплением: от многорукого бандита до полноценного RL агента// Хабр : [сайт]. – 2017. – URL.:https://habr.com/ru/companies/newprolab/articles/343834 (дата обращения 11.11.2024).

3. *Горбань А. Н.* Обучение нейронных сетей. – М. : СП «ПараГраф», 1990. – 159 с.

4. *Калан Р.* Основные концепции нейронных сетей / Р. Калан. – М. : Издательский дом «Вильямс», 2001. – 288 с.

5. *Харрисон М.* Машинное обучение. Карманный справочник. – Диалектика, 2020. – 320 с.

УДК 004.838.5

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ В МОБИЛЬНЫХ ПРИЛОЖЕНИЯХ: НОВЫЕ ГОРИЗОНТЫ

П. А. Никольникова

Воронежский государственный университет

Аннотация. В статье анализируются ключевые области применения ИИ, включая персонализацию пользовательского опыта, анализ больших данных и компьютерное зрение, рассматриваются преимущества и недостатки, связанные с внедрением ИИ. Статья также освещает перспективные направления развития ИИ в мобильной сфере и его потенциальное влияние на будущее взаимодействия пользователей с мобильными устройствами. В целом, статья представляет обзор современного состояния и будущих перспектив использования ИИ в мобильных приложениях.
Ключевые слова: искусственный интеллект, мобильные приложения, персонализация, обработка естественного языка, компьютерное зрение, анализ больших данных, конфиденциальность данных, доступность, будущее мобильных технологий.

Введение

Мобильные приложения прочно вошли в нашу жизнь, став неотъемлемой частью повседневности. Их функциональность постоянно расширяется, и одним из наиболее перспективных направлений развития является интеграция искусственного интеллекта (ИИ). В статье рассмотрено: как ИИ трансформирует пользовательский опыт, улучшает функциональность приложений, и какие новые возможности он предоставляет в различных сферах, от персонализированных рекомендаций до сложных аналитических задач. От анализа существующих решений до прогнозирования будущих трендов.

Искусственный интеллект — это область компьютерных наук, занимающаяся созданием систем, способных выполнять задачи, требующие умственных усилий. Эти задачи включают понимание естественного языка, распознавание образов, обучение на основе данных и принятие решений [1].

1. Области применения ИИ в мобильных приложениях

Искусственный интеллект находит широкое применение в мобильных приложениях, улучшая пользовательский опыт и повышая эффективность работы. Например, ИИ используется для разработки персонализированных рекомендаций в сферах электронной коммерции и медиаплатформ, а также для реализации интеллектуальных ассистентов, которые способны обрабатывать естественный язык и выполнять команды пользователей. Кроме того, технологии машинного обучения позволяют анализировать большие объемы данных, что способствует оптимизации бизнес-процессов и повышению качества обслуживания клиентов в реальном времени.

1.1. Персонализация

Персонализация, обеспечиваемая с помощью искусственного интеллекта, представляет собой процесс адаптации контента, интерфейсов и функциональности мобильных приложений к уникальным предпочтениям и поведению пользователей. Этот процесс включает в себя несколько ключевых этапов и технологий.

Первым шагом в персонализации является сбор и анализ данных о пользователях. ИИ использует информацию о поведении пользователя, такую как история поиска, выбор товаров, время взаимодействия с приложением и демографические данные. С помощью алгоритмов машинного обучения происходит выявление паттернов и тенденций, что позволяет создать детализированный профиль пользователя.

На основе собранного анализа искусственный интеллект может предлагать персонализированные рекомендации в режиме реального времени. Например, в мобильных приложениях для электронной коммерции ИИ может показывать товары, которые соответствуют интересам пользователя, основываясь на его предыдущих покупках и взаимодействиях. В стриминговых сервисах, таких как Spotify или Netflix, Искусственный интеллект генерирует музыкальные или видеоплейлисты, основываясь на предпочтениях пользователя и его активности.

Ключевым аспектом персонализации является динамичность: алгоритмы ИИ могут адаптироваться к изменениям в поведении и предпочтениях пользователей, обеспечивая актуальные рекомендации. Кроме того, применение методов обработки естественного языка (NLP) позволяет создавать более интуитивные интерфейсы, такие как чат-боты, которые понимают и учитывают запросы пользователей, предлагая им персонализированные решения.

Важно отметить, что успешная персонализация не только повышает уровень удовлетворенности пользователей, но и может значительно увеличить конверсию и удержание клиентов. Однако вместе с этим возникают и вызовы, связанные с соблюдением конфиденциальности данных и этическими аспектами использования искусственного интеллекта, что требует от разработчиков мобильных приложений тщательного подхода к обработке и защите пользовательской информации.

Таким образом, персонализация с использованием ИИ представляет собой мощный инструмент, который, при правильной реализации, способен значительно улучшить пользовательский опыт и повысить конкурентоспособность мобильных приложений.

1.2. Обработка естественного языка (NLP)

Обработка естественного языка (NLP) представляет собой область искусственного интеллекта, занимающуюся взаимодействием между компьютерами и человеческим (естественным) языком. Цель NLP состоит в том, чтобы позволить компьютерам понимать, интерпретировать и генерировать человеческий язык, в результате чего возникают различные приложения, от автоматического перевода до анализа эмоций.

Первые попытки обработки языка начались в 1950-х годах, когда алгоритмы машинного перевода исследовались для перевода текстов с одного языка на другой. Однако начальный успех был ограничен, и исследования замедлились.

Эволюция методов:

• правила и грамматики: первоначально NLP полагалось на использование правил и грамматик, таких как фиксированные шаблоны и контекстно-свободные грамматики;

• статистические методы: в 1990-х годах произошел переход к статистическим методам, которые стали возможны благодаря увеличению объема данных и вычислительных мощностей;

• машинное обучение: в начале 2000-х годов методы машинного обучения, такие как наивный байесовский классификатор и скрытые марковские модели, начали активно использоваться;

• глубокое обучение: совсем недавно, с приходом глубокого обучения, появились мощные модели, такие как recurrent neural networks (RNN) и transformers, что в значительной степени улучшило качество обработки языка.

Перед тем как применять модели, данные часто требуют предварительной обработки, включающей шаги:

• токенизация: разделение текста на токены (слова или фразы);
• удаление стоп-слов: исключение часто встречающихся, но неинформативных слов;
• стемминг и лемматизация: приведение слов к их базовым формам.

Несмотря на прогресс, NLP сталкивается с рядом вызовов:

• амфиболия и неоднозначность языка;
• ограничения в обработке языков с динамичными грамматиками;
• этические проблемы, включая предвзятость в данных и моделях.

Обработка естественного языка продолжает развиваться с учетом новых технологических достижений и улучшения существующих методов. Она открывает новые горизонты для научных исследований и практических приложений, и в будущем можно ожидать еще более глубокого взаимодействия между человеко-ориентированным и компьютерным языком.

1.3. Компьютерное зрение (CV)

Компьютерное зрение (Computer Vision, CV) — это область искусственного интеллекта, нацеленная на предоставление компьютерам способности «видеть» и интерпретировать изображения и видео так же, как это делает человек [2]. Это достигается путем разработки алгоритмов и моделей, которые извлекают информацию из визуальных данных и преобразуют её в понятный для компьютера формат. В основе компьютерного зрения лежат следующие этапы:

1 этап. Получение изображения: процесс захвата изображения с помощью различных устройств, таких как камеры, сканеры или другие сенсоры. Качество изображения существенно влияет на дальнейшую обработку.

2 этап. Предварительная обработка: этап подготовки изображения к дальнейшему анализу. Он включает в себя такие операции, как:

• уменьшение шума: удаление нежелательных артефактов с изображения;
• улучшение контраста: повышение различимости между яркими и тёмными областями;
• выравнивание: коррекция геометрических искажений;
• сегментация: разделение изображения на значимые области (объекты, фоны).

3 этап. Извлечение признаков: этот критически важный этап заключается в выделении характерных особенностей изображения, которые помогают классифицировать или анализировать его содержимое. Методы извлечения признаков варьируются от простых (например, гистограммы) до сложных (например, глубокие сверточные нейронные сети — CNN). CNN являются доминирующим методом в современном компьютерном зрении, извлекая иерархические признаки с различных уровней абстракции.

4 этап. Анализ и интерпретация: на этом этапе извлеченные признаки используются для решения специфических задач, таких как:

• классификация изображений: определение класса объекта (например, кот, собака, автомобиль);
• обнаружение объектов: локализация и классификация объектов на изображении;
• сегментация изображений: разделение изображения на пиксельные области, соответствующие различным объектам;
• трехмерное моделирование: восстановление трехмерной структуры объекта из двухмерных изображений;
• распознавание лиц: идентификация лиц на изображении;
• оптическое распознавание символов (OCR): извлечение текста из изображений.

5 этап. Пост-обработка: на этом этапе результаты анализа обрабатываются и представляются в удобном для пользователя формате.

В современном компьютерном зрении широко используются методы глубокого обучения, особенно CNN, которые продемонстрировали выдающиеся результаты в различных задачах.

Однако, классические методы, такие как методы обработки изображений на основе математической морфологии и алгоритмы Хафа, по-прежнему находят применение в некоторых специализированных задачах.

Мобильные приложения широко используют компьютерное зрение для различных целей: от фильтров и эффектов в приложениях для обработки фотографий до систем дополненной реальности (AR), автоматического перевода и медицинской диагностики. Однако ограничения мобильных устройств, такие как вычислительная мощность и энергопотребление, требуют оптимизации алгоритмов и моделей компьютерного зрения для обеспечения эффективной работы приложений.

2. Преимущества ИИ в мобильных приложениях

Интеграция искусственного интеллекта в мобильные приложения приводит к существенному улучшению пользовательского опыта и расширению функциональности. В следующем разделе будут рассмотрены ключевые преимущества ИИ, позволяющие создавать более интеллектуальные, персонализированные и эффективные мобильные приложения

2.1. Улучшение пользовательского опыта

Пользовательский опыт (UX) является критически важным аспектом разработки мобильных приложений. Конкуренция на рынке приложений растет с каждым днем, и потребители становятся все более требовательными к интерфейсам, функциональности и удобству использования. Искусственный интеллект (ИИ) представляет собой мощный инструмент, который может значительно улучшить UX, предлагая персонализированные решения и автоматизированные процессы. В этом разделе рассматриваются ключевые способы, с помощью которых ИИ меняет подход к пользовательскому опыту.

Один из основных способов, с помощью которых ИИ улучшает пользовательский опыт, — это персонализация. Адаптивные алгоритмы способны анализировать данные о поведении пользователей, их предпочтениях и привычках, что позволяет приложениям предлагать персонализированный контент. Например, рекомендуемые фильмы, музыка или товары могут быть автоматически подстроены под каждого пользователя, что делает взаимодействие более привлекательным и удобным.

Интеграция ИИ в виде чат-ботов и виртуальных помощников позволяет значительно улучшить взаимодействие пользователей с приложениями. Эти системы способны предоставлять мгновенные ответы на вопросы, помогать в навигации и поддерживать пользователей в процессе выполнения задач. Благодаря способности учиться на основе взаимодействий с пользователями, такие технологии становятся все более эффективными и способны предлагать актуальные и полезные решения.

ИИ также имеет значительное влияние на доступность мобильных приложений. Технологии распознавания речи, текстовых и графических интерфейсов помогают сделать приложения более доступными для людей с ограниченными возможностями. Например, голосовые команды и текстовые подсказки могут упростить использование приложения для пользователей с нарушениями зрения или двигательной активности, предоставляя более равные условия для всех.

Искусственный интеллект может анализировать большие объемы данных о пользователях, что позволяет выявлять проблемы в UX и прогнозировать потребности аудитории. С помощью аналитики на основе ИИ разработчики могут лучше понимать, как пользователи взаимодействуют с приложением, что дает возможность оперативно вносить изменения, улучшая интерфейс и функции под конкретные требования пользователей.

Интеграция искусственного интеллекта в мобильные приложения открывает новые горизонты для улучшения пользовательского опыта. Персонализация, интеллектуальные помощники, повышение доступности и анализ данных — все это является важными аспектами, которые помогают создать более интуитивно понятные и привлекательные продукты. В будущем мы можем ожидать, что ИИ станет еще более интегрированным в UX-дизайн, что приведет к новым стандартам взаимодействия пользователя с технологиями.

2.2. Анализ больших данных

Мобильные приложения генерируют огромные объемы данных, которые содержат ценную информацию о поведении пользователей, их предпочтениях и взаимодействии с приложением. Анализ этих данных с помощью искусственного интеллекта позволяет извлекать значимые инсайты, которые могут быть использованы для улучшения приложения, персонализации пользовательского опыта и принятия более информированных бизнес-решений.

• обработка и предобработка данных: перед анализом данные требуют тщательной обработки и предобработки. Этот этап включает в себя очистку данных от шума и пропущенных значений, преобразование данных в подходящий формат, а также извлечение релевантных признаков. Для эффективной обработки больших объемов данных часто используются распределенные системы и технологии Big Data, такие как Hadoop и Spark;

• машинное обучение для анализа данных: для извлечения значимых инсайтов из больших объемов данных применяются различные методы машинного обучения. Например, классификация используется для сегментации пользователей на основе их поведения, регрессия — для прогнозирования будущих действий пользователей, а кластеризация — для обнаружения скрытых паттернов в данных. Глубокое обучение, в частности, рекуррентные нейронные сети (RNN) и трансформеры, могут быть использованы для анализа последовательностей действий пользователей;

• визуализация данных: визуализация результатов анализа данных играет ключевую роль в понимании полученной информации. Интерактивные графики и диаграммы позволяют выявить тенденции, корреляции и другие важные паттерны в данных. Это помогает принимать информированные решения и эффективно коммуницировать результаты анализа заинтересованным сторонам;

• применение результатов анализа: полученные в результате анализа данных инсайты могут быть использованы для улучшения приложения различными способами. Например, результаты классификации пользователей могут быть использованы для персонализации контента, результаты прогнозирования — для проактивной помощи пользователям, а результаты обнаружения скрытых паттернов — для совершенствования функционала и дизайна приложения;

• вызовы и ограничения: анализ больших данных в мобильных приложениях сопряжен с ряд вызовов и ограничений. Это включает в себя обеспечение конфиденциальности и безопасности данных, управление вычислительными ресурсами мобильных устройств, а также необходимость разработки эффективных алгоритмов и моделей, способных работать с большими объемами данных в условиях ограниченной вычислительной мощности.

Анализ больших данных с помощью ИИ является важным инструментом для улучшения мобильных приложений. Правильный подход к обработке, анализу и визуализации данных позволяет извлекать ценную информацию, которую можно использовать для создания более эффективных, персонализированных и удобных приложений. Однако, необходимо учитывать вызовы и ограничения, связанные с анализом больших данных, и обеспечивать соответствие этических норм и законодательных требований.

3. Проблемы, связанные с применением ИИ в мобильных приложениях

Несмотря на многочисленные преимущества, внедрение ИИ в мобильные приложения сопряжено с рядом вызовов и потенциальных проблем. В следующем разделе будут рассмотрены ключевые сложности, которые необходимо учитывать при разработке и внедрении ИИ-решений в мобильные приложения.

3.1. Конфиденциальность

Применение искусственного интеллекта в мобильных приложениях открывает широкие возможности, но одновременно поднимает серьезные вопросы конфиденциальности данных пользователей. Алгоритмы машинного обучения, лежащие в основе многих ИИ-функций, требуют больших объемов данных для обучения и эффективной работы. Эти данные часто включают в себя персональную информацию, такую как местоположение, история поиска, контакты, данные о покупках и другие сведения, которые могут быть использованы для создания подробного профиля пользователя. Несанкционированный доступ к этим данным или их утечка могут привести к серьезным последствиям для пользователей, включая финансовые потери, кражу личных данных и ущерб репутации.

Особенно уязвимы приложения, использующие технологии распознавания лиц, анализа речи и обработки изображений. Эти технологии позволяют собирать и обрабатывать биометрические данные, которые являются особо чувствительной информацией. Недостаточно строгие меры защиты могут привести к злоупотреблению этими данными, например, к созданию фальшивых аккаунтов или использованию биометрических данных для несанкционированного доступа к учетным записям пользователей. Кроме того, отсутствие прозрачности в отношении того, какие данные собираются и как они используются, вызывает обеспокоенность у пользователей и может привести к недоверию к приложениям.

Для решения проблем конфиденциальности необходимо обеспечить соблюдение строгих стандартов защиты данных, включая шифрование, анонимизацию и деидентификацию данных. Важно также разработать прозрачные политики конфиденциальности, которые ясно описывают, какие данные собираются, как они используются и какие меры безопасности применяются. Кроме того, необходимо обеспечить пользователям контроль над своими данными, предоставив им возможность управлять сбором и использованием своей информации. Только при соблюдении этих условий можно минимизировать риски, связанные с использованием ИИ в мобильных приложениях и обеспечить доверие пользователей.

3.2. Доступность

Применение искусственного интеллекта в мобильных приложениях, несмотря на огромный потенциал, сталкивается с проблемой доступности для широкого круга пользователей. Во-первых, многие ИИ-модели требуют значительных вычислительных ресурсов, которые не всегда доступны на мобильных устройствах с ограниченной вычислительной мощностью и объемом памяти. Запуск сложных алгоритмов машинного обучения на смартфонах может приводить к замедлению работы приложения, повышенному потреблению энергии и быстрому разряду батареи. Это особенно актуально для пользователей с устройствами низкого и среднего класса.

Во-вторых, доступность ИИ в мобильных приложениях ограничена также из-за высокой стоимости разработки и обслуживания сложных ИИ-систем. Требуются специалисты с высокой квалификацией в области машинного обучения и глубокого обучения, а также значительные инвестиции в инфраструктуру для обучения и развертывания ИИ-моделей. Эти факторы

могут привести к тому, что ИИ-функциональность будет доступна только в премиум-приложениях или приложениях крупных корпораций, ограничивая доступ к инновациям для небольших компаний и разработчиков.

В-третьих, доступность ИИ зависит от наличия качественных наборов данных для обучения ИИ-моделей. Создание таких наборов данных — задача сложная и затратная. Кроме того, необходимо учитывать вопросы предвзятости данных и их представительности. Использование некачественных или непредставительных наборов данных может привести к неправильной работе ИИ-моделей и к дискриминации отдельных групп пользователей. Решение этих проблем требует разработки новых методов обучения ИИ-моделей с ограниченными ресурсами и более эффективного использования доступных данных.

4. Будущее ИИ в мобильных приложениях

Будущее мобильных приложений неразрывно связано с дальнейшим развитием и внедрением искусственного интеллекта. Ожидается, что ИИ станет не просто дополнительной функцией, а неотъемлемой частью пользовательского опыта, трансформируя взаимодействие с мобильными устройствами и открывая новые возможности. Ключевые направления развития включают:

• расширенная персонализация: ИИ будет использовать еще более сложные модели для предсказания потребностей и предпочтений пользователей. Персонализация выйдет за рамки простых рекомендаций, затрагивая все аспекты взаимодействия с приложением, от дизайна интерфейса до функциональности и контента. Ожидается появление приложений, которые будут адаптироваться к индивидуальным стилям и предпочтениям каждого пользователя в реальном времени;

• улучшенное взаимодействие человек-машина: ИИ будет обеспечивать более естественное и интуитивное взаимодействие с мобильными приложениями. Голосовые помощники станут более интеллектуальными и способными понимать сложные запросы, а системы распознавания образов — более точными и быстрыми. Технологии виртуальной и дополненной реальности будут интегрированы с ИИ для создания immersive пользовательских опытов;

• расширенная автоматизация: ИИ будет использоваться для автоматизации все большего количества задач в мобильных приложениях. Это может включать в себя автоматический контроль и управление системами, автоматическое обслуживание клиентов, а также автоматизированное решение проблем и оптимизацию работы приложения. Автоматизация будет ориентирована на улучшение как пользовательского опыта, так и эффективности работы приложения;

• интеграция с другими устройствами и сервисами: ИИ будет играть ключевую роль в интеграции мобильных приложений с другими устройствами и сервисами, такими как умный дом, умные часы и другие IoT-устройства. ИИ будет обеспечивать бесшовную синхронизацию данных и функциональности между различными устройствами и сервисами, создавая единую экосистему для пользователя

Для реализации полного потенциала ИИ необходимо решать возникающие вызовы и обеспечивать ответственное и этичное использование этих технологий. Только в этом случае мы сможем максимизировать пользу от внедрения ИИ в мобильных приложениях и создать более удобные, эффективные и инновационные продукты для пользователей.

Заключение

Внедрение искусственного интеллекта в мобильные приложения знаменует собой новый этап в развитии мобильных технологий, открывая перед разработчиками и пользователя-

ми безграничные возможности. Мы проанализировали ключевые аспекты применения ИИ в мобильных приложениях, от персонализации пользовательского опыта до автоматизации сложных задач и анализа больших данных. Несмотря на существующие вызовы, связанные с конфиденциальностью, доступностью и этическими аспектами, потенциал ИИ огромный. Дальнейшее развитие этой области приведет к созданию еще более интеллектуальных, персонализированных и интегрированных мобильных приложений, трансформируя способы взаимодействия пользователей с цифровым миром. Однако, критическим фактором успешного внедрения ИИ является ответственный подход, приоритезирующая этичность, безопасность и защита прав пользователей. Только в этом случае можно полностью реализовать потенциал ИИ и создать истинно инновационные и полезные мобильные приложения для всех.

Литература

1. *Russell S.* Artificial Intelligence: A Modern Approach / S. Russell, P. Norvig – 4-е изд. – United Kingdom : Pearson Education, 2021. – 1166 с.

2. *Коул А.* Искусственный интеллект и компьютерное зрение. Реальные проекты на Python, Keras и TensorFlow / А. Коул, С. Ганджу, М. Казам – Санкт-Петербург : Питер, 2023. – 624 с.

УДК 629.113.066

АНАЛИЗ СТАНДАРТНЫХ ПРОТОКОЛОВ ДЛЯ ДИАГНОСТИКИ СПЕЦИАЛЬНЫХ ТРАНСПОРТНЫХ СРЕДСТВ НА ОСНОВЕ ТЕХНОЛОГИИ PASS-THRU И CAN-СТАНДАРТА ПРОМЫШЛЕННОЙ СЕТИ

Н. А. Пеньков, О. А. Сидоркин, С. В. Борисов, И. Али, Д. С. Кирнос

Военный учебно-научный центр Военно-воздушных сил «Военно-воздушная академия имени профессора Н. Е. Жуковского и Ю. А. Гагарина», г. Воронеж

Аннотация. В статье рассматриваются два основных протокола для диагностики специальных транспортных средств. Проводится сравнительный анализ основных способов диагностики транспортного средства. Показано, что для этих целей применяются протоколы J 2534 (Pass-Thru) и ISO 15765-4, которые разработаны на базе CAN-стандарта промышленной сети, объединяющей различные исполнительные устройства и датчики в единую систему. Указаны преимущества и слабые стороны рассматриваемых протоколов. Даются рекомендации, пользуясь которыми пользователь сможет принять решение о целесообразности проведения компьютерной диагностики специального транспортного средства с использованием рассматриваемых протоколов.
Ключевые слова: протокол данных, арбитраж данных, сеть, компьютерная диагностика, транспортные средства, CAN-стандарт, бортовая электросеть, стандарт OBD-II, адаптер.

Введение

Как показывает практика эксплуатации автомобильного парка нашей страны, компьютерная диагностика может выявить до 90 % неисправностей двигателя, коробки передач и бортовой электросети техники различного назначения. Она также позволяет обнаружить проблемы с работой АБС и других электронных систем тормозной системы. В случае с подвеской, диагностируются только активные компоненты, управляемые электроникой. Компьютерная диагностика может помочь выявить скрытые проблемы, которые не проявляются на ранних стадиях. Однако важно помнить, что она не является готовым решением в любой ситуации и не может выявить все неисправности, возникающие в современном автомобиле.

Диагностика автомобилей с помощью компьютера — OBD (On-board diagnostics), представляет собой процесс проверки состояния различных систем машины, осуществляемый бортовым компьютером. Результаты этой проверки доступны водителю, например, через сигнализацию о поломке на приборной панели, а также применяются механиками и специалистами по диагностике. Данные системы начали внедряться с 1980-х годов, а начиная с 1996 года появилась версия OBD-2. В современных системах используются стандартные цифровые порты для передачи актуальных данных и выдачи стандартных кодов ошибок под названием DTC (Diagnostic Trouble Code).

Стандарт OBD-II (On-board Diagnostics) предлагает комплексный контроль за работой двигателя. Он позволяет отслеживать состояние компонентов кузова и вспомогательных устройств, а также проводит диагностику сети управления автомобилем. Различные производители используют разные протоколы связи с автомобилем в рамках этого стандарта, включая варианты, представленные на рис. 1.

Спецификация OBD-II включает стандартизированный аппаратный интерфейс, представленный в виде колодки диагностического разъема (DLC — Diagnostic Link Connector), которая соответствует стандарту SAE J1962 и имеет 16 контактов (2x8) для подключения диагностического оборудования к автомобилю. Этот разъем выполнен в форме трапеции. Отличие от разъема OBD-I, который иногда можно найти под капотом, заключается в том, что разъем OBD-II должен находиться в пределах досягаемости водителя, обычно в районе рулевого колеса.

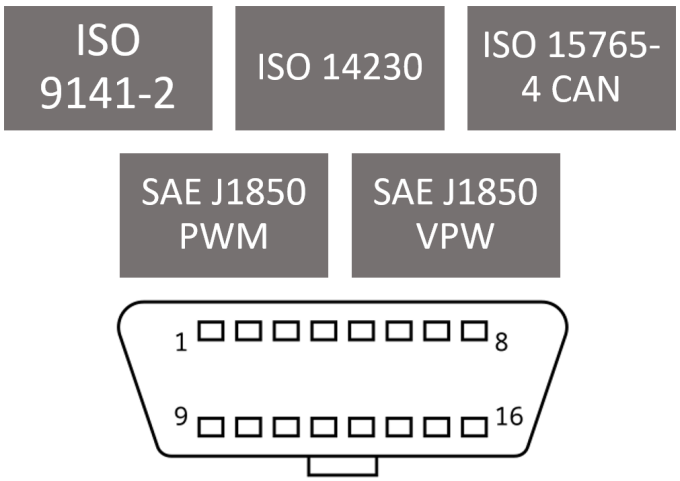


Рис. 1. Диагностический разъем OBD-II и различные протоколы

1. Протокол J 2534 (Pass-Thru)

Стандарт SAE J 2534 был введен в 2002 году для того, чтобы предоставить возможность перепрограммирования электронных блоков управления (ЭБУ) через интерфейсы разных производителей. Этот стандарт определяет общий интерфейс для обмена данными между компьютером и адаптером, позволяя разработчикам ПО использовать адаптеры, использование которых снимает необходимость знания их аппаратной реализации. Это значит, что производитель ПО может применять адаптеры других компаний для программирования блоков управления.

Большинство производителей автомобилей приняли этот стандарт, используя его для программирования и всесторонней диагностики ЭБУ. Компания «Drew Tech» была первой в разработке этого стандарта. Эксперты компании «Toyota» использовали этот интерфейс для реализации полного спектра диагностических процедур на всех моделях Lexus и Toyota. В настоящее время большинство адаптеров поддерживают этот стандарт, что значительно расширяет их функциональные возможности. Стандарт широко распространен в автомобилестроительной индустрии и считается стандартом промышленного применения, способствуя унификации и структурированию систем обмена данными. В работе с этим стандартом задействован обширный спектр специализированных программных решений, которые применяются для перепрограммирования электронных модулей, выполнения диагностических процедур и обслуживания автомобилей.

Текущий адаптер загрузчика CombiLoader не способен работать с интерфейсами типа CAN, поэтому использование адаптера с использованием рассматриваемого протокола J2534 расширяет список поддерживаемых ЭБУ.

На рынке автомобильного оборудования существует множество различных адаптеров, которые поддерживают данный стандарт при выполнении различных задач. Теоретически, любой из таких адаптеров может взаимодействовать с загрузчиком, используемом на специальной технике, но перед использованием неизвестного адаптера рекомендуется проверить его совместимость с загрузчиком на ЭБУ конкретного образца автомобиля.

Преимуществами рассматриваемых адаптеров являются:

– наличие возможности работы с грузовыми автомобилями (бортсеть 24 В);

– поддержка возможности работы на более широком спектре транспортных средств, включая грузовые и специальные автомобили;

– лучшая совместимость с требованиями стандартов J 2534 (*1);

– обеспечение большей гибкости и совместимости с различными стандартными адаптерами, что делает процесс диагностики и программирования более универсальным;

– высокая скорость передачи данных в сравнении с другими адаптерами (скорость обмена данными по протоколу ISO-11898 (CAN) достигает максимальной пропускной способности шины, что значительно превышает скорость работы популярных адаптеров, таких как OpenPort;

– в сравнении с другими адаптерами — лучшее соотношение цены, качества и возможностей.

Однако, предлагаемый адаптер обладает и рядом недостатков:

– высокая стоимость в абсолютном значении;

– для корректной работы необходимо использование дополнительных переходных устройств.

2. Протокол ISO 15765-4

CAN (Controller Area Network) — это стандарт промышленной сети, предназначенный для объединения различных исполнительных устройств и датчиков в единую систему. Этот стандарт может использовать последовательную, широковещательную и пакетную передачу данных.

Стандарт CAN был разработан компанией Robert Bosch GmbH в середине 1980-х годов и широко используется в автомобильной промышленности, промышленной автоматизации, умных домах и других областях использующих микроконтроллеры в своей элементной базе.

В 2000 году ведущие мировые производители признали необходимость единой системы управления и диагностики автомобилей, что привело к созданию нового стандарта — EOBD (EURO OBD). Основным отличием EOBD стало применение протокола управления автомобильными подсистемами CAN, разработанного компанией Bosch и реализованного на моделях автомобилей европейских и азиатских производителей. Однако, диагностический разъем OBD-II остался неизменным, так как в нем уже были предусмотрены контакты для связи по интерфейсу CAN.

Хотя EOBD иногда называют европейским стандартом, а OBD-II — американским, терминология эта некорректна, поскольку EOBD представляет собой стандарт управления, а не диагностики. В США продолжают использовать термин OBD-II, даже если сканеры поддерживают диагностику и управление по CAN.

Протоколы ISO 15765-1, ISO 15765-2, ISO 15765-3, ISO 15765-4 (или ISO 15031-1 и другие для CAN ранних версий) обеспечивают высокую скорость передачи данных (до 1 Мбит/с), что делает их самыми быстрыми среди существующих протоколов.

Преимущества использования такого протокола:

– работа в режиме жесткого реального времени заложена в базовый функционал (устройства могут обмениваться данными с предсказуемым временем реакции, что особенно важно для критических приложений, требующих мгновенного реагирования);

– простота реализации и минимальные затраты на использование (легкость внедрения и поддержки — снижает общую стоимость эксплуатации сети);

– высокая помехоустойчивость (способность эффективно работать в условиях электрических помех благодаря высокой надежности передачи данных);

– наличие арбитража доступа к сети без потерь пропускной способности (распределение доступа к каналу связи между устройствами осуществляется таким образом, чтобы исключить потери производительности из-за конфликтов в формирующихся запросах);

– надёжный контроль ошибок во время процесса приёма-передачи информации (наличие механизмов обнаружения и исправления ошибок, позволяющих повысит надежность передачи данных);

– широкий диапазон скоростей работы (возможность работы на различных скоростях передачи данных, что позволяет адаптироваться к различным приложениям и требованиям);

– широкое распространение технологии, наличие богатого ассортимента продуктов от различных поставщиков (доступность большого количества готовых решений от разных производителей, облегчающая выбор и интеграцию системы).

К недостаткам рассматриваемого протокола следует отнести:

– максимальную длину сети, обратно пропорциональной скорости передачи, что накладывает дополнительные ограничения на возможность её использования (чем выше скорость передачи данных, тем короче допустимая длина линии связи);

– значительный по объему размер служебных данных в транслируемом пакете, что приводит к увеличению нагрузки на сеть и снижению эффективности передачи.

Заключение

Таким образом, приведенный сравнительный анализ протоколов для компьютерной диагностики специальных транспортных средств показал допустимость использования рассмотренных протоколов по своему функционалу, характеристикам, достоинствам и недостаткам для различных потребителей. Однако, для диагностики специальной транспортной техники предпочтительнее использование протокола J 2534 (Pass-Thru), поскольку он подходит для диагностики сети напряжением 24 В, что широко используется в грузовых автомобилях. Кроме того, немаловажным фактором, определяющим выбор протокола J 2534, является возможность получения доступа к программному обеспечению и электронной библиотеке транспортных средств различных сборок и производителей, что позволяет работать с большой базой данных, содержащей информацию об особенностях эксплуатации схожей конструктивно автомобильной техники, что положительно сказывается на диагностике возникающих неисправностей у таких образцов.

Литература

1. *Яковлев В. Ф.* Диагностика электронных систем автомобиля / В. Ф. Яковлев. – «СОЛОН-ПРЕСС». – 200*3. – С. 144–178.*

2. Смирнов Ю. А. Автомобильная электроника и электрооборудование / Ю. А. Смирнов, В. А. Детистов. – М. : Лань. – 2023. – С. 124–150.

УДК 004.8

ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ УЛУЧШЕНИЯ КАЧЕСТВА ПЕРЕДАЧИ ДАННЫХ ПО ЦИФРОВЫМ КАНАЛАМ СВЯЗИ

А. Е. Петров

Курский государственный университет

Аннотация. В статье рассматривается применение нейронных сетей для повышения эффективности цифровых систем связи. Исследуются возможности использования глубокого обучения для задач модуляции, демодуляции и декодирования сигналов в условиях сложных каналов связи. Рассмотренные варианты, включая сверточные, рекуррентные нейронные сети и их модификации, демонстрируют снижение битовой ошибки и приближаются к оптимальной производительности. Уделяется внимание адаптивным методам, способным улучшить качество передачи данных в реальных условиях эксплуатации. Приводятся достоинства моделей цифровых систем с использованием нейронных сетей над традиционными методами.

Ключевые слова: нейронные сети, цифровые каналы связи, помехоустойчивое кодирование, декодирование, модуляция, демодуляция, глубокое обучения, блочные коды, сверточные коды, рекуррентные нейронные сети, сверточные нейронные сети.

Введение

Важной чертой современного мира является огромная связанность общества посредством различных систем коммуникации. Люди со всей планеты могут быстро получить информацию, не выходя из своего дома. Всё это реализуется благодаря сложным цифровым системам связи.

Вместе с развитием технологий, появляются новые методы передачи данных, которые стремятся не только повысить скорость, но и увеличить качество связи в целом. Одной из главной проблем цифровых систем связи являются шумы, которые возникают в каналах связи по различным причинам (например, белый гауссовский шум). Для борьбы с этим используются различные методы, которые позволяют избегать негативных последствий при получении информации. Основными направлениями по борьбе с шумами в каналах связи является разработка новых методов модулирования и демодулирования сигналов, а также применение различных помехоустойчивых кодов, которые позволяют проверять полученные данные на ошибки и исправлять их (например, LDPC коды, турбо-коды и д. р.).

Сегодня большое развитие получили нейронные сети, которые показывают высокие результаты в различных областях деятельности. Их применение может позволить создать новые цифровые системы связи, которые будут более эффективны, чем прежние варианты. На основе нейронных сетей создаются различные демодуляторы и декодеры, которые отлично справляются с последствиями высокой зашумленности каналов связи.

1. Цифровая система связи

Работы цифровой системы связи можно описать несколькими этапами, каждый из которых играет важную роль в эффективности и качестве передачи данных [1]. Можно выделить пять основных этапов.

1. Преобразование исходной информации и кодирование. Данный этап включает в себя обработку исходного вида информации в цифровую форму. Также на данном этапе применяются различные виды помехоустойчивых кодов, для обеспечения надежной передачи данных.

2. Модуляция. Для передачи данных по каналам связи информация из цифрового вида преобразуется в аналоговые сигналы с применением различных видов модуляции.

3. Передача сигнала. На этом этапе сигнал передается по каналу связи, который может быть проводным (например, оптоволоконным или медным) или беспроводным (например, через радиоволны, спутниковые каналы). Сигнал в процессе передачи может быть искажен различными помехами, что требует дополнительных механизмов для исправления ошибок.

4. Прием сигнала и демодуляция. После того как сигнал достигнет приемного устройства, он подвергнется процессу демодуляции, где аналоговый сигнал преобразуется обратно в цифровую форму. Демодуляция включается извлечение переданных бит из полученного сигнала с помощью алгоритмов, соответствующих типу использованной модуляции.

5. Декодирование и обработка ошибок. На данном этапе применяется декодирование с использованием различных методов коррекции ошибок, такие как коды Хэмминга, LDPC или турбо-коды, чтобы восстановить исходные передаваемые данные.

На рис. 1 представлена схема цифровой системы связи.

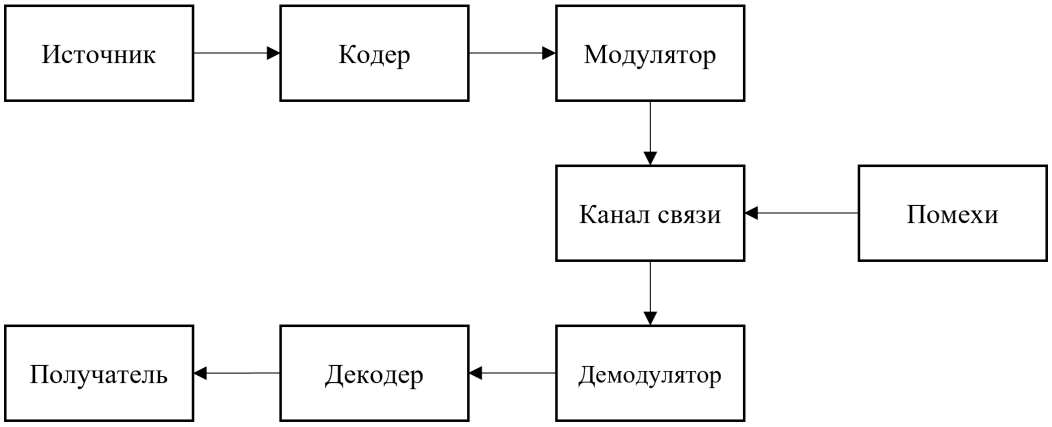


Рис. 1. Схема цифровой системы связи

Нейронные сети в основном могут быть применены для процессов модуляции и демодуляции сигналов, а также для разработки новых декодеров помехоустойчивых кодов. Именно на этих аспектах остановимся подробнее.

2. Модуляция и демодуляция сигнала с применением нейронных сетей

В последнее время нейронные сети стали активно использоваться для создание новых видов модуляторов и демодуляторов сигналов, которые позволяются повысить общую эффективность и помехоустойчивость цифровых систем связи. Новые методы позволяют оптимизировать классические методы, а также создать новые решения, которые могут адаптироваться к изменяющимся условиям передачи данных.

Одной из ветвей развития нейронных сетей в сфере модуляции и демодуляции является разработка адаптивных систем, которые настраивают свои параметры в зависимости от уровня зашумленности каналов связи [2]. Для разработки таких моделей используются различные виды архитектур. Довольно популярными являются сверточные и рекуррентные нейронные сети, а также их гибридные варианты [2, 3]. Такие решения показывают высокие результаты сравнимые с классическими схемами, а также могут их превосходить в некоторых ситуациях.

Другой актуальной проблемой в этой области является распознавание видов модуляции сигналов. Нейронные сети могут существенно облегчить данный процесс, так как уже показали свои возможности в различных задачах классификации. Сверточные модели отлично подходят для распознавания видов модуляции в виде изображений (например, спектрограммы или ди-

аграммы созвездий), где они идентифицируют паттерны соответствующие различным видам модуляции. Для этого могут использоваться известные архитектуры, к примеру, ResNet-50 или Inception-V3, которые показывают высокие результаты в исследованиях [4]. На рис. 2 представлены результаты классификации различных видов модуляции по спектрограммам.

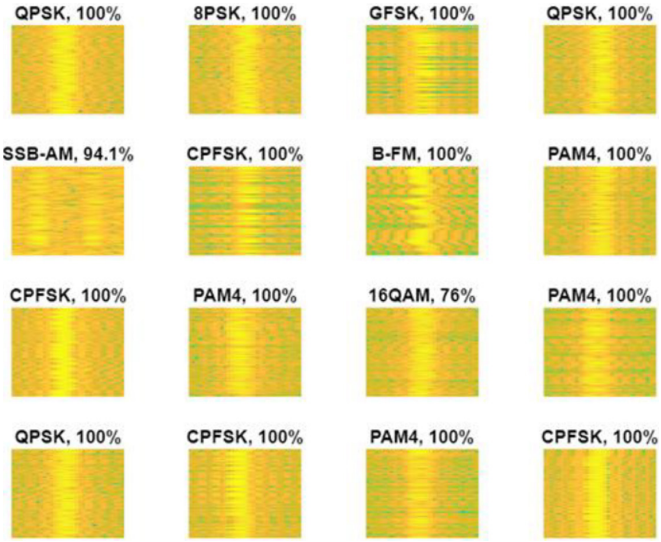


Рис. 2. Результаты работы сети Resnet-50 для классификации модуляции сигнала по спектрограмме

Другой важной стороной исследований применения нейронных сетей в процессе модуляции и демодуляции сигналов является использование автоэнкодеров. Такой тип нейронной сети используются для кодирования и модуляции сигналов. Она может быть обучена восстанавливать сигналы, что позволяет эффективно компрессировать данные перед передачей и минимизировать потерю информации из-за канала с помехами. На основе автоэнкодеров можно построить комплексную систему, в которой передатчик и приемник обучается как единая сеть [5]. Такая архитектура позволяется оптимизировать передачу данных по заданному каналу связи с учетом его характеристик. На рис. 3 представлена схема сквозной системы, основанной на автоэкнодере.

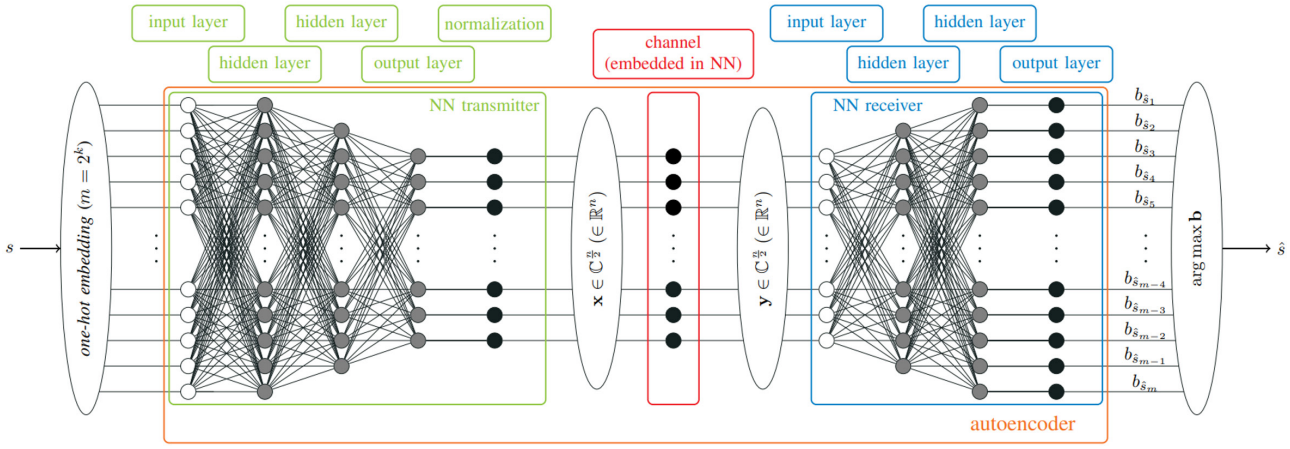


Рис. 3. Схема сквозной системы связи в виде автоэнкодера

По результатам тестирования, было выявлено, что такая система предоставляет более низкий уровень ошибок на блок (BLER) по сравнению с базовой системой QPSK с минимальной средней квадратической ошибкой (MMSE) на всем диапазоне SNR. Помимо этого, включение нелинейного усилителя показало, что автоэнкодеры эффективно справляются с искажениями без дополнительных алгоритмов компенсации [5].

Можно сделать вывод, что такие сети предлагают универсальный инструмент для адаптации к сложным и изменяющимся условиям передачи данных, а также снижается сложность проектирования систем и повышается их гибкость.

3. Применение нейронных сетей для декодирования помехоустойчивых кодов

Помехоустойчивое кодирование играет ключевую роль в обеспечении надежной цифровой передачи данных, особенно в условиях зашумленных и ненадежных каналов связи. Суть этого метода заключается в добавлении избыточной информации к передаваемым данным, что позволяет обнаруживать и исправлять ошибка, вызванные искажениями в процессе передачи. Это особенно важно в системах связи, где сигнал может быть подвержен влиянию различных видов шумов. Благодаря таким кодам, как коды Хэмминга, БЧХ-коды или турбо-коды, передача данных становится более устойчивой к ошибкам, что критически важно для обеспечения качества связи в современных телекоммуникационных системах, интернете и мобильных сетях.

Кроме того, помехоустойчивое кодирование значительно повышает общую эффективность системы, позволяя использовать каналы с низким уровнем сигнала и минимизировать необходимость в повторной передаче поврежденных данных. Основной проблемой является именно декодирование помехоустойчивых кодов при приеме сигнала по каналу связи и именно её рассматривают во многих исследованиях.

С развитием технологий появились новые алгоритмы кодирования и декодирования различных кодов. На замену классическим «жестким» методам декодирования стали приходить новые алгоритмы «мягкого» декодирования, которые предлагают более высокое качество полученных данных, даже в каналах связи с высоким уровнем шума. Для получения более качественных результатов «мягкие» декодеры проводят большее количество вычислительных операций, что сильно может сказаться на эффективности системы в целом. Для решения данной проблемы активно используются нейронные сети [6].

Нейросетевые декодеры, построенные на полносвязных сетях, имеют довольно высокий результат по сравнению с «жесткими» методами декодирования и достигают точность, сравнимую с «мягкими» алгоритмами декодирования [6]. Сейчас в исследованиях используются различные варианты архитектур, для поиска новых решений, которые бы смогли предоставить варианты с высоким качеством декодирования и низкими вычислительными затратами.

Одним из вариантов является использование рекуррентных нейронных сетей. Их преимуществом для декодирования помехоустойчивого кода является то, что они разработаны для выявления зависимостей между элементами данных, что позволяет им эффективно моделировать декодирование кодов, которые требуют учета контекста, как в сверточных или блочных кодах, где структура последовательностей играет ключевую роль. Использование таких моделей показывает в тестах высокие результаты, а именно для линейных кодов можно достичь значения битовой ошибки (BER), которые ниже, чем у традиционных методов декодирования [7]. На рис. 4 представлены результаты сравнения метода декодирования с помощью алгоритма распространения ошибки (BP) и усовершенствованного алгоритма, с использованием рекуррентных нейронных сетей (BP-RNN).

Как можно заметить выигрыш алгоритма BP-RNN варьируется от 0.9 dB до 1.0 dB на различных уровнях SNR. Таким образом, можно сделать вывод о высокой эффективности применения данной архитектуры нейронных сетей для декодирования различных помехоустойчивых кодов в цифровых системах связи.

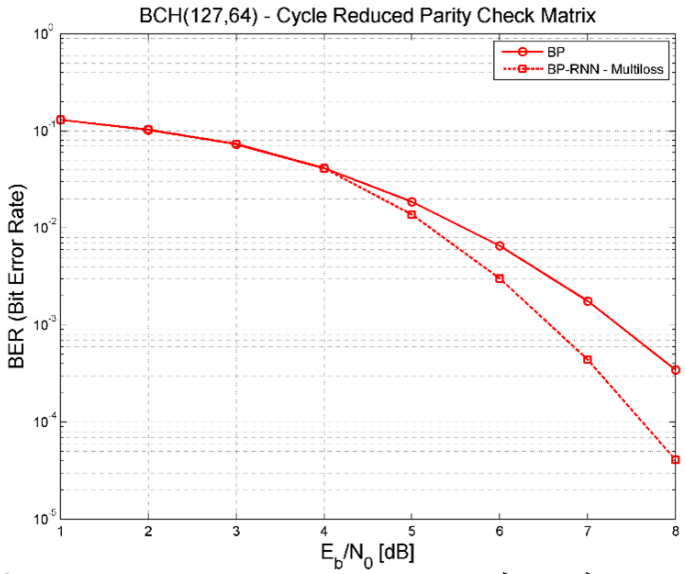


Рис. 4. Сравнение алгоритмов BP и BP-RNN для кода БЧХ (127,64)

Заключение

Современные достижения в области нейронных сетей существенно продвинули цифровые системы связи, предоставив эффективные инструменты для повышения качества передачи данных. Использования различных архитектур нейронных сетей позволяет адаптировать процессы модуляции, демодуляции и декодирования сигналов к сложным условиям каналов, включая шумы и замирания.

Эти успехи открывают перспективы для создания адаптивных и устойчивых коммуникационных систем. Обучаемые алгоритмы нейронных сетей показывают высокую гибкость и эффективность работы. В условиях стремительного роста объемов данных и ужесточения требований к производительности связи нейронные сети становятся ключевым компонентом современных цифровых систем. В дальнейшем развитие этой области будет способствовать еще большему повышению надежности и эффективности телекоммуникационных технологий.

Литература

1. *Скляр Б.* Цифровая связь. Теоретические основы и практическое применение. – 3-е изд. – Москва : Техносфера, 2014. — 1104 с.

2. *Kim H., Jiang Y., Rana R., Kannan S., Oh S., Viswanath P.* Communication algorithms via deep learning // 2018 IEEE International Conference on Communications (ICC). — 2018.

3. *Wu T.* CNN and RNN-based deep learning methods for digital signal demodulation // 2019 IEEE International Conference on Communications (ICC). – 2019. – С. 122–127.

4. *Waqas M., Ashraf M., Zakwan M.* Modulation classification through deep learning using resolution transformed spectrograms // 2023 IEEE International Conference on Communications (ICC). – 2023. – С. 1–6.

5. *Felix A., Cammerer S., Dorner S., Hoydis J.* OFDM-Autoencoder for end-to-end learning of communications systems // 2018 IEEE International Conference on Communications (ICC). – 2018.

6. *Петров А. Е., Кривонос А. В.* Применение нейронных сетей для декодирования блочных турбо-кодов. // Интеллектуальные информационные системы: теория и практика. – 2023. – С. 65–72.

7. Nachmani E., Be'ery Y., Burshtein D. RNN decoding of linear block codes // 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – 2017. – С. 1135–1139.

УДК 004.9

РАЗВИТИЕ ТЕХНОЛОГИИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ ПОСРЕДСТВОМ ИНТЕГРАЦИИ С НЕЙРОННЫМИ СЕТЯМИ

А. И. Расторгуева

Военный учебно-научный центр Военно-воздушных сил «Военно-воздушная академия имени профессора Н. Е. Жуковского и Ю. А. Гагарина», г. Воронеж

Аннотация. Одно из самых популярных направлений информационных технологий дополненная реальность, активно используется для решения разного типа задач. При этом могут возникать ситуации, осложняющие работу с такими приложениями, не позволяющие функционировать им корректно. Подобные вопросы предлагается решить с применением нейронных сетей.
Ключевые слова: дополненная реальность, нейронные сети, библиотеки дополненной реальности.

Введение

Технология дополненной реальности не только развивается, но и активно внедряется в различные сферы деятельности человека. Теперь это не только развлечение, но и элемент навигации (в том числе транспортной), фактор повышения эффективности образовательного процесса, подготовки и переподготовки специалистов различных профилей, инструктаж при выполнении технического обслуживания и ремонта, сопровождение сложных работ на производстве и т. д.

До недавнего времени главными недостатками технологии дополненной реальности были:

1) ограничение по считываю сложных сцен (без однотонного фона), влекущее некорректное распознавание образов;

2) отказ выполнения при недостаточно или при чрезмерно ярком освещении;

3) невозможность различать объекты при условии, что они частично скрыты [1];

4) затруднения с выполнением распознавания, если объект перемещается в поле видения веб-камеры, вращается или меняет свои размеры (приближается/удаляется).

1. Библиотеки дополненной реальности

Приложения дополненной реальности ранее основывались на двух видах подходов: безмаркерный и маркерный. В первом случае дополнительная информация (графическая, текстовая, 3d-объекты и др.) появлялась при запуске приложений или воспроизведение основывалось на геолокации. Во втором случае либо на объект помещалась и в дальнейшем распознавалась специальная метка (маркер), либо объект сам служил меткой для распознавания. Для организации этого использовались библиотеки дополненной реальности, которые сравнивали, полученную с веб-камеры информацию, с шаблонами, внесёнными в программу. К библиотекам дополненной реальности относятся:

1) ARToolKit;

2) OpenCV;

3) NYARToolKit;

4) Vuforia.

Все эти библиотеки приемлемо работают с маркерами, но имеют ряд неостатков:

1) подходят не для всех языков программирования (объектно-ориентированных);

2) в большинстве вариантов библиотек требуется регистрация на сайте производителя;

3) нестабильность при работе с объектами, выступающими вместо маркеров, метками для распознавания;

4) большое количество программных ошибок в самих библиотеках, приводящих к нестандартному поведению приложений.

2. Нейронные сети и дополненная реальность

Развитие современных технологий таково, что появление и развитие нейронных сетей может решить вышеперечисленные проблемы дополненной реальности.

Нейронная сеть (*neural network*) — сеть примитивных обрабатывающих элементов, соединенных взвешенными связями с регулируемыми весами, в которой каждый элемент выдает значение, применяя нелинейную функцию к своим входным значениям, и передает его другим элементам или представляет его в качестве выходного значения [2].

Нейронные сети позволяют дополненной реальности более точно выполнять задачи. При этом работа с приложениями становится более интерактивной и эффективной. Происходит более корректное распознавание объектов и реагирование на их изменения и перемещения.

Становиться значительно шире и спектр возможностей данной технологии:

1) анализ происходящего позволит предлагать рекомендации по выполнению задач, обеспечивая эффективность работы и безопасность (движение на дороге, выполнение опасных работ);

2) анализ полученного изображения в режиме реального времени предоставит рекомендации по дальнейшим действиям (покупки, образовательный процесс);

3) работа с приложениями на основе дополненной реальности теперь может быть индивидуально настроена на конкретного пользователя (важными становятся скорость его реакции, характер, темперамент);

4) улучшаются характеристики навигации (пользователь может точно определить своё место положения);

5) обеспечивается бесперебойность работы дополненной реальности (при условии соответствия системного предложения).

Дополненная реальность расширяет возможности для человека, а нейронные сети расширяют возможности дополненной реальности.

При всех положительных аспектах не стоит забывать, что чем больше информации обрабатывается, тем больше требуются ресурсы, в том числе и энергетические.

Ещё одним недостатком подобного соединения технологий является вопрос сохранения личных данных и их защита от мошенников.

Решение этих проблем — вопрос дальнейших исследований

Заключение

Совместное использование технологии дополненной реальности и нейронных сетей раскрывает невероятный потенциал для работы в разных отраслях. Теперь можно не только дополнять реальный мир вспомогательными объектами, но достичь более чуткой реакции программного обеспечения на действия пользователя, на обстановку, на малейшие изменения и откликаться на возникающие потребности.

Возможности технологии дополненной реальности значительно расширяются благодаря нейронным сетям. Однако, и здесь есть ряд задач, которые ещё предстоит решить. И одна из них это ресурсоемкость, в том числе энергоёмкость.

Литература

1. Хабр: официальный сайт. URL: https://habr.com/ru/articles/709432/ (дата обращения: (19.09.2024)

2. ГОСТ Р 70462.1-2022. Информационные технологии. Искусственный интеллект. Оценка робастности нейронных сетей. Часть 1. Обзор: национальный стандарт Российской Федерации : дата введения 2023-01-01/ Федеральное агентство по техническому регулированию. – Изд. официальное. – Москва : Стандартинформ, 2022. – 32 с.

УДК 004.89

ГРАФОВЫЕ НЕЙРОННЫЕ СЕТИ, ИХ ОСОБЕННОСТИ И ВОЗМОЖНЫЕ ОБЛАСТИ ПРИМЕНЕНИЯ

В. О. Решетов[1], А. А. Арзамасцев[1,2]

[1]Воронежский государственный университет
[2]Тамбовский филиал Федерального государственного автономного учреждения «Национальный медицинский исследовательский центр «Межотраслевой научно-технический комплекс «Микрохирургия глаза» имени академика С. Н. Федорова»

Аннотация. Графовые нейронные сети (GNN) представляют собой мощный инструмент для обработки и анализа данных, представленных в виде графов, что делает их актуальными для широкого круга задач в различных областях науки и техники. В статье рассматриваются основные архитектуры GNN, такие как Graph Convolutional Networks (GCN) и Graph Attention Networks (GAT), их ключевые особенности и принципы работы. Освещены существующие проблемы и ограничения, включая вычислительную сложность и обработку разреженных графов. Особое внимание уделено областям применения GNN: от социальных сетей и биоинформатики до финансовых технологий.
Ключевые слова: графовые нейронные сети (GNN), архитектуры GNN, graph convolutional networks (GCN), graph attention networks (GAT), приложения GNN.

Введение

В последние годы наблюдается значительный рост интереса к графовым нейронным сетям, которые позволяют эффективно работать с данными, представленными в виде графов. Графы широко применяются в различных областях: от социальных сетей и рекомендационных систем до анализа биологических структур, таких как молекулы и белки. В отличие от традиционных нейронных сетей, которые работают с данными, представленными в виде табличных или последовательных структур (например, изображений или временных рядов), графы обладают сложной нерегулярной структурой, что требует создания специализированных методов обработки информации.

Графовые нейронные сети предоставляют инструменты для распространения и агрегации информации между вершинами графа, что позволяет моделям обучаться на основе локальных и глобальных взаимосвязей. Это открывает широкие возможности для решения задач, где важны не только свойства отдельных элементов, но и их взаимосвязи. Например, такие задачи включают предсказание взаимодействий между пользователями в социальных сетях, моделирование молекулярных взаимодействий или анализ сложных физических систем.

Цель работы — рассмотреть основные архитектуры графовых нейронных сетей, их ключевые особенности, а также существующие проблемы и ограничения. Кроме того, в статье проанализированы основные области применения GNN, в которых применение графовых нейронных сетей демонстрирует значительные результаты.

1. Особенности графовых нейронных сетей

1.1. Определение графовых нейронных сетей

Графовые нейронные сети — это обобщение традиционных нейронных сетей для работы с графовыми данными. В отличие от стандартных архитектур нейронных сетей, таких как полносвязные сети (Fully Connected Networks) или сверточные сети (Convolutional Neural

Networks), которые работают с данными фиксированной структуры, GNN способны обрабатывать информацию, представленную в виде графов с произвольной структурой.

Графы — это математические конструкции, которые используются для представления объектов и их связей, где узлы — объекты, а ребра — связи между ними. Каждый узел в графе имеет вектор опций, представляющий его атрибуты. Подобные опции могут быть разных типов. Ребра между узлами также могут иметь связанные опции, собирая более подробную информацию об отношениях между узлами. Это позволяет извлекать более глубокий смысл из данных, расширяя спектр задач в применении глубокого обучения. Графовые нейронные сети применяют прогностические возможности глубокого обучения к структурам данных, которые изображают объекты и их взаимосвязи в виде точек, соединенных линиями на графике. Например, в социальных сетях вершины могут представлять пользователей, а рёбра — связи между ними, такие как дружба или подписки. В транспортных сетях вершинами могут быть города, а рёбрами — дороги между ними. Химические соединения также можно представить в виде графов, где атомы — это вершины, а химические связи — рёбра [1].

Существует три общих типа задач прогнозирования на графах: на уровне графа, на уровне узла и на уровне ребра. В задаче уровня графа предсказывается одно общее для всего графа. Для задачи уровня узла предсказывается некоторое свойство для каждого узла в графе. Для задачи уровня ребра, соответственно, предсказание строится для свойства или наличия ребер в графе. Для трех уровней задач прогнозирования, описанных выше (уровень графа, уровень узла и уровень ребра), важно, что все следующие проблемы могут быть решены с помощью одного класса моделей, GNN.

1.2. Проблема использования графов в машинном обучении

Традиционные нейронные сети, такие как сверточные нейронные сети и рекуррентные нейронные сети, предназначены для работы с данными, имеющими строгую структуру. CNN отлично справляются с анализом изображений, где пиксели упорядочены в виде сетки, а RNN хорошо работают с последовательными данными, такими как текст или временные ряды, где информация поступает по временной оси [2].

Однако многие реальные данные, например, социальные сети, биологические структуры или транспортные системы, не могут быть представлены в виде упорядоченных сетей или последовательностей [4]. Такие данные имеют нерегулярную структуру, где связи между элементами варьируются и могут быть весьма сложными. Это и делает графы отличным способом представления таких данных, однако не очевидно, каким образом представить графы в формате, совместимом с глубоким обучением, ведь модели машинного обучения обычно принимают на вход прямоугольные или сеточные массивы данных.

Графы содержат до четырех типов информации, которые могут быть потенциально полезны для построения предсказаний: узлы, ребра, глобальный контекст и связность. Описать первые три можно достаточно просто: например, для узлов можно сформировать матрицу признаков узлов N путем присвоения каждому признаку индекса i и хранения признака $node_i$ в N. Однако, представление связанности графа немного сложнее, потому что здесь можно применять разные подходы в зависимости от конфигурации графа: матрицы или списки смежности. Применение матриц смежности удобно с точки зрения того, что они легко тензоризируются, но так как количество узлов в графе может быть порядка миллиона и более, а количество ребер между ними изменчиво, то это может приводить к построению очень разреженных матриц смежности, крайне неэффективно расходующих память, к тому же можно построить несколько матриц смежности, которые будут описывать одну и ту же структуру смежности, однако не существует гарантии, что разные матрицы смежности будут давать эквивалентные результаты в глубокой нейронной сети, то есть они не являются инвариантны-

ми к перестановкам. Эти проблемы решают списки смежности. Они описывают связность ребра e_k между вершинами n_i и n_j в виде кортежа (i, j), за счет этого достигается оптимальность хранения графа в случае, если количество вершин в графе сильно превосходит количество ребер [3].

2. Архитектуры графовых нейронных сетей

2.1. Графовые сверточные сети

Графовые сверточные сети — это одна из ключевых архитектур графовых нейронных сетей, которая адаптирует идею сверток из традиционных сверточных нейронных сетей для работы с графами. GCN были предложены для того, чтобы эффективно обрабатывать данные, представленные в виде графов, и использовать информацию о взаимосвязях между вершинами для улучшения предсказательной способности модели.

В основе GCN лежит идея свертки на графе — операция, позволяющая объединять информацию от соседних вершин и обновлять представление каждой вершины на основе этих данных. В случае графов это аналогично тому, как в CNN каждая точка изображения (пиксель) обновляется на основе соседних пикселей. Однако в графах мы имеем нерегулярную структуру, где у каждой вершины разное количество соседей, и ребра могут быть ненаправленными или взвешенными.

Свертка по графу происходит через следующие шаги:

• агрегация, где каждый узел получает представления своих соседей и объединяет их в единое представление;

• нормализация, которая делит вклады соседей на степень вершины (количество её соседей) и помогает сбалансировать вклад каждого соседа, избежать проблемы избыточного усреднения;

• линейная трансформация.

После того как агрегированная информация собрана, она преобразуется с помощью линейной трансформации. Это похоже на линейный слой в традиционной нейронной сети, где к данным применяется обучаемая матрица весов для того, чтобы «обучить» модель выявлять важные признаки. Формально это можно записать как:

$$H^{(k+1)} = \sigma(\tilde{A}H^k W^k), \tag{1}$$

где H^k — матрица признаков на слое k, где каждая строка соответствует представлению (вектору признаков) каждой вершины,

W^k — обучаемая матрица весов для слоя k,

$\tilde{A}$ — нормализованная матрица смежности графа с добавлением единичной матрицы для учёта самой вершины (self-loop),

σ — нелинейная функция активации, например ReLU.

После применения линейной трансформации и нелинейной активации (например, ReLU) каждая вершина получает обновленное представление, которое учитывает информацию не только о самой вершине, но и о её соседях. Этот процесс может повторяться на нескольких слоях сети, чтобы захватить информацию от всё более дальних соседей. GCN использует нормализованную матрицу смежности графа для обеспечения устойчивой агрегации информации. Эта нормализация обеспечивает равномерное распределение вклада каждой вершины в обновлённое представление и предотвращает «раздувание» представлений у вершин с большим количеством связей.

Одной из ключевых характеристик GCN является количество слоёв — глубина сети. Обычно в практике используют от 2 до 3 слоёв, так как более глубокие GCN сталкиваются с пробле-

мой размывания признаков (over-smoothing). Это явление возникает, когда после большого числа слоёв представления всех вершин становятся очень похожими, что снижает способность модели различать разные вершины.

2.2. Graph Attention Networks

Graph Attention Networks — это одна из продвинутых архитектур графовых нейронных сетей, которая использует механизм внимания для более эффективной работы с графовыми данными. В отличие от Graph Convolutional Networks, где информация от всех соседей усредняется с одинаковыми весами, GAT позволяет сети обучать веса, которые отражают важность каждого соседа для текущей вершины. Это делает GAT более гибкими и мощными для задач, где значение связей между вершинами может варьироваться.

Вместо того чтобы одинаково учитывать всех соседей (как это делается в GCN), GAT динамически обучает веса внимания для каждого ребра графа. Модель определяет, какие вершины (соседи) наиболее важны для данной вершины и какова их относительная значимость. Этот процесс осуществляется следующим образом:

• каждая вершина v вычисляет коэффициенты внимания a_{vu} для своих соседей $u \in N(v)$;

• эти коэффициенты показывают, насколько важен каждый сосед u для вершины v при обновлении её состояния. Значения коэффициентов зависят от представлений самой вершины v и её соседа u.

Коэффициенты внимания a_{vu} вычисляются через механизм точечного внимания (dot-product attention), который принимает на вход векторные представления двух вершин — целевой вершины v и её соседа u. Формально, коэффициенты внимания можно описать следующим образом:

$$e_{vu} = \text{LeakyReLU}(a^T [Wh_v \,||\, Wh_u]) \tag{2}$$

где h_v, h_u — это входные представления (векторы признаков) вершин v и u,

W — обучаемая матрица весов, применяемая для линейной трансформации признаков,

a — вектор, который определяет важность взаимодействия между вершинами,

|| — операция конкатенации (объединения) векторов h_v и h_u,

LeakyReLU — нелинейная функция активации для моделирования зависимости между вершинами.

Значение e_{vu} представляет собой ненормализованный коэффициент важности, который затем нормализуется с помощью softmax-функции по всем соседям вершины v. Softmax делает сумму всех коэффициентов внимания для соседей вершины v равной 1. Таким образом, каждый сосед вносит вклад в обновление представления вершины в соответствии с его важностью.

После вычисления коэффициентов внимания a_{vu}, происходит процесс агрегации. Обновлённое представление вершины v формируется как взвешенная сумма представлений её соседей u, где веса — это коэффициенты внимания:

$$h_v^i = \sigma\left(\sum_{u \in N(v)} a_{vu} W h_u \right), \tag{3}$$

где h_v^i — обновлённое представление вершины v,

a_{vu} — коэффициент внимания для соседа u,

W — обучаемая матрица весов для линейной трансформации,

σ — нелинейная функция активации, например, ReLU.

Этот процесс позволяет каждому узлу обновлять свои признаки, уделяя больше внимания важным соседям и игнорируя менее значимых.

GAT представляют собой мощную архитектуру для работы с графами, которая использует механизм внимания для более гибкой и точной обработки информации. В отличие от GCN,

где все соседи вершины вносят равный вклад, GAT позволяет обучать значимость каждого соседа, что делает эту модель более подходящей для задач, где важность связей варьируется.

3. Возможные области применения

Графовые нейронные сети, включая такие архитектуры, как Graph Convolutional Networks и Graph Attention Networks, находят применение в широком спектре задач, где данные естественно представляются в виде графов или имеют сложную структуру взаимосвязей. Рассмотрим ключевые области, где GNN играют важную роль.

3.1. Социальные сети и анализ взаимодействий

Одной из очевидных и популярных областей применения GNN являются социальные сети. В социальных сетях пользователи и их взаимодействия могут быть естественно представлены в виде графа, где вершинами являются пользователи, а рёбрами — их связи (дружба, подписки, сообщения и т. д.). GNN позволяют эффективно моделировать влияние друзей, рекомендовать контент или новых друзей, а также анализировать, как распространяется информация по сети [5].

Применения:

• Рекомендательные системы: GNN используются для персонализированных рекомендаций контента (например, фильмов, музыки или новостей) на основе предпочтений пользователя и его социальной активности.

• Предсказание взаимодействий: Модели на основе GNN могут предсказывать вероятные новые взаимодействия между пользователями (например, предложить подружиться).

• Анализ влияния и распространение информации: GNN позволяют моделировать, как распространяются новости, мемы или вирусные кампании через социальные сети.

3.2. Биоинформатика

Графовые структуры широко используются для моделирования биологических и химических данных. Молекулы можно представить как графы, где вершины — это атомы, а рёбра — химические связи. Белки, гены и их взаимодействия также могут быть представлены как графовые структуры, что делает GNN особенно полезными в биоинформатике [3].

Применения:

• Предсказание свойств молекул: GNN применяются для предсказания физических, химических и биологических свойств молекул на основе их структур. Например, можно предсказать токсичность нового соединения или его активность как лекарственного средства.

• Анализ белковых взаимодействий: Белки взаимодействуют друг с другом в сложных сетях, и GNN могут использоваться для предсказания взаимодействий между белками, что важно для разработки новых лекарств.

• Генетические сети: GNN могут помочь анализировать сети взаимодействий между генами, что может быть полезно в исследованиях генетических заболеваний.

3.3. Распознавание изображений и обработка визуальных данных

Хотя обработка изображений традиционно ассоциируется с Convolutional Neural Networks, GNN могут использоваться для анализа структурированных визуальных данных, таких как 3D-сцены или объекты, представленные в виде графов.

Применения:

• Сегментация и анализ 3D-объектов: GNN могут обрабатывать 3D-объекты и сцены, где поверхности объектов представлены как графы, для задач распознавания и сегментации.

• Обработка изображений с нерегулярной структурой: GNN могут эффективно работать с изображениями, имеющими сложные или нерегулярные структуры, например, в медицинских снимках или спутниковых изображениях.

3.4. Финансовые технологии и анализ транзакций

В финансовой сфере GNN могут помочь анализировать транзакции между людьми, компаниями или организациями, выявляя подозрительные операции или предсказывая важные тенденции в рынке.

Применения:

• Выявление мошенничества: GNN используются для выявления подозрительных транзакций и мошеннических схем, анализируя граф транзакций между пользователями и компаниями.

• Оценка кредитных рисков: В банках и кредитных организациях GNN могут использоваться для оценки финансового риска и определения кредитного рейтинга на основе анализа взаимодействий клиентов и их финансового поведения.

• Инвестиционный анализ: GNN могут помочь предсказать движение цен на рынке акций или криптовалют, анализируя графы финансовых транзакций и взаимодействия участников рынка.

Заключение

В последние годы графовые нейронные сети стали важным инструментом в анализе сложных данных, представленных в виде графов. Способность эффективно обрабатывать и интерпретировать информацию о взаимосвязях между объектами позволяет решать разнообразные задачи в различных областях, таких как социальные сети, биоинформатика, транспорт, финансовые технологии и многие другие. Архитектуры, такие как GCN и GAT, продемонстрировали свои преимущества в моделировании взаимосвязей и динамическом обучении важности соседей.

Ключевые особенности GNN, такие как механизм агрегации и внимательность, позволяют выявлять сложные зависимости и структуру данных, улучшая качество предсказаний и рекомендаций. Благодаря гибкости и способности к обобщению, GNN находят применение в решении практических задач, от анализа социальных взаимодействий до предсказания химических свойств молекул.

С учетом быстрого роста объема данных и сложности современных систем применение графовых нейронных сетей будет лишь расширяться. Однако с увеличением популярности возникают и новые вызовы, такие как необходимость обработки больших графов, обучение на неструктурированных данных и улучшение устойчивости моделей к шуму и изменениям в данных. Будущие исследования должны сосредоточиться на этих аспектах, а также на разработке новых методов и подходов, которые помогут раскрыть весь потенциал GNN.

Таким образом, графовые нейронные сети представляют собой захватывающую и быстро развивающуюся область, открывающую новые горизонты для анализа и обработки данных, предлагая эффективные решения для сложных задач в различных отраслях.

Литература

1. *Stokes J. M.* A deep learning approach to antibiotic discovery // Cell. – 2020. – № 4. – С. 688–702. e13.

2. *Prates M. et al.* Learning to solve np-complete problems: A graph neural network for decision tsp // Proceedings of the AAAI Conference on Artificial Intelligence. – 2019. – № 01. – С. 4731–4738.

3. *Pearce A. A.* Gentle Introduction to Graph Neural Networks. – 2021. – URL: https://distill.pub/2021/gnn-intro/ (дата обращения: 15.10.2024).

4. *Cui G., Wei Z., Su H. H.* Rethinking the Expressiveness of GNNs: A Computational Model Perspective // arXiv preprint arXiv:2410.01308. – 2024.

5. *Liu H.* An Unsupervised Learning Framework Combined with Heuristics for the Maximum Minimal Cut Problem //Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. – 2024. – С. 1921–1932.

6. *Liu Y.* Combinatorial Optimization with Automated Graph Neural Networks //arXiv preprint arXiv:2406.02872. – 2024.

7. *Gavris G. B., Sun W.* Topology optimization with graph neural network enabled regularized thresholding //Extreme Mechanics Letters. – 2024. – С. 102215.

8. *Seo M., Min S.* Graph neural networks and implicit neural representation for near-optimal topology prediction over irregular design domains //Engineering Applications of Artificial Intelligence. – 2023. – С. 106284.

УДК 004.855.5

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ВЫЯВЛЕНИЯ ЦЕНООБРАЗУЮЩИХ ФАКТОРОВ СТОИМОСТИ АВТОМОБИЛЯ

С. В. Решетов

Воронежский государственный университет

Аннотация. В работе произведен анализ машинного обучения для выявления факторов, определяющих стоимость автомобилей. Рассмотрены и использованы такие модели как дерево решений, случайный лес, градиентный бустинг.
Ключевые слова: машинное обучение, дерево решений, случайный лес, градиентный бустинг, автомобили, прогнозирование цен.

Введение

В последние годы рынок автомобилей претерпел значительные трансформации, обусловленные стремительным технологическим прогрессом, изменением предпочтений потребителей и глобальными экономическими тенденциями. В условиях жесткой конкуренции производители стремятся совершенствовать свои предложения, а покупатели ищут оптимальное соотношение цены и качества. В этом контексте становится критически важным провести детальный анализ факторов, влияющих на стоимость автомобилей, чтобы не только понять текущую динамику цен, но и предсказать их будущие изменения.

Машинное обучение, как один из наиболее перспективных методов анализа данных, предоставляет уникальные возможности для выявления скрытых паттернов и взаимосвязей между различными характеристиками автомобилей и их ценами. В рамках исследования будут рассмотрены различные алгоритмы машинного обучения, проведен анализ данных и оценка эффективности моделей. Результаты данного исследования могут быть полезны как для производителей при формировании ценовой стратегии, так и для потребителей при выборе автомобиля.

1. Параметры

На цену автомобилей влияют различные факторы. В данной работе стоимость автомобилей будет оцениваться с помощью следующих параметров:

- компания, производящая автомобиль, марка;
- модель;
- год выпуска;
- пробег;
- состояние.

2. Обработка данных

Перед обучением модели исходные данные требуют предварительной обработки. Первые несколько строк загруженного набора данных представлены на рис. 1.

В наборе данных могут оказаться нулевые значения, поэтому нужно проверить, есть ли пропуски в параметрах, с помощью функции *info*(). На рис. 2 представлена информация о ненулевых параметрах и типах данных. Дополнительно проставлять значения не нужно, нулевых параметров нет.

	Unnamed: 0	Make	Model	Year	Mileage	Condition	Price
0	0	Ford	Silverado	2022	18107	Excellent	19094.75
1	1	Toyota	Silverado	2014	13578	Excellent	27321.10
2	2	Chevrolet	Civic	2016	46054	Good	23697.30
3	3	Ford	Civic	2022	34981	Excellent	18251.05
4	4	Chevrolet	Civic	2019	63565	Excellent	19821.85

Рис. 1. Первые строки загруженного набора данных

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  1000 non-null   int64
 1   Make        1000 non-null   object
 2   Model       1000 non-null   object
 3   Year        1000 non-null   int64
 4   Mileage     1000 non-null   int64
 5   Condition   1000 non-null   object
 6   Price       1000 non-null   float64
dtypes: float64(1), int64(3), object(3)
memory usage: 54.8+ KB
None
```

Рис. 2. Информация о параметрах

На рис. 2 видно, что три параметра не имеют численного представления. Для кодирования данных, не имеющих численного представления, использован метод dummy-кодирования, который заключается в кодировании категориального признака с n возможными значениями с помощью n бинарных признаков. Каждый бинарный признак соответствует одному из возможных значений категориального признака и является индикатором того, что на данном объекте он принимает данное значение. После кодирования всех признаков удаляются лишние исходные столбцы и получается набор данных, представленный на рис. 3.

	Year	Mileage	Price	Make_Chevrolet	Make_Ford	Make_Honda	Make_Nissan	Make_Toyota	Model_Altima	Model_Camry	Model_Civic	Model_F-150	Model_Silverado	Condition_Excellent	Condition_Fair	Condition_Good
0	2022	18107	19094.75	False	True	False	False	False	False	False	False	False	True	True	False	False
1	2014	13578	27321.10	False	False	False	False	True	False	False	False	False	True	True	False	False
2	2016	46054	23697.30	True	False	False	False	False	False	False	True	False	False	False	False	True
3	2022	34981	18251.05	False	True	False	False	False	False	False	True	False	False	True	False	False
4	2019	63565	19821.85	True	False	False	False	False	False	False	True	False	False	True	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	2010	149032	24548.50	False	False	False	True	False	False	True	False	False	False	True	False	False
996	2014	20608	26969.70	True	False	False	False	False	False	False	False	True	False	True	False	False
997	2016	109851	20507.55	False	True	False	False	False	True	False	False	False	False	False	False	True
998	2010	11704	31414.90	False	False	False	False	True	False	False	False	False	True	False	False	True
999	2017	126396	18580.60	False	False	False	True	False	False	False	False	False	True	True	False	False

1000 rows × 16 columns

Рис. 3. Подготовленные для обучения данные

Используемые модели машинного обучения

Для анализа были выбраны следующие модели машинного обучения:
- модель дерева решений;
- модель случайного леса;
- модель градиентного бустинга.

3. Оценка качества регресии

Коэффициент детерминации R^2 — статистический показатель, отражающий объясняющую способность регрессии $f : X \to X$ и определяемый как доля дисперсии зависимой переменной, объяснённая регрессионной моделью с данным набором независимых переменных.

Он показывает, какую часть изменчивости наблюдаемой переменной можно объяснить с помощью построенной модели, то есть определяет долю (в процентах) изменений, обусловленных влиянием факторных признаков, в общей изменчивости результативного признака.

Дерево решений

Дерево решений — это метод, позволяющий предсказывать значения зависимой переменной в зависимости от соответствующих значений одной или нескольких независимых переменных. У данного метода точность R^2 составила 0.9988944007522117 при максимальной глубине дерева (max_depth) равной 8. На рис. 4 представлена диаграмма важности признаков для модели «Дерево решений», которая показывает, как параметры набора данных влияют на модель. Можно сделать вывод, что признаки «Год», «Пробег», влияют на модель, а значит остальные признаки можно удалить для более быстрого обучения модели.

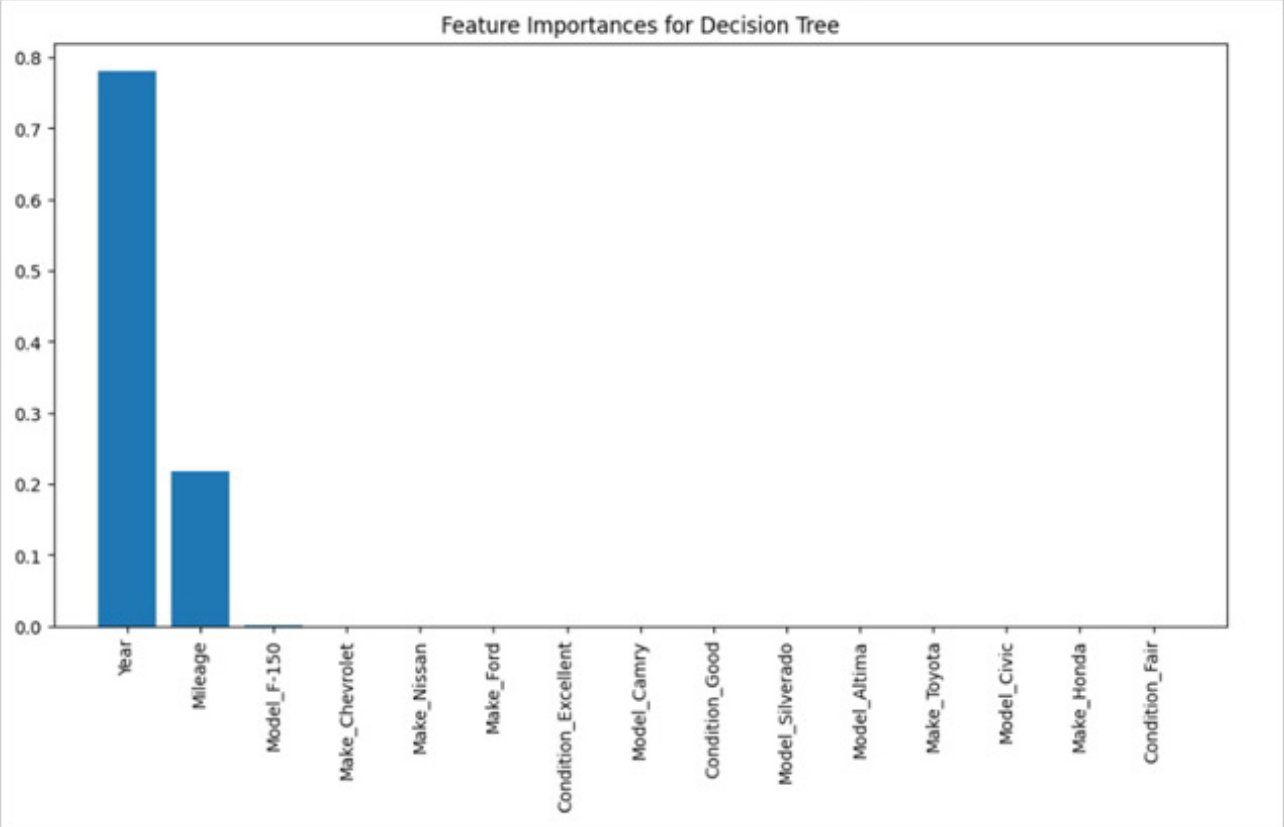


Рис. 4. Диаграмма важности признаков для модели «Дерево решений»

Случайный лес

Модель случайного леса — алгоритм машинного обучения, заключающийся в использовании ансамбля решающих деревьев. У данного метода точность R^2 составила 0.999337978715414 при оптимальном числе деревьев (n_estimators) равным 100 и максимальной глубине дерева (max_depth) равной 8.

На рис. 5 представлена диаграмма важности признаков для модели «Случайный лес», которая показывает, как параметры набора данных влияют на модель. Можно сделать вывод, что признаки «Год», «Пробег», влияют на модель, а значит остальные признаки можно удалить для более быстрого обучения модели.

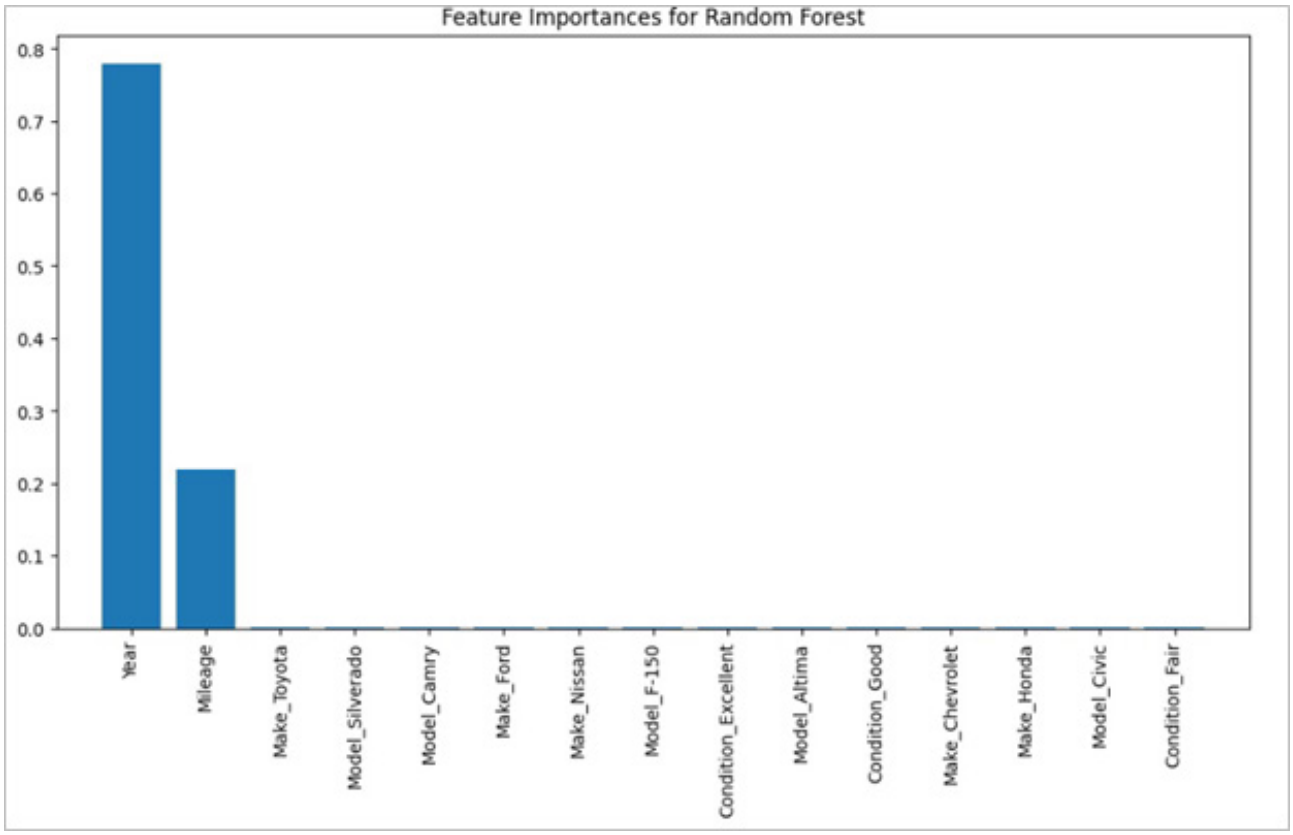


Рис. 5. Диаграмма важности признаков для модели «Случайный лес»

Градиентный бустинг

Градиентный бустинг — это техника машинного обучения для задач классификации и регрессии, которая строит модель предсказания в форме ансамбля слабых предсказывающих моделей, обычно деревьев решений. У данного метода точность R^2 составила 0.85 при оптимальном числе деревьев (n_estimators) равным 100 и максимальной глубине дерева (max_depth) равной 8.

На рис. 6 представлена диаграмма важности признаков для модели «Градиентный бустинг», которая показывает, как параметры набора данных влияют на модель. Можно сделать вывод, что признаки «Год», «Пробег», влияют на модель, а значит остальные признаки можно удалить для более быстрого обучения модели.

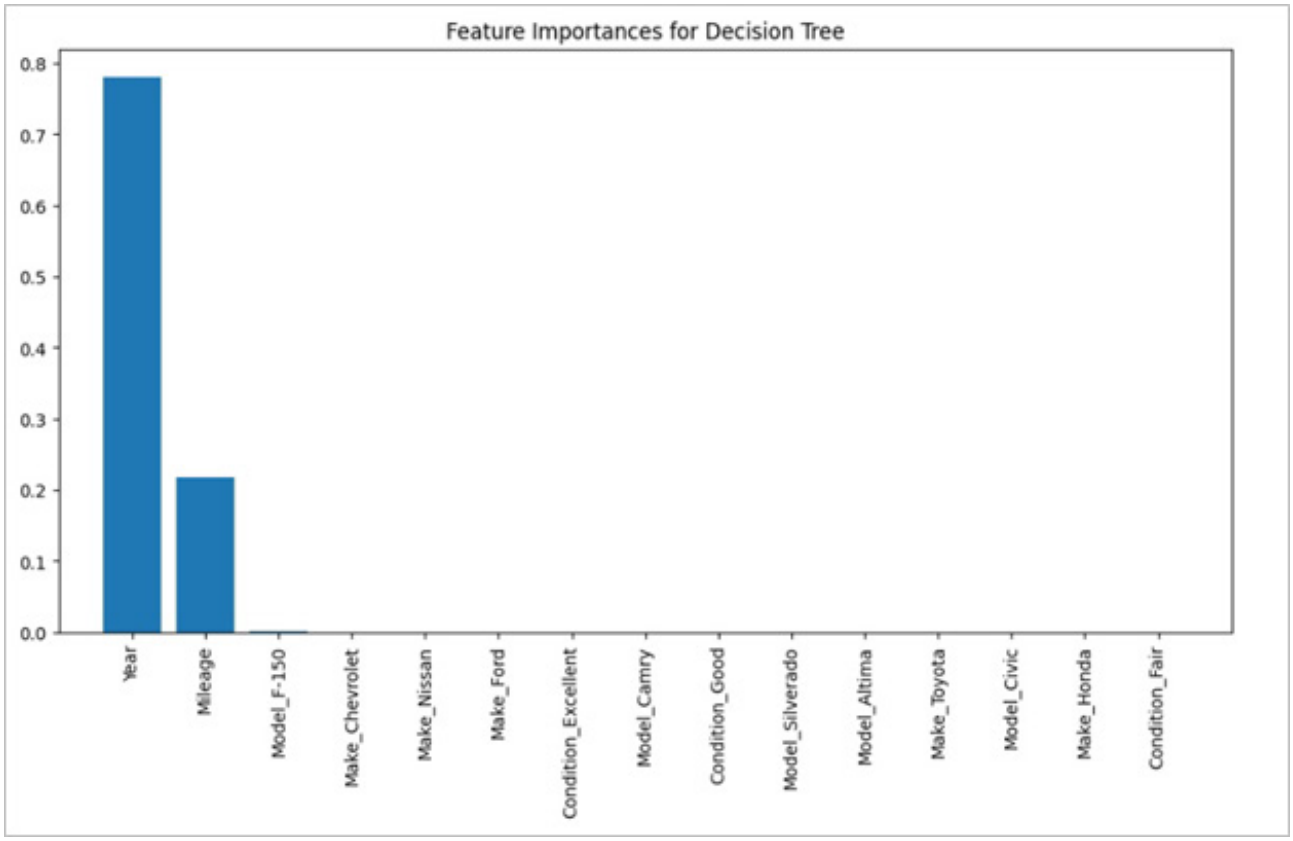


Рис. 6. Диаграмма важности признаков для модели «Градиентный бустинг»

Диаграмма рассеивания

Для оценки точности каждой из рассмотренных моделей помимо метрики R^2 была построена диаграмма рассеивания, которая показывает соотношение предсказанных и реальных значений. На рис. 7 видно, что больше совпадений в диапазоне цен от 22500 до 25000. Значительно хуже предсказываются цены на автомобили до 17500 и от 30000.

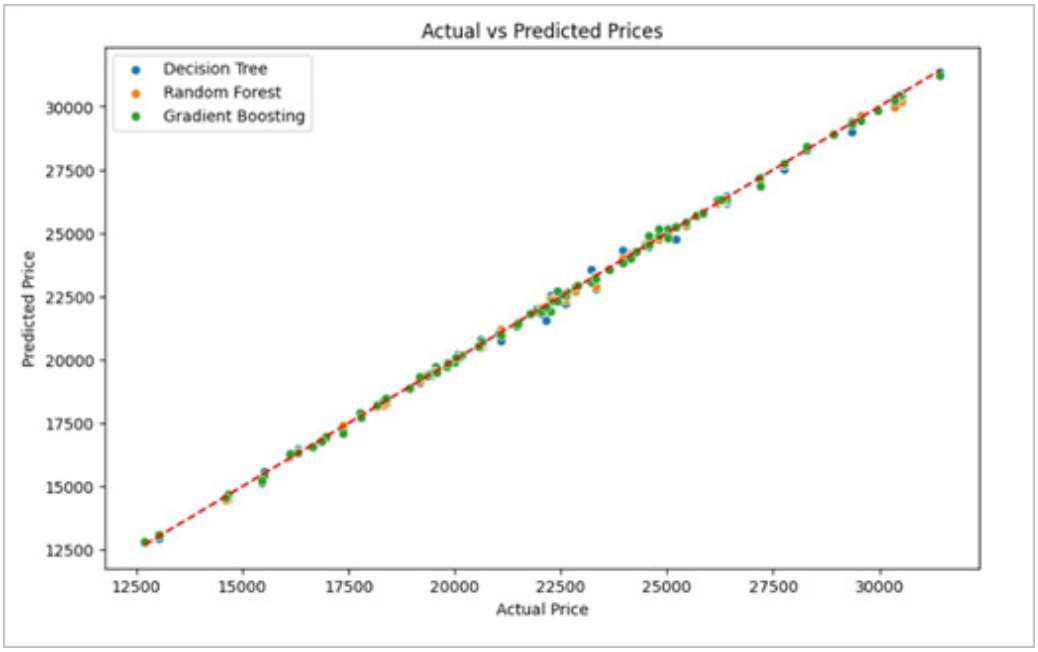


Рис. 7. Диаграмма рассеивания

Заключение

В результате сравнения наилучший коэффициент детерминации показала модель случайного леса. Среди рассмотренных моделей она лучше всего подходит для прогнозирования цен на автомобили. Главным фактором, влияющим на стоимость автомобиля, у трёх моделей является год и пробег.

Литература

1. Теория и практика машинного обучения : учеб. пособие / В. В. Воронина, А. В. Михеев, Н. Г. Ярушкина, К. В. Святов. – Ульяновск : УлГТУ, 2017. – 290 с.

2. Что влияет на стоимость автомобиля. – URL:https://avtocod.ru/chto-vliyaet-na-stoimost-avtomobilya (дата обращения 15.09.2024).

3. Метод случайного леса. – URL: https://ru.wikipedia.org/wiki/Метод_случайного_леса (дата обращения 22.09.2024).

4. Градиентный бустинг. – URL:https://ru.wikipedia.org/wiki/Бустинг (дата обращения 15.09.2024).

5. Модели классификации: дерево решений. – URL:https://ru.wikipedia.org/wiki/Дерево_решений (дата обращения 17.09.2024).

УДК 004.8

АВТОМАТИЗАЦИЯ ПРОГРАММИРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

А. А. Рябчиков

Воронежский государственный университет

Аннотация. Современные методы искусственного интеллекта (ИИ) находят всё более широкое применение в области программирования, начиная с автоматизации рутинных задач и заканчивая генерацией программного кода для сложных систем. В данной статье анализируются основные методы и подходы, используемые в автоматизации программирования, включая машинное обучение, глубокие нейронные сети и языковые модели. Исследуются текущие достижения в этой области, а также рассматриваются возможности и ограничения применения ИИ в создании кода и разработке программных систем.
Ключевые слова: автоматизация программирования, машинное обучение, глубокие нейронные сети, языковые модели, искусственный интеллект, генерация кода, оптимизация кода, обработка естественного языка, Python, программирование для ИИ, синтез программ.

Введение

В последние годы искусственный интеллект стремительно внедряется в разнообразные области науки и техники. В частности, он стал играть ключевую роль в разработке и оптимизации программного обеспечения, предлагая возможности автоматизации на уровнях, которые ранее казались невозможными. Методы ИИ позволяют создавать программы, анализировать код, оптимизировать алгоритмы и находить ошибки с минимальным участием человека. Применение ИИ в программировании не только сокращает время разработки, но и повышает качество и производительность программных продуктов.

Существует несколько подходов к автоматизации программирования с использованием ИИ, включая глубокие нейронные сети, методы машинного обучения и обработку естественного языка. Наиболее передовые языковые модели, такие как GPT и Codex, могут генерировать код на популярных языках программирования, таких как Python, JavaScript и C++, значительно сокращая объем рутинной работы программистов. В данной статье рассматриваются ключевые аспекты применения ИИ для создания и оптимизации программного обеспечения, его достижения и перспективы.

1. История автоматизации программирования с использованием ИИ

История автоматизации программирования берет свое начало в 1950-х годах, когда были разработаны первые алгоритмы, ставшие основой для машинного обучения. На этом этапе компьютерные программы выполняли лишь самые базовые операции, но уже тогда идеи автоматизации и использования ИИ привлекали большое внимание. С появлением первых языков программирования, таких как Fortran и Lisp, стало возможным писать более сложные программы, а разработка автоматизированных инструментов для программирования стала реальной перспективой.

В 1980-х годах появился важный шаг к применению ИИ в программировании: экспертные системы, которые могли решать задачи в узких областях. Эти системы были запрограммированы на выполнение определённых действий, используя заранее известные правила и базы знаний. Это сделало возможным автоматизацию некоторых задач, и экспертные системы

стали использоваться в таких областях, как медицина и инженерия. Однако их возможности были ограничены: они могли выполнять только те действия, которые были заранее запрограммированы, что ограничивало их гибкость и адаптивность.

В 2010-х годах произошел прорыв с развитием глубинного обучения и нейронных сетей. Благодаря им системы смогли анализировать и обрабатывать естественный язык, что стало важной вехой на пути к созданию моделей, способных генерировать и анализировать код. Программы, основанные на нейросетях, научились предсказывать и автоматизировать действия разработчиков, помогая им в написании и тестировании кода. Это стало возможным благодаря новым методам машинного обучения, которые позволили моделям обрабатывать большие объемы данных и лучше понимать структуру программных текстов.

С 2020-х годов крупные языковые модели, такие как GPT, начали активно использоваться для автоматизации программирования. Благодаря обучению на огромных массивах данных, эти модели могут генерировать код на различных языках и помогать разработчикам решать задачи без необходимости досконально изучать синтаксис и правила. Эти технологии делают ИИ мощным инструментом для программистов, ускоряя процессы разработки, а также обеспечивая возможности автоматического тестирования и улучшения кода. В итоге ИИ стал важным помощником, помогающим разработчикам сосредоточиться на более сложных задачах и повышающим качество программного обеспечения.

2. Архитектура и методы ИИ в автоматизации программирования

Современные технологии искусственного интеллекта, такие как глубокие нейронные сети и большие языковые модели, существенно продвинули автоматизацию программирования. Глубокие нейронные сети, обученные на огромных наборах данных, включающих как текст, так и программный код, используют многослойную архитектуру, благодаря которой они могут не только анализировать, но и генерировать код, учитывая его синтаксис и логику. Архитектура Transformer, лежащая в основе многих современных языковых моделей, позволяет этим системам эффективно анализировать структуру кода, предсказывать его дальнейшие части и автоматически исправлять ошибки. Это помогает разработчикам автоматизировать процессы дополнения, исправления и даже оптимизации программного кода.

Для создания кода с использованием ИИ применяются такие крупные модели, как GPT-3 и Codex, которые обучены на миллионах строк программного кода. Благодаря этому они способны «понимать» синтаксические и логические структуры множества языков программирования, включая как распространенные языки вроде Python или Java, так и специализированные. Такие модели позволяют автоматизировать множество типичных задач, которые раньше требовали большого объема ручной работы. Это значительно ускоряет процесс разработки, позволяет оптимизировать код, а также сократить время на выполнение рутинных операций.

3. Программные инструменты и платформы для автоматизации программирования

Существует несколько инструментов и платформ, созданных для поддержки ИИ в автоматизации программирования. Например, GitHub Copilot и Tabnine помогают программистам с автодополнением и генерацией кода, значительно сокращая время на написание типичных функций и блоков кода. Эти системы основаны на языковых моделях, которые могут предлагать контекстно-осмысленный код, соответствующий текущей задаче программиста.

Платформа TensorFlow и библиотека PyTorch также позволяют использовать ИИ для создания и тестирования моделей, которые автоматизируют части процесса разработки. Используя эти инструменты, разработчики могут обучать свои модели на примерах кода и автоматически генерировать решение для новых задач.

4. Примеры практического применения ИИ в программировании

Автоматизация программирования с использованием искусственного интеллекта охватывает всё больше сфер и находит применение в самых разных отраслях. Одним из ярких примеров использования ИИ является автоматическое тестирование, которое значительно упростилось благодаря языковым моделям, способным генерировать тестовые сценарии для проверки кода. Такие модели могут автоматически анализировать написанный код, выявлять потенциальные ошибки и создавать тесты, которые проверяют функциональность программного обеспечения на различных уровнях. Например, для веб- приложений языковые модели могут генерировать юнит-тесты для проверки отдельных функций или компонентов, а также интеграционные и системные тесты для проверки взаимодействия между ними. Это позволяет разработчикам более оперативно находить и исправлять ошибки, тем самым улучшая качество конечного продукта и сокращая временные затраты на ручное тестирование. В некоторых случаях автоматическое тестирование на базе ИИ способно предложить оптимальные тест-кейсы, которые разработчики могли бы упустить.

Другой значимой областью применения ИИ в программировании стала оптимизация кода. Сегодня ИИ помогает не только обнаруживать узкие места в производительности программного обеспечения, но и автоматически анализировать и оптимизировать существующий код для его ускорения. Например, языковые модели могут анализировать код и предлагать улучшенные версии, устраняя избыточные или избыточные вычисления, что позволяет оптимизировать выполнение программы. В крупных системах, где сложные алгоритмы требуют множества вычислительных ресурсов, такие улучшения могут быть критически важными и значительно сократить время выполнения кода и расход ресурсов. Также ИИ может помогать с оптимизацией памяти, предлагать более эффективные структуры данных и алгоритмы, что особенно актуально в разработке для мобильных устройств и встроенных систем, где вычислительные ресурсы часто ограничены.

Еще одним важным направлением использования ИИ стало создание генеративных инструментов для автоматического написания кода. Генерация кода для веб-приложений и мобильных приложений является популярной задачей, которую ИИ помогает решать, автоматически генерируя основные структуры и элементы приложения. Например, языковые модели могут сгенерировать шаблонный код для интерфейса веб-приложения на основе описания, которое предоставил разработчик. Системы на базе ИИ могут автоматически создавать компоненты HTML, CSS и JavaScript, которые являются основными блоками для построения веб-страниц. Это делает процесс разработки веб-приложений более быстрым и удобным, так как разработчики могут сосредоточиться на решении более сложных задач, доверяя ИИ генерацию базовой структуры.

Что касается мобильных приложений, ИИ также находит здесь широкое применение. С помощью языковых моделей разработчики могут генерировать код для прототипов мобильных приложений, выполняющих базовые функции. Такие прототипы, созданные с использованием ИИ, можно адаптировать для различных платформ, таких как Android и iOS, благодаря совместимости языковых моделей с популярными языками программирования и фреймворками, такими как Swift и Kotlin. Автоматическая генерация кода для интерфейса пользователя, навигации и базовой логики приложения позволяет быстро создавать работающие прототипы, которые можно тестировать и дорабатывать для использования в реальных проектах. Такой подход помогает значительно ускорить процесс создания новых продуктов, так как разработчики могут быстрее получать обратную связь о работоспособности приложения и оценивать его функциональность в действии.

5. Преимущества и ограничения использования ИИ в автоматизации программирования

Преимущества:

Сокращение времени на разработку: использование ИИ позволяет быстро генерировать код, особенно для стандартных задач.

Повышение качества кода: ИИ может анализировать код на наличие ошибок и предлагать оптимальные решения.

Автоматизация тестирования: ИИ способен создавать тесты, ускоряя процесс выявления багов.

Ограничения:

Ограниченная универсальность: несмотря на возможности ИИ, не все задачи можно эффективно автоматизировать.

Необходимость корректировки и проверки: код, сгенерированный ИИ, требует проверки человеком, особенно для сложных задач.

Трудности с обучением: создание и обучение моделей требует больших ресурсов и знаний в области ИИ.

Заключение

Использование методов искусственного интеллекта для автоматизации программирования открывает новые возможности в разработке программного обеспечения. ИИ может сократить время на разработку, повысить качество кода и автоматизировать многие рутинные процессы. Несмотря на некоторые ограничения, такие как необходимость верификации и высокие требования к ресурсам, будущее автоматизированного программирования с использованием ИИ выглядит многообещающим и перспективным.

Литература

1. *Smith A*. GPU Computing in Modern Physics // Journal of Computational Physics. – 2021.

2. NVIDIA. NVIDIA Tesla V100 Architecture // NVIDIA Whitepapers. – 2017.

3. Owens J.D. GPU Computing in Modern Research // High-Performance Computing Journal. – 2008.

4. NVIDIA. Технические характеристики архитектуры GPU Tesla V100 // NVIDIA Whitepapers. – 2017.

УДК 519.8

МОДЕЛИ ВОЗМОЖНОСТНО-ВЕРОЯТНОСТНОЙ ОПТИМИЗАЦИИ С ОГРАНИЧЕНИЯМИ ПО ВЕРОЯТНОСТИ И ВОЗМОЖНОСТИ ПРИ ИСПОЛЬЗОВАНИИ СЛАБЕЙШЕЙ ТРЕУГОЛЬНОЙ НОРМЫ

И. С. Солдатенко

Тверской государственный университет

Аннотация. В работе исследованы модели оптимизации в контексте гибридной неопределенности возможностно-вероятностного типа. Для агрегирования возможностной информации используется слабейшая треугольная норма. Рассмотрены два принципа принятия решений, отличающиеся порядком снятия неопределенности: наложение ограничений по возможности/необходимости — вероятности и по вероятности — возможности/необходимости. В первом случае для нормально распределенных случайных факторов модели построен эквивалентный аналог системы. Во втором случае получена эквивалентная модель стохастической оптимизации. Показано, как с помощью индикаторных функций можно построить эквивалентный стохастический аналог, для решения которого можно использовать методы стохастического квазиградиента.
Ключевые слова: возможностно-вероятностное программирование; эквивалентный детерминированный аналог; эквивалентный стохастический аналог; неопределенность гибридного типа; возможность; необходимость; агрегирование возможностной информации; слабейшая треугольная норма.

Введение

В настоящей статье продолжены исследования задач возможностно-вероятностного программирования, начатые в [1–4]. В работе [1] был построен детерминированный эквивалент задачи при ограничениях по возможности/необходимости — вероятности, а в работе [2] — эквивалентный стохастический аналог для случая, когда неопределенность снимается в другом порядке – сначала вероятностная, потом возможностная. В [3, 4] был изучен вопрос коммутативности порядка снятия неопределенности и показано, что данное свойство не выполняется. Во всех перечисленных выше работах для агрегирования возможностной информации использовалась сильнейшая треугольная норма. В настоящей статье все указанные модели изучаются для случая слабейшей треугольной нормы. Для этого строится исчисление возможностей на основе результатов Р. Фуллера и Х. Нгуена [10, 11], позволяющих вычислять α -уровневое множество функций от нечётких параметров с помощью функций от α-уровней самих параметров. В настоящей работе указанные результаты обобщаются на случай нечетких чисел и отрезков, имеющих, в общем случае, неограниченные носители.

1. Основные определения из теории возможностей

Введем ряд требуемых нам понятий и определений из теории возможностей [5–7]. Пусть $(\Gamma, P(\Gamma), \tau)$ и (Ω, B, P) есть, соответственно, возможностное и вероятностное пространства, в которых Ω — пространство элементарных событий $\omega \in \Omega$, Γ — модельное пространство с элементами $\gamma \in \Gamma$, B — σ-алгебра событий, $P(\Gamma)$ — множество всех подмножеств $\Gamma, \tau \in \{\pi, \upsilon\}$, π и υ — меры возможности и необходимости, а P — мера вероятности; $\mathbb{R}$ — числовая прямая.

Определение 1. Возможностной (нечеткой) величиной $X(\gamma)$ называется вещественная функция $X : \Gamma \to \mathbb{R}^1$, характеризующаяся функцией распределения возможностей $\mu_X(t) = \pi\{\gamma \in \Gamma : X(\gamma) = t\}$, $\forall t \in \mathbb{R}$.

Определение 2. Для нечеткой величины $X(\gamma)$ и любого $\alpha \in (0,1]$ α-уровневым множеством называется четкое подмножество $X_\alpha = \{x \in \mathbb{R} \mid \mu_X(x) \geq \alpha\}$.

Определение 3. Модальным значением нечеткой величины $X(\gamma)$ называется четкое подмножество $\{x \in \mathbb{R} \mid \mu_X(x) = 1\}$.

Определение 4. Нечеткая величина называется (строго) унимодальной, если ее модальное значение есть интервал (точка).

Определение 5. Функция распределения возможностей μ_X нечеткой величины $X(\gamma)$ называется квазивогнутой, если $\mu_X(\lambda x_1 + (1-\lambda)x_2) \geq \min\{\mu_X(x_1), \mu_X(x_2)\}$, $\lambda \in [0,1]$, $x_1, x_2 \in \mathbb{R}$.

Определение 6. Функция распределения возможностей μ_X называется полунепрерывной сверху, если $\forall \alpha \in (0,1]$ множество X_α является замкнутым.

Определение 7. Нечеткая величина $X(\gamma)$ называется нечетким числом (отрезком), если выполняются следующие условия: μ_X полунепрерывна сверху и квазивогнута, $X(\gamma)$ является строго (нестрого) унимодальной, $\forall \alpha \in (0,1]$ множество X_α ограничено.

Определение 8. Слабейшей треугольной нормой T_W называется отображение $T : [0,1] \times [0,1] \to [0,1]$ следующего вида $T_W = \begin{cases} \min\{x, y\}, & \textit{если } \max\{x, y\} = 1, \\ 0, & \textit{иначе.} \end{cases}$

Неопределенность гибридного возможностно-вероятностного типа будем моделировать с помощью нечетких случайных величин.

Определение 9. Нечеткая случайная величина $Y(\omega, \gamma)$ есть вещественная функция $Y : \Omega \times \Gamma \to \mathbb{R}^1$ B-измеримая для каждого γ, а $\mu_Y(\omega, t) = \pi\{\gamma \in \Gamma : Y(\omega, \gamma) = t\}$, $\forall t \in \mathbb{R}$ называется ее функцией распределения.

Определение 10. Пусть $Y(\omega, \gamma)$ — нечеткая случайная величина. Ее ожидаемым значением $\mathbf{E}[Y]$ называется нечеткая величина, имеющая функцию распределения возможностей $\mu_{\mathbf{E}[Y]}(t) = \pi\{\gamma \in \Gamma : \mathbf{E}[Y(\omega, \gamma)] = t\}$, $\forall t \in \mathbb{R}^1$, где $\mathbf{E}$ — оператор взятия математического ожидания $\mathbf{E}[Y(\omega, \gamma)] = \int_\Omega Y(\omega, \gamma)\mathbf{P}(d\omega)$.

На практике удобно представлять нечеткую случайную величину в сдвиг-масштабном представлении [5].

Определение 11. Нечеткая случайная величина $Y(\omega, \gamma)$ имеет сдвиг-масштабное представление, если $Y(\omega, \gamma) = a(\omega) + \sigma(\omega)X(\gamma)$, где $a(\omega)$, $\sigma(\omega)$ — случайные факторы распределения, задающие, соответственно, сдвиг и масштаб, а $X(\gamma)$ — нечеткая компонента распределения.

2. Элементы исчисления возможностей при слабейшей треугольной норме

Под исчислением возможностей понимается набор правил для идентификации возможностных распределений простейших операций и функций от нечетких и нечетких случайных величин. В данном разделе мы идентифицируем основные операции, которые потребуются в дальнейшем, для случая слабейшей t-нормы T_W.

Часто получается найти границы α-уровневых множеств результата в явном виде, но иногда удобно их вычислять с помощью границ α-уровневых множеств операндов. Приведем основные результаты, которые позволяют это сделать.

В [10, 11] сформулированы и доказаны условия, позволяющие вычислять α-уровневое множество функции от нечётких параметров с помощью функции от α-уровней самих параметров. В [5] данные результаты погружены в контекст теории возможностей.

Утверждение 1 (см. [5,10,11]). *Пусть $A(\gamma)$ и $B(\gamma)$ — возможностные величины с функциями распределения возможностей μ_A и μ_B, соответственно; T — произвольная треугольная норма, а $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ — функция от двух аргументов. Тогда необходимым и достаточным условием выполнения равенства*

$$\left[f(A,B)\right]_\alpha = \bigcup_{\xi,\eta:T(\xi,\eta)\geq\alpha} f\left(A_\xi, B_\xi\right),\ \alpha\in(0,1] \tag{1}$$

является то, что $\forall z\in\mathbb{R}\ \sup\limits_{(x,y):f(x,y)=z} T\left(\mu_A(x),\mu_B(y)\right)$ *достигается.*

Утверждение 2 (см. [5,10,11]). *Если функция* $f:\mathbb{R}\times\mathbb{R}\to\mathbb{R}$ *— непрерывная, треугольная норма* T *— полунепрерывная сверху, то равенство (1) выполняется для любых возможностных величин* $A(\gamma)$ *и* $B(\gamma)$*, обладающих полунепрерывными сверху функциями распределения возможностей и компактными носителями.*

Как видим, в этих результатах используется условие компактности носителей возможностных операндов $A(\gamma)$ и $B(\gamma)$, что делает невозможным, в частности, их применение для нормальных нечетких чисел. Обобщим их на случай произвольных нечетких чисел или отрезков, заданных в соответствии с Определением 7, убрав условие компактности. Может быть доказана следующая теорема.

Теорема 1. *Если функция* $f:\mathbb{R}\times\mathbb{R}\to\mathbb{R}$ *— непрерывная, треугольная норма* T *— полунепрерывная сверху, то равенство (1) выполняется для любых нечетких чисел (отрезков)* $A(\gamma)$ *и* $B(\gamma)$*, заданных в соответствии с Определением 7.*

Используем этот результат для идентификации границ α -уровневых множеств основных арифметических операций.

Следствие 1. *Если* $A(\gamma)$ *и* $B(\gamma)$ *— нечеткие числа (отрезки), а для агрегирования возможностной информации используется слабейшая треугольная норма, то* α*-уровневые множества основных арифметических операций вычисляются следующим образом:*

$$\left[X+Y\right]_\alpha = \left[\min\left\{X_1^- + Y_\alpha^-, X_\alpha^- + Y_1^-\right\}, \max\left\{X_1^+ + Y_\alpha^+, X_\alpha^+ + Y_1^+\right\}\right],$$

$$\left[X-Y\right]_\alpha = \left[\min\left\{X_1^- - Y_\alpha^+, X_\alpha^- - Y_1^+\right\}, \max\left\{X_1^+ - Y_\alpha^-, X_\alpha^+ - Y_1^-\right\}\right],$$

$$\left[X\times Y\right]_\alpha = \left[\min\left\{X_1^\pm \cdot Y_\alpha^\pm, X_\alpha^\pm \cdot Y_1^\pm\right\}, \max\left\{X_1^\pm \cdot Y_\alpha^\pm, X_\alpha^\pm \cdot Y_1^\pm\right\}\right],$$

где $X_*^\pm \cdot Y_*^\pm = \left\{X_*^- \cdot Y_*^-, X_*^- \cdot Y_*^+, X_*^+ \cdot Y_*^-, X_*^+ \cdot Y_*^+\right\}$.

Легко индукцией доказывается следующий результат.

Следствие 2. Пусть $X_1(\gamma)$, $X_2(\gamma)$, …, $X_n(\gamma)$ — нечеткие числа (отрезки), а для агрегирования возможностной информации используется слабейшая t-норма. Тогда

$$\left[\sum_{i=1}^n X_i(\gamma)\right]_\alpha^- = \min_{i=1,\ldots,n}\left\{\left[X_i(\gamma)\right]_\alpha^- + \sum_{\substack{j=1,\ldots,n\\ j\neq i}}\left[X_j(\gamma)\right]_1^-\right\}\ \left[\sum_{i=1}^n X_i(\gamma)\right]_\alpha^+ = \max_{i=1,\ldots,n}\left\{\left[X_i(\gamma)\right]_\alpha^+ + \sum_{\substack{j=1,\ldots,n\\ j\neq i}}\left[X_j(\gamma)\right]_1^+\right\}.$$

Наконец, мы можем получить еще один важный с практической точки зрения результат.

Следствие 3. *Если в условиях Теоремы 1 функция f будет монотонно неубывающей по нечетким аргументам, то для слабейшей t-нормы левую и правую границы* α*-уровневого множества* f_α *можно найти по формулам:*

$$f_\alpha^- = \min\left\{f\left(X_1^-, Y_\alpha^-\right), f\left(X_\alpha^-, Y_1^-\right)\right\},\ f_\alpha^+ = \max\left\{f\left(X_1^+, Y_\alpha^+\right), f\left(X_\alpha^+, Y_1^+\right)\right\}.$$

3. Модели возможностно-вероятностного программирования при ограничениях по возможности/необходимости — вероятности и вероятности-возможности/необходимости

В настоящей работе мы продолжим изучение следующих двух моделей возможностно-вероятностного линейного программирования. Первая — модель уровневой оптимизации при ограничениях по возможности/необходимости — вероятности:

$$k\to\min,$$

$$\tau\left\{\mathbf{P}\left\{f_0(x,\omega,\gamma)\le k\right\}\ge p_0\right\}\ge\alpha_0, \tag{2}$$

$$\begin{cases}\tau\left\{\mathbf{P}\left\{f_i(x,\omega,\gamma)\le 0\right\}\ge p_i\right\}\ge\alpha_i, i=\overline{1,m},\\ \mathbf{x}\in X.\end{cases} \tag{3}$$

а вторая — модель уровневой оптимизации при ограничениях по вероятности — возможности/необходимости:

$$k\to\min,$$

$$\mathbf{P}\left\{\tau\left\{f_0(x,\omega,\gamma)\le k\right\}\ge\alpha_0\right\}\ge p_0, \tag{4}$$

$$\begin{cases}\mathbf{P}\left\{\tau\left\{f_i(x,\omega,\gamma)\le 0\right\}\ge\alpha_i\right\}\ge p_i, i=\overline{1,m},\\ \mathbf{x}\in X.\end{cases} \tag{5}$$

В этих моделях $f_i(x,\omega,\gamma),\ i=\overline{0,m}$ — это нечеткие случайные функции, являющиеся отображениями $f_i(\cdot,\cdot,\cdot): X\times\Omega\times\Gamma\to\mathbb{R}^1,\ X\subset\mathbb{R}^n,\ \tau\in\{\pi,\nu\}$ — одна из нечетких мер (возможности или необходимости), $p_i,\alpha_i\in(0,1],\ i=\overline{0,m}$ — заданные уровни вероятности и возможности, k — дополнительная переменная.

Будем рассматривать следующие представления функций $f_i(x,\omega,\gamma)$:

$$f_i(x,\omega,\gamma)=\sum_{j=1}^{n}A_{ij}(\omega,\gamma)x_j-B_i(\omega,\gamma),\ i=\overline{0,m}, \tag{6}$$

где $B_0(\omega,\gamma)\equiv 0$. Нечеткие случайные коэффициенты $A_{ij}(\omega,\gamma)$ и $B_i(\omega,\gamma)$ имеют сдвиг-масштабное представление со следующими коэффициентами:

$$A_{ij}(\omega,\gamma)=a_{ij}(\omega)+\sigma_{ij}(\omega)Y_{ij}(\gamma),\qquad B_i(\omega,\gamma)=b_i(\omega)+\sigma_i(\omega)Y_i(\gamma).$$

В работах [3, 4] модели (2)–(3) и (4)–(5) рассматривались в случае, когда для агрегирования возможностной информации используется сильнейшая t-норма T_M. В следующем разделе мы рассмотрим эквивалентные четкие и стохастические аналоги указанных моделей для слабейшей треугольной нормы T_W.

4. Эквивалентные четкий и стохастический аналоги задач (2)–(3) и (4)–(5) при слабейшей треугольной норме

Начнем с эквивалентного четкого аналога задачи (2)–(3). Пусть в сдвиг-масштабном представлении нечетких случайных коэффициентов $A_{ij}(\omega,\gamma)$ и $B_i(\omega,\gamma)$ случайные параметры принадлежат классу нормальных вероятностных распределений: $a_{ij}(\omega)\in\mathrm{N}_p(a_{ij}^0,d_{ij}^a)$, $\sigma_{ij}(\omega)\in\mathrm{N}_p(\sigma_{ij}^0,d_{ij}^\sigma)$, $b_i(\omega)\in\mathrm{N}_p(b_i^0,d_i^b)$, $\sigma_i(\omega)\in\mathrm{N}_p(\sigma_i^0,d_i^\sigma)$. Для упрощения записи ковариацию случайных параметров обозначим следующим образом: $C_{\alpha\beta}=\mathrm{cov}(\alpha,\beta)$, а ковариационную матрицу как $\mathbb{C}^i=\{\mathbb{C}_{ijk}(t_{ik},t_{ij})\}_{j,k=1}^n,\ i=\overline{0,m}$ с коэффициентами $\mathbb{C}_{ijk}(t_{ik},t_{ij})=C_{a_{ij}a_{ik}}+C_{a_{ij}\sigma_{ik}}t_{ik}+C_{\sigma_{ij}a_{ik}}t_{ij}+$ $+C_{\sigma_{ij}\sigma_{ik}}t_{ij}t_{ik}$.

Формулы для расчета математического ожидания нечетких случайных функций (6), используемых для построения эквивалентных детерминированного аналога модели (2)–(3) можно найти в [5, 6].

В [1], опираясь на результаты [6, 8], доказано, что при сделанных предположениях о распределениях параметров модели, эквивалентный детерминированный аналог для i-го ограничения модели (3) содержит только неопределенность возможностного типа и имеет следующий вид:

$$\tau\left\{\sum_{j=1}^{n}\left(a_{ij}^0+\sigma_{ij}^0Y_{ij}(\gamma)\right)x_j-\beta_i\sqrt{\tilde{d}_i(x,\gamma)}\le b_i^0+\sigma_i^0Y_i^0(\gamma)\right\}\ge\alpha_i,$$

где β_i — решение уравнения $F_i(t) = 1 - p_i$, $F_i(t)$ — функция стандартного нормального распределения, а $\tilde{d}_i(x,\gamma) = \sum_{j=1}^{n}\left(d_{ij}^a + d_{ij}^\sigma Y_{ij}^2(\gamma)\right)x_j^2 + d_i^b + d_i^\sigma Y_i^2(\gamma)$. С помощью данного выражения можно доказать следующие теоремы, дающие эквивалентные четкие аналоги модели критерия (2) и модели ограничений (3).

Теорема 2. *Пусть в модели критерия (2) $\tau = \pi$, $p_0 > 0.5$, все нечеткие компоненты $Y_{0j}(\gamma)$ в сдвиг-масштабном представлении нечетких случайных величин $A_{0j}(\omega,\gamma)$ являются взаимно T_W-связанными нечеткими числами, все случайные параметры сдвига и масштаба являются нормально распределенными случайными величинами. Тогда эквивалентный детерминированный аналоги модели критерия (2) есть*

$$\left[\sum_{j=1}^{n}\left(a_{0j}^0 + \sigma_{0j}^0 Y_{0j}(\gamma)\right)x_j - \beta_0\sqrt{\sum_{j=1}^{n}\sum_{k=1}^{n}C_{0jk}(\gamma)x_j x_k}\right]_{\alpha_0}^{-} \to \min_{x\in X}.$$

Теорема 3. *Пусть в модели ограничений (3) $\tau = \pi$, $p_i > 0.5$, $i = \overline{1,m}$, все нечеткие компоненты $Y_{ij}(\gamma)$, $Y_i(\gamma)$ в сдвиг-масштабном представлении нечетких случайных величин $A_{ij}(\omega,\gamma)$ и $B_i(\omega,\gamma)$ являются взаимно T_W-связанными нечеткими числами, а коэффициенты сдвига и масштаба являются независимыми нормально распределенными случайными величинами. Тогда система ограничений (3) эквивалентна детерминированной модели ограничений*

$$\begin{cases}\left[\sum_{j=1}^{n}\left(a_{ij}^0 + \sigma_{ij}^0 Y_{ij}(\gamma)\right)x_j - \beta_i\sqrt{d_i(x,\gamma)} - b_i^0 - \sigma_i^0 Y_i(\gamma)\right]_{\alpha_i}^{-} \le 0, i = \overline{1,m},\\ \mathbf{x}\in X.\end{cases}$$

Для расчёта левых границ уровневых множеств в Теоремах 2 и 3 можно сначала построить вычислительный граф, в узлах которого будут находиться операции, а в листьях константы и нечёткие величины, а затем с помощью постфиксного обхода выполнить все вычисления на α-уровнях с помощью формул из Следствий 1–3. Приведем простой пример.

Пример 1. Пусть нам дана возможностная функция

$$f(X,Y,Z) = \left(a_1 + b_1 X(\gamma)\right) + \left(a_2 + b_2 Y(\gamma)\right) + \beta\sqrt{c + Z(\gamma)},$$

где $a_1 = 2.15$, $a_2 = 2$, $b_1 = -1$, $\beta = 0.5$, $c = 3$, $X = [2,3,3,4]$, $Y = [1,3,4,5]$, $Z = [-2,0,0,1]$. Нечеткие величины заданы в виде $[X_\alpha^-, X_1^-, X_1^+, X_\alpha^+]$, то есть с помощью своих модальных значений и α-уровневых множеств, вычисленных для какого-то конкретного значения α. Для агрегирования возможностной информации мы используем T_W. Вычислительный граф изображен на рис. 1. Для расчёта модальных значений и границ уровневых множеств используются правила работы с функциями от возможностных величин и результаты Следствий 1–3. В итоге мы получим следующие характеристики для функции: $f_\alpha = [3,6]$, $f_1 = [4,5]$.

С помощью этого же графа можно вычислять значения обобщенного градиента, если это необходимо для численного метода решения.

Рассмотрим теперь модель (4)–(5). В ней мы не требуем, чтобы случайные факторы были распределены по нормальному закону.

Теорема 4. *Пусть в модели ограничений (5) $\tau \in \{\pi,\nu\}$, для агрегирования возможностной информации используется слабейшая t-норма T_W, возможностные составляющие в сдвиг-масштабном представлении нечетких случайных величин $A_{ij}(\omega,\gamma)$ и $B_i(\omega,\gamma)$ являются симметричными нечеткими числами, задаваемыми с помощью (L,R)-распределений: $X_{ij}(\gamma) = (m_{ij}, d_{ij})_{S_{ii}}$, $X_i(\gamma) = (m_i, d_i)_{S_i}$ и имеющими в случае $\tau = \nu$ ограниченные носители. Тогда модель построчных ограничений по вероятности и возможности/необходимости (5) имеет эквивалентный стохастический аналог для меры возможности π:*

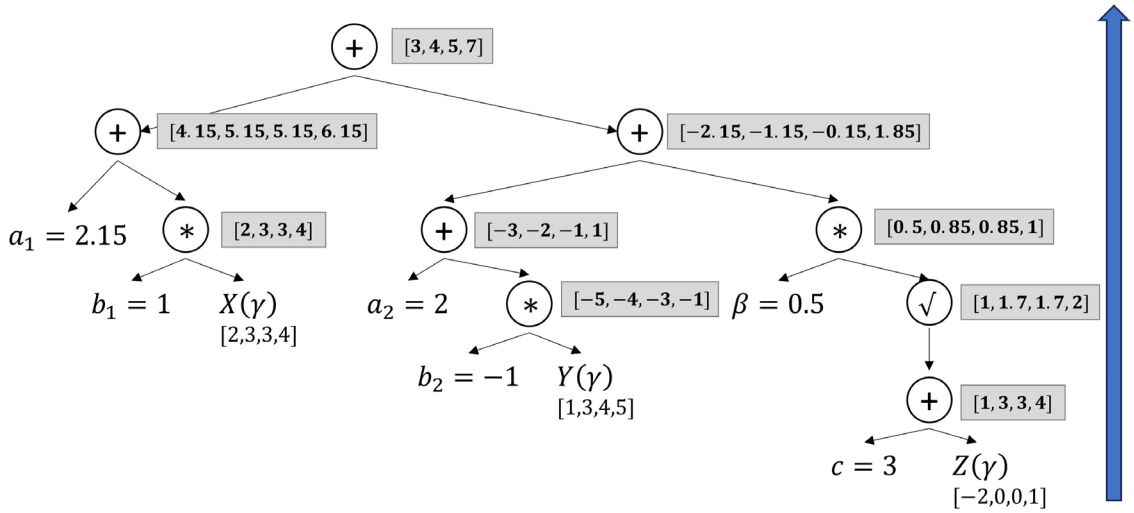


Рис. 1. Расчёт α-уровневого множества монотонной функции при слабейшей t-норме с помощью вычислительного графа

$$\begin{cases} \mathbf{P}\left\{\min\left\{A_i^-(x,\omega,\alpha_i) - B_i^+(\omega,1), A_i^-(x,\omega,1) - B_i^+(\omega,\alpha_i)\right\} \le 0\right\} \ge p_i, i = \overline{1,m}, \\ \mathbf{x} \in X, \end{cases} \tag{7}$$

и следующий для меры необходимости ν:

$$\begin{cases} \mathbf{P}\left\{\min\left\{A_i^+(x,\omega,1-\alpha_i) - B_i^-(\omega,0), A_i^+(x,\omega,0) - B_i^-(\omega,1-\alpha_i)\right\} \le 0\right\} \ge p_i, i = \overline{1,m}, \\ \mathbf{x} \in X, \end{cases} \tag{8}$$

в которых

$$A_i^{\pm}(x,\omega,\beta) = \sum_{j=1}^{n}\left(a_{ij}(\omega) + \sigma_{ij}(\omega)m_{ij}\right)x_j \pm S_{ii}^{-1}(\beta)\max_{j=1,\ldots,n}\left\{d_{ij}\left|\sigma_{ij}(\omega)x_j\right|\right\},$$

$$B_i^{\pm}(\omega,\beta) = a_i(\omega) + \sigma_i(\omega)m_i \pm S_i^{-1}(\beta)\left|\sigma_i(\omega)\right|d_i.$$

В (7)–(8) для $\beta = 1$ и $\beta = 0$ мы получаем, соответственно, формулы для левых/правых модальных значений и границ носителей.

Теорема 5. *Пусть в модели критерия (4) $\tau \in \{\pi, \nu\}$, для агрегирования возможностной информации используется слабейшая t-норма T_W, возможностные составляющие в сдвиг-масштабном представлении нечетких случайных величин $A_{0j}(\omega,\gamma)$ являются симметричными нечеткими числами, задаваемыми с помощью (L,R)-распределений: $X_{0j}(\gamma) = (m_{0j}, d_{0j})_{S_{00}}$. Тогда модель критерия (4) имеет следующий эквивалентный стохастический аналог для меры возможности π:*

$$k \to \min,$$
$$\mathbf{P}\left\{A_0^-(x,\omega,\alpha_0) \le k\right\} \ge p_0,$$

и следующий для меры необходимости ν:

$$k \to \min,$$
$$\mathbf{P}\left\{A_0^+(x,\omega,1-\alpha_0) \le k\right\} \ge p_0,$$

в которых

$$A_0^{\pm}(x,\omega,\beta) = \sum_{j=1}^{n}\left(a_{0j}(\omega) + \sigma_{0j}(\omega)m_{0j}\right)x_j \pm S_{00}^{-1}(\beta)\max_{j=1,\ldots,n}\left\{d_{0j}\left|\sigma_{0j}(\omega)x_j\right|\right\}.$$

Следствие 4. *Пусть в модели критерия (4) $\tau \in \{\pi, \nu\}$, для агрегирования возможностной информации используется слабейшая t-норма T_W, возможностные составляющие в сдвиг-мас-*

штабном представлении нечетких случайных величин $A_{0j}(\omega,\gamma)$ *являются нечеткими числами или отрезками, задаваемыми с помощью (L,R)-распределений:* $X_{0j}(\gamma)=\left(\underline{m}_{0j},\overline{m}_{0j},\underline{d}_{0j},\overline{d}_{0j}\right)_{L_{00}R_{00}}$ *и имеющими в случае* $\tau\in\nu$ *ограниченные носители, случайные коэффициенты масштаба* $\sigma_{0j}(\omega)\in\mathbb{R}_+,\ \forall\omega\in\Omega,\forall j,$ *а компоненты вектора неизвестных принимают только положительные значения:* $x\in X\bigcap\mathbb{R}_+$. *Тогда модель критерия (4) имеет следующий эквивалентный стохастический аналог для меры возможности* π*:*

$$k\to\min,$$
$$\mathbf{P}\left\{A_0^-(x,\omega,\alpha_0)\le k\right\}\ge p_0,$$

и следующий для меры необходимости ν*:*

$$k\to\min,$$
$$\mathbf{P}\left\{A_0^+(x,\omega,1-\alpha_0)\le k\right\}\ge p_0,$$

в которых

$$A_0^-(x,\omega,\beta)=\sum_{j=1}^{n}\left(a_{0j}(\omega)+\sigma_{0j}(\omega)\underline{m}_{0j}\right)x_j-L_{00}^{-1}(\beta)\max_{j=1,\dots,n}\left\{\underline{d}_{0j}\sigma_{0j}(\omega)x_j\right\},$$

$$A_0^+(x,\omega,\beta)=\sum_{j=1}^{n}\left(a_{0j}(\omega)+\sigma_{0j}(\omega)\overline{m}_{0j}\right)x_j-R_{00}^{-1}(\beta)\max_{j=1,\dots,n}\left\{\overline{d}_{0j}\sigma_{0j}(\omega)x_j\right\}.$$

Полученные модели, которые в общем виде можно записать как

$$\begin{cases}\mathbf{P}\left\{f(x,\omega)\le 0\right\}\ge p,\\ \mathbf{x}\in X\ \textit{или}\ \mathbf{x}\in X\bigcap\mathbb{R}_+^n,\end{cases}$$

представляют собой построчные ограничения по вероятности. Построение их эквивалентного детерминированного аналога, то есть снятие неопределенности вероятностного типа, затруднено даже для класса нормальных случайных величин в связи с присутствием в соответствующих выражениях модулей случайных величин или условий на их неотрицательность. Однако с помощью индикаторных функций

$$\mathrm{I}(x,\omega)=\begin{cases}1, & f(x,\omega)\le 0,\\ 0, & f(x,\omega)>0,\end{cases}$$

и свойства $\mathbf{E}[\mathrm{I}(x,\omega)]=\mathbf{P}\{f(x,\omega)\le 0\}$ их можно свести к системе ограничений

$$\begin{cases}\mathbf{E}\left[\mathrm{I}(x,\omega)\right]\ge p,\\ \mathbf{x}\in X\ \textit{или}\ \mathbf{x}\in X\bigcap\mathbb{R}_+^n,\end{cases}$$

и применить прямые методы стохастического программирования, основанные на методе стохастических квазиградиентов [9], которые не требуют построения эквивалентных детерминированных аналогов.

Заключение

В работе рассмотрена проблема возможностно-вероятностного линейного программирования (задача уровневой оптимизации) с ограничениями по возможности/ необходимости — вероятности и вероятности — возможности/необходимости в случае использования слабейшей треугольной нормы для агрегирования возможностной информации. Построены элементы исчисления возможностей, необходимые для идентификации границ уровневых множеств простейших операций над нечеткими величинами. Для этого использованы результаты Р. Фуллера и Х. Нгуена [10, 11], позволяющие вычислять α-уровневые множества функций от нечётких параметров с помощью функций от α-уровней самих параметров. Указанные результаты обобщены на случай нечетких чисел и отрезков, имеющих, в общем случае, неограниченные носители.

В плане перспективных исследований представляется интересным спецификация методов стохастического квазиградиента для решения стохастических аналогов задач.

Литература

1. *Yazenin A.* On the Problem of Possibilistic-Probabilistic Optimization with Constraints on Possibility/Probability / A. Yazenin, I. Soldatenko // Lecture Notes in Computer Science, WILF 2018. Giove S., Masulli F., Fuller R. (eds.). Advances in Intelligent Systems and Computing. – Vol. 11291. – Springer, Cham, 2019. – P. 43–54.

2. *Язенин А. В.* Задача возможностно-вероятностной оптимизации с ограничениями по возможности/необходимости — вероятности и вероятности — возможности/необходимости / А. В. Язенин, И. С. Солдатенко // Интегрированные модели и мягкие вычисления в искусственном интеллекте (ИММВ-2021). Сборник научных трудов X-й Международной научно-технической конференции. – Смоленск, 2021. – С. 271–283.

3. *Солдатенко И. С.* Об очередности принципов снятия неопределенности в задачах возможностно-вероятностного программирования и эволюционном методе их решения / И. С. Солдатенко, А. В. Язенин // Актуальные проблемы прикладной математики, информатики и механики (АППМ-2021). Сборник трудов Международной научной конференции. – Воронеж, 2022. – С. 748–754.

4. *Soldatenko I.* On the Order of Removing of Uncertainty Principles in the Problems of Possibilistic-Probabilistic Programming and the Evolutionary Method of their Solution / I. Soldatenko, A. Yazenin // 2023 Applied Mathematics, Computational Science and Mechanics: Current Problems (AMCSM). – Voronezh, 2023. – P. 1–5.

5. *Язенин А. В.* Основные понятия теории возможностей. – М. : Физматлит, 2016. – 142 с.

6. *Yazenin A.* Possibilistic Optimization / A. Yazenin, M. Wagenknecht. – Cottbus: Brandenburgische Technische Universitat, 1996.

7. *Nguyen H. T.* A First Course in Fuzzy Logic / H. T. Nguyen, E. A. Walker. – CRC Press, Boca Raton, 1997.

8. *Yazenin A. V.* On the problem of possibilistic optimization // Fuzzy Sets and Systems. – 1996. – Vol. 81(1). – P. 133–140.

9. *Ермольев Ю. М.* Методы стохастического программирования. – М. : Наука, 1976.

10. *Fuller R.* On generalization of Nguyen's theorem / R. Fuller, T. Keresztfalvi // Fuzzy Sets and Systems. – 1991. – Vol. 41. – P. 371–374.

11. *Nguen H. T.* A note on the extension principle for fuzzy sets // Journal of Mathematical Analysis and Applications. – 1978. – Vol. 64. – P. 369–380.

УДК 004.021

ИССЛЕДОВАНИЕ И РАЗРАБОТКА МЕТОДОВ ПОВЫШЕНИЯ ТОЧНОСТИ СЕМАНТИЧЕСКОЙ СЕГМЕНТАЦИИ НА ОСНОВЕ КАЛИБРОВКИ ВЕРОЯТНОСТНЫХ ОЦЕНОК ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ

А. М. Субботин, И. Л. Каширина

Воронежский государственный университет

Аннотация. В работе представлен метод повышения точности семантической сегментации патологических структур на основе анализа вероятностных карт, созданных глубокими нейронными сетями. Предложенный подход включает уточнение области анализа, фильтрацию шумовых объектов, исключение вложенных патологий и классификацию сегментированных областей.
Ключевые слова: семантическая сегментация, глубокие нейронные сети, вероятностные карты, медицинская диагностика, патологические структуры, фильтрация шумов.

Введение

Семантическая сегментация — это одна из ключевых задач машинного обучения, направленная на присвоение меток каждому пикселю изображения. Она широко применяется в медицине, автономном вождении, анализе спутниковых данных и других областях. В медицинской диагностике, например, сегментация опухолей и патологий в трехмерных изображениях, таких как МРТ и КТ, играет критически важную роль для своевременного выявления и лечения заболеваний.

Современные глубокие нейронные сети, такие как U-Net и её модификации, достигли значительных успехов в решении задачи сегментации. Однако, несмотря на их высокую точность, вероятностные оценки, генерируемые такими моделями, часто оказываются плохо откалиброванными. Это означает, что вероятность, которую модель присваивает предсказанию, не всегда соответствует истинной вероятности того, что предсказание верно. Некачественная калибровка может привести к снижению точности при интеграции модели в реальных приложениях, где важно учитывать не только предсказания, но и их достоверность.

Целью данной работы является исследование и разработка методов повышения точности семантической сегментации через улучшение вероятностных оценок, генерируемых глубокими нейронными сетями. Для достижения этой цели в работе предлагается подход, адаптированный для задач семантической сегментации, и выполняется экспериментальная проверка подхода на реальных данных.

Работа направлена на преодоление текущих ограничений моделей сегментации и улучшение их точности за счет калибровки вероятностных оценок, что особенно важно для задач с высокой степенью ответственности, таких как медицинская диагностика.

1. Постановка задачи и этапы решения

В рамках данной работы предполагается разработка метода, который улучшает точность сегментации патологических областей медицинских изображений на основе вероятностных оценок, полученных с использованием глубокой нейронной сети. Метод должен учитывать вложенные структуры, отсеивать незначимые объекты и улучшать интерпретацию вероятностных оценок.

Процесс решения задачи по разработке метода и его внедрению состоит из следующих этапов:

1. Уточнение области анализа с использованием маски органа
2. Реализация метода выделения и фильтрации объектов на основе вероятностных карт
3. Исключение шумовых объектов, объем которых ниже заданного порога
4. Проверка корректности выделения патологий по реальным клиническим данным

2. Алгоритм постобработки выходных вероятностей

Для разработки предложенного алгоритма был выбран язык программирования Python благодаря его широким возможностям для научных расчетов и обработки данных. В качестве ключевых библиотек для реализации алгоритма использовались NumPy [1] и SciPy [2], обеспечивающие высокую производительность вычислений и удобные инструменты для работы с массивами данных и числовым анализом. Для визуализации результатов применялась библиотека Plotly [3], которая позволила наглядно отображать трехмерные вероятностные карты и результаты классификации в интерактивном формате, что облегчило анализ работы алгоритма и оценку его эффективности.

Для проведения исследования и проверки разработанного алгоритма были использованы данные из открытого набора Medical Segmentation Decathlon (MSD) [4], предоставленного для задач анализа медицинских изображений. Этот набор включает трехмерные томографические изображения различных органов и патологий в формате NIfTI, что позволяет эффективно работать с объемными данными. Набор MSD широко применяется в научных исследованиях благодаря своему качеству и разнообразию предоставляемых данных, что делает его идеальной платформой для разработки и тестирования алгоритмов сегментации.

Вероятностные карты представляют собой многоканальные массивы, где каждый канал соответствует определённым типам патологий. Эти карты являются выходом нейронной сети, обученной на задаче семантической сегментации [5, 6]. Значения в каждом вокселе таких карт указывают вероятность принадлежности данного вокселя к конкретному классу. Например, если воксель имеет значение 0,85 в канале «Киста», это означает, что модель с вероятностью 85% относит его к этому типу патологии.

Первый канал вероятностной карты традиционно соответствует фону — областям, которые модель не относит ни к одному из классов, оставшиеся каналы соответствуют различным типам патологий. Такое разбиение позволяет детализировать результаты сегментации и использовать вероятностные значения для дальнейшей классификации и анализа. Вероятностные карты являются важным инструментом для точного определения границ объектов, поскольку они предоставляют не бинарную, а вероятностную интерпретацию предсказаний, что особенно полезно при обработке сложных медицинских данных, где границы между классами могут быть нечеткими.

Входными данными алгоритма являются вероятностные карты патологий, полученные путем обработки трехмерных медицинских изображений базовой моделью сегментации. Вероятностные карты имеют размерность (C, H, W, D), где C – количество классов сегментации, H, W, D — пространственные размеры изображения.

На первом этапе работы уточняется область, в которой будет производиться анализ вероятностных карт. Это достигается с использованием предварительно подготовленной маски органа. Маска ограничивает область обработки изображений только теми участками, которые относятся к органу интереса, исключая области, не содержащие диагностически значимой информации и рассчитывается по формуле

$$A = \min\left(background_{i,j,k}, liver_{i,j,k}\right),$$

где i, j, k — координаты вокселей, $background$ — карта вокселей фона, $liver$ — маска органа.

На следующем этапе по значениям вокселей полученной фоновой карты создаётся бинарная маска B, основанная на пороговом значении вероятности ($threshold$), которая позволяет выделить области, потенциально относящиеся к патологиям. Маска рассчитывается по формуле

$$B_{i,j,k} = \begin{cases} 1, & если \left(1 - A_{i,j,k}\right) > threshold \\ 0, & иначе. \end{cases}$$

Для выделения отдельных объектов применяется метод связных компонентов label, реализованный в библиотеке scipy, который идентифицирует отдельно стоящие объекты маски B и присваивает каждому вокселю значение индекса идентифицированной патологии. Для каждого выделенного объекта рассчитывается его объём в вокселях. Небольшие по объему объекты зачастую являются артефактами модели или шумом и несут мало диагностической значимости. Поэтому объем объекта сравнивается с заданным минимальным порогом min_volume. Если объём меньше порогового значения, объект удаляется из маски.

Процесс определения класса патологии базируется на анализе вероятностных карт, созданных моделью нейронной сети. Каждый выделенный объект исследуется по всем каналам вероятностных карт, чтобы определить, к какому типу патологии он принадлежит. Это делается с помощью расчёта метрик вероятности для каждого канала, таких как сумма вероятностей, средняя вероятность или медианная вероятность. В данном исследовании было принято решение использовать среднюю вероятность как основной критерий, поскольку она наиболее точно отражает общую вероятность принадлежности объекта к определённому классу, сглаживая влияние локальных аномалий и обеспечивая более стабильные результаты классификации. Средняя вероятность принадлежности патологии к выбранному типу, рассчитанная как взвешенное среднее вероятностей по всем вокселям

$$Avg_{type} = \frac{\sum_{v \in V} p\left(v, type\right)}{\left|V\right|},$$

где $type$ — вид патологии, $p(v, type)$ — вероятность конкретной патологии в вокселе v, V — множество вокселей найденной патологии.

Вложенные или пересекающиеся области, которые могут возникать из-за ошибок сегментации или особенности данных, представляют собой серьёзную проблему. Если обнаруживается вложенный объект, его средняя вероятность сравнивается с вероятностью объекта. Объект с меньшей средней вероятностью поглощается как менее значимый. После удаления вложенных объектов итоговая маска обновляется, чтобы исключить пересечения и дублирования. Эта проверка основана на принципе, что более крупные области, как правило, представляют основную патологию, тогда как вложенные области могут быть результатом шумов или недостаточной точности модели.

Основным выходным продуктом метода является многоканальная маска, где каждый канал соответствует определённому классу патологии. Эта маска содержит информацию о пространственном расположении объектов, причём:

1. Каждый воксель маски имеет значение, указывающее принадлежность к определённому классу или к фону.

2. Вложенные и шумовые объекты, которые были удалены в ходе фильтрации, не включены в итоговую маску, что улучшает её интерпретируемость.

3. Анализ полученных результатов

Результаты работы алгоритма представлены в виде объемной визуализации, которая наглядно иллюстрирует процесс выделения и анализа патологических областей в исследуемой области органа.

На рис. 1 можно видеть, как происходит определение патологий без применения алгоритма. Цветовая палитра иллюстрирует принадлежность выделенных зон к определенным классам патологий, где каждый цвет означает конкретный вид патологии. Желтая область занимает центральное место и имеет четкие границы. Внутри нее выделяются другие области. В то же время были идентифицированы патологии, имеющие малый объем.

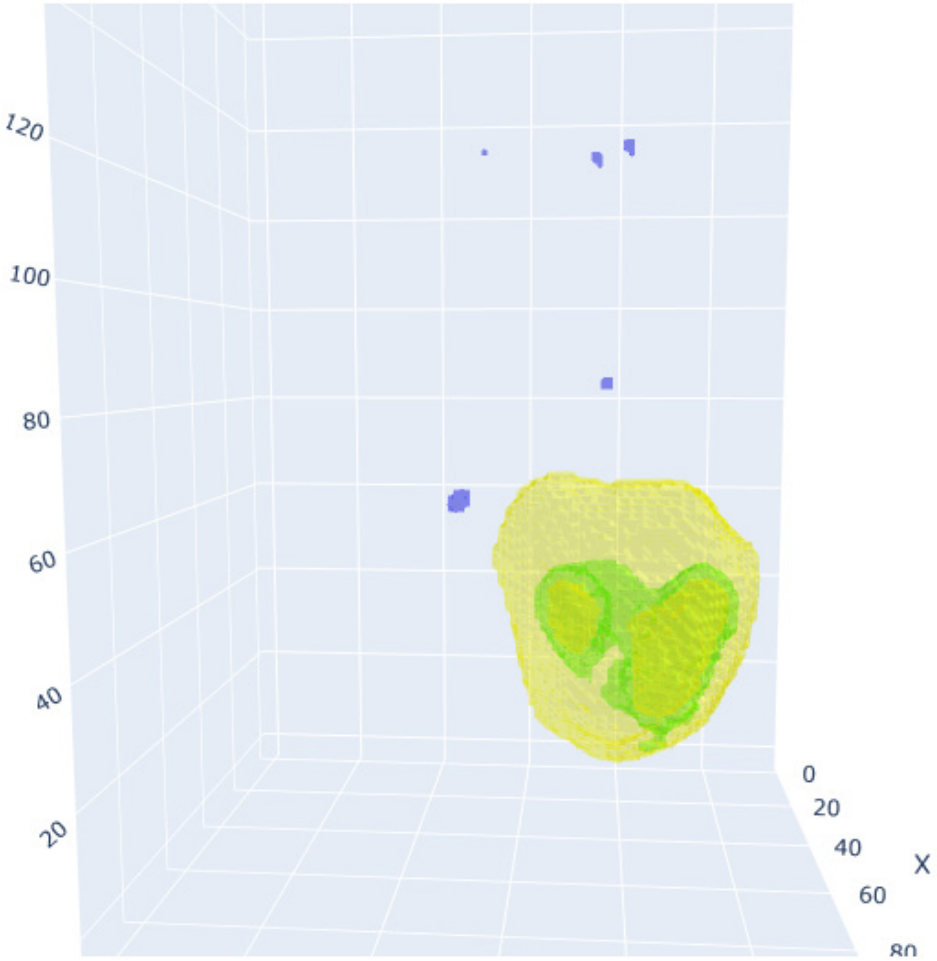


Рис. 1. Визуализация патологий до применения алгоритма

На рис. 2 представлен итоговый результат, который демонстрирует работу алгоритма. Границы желтой области остались неизменными, что говорит о сохранении базовой структуры патологии. Зеленые зоны корректно интегрированы в контекст основной области, что предотвращает их ошибочное выделение в качестве отдельных патологий. Патологии, имеющие малый объем были устранены, что подтверждает успешное исключение мелких шумовых объектов, не соответствующих заданным критериям значимости.

В табл. 1 представлены средние вероятности до (Avg) и после применения алгоритма (Avg'), а также суммарный объем в вокселях (V) для каждой из патологий при заданном пороге $threshold = 0,2$. Исходя из значений, можно увидеть, что алгоритм улучшает значения показателей средних вероятностей. В большинстве случаев значения после его применения превышают начальные. Обнуление или уменьшение значений является признаком того, что найденные патологии были незначительны по объему или были найдены вложенные патологии.

Таким образом, результаты показывают, что метод не только точно выделяет основные патологические образования, но и эффективно классифицирует их на основании вероятностных карт.

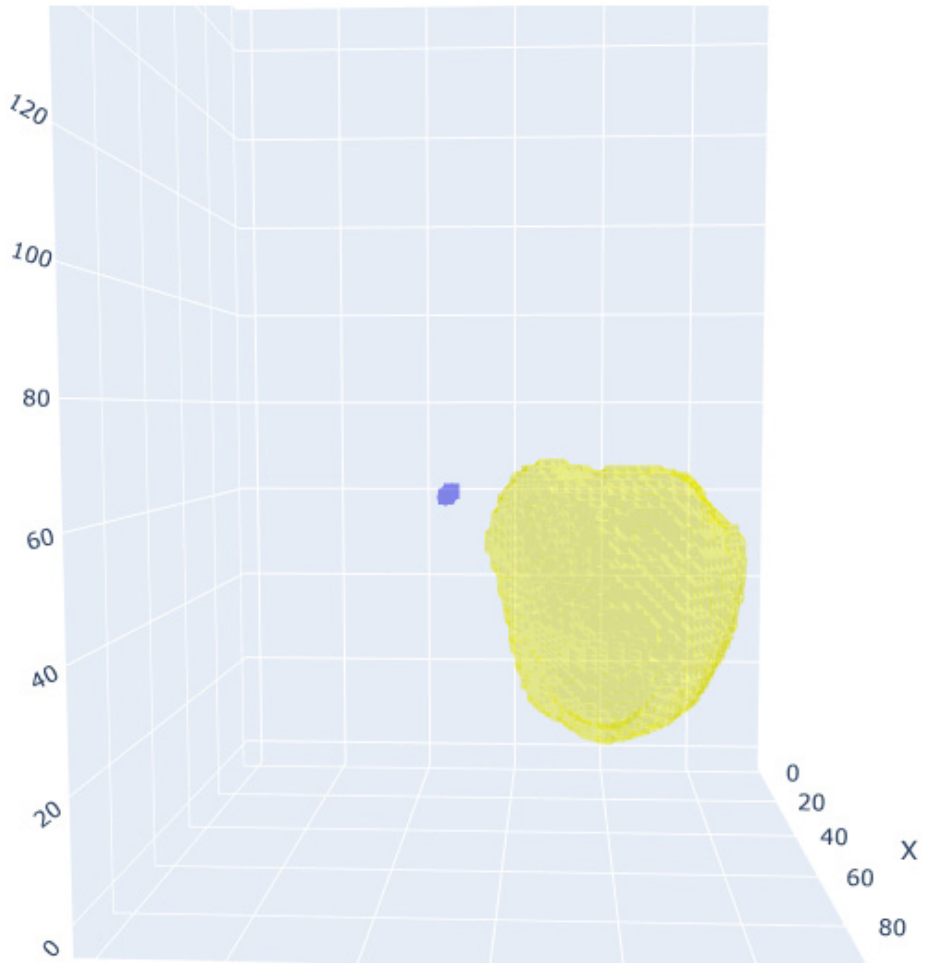


Рис. 2. Визуализация после применения алгоритма

Таблица 1

Средние вероятности патологий до и после применения алгоритма

Avg_1	Avg'_1	V_1	Avg_2	Avg'_2	V_2	Avg_3	Avg'_3	V_3	Avg_4	Avg'_4	V_4
0,66	0,79	189	0,68	0,00	0	0,00	0,00	0	0,85	0,87	303610
0,00	0,00	0	0,73	0,00	0	0,81	0,81	12346	0,84	0,86	1031
0,00	0,00	0	0,67	0,72	3383	0,27	0,38	364	0,75	0,71	181388
0,00	0,00	0	0,25	0,00	0	0,22	0,00	0	0,88	0,88	71578
0,65	0,73	226	0,82	0,80	78933	0,40	0,40	3423	0,53	0,52	827

Заключение

В ходе работы был разработан и реализован метод повышения точности семантической сегментации патологических структур с использованием вероятностных карт, сгенерированных глубокими нейронными сетями. Предложенный подход успешно решает ключевые задачи, возникающие при анализе медицинских изображений, включая исключение вложенных патологий, фильтрацию шумовых объектов с малым объемом и корректное определение классов патологий на основе вероятностных оценок.

Разработанный подход может быть интегрирован в системы автоматизированной диагностики, что сократит время анализа данных и повысит точность сегментации патологий.

Литература

1. NumPy Documentation // NumPy URL: https://numpy.org/doc/ (дата обращения: 15.10.2024).

2. SciPy API // SciPy v1.14.1 Manual URL: https://docs.scipy.org/doc/scipy/reference/index.html#scipy-api (дата обращения: 22.10.2024).

3. Python API reference for plotly // Plotly documentation URL: https://plotly.com/python-api-reference/ (дата обращения: 05.11.2024).

4. *Antonelli M., Reinke A.* The Medical Segmentation Decathlon / M. Antonelli // Nature communications. – 2022 – № 13. – С. 4128

5. Лукашик Д. В. Анализ современных методов сегментации изображений [Текст] / Д. В. Лукашик // Экономика и качество систем связи. – 2022. – № 24. – С. 57–65.

6. *Al-Shayea Q. K.* Artificial Neural Networks in Medical Diagnosis / Q. K. Al Shayea // IJCSI International Journal of Computer Science Issues. – 2011 – Vol. 8, № 2. – P. 150–154.

УДК 004.048

КОНЦЕПТУАЛЬНЫЙ ПОДХОД КЛАССИФИКАЦИИ ДЕСТРУКТИВНОГО ПОВЕДЕНИЯ С ПРИМЕНЕНИЕМ ТЕХНОЛОГИЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Н. Н. Тетерин[1], В. В. Смоленцева[2]

[1]МИРЭА – Российский технологический университет
[2]Российский государственный университет социальных технологий

Аннотация. Рассмотренный в работе концептуальный подход к классификации с применением технологий искусственного интеллекта (ИИ) является эффективным инструментом для предотвращения последствий от деструктивного поведения сотрудников. В условиях растущей роли информационных технологий в организациях и управлении персоналом, проблема выявления и предотвращения деструктивного поведения бесспорно является актуальной. Деструктивное поведение может проявляться в различных формах, нанося ущерб репутации и безопасности компании. Основные этапы подхода включают сбор данных, предварительную обработку, выбор признаков, обучение модели, тестирование, классификацию. Преимущества подхода включают автоматизацию рассматриваемого в работе процесса и своевременное и обоснованное принятие решений руководством.
Ключевые слова: концептуальный подход, искусственный интеллект, деструктивное поведение, алгоритмы машинного обучения, социальная сеть.

Введение

Деструктивное поведение сотрудника в социальных сетях — это поведение, которое наносит ущерб организации, её репутации или другим сотрудникам.

Такое поведение может включать в себя:

– Распространение негативной информации о компании или коллегах. Например: критика руководства, претензии на условия работы или ложная информация, которая вредит репутации компании.

– Дезинформация. Например: деструктивные сотрудники распространяют ложную информацию, которая вводит в заблуждение коллег или клиентов.

– Оскорбления. Например: поведение сотрудника приводит к психологическому дискомфорту коллег и создает негативную атмосферу в коллективе.

– Спам и фишинг. Например: деструктивные сотрудники используют социальные сети для распространения спама или фишинговых атак, что приводит к утечке данных или финансовым потерям.

Для классификации поведения деструктивного сотрудника в социальных сетях по большей части используют следующие методы:

Анализ содержания — метод, который используется для изучения и интерпретации текстовых данных, размещённых сотрудниками в социальных сетях. С помощью алгоритмов машинного обучения можно автоматически анализировать тексты и выявлять в них негативные высказывания или распространение ложной информации.

Анализ содержания полезен для организаций, так как он позволяет наблюдать за настроением сотрудников, выявлять проблемы и конфликты, а также предотвращать распространение негативной информации. Однако при использовании этого метода необходимо учитывать возможные ограничения и риски, такие как возможность ложных срабатываний алгоритмов или нарушение приватности сотрудников.

Важно разработать чёткие критерии и правила для анализа содержания. Также необходимо учитывать контекст и особенности каждого конкретного случая.

В целом, анализ содержания может быть полезным инструментом для компаний, но его применение должно быть обоснованным и соответствовать законодательству.

Анализ связей — метод, который предполагает изучение взаимодействий сотрудника в социальных сетях. Он позволяет определить, с кем сотрудник общается и какие темы обсуждаются, для выявления деструктивного поведения, связанного с распространением негативной информации.

Анализ взаимодействий в социальных сетях дает представление о корпоративной культуре компании, что помогает понять, какие ценности и нормы приняты в коллективе. Анализ может помочь выявить потенциальные конфликты до их эскалации, что позволяет своевременно принять меры для их предотвращения.

Сетевой анализ в социальных сетях — метод исследования, который позволяет визуализировать и анализировать связи между участниками сети. В контексте изучения взаимодействий сотрудников, сетевой анализ может помочь выявить структуру коммуникаций внутри организации.

Для проведения сетевого анализа необходимо собрать данные о взаимодействиях между сотрудниками. Это могут быть сведения о частоте общения, содержании разговоров, а также о наличии или отсутствии связей между участниками сети. Собранные данные можно представить в виде графа, где вершины обозначают сотрудников, а рёбра — связи между ними.

Мониторинг упоминаний — процесс сбора и анализа информации о том, как сотрудники обсуждают компанию, коллег и руководство.

Мониторинг позволяет обнаружить негативные высказывания о компании, коллегах или руководстве. Анализ упоминаний позволяет выявить, кто именно оставляет негативные отзывы, что помогает сосредоточиться на работе с конкретными сотрудниками или группами, чтобы изменить их отношение к компании.

Внутренние каналы связи для анализа мониторинга упоминаний: корпоративный чат, электронная почта, внутренние новостные ленты, социальные сети, форумы и другие ресурсы, где сотрудники могут высказываться о компании.

Анализ тональности текста — метод, который позволяет определить эмоциональную окраску сообщения (тональность) и выявить в нём присутствие негативных или деструктивных элементов. Такой анализ может проводиться с помощью алгоритмов и компьютерных программ.

Анализ тональности может быть как бинарным, так и многоклассовым. В первом случае тексты относят к одной из двух категорий: положительные или отрицательные. Во втором случае выделяют больше двух категорий, например: положительные, отрицательные, нейтральные, смешанные.

Существуют разные подходы к анализу тональности текста. Одни методы анализируют отдельные слова и их сочетания, другие учитывают контекст и структуру предложений. Наиболее распространёнными алгоритмами являются методы, основанные на машинном обучении, такие как SVM (метод опорных векторов), Random Forest (случайный лес) и нейронные сети [1].

Анализ тональности текста может быть полезен для бизнеса, государственных органов, научных исследователей и обычных пользователей. Он помогает принимать обоснованные решения, следить за общественным мнением и выявлять потенциально опасные тенденции.

Анализ контекста — метод, который позволяет выявить причины деструктивного поведения сотрудника через анализ контекста публикаций негативных высказываний или ложной информации. Контекст — это обстоятельства, факты и условия, которые окружают публикацию.

Анализ тональности публикации. Необходимо определить, какой тон имеет публикация — агрессивный, обидный, саркастический или нейтральный.

Для применения описанных выше методов необходимо сформировать входной набор данных, схема формирования единой базы входного набора представлена на рис. 1.

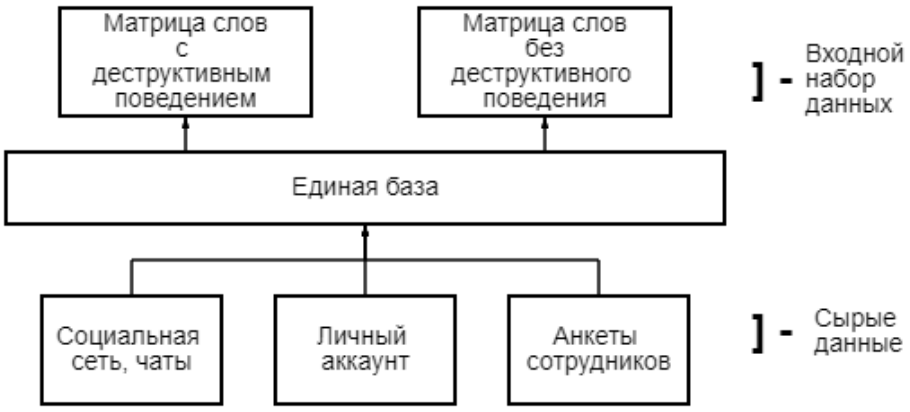


Рис. 1. Схема концептуального подхода формирования входного набора данных

Матрица слов и словосочетаний для классификации деструктивного и недеструктивного поведения, является полезным инструментом для идентификации и анализа текстовых данных.

К ключевым этапам обработки текста относятся:

1. Токенизация — процесс разделения текста на отдельные слова или токены.

2. Удаление стоп-слов — исключение часто встречающихся слов, которые не несут значимого смысла.

3. Лемматизация и стемминг — методы приведения слов к их базовым формам, что позволяет унифицировать различные морфологические варианты.

Анализ текстовых данных с помощью алгоритмов обработки естественного языка (NLP) повысит точность анализа и классификации текстов, и эффективно определит деструктивные и недеструктивные элементы в социальных сетях пользователей.

Для повышения качества анализа текстовых данных в комбинации с другими методами NLP предлагается использование TF-IDF.

TF-IDF подходит хорошо для задач, связанных с извлечением ключевых слов, кластеризацией и классификацией. Применение позволяет улучшить точность анализа и классификации текстов, а также эффективно выявлять признаки деструктивного и недеструктивного поведения в социальных сетях.

Для применения методов классификации поведения деструктивного сотрудника в социальных сетях можно использовать специальные программы и алгоритмы машинного обучения. С их помощью проанализировать тексты, связи и поведение сотрудников в социальных сетях для выявления деструктивного поведения и предоставлении информации для принятия мер [2].

На рис. 2 приведены возможные этапы процесса классификации текста деструктивного сотрудника в социальных сетях.

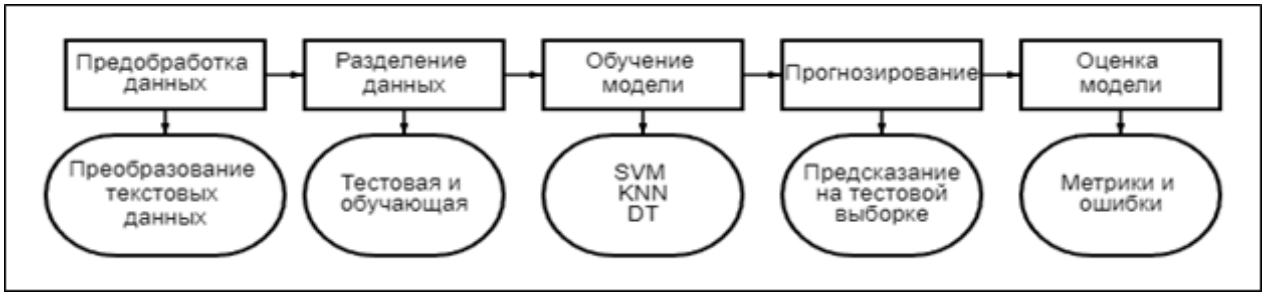


Рис. 2. Этапы процесса классификации текста деструктивного сотрудника в социальных сетях

При выявлении деструктивного поведения сотрудника в социальных сетях можно принять следующие меры: обсудить поведение сотрудника с ним лично, провести обучение и тренинги для сотрудников, что поможет повысить осведомлённость о деструктивном поведении и научить сотрудников распознавать его, создать политику использования социальных сетей в организации [3].

Заключение

Применение методов классификации в контексте формирования концептуального подхода выявления деструктивного сотрудника в социальных сетях может помочь выявить и предотвратить деструктивное поведение, которое может нанести ущерб организации. Для этого можно использовать специальные программы и алгоритмы машинного обучения, которые могут анализировать тексты, связи и поведение сотрудников в социальных сетях. При выявлении деструктивного поведения можно принять меры, такие как обсуждение поведения с сотрудником, проведение корпоративного обучения и создание политики использования социальных сетей в организации.

Литература

1. *Семериков А. В.* Классификация объектов на основе нейронной сети и методами дерева решения и ближайших соседей : учебное пособие / А. В. Семериков, М. А. Глазырин. – Ухта : УГТУ, 2022. – 68 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/267857 (дата обращения: 29.10.2024). – Режим доступа: по подписке.

2. *Благов А. В.* Анализ социальных сетей : учебное пособие / А. В. Благов, И. А. Рыцарев. – Самара : Самарский университет, 2020. – 104 с. – ISBN 978-5-7883-1556-0. – Текст : электронный // Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/188862 (дата обращения: 01.11.2024). – Режим доступа: по подписке.

3. *Базерман М.* Как принять правильное управленческое решение / М. Базерман, Д. Мур ; Александр Алексеев. – Москва : Альпина Паблишер, 2024. – 400 с. – ISBN 978-5-206-00336-9. – Текст : электронный // Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/425585 (дата обращения: 01.11.2024). – Режим доступа: по подписке.

УДК 004.42

АНАЛИТИКА РЫНКА ТРУДА И ЗАНЯТОСТИ НА ОСНОВЕ ПРИМЕНЕНИЯ МЕТОДОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Ф. Г. Типикина

Воронежский государственный университет

Аннотация. Работа посвящена обзору исследований в области применения методов искусственного интеллекта (ИИ) к аналитике рынка труда и занятости. В ней анализируются как позитивные аспекты внедрения технологий, таких как машинное обучение и обработка естественного языка, так и возникающие проблемы, включая качество данных и необходимость подготовки сотрудников к новым требованиям. В заключении подчеркивается, что использование ИИ постепенно становится ключевым фактором для организаций, стремящихся оставаться конкурентоспособными в современных условиях.
Ключевые слова: искусственный интеллект, рынок труда, машинное обучение, обработка естественного языка, качество данных, большие данные, прогнозирование спроса и предложения, анализ навыков, автоматизация подбора кадров, эффективность рекрутинга, стратегическое планирование.

Введение

Рынок труда находится в постоянном движении, реагируя на технологические изменения, глобализацию и демографические факторы. Этот динамичный процесс обусловлен множеством причин, включая быстрое развитие информационных технологий, изменение потребностей работодателей и поведение работников, а также влияние международных экономических отношений [5].

Понимание этих тенденций и прогнозирование будущих изменений становится все более сложным, особенно в условиях высоких темпов развития. В такой сложной и постоянно меняющейся среде искусственный интеллект становится мощным инструментом, способным анализировать огромные объемы данных и выявлять скрытые закономерности, которые человеку было бы трудно заметить. Он предоставляет новые возможности для глубокого анализа рынка труда, позволяя организациям не только реагировать на текущие изменения, но и предсказывать их.

1. Применение искусственного интеллекта при анализе рынка занятости

1. Обработка больших данных.

ИИ может анализировать данные из различных источников, включая онлайн- резюме, веб-сайты вакансий, социальные сети и профессиональные платформы. Используя алгоритмы обработки естественного языка, ИИ способен извлекать ключевую информацию о навыках, опыте и предпочтениях соискателей, что позволяет выявлять текущие тренды и изменения на рынке труда.

Одним из исследований, посвященных анализу больших данных в области найма и рынка труда, стало исследование, проведенное компанией LinkedIn. В 2021 году она выпустила отчет под названием «2021 Workplace Learning Report» [2], в котором рассмотрели данные о профессиональных навыках и карьерных предпочтениях пользователей платформы.

Для выявления актуальных трендов в востребованных навыках LinkedIn использовал алгоритмы обработки естественного языка (NLP), анализируя обширный массив резюме и профилей пользователей. В процессе исследования были собраны данные из различных источников на платформе, включая онлайн-резюме, вакансии и профили пользователей, что дало

возможность определить, какие навыки становятся все более актуальными, какие профессии пользуются спросом и какие изменения происходят на рынке труда.

1. Прогнозирование спроса и предложения.

Системы машинного обучения могут использовать исторические данные для выявления закономерностей и прогнозирования будущих изменений в спросе и предложении на определенные профессии. Учитывая различные факторы, такие как экономические условия, изменения в законодательстве и бурный й рост технологий, такие прогнозы могут быть полезны для стратегического планирования как для работодателей, так и для соискателей.

Одним из примеров исследований, посвящённых прогнозированию спроса и предложения на рынке труда, является работа, проведённая компанией McKinsey & Company в 2020 году под названием «The Future of Work after COVID-19» (Будущее работы после COVID-19) [3]. В ходе данного исследования McKinsey использовал модели машинного обучения и методы анализа больших данных для оценки влияния пандемии на рынок труда в различных отраслях экономики. Исследование включало анализ исторических данных по занятости, изменению экономических показателей и колебаниям потребительских предпочтений. Основное внимание уделялось тому, как будут изменяться актуальные навыки и профессии в ближайшие годы, а также тому, как работодатели могут адаптироваться к новым условиям.

2. Анализ навыков.

ИИ-системы могут анализировать требования работодателей к навыкам и умениям, что позволяет не только выявить профессии с дефицитом квалифицированных кадров, но и предлагать модули обучения и курсы. Используя кластеризацию данных, ИИ может также выявлять связанные навыки и предлагать многопрофильные карьерные пути для соискателей.

Например, HT Lab (лаборатория «Гуманитарные технологии») создала инструмент на основе искусственного интеллекта — комплексную методику «Бизнес-Профиль», которая сочетает в себе личностные, интеллектуальные и мотивационные тесты [4]. Эта методика идеально подходит для детального анализа кандидата, помогает выяснить его мотивацию, профессиональные возможности, структуру интеллекта, а также оценить его компетенции, управленческие стили и уровень конфликтности. Таким образом, можно получить больше информации о кандидате и определить, в каком направлении ему следует развиваться в компании.

3. Обработка резюме и автоматизация подбора кадров.

ИИ может существенно ускорить процесс рекрутинга, автоматизируя анализ резюме. Алгоритмы могут сопоставлять резюме с вакансиями, выделяя наиболее подходящих кандидатов и минимизируя человеческий фактор. Это повышает эффективность подбора, позволяя HR-специалистам сосредоточиться на более важных аспектах работы, таких как оценка культурной совместимости и выражение интереса к кандидатам.

Компания LinkedIn, специализированная социальная сеть для поиска и установления деловых контактов, представила своего первого ИИ-агента, который исполняет роль рекрутера. В настоящее время этот инструмент доступен лишь ограниченному числу клиентов, среди которых находятся крупные компании, такие как AMD, Canva, Siemens и Zurich Insurance. Тем не менее, в ближайшем будущем LinkedIn планирует предоставить доступ к этому инструменту большему числу пользователей [1].

Hiring Assistant — это последняя и наиболее значимая разработка компании в данной области. По словам вице-президента по продуктам LinkedIn Хари Шринивасана, этот ИИ- решение выполняет рутинные задачи, что позволяет рекрутерам сосредоточиться на более стратегических аспектах своей работы.

Инструмент предоставляет возможность загружать или создавать описания вакансий, выделять ключевые навыки и автоматически искать подходящих кандидатов, основываясь на их профессиональных компетенциях, а не только на таких факторах, как местонахождение или образование.

В будущем LinkedIn планирует расширить функциональные возможности помощника, добавив функции для обмена сообщениями, планирования интервью и взаимодействия с кандидатами на всех этапах процесса найма.

2. Преимущества использования ИИ при анализе рынка занятости

1. Ускорение процессов.

Автоматизация рутинных задач, таких как отбор резюме и анализ данных, позволяет HR-менеджерам и рекрутерам быстрее реагировать на изменения в потребностях компании.

2. Точность прогнозов

Алгоритмы ИИ способны обрабатывать данные с высокой скоростью и точностью, что значительно улучшает качество и надежность прогнозов относительно динамики рынка труда.

3. Индивидуализация подбора.

ИИ может предоставлять соискателям наиболее релевантные вакансии, учитывая их опыт и предпочтения, что улучшает их шансы на трудоустройство и повышает удовлетворенность работой.

4. Улучшение принятия решений.

С помощью аналитики и прогнозов, основанных на данных, работодатели могут более эффективно принимать стратегические решения относительно выставления вакансий, повышения квалификации сотрудников и адаптации к изменениям в экономике.

3. Проблемы, связанные с применением ИИ при анализе рынка занятости

1. Качество данных.

Эффективность анализа рынка занятости существенно зависит от качества изначально используемых данных. Неполные, устаревшие или неточные данные могут снижать точность прогнозов и рекомендаций, что повышает важность обеспечения высокого качества собираемых данных.

2. Технические проблемы.

Поскольку ИИ-системы зависят от алгоритмов и программного обеспечения, существует риск технических сбоев и ошибок. Неверные выводы, основанные на недостаточных или некорректных данных, могут негативно сказаться на процессе подбора кадров и принятии решений.

3. Сопротивление внедрению технологий.

Внедрение систем ИИ может вызывать сопротивление со стороны сотрудников и руководства. Опасения, связанные с потерей рабочих мест, недовольство по поводу непрозрачности процессов принятия решений и необходимость изменений в организационной культуре могут замедлить адаптацию новых технологий.

4. Правовые и регуляторные рамки.

Отсутствие четких правовых норм и регуляций в области использования ИИ для анализа занятости также может стать серьезным ограничением. Законодательство должно развиваться в ногу с технологическими изменениями, чтобы обеспечить защиту прав работников и предотвратить возможные злоупотребления.

5. Зависимость от технологий.

Увеличение автоматизации и применения ИИ может привести к слишком большой зависимости компаний от технологий, что ставит под угрозу устойчивость бизнеса в случае технических сбоев или кибератак. Необходима диверсификация подходов к управлению работой, чтобы минимизировать риски, связанные с нарушением работы систем ИИ.

Заключение

Рынок труда постоянно развивается под влиянием технологического прогресса, глобализации и демографических факторов [6]. Анализ рынка занятости с помощью искусственного интеллекта предоставляет организациям возможности для понимания текущих и прогнозирования будущих тенденций, позволяя им адаптироваться к изменяющимся условиям.

ИИ-системы способны обрабатывать большие объемы данных, выявлять закономерности и прогнозировать спрос и предложение на рынке труда. Они также могут анализировать навыки, автоматизировать подбор кадров и предоставлять персонализированные рекомендации. Однако существуют проблемы, связанные с применением ИИ, такие как качество данных, технические сбои и сопротивление внедрению.

По мере развития технологий и появления новых инструментов на основе ИИ организации будут все больше полагаться на анализ рынка труда с помощью ИИ, чтобы оставаться конкурентоспособными и принимать обоснованные решения относительно своего персонала. Понимание динамики рынка труда и использование ИИ-инструментов для ее прогнозирования станут ключевыми факторами успеха на современном динамичном рынке.

Литература

1. LinkedIn launches its first AI agent to take on the role of job recruiters // TechCrunch. URL: https://techcrunch.com/2024/10/29/linkedin-launches-its-first-ai-agent-to- take-on-the-role-of-job-recruiters/ (дата обращения: 14.10.2024).

2. LinkedIn Learning's 2021 Workplace Learning Report – It's all about rapid skill building at scale // Job Market Monitor. URL: https://jobmarketmonitor.com/2021/04/26/linkedin- learnings-2021-workplace-learning-report-its-all-about-rapid-skill-building-at-scale/ (дата обращения: 14.10.2024).

3. The future of work after COVID-19 // McKinsey & Company. URL: https://www.mckinsey.com/featured-insights/future-of-work/the-future-of-work-after-covid-19 (дата обращения: 14.10.2024).

4. Бизнес-Профиль Универсальный психодиагностический комплекс // HT LAB. URL: https://ht-lab.ru/testy/biznes-profil/ (дата обращения: 21.10.2024).

5. *Игнатьева Е. А., Базылев Я. С.* Оценка влияния цифровизации на рынок труда: возможности и риски // Оригинальные исследования. – 2022. – Т. 12, № 11. – С. 414–421.

6. *Романова Е. С.* Прогнозируемые последствия внедрения искусственного интеллекта на рынке труда в сфере информационных технологий // Международные коммуникации. – 2019. – №1 (10). – С. 6.

УДК 004.8

ПОДХОД К РЕШЕНИЮ ЗАДАЧИ КЛАССИФИКАЦИИ СЛУХОВ В ИНТЕРНЕТ-НОВОСТЯХ

А. Д. Худобин, И. Е. Воронина

Воронежский государственный университет

Аннотация. Рассматривается подход к решению задачи классификации слухов в новостях на основе продукционных правил. Представлены сравнительные результаты применения разного рода методов машинного обучения и нейронных сетей с использованием различных способов векторизации. Непроверенная информация в новостных сайтах приобретает характер не только информационного мусора, но и способна в отдельных случаях нанести существенный вред потребителям. Решаемая задача носит нетривиальный характер, актуальна и не имеет стандартного решения.
Ключевые слова: слухи, классификация, машинное обучение, нейронные сети, TF-IDF, Bag-of-words, Word2Vec, N-граммы, LSTM, RNN, продукционные правила.

Введение

Новостные сайты в интернет-пространстве содержат достаточное количество непроверенной информации. Это становится достаточно серьезной проблемой.

Во многих современных СМИ «желтой» направленности есть рубрики с названиями: «Слухи», «Сплетни», «Версия». Формально подобные рубрики анонсируют, что в них содержится непроверенная информация, а фактически они часто используются для очернения действующих лиц [1]. Сейчас новости, которые можно отметить как слухи, встречаются в СМИ не только в отдельных, предназначенных для них разделах, но и там, где должны быть факты [2].

Слух — высказывание, имеющее особую композицию, стилистические особенности и типичное содержание, то есть хорошо узнаваемое. Это массовое распространение актуальной для людей информации, авторство и достоверность которой не установлены и ответственность за которую никто не несет [3].

Факт — событие, которое уже произошло, его достоверность установлена.

Объемы информации таковы, что актуальна задача автоматического разделения слухов и фактов. Это важно, потому что слухов в новостях становится все больше, а реализаций способных их удалить, нет. Для решения данной задачи необходимо разработать подход, позволяющий выполнить классификацию слухов и фактов.

1. Сложности решения и предлагаемый подход

Для решения задачи отделения слухов от фактов предлагается использовать методы машинного обучения. Важной частью работы является создание исходного набора данных, которого нет. Для этого необходимо выгрузить новости с различных сайтов и провести разметку данных вручную, несмотря на очевидную трудоемкость процесса. В ходе работы возникла проблема: новости, относящиеся к классу «Политика», не получалось разделить на слухи и факты [2]. Поэтому понадобилось разработать набор правил, который позволил бы выполнить эти действия. Правила должны показать, существует ли возможность корректного разбиения на классы.

Предлагается следующий подход к решению:

1. Выгрузить новости с различных сайтов, провести разметку данных вручную.
2. Сформулировать продукционные правила для каждого класса.

3. Провести предварительную обработку данных.

4. Провести векторизацию различными способами (TF-IDF, Bag-of-words, Word2Vec, Pretrained Word2Vec) с целью выбора наиболее подходящего.

5. Решить задачу классификации, используя нейронные сети и методы машинного обучения.

6. Провести сравнение методов, сделать выводы об их эффективности.

2. Набор данных

В [2] приводятся результаты создания набора данных из 6 классов: «Политика», «Политика слухи», «Спорт», «Спортивные слухи», «Наука», «Научные слухи». Новости были выгружены с сайтов sports.ru, argumenti.ru.

Новости, которые относились к политике, не удалось корректно разбить на классы. Поэтому классы, относящиеся к политике, было решено удалить.

В датасет добавлены классы «Шоу-бизнес» и «Шоу-бизнес слухи». Новости для них были выгружены с сайтов 7days.ru и argumenti.ru.

Набор данных содержит 6570 новостей.

В табл. 1 показано, сколько данных содержит каждый класс и сколько новостей есть в тестовом и тренировочном наборах данных.

Таблица 1

Данные в датасете

Наука	Наука слухи	Спорт	Спорт слухи	Шоу-бизнес	Шоу-бизнес слухи
1095	1095	1095	1095	1095	1095
Тренировочная часть			Тестовая часть		
5256			1314		

В [2] данные были несбалансированными. В новом наборе данных количество новостей в каждом классе одинаково.

3. Набор правил

Для разбиения новостей на классы были разработаны продукционные правила.

Правило1 для каждого абзаца:

Если абзац не содержит «недавно/давно появилась новость» и абзац не содержит «читайте далее новость», то абзац_смысл=да

(Правило необходимо для того, чтобы отбросить текст, в котором идет описание предыдущих новостей, так как необходимо сделать вывод по конкретной новости, а не по связке).

Правило2:

Если абзац_смысл==да и в новости человек говорит о себе и событие, о котором говорят в новости уже произошло, то шоу-бизнес факт

Правило3:

Если абзац_смысл==да и в новости говорит официальный представитель и событие, о котором говорят в новости уже произошло, то шоу-бизнес факт

Правило4:

Если абзац_смысл==да и новость содержит опровержение, то шоу-бизнес факт

Правило5:

Если абзац_смысл==да и в новости говорит не сам человек и в новости говорит не официальный представитель, то шоу-бизнес слух

Правило6:

Если абзац_смысл==да и (новость содержит «по слухам» или новость содержит «СМИ узнали» или новость содержит «по слухам»), то шоу-бизнес слух

Правило7:

Если абзац_смысл==да и человек говорит о событии, которое еще не произошло, то шоу-бизнес слух

Правило8:

Если абзац_смысл==да и (новость содержит «широкая одежда» или новость содержит «тайная свадьба»), то шоу-бизнес слух

Правило9:

Если абзац_смысл==да и в новости пишут об уфологах, то наука слух

Правило10:

Если абзац_смысл==да и (в новости говорит представитель НАСА или в новости говорит представитель Роскосмоса) и говорят о планах на что-то, то наука слух

Правило11:

Если абзац_смысл==да и (в новости говорит представитель НАСА или в новости говорит представитель Роскосмоса) и говорят о результатах какого-либо процесса, то наука факт

Правило12:

Если абзац_смысл==да и (в новости говорит представитель НАСА или в новости говорит представитель Роскосмоса) и называют точную дату начала какого-либо процесса, то наука факт

Правило13:

Если абзац_смысл==да и в новости пишут о раскопках древних сооружений, то наука факт

Правило14:

Если абзац_смысл==да и в новости пишут об уже произошедшем запуске ракеты (спутника), то наука факт

4. Предварительная обработка данных

Для решения задачи классификации необходимо провести предварительную обработку данных:

1. Очистить новости, удалить лишние символы.
2. Обрезать каждую новость, оставить только первые 1350 символов.
3. Разбить новости на токены.
4. Провести лемматизацию всех токенов.

5. Решение методами машинного обучения и нейронными сетями

Результаты оцениваются с помощью метрики точности (Accuracy), которая показывает долю правильных классификаций. Будем использовать следующие методы машинного обучения: градиентный бустинг, гребневая регрессия, случайный лес, метод опорных векторов, стохастический градиентный спуск, логистическая регрессия, метод k ближайших соседей, дерево решений.

В табл. 2 представлена точность классификации методами машинного обучения для способов векторизации TF-IDF и Bag-of-words. Указан только лучший результат для каждого способа векторизации.

На рис. 1. Изображена матрица ошибок для классификации со способом векторизации TF-IDF (1, 2) и методом машинного обучения гребневая регрессия.

Таблица 2

Точность классификации для TF-IDF и Bag-of-words

Способ векторизации	Общая точность	Наука	Шоу-бизнес	Метод машинного обучения
TF-IDF (1, 1)	0.820	0,746	0,727	Метод опорных векторов
TF-IDF (1, 2)	0,829	0,774	0,744	Гребневая регрессия
TF-IDF (1, 3)	0,820	0,771	0,717	Градиентный спуск
TF-IDF (2, 2)	0,792	0,745	0,671	Градиентный спуск
TF-IDF (1, 6)	0,807	0,759	0,703	Градиентный спуск
BOW (1, 1)	0,787	0,718	0,684	Логистическая регрессия
BOW (1, 2)	0,805	0,734	0,705	Гребневая регрессия
BOW (1, 3)	0,804	0,743	0,680	Логистическая регрессия
BOW (2, 2)	0,780	0,723	0,662	Градиентный спуск

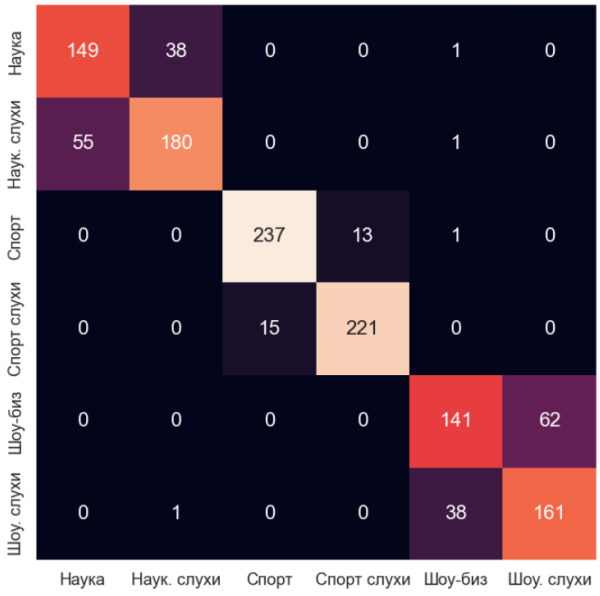


Рис. 1. Матрица ошибок для TF-IDF (1, 2)

С использованием Word2Vec было проведено сравнение предобученных моделей и собственных, созданных на основе обучающего набора данных. В собственных моделях проводилось сравнение нескольких параметров модели: количество эпох, размер окна, алгоритм обучения. Эпоха — итерация прохода текста. Размер окна — количество слов в контексте текущего слова.

В табл. 3 показана точность классификации методами машинного обучения для способа векторизации Word2Vec. Указан только лучший результат для каждого способа векторизации.

При увеличении количества итераций в собственном Word2Vec, увеличивается точность. Изменение размеров окна не оказывает большого влияния на результат.

RNN — рекуррентная нейронная сеть. LSTM — рекуррентная нейронная сеть, которая способна запоминать значения как на короткие, так и на длинные промежутки времени.

Таблица 3

Точность классификации для Word2Vec

Способ векторизации, количество эпох, окно	Общая точность	Наука	Шоу-бизнес	Метод машинного обучения
CBOW, 100, 2	0,800	0,723	0,718	Логистическая регрессия
CBOW, 100, 5	0,796	0,714	0,733	Логистическая регрессия
CBOW, 300, 2	0,800	0,716	0,728	Логистическая регрессия
CBOW, 300, 5	0,796	0,728	0,717	Гребневая регрессия
CBOW, 10, 2	0,778	0,740	0,702	Градиентный бустинг
CBOW, 10, 5	0,769	0,702	0,690	Метод опорных векторов
CC.RU pretrained, 10, 5	0,737	0,721	0,618	Градиентный бустинг
RUWIKI pretrained, 10, 5	0,791	0,738	0,707	Градиентный бустинг
WIKI-LENTA pretrained, 10, 5	0,790	0,735	0,700	Градиентный спуск
SG, 10, 2	0,772	0,706	0,694	Метод опорных векторов
SG, 10, 5	0,772	0,710	0,659	Градиентный спуск
SG, 100, 2	0,796	0,730	0,713	Метод опорных векторов
SG, 100, 5	0,802	0,735	0,738	Логистическая регрессия
SG, 300, 2	0,788	0,711	0,706	Градиентный спуск
SG, 300, 5	0,802	0,735	0,734	Логистическая регрессия

В табл. 4 показана точность классификации нейронными сетями для способов векторизации SG, 100, 5, RUWIKI pretrained, 10, 5.

Таблица 4

Точность классификации нейронными сетями LSTM и RNN

Способ векторизации, нейронная сеть	Общая точность	Наука	Шоу-бизнес
SG, 100, 5, LSTM	0,788	0,739	0,644
SG, 100, 5, RNN	0,588	0,569	0,528
RUWIKI pretrained, 10, 5, LSTM	0,770	0,725	0,621
RUWIKI pretrained, 10, 5, RNN	0,544	0,493	0,522

На рис. 2. Изображена матрица ошибок для классификации со способом векторизации SG, 100, 5 и нейронной сетью LSTM.

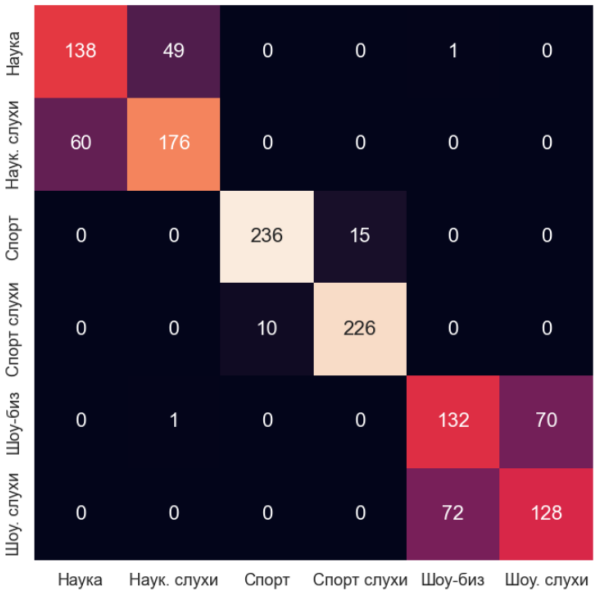


Рис. 2. Матрица ошибок для SG, 100, 5, нейронная сеть LSTM

Заключение

Представим основные выводы.

Таким образом, предлагается подход к решению задачи классификации слухов в новостях. В [2] был создан набор данных, который содержит новости науки, политики и спорта. Новости политики не удалось корректно разбить на классы, поэтому классы, относящиеся к политике, было решено удалить. Для корректного разбиения новостей на классы были созданы продукционные правила, которые позволяют выполнить эти действия. В набор данных были добавлены новые классы: «Шоу-бизнес», «Шоу-бизнес слухи».

В [2] данные были несбалансированными. В новом наборе данных количество новостей в каждом классе одинаково.

Была проведена предобработка данных и решена задача классификации различными методами машинного обучения и нейронными сетями с использованием различных способов векторизации.

Лучшая точность была получена с использованием алгоритмов машинного обучения (гребневая регрессия) и способа векторизации TF-IDF (1, 2). Решения с использованием способа векторизации TF-IDF показывали более высокую точность, чем решения с использованием Word2Vec. Решение с использованием собственных Word2Vec показали более высокую точность, чем решения с использованием предобученных Word2Vec. При увеличении количества итераций в собственном Word2Vec, увеличивается точность.

Решения с использованием методов машинного обучения показали более высокую точность, чем решения с использованием нейронных сетей. Решения с использованием RNN не справились с задачей и показали очень низкую точность.

Литература

1. Лингвистическая экспертиза в трудах Воронежской ассоциации экспертов-лингвистов : монография / Ж. В. Грачева [и др.] ; под ред. М. Е. Новичихиной, А. В. Рудаковой. – Воронеж : Издательский дом ВГУ, 2023. – 357 с.

2. *Худобин А. Д.* TF-IDF, Bag-of-words, Word2Vec и N-граммы для решения задачи классификации слухов в новостях / А. Д. Худобин, И. Е. Воронина // Информатика: проблемы, методы, технологии. – 2024. – С. 1285–1294.

3. *Осетрова Е. В.* Слухи в парадигме лингвистической генристики / Е. В. Осетрова // Жанры речи 2. – 2015. – № 12. – С. 80–89.

УДК 004.932

СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ АНАЛИЗА ИЗОБРАЖЕНИЙ В ОБЛАСТИ ОФТАЛЬМОЛОГИИ

А. А. Худяков[1], А. А. Арзамасцев[1, 2]

[1]Воронежский государственный университет
[2]Тамбовский филиал Федерального государственного автономного учреждения «Национальный медицинский исследовательский центр «Межотраслевой научно-технический комплекс «Микрохирургия глаза» имени академика С. Н. Федорова»

Аннотация. Данная статья представляет собой обзор возможностей применения систем искусственного интеллекта в офтальмологии для анализа изображений, полученных с различных медицинских приборов. Разбираются недостатки и ограничения классических подходов, таких как мультиклассовая классификация с использованием сверточных нейронных сетей. Рассматриваются последние исследования и достижения на стыке офтальмологии и машинного обучения, предлагающие передовые подходы с применением механизмов внимания и ансамблей моделей, а также алгоритмы преобразования изображений и удаления шумов.
Ключевые слова: искусственный интеллект, нейронные сети, сверточные нейронные сети, визуальные трансформеры, компьютерное зрение, офтальмология.

Введение

Как и во многих других областях медицины, в офтальмологии для постановки диагноза широко используется анализ изображений, полученных при помощи различных приборов, будь то оптическая когерентная томография (ОКТ), сканирующая лазерная офтальмоскопия (СЛО) и другие методы проведения исследования. Постановка диагноза в большинстве случаев может быть сведена к присвоению такому изображению соответствующей метки, обозначающей найденное заболевание или отклонение от нормы, то есть к широко распространенной и изученной в машинном обучении задаче классификации.

Помимо непосредственно задачи классификации, для работы с каждым видом офтальмологических изображений требуется индивидуальный подход, включая различия в размерах изображения, количестве цветов, подверженности появлению шумов и артефактов, смазанных снимков, поворотов. Это требует предварительной обработки снимков перед передачей их в нейронную сеть, а в некоторых случаях и перед постановкой диагноза самим врачом-диагностом.

Эти и другие задачи анализа изображений в офтальмологии, ранее казавшиеся неприступными для традиционных алгоритмов, сегодня можно решить при помощи различных методов машинного обучения, а появление и адаптация продвинутых архитектур, таких как визуальные трансформеры, делают возможными использование нейросетевых моделей в качестве полноценных ассистентов для специалистов отрасли, особенно на ранних стадиях прогрессирования заболевания.

1. Применение сверточных нейронных сетей в офтальмологии

При работе с изображениями в машинном обучении хорошо себя зарекомендовали сверточные нейронные сети (СНС). Первая часть такой сети представляет собой чередование сверточных и пулинговых слоев, отбирающих из данных существенные признаки, а вторая

состоит из нескольких полносвязных слоев. Результатом работы такой модели являются вероятности принадлежности изображения к одному из классов.

Этот подход достаточно прост и может быть довольно эффективен, но все же имеет ряд недостатков, большинство из которых заложены в его собственную архитектуру. Как и многие нейросетевые подходы, он предоставляет лишь конечный результат своей работы (вероятности классов), но не позволяет анализировать и интерпретировать промежуточные результаты и сам процесс принятия решения, который привел к итоговому ответу, что ограничивает применение таких моделей на практике. Монолитная и неделимая архитектура модели также требует полного переобучения при любом изменении своей структуры или структуры данных, например, при появлении нового класса, который представляет собой ранее не встречающееся для модели заболевание. Архитектура также не позволяет эффективно и полноценно диагностировать по изображению наличие у пациента сразу нескольких заболеваний, так как предсказанные вероятности предполагают лишь взаимоисключающие варианты.

На данный момент существует множество разновидностей архитектур СНС, отличающихся размерами сверточных ядер, количеством и порядком слоев и другими характеристиками, и многие из этих архитектур показывают достойные результаты в различных задачах офтальмологии. Найим А. и соавт. в своей работе [1] производят сравнение большого количества популярных архитектур на задаче диагностирования кератоконуса – дегенеративного невоспалительного заболевания глаза. Для сравнения были взяты такие архитектуры, как VGG16, ResNet50, EfficientNetB0, MobileNetV2, DenseNet121 и другие. Лучшие результаты показала модель MobileNetV2, успешно отличающая здоровый глаз от больного и справляясь со спорными ситуациями, показав впечатляющую точность в 98% на тестовой выборке. Модель с подобным уровнем точности уже может уверенно использоваться на практике и помогать в раннем диагностировании заболеваний.

Помимо задач классификации, модели на основе СНС могут использоваться для преобразования офтальмологических изображений. Частный случай таких моделей — генеративно-состязательные сети (GAN). Модель состоит из двух сетей, первая из которых генерирует данные, а вторая старается отличить подлинные данные от созданных. В работе Ксин Т. и соавт. [2] подобная сеть используется для преобразования изображений ОКТ в изображения конфокальной микроскопии. ОКТ является достаточно быстрым и неинвазивным методом получения трехмерных изображений, представляющих собой поперечные срезы различных структур глаза. Главным недостатком этого метода является большая зашумленность и артефакты, вызванные движением пациента или другими внешними факторами, что может сделать незаметными важные для диагноза детали. Это особенно критично на ранних стадиях заболевания, так как детали могут быть легко приняты за шум и пропущены как нейронной сетью, так и опытным врачом-диагностом. Более точные методы, такие как конфокальная микроскопия, являются инвазивными и занимают больше времени, но дают чистые изображения, хорошо подходящие для диагностики. Использование нейронных сетей для преобразования изображений в первую очередь является мощным инструментом удаления шумов из изображений ОКТ, существенно не увеличивая при этом время и сложность обследования.

В работе Слутвега И. и соавт. [3] СНС используются для классификации и расширения исходного набора данных при определении ранних стадий болезни Альцгеймера. Исходный набор данных состоит из изображений СЛО глазного дна, ОКТ и ОКТ-ангиографии сетчатки пациентов. Для обучения СНС нужна достаточно большая выборка, которую не всегда возможно получить или собрать из-за различных этических и других ограничений, присутствующих в медицине. Чтобы решить эту проблему, авторы используют особый вид генеративной модели — диффузионную вероятностную модель, идея которой была взята из неравновесной термодинамики. Она использует цепь Маркова, которая из исходного изображения, представленного в виде гауссовского шума, постепенно убирает шумы, тем самым генерируя новые

уникальные примеры для обучения мультимодального классификатора. Данное исследование является особенно важным в контексте того, что для полноценного диагностирования болезни Альцгеймера используются инвазивные и дорогостоящие методы, такие как AmyloidPET и анализ спинномозговой жидкости пациента, из-за чего их нецелесообразно и неудобно использовать на ранних стадиях болезни. Предложенное в статье неинвазивное офтальмологическое исследование значительно упрощает и удешевляет диагностику заболевания на ранних стадиях, позволяя раньше начать его купирование и замедлить развитие.

2. Использование визуальных трансформеров и механизма внимания

СНС хорошо обнаруживают локальные признаки в пределах небольших областей изображения, но данная локальность имеет недостаток — при обработке каждого пикселя ядром свертки он взаимодействует только с пикселями в небольшой области вокруг самого себя, а для сбора глобальной информации об изображении и связях между его частями, требуется большое количество слоев. Архитектура трансформера, использующая механизм внимания и хорошо зарекомендовавшая себя для обработки естественного языка, также получила свое применение в задачах компьютерного зрения. Она позволяет более эффективно обнаруживать признаки в глобальном контексте изображения и находить связи между отдаленными областями. Из недостатков таких моделей стоит выделить большой размер и требования к вычислительным ресурсам, тогда как модели с небольшим количеством параметров проигрывают СНС в точности предсказаний.

Так, работа Тухидул И. и соавт. [4] показывает возможности применения ансамбля моделей, состоящего из СНС и визуального трансформера, для обнаружения ключевых биомаркеров по изображениям ОКТ из набора данных OLIVES [5]. Использованные в исследовании модели по отдельности значительно уступают передовым решениям, в частности, из-за своего небольшого размера, но большую роль в этом играют и сами данные. Изображения с томографа могут отличаться яркостью, ориентацией и, как было упомянуто ранее, наличием большого количества шумов, которые очень непросто убрать. Использованный ансамбль моделей представляет собой параллельное вычисление результата классификации СНС и визуальным трансформером, а итоговый ответ является взвешенным средним полученных результатов, что позволяет компенсировать ошибки каждой модели по отдельности. Полученный ансамбль значительно превосходит входящие в него модели по метрике F1 и обладает небольшим общим количеством параметров по сравнению с передовыми решениями в области, что положительно сказывается на времени обучения и генерации ответов.

В совместной работе [6] ученых из СПбГЭТУ «ЛЭТИ» и Джадавпурского университета Индии показана модель классификатора для распознавания глаукомы по фотографиям глазного дна. Два последовательных блока с механизмом внимания используются поверх сверточной нейронной сети DenseNet121, которая отвечает за выделение признаков, но не за итоговый результат классификации. Полученная модель превосходит существующие передовые решения в точности прогнозов и метрике F1, используя наборы данных RIM-ONE [7] и ACRIMA для обучения и валидации.

Как и СНС, модели на основе механизма внимания помимо задач классификации могут использоваться для подготовки данных. В работе Йепенг Л. и соавт. [8] описывается использование сильно модифицированной модели на основе трансформеров для регистрации изображений глазного дна — нахождения соответствия между несколькими искаженными изображениями и приведение их к единой системе координат путем деформаций. Это позволяет привести разнородные или некорректно собранные данные к единому виду для упрощения диагностики.

Заключение

Современные архитектуры нейросетевых моделей получили широкое распространение в офтальмологии и используются как для диагностики, так и для обработки и подготовки данных с различных приборов. Результаты исследований показывают, что данные модели в некоторых случаях уже обладают достаточно высокой точностью предсказаний, чтобы использоваться в качестве ассистента врача-диагноста, и позволяют снизить нагрузку на медицинский персонал, заменяя собой некоторые дорогостоящие инвазивные методы. Помимо задач классификации, методы машинного обучения также могут использоваться для аугментации исходного набора данных, очистки изображений от шумов и других артефактов.

Благодарности

Работа выполнена в соответствии с договором о научно-техническом сотрудничестве Воронежского государственного университета и Федерального государственного автономного учреждения «Национальный медицинский исследовательский центр «Межотраслевой научно-технический комплекс «Микрохирургия глаза» имени академика С.Н. Федорова», Тамбовского филиала от 28.11.2022.

Литература

1. *Nayeem A.* Comparative Performance Analysis of Transformer-Based Pre-Trained Models for Detecting Keratoconus Disease / A. Nayeem, R. Maruf, I. Fatin, K. Imran, S. Sanowar, R. Sadekur. – URL: https://arxiv.org/abs/2408.09005 (дата обращения 05.10.2024).

2. *Xin T.* The Quest for Early Detection of Retinal Disease: 3D CycleGAN-based Translation of Optical Coherence Tomography into Confocal Microscopy / T. Xin, A. Nantheera, N. Lindsay, A. Alin. – URL: https://arxiv.org/abs/2408.04091 (дата обращения 03.10.2024).

3. *Slootweg I.* Generative artificial intelligence in ophthalmology: multimodal retinal images for the diagnosis of Alzheimer's disease with convolutional neural networks / I. Slootweg, M. Thach, K. Curro-Tafili [и др.]. – URL: https://arxiv.org/abs/2406.18247 (дата обращения 06.10.2024).

4. *Touhidul I.* Ophthalmic Biomarker Detection with Parallel Prediction of Transformer and Convolutional Architecture / I. Touhidul, A. Chowdhury, M. Hasan [и др.]. – URL: https://arxiv.org/abs/2409.17788 (дата обращения 10.10.2024).

5. *Prabhushankar M.* OLIVES Dataset: Ophthalmic Labels for Investigating Visual Eye Semantics / M. Prabhushankar, K. Kokilepersaud [и др.]. – URL: https://arxiv.org/abs/2209.11195 (дата обращения 10.10.2024).

6. *Chakraborty S.* A Dual Attention-aided DenseNet-121 for Classification of Glaucoma from Fundus Images / S. Chakraborty, A. Roy [и др.]. – URL: https://arxiv.org/abs/2406.15113 (дата обращения 09.10.2024).

7. *Fumero F.* RIM-ONE DL: A Unified Retinal Image Database for Assessing Glaucoma Using Deep Learning / F. Fumero, T. Diaz-Aleman [и др.]. – URL: https://www.ias-iss.org/ojs/IAS/article/view/2346 (дата обращения 09.10.2024).

8. *Liu Y.* Progressive Retinal Image Registration via Global and Local Deformable Transformations / Y. Liu, B. Yu [и др.]. – URL: https://arxiv.org/abs/2409.01068 (дата обращения 13.10.2024).

УДК 004.932.72

АЛГОРИТМ ПОСТОБРАБОТКИ ДЛЯ ПОВЫШЕНИЯ ТОЧНОСТИ СЕГМЕНТАЦИИ МУЛЬТИФАЗНЫХ КТ-ИЗОБРАЖЕНИЙ

А. В. Черемискин, И. Л. Каширина

Воронежский государственный университет

Аннотация. В данной работе рассматривается алгоритм постобработки для повышения точности сегментации мультифазных КТ-изображений, полученных с помощью нейронной сети U-Net. Основной целью является улучшение точности сегментации, устранение топологических ошибок и минимизация ложноположительных результатов. Алгоритм включает три этапа: удаление шумов, объединение пересекающихся патологий и объединение фаз контрастирования.
Ключевые слова: нейронные сети, сегментация, U-Net, постобработка.

Введение

Сегментация медицинских изображений является ключевой задачей в анализе данных. Однако разнообразие исследуемых объектов создает необходимость в универсальных инструментах. Исследования показали, что модели, хорошо работающие на разных типах задач, оказываются более эффективными, чем узкоспециализированные алгориитмы [1]. В контексте автоматизированного анализа компьютерных томограмм используется сверточная нейронная сеть архитектуры U-Net для мультиклассовой сегментации патологических образований. Выходные данные сети представляют собой вероятностные карты принадлежности каждого вокселя к различным классам патологий либо к фоновому классу. Текущий алгоритм классификации вокселей основан на принципе максимума, где каждому вокселю присваивается класс с наибольшей вероятностью. Однако данный подход не учитывает анатомические и патофизиологические особенности взаимного расположения патологических образований, что приводит к появлению топологически некорректных артефактов сегментации, в частности, к ошибочной идентификации патологических структур внутри других патологических образований. Данная проблема обуславливает необходимость разработки универсального алгоритма постобработки результатов сегментации, учитывающего топологические ограничения и анатомические особенности взаимного расположения патологических структур.

1. Описание исходных данных

В качестве исходных данных в этой работе использовались результаты сегментации нейронной сети, в которых наряду с истинными патологиями присутствовали ложные классы. Нейронная сеть тестировалась на открытом наборе данных из проекта The Medical Segmentation Decathlon (MSD) [2], содержащем трехмерные медицинские изображения. Для исследования был выбран поднабор данных, который содержит изображения печени в формате NIfTI, полученные в четырех фазах контрастирования. В рамках работы рассматривались две фазы с целью проверки гипотезы о повышении диагностической точности учета результатов на разных фазах. Использование изображений с разных фаз КТ при разработке автоматизированной модели сегментации патологий важно, потому что каждая фаза предоставляет уникальную информацию о тканях и структурах организма. Нативная фаза (NA) позволяет оценить естественные характеристики тканей без контраста и четко выделяет жировую ткань, кисты, кальцинаты. Портально-венозная фаза (PV) улучшает визуализацию сосудистых структур,

помогая обнаруживать патологии, связанные с кровотоком, такие как злокачественные новообразования. Комбинирование данных из разных фаз повышает точность модели, позволяя ей более эффективно различать и сегментировать различные типы патологий. Однако различия в интенсивности сигнала и присутствие шумов и артефактов затрудняют надежную сегментацию и распознавание патологий, требуя разработки сложных алгоритмов обработки и анализа изображений, способных адаптироваться к этим вариациям и обеспечивать высокую точность диагностики.

2. Предварительные эксперименты

В предварительных экспериментах выполнялась оценка точности модели сегментации патологий в двух фазах контрастирования (NA и PV) для выявления оптимальной фазы, при которой модель наилучшим образом обнаруживает истинные патологии. Результаты показали, что точность сегментации патологий в фазе PV превышала соответствующие значения в фазе NA. В частности, в фазе PV было выявлено больше патологий с истинным классом и меньше ложноположительных результатов по сравнению с фазой NA.

3. Алгоритм постобработки

В данной работе представлен алгоритм постобработки сегментации патологий на мультифазных КТ-изображениях, состоящий из трех последовательных этапов: удаление шумов, объединение пересекающихся патологий, объединение фаз контрастирования.

3.1. Удаление шумов

На первом этапе для фильтрации избыточных данных удаляются все найденные патологии объемом меньше заданного порога. Для этого создаётся трёхмерная бинарная карта, где каждому элементу присваивается значение 1 (наличие патологического образования) или 0 (отсутствие патологического образования). Затем для каждой связной области, отмеченной на бинарной карте значением 1, рассчитывается объем вокселей. Если объем области меньше порогового значения, она классифицируется как шум и исключается из дальнейшего анализа. Данный подход позволяет уменьшить количество ложноположительных результатов, связанных с мелкими артефактами и незначительными изменениями, не имеющими клинической значимости. После удаления таких областей остаются только те образования, которые потенциально могут представлять практический интерес, что улучшает точность последующих этапов алгоритма.

3.2. Объединение пересекающихся патологий

На втором этапе алгоритма проводится анализ пространственных пересечений между патологическими областями различных классов, выявленных на этапе сегментации. Для устранения невозможных с клинической точки зрения ситуаций, где разные патологии перекрываются, предложен следующий алгоритм:

1. Идентификация всех патологий независимо от их исходной классификации.
2. Разделение всех найденных патологий в связные компоненты для последующего объединения пересекающихся патологий.
3. Слияние областей внутри каждой найденной связной компоненты и суммирование вероятностей принадлежности к каждому классу патологии, что даст интегральную оценку присутствия каждого класса в объединенной области.

4. Присвоение объединенной области класса с наибольшей суммарной вероятностью на основе найденных вероятностных оценок.

Данный алгоритм корректирует результаты модели, устраняя невозможные с клинической точки зрения ситуации, где разные патологии перекрываются, и обеспечивает более достоверную идентификацию патологий с учетом их пространственного расположения и вероятностных характеристик.

3.2. Объединение фаз контрастирования

На финальной стадии алгоритма происходит объединение результатов сегментации, полученных на двух фазах контрастирования. Цель объединения заключается в нахождении оптимального баланса между двумя противоположными задачами: сохранением всех истинных патологических образований и исключением ошибочных патологий. Эти задачи взаимно противоречат друг другу, поскольку увеличение числа выявленных истинных патологий приводит к росту количества ложных данных и наоборот. Для достижения оптимального результата было рассмотрено три метода объединения фаз:

1. Метод строгой фильтрации: сохраняются только те патологии, которые обнаружены на обеих фазах. Этот метод позволяет эффективно удалять ложные патологии, однако увеличивает риск потери значимых патологических образований, если они выявляются только на одной из фаз.

2. Метод максимизации данных: учитываются все патологии, найденные на обеих фазах. Такой подход гарантирует максимальное сохранение всех истинных патологий, однако приводит к высокому числу ложноположительных результатов.

3. Метод приоритета PV-фазы: сохраняются патологии, которые хотя бы частично обнаружены на PV-фазе. Этот подход основан на предварительных экспериментах, которые показали, что PV-фаза наиболее эффективно выявляет значимые патологические образования.

4. Результаты тестирования

Первые два этапа алгоритма были протестированы для каждого изображения, в котором нейронная сеть допустила ошибку. В результате выполнения этих этапов патологии объемом меньше порогового значения были успешно удалены, а пересекающиеся патологии объединены. В 65 % случаев в ходе применения первых двух этапов алгоритма были устранены избыточные данные. Однако в остальных случаях из-за значительных начальных ошибок модели был допущен неправильный выбор класса объединенной структуры.

Кроме того, были проведены тесты с использованием всех трех шагов алгоритма. При тестировании алгоритма для каждого из методов объединения фаз были вычислены: средний процент изменения объема патологий истинного класса относительно фазы с наибольшим объемом истинных патологий, средний процент изменения объема патологий ложного класса относительно фазы с минимальным объемом ложных патологий, а также изменение доли патологий истинного класса. Результаты тестирования представлены в табл. 1.

На основе полученных результатов можно сделать следующие выводы относительно влияния методов объединения фаз на точность сегментации:

1. Метод строгой фильтрации показал наибольшее улучшение доли патологий истинного класса (+60 %), что свидетельствует о значительном уменьшении числа ложных патологий. Однако увеличение объема патологий истинного класса при объединении двух фаз составило лишь +8 %, что указывает на некоторую потерю истинных патологий при значительном удалении ложных. Этот метод эффективен в задачах, где приоритет отдается исключению ложных патологий, но он может привести к потере части значимых данных.

Таблица 1

Сравнение методов

	Изменение доли патологий истинного класса	Изменение объема патологий истинного класса	Изменение объема патологий ложного класса
Метод строгой фильтрации	+60 %	+8 %	+45
Метод максимизации данных	+8 %	+21 %	+119 %
Метод приоритета PV-фазы	+35 %	+20 %	+84 %

2. Метод максимизации данных показал значительное увеличение объема патологий ложного класса (+119 %), что привело к существенному росту ложноположительных результатов. В то же время, увеличение объема патологий истинного класса составило лишь +21 %, а изменение доли патологий истинного класса оказалось минимальным (+8 %). Этот метод может быть полезен в случаях, когда важно сохранить все обнаруженные патологии, но он приводит к ухудшению качества сегментации за счет увеличения ложных данных.

3. Метод приоритета PV-фазы продемонстрировал сбалансированные результаты: увеличение объема патологий истинного класса составило +20 %, а ложных патологий — +84 %. Изменение доли патологий истинного класса составило +35 %, что свидетельствует о значительном улучшении сегментации при умеренном увеличении ложных данных. Этот метод является наиболее сбалансированным, обеспечивая хороший компромисс между сохранением истинных патологий и минимизацией ложных.

Таким образом, алгоритм постобработки с использованием метода приоритета PV-фазы является оптимальным, так как демонстрирует наилучший баланс между увеличением объема патологий истинного класса и контролем за числом ложных патологий, обеспечивая улучшение отношения между объемами истинных и ложных патологий. Этот метод обеспечивает наиболее точную сегментацию, что делает его предпочтительным для практического применения.

Заключение

В данной работе разработан алгоритм постобработки сегментации патологий на мультифазных КТ-изображениях, который включает удаление шумов, объединение пересекающихся патологий и оптимизации результатов с использованием разных фаз контрастирования. Тестирование показало, что предложенные методы эффективно повышают точность сегментации, обеспечивая баланс между сохранением истинных объектов и минимизацией ложных данных. Этот подход улучшает качество результатов и может быть полезен в клинической практике для автоматизированного анализа медицинских изображений.

Литература

1. *Antonelli M., Reinke A., Bakas S. et al.* The Medical Segmentation Decathlon // Nature Communications. 2022. Т. 13. С. 4128. URL: https://doi.org/10.1038/s41467-022-30695-9 (дата обращения: 25.11.2024).

2. *Simpson A. L., Antonelli M., Bakas S., et al.* A large annotated medical image dataset for the development and evaluation of segmentation algorithms. arXiv.org, 2019. URL: https://doi.org/10.48550/arXiv.1902.09063 (дата обращения: 25.11.2024).

3. *Cao K., Xia Y., Yao J. et al.* Large-scale pancreatic cancer detection via non-contrast CT and deep learning // Nat Med. – 2023. – T. 29. – C. 3033–3043. URL: https://doi.org/10.1038/s41591-023-02640-w (дата обращения: 25.11.2024).

4. *Ma, J., He, Y., Li, F. et al.* Segment anything in medical images. Nat. Commun. 15, 654 (2024). URL: https://doi.org/10.1038/s41467-024-44824-z (дата обращения: 25.11.2024).

УДК 004.932.72.1

ИСПОЛЬЗОВАНИЕ НЕЙРОСЕТИ WHISPER ДЛЯ РАСПОЗНАВАНИЯ МЕДИЦИНСКОЙ ТЕРМИНОЛОГИИ И УЛУЧШЕНИЯ РАННЕЙ МЕДИЦИНСКОЙ ДИАГНОСТИКИ

Д. Е. Черняев, А. А. Ворсунов, Д. В. Лебедев

Воронежский государственный университет

Аннотация. В данной статье рассматривается использование нейросетевой методики автоматического распознавания речи в медицинской диагностике рака молочной железы и для улучшения и ускорения её процессов.
Ключевые слова: автоматическое распознавание речи, медицинская терминология, медицинская диагностика, рак молочной железы.

Введение

Рак молочной железы — одно из самых распространенных и изучаемых злокачественных новообразований, однако он остается серьезной угрозой для здоровья женщин во всем мире. Раннее выявление и точная диагностика играют ключевую роль в улучшении результатов лечения и увеличении выживаемости пациентов. Современные достижения в области обработки естественного языка (NLP) и технологий распознавания голоса позволяют использовать голосовые помощники как перспективные инструменты для содействия медицинским исследованиям и принятия клинических решений [8].

Целью данного исследования является разработка голосового помощника, способного точно распознавать термины, связанные с раком молочной железы, из речевого ввода человека. Такой помощник будет использовать алгоритмы автоматического распознавания речи (ASR) и NLP для анализа произнесенных слов и выделения ключевых терминов, связанных с раком молочной железы, таких как «рак молочной железы», «опухоль груди», «карцинома груди», «HER2», «маммография», «BRCA1/2» и другие.

Голосовой помощник может быть интегрирован с научными базами данных для поиска релевантных научных исследований, используя выделенные ключевые слова. Это обеспечит доступ к актуальной научной литературе, связанной с раком молочной железы, и позволит исследователям, врачам и пациентам эффективно находить и анализировать новейшие данные.

Предоставляя информацию на основе актуальной научной литературы, голосовой помощник может поддерживать принятие клинических решений, основанных на фактических данных, способствовать новым открытиям и ускорять распространение знаний в области лечения и исследования рака молочной железы.

Предлагаемая медицинская система автоматизированной диагностики выполняет несколько этапов обработки медицинских изображений, таких как предварительная обработка, сегментация, функция извлечения признаков и классификация.

Технология автоматического распознавания речи (ASR), преобразующая устную речь в письменный текст, играет важную роль в правильном распознавании химических терминов из человеческой речи. Система Whisper ASR используется для приема ввода от пользователя и предоставления транскрипций на его основе. Транскрипции медицинских терминов из человеческой речи, созданные ASR, могут использоваться для анализа данных и поиска. Точное распознавание медицинских терминов с помощью ASR позволяет эффективно искать, индексировать и извлекать соответствующую информацию из транскрипций, что позволяет исследователям анализировать и извлекать значимые идеи из своих данных.

1. Нейросеть Whisper ASR

Whisper ASR основана на архитектуре глубокого обучения с использованием рекуррентных нейронных сетей (RNN), а точнее их варианта — сети с долговременной краткосрочной памятью (LSTM). Этот тип RNN оснащён специальными механизмами, которые позволяют эффективно учитывать долгосрочные зависимости в последовательных данных. LSTM особенно хорошо справляются с моделированием последовательностей, таких как речевые сигналы, и широко применяются в системах автоматического распознавания речи (ASR). Модель Whisper ASR обучена на обширном наборе многоязычных и многозадачных данных, собранных из Интернета. Благодаря этому она способна преобразовывать устную речь в текст на разных языках и для различных тематических областей. Whisper ASR демонстрирует высокую точность и производительность, что обусловлено использованием большого объёма данных в процессе обучения.

API предоставляет разработчикам удобный интерфейс для интеграции возможностей Whisper ASR в приложения или сервисы. Это позволяет использовать систему для распознавания речи без необходимости самостоятельного обучения или настройки модели.

Предлагаемая система использует возможности Whisper ASR для обработки неструктурированных речевых данных и выделения информации, связанной с диагностикой рака молочной железы. В рамках исследования изучаются методы предварительной обработки данных, представления признаков и конфигурации модели, чтобы оптимизировать её производительность для этой специфической задачи.

На рис. 1 и рис. 2 представлены архитектура нейронной сети и схема её обучения:

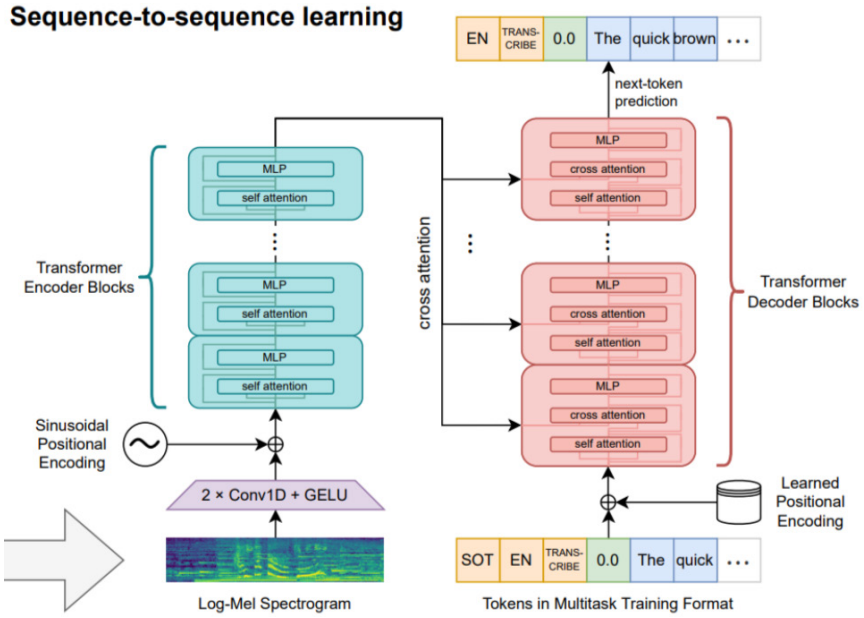


Рис. 1. Архитектура Whisper ASR

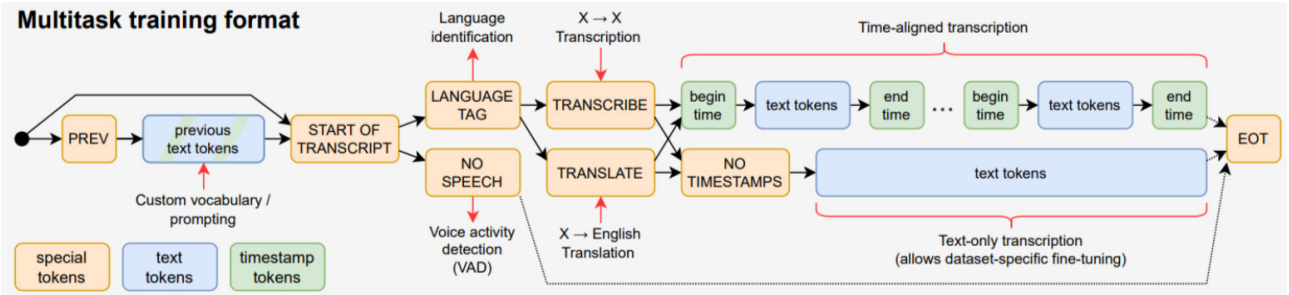


Рис. 2. Схема многозадачного обучения Whisper ASR

2. Обзор методологии и литературы

Разработка интерфейса модуля медицинской информационной системы для преобразования речи в текст обсуждает реализацию используемой оболочки Whisper. Whisper MR Wrapper — это легкий и эффективный программный модуль, который обеспечивает автоматическое распознавание речи (ASR) во встроенных системах с использованием фреймворка MediaRecorder API (MR). Он предоставляет простой интерфейс для захвата звука, распознавания речи.

Оболочка используется в различных приложениях, включая мобильную робототехнику, умные дома, системы роботизированных рук и носимые устройства [1].

В [2], автор обсуждает различные методы распознавания речи. В статье представлен всесторонний обзор традиционных и современных методов, используемых для распознавания речи, включая акустическую фонетику, динамическую временную деформацию, скрытые марковские модели, искусственные нейронные сети и методы глубокого обучения. Автор выделяет преимущества и ограничения каждого метода и приходит к выводу, что подходы, основанные на глубоком обучении, показали многообещающие результаты в повышении точности распознавания речи.

На рис. 3 описывается функционирование системы распознавания речи:

Рис. 3. Процесс работы системы автоматического распознавания речи

Внедрение слабо контролируемого метода для автоматизированного распознавания речи (ASR) вызвало прогресс в распознавании речи за счет разработки неконтролируемых методов предварительного обучения. Подход основан на крупномасштабном наборе данных слабо размеченных аудио- и текстовых данных, которые используются для обучения системы распознавания речи. Показано, что метод достигает самых современных показателей на нескольких тестах ASR, включая чистую и зашумленную речь [3].

Другой эффективный метод использует автоматическое распознавание речи (ASR) и обработку естественного языка (NLP) для транскрипции устной речи в письменный текст и создания результирующего текста.

Предлагаемый подход включает следующие этапы: сегментация аудиозаписи на предложения, преобразование речи в текст с использованием системы автоматического распознавания речи (ASR), а затем применение методов обработки естественного языка (NLP) для формирования итогового текста. Данный подход доказал свою эффективность при создании кратких резюме аудиоконтента. Он находит применение в таких задачах, как ведение заметок, обеспечение доступности информации и управление знаниями [4].

Для преобразования речи в текст также могут использоваться модели гауссовой смеси (GMM). Эти модели подходят для представления нормально распределённых субпопуляций в общей популяции. В контексте распознавания речи GMM применяются для обучения на наборах данных, содержащих речевые сигналы и соответствующие текстовые транскрипции. Процесс [5] включает в себя сегментацию речи на фонемы, оценку вероятности каждой фонемы на основе речевого сигнала и объединение последовательности фонем для формирования итоговой текстовой транскрипции [5].

Общий поток обработки и распознавания речевого сигнала показан на рис. 4.

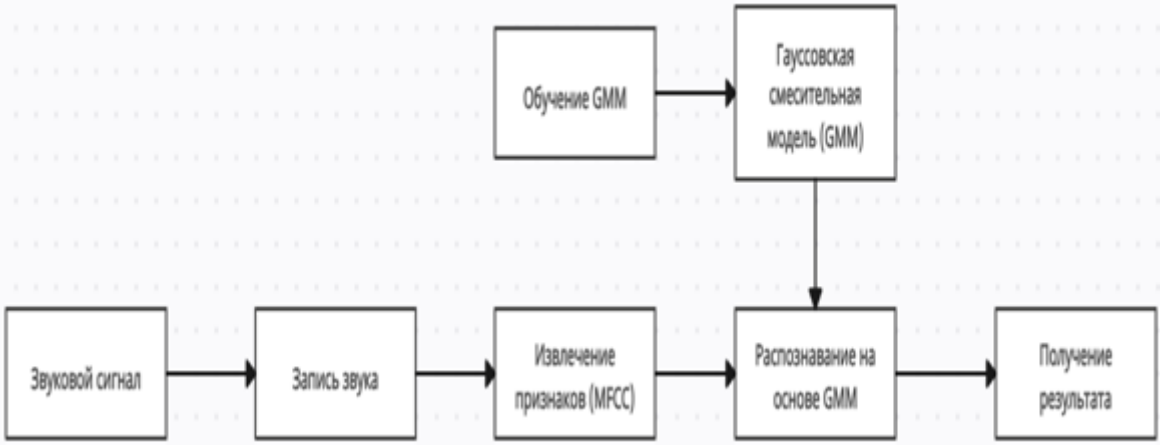


Рис. 4. Обработка и распознавание звука

Системы управления естественным языком для работы с различными акцентами используют алгоритмы распознавания речи, основанные на глубоких нейронных сетях (DNN). Эти алгоритмы повышают производительность систем автоматического распознавания речи (ASR) при обработке речи с различными акцентами. В статье представлена методология, включающая сбор разнообразных данных речевых сигналов и их текстовых транскрипций, а также обучение DNN для улучшения распознавания речи с учётом акцентов [6].

В работе [7], автор анализирует влияние изменчивости речи на системы ASR. Он рассматривает различные подходы к решению этой проблемы, включая нормализацию признаков, адаптацию под говорящего и компенсацию каналов. Обзор подчёркивает важность устранения изменчивости речи для разработки и внедрения высокоточных и надёжных систем ASR, применимых в различных реальных условиях.

3. Методология

Для обеспечения высококачественной записи речевых сигналов осуществляется выбор профессионального микрофона с улучшенными акустическими характеристиками, а его размещение и позиционирование оптимизируются в соответствии с установленными стандартами записи. В данном процессе предпочтение отдаётся направленным микрофонам, которые обладают способностью селективно улавливать звуковую волну, исходящую непосредственно изо рта говорящего, при этом эффективно подавляя фоновый шум и посторонние акустические помехи.

Размещение микрофона определяется с учётом акустических свойств и направленности устройства. Оно устанавливается на определённом расстоянии от источника звука, чтобы избежать эффекта перегрузки сигнала или излишнего затухания. Угол наклона микрофона регулируется таким образом, чтобы максимально точно фиксировать звуковые колебания, обеспечивая при этом минимизацию аэродинамических шумов, возникающих при артикуляции согласных звуков.

Для повышения качества записи могут использоваться дополнительные аксессуары, такие как поп-фильтры, которые эффективно минимизируют воздействие ударных воздушных потоков, возникающих при произношении определённых звуков, и ветрозащитные экраны, снижающие восприимчивость микрофона к шумам, вызванным движением воздуха. Для устранения механических вибраций микрофон размещается на специальных амортизирующих креплениях, которые предотвращают передачу внешних колебаний.

На этапе предварительной обработки аудиоданных применяется кодер модели Whisper — сложная нейронная сеть, способная работать с необработанными звуковыми волнами. Кодер преобразует входной сигнал через несколько слоёв, которые включают сверточные или рекуррентные нейронные сети (RNN). Первый шаг обработки состоит в преобразовании звуковой волны в спектрограмму — графическое представление частотных составляющих сигнала во времени. Этот процесс позволяет выделить ключевые спектральные характеристики аудиосигнала для последующего анализа.

Кодировочная сеть Whisper часто реализована в виде сверточной нейронной сети (CNN) или двунаправленной рекуррентной нейронной сети (BiRNN). Эти архитектуры обеспечивают эффективное извлечение как временных, так и спектральных признаков аудиосигнала. На выходе кодировочная сеть формирует сжатый вектор фиксированной длины, который аккумулирует наиболее значимые характеристики входного аудиосигнала, упрощая дальнейшую обработку и анализ.

Сжатый вектор передаётся в декодер, который отвечает за генерацию выходного речевого сигнала. В модели Whisper декодер, как правило, представлен рекуррентной нейронной сетью (RNN), которая принимает вектор узкого места на входе и преобразует его в спектрограмму. На следующем этапе спектрограмма обрабатывается вокодером — специализированным алгоритмом обработки сигналов, способным преобразовать её обратно в звуковую волну, формируя итоговый речевой сигнал.

Вектор узкого места передаётся в декодер, который отвечает за генерацию выходного речевого сигнала. В модели Whisper декодер, как правило, представлен рекуррентной нейронной сетью (RNN), которая принимает вектор узкого места на входе и преобразует его в спектрограмму. На следующем этапе спектрограмма обрабатывается вокодером — специализированным алгоритмом обработки сигналов, способным преобразовать её обратно в звуковую волну, формируя итоговый речевой сигнал.

Рис. 5. Общий процесс записи речи

Далее результаты преобразования речи в текст проходят этап нормализации, целью которого является удаление стоп-слов, знаков препинания и других нетекстовых элементов. Этот процесс включает преобразование текста в нижний регистр, удаление цифр и специальных символов, а также разбиение текста на отдельные слова с помощью токенизации.

После этого происходит извлечение ключевых слов (терминов). Это можно сделать двумя способами: вручную, посредством создания словаря химической терминологии на основе опубликованных материалов, или автоматически, с использованием программного обеспечения. После предварительной обработки текст сопоставляется с указанным словарём для выявления химических сущностей. Все слова и фразы, соответствующие терминам из словаря, извлекаются и идентифицируются как химические сущности.

После извлечения ключевых слов они используются для поиска соответствующих статей в базе данных медицинской терминологии. Это включает в себя написание скрипта, который отправляет запрос в базу данных, извлекает результаты поиска и извлекает соответствующую информацию.

4. Результаты

Была разработана система искусственного интеллекта (ИИ), интегрируемая в медицинские информационные системы (МИС), которая значительно упрощает процесс взаимодействия с пациентами и обеспечивает глубокий анализ медицинских данных. Эта система направлена на автоматизацию рутинных задач, ускорение документооборота и повышение точности диагностического процесса.

В основе системы лежит нейросетевая модель, которая преобразует голосовые данные в текст, выделяет ключевую информацию и структурирует её для последующего использования. Общий план работы данной модели отображает рис. 6.

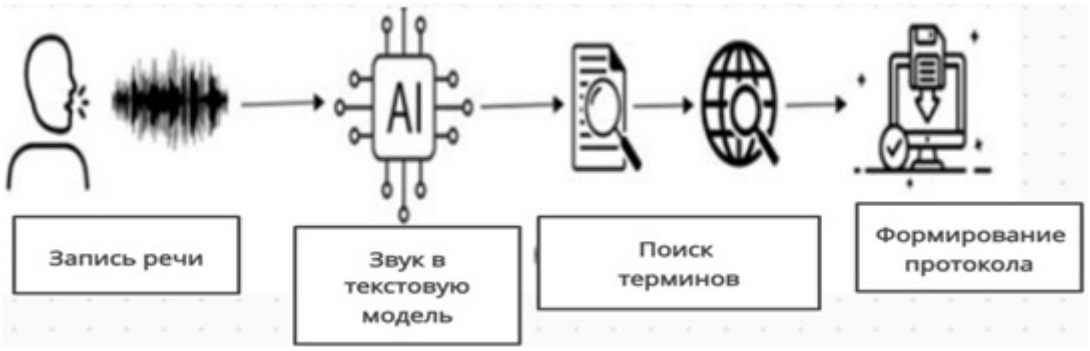


Рис. 6. Процесс работы реализованной системы

Основные этапы работы системы:

• Распознавание речи: аудиозапись, активируемая врачом, отправляется на сервер ИИ, где происходит её обработка и преобразование в текстовый формат.

• Анализ текста: нейросеть выделяет ключевые медицинские термины, такие как названия заболеваний, симптомы, лекарства и анализы, структурируя информацию в удобном для использования формате.

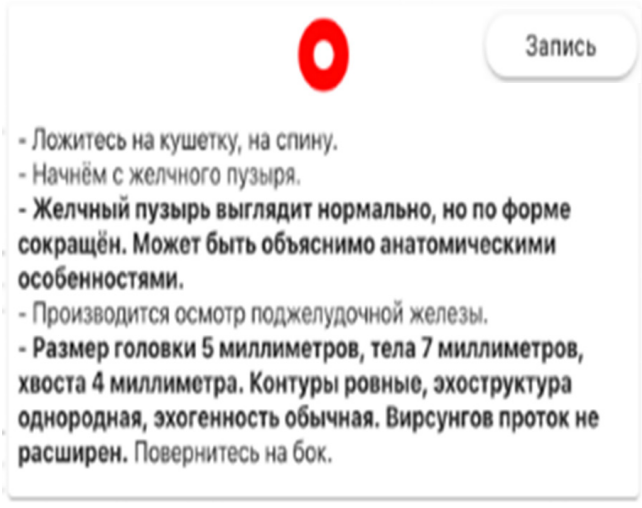


Рис. 7. Пример интерфейса записанного диалога

• После обработки текст передаётся клиентскому приложению, где реализованы следующие функции:

• Вывод текста на экран: врач видит полный текст диалога с пациентом, где ключевые слова и термины автоматически подсвечиваются, что упрощает восприятие данных (рис. 7).

• Автозаполнение медицинских протоколов: система автоматически заполняет шаблоны медицинских протоколов на основе структурированных данных, позволяя врачу при необходимости вносить коррективы (рис. 8).

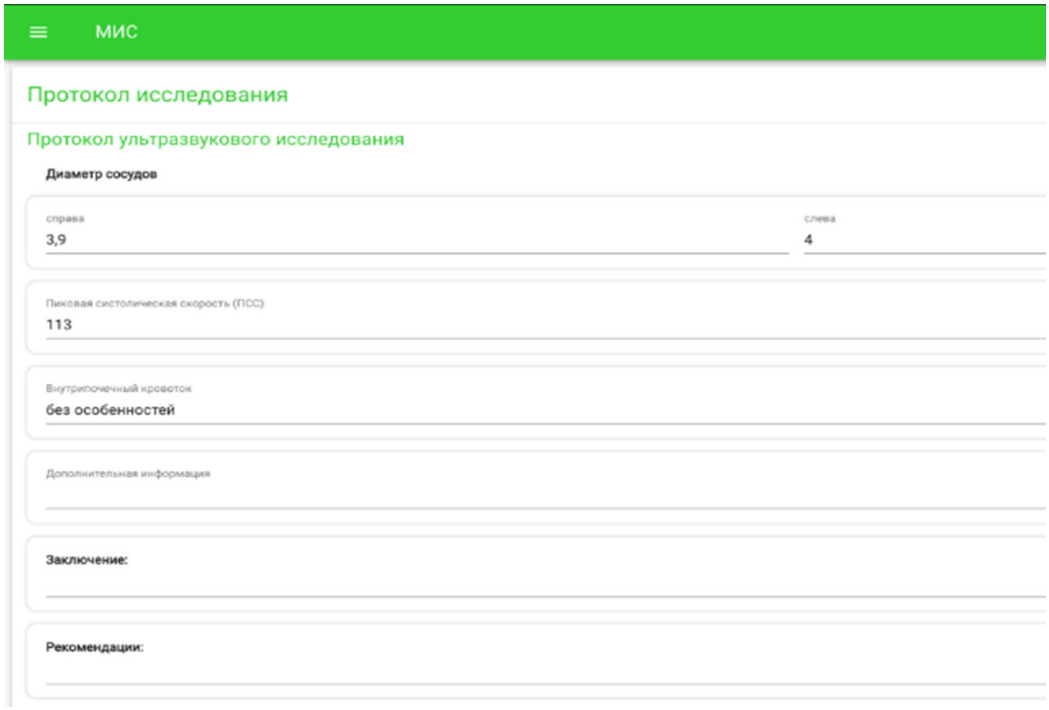


Рис. 8. Пример заполненного протокола

Основными преимуществами системы являются:

• Скорость: автоматизация позволяет сократить время на заполнение документации, освобождая больше времени для общения с пациентами.

• Качество диагностики: подсветка важных медицинских терминов помогает врачу фокусироваться на ключевых аспектах при анализе данных.

• Интеграция с МИС: система бесшовно подключается к существующим МИС без значительных изменений в инфраструктуре.

Система построена на современной модульной архитектуре, включающей:

• Основной сервер МИС: управляет пользователями, хранит и обрабатывает данные пациентов.

• Сервер ИИ: обрабатывает аудиоданные, выполняет распознавание речи и анализ текста.

• Клиентскую часть: отображает текст, обеспечивает подсветку терминов и заполняет протоколы.

Обмен данными между клиентом и сервером ИИ осуществляется через WebSocket, что обеспечивает высокую скорость обработки и минимальные задержки.

Технологический стек, используемый для создания системы:

• React: используется для создания динамичного и удобного интерфейса.

• Material-UI: обеспечивает современный дизайн и удобство работы с приложением.

• @wmik/use-media-recorder: Предназначен для записи аудиоданных.

• jszip: применяется для сжатия и передачи аудиофайлов, что оптимизирует использование сети.

На серверной части используется специализированная нейросетевая модель, обученная на медицинских данных. Она обеспечивает точное преобразование речи и выделение значимых медицинских сущностей.

Система функционирует в защищённой телекоммуникационной сети (ЗТКИ), что исключает вероятность внешних атак. Для авторизации используется стандартная схема логин/пароль (рис. 9). Такой уровень защиты является достаточным для работы внутри закрытой инфраструктуры.

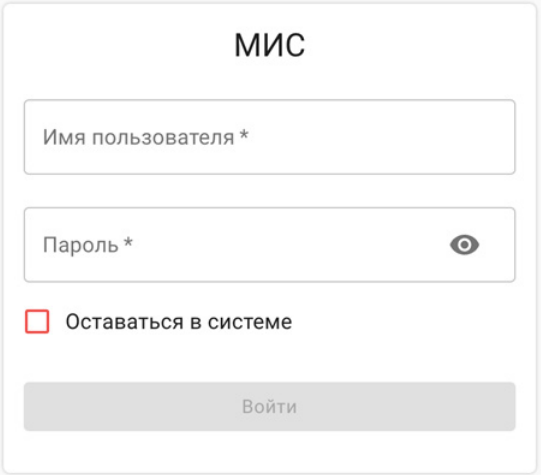


Рис. 9. Окно авторизации МИС

Заключение

В этой статье рассмотрены возможности применения современных технологий автоматического распознавания речи и обработки естественного языка (NLP) для разработки голосового помощника, способного распознавать ключевые термины, связанные с раком молочной железы. Было проанализировано использование таких инструментов для поддержки клинических решений, поиска релевантной научной литературы и повышения доступности знаний в области онкологи, а также показана реализация взаимодействия данного нейросетевого метода с МИС.

Современные достижения в области искусственного интеллекта и обработки естественного языка открыли возможность разработки сложных моделей, способных эффективно идентифицировать термины и концепции, связанные с онкологией, на основе аудиоданных. Обучение таких моделей проводится на обширных массивах данных, включающих лекции и научные статьи по онкологии, что позволяет им усваивать наиболее распространённую терминологию, ключевые идеи и фразы, характерные для этой области. После завершения обучения эти модели могут быть применены для автоматической транскрипции разговоров, извлечения ключевых понятий и терминов, связанных с раком, а также для генерации письменных резюме или научных текстов.

В перспективе использование нейросетевых технологий для выявления онкологических речевых оборотов с помощью синтеза речи моделей обещает значительный прогресс. Развитие технологий ИИ и NLP, вероятно, приведёт к созданию более сложных и точных моделей, которые смогут значительно облегчить работу исследователей и врачей в области онкологии.

Литература

1. *Ramírez-Duque A. A.* A Whisper ROS Wrapper to Enable Automatic Speech Recognition in Embedded Systems / A. A. Ramírez-Duque, M. E. Foster // HRCI 2023. – 2023.

2. *Saksamudre S. K.* A Review on Different Approaches for Speech Recognition System / S. K. Saksamudre, P. P. Shrishrima, R. R. Deshmukh // International Journal of Computer Applications. – 2015.

3. *Radford A.* Robust Speech Recognition via Large-Scale Weak Supervision / A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, I. Sutskever // IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) – 2019. – P. 5961–5965. DOI: 10.1109/ICASSP.2019.8682574.

4. *Vinnarasu A.* Speech to Text Conversion and Summarization for Effective Understanding and Documentation / A. Vinnarasu, D. V. Jose // International Journal of Electrical and Computer Engineering (IJECE). – 2019.

5. *Chauhan V.* Speech to Text Converter Using Gaussian Mixture Model / V. Chauhan, S. Dwivedi, P. Karale, S. M. Potdar // International Research Journal of Engineering and Technology (IRJET). – 2016.

6. *Gurtueva I.* Speech Recognition Algorithm for Natural Language Management Systems Under Variety of Accents / I. Gurtueva, O. Nagoeva, I. Pshenokova // E3S Web of Conferences. – 2020.

7. *Бензегиба М.* Автоматическое распознавание речи и речевая изменчивость: обзор / М. Бензегиба, Р. Де Мори, Ф. Дюфур, П. Дж. Годфри // Speech Communication – 2014. – Т. 57. – С. 109–129.

8. *Черняев Д. Е.* Диагностика рака груди при помощи системы поддержки принятия врачебных решений / Д. Е. Черняев, В. М. Павлова // Актуальные проблемы прикладной математики, информатики и механики: Сборник трудов Международной научной конференции, Воронеж – 2024. – С. 1079–1085. URL: https://elibrary.ru/item.asp?id=66563100 (дата обращения: 01.01.2024).

9. *Faris W. F.* Cataract Eye Detection Using Deep Learning Based Feature Extraction with Classification // Research Journal of Computer Systems and Engineering. – 2020. – 1(2). – P. 20–25.

10. *Ajami S.* Use of speech-to-text technology for documentation by healthcare providers // Int. J. Healthc. Technol. Manag. – 2016. – 15(1). – P. 23–32.

11. *Cutajar M., Gatt E., Grech I., Casha O., Micallef J.* Comparative Study of Automatic Speech Recognition techniques // IET Signal Processing. – 2013. – 7(1).– P. 1–8.

УДК 004.912

СРАВНИТЕЛЬНЫЙ АНАЛИЗ АРХИТЕКТУР НЕЙРОННЫХ СЕТЕЙ ДЛЯ КЛАССИФИКАЦИИ НОВОСТНЫХ СТАТЕЙ

В. И. Шиян, М. Е. Кокоровец, Ю. Д. Кривошей

Кубанский государственный университет

Аннотация. В данной работе проведено сравнительное исследование различных архитектур нейронных сетей, применяемых для задачи классификации новостных статей по тематикам. Рассмотрены одномерные свёрточные нейронные сети (1D-CNN), рекуррентные сети на основе LSTM и GRU. Представлены детальные описания структур данных моделей, а также результаты экспериментальной оценки их эффективности на наборе данных с новостными текстами. Полученные результаты демонстрируют различия в способности моделей к обобщению и выявляют преимущества рекуррентных архитектур в задачах классификации текстовых данных.
Ключевые слова: классификация новостных статей, нейронные сети, свёрточные нейронные сети, LSTM, GRU, сравнительный анализ, обобщающая способность.

Введение

С одной стороны, задача классификации новостных статей по тематикам является важной для автоматизации процессов обработки и анализа больших объёмов текстовой информации. Эффективное решение этой задачи позволяет организовывать и структурировать новостной контент, облегчая его поиск и навигацию [1]. С другой стороны, разработка моделей машинного обучения, способных точно и надёжно классифицировать новостные тексты, представляет собой сложную научно-техническую проблему, требующую тщательного изучения различных архитектур нейронных сетей и их сравнительной оценки [2, 3].

1. Применение нейронных сетей для классификации новостных статей

Для решения задачи классификации новостных статей широко применяются методы машинного обучения, в частности, различные архитектуры нейронных сетей [4, 5]. Одномерные свёрточные нейронные сети способны эффективно извлекать локальные признаки из текста, что позволяет им успешно справляться с задачей классификации [6]. Рекуррентные нейронные сети, такие как LSTM и GRU, демонстрируют высокую эффективность в моделировании последовательной структуры текста, что также делает их подходящим выбором для классификации новостных статей [7, 8].

Переходя к более детальному рассмотрению архитектур, следует отметить, что одномерные свёрточные нейронные сети (1D-CNN) представляют собой мощный инструмент для извлечения локальных признаков из текстовых последовательностей. Эти сети используют свёрточные операции для выявления локальных паттернов, которые могут быть полезными для классификации текстов. Например, 1D-CNN способны эффективно обнаруживать ключевые фразы или сочетания слов, которые имеют высокую значимость для определения тематики статьи. Благодаря своей архитектуре они могут параллельно обрабатывать данные, что обеспечивает высокую вычислительную эффективность по сравнению с рекуррентными подходами. Кроме того, использование пулинговых операций позволяет агрегировать информацию из различных частей текста, что помогает модели сосредотачиваться на наиболее значимых признаках.

Рекуррентные нейронные сети (RNN), включая их усовершенствованные варианты, такие как LSTM (Long Short-Term Memory) и GRU (Gated Recurrent Unit), демонстрируют высокую эффективность при работе с последовательными данными. Эти архитектуры разработаны для учёта контекста в последовательностях, что делает их особенно полезными для анализа текстов. LSTM и GRU способны запоминать долгосрочные зависимости в тексте благодаря механизмам управления памятью и забыванием. Например, при анализе новостного текста модель может учитывать не только локальные признаки, но и глобальный контекст, такой как общий тон статьи или её тематическая направленность. Это позволяет рекуррентным сетям более точно классифицировать тексты, особенно если они содержат сложную структуру или выраженные взаимосвязи между словами.

Для успешного применения нейронных сетей в задаче классификации новостных статей важным этапом является предварительная обработка текстовых данных. В данной работе использовалась токенизация текста, построение словаря наиболее частотных слов и преобразование текстов в числовые представления, пригодные для работы с нейронными сетями. Эти шаги обеспечивают эффективное кодирование текстовой информации, что позволяет моделям извлекать значимые признаки для обучения.

Обучение нейронных сетей проводилось на наборах данных, содержащих новостные тексты с метками классов, соответствующих различным тематикам. В процессе обучения модели оптимизировались для минимизации ошибки классификации на обучающей выборке [9]. Такой подход позволил модели научиться точно предсказывать классы новостных текстов, основываясь на их содержании. Благодаря этому нейронные сети демонстрируют высокую точность в задаче автоматической классификации новостных статей по тематикам [10].

2. Свёрточная нейронная сеть

Свёрточная нейронная сеть представляет собой архитектуру, ориентированную на обработку данных с локальными пространственными связями. На вход подаются данные в виде последовательностей, которые проходят через слой вложений (Embedding), преобразующий дискретные индексы входных данных в плотные векторы фиксированной размерности. Этот слой создаёт компактное представление входной информации, которое затем передаётся в свёрточный слой (Conv1D). Свёрточный слой выполняет фильтрацию данных с использованием набора обучаемых фильтров, которые сканируют последовательность с фиксированным размером окна. В данном случае фильтры имеют размер 5 и производят 250 выходных каналов. Результаты свёрточной операции акцентируют внимание на локальных паттернах в данных, таких как сочетания признаков, релевантные для задачи классификации.

После свёртки применяется глобальный слой максимального объединения (GlobalMaxPooling1D), который извлекает наиболее значимые признаки из каждого канала, снижая размерность данных и сохраняя наиболее информативные компоненты. Далее сеть содержит плотный слой (Dense) с 128 нейронами, который выполняет нелинейное преобразование данных и способствует выявлению сложных зависимостей между признаками. Завершающий плотный слой с 4 нейронами обеспечивает классификацию на 4 класса, соответствующие целевой задаче. Общая структура сети позволяет эффективно извлекать локальные признаки из последовательностей и использовать их для решения задач классификации.

3. Рекуррентная нейронная сеть на основе LSTM

Рекуррентная нейронная сеть на основе LSTM предназначена для обработки последовательных данных с учётом их временной структуры. Входные данные также проходят через слой вложений (Embedding), преобразующий дискретные представления в плотные векторы

размерности 32. Затем данные поступают в слой LSTM, который обладает способностью сохранять долгосрочные зависимости благодаря наличию управляющих механизмов: входного, выходного и забывающего «вентилей». Эти механизмы позволяют модели избирательно запоминать или забывать информацию, что особенно важно для работы с длинными последовательностями. В данном случае слой LSTM имеет 16 выходных нейронов, что обеспечивает компактное представление временных зависимостей в данных. На выходе LSTM-слоя информация передаётся в плотный слой (Dense) с 4 нейронами, который выполняет классификацию на основе извлечённых временных признаков.

4. Рекуррентная нейронная сеть на основе GRU

Рекуррентная нейронная сеть на основе GRU является упрощённой версией LSTM, сохраняя при этом способность учитывать временные зависимости в данных. Как и в предыдущих архитектурах, входные данные проходят через слой вложений (Embedding), преобразующий их в плотные векторы. Затем данные поступают в GRU-слой, который использует два управляющих механизма: обновляющий и сбрасывающий «вентили». Эти механизмы регулируют поток информации через сеть, позволяя модели запоминать или игнорировать определённые временные зависимости. GRU-слой имеет 16 выходных нейронов, что обеспечивает компактное представление последовательности. Завершающий плотный слой (Dense) с 4 нейронами выполняет классификацию на основе временных признаков, извлечённых GRU-слоем.

5. Результаты экспериментов

В ходе исследования были проведены эксперименты по обучению и оценке трёх архитектур нейронных сетей: свёрточной нейронной сети (CNN), сети с долговременной краткосрочной памятью (LSTM) и сети на основе механизмов обновления (GRU). Основной целью экспериментов являлось сравнение точности моделей как на обучающем, так и на проверочном наборах данных. Все модели обучались на протяжении пяти эпох, и их эффективность оценивалась на каждом этапе обучения.

На графике обучения свёрточной нейронной сети (рис. 1) наблюдается стабильное увеличение точности на обучающем наборе данных в течение всех пяти эпох. Начальная точность составила 84,02 %, а к пятой эпохе она достигла 96,93 %. Однако точность на проверочном наборе демонстрирует иную динамику: после первоначального роста (88,42 % на первой эпохе), начиная со второй эпохи, точность постепенно снижается, достигнув 87,29 % к пятой эпохе. Это может свидетельствовать о переобучении модели.

График обучения LSTM-сети (рис. 2) показывает аналогичную тенденцию роста точности на обучающем наборе данных, которая увеличилась с 82,56 % до 93,69 %. Однако точность на проверочном наборе остаётся относительно стабильной в диапазоне от 88,15 % до 88,79 %. Это свидетельствует о лучшей способности LSTM-сети к обобщению по сравнению с CNN.

На графике обучения GRU-сети (рис. 3) также виден рост точности на обучающем наборе данных: с 80,03 % до 93,22 %. Однако, как и в случае с LSTM, точность на проверочном наборе остаётся практически неизменной, колеблясь между 87,79 % и 88,21 %. Это указывает на схожее поведение GRU и LSTM в контексте обобщающей способности.

Для оценки итоговой эффективности моделей на проверочном наборе данных был построен график сравнения точности всех трёх архитектур (рис. 4). На нём видно, что свёрточная нейронная сеть достигает максимальной точности на проверочном наборе (88,42 %) уже на первой эпохе, однако её эффективность снижается в последующих эпохах. В то же время LSTM и GRU демонстрируют более стабильные результаты, достигая близких значений точности (около 88 %) без значительных отклонений.

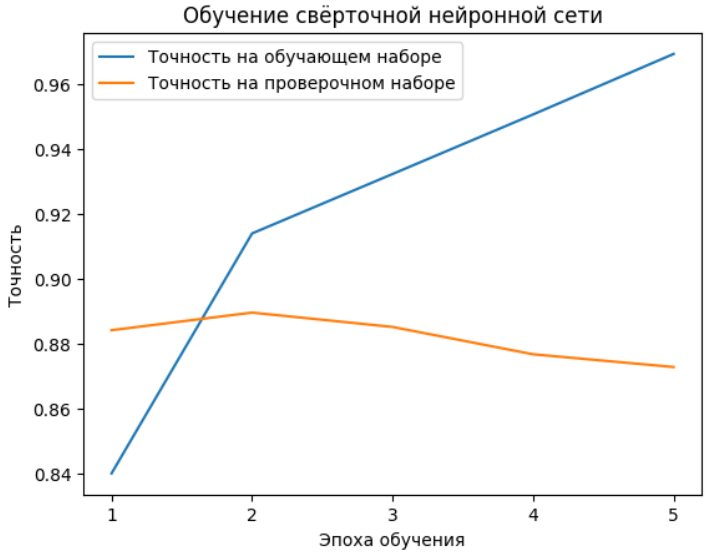


Рис. 1. График обучения свёрточной нейронной сети

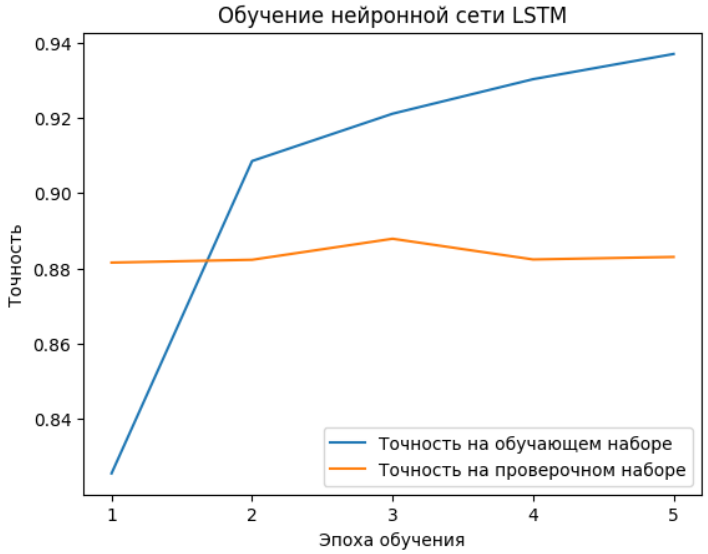


Рис. 2. График обучения нейронной сети LSTM

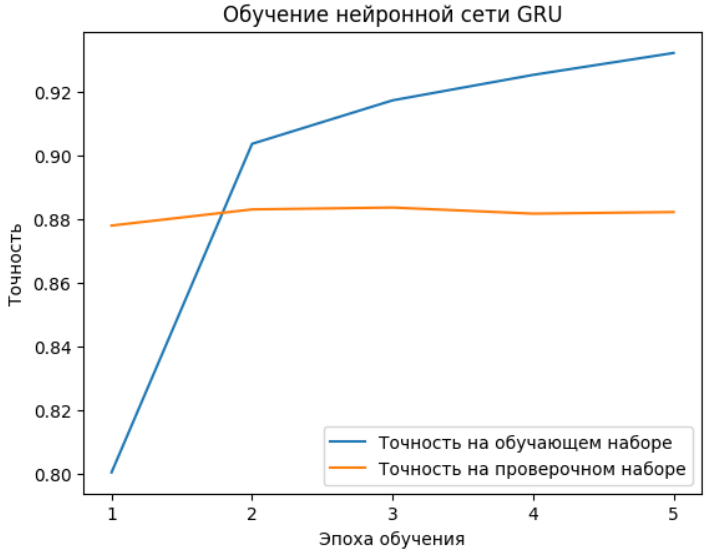


Рис. 3. График обучения нейронной сети GRU

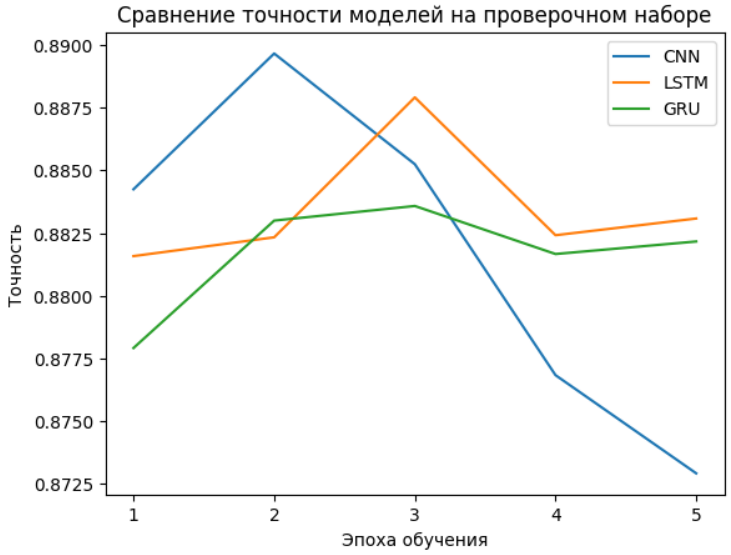


Рис. 4. Сравнение точности моделей на проверочном наборе

На финальном этапе экспериментов была рассчитана итоговая точность каждой модели после завершения всех эпох обучения (рис. 5). Наилучшая итоговая точность была достигнута LSTM-сетью (88,31 %), за которой следует GRU-сеть (88,22 %). Свёрточная нейронная сеть показала несколько худший результат (87,29 %), что подтверждает её склонность к переобучению при увеличении числа эпох.

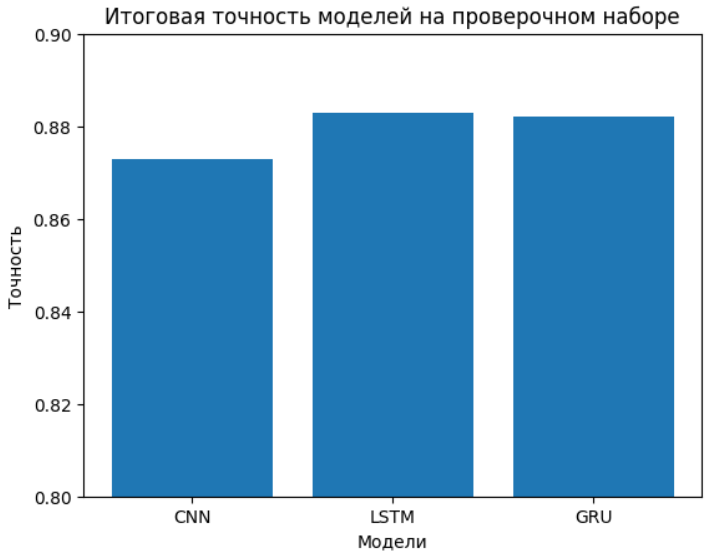


Рис. 5. Итоговая точность моделей на проверочном наборе

Результаты экспериментов демонстрируют различия в поведении изученных архитектур нейронных сетей. Свёрточная нейронная сеть показала высокую точность на обучающем наборе данных, но её способность к обобщению оказалась ниже из-за переобучения. Напротив, LSTM и GRU продемонстрировали более стабильные результаты на проверочном наборе данных, что делает их предпочтительными для задач, требующих высокой обобщающей способности.

Заключение

В заключение следует отметить, что проведённое исследование продемонстрировало различия в способности рассмотренных архитектур нейронных сетей к обобщению на задаче

классификации новостных статей. Свёрточная нейронная сеть показала высокую точность на обучающем наборе данных, но её эффективность снизилась на проверочном наборе, что указывает на склонность к переобучению. В то же время рекуррентные сети на основе LSTM и GRU продемонстрировали более стабильные результаты, сохраняя высокую точность на проверочном наборе данных. Эти результаты свидетельствуют о преимуществах рекуррентных архитектур в задачах, требующих высокой обобщающей способности, таких как классификация новостных текстов. Полученные выводы могут быть полезны при выборе наиболее подходящих моделей для решения аналогичных задач обработки и анализа текстовых данных.

Литература

1. *Айвазян С. А., Бухштабер В. М., Енюков И. С., Мешалкин Л. Д.* Прикладная статистика: классификация и снижение размерности. – М. : Финансы и статистика, 1989. – 607 с.

2. *Вапник В. Н.* Восстановление зависимостей по эмпирическим данным. – М.: Наука, 1979. – 334 с.

3. *Журавлев Ю. И., Рязанов В. В., Сенько О. В.* «Распознавание». Математические методы. Программная система. Практические применения. – М. : Фазис, 2006. – 416 с.

4. *Загоруйко Н. Г.* Прикладные методы анализа данных и знаний. – Новосибирск : ИМ СО РАН, 1999. – 272 с.

5. *Шлезингер М., Главач В.* Десять лекций по статистическому и структурному распознаванию. – Киев : Наукова думка, 2004. – 324 с.

6. *Hastie T., Tibshirani R., Friedman J.* The Elements of Statistical Learning: Data Mining, Inference, and Prediction. – New York : Springer, 2001. – 533 p.

7. *Mitchell T.* Machine Learning. – New York : McGraw-Hill Science/Engineering/Math, 1997. – 414 p.

8. *Николенко С., Кадурин А., Архангельская Е.* Глубокое обучение. – СПб. : Питер, 2018. – 480 с.

9. *Осовский С.* Нейронные сети для обработки информации. – М. : Финансы и статистика, 2004. – 344 с.

10. *Хайкин С.* Нейронные сети: полный курс. 2-е изд. – М. : Вильямс, 2006. – 1104 с.

УДК 004.8

ИСПОЛЬЗОВАНИЕ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ ДЛЯ ОЦЕНКИ ОТЗЫВОВ НА ПРОГРАММНЫЕ ПРОДУКТЫ

Н. А. Экерт, И. Е. Воронина

Воронежский государственный университет

Аннотация. Рассматривается актуальная проблема регулировки входных параметров алгоритма для разделения корпуса текста на кластеры и выдачи наименований для выявленных кластеров. В качестве предмета исследования выбраны большие языковые модели (Large Language Model), используемые для задач краткого изложения (суммаризации) и извлечения смыслов из текста. Приводятся результаты исследования по оценке применимости наиболее популярных моделей к обозначенной задаче. Предлагаются два подхода по использованию больших языковых моделей для задачи оценки отзывов на программные продукты. Новизна результата заключается в доработке существующего алгоритма оценки обратной связи пользователей.
Ключевые слова: большие языковые модели, Large Language Model, LLM, оценка отзывов, Llama, GPT, Claude, Gemini, GigaChat, Phi3, Gemma, Mistral, сравнение языковых моделей, суммаризация текстов, извлечение смыслов, кластеризация.

Введение

Современный рынок программного обеспечения характеризуется высокой конкуренцией и постоянным стремлением к повышению качества с целью удовлетворения потребностям пользователя. Одним из ключевых критериев успешности программного продукта является его способность соответствовать ожиданиям клиента. К счастью, клиенты сами способны сообщить разработчикам о своих потребностях. Они могут предоставлять обратную связь в виде отзывов, которые в дальнейшем могут быть проанализированы отделом качества.

С ростом популярности приложения, отзывов может становиться все больше, а их анализ может потребовать более детальной проработки. Количество факторов (критериев), которые необходимо учитывать при оценке отзывов, растет вместе с пользовательской базой и функциональными возможностями программного продукта. Качество оценки обратной связи напрямую влияет на наполнение дорожной карты с задачами, а значит, на вносимые изменения в конечный продукт.

Очевидна потребность в автоматизации процесса выявления отклонений в работе программного обеспечения путем оценки отзывов, предоставляемых пользователями. Задача является объемной и нетривиальной. Для оценки отзывов было разработано несколько подходов и алгоритмов. При этом, выявлена потребность в привлечении экспертов для наименования выявляемых алгоритмом кластеров и регулировки входных параметров.

Именно в решении данной задачи использование больших языковых моделей становится особенно актуальным. Большие языковые модели (Large Language Model) способны понимать контекст, выявлять ключевые идеи и генерировать сжатые, информативные отрезки текста. В исследовании планируется произвести анализ и сравнение существующих больших языковых моделей в задачах краткого изложения (суммаризации) и извлечения смыслов из текста.

1. Предварительная обработка данных

Формат обратной связи пользователей — это отзывы, представленные в электронной текстовой форме на русском языке. Допускаются вставки английских слов. Исключается исполь-

зование ненормативной лексики. Отзывы представлены в виде набора данных, который содержит оценки всех доступных версий программного продукта.

Для корректной работы с неструктурированными текстовыми данными необходимо выполнить их предварительную обработку. Текст каждого отзыва разделяется на предложения, каждое из которых проходит процедуру токенизации (разделения на токены или слова) и лемматизации (приведения словоформы к начальной форме).

Для решения проблемы нахождения опечаток и орфографических ошибок в тексте был разработан контекстно-независимый алгоритм выбора варианта при нечетком сравнении строк [1]. Суть его работы заключается в том, что для слова составляется матрица символов-заменителей и каждому из этих символов ставится в соответствие весовой коэффициент из словаря. Весовой коэффициент для каждого слова-заменителя рассчитывается с учетом редакционного расстояния (в данном случае использовалось редакционное расстояние Левенштейна). Результатом работы алгоритма является подстановка на место слова с неверным написанием слова-заменителя с наибольшим коэффициентом уверенности. Таким образом, слово, содержащее орфографическую ошибку или опечатку, заменяется на наиболее подходящее слово из словаря.

Данные были переведены в формат, пригодный для построения векторов слов: предложения разделены на массивы, содержащие отдельные слова, и объединены в один большой массив. Для каждого слова было составлено его векторное представление с помощью обучения модели Word2Vec.

Для корректной интерпретации отзывов на программный продукт был разработан алгоритм формирования критериев оценивания [2]. В основе его идеи лежит принцип разделения корпуса текста на соответствующие каждому критерию кластеры. Разделение на кластеры осуществляется с помощью создания графового представления всего корпуса. Вершинами в таком графе являются слова, ребрами — отношения между словами, весовым коэффициентом ребер — значение сходства слов в векторном представлении. Рёбра, чьё значение весового коэффициента выходило за рамки выбранных значений, удалялись из графа.

Для разделения графа на кластеры был выбран алгоритм поиска слабо связанных компонентов, так как он позволяет разделить граф на подграфы, между вершинами которых существует хотя бы одна связь.

С помощью данного способа удалось выявить неочевидные критерии оценивания программного обеспечения, характеризующие потребности пользователя.

Важно заметить, что наименование кластеров производилось с помощью привлечения экспертов, что является не эффективным способом, особенно, если таких кластеров большое количество.

Также, в процессе эксперимента, необходимо было эмпирическим методом подбирать минимальное значение сходства слов для разделения графа на тематические кластеры. При такой настройке необходимо производить множество итераций для поиска оптимальных значений. Тематики разных кластеров не должны накладываться друг на друга. В то же время, нужно найти максимальное количество кластеров для учитывания разнообразных критериев.

а) *б)* *в)*

а) группа отзывов с оценкой стабильности работы приложения; б) группа отзывов с оценкой удобства использования приложения; в) группа отзывов с оценкой читаемости текста приложения

Рис. 1. Результат разделения корпуса текста на тематические кластеры

2. Использование больших языковых моделей

Учитывая наличие нечетких критериев оценивания при формировании кластеров, а также взрывной рост использования больших языковых моделей для решения задач, связанных с распознаванием естественного языка и анализом текстовых данных, можно сделать эксперимент по использованию большой языковой модели для решения проблемы оценки отзывов на программные продукты.

Large Language Model (LLM) (рус. Большая языковая модель) — это модель искусственной нейронной сети, обученная на большом наборе данных, предназначенная для обработки естественного языка. Такие модели способны распознавать закономерности и взаимосвязи между словами, фразами и идеями в языке.

LLM используются для задач:

- автоматизированной генерации текста на основе заданной темы, стиля или формата;
- перевода текстов с одного языка на другой;
- суммаризации или краткого изложения длинного текста;
- анализа эмоциональной окраски текста;
- классификации текста.

Языковые модели обладают контекстом. Это означает, что они могут формировать ответ на основе переданных данных: базы знаний и установок оператора. Размер контекста измеряется в токенах и в разных моделях может варьироваться от тысяч до десятков тысяч токенов [3, 4].

Токен (token) — основная единица текстовой информации, используемая для представления текста в виде, пригодном для обработки моделью. Это может быть слово, символ или предложение. Процесс токенизации позволяет модели обрабатывать текстовую информацию на гранулярном уровне, что значительно повышает ее гибкость.

Был проведен анализ наиболее известных на сегодняшний день языковых моделей, данные приведены в табл. 1.

OpenAI предлагает выбор из четырех моделей, среди которых GPT4 Omni является наиболее современной [5]. Она предназначена для решения сложных задач по анализу и обработке текстов, поддерживает вставку дополнений в текст, может использоваться для перевода, сложной классификации, анализа и резюмирования текстов. Данная модель нацелена на снижение случаев «лени», когда модель не выполняет поставленную задачу в полной мере. Помимо этого, она имеет возможность получать данные из Интернета, что позволяет актуализировать ту или иную предметную область для генерации наиболее релевантных ответов.

Таблица 1

Сравнение характеристик больших языковых моделей

Семейство	Модель	Кол-во токенов	Обучающие данные	Количество параметров	Использование
GPT4	Omni	131,072	до 12.2023 + Интернет	Не указано	API + VPN
Claude 3.5	Sonnet	204,800	Не указано	Не указано	API + VPN
Llama	3.1:405b	131,072	Не указано	405b	Local
Llama	3.1:70b	131,072	Не указано	70b	Local
Llama	3.1:8b	131,072	Не указано	8b	Local
Gemini	1.5 Pro	1,024,000	Не указано	Не указано	API + VPN
GigaChat	Pro	8,192	Не указано	29b	API
GigaChat	Lite	8,192	Не указано	7b	API
Phi 3	Medium	131,072	Не указано	14b	Local
Gemma	2	8,192	Не указано	27b	Local
Mistral	Nemo	131,072	Не указано	12b	Local

Модели Claude 3.5 Sonnet от компании Anthropic и Gemini 1.5 Pro от Google являются прямым конкурентом GPT4 Omni [6, 7]. Они могут выполнять те же манипуляции с текстовыми данными: перевод, суммаризацию, дополнение, исправление ошибок, классификацию и многое другое. У этих моделей отсутствует функция поиска в Интернете, но можно передавать им необходимые для обучения данные в контекст, как и любым большим языковым моделям, представленным в статье.

Llama 3.1 от компании Facebook выделяется на фоне других т. к. ее веса можно загрузить на компьютер и использовать модель локально [8]. Такой подход открывает значительно больше возможностей для проведения экспериментов. По функциональным возможностям она не уступает своим конкурентам от OpenAI, Anthropic и Google.

Отечественный GigaChat предлагает несколько моделей [9]. Базовая версия Lite подходит для решения тривиальных задач, требующих быстрой отзывчивости при работе. GigaChat Pro лучше следует сложным инструкциям и может выполнять более комплексные задачи: суммаризацию, рерайтинг и редактирование текстов, ответы на различные вопросы.

Примечательны также языковые модели Phi 3 Medium от Microsoft, Gemma 2 от Google и Mistral Nemo. По аналогии с Llama 3.1, они разворачиваются локально, имеют очень высокую скорость ответа и следуют инструкциям из контекста.

3. Алгоритм работы

Предлагается сравнить два подхода:

1. С применением большой языковой модели для задачи наименования кластеров и определения пороговых значений при разделении графа на кластеры.

2. С применением большой языковой модели в качестве экспертной системы, способной самостоятельно провести анализ отзывов и выявить приоритетные зоны развития программного продукта.

В обоих случаях предлагается использовать RAG-подход для дообучения модели [10]. Этот подход предполагает использование контекста модели для указания правил формирования ответа и дополнительной информации, которую модель может использовать в качестве базы знаний. Правила и база знаний объединяются в одно сообщение, называемое «промпт» (англ. Prompt). Промпт передается в модель, которая, в свою очередь, формирует ответ на естественном языке. Контекст модели ограничен в размерах и измеряется в токенах. Контекст действителен в рамках одной сессии. Сессия – это временной промежуток, в рамках которого пользователь взаимодействует с моделью. В рамках одной сессии контекст может быть очищен по инициативе пользователя неограниченное количество раз.

Тестирование проводилось на моделях GPT4 Omni, Llama 3.1:70b, Llama 3.1:8b, Gemini 1.5 Pro, Claude 3.5 Sonnet, GigaChat Pro, Phi3 Medium, Gemma 2, Mistral Nemo. Модели GPT4 Omni, Gemini 1.5 Pro и Claude 3.5 Sonnet имеют региональные ограничения, однако доступен тестовый период, в рамках которого удалось произвести тестирование. Несмотря на положительные результаты, коммерческое использование вышеперечисленных моделей крайне затруднено.

Модели Llama запускаются локально на компьютере пользователя. Это дает дополнительные преимущества в гибкости развертывания и избавляет от необходимости оплачивать доступ для использования модели. Без недостатков не обошлось – Llama 3.1:405b требует значительных ресурсов для запуска, поэтому тестирование производилось на моделях того же семейства с меньшим количеством параметров. Модели Llama 3.1:70b и Llama 3.1:8b показали себя очень хорошо. Они работают быстро и следуют правилам, указанным в промпте. Эти правила легко дополнить в рамках одной сессии. Модель поддерживает размер контекста 131072 токенов, что соответствует приблизительно 30 тыс. слов на русском языке. Это дает

возможность использовать не только объемный свод правил, но и передавать достаточно большое количество данных для анализа.

Нас интересует качество решения задачи извлечения смыслов из текста, поэтому для проведения тестирования языковых моделей были сгенерированы соответствующие запросы.

Ответ модели также необходимо корректно интерпретировать и верифицировать.

Процесс интерпретации результатов включает:

- понимание смысла и значимости полученных результатов;
- оценку достоверности результатов в выбранной предметной области;
- анализ возможных альтернативных объяснений выявленных закономерностей.

Процесс верификации результатов включает:

- проверку воспроизводимости результатов при повторном запуске;
- сравнение с ранее известными фактами и зависимостями;
- валидацию на данных, не используемых при основном тестировании;
- оценку значимости результатов с помощью статистических критериев.

Первый запрос: «*Я предоставлю тебе несколько кластеров, состоящих из набора слов. Твоя задача — суммаризировать смысловое значение этих слов. Кластер 1: [пельмени, хинкали, манты]. Кластер 2: [ноутбук, телефон, сервер, домофон]. Кластер 3: [iphone 11, samsung galaxy s3, nokia 3310]. Кластер 4: [ночь, дождь, темнота, тоска]. Кластер 5: [банан, апельсин, мандарин, киви, яблоко, манго]. Кластер 6: [репозиторий, среда разработки, линтер, консоль]. Старайся придумать максимально точную ассоциацию. Сопоставь каждому кластеру одно слово или словосочетание.*». Результат тестирования первого запроса для разных языковых моделей представлен в табл. 2.

Таблица 2

Результаты тестирования больших языковых моделей в задаче краткого изложения длинного текста

№	Ответ модели								
	GPT4 Omni	Llama 3.1:70b	Llama 3.1:8b	Gemini 1.5 Pro	Claude 3.5 Sonnet	Giga-Chat Pro	Phi3 Medium	Gemma 2	Mistral Nemo
1	Кулинария	Восточная кухня	Еда	Тестовые изделия с начинкой	Блюда	Пельмени	Пельмени	Тестовые блюда	Традиционная еда
2	Устройства для связи и работы	Устройство связи	Техника	Цифровые устройства	Электронные устройства	Техника	Устройство	Электроника	Техника и коммуникация
3	Телефоны	Смартфон	Смартфоны	Мобильные телефоны	Мобильные телефоны	Смартфон	Смартфон	Мобильные телефоны	Мобильные телефоны
4	Грусть	Меланхолия	Покой	Меланхолия	Меланхолия	Погода	Одиночество	Меланхолия	Тьма
5	Фрукты	Фрукт	Фрукты	Тропические фрукты	Тропические фрукты	Фрукты	Фрукты	Фрукты	Тропические фрукты
6	Программирование	Программирование	Программирование	Программирование	Разработка ПО	Инструменты	Разработка	Разработка ПО	Разработка ПО

Стоит отметить, что модели GPT4 Omni, Llama 3.1:70b, Gemini 1.5 Pro и Claude 3.5 Sonnet справились лучше всего. Модели Llama 3.1:8b, Mistral Nemo, Phi3 Medium и GigaChat Pro выдают не конкретные формулировки, но следуют инструкции, переданной в промпте. Модель Gemma 2 правильно именует кластеры в большинстве случаев, но не следует инструкции и пытается дополнить свой ответ.

Второй запрос был связан с определением краткого содержания текста отзывов на различные программные продукты: компьютерную игру, мобильное приложение для контроля финансов, десктопное приложение для редактирования видео. Текст запроса: «*Я предоставлю тебе отзыв пользователя. Начало отзыва. {Отзыв} Конец отзыва. Расскажи в одном предложении, чем недоволен пользователь?*».

Модели GPT4 Omni, Llama 3.1:70b, Llama 3.1:8b, Gemini 1.5 Pro, Claude 3.5 Sonnet, GigaChat Pro и Gemma 2 зачастую используют краткое и емкое повествование и вмещают в одном предложении достаточно информации для понимания основной претензии пользователя. Модели Llama 3.1:70b, Llama 3.1:8b, GPT4 Omni, Gemini 1.5 Pro и Claude 3.5 Sonnet передают смысл точнее всех. GigaChat Pro периодически пытается дать слишком развернутый ответ и ошибается в логических конструкциях. Gemma 2 дает слишком развернутый ответ, даже если дополнительно попросить ее об обратном. Модели Phi3 Medium и Mistral Nemo справились хуже всех. Они выделяют не ключевые мысли из текста и совершают логические ошибки. Результаты представлены в табл. 3.

Таблица 3

Результаты тестирования больших языковых моделей в задаче извлечения смысла из текста

Модель	Статус выполнения задания			
	Справилась с заданием	Справилась с незначительными ошибками	Справилась со значительными ошибками	Не справилась с заданием
GPT4 Omni	+			
Llama 3.1:70b	+			
Llama 3.1:8b	+			
Gemini 1.5 Pro	+			
Claude 3.5 Sonnet	+			
GigaChat Pro			+	
Gemma 2		+		
Phi3 Medium				+
Mistral Nemo			+	

Если в качестве вопроса в промпте использовать «*Расскажи коротким словосочетанием, что нужно улучшить в приложении, чтобы пользователь остался доволен?*», то для моделей GPT4 Omni, Llama 3.1:70b, Gemini 1.5 Pro и Claude 3.5 Sonnet результаты не изменяются – они все так же корректно отвечают на вопрос в запрашиваемом формате. Llama 3.1:8b и Mistral отвечают общими формулировками, не отражающими суть проблемы. GigaChat Pro предоставляет понятный ответ, но добавляет к ответу дополнительный контекст. Gemma 2 и Phi3 Medium предоставляет короткий и понятный ответ. Результаты представлены в табл. 4.

По результатам предварительного тестирования, для дальнейшего исследования была выбрана большая языковая модель Llama 3.1 на 70 миллиардов параметров. Ее результаты в задаче выделения основной мысли из текстовых данных оказались выше или наравне с другими проанализированными моделями. Ее преимуществом также является возможность локального развертывания и простой интеграции с уже имеющимися программными продуктами.

Таблица 4

Результаты тестирования больших языковых моделей в задаче извлечения смысла из текста

Модель	Статус выполнения задания		
	Справилась с заданием	Справилась с незначительными ошибками	Справилась со значительными ошибками
GPT4 Omni	+		
Llama 3.1:70b	+		
Llama 3.1:8b			+
Gemini 1.5 Pro	+		
Claude 3.5 Sonnet	+		
GigaChat Pro		+	
Gemma 2	+		
Phi3 Medium	+		
Mistral Nemo			+

Заключение

Обозначена необходимость использования экспертной системы для задач краткого изложения (суммаризации) и извлечения смыслов из текста. В качестве инструмента решения выбраны большие языковые модели (Large Language Model). Проведен анализ существующих на рынке решений. По результатам предварительного тестирования было принято решение использовать для дальнейшего исследования модель Llama 3.1 на 70 миллиардов параметров. Она лучше других доступных больших языковых моделей справляется с поставленными задачами и имеет дополнительные конкурентные преимущества.

Литература

1. *Воронина И. Е.* Выбор варианта из множества решений при нечетком сравнении строк / И. Е. Воронина, Н. А. Экерт // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. – 2023. – No 2. – С. 181–191.

2. *Воронина И. Е.* Автоматизированное формирование критериев оценивания отзывов на программное обеспечение / И. Е. Воронина, Н. А. Экерт // Сборник трудов Международной научной конференции «Актуальные проблемы прикладной математики, информатики и механики». – 2023 – С. 169–174.

3. *Ингерсолл Г. С.* Обработка неструктурированных текстов / Г. С. Ингерсолл, Т. С. Мортон, Э. Л. Фэррис. – Москва : ДМК Пресс, 2015. – 414 с.

4. *Хобсон Л.* Обработка естественного языка в действии / Л. Хобсон, Х. Ханнес, Х. Коул. – СПб. : Питер, 2020. – 576 с.

5. Документация GPT4 Omni. – URL: https://platform.openai.com/docs/overview (дата обращения: 05.11.2024).

6. Документация Claude 3.5 Sonnet. – URL: https://docs.anthropic.com/en/home (дата обращения: 14.11.2024).

7. Документация Gemini 1.5 Pro. – URL: https://ai.google.dev/gemini-api/docs (дата обращения: 12.11.2024).

8. Документация Llama 3.1. – URL: https://docs.llama-api.com/quickstart (дата обращения: 07.11.2024).

9. Документация GigaChat Pro. – URL: https://developers.sber.ru/docs/ru/gigachat/api/reference/rest/gigachat-api (дата обращения: 04.11.2024).

10. Келен О. Разработка приложений на базе GPT-4 и ChatGPT / О. Келен, М. Блете. – СПб. : Питер, 2024. – 192 с.

УДК 621.865:004.896

О РЕГРЕССИОННОМ МОДЕЛИРОВАНИИ В ЗАДАЧЕ ОБ УПРАВЛЕНИИ ПЛОСКИМ ЧЕТЫРЕХЗВЕННОМ МЕХАНИЗМОМ С ПРИВОДОМ

А. Ю. Яковлев, М. А. Релина, Р. Ю. Комнатный

Воронежский государственный университет

Аннотация. Работа посвящена разработке алгоритма управления плоским четырехзвенным механизмом с приводом, основанного на методе регрессии с использованием гауссовских процессов [1]. Представлен метод построения регрессионной модели для прогнозирования угловой скорости корневого звена, а также алгоритм управления механизмом. Разработано программное обеспечение, реализующее модель и алгоритм управления, выполнены компьютерные эксперименты. Результаты демонстрируют высокую точность прогнозов при ограниченном объеме обучающих данных. Исследование ориентировано на применение в робототехническом комплексе РОИН [5], где внедрение предложенных решений позволит улучшить управление гидравлическими приводами.
Ключевые слова: регрессионное моделирование, гауссовские процессы, четырехзвенный механизм, робототехнический комплекс, байесовский вывод, программное обеспечение, управление, угловая скорость, гидравлический привод, мехатроника, машинное обучение, компьютерное моделирование.

Введение

Создание нелинейных систем управления на основе методов машинного обучения для мехатронных и робототехнических устройств является актуальной задачей современной мехатроники и робототехники. Активное внедрение машинного обучения в управлении мехатронными и робототехническими комплексами находит отражение, в частности, в научных статьях, посвященных построению регрессионных моделей в робототехнике [2, 4].

Авторами настоящей работы разработано программное обеспечение, реализующее регрессионную модель плоского четырехзвенного механизма с приводом и алгоритм управления угловой скоростью корневого звена механизма на ее основе. Для построения регрессионной модели был выбран метод, основанный на гауссовских процессах [1, 3].

Практическая значимость работы определяется возможностью внедрения ее результатов в систему управления реального робототехнического комплекса РОИН [5]. Робототехнический комплекс РОИН оснащен гидравлическими приводами и представляет собой промышленный манипулятор. Применяемая система гидравлических приводов является одноконтурной и позволяет осуществлять управление движением приводов по одному в единицу времени. Поэтому полученная система управления корневым сегментом четырехзвенного механизма может быть использована для управления остальными приводами без дополнительных изменений.

1. Задача о кинематике движения плоского четырехзвенного механизма с приводом

На рис. 1 показана конструкция четырехзвенного механизма крепления рабочего инструмента робототехнического комплекса РОИН.

Для построения системы управления корневым сегментом четырехзвенного механизма рассмотрим его математическую модель. Схематически плоский кинематический механизм, состоящий из четырех шарнирно соединенных сегментов и гидравлического привода, представлен на рис. 2.

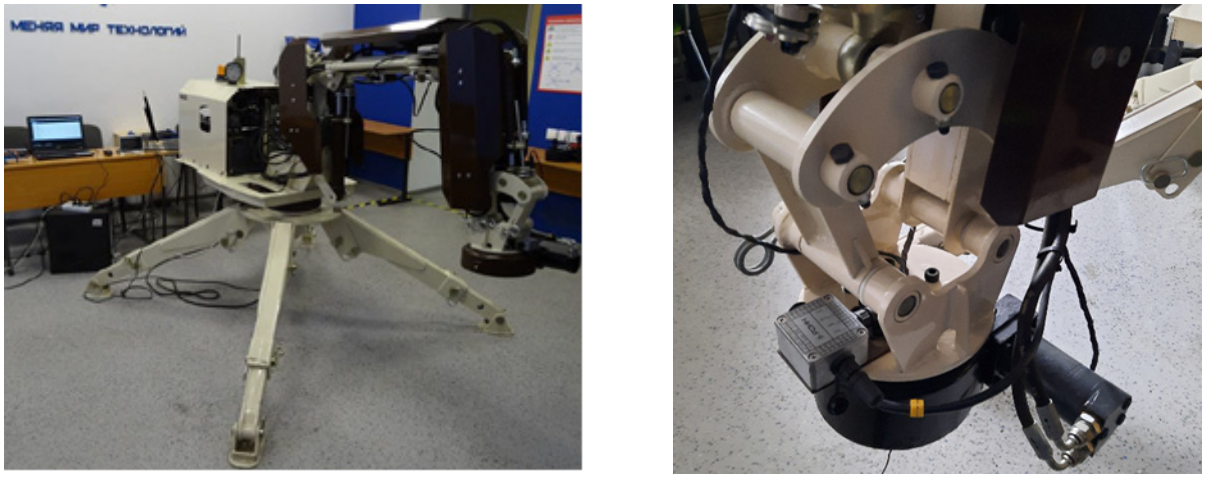


Рис. 1. Общий вид робототехнического комплекса РОИН (слева) и внешний вид четырехзвенного элемента стрелы комплекса (справа)

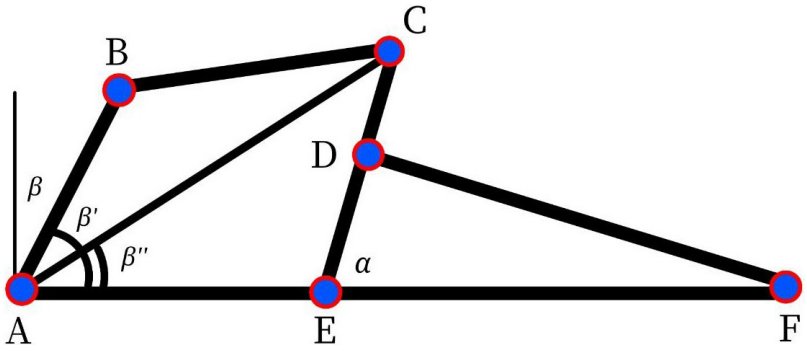


Рис. 2. Схематическое изображение плоского четырехзвенного механизма с приводом

Сочленения A, B, C, D, E, F представляют собой одноосевые шарниры. Сегмент AB будем называть корневым сегментом механизма, скорость его поворота в зависимости от скорости движения гидропривода есть параметр управления. Цель управления — удержание заданной скорости поворота звена AB. Гидропривод моделирует сегмент FD (далее — привод), его длину будем обозначать символом u. Величины длин сегментов AB = amdl, BC = d1, CD = bmdl/2, CE = bmdl, AE = cmdl и расстояние EF = fmdl заданы. Длина привода FD ограничена минимальным и максимальным значениями.

Для описания математической модели механизма запишем систему кинематических выражений, пригодную для компьютерной реализации.

Расстояние между точками A и C, определяющее длину одного из звеньев механизма, вычисляется по соотношению

$$d_1 = \sqrt{(A_x - C_x)^2 + (A_y - C_y)^2}. \quad (1)$$

Угол поворота звена CE относительно шарнира E определяется согласно выражению (2)

$$\alpha = \arccos\left(-\frac{(u+u) - (b_{\text{mdl}} \cdot 0.5) \cdot (b_{\text{mdl}} \cdot 0.5) - (f_{\text{mdl}} \cdot f_{\text{mdl}})}{b_{\text{mdl}} \cdot f_{\text{mdl}}} \right). \quad (2)$$

Для расчетов используются углы β_1 и β_2, описывающие взаимное расположение звеньев (3)

$$\beta_1 = \arccos\left(\frac{(b_{\text{mdl}}^2) - (d_1^2) - (d_{\text{mdl}}^2)}{2.0 \cdot d_1 \cdot d_{\text{mdl}}} \right), \; \beta_2 = \arccos\left(\frac{(c_{\text{mdl}}^2) - (a_{\text{mdl}}^2) - (d_1^2)}{2.0 \cdot a_{\text{mdl}} \cdot d_1} \right). \quad (3)$$

Координаты ключевых точек механизма можно определить следующим образом.

Координата точки C:

$$C_x = b_{\text{mdl}} \cdot \cos(\alpha), \quad C_y = b_{\text{mdl}} \cdot \sin(\alpha). \tag{4}$$

Координата точки D:

$$F_x = b_{\text{mdl}} \cdot 0.5 \cdot \cos(\alpha), \quad F_y = b_{\text{mdl}} \cdot 0.5 \cdot \sin(\alpha). \tag{5}$$

Координата точки B:

$$B_x = a_{\text{mdl}} \cdot \cos(\beta_1 + \beta_2) - d_{\text{mdl}}, \quad B_y = a_{\text{mdl}} \cdot \sin(\beta_1 + \beta_2). \tag{6}$$

Согласно (1)–(6), угол положения корневого сегмента относительно вертикали вычисляется по соотношению (7)

$$\beta = \left(\beta_1 + \beta_2 - \frac{\pi}{2} \right). \tag{7}$$

2. Описание метода построения регрессионной модели

Для построения регрессионной модели кинематики четырехзвенного механизма с приводом воспользуемся подходом, основанном на моделировании гауссовскими процессами [1, 2].

При таком подходе необходимо, чтобы обучающие данные $D = \{X = [\mathbf{x}_1, ..., \mathbf{x}_n]^T, Y[y_1, ..., y_n]^T\}$ описывали скрытую шумами функцию кинематической связи элементов четырехзвенного механизма $y_i = h(x_i) + \varepsilon_i$, где $h : R^D \to R$, $\varepsilon \sim N(0, \sigma_\varepsilon^2)$ — независимый гауссовский шум измерения. Функцию h будем рассматривать как случайную функцию и получим апостериорное распределение $p(h \mid D)$ по h из априорного распределения $p(h)$ и данных D, а также из предположения о гладкости функции h [2]. Полученное таким образом апостериорное распределение $p(h \mid D)$ позволит получать прогнозные значения функции $h(\mathbf{x}_*)$ для произвольных входных данных $x_* \in R^D$.

В задаче о четырехзвенном механизме вектор состояния системы может быть выбран различными способами, в данном случае $\mathbf{x} = [\theta, u]^T \in R^2$. Вектор $\mathbf{x}$ является независимым параметром.

Целью прогнозирования удобно выбрать соответствующую скорость поворота корневого сегмента, поэтому зависимый параметр является угловой скоростью $y = \dot{\theta} = \omega \in R$.

Апостериорное распределение случайной функции h найдем методом байесовского вывода в рамках гауссовских процессов (ГП) [2]. Байесовский вывод представляет собой трехэтапную процедуру:

Этап 1. Выбираем априорное распределение для неизвестной функции h.

Этап 2. Выполнение экспериментальных наблюдений с сохранением данных D.

Этап 3. Вычисление апостериорного распределения по функции h, которое уточняет априорное значение на основе данных наблюдений D.

Согласно [1, 2] априорное среднее значение принимается тождественно равным нулю, а ковариационная функция имеет следующий вид

$$k_h = k_{SE} + \delta_{pq}\sigma_\varepsilon^2, \; k_{SE}\left(\mathbf{x}_p, \mathbf{x}_q\right) = \alpha^2 \exp\left(-\frac{1}{2}\left(\mathbf{x}_p, \mathbf{x}_q\right)^T \Lambda^{-1}\left(\mathbf{x}_p, \mathbf{x}_q\right) \right), \; \mathbf{x}_p, \mathbf{x}_q \in R^D. \tag{8}$$

В соотношении (8) обозначены: α^2 — дисперсия значений функции h, $\Lambda = diag([l_1^2, ..., l_D^2])$ — диагональная матрица характерных масштабов длины l_i, δ_{pq} — символ Кронекера [2]. Величины $l_1, ..., l_D$, дисперсия α^2 значений функции h и дисперсия шума σ_ε^2 называются гиперпараметрами, которые группируются в векторе гиперпараметров $\boldsymbol{\theta}$.

Следующим действием необходимо сформировать обучающий набор данных, который должен состоять из n входных векторов $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)$ и соответствующих им целевых значений $\mathbf{y} = (y_1, ..., y_n)^T$.

После определения значений функции h для входных векторов вычисляем значения гиперпараметров. Для этого решаем задачу нелинейной оптимизации для определения вектора

гиперпараметров, который доставляет максимум логарифму функции правдоподобия. Логарифм функции правдоподобия представляет выражение [2]

$$\log p\left(\mathbf{y}\middle|\mathbf{X},\boldsymbol{\theta}\right)=-\frac{1}{2}\mathbf{y}^T\left(\mathbf{K}_{\grave{e}}+\sigma_\varepsilon^2\mathbf{I}\right)^{-1}\mathbf{y}-\frac{1}{2}\log\left|\mathbf{K}_{\grave{e}}+\sigma_\varepsilon^2\mathbf{I}\right|-\frac{D}{2}\log\left(2\pi\right), \tag{9}$$

где $\mathbf{K}_{\boldsymbol{\theta}}=\begin{pmatrix} k_h\left(\mathbf{x}_1,\mathbf{x}_1,\boldsymbol{\theta}\right) & . & . & k_h\left(\mathbf{x}_1,\mathbf{x}_n,\boldsymbol{\theta}\right) \\ . & . & . & . \\ . & . & . & . \\ k_h\left(\mathbf{x}_n,\mathbf{x}_1,\boldsymbol{\theta}\right) & . & . & k_h\left(\mathbf{x}_n,\mathbf{x}_n,\boldsymbol{\theta}\right) \end{pmatrix}$, $\mathbf{I}$ — единичная матрица размером $n\times n$.

Находим вектор оптимальных гиперпараметров $\hat{\boldsymbol{\theta}}\in\arg\max\log p(\mathbf{y}|\mathbf{X},\boldsymbol{\theta})$.

Следуя [2], можно заметить, что значения функции для тестовых и обучающих входных данных должны быть совместно гауссовыми

$$p\left(\mathbf{h},\mathbf{h}_*\middle|\mathbf{X},\mathbf{X}_*\right)=N\left(\begin{bmatrix} m_h(\mathbf{X}) \\ m_h(\mathbf{X}_*) \end{bmatrix},\begin{bmatrix} \mathbf{K} & \mathbf{K}_{\mathbf{h}}(\mathbf{X},\mathbf{X}_*) \\ \mathbf{K}_{\mathbf{h}}(\mathbf{X},\mathbf{X}_*) & \mathbf{K}_* \end{bmatrix}\right), \tag{10}$$

где $\mathbf{h}=[h(\mathbf{x}_1),...,h(\mathbf{x}_n)]^T$, $\mathbf{h}_*=[h(\mathbf{x}_{*1}),...,h(\mathbf{x}_{*n})]^T$, n — количество обучающих векторов, m — количество тестовых векторов для прогнозирования.

В работе применен случай определения прогнозного значения функции $h(\mathbf{x}_*)$, $h:R^D\to R$, при неопределенном тестовом входе $\mathbf{x}_*\sim N(\boldsymbol{\mu},\boldsymbol{\Sigma})\in R^D$, $y_*\in R$, где h есть ГП с функцией ковариации k_h (8) плюс функция ковариации шума. При этом среднее значение μ_* прогнозируемого распределения, согласно [2], запишется в виде

$$\mu_*=\boldsymbol{\beta}^T\mathbf{q}, \tag{11}$$

где

$$\boldsymbol{\beta}=\left(\mathbf{K}+\sigma_\varepsilon^2\mathbf{I}\right)^{-1}\mathbf{y},$$

$$\mathbf{q}=\left[q_1,...,q_n\right]^T\in R^n,\ q_i=\alpha^2\left|\boldsymbol{\Sigma}\Lambda^{-1}+\mathbf{I}\right|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}\left(\mathbf{x}_i-\boldsymbol{\mu}\right)^T\left(\boldsymbol{\Sigma}+\Lambda\right)^{-1}\left(\mathbf{x}_i-\boldsymbol{\mu}\right)\right),$$

$$\mathbf{K}=\begin{pmatrix} k_h\left(\mathbf{x}_1,\mathbf{x}_1\right) & . & . & k_h\left(\mathbf{x}_1,\mathbf{x}_n\right) \\ . & . & . & . \\ . & . & . & . \\ k_h\left(\mathbf{x}_n,\mathbf{x}_1\right) & . & . & k_h\left(\mathbf{x}_n,\mathbf{x}_n\right) \end{pmatrix}.$$

3. Описание алгоритма построение регрессионной модели

Рассмотрим алгоритм построения регрессионной модели четырехзвенного механизма с приводом методом гауссовских процессов с применением трёхэтапной байесовской процедуры [2].

1. Используя математическую модель кинематики четырехзвенного механизма с приводом, фиксируем набор данных в виде кортежей, состоящих из угла положения корневого сегмента и скорости движения привода (входные данные) $(\theta,u)^T$ и соответствующих целевых значений угловой скорости корневого сегмента.

2. Априорное распределение моделируемой функции h задается выражением (8).

3. Для сформированного набора входных и выходных данных функции h рассчитываем оптимальный вектор гиперпараметров методом определения экстремума логарифма правдоподобия (9) [2].

4. Вычисляем апостериорное распределение значений моделируемой функции h, используя соотношение (11).

4. Результаты моделирования

Для компьютерной реализации рассмотренной математической модели четырехзвенного механизма с приводом и алгоритма регрессионного моделирования гауссовскими процессами была разработана программа на языке C++ в среде Processing [6].

Программа позволяет проводить все необходимые расчеты и демонстрировать визуализацию процессов (рис. 3).

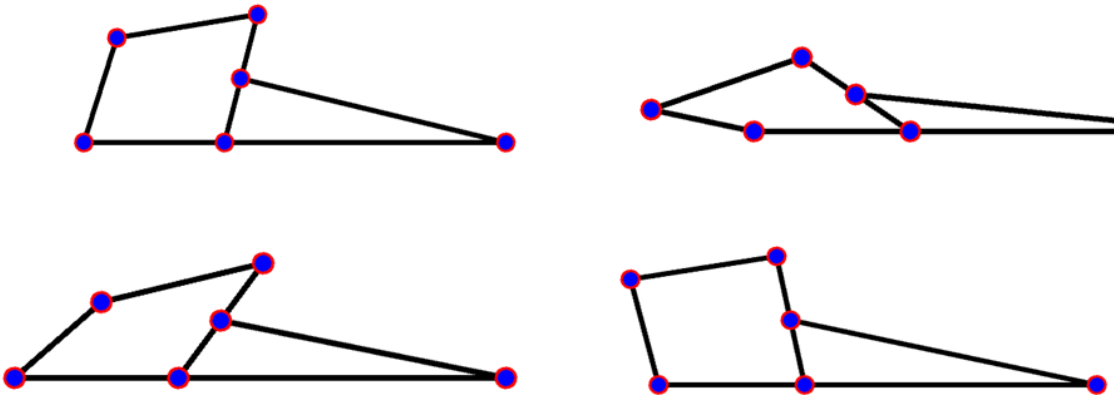


Рис. 3. Общий вид компьютерной модели четырехзвенного механизма с приводом в разных фазах движения

Отметим, что при разработке программы не использовались сторонние программные библиотеки. В состав программы входит девять модулей, и ее общий размер составляет 114 Кб.

Базовыми модулями являются:

GP.pde — реализует функцию логарифма правдоподобия (9), функцию вычисления прогнозного значения по соотношению (11);

Unit.pde — реализует функцию нелинейного поиска максимума логарифма функции правдоподобия (9) методом BFGS [1] для получения оптимального вектора гиперпараметров;

Matrix.pde — содержит необходимый для проведения матричных операций набор функций;

Model.pde — реализует математическую модель кинематики четырехзвенного механизма с приводом. Математическая модель используется для формирования обучающего набора данных и проведения компьютерного эксперимента с прогнозированием.

На рис. 4 приведен график изменения угловой скорости в процессе компьютерного моделирования движения четырехзвенного механизма с приводом в зависимости от угла положения корневого сегмента и скорости движения привода. Вертикальными линиями указаны положения механизма, при которых сохранялись данные в обучающий набор. Общее количество кортежей обучающего набора составляет 48 элементов.

Полученная регрессионная модель четырехзвенного механизма с приводом позволит создать нелинейную систему управления корневым сегментом механизма. Данный результат связан с созданием системы управления робототехническим комплексом РОИН [5]. В частности, изменение управляющего сигнала (скорости движения) для привода в зависимости от положения корневого сегмента механизма для удержания заданного значения его угловой скорости приведено на рис. 5.

На рис. 5 можно наблюдать график управляющего сигнала на привод и скорость поворота корневого сегмента, вычисляемые с использованием регрессионной модели механизма.

По итогам проведенной работы можно наблюдать высокую эффективность и точность моделирования гауссовскими процессами при небольшом объеме обучающих данных. Результаты, полученные в работе, могут быть использованы при создании макета этого механизма. В качестве привода будет использован линейный актуатор, угловое положение и угловая ско-

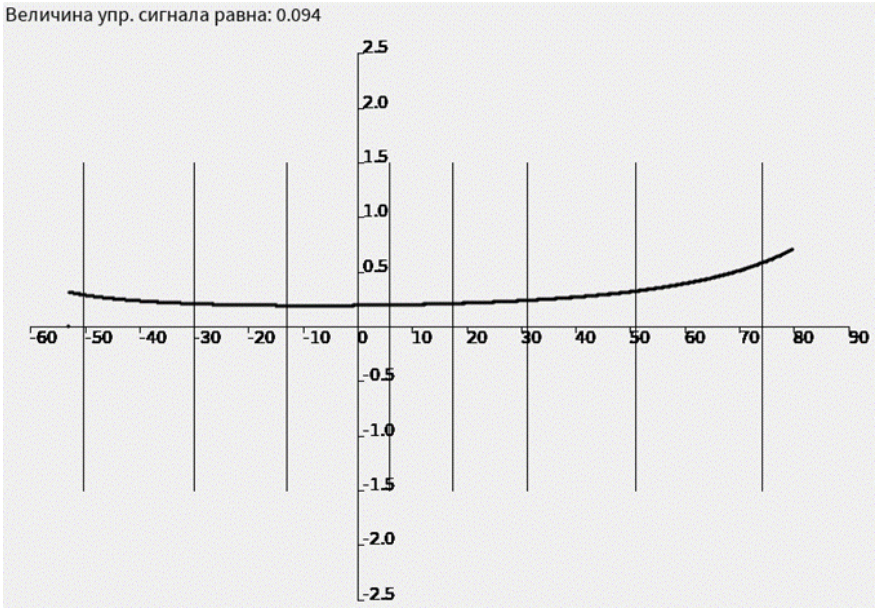


Рис. 4. График изменения скорости корневого сегмента в зависимости от его углового положения и скорости привода. По оси абсцисс откладывается угол положения, по оси ординат откладывается угловая скорость корневого сегмента. График получен при постоянной скорости привода

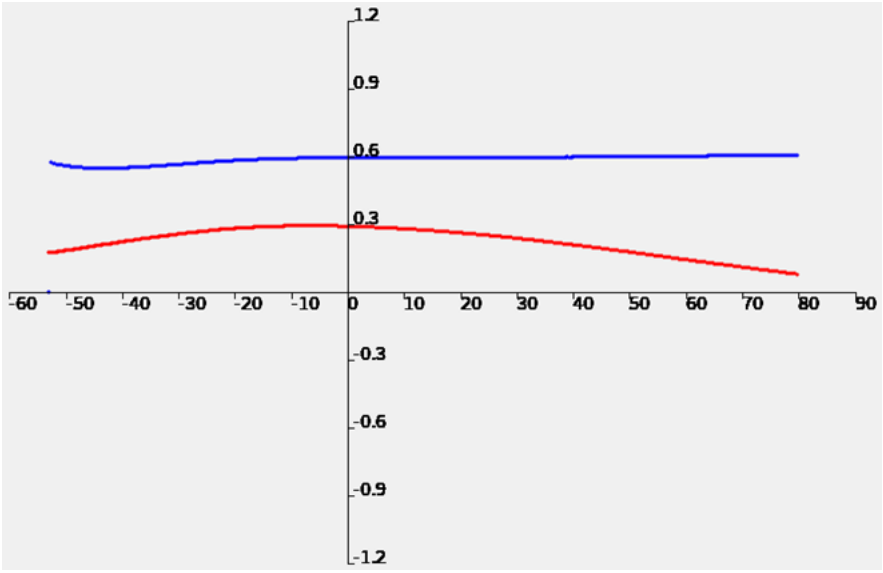


Рис. 5. Графики отражают процесс управления корневым сегментом. Целевая угловая скорость задана на уровне 0.6 с–1. Фактическая угловая скорость корневого сегмента механизма — синий (верхний) график. Красный график отображает величину управляющего сигнала (скорость движения) на привод

рость корневого сегмента будут определяться с помощью модуля инерциальных датчиков. Система управления на основе регрессионной модели с использованием гауссовских процессов позволит механизму «самостоятельно» обучаться и впоследствии выполнять повороты корневым сегментом по заданному закону движения.

Заключение

В работе представлен алгоритм управления четырехзвенным механизмом с приводом на основе регрессии гауссовскими процессами. Разработана компьютерная модель механизма и

реализована программно процедура построения нелинейной регрессионной модели на основе гауссовских процессов. Проведены компьютерные эксперименты с моделью механизма по управлению его корневым сегментом. Модуль, представляющий программную реализацию регрессионной модели, может быть применен без значительных изменений для решения широкого класса отдельных задач. Он также может быть использован в программном обеспечении микроконтроллеров мехатронных и робототехнических устройств.

Литература

1. *Бишоп Кристофер М.* Распознавание образов и машинное обучение : пер. с англ. – СПб. : ООО «Диалектика», 2020. – 960 с.

2. *Deisenroth M. P.* Efficient Reinforcement Learning using Gaussian Processes. – KIT Scinticfic Publishing, 2011.

3. *Измаилов П. А.* Алгоритмы обучения гауссовских процессов для больших объемов данных. – Москва, 2017.

4. PILCO: A Model-Based and Data-Efficient Approach to Policy Search, 2011. – 1–4 с.

5. ИНТЕХРОС : официальный сайт. — URL: http://intehros.ru (дата обращения: 20.11.2024).

6. Processing : официальный сайт. — URL: https://processing.org (дата обращения: 20.11.2024).