Федеральное государственное бюджетное учреждение науки Институт системного программирования им. В.П. Иванникова Российской академии наук

На правах рукописи

Перминов Андрей Игоревич

Доверенный байесовский классификатор для данных малой размерности на основе многослойного персептрона

Специальность 2.3.5 —
«Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей»

Диссертация на соискание учёной степени
кандидата физико-математических наук

Научный руководитель:
кандидат физико-математических наук
Турдаков Денис Юрьевич

Москва — 2026

Оглавление

Стр.

Стр.

Стр.

Стр.

Стр.

Введение

Парадигма доверенного искусственного интеллекта направлена на обеспечение возможности применения методов машинного обучения в критически важных областях, включая государственное управление, критическую инфраструктуру, медицину и финансовые системы. В рамках этой парадигмы модели должны не только демонстрировать высокое качество предсказаний, но и обладать формализованными механизмами оценки собственной уверенности, определения границ применимости и принятия статистически обоснованных решений [1]. Реализация этих требований в общем виде невозможна без строгой математической теории, обеспечивающей формальные гарантии корректности работы моделей.

Однако в настоящее время такая теория для современных методов машинного обучения в целом отсутствует. На практике преобладают эмпирические подходы, ориентированные главным образом на оптимизацию стандартных метрик качества. Несмотря на это, нейросетевые модели получили широкое распространение благодаря своей универсальности и способности эффективно решать широкий круг прикладных задач. Но, как правило, они не обеспечивают строгого статистического обоснования принимаемых решений и контроля области своей применимости.

В то же время в математической статистике разработан развитый теоретический аппарат для задач классификации, позволяющий получать интерпретируемые результаты и строгие вероятностные гарантии. Однако область применимости классических статистических методов существенно уже по сравнению с методами машинного обучения и, в частности, нейросетевыми моделями, что ограничивает их использование в современных прикладных задачах.

Исследование основано на положениях российских и зарубежных научных школ теории распознавания образов. Методологическую базу составляют труды Ю. И. Журавлёва, К. В. Рудакова и К. В. Воронцова, а также исследования М. И. Забежайло, А. А. Грушо и А. К. Горшенина. Значительное влияние оказали фундаментальные работы по теории статистического обучения В. Н. Вапника, А. Я. Червоненкиса и Л. Девроя и вероятностным моделям К. Бишопа.

Настоящая работа делает шаг в направлении построения математической теории нейросетевых моделей на основе методов математической статистики. Исследование сосредоточено на задачах классификации в пространствах малой размерности, что позволяет сохранить формальную строгость получаемых результатов. В рамках работы предлагается статистически обоснованный доверенный классификатор на основе многослойного персептрона, обеспечивающий формализованную оценку уверенности и определение границ компетенции модели, тем самым закладывая основу для дальнейшего развития теории доверенного искусственного интеллекта.

Целью данной работы является разработка методики построения доверенных классификаторов на основе многослойного персептрона для данных малой размерности, обеспечивающей способность к отказу от классификации вне носителя распределения, устойчивость к дисбалансу классов и интерпретируемость принимаемых решений.

Для достижения поставленной цели необходимо было решить следующие **задачи**:

1. Разработать метод построения доверенного объяснимого классификатора на основе многослойного персептрона, обеспечивающего статистически обоснованное оценивание апостериорных вероятностей и устойчивость к дисбалансу классов.
2. Разработать метод генерации синтетических данных, сохраняющих геометрические и статистические свойства исходного распределения, на основе разработанного метода.
3. Провести экспериментальное исследование разработанных методов для оценки устойчивости классификатора к дисбалансу классов, корректности работы вне носителя обучающего распределения и качества генерируемых синтетических данных.
4. Разработать интеллектуальную систему машинного обучения, реализующую предложенные методы и обеспечивающую решение задач классификации данных малой размерности в условиях дисбаланса классов и высокой неопределённости вне носителя распределения.

Основные положения, выносимые на защиту:

1. Теоретическая база непараметрического оценивания в условиях дисбаланса классов и малой размерности. Сформулированы и доказаны

теоремы, обосновывающие асимптотическую связь между нейросетевой и гистограммной оценками апостериорной вероятности.

2. Метод построения статистически обоснованного объяснимого байесовского классификатора на основе многослойного персептрона и дерева решений.
3. Метод построения унарного классификатора, устойчивого к дисбалансу классов и позволяющего генерировать синтетические данные.
4. Интеллектуальная система машинного обучения, реализующая предложенные методы и обеспечивающая решение задач классификации данных малой размерности в условиях дисбаланса классов и высокой неопределённости вне носителя распределения.

Перечисленные положения относятся к направлениям исследований 4, 7, 8 и 9 паспорта специальности 2.3.5 «Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей»:

- п. 4. Интеллектуальные системы машинного обучения, управления базами данных и знаний, инструментальные средства разработки цифровых продуктов.
- п. 7. Модели, методы, архитектуры, алгоритмы, форматы, протоколы и программные средства человеко-машинных интерфейсов, компьютерной графики, визуализации, обработки изображений и видеоданных, систем виртуальной реальности, многомодального взаимодействия в социокиберфизических системах.
- п. 8. Модели и методы создания программ и программных систем для параллельной и распределенной обработки данных, языки и инструментальные средства параллельного программирования.
- п. 9. Модели, методы, алгоритмы, облачные технологии и программная инфраструктура организации глобально распределенной обработки данных.

Научная новизна: разработан метод построения доверенного классификатора на основе многослойного персептрона, обеспечивающего формальные гарантии корректного поведения модели. Предложен подход, позволяющий трактовать выход персептрона как статистически обоснованную оценку апостериорной вероятности и реализующий механизм осознанного отказа от классификации для объектов вне носителя обучающего распределения, что отличает его от стандартных нейросетевых методов, не имеющих подобного

теоретического обоснования. Предложен метод унарной классификации, устраняющий проблему дисбаланса классов без искажающих процедур балансировки и позволяющий генерировать синтетические данные, сохраняющие геометрические и статистические свойства исходной выборки. Предложен инструмент объяснения решений классификатора – дерево eXBTree, обеспечивающее интерпретируемость модели за счёт анализа правил принятия решений и схожих прецедентов. Теоретической основой подхода является доказанная теорема о корректном поведении классификатора вне носителя распределения, что вносит вклад в развитие математических основ доверенного искусственного интеллекта для нейросетевых моделей.

Теоретическая и практическая значимость

Теоретическая значимость работы заключается в развитии статистических основ доверенной классификации на основе многослойного персептрона и в формировании формализованного подхода к оценке уверенности предсказаний нейросетевых моделей. В работе сформулирован и доказан ряд теорем, позволяющих статистически обосновать построение доверенных классификаторов, определить границы компетенции модели и описать её поведение вне носителя распределения, включая механизм отказа от классификации. Установлена асимптотическая связь между нейросетевой и гистограммной оценками апостериорной вероятности, что подтверждает состоятельность предложенного подхода. Полученные результаты расширяют теоретическую базу непараметрического оценивания в условиях дисбаланса классов и малой размерности и формируют основу для построения математически строгой теории доверенного искусственного интеллекта.

Практическая значимость работы заключается в использовании предложенных методов при разработке инструментов доверенного искусственного интеллекта в Исследовательском Центре Доверенного Искусственного Интеллекта (ИЦДИИ) ИСП РАН. Разработанный классификатор применяется для анализа данных в условиях дисбаланса классов, обеспечивая интерпретируемость решений и повышение надёжности за счёт механизма автоматического отказа от классификации в недостоверных областях. Метод генерации синтетических данных, сохраняющих статистическую структуру оригинала, используется для безопасного расширения обучающих выборок. Реализованная система обеспечивает воспроизводимость и практическое применение подхода в задачах, требующих доверенного принятия решений.

Апробация работы. Основные результаты работы были представлены на следующих конференциях и семинарах:

- Форум «Цифровая экономика. Технологии доверенного искусственного интеллекта», Москва, 25 мая 2023 г.
- 32-я научно-техническая конференция «Методы и технические средства обеспечения безопасности информации» (МиТСОБИ), Санкт-Петербург, 26-29 июня 2023 г.
- WAIT: Workshop on Artificial Intelligence Trustworthiness, Almaty, Kazakhstan, 24 апреля 2024 г.
- Международная конференция «Иванниковские чтения», Великий Новгород, 17-18 мая 2024 г.
- II форум «Технологии доверенного искусственного интеллекта», Москва, 27 мая 2024 г.
- 33-я научно-техническая конференция «Методы и технические средства обеспечения безопасности информации» (МиТСОБИ), Санкт-Петербург, 24-27 июня 2024 г.
- MathAI 2025 The International Conference dedicated to mathematics in artificial intelligence, March 24-28, 2025 г.
- III форум «Технологии Доверенного Искусственного Интеллекта», Москва, 20 мая 2025 г.
- 34-я всероссийская конференция «Методы и технические средства обеспечения безопасности информации» (МиТСОБИ), Санкт-Петербург, 23-26 июня 2025 г.
- Международная конференция «Иванниковские чтения», Иркутск, 26-27 июня 2025 г.

Личный вклад. Все выносимые на защиту результаты получены лично автором.

Публикации. Основные результаты по теме диссертации изложены в 9 печатных изданиях, 6 из которых изданы в журналах, рекомендованных ВАК, 4 — в периодических научных журналах, индексируемых Web of Science и Scopus, 3 — в тезисах докладов. Зарегистрированы 4 программы для ЭВМ.

Личный вклад в совместные публикации является определяющим. Из 6 основных публикаций по теме диссертации одна работа [2] выполнена без соавторов. В основных публикациях по теме диссертации автору принадлежат: метод статистически обоснованного объяснимого байесовского классификатора на основе многослойного персептрона и дерева решений eXBTree и соответствующая формулировка теоремы, а также разработка системы визуализации DenseNetworkVisualizer [3], теорема, обосновывающая асимптотическую связь между нейросетевой и гистограммной оценками апостериорной вероятности, и метод построения унарного классификатора, устойчивого к дисбалансу классов [4], метод генерации синтетических данных [5; 6], метод обучения классификатора на основе многослойного персептрона на данных с пропусками [7].

Объём и структура работы. Диссертация состоит из введения, 5 глав, заключения и 2 приложений. Полный объём диссертации составляет 154 страницы, включая 41 рисунок и 6 таблиц. Список литературы содержит 102 наименования.

Глава 1. Проблемы доверия в задаче классификации

1.1 Введение в задачу классификации и требования доверенных систем

Задача классификации является одной из базовых и наиболее изученных задач машинного обучения с учителем. В формальной постановке по заданной обучающей выборке

$$\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n,$$

где $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$ – вектор признаков объекта, а $y_i \in \mathcal{Y} = \{1, 2, \ldots, C\}$ – соответствующая метка класса, требуется построить решающую функцию

$$h : \mathcal{X} \to \mathcal{Y},$$

минимизирующую вероятность ошибочной классификации на новых объектах, порождённых тем же, неизвестным распределением данных [8].

В традиционной постановке эффективность классификатора оценивается по его обобщающей способности, измеряемой с помощью стандартных метрик качества, таких как accuracy, precision, recall и f_1-мера, вычисляемых на независимой тестовой выборке. Однако в задачах, относящихся к ответственным прикладным областям – включая медицинскую диагностику, анализ клинических рисков, финансовые и технические системы поддержки принятия решений, – данных критериев оказывается недостаточно. В таких сценариях возрастает значимость не только точности предсказаний, но и их надёжности, интерпретируемости и статистической обоснованности, что приводит к необходимости разработки так называемых доверенных интеллектуальных систем, которые должны удовлетворять комплексу дополнительных требований, обеспечивающих их надёжность, прозрачность и безопасность.

– **Способность к оценке собственной уверенности.** Классификатор должен предоставлять не только точечное предсказание класса, но и количественную оценку уверенности или апостериорного распределения вероятностей по классам. Это позволяет различать ситуации, в которых принятое решение является статистически обоснованным, и случаи, характеризующиеся высокой степенью неопределённости.

– **Корректное определение области компетенции и обнаружение выхода за распределение.** Модель должна учитывать ограничения, накладываемые обучающим распределением данных. При поступлении объекта, существенно отличающегося от обучающих примеров и находящегося вне носителя распределения, система не должна формировать произвольное предсказание. Необходим формальный механизм выявления таких наблюдений.
– **Наличие механизма отказа от принятия решения.** В ситуациях, когда уровень уверенности модели оказывается ниже допустимого порога либо объект идентифицируется как находящийся вне области компетенции, классификатор должен обладать формализованной процедурой отказа от автоматического решения с возможностью передачи управления эксперту или запроса дополнительной информации.
– **Интерпретируемость и объяснимость принимаемых решений.** В критически значимых приложениях требуется не только результат классификации, но и возможность анализа факторов, повлиявших на его получение. Объяснимость решений является необходимым условием аудита, повышения доверия пользователей и последующего совершенствования моделей в рамках парадигмы объяснимого искусственного интеллекта (XAI).
– **Устойчивость к неблагоприятным и аномальным условиям.** Классификатор должен сохранять корректность поведения при наличии шума, выбросов, существенного дисбаланса классов, а также в условиях целенаправленных состязательных воздействий, направленных на искажение его выходных решений.

Парадигма доверенного искусственного интеллекта смещает акцент с экстремальной оптимизации точечных метрик качества в сторону построения надёжных, предсказуемых и прозрачных моделей, корректно интегрируемых в сложные процессы принятия решений. Вместе с тем, как будет показано в последующих разделах данной главы, большинство широко используемых методов классификации в той или иной степени не удовлетворяют указанным требованиям, в особенности в части корректной работы с неопределённостью, определения границ собственной компетенции и обеспечения объяснимости принимаемых решений.

1.2 Методы классификации

Историческое развитие методов классификации отражает постепенный переход от статистически строгих моделей, основанных на явно сформулированных предположениях о природе данных, к более универсальным и гибким алгоритмам, ориентированным преимущественно на достижение высокой предсказательной точности на выборках сложной и слабо структурированной природы. Каждый из сформировавшихся классов методов обладает собственными достоинствами, однако одновременно характеризуется ограничениями, существенными при построении доверенных и статистически обоснованных систем принятия решений.

К числу наиболее ранних и теоретически проработанных подходов относятся непараметрические методы классификации, основанные на оценке плотности распределения или локальной структуры данных. В частности, гистограммные методы и метод k ближайших соседей (k-NN) опираются на минимальные априорные предположения и обладают асимптотическими гарантиями состоятельности при выполнении достаточно общих условий [9]. Гистограммные классификаторы аппроксимируют распределение данных путём разбиения пространства признаков на ячейки, тогда как k-NN принимает решение на основе локального большинства в окрестности объекта. Эти методы имеют прозрачную статистическую интерпретацию и естественным образом отражают локальную структуру данных, однако их практическое применение существенно ограничено ростом размерности пространства признаков, что приводит к резкому ухудшению обобщающей способности и вычислительной эффективности.

Классические линейные методы, к которым относятся логистическая регрессия и метод опорных векторов (Support Vector Machine, SVM), основаны на построении линейной разделяющей поверхности в пространстве признаков [10]. Их ключевыми преимуществами являются сравнительная простота, хорошая теоретическая изученность, а в случае SVM — строгое обоснование в рамках принципа минимизации структурного риска. Вместе с тем выразительная способность данных моделей ограничена предположением о линейной разделимости классов. На практике это предположение часто нарушается и компенсируется за счёт нелинейных преобразований признакового пространства, в

частности с использованием ядерных функций, что приводит к усложнению модели и снижению прозрачности интерпретации получаемых решений.

Деревья решений реализуют принципиально иной, непараметрический подход к классификации, формируя решающее правило в виде иерархической структуры элементарных условий на значения признаков [11]. К их существенным достоинствам относятся устойчивость к монотонным преобразованиям признаков, возможность работы с данными смешанной природы, а также высокая интерпретируемость, поскольку структура дерева непосредственно отражает логику принятия решений. В то же время одиночные деревья решений обладают высокой дисперсией и склонны к переобучению, что выражается в значительной чувствительности к малым изменениям обучающей выборки.

Для повышения устойчивости и качества обобщения были разработаны ансамблевые методы, агрегирующие предсказания множества базовых классификаторов. Метод случайного леса (Random Forest) сочетает идеи бутстреп-агрегирования и случайного подмножества признаков при построении каждого дерева, формируя ансамбль слабо коррелированных моделей [12]. Такой подход позволяет существенно снизить дисперсию по сравнению с одиночным деревом и одновременно сохранить возможность оценки относительной важности признаков. Градиентный бустинг (Gradient Boosting), напротив, строит ансамбль последовательно, обучая каждую последующую модель аппроксимировать ошибки предыдущих [13]. Это обеспечивает высокую аппроксимационную способность и, как правило, превосходную предсказательную точность. Однако за счёт усложнения структуры ансамбля данные методы утрачивают интерпретируемость, превращаясь в модели с трудно прослеживаемой логикой принятия решений.

Наиболее гибкими с точки зрения аппроксимации сложных нелинейных зависимостей являются нейросетевые модели, в частности многослойные персептроны (Multilayer Perceptron, MLP) [14]. Их архитектура, представляющая собой композицию линейных преобразований и нелинейных функций активации, теоретически позволяет аппроксимировать произвольные непрерывные функции на компакте. В частности, показано, что глубинные ReLU-сети могут достигать оптимальных скоростей аппроксимации в зависимости от гладкости функции и числа параметров, причём существуют различимые фазы аппроксимации, характеризующие соотношение глубины сети и числа весов [15]. Вместе с тем высокая выразительная способность нейронных сетей достигается ценой

значительного увеличения числа параметров, утраты интерпретируемости и высокой зависимости качества обучения от объёма и репрезентативности доступных данных.

Таким образом, в современных методах классификации наблюдается фундаментальный компромисс между интерпретируемостью, теоретической обоснованностью и гибкостью модели. Методы с прозрачной структурой и строгими статистическими свойствами, включая гистограммные классификаторы, k-NN, линейные модели и деревья решений, ограничены в способности описывать сложные зависимости и плохо масштабируются по размерности. В то же время наиболее мощные по точности подходы, такие как ансамблевые методы и глубокие нейросетевые модели, функционируют как "чёрные ящики" и не содержат встроенных механизмов для оценки достоверности предсказаний, особенно в условиях, выходящих за пределы обучающего распределения. Ни один из широко применяемых классов методов не предлагает целостного и статистически обоснованного решения для работы с объектами вне носителя распределения и для принятия решений в условиях высокой неопределённости, что и определяет актуальность дальнейшего рассмотрения данной проблемы.

1.3 Подходы к решению проблем доверия: детекция выхода за распределение и обработка дисбаланса

Фундаментальные проблемы доверия, присущие большинству классических и современных методов классификации, связаны с некорректным поведением на объектах вне носителя обучающего распределения и чувствительностью к дисбалансу классов. Существенную роль при этом играет неопределённость предсказаний моделей, обусловленная как стохастической природой данных, так и ограниченностью обучающей выборки и модели. Эти эффекты приводят к снижению надёжности принимаемых решений и являются предметом активных исследований. Для их компенсации предложен ряд специализированных подходов, которые, как правило, не модифицируют ядро самого алгоритма классификации, а реализуются в виде внешних методов.

1.3.1 Методы оценки неопределённости

В задачах доверенного машинного обучения ключевую роль играет анализ неопределённости предсказаний модели. В современной теории машинного обучения принято выделять два фундаментальных типа неопределённости: алеаторную и эпистемическую.

Алеаторная неопределённость обусловлена внутренней стохастичностью данных и шумом измерений и отражает вариативность наблюдений при фиксированном входе. Она является свойством самого распределения данных и, в отличие от эпистемической неопределённости, принципиально не может быть устранена путём увеличения объёма обучающей выборки [16].

Эпистемическая неопределённость связана с ограниченностью знаний модели о структуре распределения данных и обусловлена конечностью обучающей выборки и ограниченной выразительной способностью модели. Данный тип неопределённости, напротив, может быть уменьшен при поступлении дополнительной информации и, как правило, возрастает в областях пространства признаков, слабо представленных в обучающей выборке [17; 18].

Различение алеаторной и эпистемической неопределённости имеет принципиальное значение для построения доверенных классификаторов. В частности, высокая эпистемическая неопределённость может интерпретироваться как индикатор статистической необоснованности применения модели и выхода за пределы области её компетенции, тогда как алеаторная неопределённость отражает фундаментальную неоднозначность классификации внутри обучающего распределения.

Существующие методы оценки неопределённости в нейросетевых моделях развиваются в нескольких направлениях. Наиболее распространённые подходы основаны на анализе вероятностных выходов модели и их энтропийных характеристик, использовании байесовских аппроксимаций и ансамблевых моделей для оценки вариативности предсказаний, а также на исследовании структуры пространства признаков и плотности данных. Несмотря на разнообразие предлагаемых решений, в большинстве случаев оценка неопределённости реализуется как внешняя процедура по отношению к базовой модели классификации и требует дополнительной калибровки или усложнения архитектуры [19]. Это ограничивает их применение в задачах доверенного машинного обучения и мо-

тивирует разработку подходов, в которых оценка неопределённости является внутренним свойством классификатора.

1.3.2 Методы обнаружения выхода за распределение

Задача обнаружения объектов, распределение которых отличается от распределения обучающей выборки (Out-of-Distribution, OOD), формулируется как построение решающего правила

$$d : \mathcal{X} \to \{\text{in}, \text{out}\},$$

определяющего, применима ли исходная модель классификации $h(\mathbf{x})$ к данному наблюдению. Целью является выявление таких объектов, для которых использование обученного классификатора является статистически необоснованным. Существующие методы OOD-детекции можно разделить три основных направления.

Первое направление основано на анализе выходных значений обученной модели. Базовым представителем данной группы является метод максимальной вероятности softmax (Maximum Softmax Probability, MSP) [20], в котором объект относится к OOD, если максимальное апостериорное значение, выдаваемое классификатором, оказывается ниже заранее заданного порога. Несмотря на простоту реализации, данный подход уязвим к проблеме избыточной уверенности нейронных сетей, проявляющейся в высоких значениях softmax даже для нерелевантных входных данных. Более развитые методы используют ансамбли моделей или стохастические аппроксимации байесовского вывода для оценки дисперсии предсказаний, интерпретируемой как мера предсказательной неопределённости [17]. Однако такие методы, как правило, требуют многократных прямых проходов модели для одного объекта и зависят от эмпирической калибровки.

Второе направление ориентировано на анализ внутренних представлений данных, формируемых моделью. Основное предположение заключается в том, что объекты, принадлежащие распределению обучения, образуют компактные области в пространстве признаков, тогда как OOD-наблюдения располагаются вне этих областей. Классическим примером является детектор на основе

расстояния Махаланобиса [21], в котором вычисляется расстояние между представлением нового объекта и центрами классов, оценёнными по промежуточным слоям нейронной сети. Данные методы, как правило, требуют хранения статистик обучающей выборки и обладают повышенной вычислительной сложностью, что ограничивает их применение в ресурсно-ограниченных сценариях.

Третье направление предполагает модификацию процедуры обучения или архитектуры модели с целью повышения способности к детекции OOD-объектов. К данной группе относятся методы, использующие контрастивное обучение, а также подходы, в которых в функцию потерь добавляется дополнительное слагаемое, направленное на увеличение различия между представлениями объектов из обучающего распределения и объектов, рассматриваемых как OOD [22]. Эти методы наиболее тесно интегрированы с процессом обучения модели, однако требуют наличия синтетических или специально отобранных OOD-данных на этапе тренировки, что во многих практических задачах является трудно реализуемым или принципиально невозможным.

Несмотря на разнообразие существующих подходов, их объединяет общий недостаток: аддитивный характер по отношению к базовой модели классификации. OOD-детекция реализуется в виде внешнего механизма, настраиваемого поверх уже обученного классификатора, а её эффективность существенно зависит от выбора порогов, архитектурных решений и наличия репрезентативных данных для калибровки. Это усложняет практическое применение и не обеспечивает надёжной работы в условиях априорно неизвестных возмущений распределения данных.

1.3.3 Методы обработки дисбаланса классов

Проблема дисбаланса классов возникает в ситуациях, когда априорное распределение классов в обучающей выборке существенно отклоняется от равномерного, что приводит к смещению решающего правила в пользу мажоритарных классов. Существующие методы противодействия дисбалансу традиционно подразделяются на подходы уровня данных и уровня алгоритма [23].

Методы уровня данных включают недовыборку (undersampling) объектов мажоритарного класса и перевыборку (oversampling) объектов миноритарных классов, в том числе с использованием синтетической генерации примеров. К последним относится, в частности, алгоритм SMOTE [24]. Достоинством этих методов является их совместимость с произвольными алгоритмами обучения. Вместе с тем они обладают принципиальными ограничениями: недовыборка приводит к утрате информации о распределении мажоритарного класса, тогда как перевыборка, особенно основанная на синтетических данных, может способствовать переобучению и искажению геометрии пространства признаков, в том числе за счёт размытия истинных границ между классами.

Методы уровня алгоритма направлены на модификацию функции потерь или процедуры оптимизации. Наиболее распространённым приёмом является взвешивание классов, при котором ошибки на объектах миноритарных классов получают больший вклад в значение функции потерь. В ансамблевых алгоритмах, таких как градиентный бустинг, аналогичный эффект достигается за счёт балансировки параметров подвыборки. Преимуществом данных подходов является сохранение исходной выборки без изменений, однако выбор весов классов представляет собой дополнительный гиперпараметр, требующий настройки. При сильном дисбалансе это может приводить к нестабильности процесса обучения и ухудшению качества обобщения на объектах мажоритарного класса.

1.3.4 Фрагментарность существующих решений

Проведённый анализ показывает, что современные методы решения задач OOD-детекции и обработки дисбаланса классов носят преимущественно фрагментарный характер. Они представляют собой совокупность разрозненных приёмов, направленных на коррекцию последствий фундаментальных ограничений стандартных моделей классификации, а не на устранение их причин. Методы обнаружения выхода за распределение реализуются в виде внешних инструментов, требующих отдельной настройки и калибровки, тогда как подходы к борьбе с дисбалансом либо искажают исходное распределение данных, либо вводят дополнительные гиперпараметры в процедуру обучения.

Ключевым недостатком существующих решений является отсутствие единого статистически обоснованного фундамента, учитывающего неопределённость предсказаний. Эта неопределённость может быть обусловлена внутренней стохастической природой данных или ограниченностью объёма обучающей выборки и модели. Большинство подходов не предусматривает встроенного, теоретически обоснованного механизма отказа от классификации, способного корректно реагировать на обе формы неопределённости. Это обстоятельство обуславливает необходимость разработки нового подхода, в котором решение указанных проблем было бы интегрировано непосредственно в математическую модель классификатора, обеспечивая согласованность и статистическую корректность его поведения в условиях неопределённости. Разработка такого подхода для задач с ограниченным объёмом данных является целью настоящего исследования.

1.4 Выводы

Рассмотренные в главе методы классификации демонстрируют значительное разнообразие архитектурных решений и высокий уровень развития с точки зрения аппроксимационных возможностей и предсказательной точности. Однако в контексте парадигмы доверенного искусственного интеллекта становится очевидным, что усложнение моделей и рост их выразительной способности сами по себе не приводят к удовлетворению ключевых требований, предъявляемых к надёжным системам принятия решений. Напротив, увеличение сложности зачастую сопровождается утратой прозрачности, ослаблением статистической интерпретации предсказаний и отсутствием формализованного контроля над областью применимости модели.

Современные классификаторы, включая ансамблевые методы и глубокие нейросетевые модели, ориентированы преимущественно на минимизацию эмпирического риска в рамках обучающего распределения. При этом поведение модели за пределами этой области, а также в условиях высокой неопределённости, как правило, не регламентировано. В результате система может демонстрировать высокую уверенность в заведомо некорректных предсказаниях, что принципиально несовместимо с требованиями доверенно-

го искусственного интеллекта. Аналогично, распространённые приёмы работы с дисбалансом классов и выходом за распределение не формируют целостного механизма управления неопределённостью, а выступают в роли внешних корректирующих процедур.

Таким образом, существующие подходы не обеспечивают согласованного и статистически обоснованного ответа на ключевой вопрос доверенного искусственного интеллекта: когда модели следует воздержаться от принятия решения. Отсутствие встроенных механизмов определения границ собственной компетенции и формализованного отказа от классификации указывает на фундаментальное несоответствие между доминирующей практикой построения высокоточных моделей и требованиями к надёжности, предсказуемости и безопасности интеллектуальных систем.

Это обстоятельство подчёркивает необходимость перехода от эвристических и аддитивных решений к методам, в которых обработка неопределённости, асимметрии данных и выхода за носитель распределения является неотъемлемой частью математической модели классификатора. Такой подход позволяет рассматривать отказ от классификации не как побочный эффект или внешнюю надстройку, а как естественный элемент процедуры принятия решения, что особенно важно для задач с ограниченным объёмом данных и повышенными требованиями к доверию и ответственности принимаемых решений.

Структура дальнейшего изложения направлена на последовательное решение обозначенных проблем. В главе 2 представлен модифицированный байесовский классификатор, снабжённый механизмом отказа на основе оценки неопределённости, что является теоретической основой для создания доверенных систем и соотносится с первым и вторым положениями на защиту. Глава 3 развивает этот подход, предлагая метод унарной классификации, устойчивый к дисбалансу и позволяющий работать в условиях высокой неопределённости, что напрямую поддерживает третье защищаемое положение. Глава 4 демонстрирует практическое применение унарного классификатора для генерации синтетических данных, расширяя сферу его использования. Наконец, глава 5 описывает интеллектуальную систему машинного обучения, которая реализует предложенные методы и обеспечивает их исследование и визуализацию, что соответствует четвёртому положению на защиту.

Глава 2. Модифицированный байесовский классификатор

2.1 Бинарная классификации и байесовский классификатор

Пусть $D = (X, Y)$ – случайный вектор с некоторым распределением P, причём $X \in [0,1]^d$ и $Y \in \{\pm 1\}$. Обозначим отвечающее P распределение X через P_X . В дальнейшем будем называть значения X признаками, Y – метками классов, а d – размерностью признакового пространства. Задача бинарной классификации заключается в построении дискриминантной функции $f\colon [0,1]^d \to \{\pm 1\}$, которая значениям признаков ставит в соответствие метки классов.

Общую задачу классификации можно записать в следующем виде:

$$\mathbb{P}(Y \neq f(X)) \to \min_f, \tag{2.1}$$

где минимум берется по всем функциям со значениями ± 1. Аналогично, в этих терминах задача регрессии принимает вид

$$\mathbb{E}\left(Y - f(X)\right)^2 \to \min_f, \tag{2.2}$$

где $\mathbb{E}$ – отвечающее P математическое ожидание, а минимум берётся по всем функциям на $[0,1]^d$.

Решение задачи регрессии – это условное математическое ожидание

$$g(x) = \mathbb{E}\left(Y | X = x\right) = 2\mathbb{P}(Y = 1 | X = x) - 1, x \in [0,1]^d,$$

как следует из соотношения

$$\mathbb{E}\left(Y - f(X)\right)^2 = \mathbb{E}\left(Y - g(X)\right)^2 + \mathbb{E}\left(g(X) - f(X)\right)^2.$$

Вообще говоря, $g(x)$ определено однозначно на $[0,1]^d$ только P_X-почти наверное. В частности, вне $\mathbb{S}$ – носителя распределения вектора X в $[0,1]^d$ – функция условного математического ожидания $g(x)$ может принимать какие угодно значения.

Решение задачи классификации – это байесовский классификатор [25]

$$s(x) = \begin{cases} 1, & \text{если } g(x) > 0 \text{ и } x \in \mathbb{S}, \\ \text{любое из значений } \pm 1, & \text{если } g(x) = 0 \text{ или } x \notin \mathbb{S}, \\ -1, & \text{если } g(x) < 0 \text{ и } x \in \mathbb{S}, \end{cases} \tag{2.3}$$

отвечающий g (данной версии условного математического ожидания). Последнее следует из того, что для f со значениями ± 1 всегда выполнены равенства

$$4\mathbb{P}(Y \neq f(X)) = \mathbb{E}\,(Y - f(X))^2 = \mathbb{E}\,(Y - g(X))^2 + \mathbb{E}\,(g(X) - f(X))^2$$

Согласно (2.3), зоной неопределённости байесовского классификатора, отвечающего g, является множество $[0,1]^d \setminus \mathbb{S} \cup \{x : g(x) = 0\}$.

На практике распределение P неизвестно, но при этом, как правило, имеется выборка из P, так что для оценки байесовского классификатора используются эмпирические аналоги (2.1) и (2.2) с регуляризацией [26] и различными ограничениями на классы функций f, по которым ведётся оптимизация.

2.2 Проблематика

Ключевая трудность, с которой сталкиваются методы машинного обучения [27], заключается в том, что как этап обучения, так и последующие выводы обоснованы лишь в пределах носителя распределения имеющихся данных. Как было отмечено ранее, область вне носителя $\mathbb{S}$ распределения случайного вектора X представляет собой зону неопределённости для байесовского классификатора. Однако распространённые алгоритмы машинного обучения, как правило, не осуществляют явную оценку границ множества $\mathbb{S}$, формируя при этом конкретные правила классификации на всём компакте $[0,1]^d$, включая точки, лежащие вне $\mathbb{S}$. При наличии сдвигов или искажений в распределении данных (как в обучающей, так и тестовой выборках) такие выводы за пределами $\mathbb{S}$ могут оказаться некорректными. В этих случаях естественным решением является отказ от классификации, однако большинство современных методов не обладают встроенными механизмами для автоматического отказа от принятия решения, что снижает их надёжность в прикладных задачах.

Рассмотрим подробнее ситуации, в которых отказ от принятия решения является обоснованным.

1. **Выброс.** Если наблюдение существенно отличается от всех прочих, то модель не располагает достаточной информацией для корректной классификации. Обычно для обнаружения таких объектов применяются специальные процедуры предварительной обработки, ориентированные на выявление выбросов [28]. Однако эти методы, как правило, требуют задания гиперпараметров [29] и применяются перед обучением модели, что не позволяет гибко учитывать особенности распределения обучающих и тестовых данных.
2. **Выход за распределение (OOD, out-of-distribution).** При изменении распределения входных данных модель может оказаться неспособной дать обоснованное решение [30]. Существующие подходы к детекции подобных случаев делятся на три класса: статистические методы [31], моделирование сдвигов [32] и применение вспомогательных моделей машинного обучения [33]. Статистические методы отличаются высокой чувствительностью к выбору конкретного подхода и параметров. Моделирование сдвигов требует априорных предположений о характере изменений распределения и его динамике во времени, что затрудняет автоматизацию. Методы на основе машинного обучения сами подвержены проблеме выхода за распределение, но уже применительно к детектору.
3. **Зона пересечения классов.** Если носители распределений нескольких классов пересекаются, то для новых наблюдений, попавших в такую область, вероятности принадлежности к разным классам могут быть примерно равны. В этом случае разумно отказаться от автоматической классификации и передать наблюдение на рассмотрение эксперту, обладающему дополнительной информацией.

Таким образом, отказ от классификации представляется оправданным в зоне неопределённости, а именно для наблюдений, принадлежащих множеству $[0,1]^d \setminus \mathbb{S} \cup \{x : g(x) = 0\}$. Главная трудность заключается в том, что множество $\mathbb{S}$ априорно неизвестно. В следующем разделе рассматривается подход, позволяющий обойтись без явной оценки носителя $\mathbb{S}$.

2.3 Модификация байесовского классификатора

Одним из возможных подходов к преодолению указанной проблемы является модификация байесовского классификатора путём экстраполяции его поведения за пределы носителя $\mathbb{S}$. Такая экстраполяция достигается за счёт добавления к обучающей выборке искусственных наблюдений, компоненты которых равномерно распределены на всём компакте $[0,1]^d$, а метки классов фиксированы и равны нулю [3].

В результате этой модификации исходное распределение случайного вектора (X, Y), принимающего значения в пространстве $[0,1]^d \times \{\pm 1\}$, заменяется на новое распределение на $[0,1]^d \times \{-1, 0, +1\}$, представляющее собой смесь двух распределений:

$$P_\alpha = (1-\alpha)P + \alpha\hat{P},$$

где $\alpha \in (0,1)$, P – исходное распределение обучающих данных, $\hat{P}$ – распределение, при котором вектор признаков равномерно распределён на $[0,1]^d$, а метка класса тождественно равна нулю.

Соответствующее маргинальное распределение признаков X при этом принимает следующий вид:

$$\lambda_\alpha = (1-\alpha)P_X + \alpha\lambda,$$

где λ – мера Лебега на $[0,1]^d$, а P_X – распределение признаков X, когда вектор (X, Y) распределён согласно P. Обозначим через $\mathbb{E}_\alpha$ математическое ожидание относительно распределения P_α, а через $\mathbb{S}$ – носитель распределения P_X.

В силу разложения Лебега и теоремы Радона–Никодима всегда найдутся неотрицательная интегрируемая функция ρ на $[0,1]^d$ и борелевское множество $A \subseteq \mathbb{S}$ нулевой лебеговой меры такие, что

$$P_X(B) = \int_B \rho(x)dx + P_X(A \cap B)$$

для всех борелевских множеств B в $[0,1]^d$.

Теорема 1. *Для всякого $\alpha \in (0,1)$ решение g_α задачи регрессии*

$$\mathbb{E}_\alpha \left(Y - f(X)\right)^2 \to \min_f \tag{2.4}$$

существует, это решение единственно P_X- и λ-п. н. и может быть задано формулой

$$g_\alpha(x) = \begin{cases} g(x), & \text{если } x \in A, \\ \dfrac{(1-\alpha)g(x)\rho(x)}{\alpha + (1-\alpha)\rho(x)}, & \text{если } \rho(x) > 0 \text{ и } x \in \mathbb{S} \setminus A, \\ 0, & \text{если или } \rho(x) = 0 \text{ и } x \in \mathbb{S} \setminus A, \text{ или } x \notin \mathbb{S}, \end{cases} \tag{2.5}$$

здесь минимум берется по всем (борелевским) функциям f и

$$g(x) = \mathbb{E}\left(Y | X = x\right) \text{ на } [0,1]^d.$$

При этом классификатор $s_\alpha = s_\alpha(x)$, $x \in [0,1]^d$, заданный формулой (2.3) с заменой g на любое решение g_α задачи (2.4) и $\mathbb{S}$ на $[0,1]^d$, обладает следующими свойствами:

- *(i). s_α реализует минимум в задаче классификации*

$$\mathbb{P}(Y \neq f(X)) \to \min_f,$$

  где минимум берется по всем (борелевским) функциям со значениями ± 1;
- *(ii). зоной неопределённости s_α является множество $\{x \in [0,1]^d : g_\alpha(x) = 0\}$, которое покрывает λ-п.н. множество $[0,1]^d \setminus \mathbb{S}$, где $\mathbb{S}$ – носитель распределения P_X.*

Доказательство теорем. 1 приведено в приложении Б.1.

Пусть вместо (2.4) рассматривается задача вида

$$\mathbb{E}_\alpha \left(Y - f(X)\right)^2 + Pen(f) \to \min_{f \in \mathcal{F}}, \tag{2.6}$$

где $\mathcal{F}$ – некоторое параметрическое семейство функций (к примеру, нейросетей заданной архитектуры), а $Pen(f)$ – регуляризационное слагаемое-штраф.

Тогда, как следует из доказательства теорем. 1, в терминах минимизирующих функций задача (2.6) будет эквивалентна задаче приближения g_α – любого решения задачи (2.4) – функцией из класса $\mathcal{F}$ с учётом штрафа $Pen(f)$:

$$(1-\alpha)\|g_\alpha - f\|^2_{P_X} + \alpha\|g_\alpha - f\|^2_{\lambda} + Pen(f) \to \min_{f\in\mathcal{F}}, \tag{2.7}$$

где $\|\cdot\|_\mu$ – это L_2-норма относительно меры P_X или λ.

Как было отмечено ранее (см. раздел 2.1), в практических задачах распределение P априорно неизвестно, а классификационное правило строится по конечной выборке, представляющей реализацию этого распределения. В таких условиях задача классификации заменяется на эмпирический аналог задачи (2.6), решаемый с использованием методов машинного обучения.

Поскольку в эмпирической постановке оценка функции принятия решения подвержена статистическим флуктуациям, область отказа от классификации следует расширить, чтобы учесть возможную неопределённость вблизи границы между классами. Для этого вводится дополнительный гиперпараметр $\beta > 0$, регулирующий ширину зоны неопределённости [34; 35]. Отказ от принятия решения осуществляется для тех наблюдений, по которым оценка эмпирической функции f, соответствующей приближённому решению задачи (2.6), по модулю не превосходит β: $|f(x)| \leqslant \beta$.

Такое уточнение позволяет повысить надёжность классификатора за счёт уменьшения числа потенциально ошибочных решений в областях, где уверенность модели недостаточна.

Параметр β имеет ясную интерпретацию: он определяет минимальный уровень уверенности классификатора, при котором принимается решение. В предельном случае $\beta = 0$ отказ от классификации осуществляется только тогда, когда значение $f(x)$ точно равно нулю, что соответствует ситуации, в которой размер обучающей выборки стремится к бесконечности, то есть распределение P считается полностью известным.

2.4 Аппроксимация байесовского классификатора

Для построения классификатора, приближающего оптимальное байесовское правило, предполагается наличие размеченного обучающего набора $(X, Y) = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$, состоящего из n независимых наблюдений. Каждое наблюдение включает вектор признаков $X_i \in [0,1]^d$ и бинарную метку класса $Y_i \in \{-1, +1\}$. Предполагается, что признаки заданы в евклидовом

пространстве фиксированной размерности d, что позволяет применять как метрические, так и нейросетевые методы приближения. В данном разделе рассматриваются различные подходы к аппроксимации байесовского классификатора, включая классические методы и нейросетевые модели, обладающие способностью к адаптации и масштабной инвариантности.

2.4.1 Аппроксимация классическими методами

Одним из подходов к приближению байесовского классификатора является использование классических непараметрических и параметрических методов [36; 37]. Эти методы позволяют получить приближение к оптимальной функции принятия решения и часто служат базой для анализа свойств более сложных моделей.

Наиболее простым из них выступает построение гистограммы [38]. Пространство признаков $[0,1]^d$ разбивается на конечное число ячеек (например, гиперкубов одинакового объёма), в каждой из которых оценивается условное распределение метки класса Y по наблюдаемым примерам. Полученная функция классификации будет иметь вид ступенчатой функции, принимающей значение класса с наибольшей эмпирической вероятностью в каждой ячейке. Однако точность метода существенно зависит от выбора размера ячейки и неустойчива к локальным вариациям плотности данных. Кроме того, для вычисления эмпирических вероятностей требуется хранение всей обучающей выборки, а число необходимых ячеек экспоненциально возрастает с размерностью пространства признаков, что делает метод крайне неэффективным уже при умеренных значениях d.

Более гибким методом является алгоритм k ближайших соседей (kNN). Для классификации новой точки $x \in [0,1]^d$ выбираются k ближайших к ней объектов из обучающего множества, а прогноз определяется как знак суммы их меток:

$$c_n(x) = \operatorname{sign}\left(\sum_{i \in \mathcal{N}_k(x)} Y_i\right),$$

где $\mathcal{N}_k(x)$ – множество индексов k ближайших к x точек в обучающей выборке. Метод обладает асимптотической состоятельностью [39] при $k \to \infty$ и $k/n \to 0$, но на практике чувствителен к выбору метрики и параметра k. В случае сложных или структурированных признаков, определение подходящей метрики может быть затруднено или неочевидно.

Одним из распространённых непараметрических подходов также является ядерная оценка условного распределения. Предполагается, что функция плотности распределения оценивается с помощью сглаживающего ядра K, а классификационное решение принимается на основе усреднённой метки с весами, зависящими от расстояния между точкой x и наблюдениями:

$$\hat{\eta}(x) = \frac{\sum_{i=1}^{n} K_h(x - X_i) Y_i}{\sum_{i=1}^{n} K_h(x - X_i)}, \quad c_n(x) = \operatorname{sign}(\hat{\eta}(x)),$$

где $K_h(u) = \frac{1}{h^d} K(u/h)$ – ядро с шириной сглаживания h. Выбор ядра и параметра h существенно влияет на результат классификации. Метод обладает хорошими аппроксимирующими свойствами, но страдает от “проклятия размерности“ и требует осторожной настройки [40; 41].

Переходя к параметрическим методам, важное место занимает метод опорных векторов (Support Vector Machines, SVM), который аппроксимирует байесовский классификатор через построение оптимальной разделяющей гиперплоскости в признаковом пространстве или его нелинейном отображении [10]. В линейном случае SVM решает задачу максимизации зазора между классами:

$$\min_{w,b} \quad \frac{1}{2}\|w\|^2 \quad \text{при} \quad Y_i(w^\top X_i + b) \geqslant 1, \quad i = 1, \ldots, n.$$

Для нелинейных границ применяется замена скалярного произведения на ядро $K(x, x')$, что позволяет эффективно аппроксимировать сложные границы раздела. SVM показывает хорошие результаты на малых выборках, устойчив к выбросам при введении мягкого зазора и имеет теоретические гарантии обобщающей способности [42].

Таким образом, классические методы аппроксимации байесовского классификатора варьируются от простых гистограмм до методов, основанных на решении задач оптимизации в пространстве функций. Их использование обосновано в задачах с ограниченным объёмом данных и понятной метрикой, но в случае высокоразмерных или структурированных данных может потребоваться более гибкая модель.

2.4.2 Аппроксимация равномерно непрерывной функцией

Пусть $c(X) : [0,1]^d \to \mathbb{R}$ – равномерно непрерывная функция на $[0,1]^d$, с помощью которой будем приближать байесовский классификатор. Рассмотрим задачу среднеквадратичной аппроксимации:

$$\mathbb{E}\left(c(X) - Y\right)^2 \to \min_{c(X)}. \tag{2.8}$$

Поскольку

$$\mathbb{E}\left(c(X) - Y\right)^2 = \mathbb{E}\left(c(X) - g(X) + g(X) - Y\right)^2 \to$$

$$\mathbb{E}\left(c(X) - Y\right)^2 = \mathbb{E}\left(c(X) - g(X)\right)^2 + \mathbb{E}\left(g(X) - Y\right)^2$$

и второе слагаемое не зависит от $c(X)$, задача (2.8) сводится к аппроксимации функции регрессии:

$$\mathbb{E}\left(c(X) - g(X)\right)^2 \to \min_{c(X)}, \tag{2.9}$$

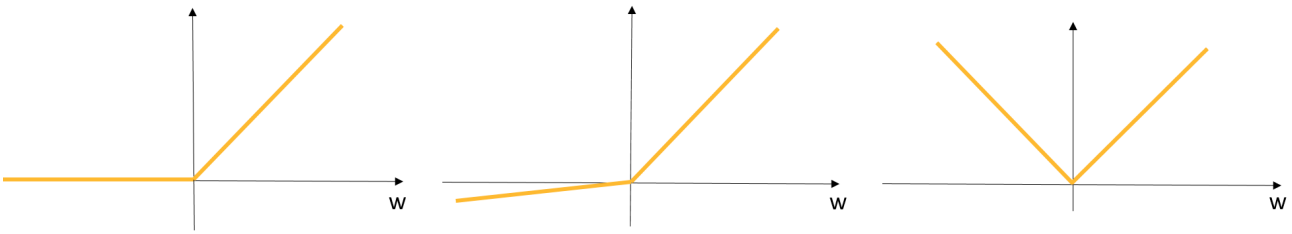
2.4.3 Нейросетевая аппроксимация

Возьмём в качестве $c(X)$ многослойный персептрон [43] (полносвязную нейронную сеть) с d-мерным входным слоем, состоящий из L скрытых слоёв по k нейронов с кусочно-линейной функцией активации $\sigma(x)$, например, ReLU, LeakyReLU, Abs (рисунок 2.1) в каждом и выходным слоем из одного нейрона. Согласно теореме об универсальной аппроксимации для нейронных сетей с неполиномиальной функцией активации [44] для любого заданного $\varepsilon > 0$ существуют такие значения параметров персептрона L и k, что для любого $x \in [0,1]^d$ выполняется условие:

$$\sup_{x \in [0,1]^d} |c(x) - g(x)| < \varepsilon.$$

То есть теоретически ε-приближенное решение задачи (2.9) существует.

Пусть выборка (X, Y) имеет на $\mathbb{S}$ равномерно непрерывную плотность $f(X)$:



а) ReLU(w) = $max(0, w)$ б) LeakyReLU(w) = $max(0.01w, w)$ в) Abs(w) = $|w|$

Рисунок 2.1 — Кусочно-линейные функции активации

$$f(X) = p_{-1}f_{-1}(X) + p_{+1}f_{+1}(X),$$

где f_{-1} и f_{+1} – плотности классов -1 и $+1$ соответственно.

Для формирования выборки из смеси реальных данных и “фона“ с плотностью $\alpha f(X) + (1 - \alpha)p(X)$ добавим к этой выборке искусственно сгенерированные данные $\{(X_{n+1}, Y_{n+1}), \ldots, (X_{2n}, Y_{2n})\}$, где векторы $\{X_{n+1}, \ldots, X_{2n}\}$ – наблюдения независимо равномерно распределённых на $[0, 1]^d$ случайных векторов с плотностью $p(X)$, а $Y_{n+i} = 0, i = 1..n$.

Пусть $C(L, k)$ – множество всех многослойных персептронов $c(X)$ с одним нейроном с линейной функцией активации в выходном слое, кусочно-линейной функцией активации $|\cdot|$ (модульная) в скрытых слоях и числом L и размером k скрытых слоёв.

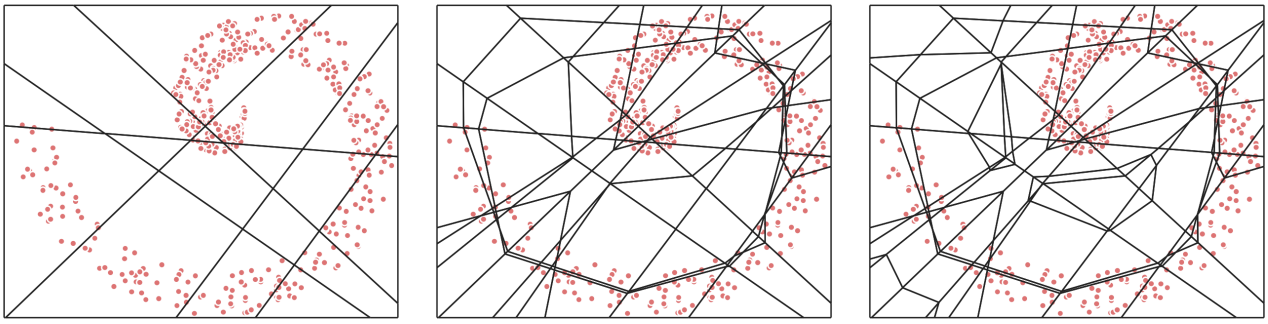
Применяя некоторый алгоритм оптимизации (градиентный спуск [45], генетический алгоритм [46] и т.д.), построим выборочную оценку решения задачи (2.9):

$$\sum_{i=1}^{2n}(c_n(X_i) - Y_i)^2 \to \min_{c_n(X)\in C(L,k)}, \tag{2.10}$$

где параметры L и k выбраны оптимально с учётом ограничений, связанных с переобучением.

Пусть функция $c_n^*(X)$ – решение оптимизационной задачи (2.10), которая в дальнейшем будет называться **функцией нейросетевой регрессии**. Соответствующий этому решению персептрон строит иерархическое (по слоям) разбиение компакта $[0, 1]^d$ на $O(k^{dL})$ непересекающихся ячеек [9] (при $k > d$).

Пример такого разбиения показан на рисунке 2.2, где персептрон имеет $L = 2$ скрытых слоя по $k = 6$ нейронов.



а) Ячейки первого слоя: 18 б) Ячейки второго слоя: 121 в) Ячейки выходного слоя: 148

Рисунок 2.2 — Пример разбиения некоторым персептроном с $L = 2$, $k = 6$

2.4.4 Адаптивная гистограммная аппроксимация

Пусть в результате построения $c_n^*(X)$ получено разбиение компакта $[0,1]^d$ на N непересекающихся ячеек $\{K_1, K_2, \ldots, K_N\}$. Рассмотрим кусочно-постоянную (в общем случае разрывную) **функцию гистограммной регрессии** $h_n(X)$, принимающую постоянные значения в ячейках разбиения $[0,1]^d$ и решим для неё оптимизационную задачу:

$$\sum_{i=1}^{2n}(h_n(X_i) - Y_i)^2 \to \min_{h_n(X)}, \tag{2.11}$$

Пусть $X \in K_r$. Тогда задачу (2.11) для этой ячейки можно представить в следующем виде:

$$n_{-1}(X)\cdot(h_n(X)+1)^2+n_0(X)\cdot(h_n(X)-0)^2+n_{+1}(X)\cdot(h_n(X)-1)^2 \to \min_{h_n(X)}, \tag{2.12}$$

где $n_j = \sum\limits_{i=1}^{2n} I_{X_i \in K_r, Y_i = j}$.

После дифференцирования функции (2.12) по $h_n(X)$ получаем решение задачи (2.11):

$$h_n^*(X) = \frac{n_{+1}(X) - n_{-1}(X)}{n_{-1}(X) + n_0(X) + n_{+1}(X)}. \tag{2.13}$$

Пример вычисления функции гистограммной регрессии показан на рисунке 2.3.

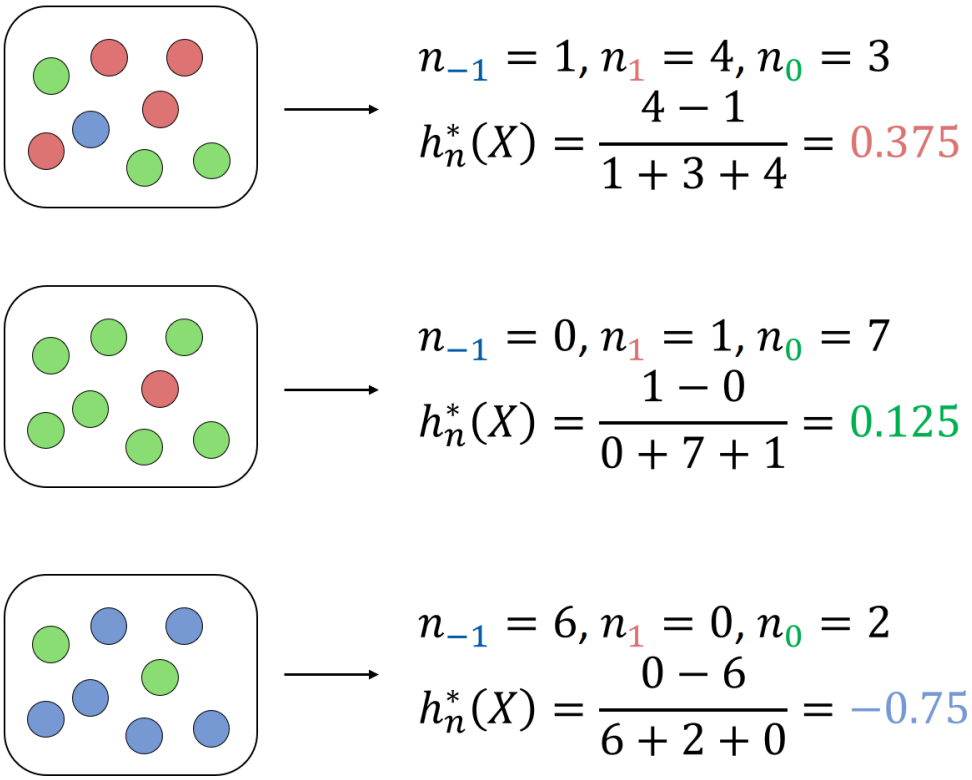


Рисунок 2.3 — Пример вычисления $h_n^*(X)$ в некоторой ячейке K_r

Основными достоинствами использования такой аппроксимации являются независимость от масштаба и отсутствие необходимости введения метрик, как того требуют методы на основе расстояний вроде k ближайших соседей.

2.5 Объясняющее двоичное дерево eXBTree

2.5.1 Построение объясняющего дерева решений

Как было сказано в разделе 2.4.3, многослойный персептрон с кусочно-линейной функцией активации разбивает входное пространство признаков на N непересекающихся ячеек $\{K_1, K_2, \ldots, K_N\}$. В каждой такой ячейке значение выходного нейрона определяется фиксированной линейной комбинацией признаков.

Рассмотрим полносвязный персептрон с L скрытыми слоями по k нейронов в каждом и одним выходным нейроном. В качестве функции активации

скрытых слоёв используется модульная функция:

$$\sigma(z) = |z|.$$

Обозначим выходы нейронов первого скрытого слоя через A_i, второго – через B_i, третьего – через C_i, и так далее (рисунок 2.4). Для произвольного нейрона слоя A выполняется:

$$A_i = \sum_{j=1}^{d} a_{ij} x_j + a_{i0},$$

где $x \in [0,1]^d$ – входной вектор признаков. Аналогично, для следующего слоя:

$$B_i = \sum_{j=1}^{k} b_{ij} |A_j| + b_{i0},$$

и так далее до выходного слоя:

$$c_n(x) = \sum_{j=1}^{k} c_j |B_j| + c_0.$$

Каждый нейрон разбивает своё входное пространство на две области: одну, в которой входная сумма положительна (в этом случае модуль раскрывается со знаком «плюс»), и другую, где сумма отрицательна (в этом случае модуль раскрывается со знаком «минус»). Равенство нулю считается переходом со знаком «плюс». Таким образом, на каждом шаге можно заменить выражение с функцией активации линейным выражением с соответствующим знаком.

Раскрывая функцию активации в нейронах первого слоя, можно сформировать дерево, в котором каждый путь соответствует определённой комбинации знаков раскрытия модулей. В узлах дерева находятся неравенства, задаваемые условиями перехода: при переходе по левой ветви знак раскрытия модуля в текущем нейроне отрицателен, по правой – положителен. После обработки всех нейронов слоя A все функции активации будут раскрыты, и входы к следующему слою B становятся кусочно-линейными выражениями, зависящими от исходных переменных x_j, и процесс повторяется.

Таким образом, можно построить объясняющее двоичное дерево решений [47], в дальнейшем называемое **eXBTree** (eXplanatory Binary Tree), в котором каждая вершина соответствует разбиению пространства по линейному неравенству одного нейрона, а каждая листовая вершина – конечной линейной

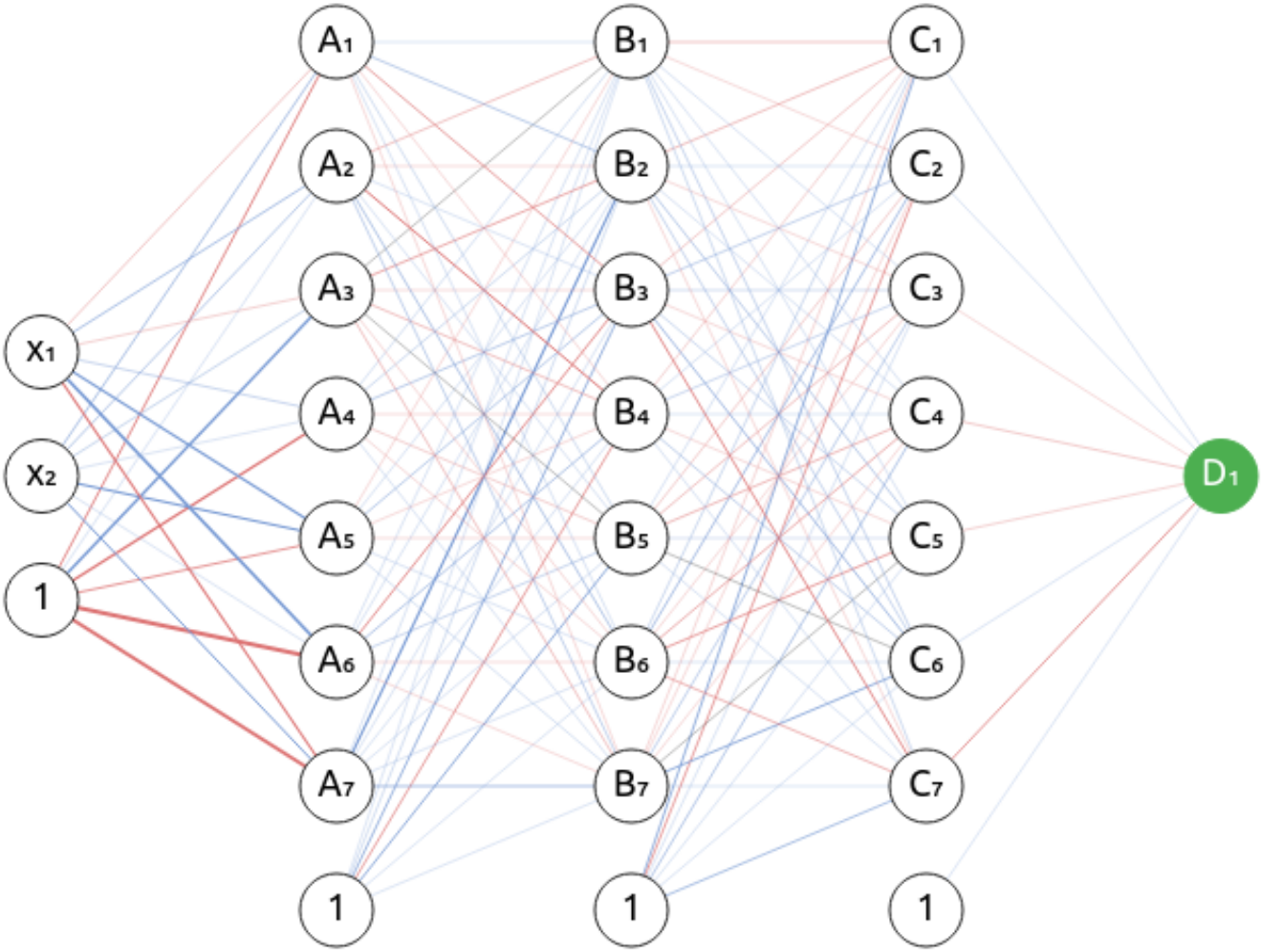
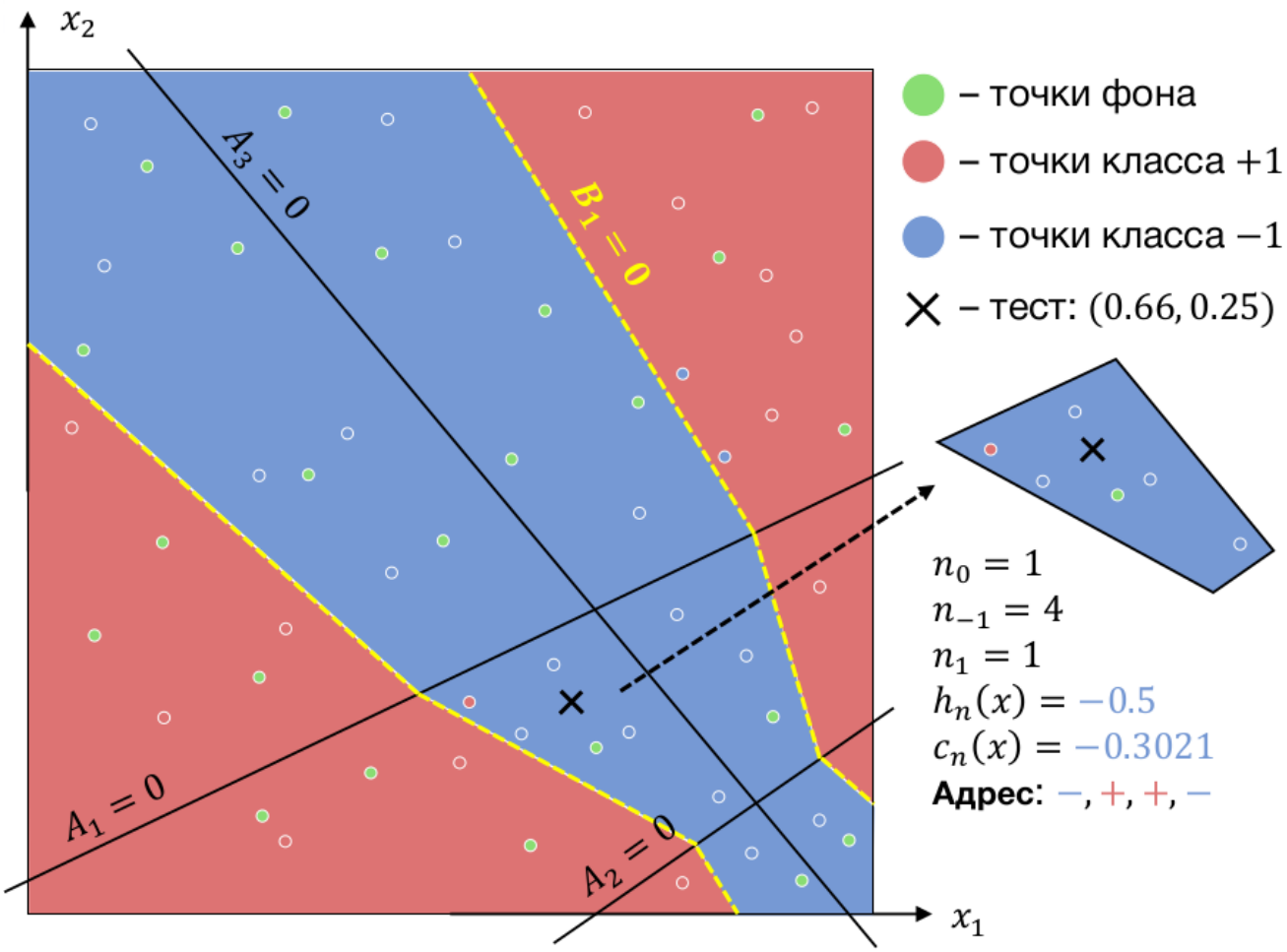


Рисунок 2.4 — Архитектура многослойного персептрона с $d = 2$, $L = 3$, $k = 7$

функции выходного слоя, полученной на конкретной ячейке пространства. Последовательность знаков, с которыми раскрывались активационные функции нейронов по пути от входа к выходному нейрону кодирует произвольную ветку в построенном дереве (рисунок 2.5).

Наличие такой структуры позволяет не только интерпретировать классификацию конкретного наблюдения (путь через дерево), но и формировать иерархию разбиений пространства, объединяя соседние ветви дерева на разных уровнях. Это открывает возможности для оценки доверия и анализа прецедентов – как в пределах отдельной ячейки, так и на объединённых областях пространства.



$$A_1 = -0.4x_1 + 0.8x_2 \qquad A_1(x) = -0.064 < 0 \rightarrow -$$
$$A_2 = -1.2x_1 + 1.8x_2 + 0.8 \qquad A_2(x) = 0.458 > 0 \rightarrow +$$
$$A_3 = -1.5x_1 - 1.3x_2 + 1.5 \qquad A_3(x) = 0.185 > 0 \rightarrow +$$

$$B_1 = 0.6|A_1| - 0.5|A_2| + 2.1|A_3| - 0.5 \qquad B_1(x) = -0.3021 < 0 \rightarrow -$$

Рисунок 2.5 — Пример eXBTree на основе персептрона с $d = 2$, $L = 1$, $k = 3$

2.5.2 Комбинаторная сложность и практическая реализация eXBTree

Теорема 2 (О верхней оценке сложности построения полного объясняющего дерева для многослойного персептрона)**.** *Рассмотрим многослойный персептрон с d-мерным входом, L скрытыми слоями, каждый из которых содержит k нейронов ($k > d$), и одним выходным нейроном. Пусть все скрытые нейроны используют кусочно-линейную функцию активации $|\cdot|$: $\sigma(x) = |x|$. Тогда временная сложность алгоритма построения полного объясняющего двоичного*

дерева (eXBTree), которое точно представляет функцию, вычисляемую персептроном, в худшем случае составляет $\mathcal{O}(k^{dL})$.

Доказательство. Каждый узел eXBTree соответствует проверке знака линейного выражения $z_i^{(l)}(x)$ перед применением активационной функции (l,i). Полный путь от корня к листу задаёт систему линейных неравенств, множество решений которой (если оно непусто) является одним из линейных регионов (ячеек K_j), на которые сеть разбивает входное пространство $\mathbb{R}^d$. На каждом таком регионе выход сети линеен. Таким образом, листья eXBTree находятся во взаимно однозначном соответствии с линейными регионами сети.

Для сети с L слоями по k нейронов с кусочно-линейной активацией максимальное число линейных регионов известно [48] и оценивается как $\mathcal{O}(k^{dL})$. На построение каждого листа (региона) алгоритм тратит полиномиальное относительно k, L и d время на проверку совместности неравенств и вычисление итоговой линейной функции. Следовательно, общая сложность есть $\mathcal{O}(k^{dL})$.

Теорема доказана. □

Теорема 3 (О временной сложности получения прогноза по дереву eXBTree)**.** *Пусть* T *– полное объясняющее дерево (eXBTree), построенное для многослойного персептрона с* L *скрытыми слоями по* k *нейронов и входной размерностью* d*. Тогда временная сложность получения прогноза для нового наблюдения* $x \in \mathbb{R}^d$ *по дереву* T *составляет* $\mathcal{O}(d \cdot (kL + 1))$.

Доказательство. Прогноз по дереву T осуществляется обходом от корня до листа. В каждом внутреннем узле дерева проверяется знак линейной комбинации вида:

$$w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + b,$$

где $w \in \mathbb{R}^d$, $b \in \mathbb{R}$ – параметры, соответствующие одному из нейронов исходной сети. Вычисление значения этой комбинации для входного вектора x требует вычисления скалярного произведения $w^\top x$ и добавления свободного члена b, что выполняется за $\mathcal{O}(d)$ операций.

Глубина полного дерева T в точности равна общему числу скрытых нейронов в сети, то есть kL. При переходе от корня к листу алгоритм выполняет проверку в каждом из kL внутренних узлов на пути. Следовательно, общее

время вычисления прогноза оценивается как:

$$\underbrace{kL}_{\text{число узлов}} \times \underbrace{\mathcal{O}(d)}_{\text{сложность проверки в узле}} = \mathcal{O}(d \cdot kL).$$

В листовой вершине хранится линейная функция $c^\top x + c_0$, вычисление которой также требует $\mathcal{O}(d)$ операций. Эта добавка не меняет асимптотическую оценку $\mathcal{O}(d \cdot kL)$.

Теорема доказана. □

Тем не менее, в рамках задач классификации или аппроксимации интерес представляют не все возможные ячейки, а лишь те, в которые попали обучающие (или тестовые) наблюдения. Таким образом, нет необходимости в полном построении дерева. Достаточно определить множество фактически реализованных путей, то есть таких комбинаций знаков при раскрытии модулей, которые соответствуют реально встречающимся входным точкам.

Это приводит к следующему практическому алгоритму: каждое наблюдение при прямом проходе через сеть порождает набор знаков уравнений нейронов (до применения активационной функции), который можно трактовать как адрес ячейки. Хранить необходимо лишь такие уникальные адреса, тем самым получая компактное и эффективное представление разбиения пространства, ограниченное данными. Если при последующем анализе наблюдение попадает в новую ячейку, то такое наблюдение считается недоверенным и требует дополнительного анализа (с возможным последующим дообучением персептрона на нём). Для хранения такого дерева требуется только оригинальный персептрон и словарь, в котором ключами выступают последовательности знаков выходов нейронов, а значениями счётчики точек каждого класса. А временная сложность получения прогноза по дереву совпадает с получением прогноза персептрона (для получения адреса ячейки).

2.5.3 Геометрический анализ построенного дерева

Рассмотрим свойства дерева, построенного на основе модифицированного обучающего множества с фоновыми наблюдениями, описанного ранее. В каждом внутреннем узле такого дерева содержится линейное неравенство,

возникающее из перехода через гиперплоскость активации некоторого нейрона персептрона $c_n^*(X)$. Каждая вершина дерева соответствует определённой комбинации знаков выражений вида $a_i^\top x + a_0$ и, следовательно, описывает подмножество признакового пространства – выпуклый многогранник, ограниченный системой линейных неравенств.

В каждом листе дерева подсчитывается число объектов обучающего множества, попавших в соответствующую ячейку, с разбиением по классам: n_{-1}, n_0 и n_{+1}. Таким образом, каждый лист фактически содержит гистограмму классов, обсуждавшуюся в разделе 2.4.4. Эти гистограммы позволяют оценивать апостериорные вероятности классов в пределах каждой ячейки и выявлять области с высокой или низкой степенью уверенности модели.

При этом следует учитывать статистическую надёжность оценок, получаемых в отдельных ячейках. Для повышения устойчивости апостериорных оценок целесообразно в первую очередь анализировать ячейки, содержащие не менее некоторого минимального числа наблюдений. В частности, практическим эмпирическим правилом математической статистики является требование наличия не менее пяти наблюдений в выборке для получения осмысленных частотных оценок [49]. Ячейки, в которых число объектов обучающего множества меньше данного порога, следует рассматривать как статистически ненадёжные, а ячейки, содержащие единичные наблюдения, – как недоверенные. Такое ограничение позволяет снизить влияние случайных флуктуаций выборки и повысить достоверность интерпретации структуры дерева в терминах вероятностных характеристик классов.

Полученное дерево может быть также рассмотрено как дерево решений с линейными функциями разделения в узлах [50], в отличие от традиционных деревьев, в которых узлы соответствуют пороговым условиям вида $x_j < c$. Такое представление делает поведение многослойного персептрона интерпретируемым: каждый путь от корня до листа соответствует системе линейных неравенств, описывающих область пространства, где модель принимает определённое решение линейным образом.

Важно подчеркнуть, что данная структура обеспечивает интерпретируемость модели в геометрических терминах, что традиционно считается слабой стороной нейронных сетей [51]. В частности, можно явно указать, при каких линейных соотношениях между признаками модель принимает то или иное решение, и каков уровень уверенности классификатора в пределах каждой

ячейки. Это позволяет использовать построенное дерево не только как аппроксиматор функции принятия решений, но и как инструмент визуального и количественного анализа поведения модели в различных областях признакового пространства.

2.5.4 Анализ прецедентов и локальной уверенности

Для повышения доверия пользователя к решению модели важным является анализ конкретных прецедентов – обучающих объектов, попавших в ту же ячейку дерева, что и тестируемое наблюдение. Вместо представления лишь числового выхода модели (например, вероятности класса 0.98), полезно показать близкие по признаковому пространству точки из обучающей выборки, что даёт наглядное представление о локальном окружении и структуре данных. Таким образом, построенное дерево служит своеобразной псевдо-метрикой, определяющей локальную близость объектов на основе разбиения пространства признаков.

Каждый лист дерева соответствует области признакового пространства, ограниченной системой линейных неравенств. В пределах этой области подсчитывается статистика по объектам различных классов. Однако в реальных задачах, особенно в медицинских и других высокорисковых прикладных областях, объёмы доступных данных могут быть недостаточными для надёжной статистической оценки на уровне отдельных ячеек.

В таких случаях целесообразно рассматривать информацию о соседних ячейках того же уровня дерева. Соседями называются ячейки, отличающиеся значением только одного из предикатов на пути от корня. Если в рассматриваемой ячейке содержится недостаточное количество наблюдений (например, менее заданного порога $n_{\min}$), то можно агрегировать информацию с её соседями для получения более устойчивой оценки локального распределения классов.

Альтернативным подходом является подъём на уровень выше по дереву, то есть укрупнение ячейки за счёт устранения одного из условий, ограничивающих пространство. Это приводит к рассмотрению более широкой области признакового пространства, в которой ожидается большее количество обучающих объектов. Полученная таким образом укрупнённая ячейка также может

быть проанализирована с точки зрения гистограммы классов, как описано выше, обеспечивая оценку апостериорной вероятности при недостаточной локальной уверенности.

Подобная стратегия, основанная на анализе прецедентов, позволяет реализовать согласованную схему оценки доверия к решению модели: в случае низкой уверенности по статистике на текущем уровне происходит адаптивное укрупнение области анализа. Это даёт практический механизм для отказа от принятия решения в условиях недостаточной информации и одновременно повышает надёжность выводов, что особенно важно в высокорисковых прикладных задачах.

2.6 Связь нейросетевой и гистограммной аппроксимаций. Асимптотические свойства гистограммной аппроксимации

Гистограммная аппроксимация, как было показано выше, представляет собой естественный способ приближённой оценки апостериорной вероятности класса по обучающим данным. В каждой ячейке пространства признаков, определённой системой линейных неравенств, оценивается эмпирическое распределение классов на основе количества объектов, попавших в соответствующую область. В частности, выход функции гистограммной аппроксимации можно записать как

$$h_n^*(X) = \frac{n_{+1}(X) - n_{-1}(X)}{n_{-1}(X) + n_0(X) + n_{+1}(X)},$$

где $n_{-1}(X)$ и $n_{+1}(x)$ – количество объектов классов -1 и $+1$ соответственно, а n_0 – количество фоновых точек в ячейке, содержащей наблюдение X.

Таким образом, $h_n^*(X)$ служит приближением разности апостериорных вероятностей.

Предполагая, что плотности распределения классов равномерно непрерывны, можно показать, что гистограмма является строго состоятельной оценкой этих плотностей. Результаты работы [9] и авторской работы [4] позволяют утверждать, что при росте объёма обучающей выборки $n \to \infty$ и одновременном увеличении числа формируемых ячеек имеет место следующее соотношение:

$$\mathbb{E}\left(h_n^*(X) - c_n^*(X)\right)^2 \to 0, \tag{2.14}$$

где $c_n^*(X)$ – функция нейросетевой регрессии.

Этот результат обосновывает возможность замены гистограммной аппроксимации на нейросетевую, сохраняющую асимптотические свойства при существенно меньших требованиях к вычислительным ресурсам. В отличие от гистограммы, для работы персептрона не требуется хранение всей обучающей выборки или экспоненциально большого числа ячеек.

Следствием приведённого утверждения является практический критерий принятия решения на основе выхода персептрона. Поскольку $c_n^*(x)$ приближает $h_n^*(x)$, то в условиях асимптотической сходимости разумно вводить порог отказа β и принимать решение о принадлежности к одному из классов только при условии

$$|c_n^*(x)| > \beta.$$

Тем самым достигается контроль над уверенностью классификатора: чем ближе значение $c_n^*(x)$ к нулю, тем ниже надёжность предсказания. Предложенная схема позволяет реализовать отказ от ответа в ситуациях, когда классификатор не обладает достаточной апостериорной уверенностью, и одновременно существенно снижает вычислительную сложность по сравнению с прямой реализацией гистограммной аппроксимации.

2.7 Случай нескольких классов

Рассмотренные ранее методы касаются задачи бинарной классификации, когда множество допустимых меток ограничено двумя классами. Однако на практике часто возникает необходимость классификации объектов в более чем два класс [52]. Переход от бинарной к многоклассовой классификации существенно усложняет как построение, так и интерпретацию модели [53].

Существуют два стандартных подхода к решению многоклассовой задачи на основе бинарных классификаторов: стратегия **один против всех** (one-vs-rest) и стратегия **попарной классификации** (one-vs-one). В первом случае для каждого из C классов обучается отдельный бинарный классификатор, который отделяет данный класс от объединения всех остальных. Во втором случае

для каждой из $\frac{C(C-1)}{2}$ пар классов строится бинарный классификатор, различающий только эти два класса, а итоговое решение принимается, например, по большинству голосов или с использованием процедуры агрегации [54].

Обе стратегии имеют как теоретические, так и практические недостатки. В стратегии один против всех возникает проблема перекоса, связанная с несбалансированностью классов. При наличии сильно преобладающего класса классификаторы могут склоняться к частому отнесению объекта к этому классу, даже если признаки ближе к другому. Это приводит к смещению аппроксимации и, как следствие, к снижению обоснованности принятого решения. Дополнительные методы борьбы с дисбалансом, такие как дублирование редких классов или уменьшение выборки преобладающих, искажают исходное распределение данных, что затрудняет интерпретацию результатов и может приводить к потере статистической достоверности.

Стратегия попарных классификаторов, напротив, требует построения большого числа моделей, число которых растёт квадратично с числом классов. Кроме того, процедура выбора итогового класса по результатам попарных голосований может быть неоднозначной [55]: возможны случаи, при которых отсутствует чёткий победитель. При этом каждая отдельная модель опирается на подмножество данных, и совокупный результат может не учитывать общую структуру пространства признаков. В результате возникает риск потери согласованности между частными классификаторами, что негативно сказывается на устойчивости системы в целом.

Таким образом, обобщение бинарной модели на многоклассовую постановку сталкивается с рядом фундаментальных затруднений. Проблемы интерпретируемости, статистической состоятельности и устойчивости принятия решений становятся особенно острыми при наличии несбалансированных классов и сложной структуры признакового пространства. Эти соображения подводят к необходимости переосмысления самой постановки задачи классификации, особенно в ситуациях, когда интерес представляет лишь один или малое число целевых классов, а остальные данные играют вспомогательную роль.

2.8 Применение

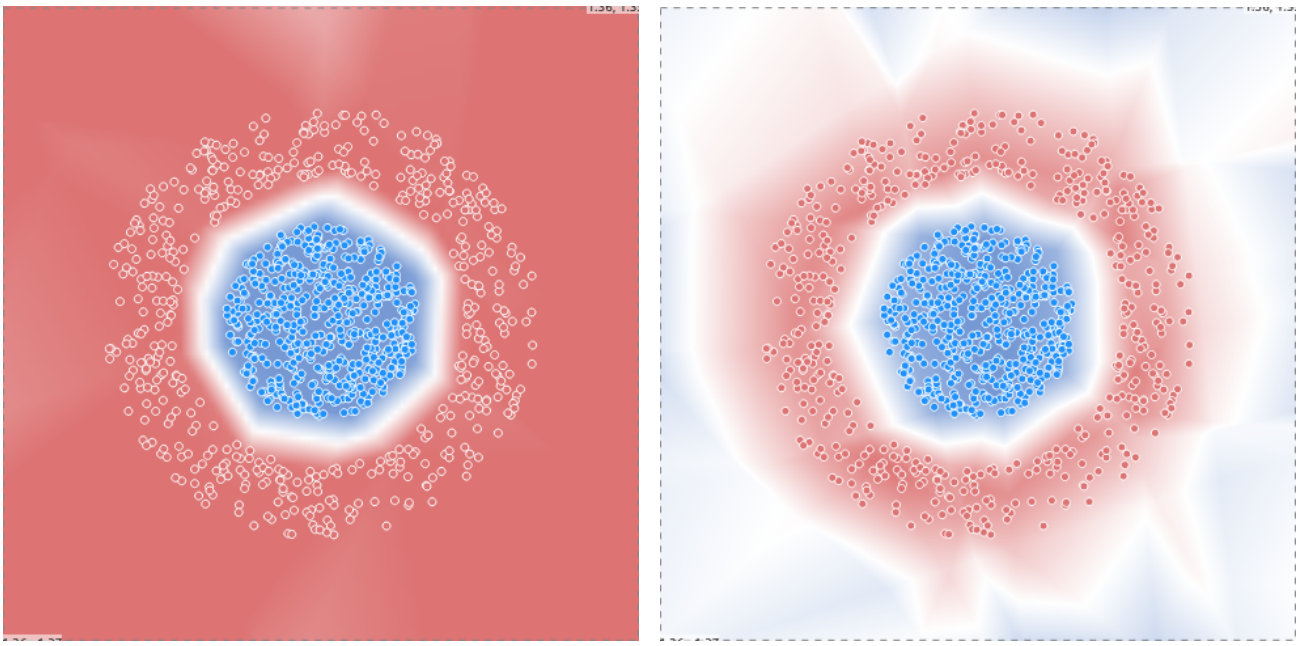
Рассмотренные в предыдущих разделах методы бинарной классификации позволяют успешно решать задачи разделения двух классов на компакте признакового пространства. В данном разделе рассматриваются примеры, иллюстрирующие особенности работы моделей, использующих описанную в разделе 2.3 модификацию, а также проблемы, которые такая модификация позволяет эффективно решать.

2.8.1 Компромисс между точностью классификации и механизмом отказа

Для оценки влияния предлагаемой модификации классификатора на качество предсказаний были рассмотрены три синтетических набора данных: структура “кольцо–круг”, двумерная шахматная разметка 2×2 и два пересекающихся гауссовских распределения. Для каждого набора данных построены решения, полученные классическим классификатором и модифицированным вариантом, что проиллюстрировано на рисунках 2.6 — 2.8.

Экспериментальные результаты показывают, что в отсутствие механизма отказа классический классификатор достигает практически максимального качества классификации, составляющего 99–100% на всех рассмотренных наборах данных. При этом модифицированный классификатор демонстрирует несколько более низкое значение точности, находящееся в диапазоне 97–98%. Данное снижение качества связано с тем, что часть объектов, расположенных в областях повышенной неопределённости, классифицируется как принадлежащая фоновому классу, а не одному из целевых классов.

Однако при исключении из оценки качества объектов, отнесённых классификатором к области отказа, точность модифицированной модели возрастает и достигает значений, сопоставимых с классическим классификатором (99–100%), при доле отказов порядка 0.5–1% от общего числа наблюдений. Таким образом, модификация классификатора реализует характерный для доверенных систем компромисс между точностью и надёжностью предсказаний:



а) классический классификатор

б) модифицированный классификатор

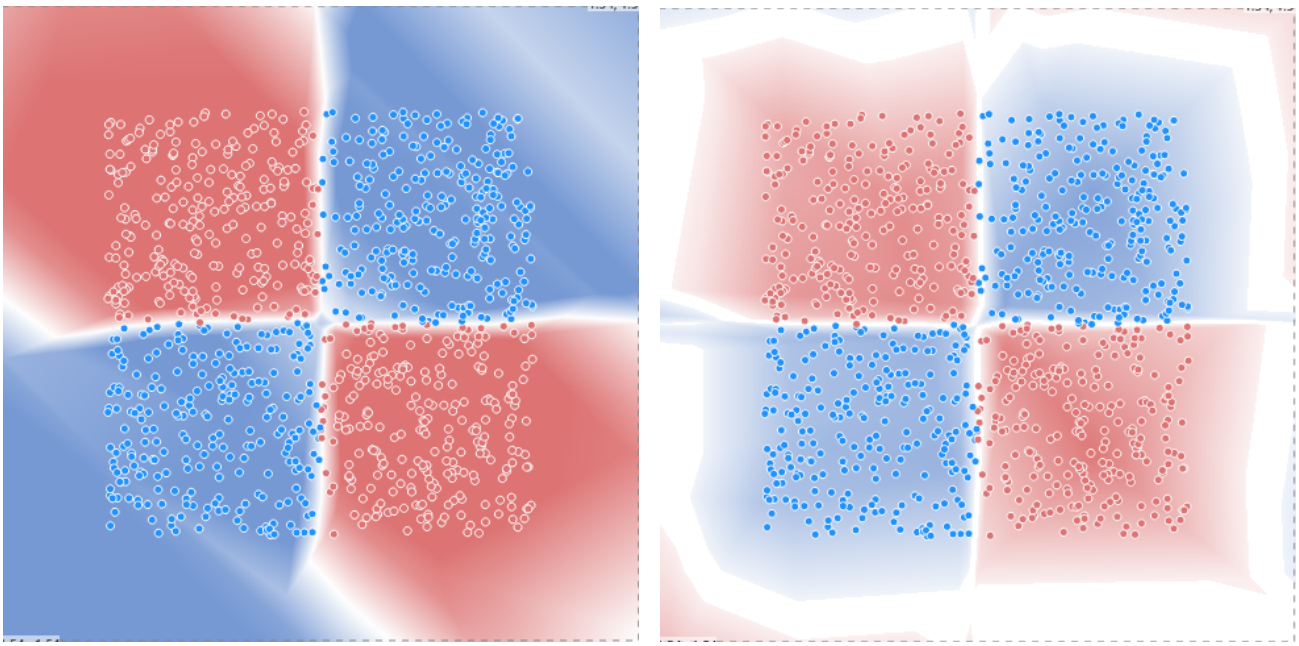
Рисунок 2.6 — Сравнение классического и модифицированных классификаторов

небольшое снижение общей точности достигается за счёт появления механизма отказа, позволяющего существенно повысить достоверность решений на подмножестве объектов, для которых классификация выполняется.

Полученные результаты подтверждают, что предлагаемая модификация не приводит к деградации качества классификации в области компетенции модели, но при этом обеспечивает принципиально новое свойство – возможность формализованного отказа от принятия решения в условиях высокой неопределённости, что является ключевым требованием доверенного искусственного интеллекта.

2.8.2 Поведение вне носителя распределения

Обычные бинарные классификаторы, обученные по конечной выборке без дополнительного “фона“, склонны выдавать уверенные предсказания даже в тех точках пространства, где отсутствуют обучающие данные. Это поведение связано с тем, что модель не знает о структуре плотности признаков и минимизирует ошибку лишь на ограниченном множестве точек.



а) классический классификатор

б) модифицированный классификатор

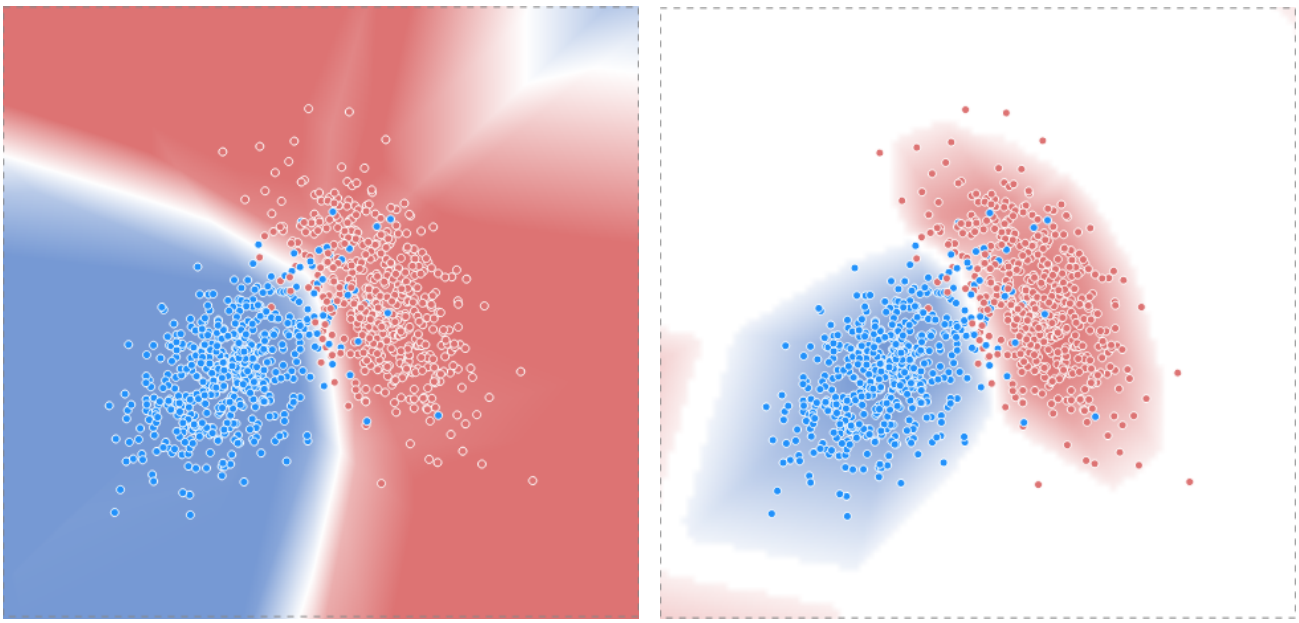
Рисунок 2.7 — Сравнение классического и модифицированных классификаторов

Рассмотрим демонстрационный пример с двумя классами, заданными в виде спиралей на двумерной плоскости (красным цветом обозначены точки класса $+1$, а синим точки класса -1). На рисунке 2.9а представлено решение, полученное обычным бинарным классификатором. Видно, что модель уверенно относит к одному из классов даже точки, расположенные далеко за пределами области, покрытой обучающими данными.

Для сравнения, если использовать, описанную в разделе 2.4.3 модифицированную процедуру, то классификатор начинает учитывать общую структуру распределения данных и классифицирует “внешние“ точки как фон (рисунок 2.9б, фон представлен белым цветов). Это значительно повышает надёжность предсказаний и позволяет говорить о появлении эффекта отказа от распознавания вне носителя распределения.

2.8.3 Устойчивость

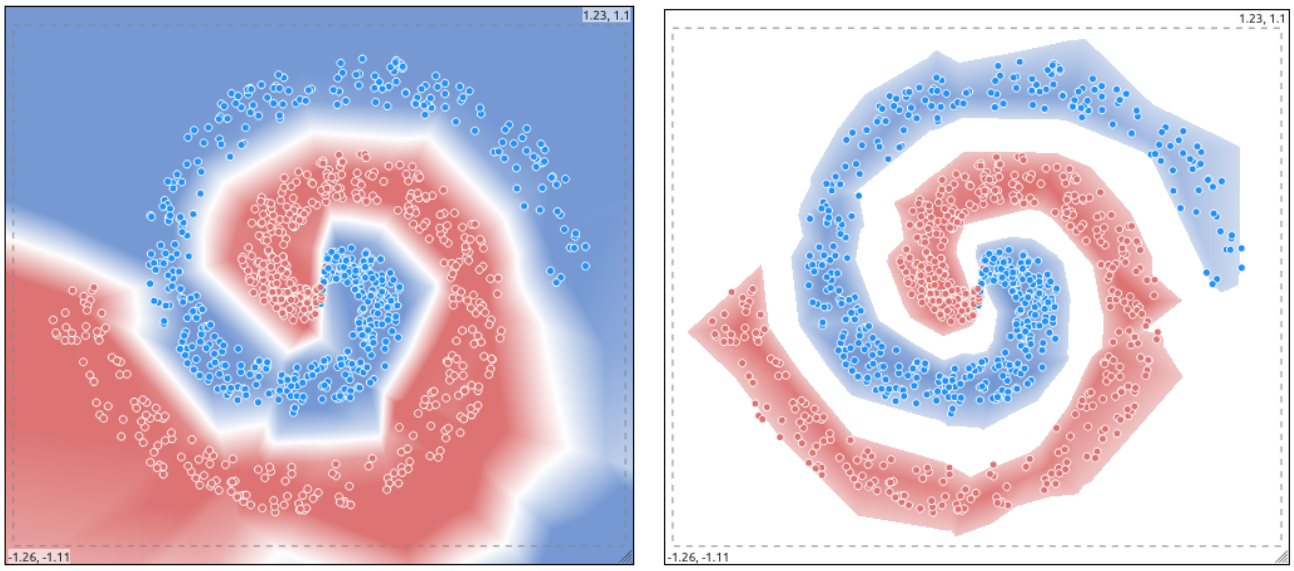
Модели, обучаемые без использования фона, оказываются чрезвычайно чувствительными к отдельным аномальным точкам. Добавление даже одной



а) классический классификатор

б) модифицированный классификатор

Рисунок 2.8 — Сравнение классического и модифицированных классификаторов



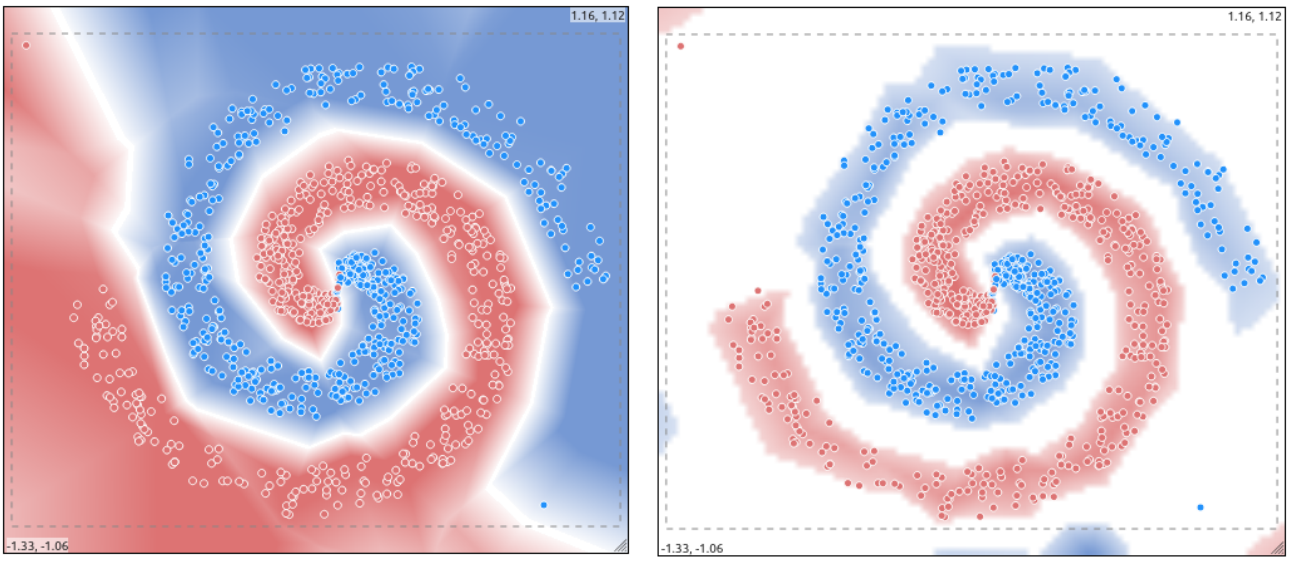
а) классический классификатор

б) модифицированный классификатор

Рисунок 2.9 — Сравнение поведения классификаторов вне носителя

точки может радикально изменить форму решающего правила (рисунок 2.10а). Это явление лежит в основе так называемых backdoor-атак [56], когда намеренно добавленные в обучающую выборку точки провоцируют нежелательное поведение модели в заранее заданной области.

Добавление фона значительно снижает эффект подобных атак (рисунок 2.10б). Чтобы в присутствии фона точка начала влиять на решение, необходимо существенно увеличить её плотность, что требует добавления множества подобных примеров. Таким образом, обучение с фоном повышает устойчивость модели к целевым модификациям данных.



а) классический классификатор

б) модифицированный классификатор

Рисунок 2.10 — Сравнение устойчивости классификаторов к backdoor-атаке

2.8.4 Сопоставление нейросетевой и гистограммной регрессии

В разделе 2.4.4 рассматривалось иерархическое разбиение компакта нейросетевой моделью на ячейки, на основе которых строилась функция гистограммной регрессии. Визуальное сопоставление результатов нейросетевой и гистограммной регрессий подтверждает близость этих методов: выход нейросети в силу своей непрерывности плавно переходит от одного класса к другому, приближая собой ступенчатую структуру гистограммы (рисунок 2.11). Ячейки гистограммы, на которые разбивает пространство персептрон, окрашены в соответствии со значением $h_n^*(X)$ в ячейке и визуально очень похожи на выход нейросети.

Это наблюдение позволяет рассматривать регрессию на выходе многослойного персептрона как более плавную версию гистограммной аппроксимации, реализуемую при помощи кусочно-линейных функций.

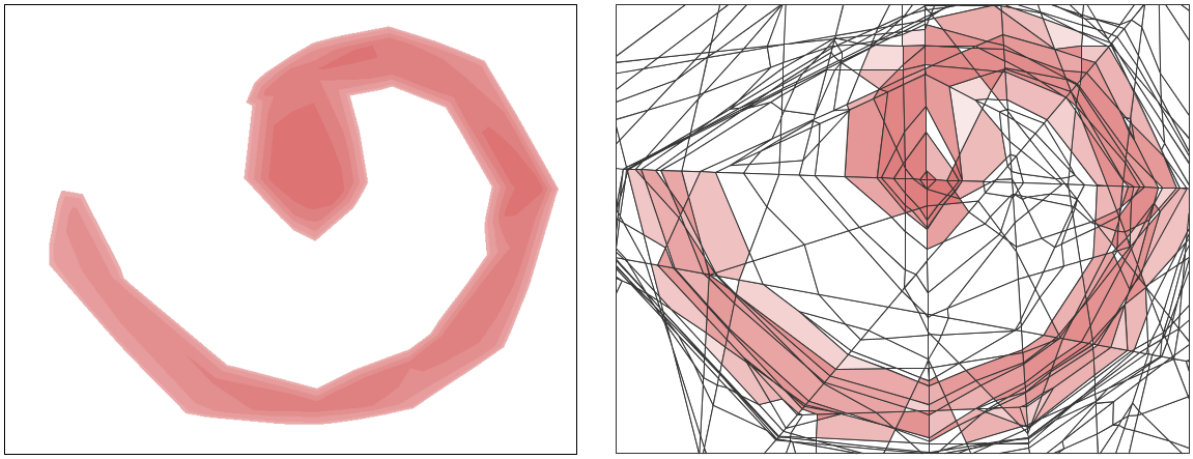


Рисунок 2.11 — Визуальное сравнение функций нейросетевой и гистограммной регрессий

2.8.5 Отказ от распознавания и интерпретация выходов

Добавление фона позволяет не только лучше моделировать границу классов, но и реализовать механизм отказа от распознавания: наблюдения, попадающие в области низкой плотности, классифицируются как “неизвестные“. Это открывает путь к более гибкому принятию решений – например, маркированию таких примеров для дополнительного анализа или анализа дополнительных признаков.

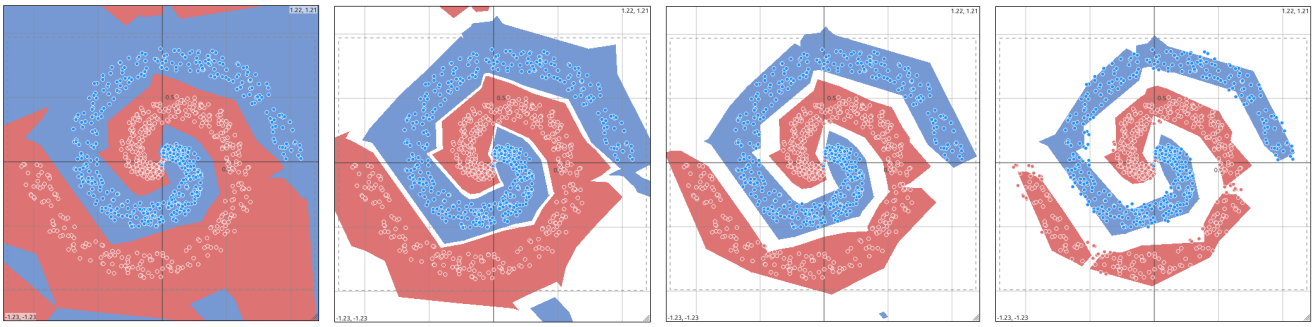
2.8.6 Влияние порога доверия на характеристики классификатора

В рамках предложенного подхода в качестве дополнительного механизма контроля за качеством классификации вводится параметр $\beta \in [0, 1)$, интерпретируемый как порог доверия. Значение β используется для принятия решения о

классификации наблюдения: если значение выхода модели по модулю не превышает β, классификатор воздерживается от принятия решения, т.е. формирует отказ от распознавания.

Введение порога β позволяет контролировать баланс между полнотой и надёжностью классификационных решений. При низких значениях β классификатор склонен выдавать решения по всем поступающим наблюдениям, включая случаи с высокой неопределённостью. При этом возрастает риск ошибочной классификации, особенно вблизи границ разделяющих поверхностей. Повышение значения β ведёт к росту количества отказов от распознавания, но одновременно повышает достоверность решений по тем наблюдениям, для которых классификация всё же производится.

На рисунке 2.12 приведена визуализация результатов классификации при различных значениях порога β: от 0 (классификация осуществляется по всем наблюдениям) до 0.5 (классификатор выдаёт решение только в случаях высокой уверенности). Видно, что при увеличении β область отказов расширяется (обозначена белым цветом), что соответствует желаемому поведению системы в условиях ограниченной уверенности модели.



а) $\beta = 0$ б) $\beta = 0.1$ в) $\beta = 0.3$ г) $\beta = 0.5$

Рисунок 2.12 — Влияние порога доверия β на пространственное распределение классификационных решений

2.9 Экспериментальное исследование доверенного классификатора

Несмотря на широкую практическую применимость нейросетевых моделей, включая многослойный персептрон, их надёжность может быть существенно снижена при воздействии целенаправленных возмущений, известных как

состязательные атаки. Эти атаки используют особенности разделяющей поверхности модели для генерации входных данных, вызывающих ошибочную классификацию, что ставит под вопрос доверие к подобным системам в критически важных приложениях. В рамках настоящей работы в авторской статье [2] предложен новый подход к формированию таких примеров специально для персептрона, получивший название “простая линейная атака на персептрон“ (SLAP, Simple Linear Attack for Perceptron). В данном разделе описывается предложенная атака и выполняется анализ устойчивости модифицированного классификатора к её воздействию, что позволяет оценить уровень доверия к разработанной модели.

2.9.1 Существующие подходы к генерации состязательных примеров

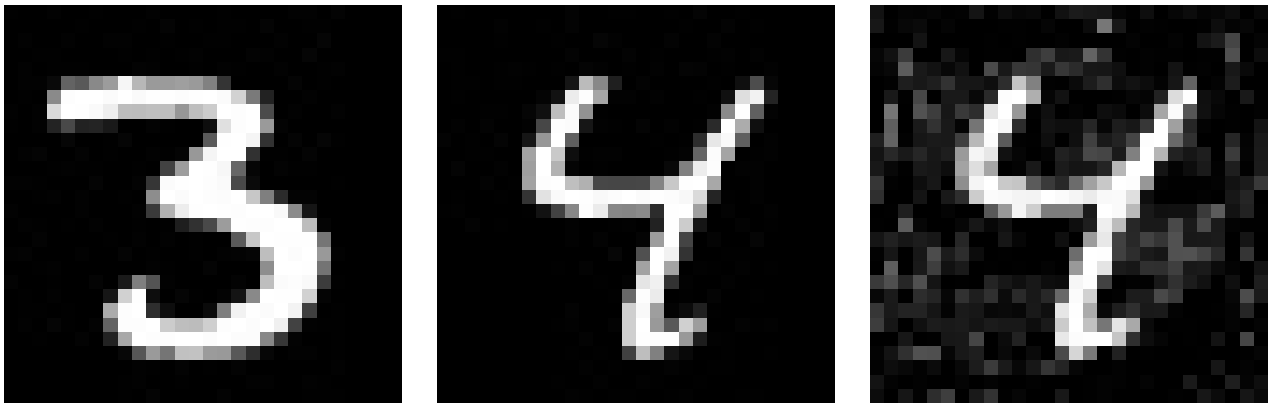
Наиболее распространённые методы формирования атакующих примеров основываются на градиентной оптимизации. В частности, метод FGSM (Fast Gradient Sign Method) [57] и метод проецированного градиентного спуска PGD (Projected Gradient Descent) [58] находят направления в пространстве входных признаков, по которым можно максимизировать ошибку классификатора. Однако такие подходы требуют итеративных вычислений и чувствительны к выбору гиперпараметров.

Альтернативой являются методы, использующие выпуклую оптимизацию или линейное программирование, например, [59; 60]. В настоящей работе предложен подход, основанный исключительно на методах линейной алгебры, позволяющий строить атакующие примеры за счёт решения систем линейных уравнений или неравенств. Подход ориентирован на персептроны с кусочно-линейными функциями активации, такими как ReLU, Leaky ReLU и Abs, что позволяет упростить структуру модели до линейных преобразований при фиксированных знаках активации.

2.9.2 Постановка задачи линейной атаки

Пусть имеется обученный персептрон $c(x)$, принимающий на вход вектор $x \in \mathbb{R}^d$ и возвращающий вектор выходных значений $y \in \mathbb{R}^C$, соответствующих C классам. Обозначим через x_t целевой пример (рисунок 2.13а), а через x_a – пример, который подвергается атаке (рисунок 2.13б). Требуется построить новый вектор x (рисунок 2.13в), близкий к x_a, но классифицируемый так же (или почти так же), как x_t. Формально, задача формулируется как:

$$\begin{cases} \|x - x_a\| \to \min, \\ \|x - x_t\| > 0, \\ c(x) = c(x_t), \end{cases} \quad \text{или} \quad \begin{cases} \|x - x_a\| \to \min, \\ \|x - x_t\| > 0, \\ \arg\max c(x) = \arg\max c(x_t). \end{cases}$$



а) x_t – целевой пример

б) x_a – пример, подвергающийся атаке

в) x – построенный атакующий пример

Рисунок 2.13 — Примеры, участвующие в атаке на многослойный персептрон

Для повышения скрытности атаки накладываются ограничения на диапазон допустимых значений $x \in [x_{\min}, x_{\max}]$, где, например, для изображений естественно полагать $x_{\min} = 0$, $x_{\max} = 1$.

2.9.3 Атака на однослойный персептрон

Рассмотрим случай одного слоя, где выходной вектор модели задаётся как $y = Wx + b$, причём $W \in \mathbb{R}^{C \times d}$, $b \in \mathbb{R}^C$.

Без учёта ограничений на входные значения

Предположим, что матрицу W можно разбить на подматрицы $W_1 \in \mathbb{R}^{C \times C}$ и $W_2 \in \mathbb{R}^{C \times (d-C)}$, выбрав, например, первые (или случайные для большей незаметности) C столбцов (предполагается, что в этом случа количество классов меньше, чем размер входного пространства). Аналогично разбиваем вектор x_a на $x_{a_1} \in \mathbb{R}^C$ и $x_{a_2} \in \mathbb{R}^{d-C}$. Тогда атакующий вектор может быть получен по формуле:

$$x^* = W_1^{-1} \cdot (b^\top - W_2 x_{a_2}),$$

а полное решение восстанавливается как конкатенация $x = [x^*, x_{a_2}]$ (рисунок 2.14).

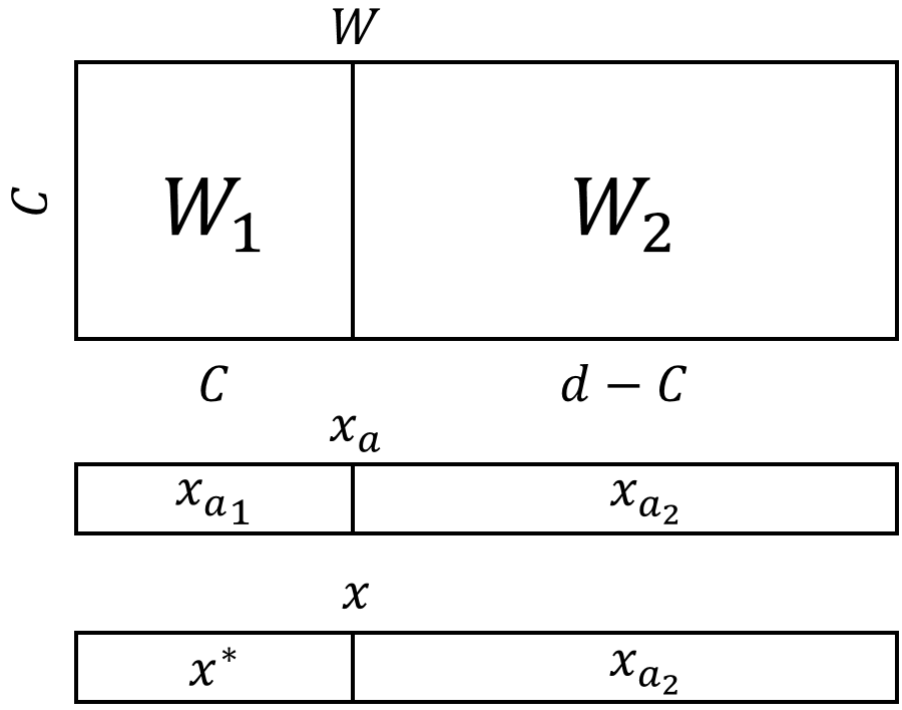
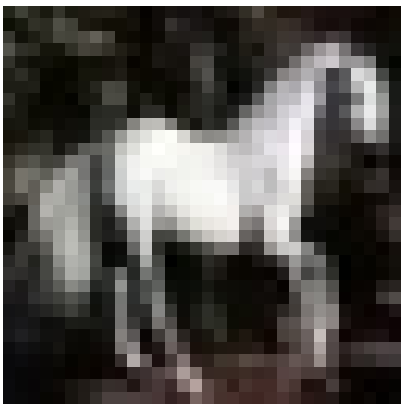


Рисунок 2.14 — Схема матричной атаки

Данный метод работает исключительно в случае, если матрица W_1 обратима (что почти всегда выполняется при случайной инициализации). Однако он не учитывает допустимые границы значений и чувствителен к квантованию, происходящему при сохранении изображения в файл (рисунок 2.15).

а) x_t – целевой пример



б) x_a – пример, который подвергается атаке

в) x – построенный атакующий пример

Рисунок 2.15 — Пример матричной атаки, $x \in [-1055, 926]$

С учётом ограничений

Более реалистичный подход включает в себя формулировку задачи как квадратичной оптимизации:

$$\begin{cases} \frac{1}{2}x^\top Px + q^\top x \to \min, \\ Ax = b, \\ x_{\min} \leqslant x \leqslant x_{\max}, \end{cases}$$

где в простейшем случае $P = E$ – единичная матрица, $q = -x_a$, $A = W$, $b = y_t - b$. Тогда задача принимает вид:

$$\begin{cases} \frac{1}{2}x^\top x + x_a^\top x \to \min, \\ Wx = y_t - b, \\ x \in [x_{\min}, x_{\max}]. \end{cases}$$

При невозможности точного воспроизведения y_t возможно ослабление условий за счёт введения допусков ε:

$$y_t - \varepsilon \leqslant Wx + b \leqslant y_t + \varepsilon.$$

Эти неравенства легко переписываются в канонической форме для QP-решателей. Пример применения данного вида атаки приведён на рисунке 2.16.

а) x_t – целевой пример

б) x_a – пример, который подвергается атаке

в) x – построенный атакующий пример

Рисунок 2.16 — Пример QP атаки

2.9.4 Атака на многослойный персептрон

При наличии нескольких слоёв в сети возникает проблема нелинейности из-за активационных функций. Однако, если эти функции кусочно-линейны (ReLU, Leaky ReLU, Abs), то при фиксированных знаках аргументов они представляют собой линейные отображения. Например, функция ReLU ведёт себя как x при $x \geqslant 0$ и как 0 при $x < 0$.

Рассмотрим модель из трёх слоёв:

$$y = W_3 \cdot f_2(W_2 \cdot f_1(W_1 x + b_1) + b_2) + b_3.$$

Предположим, что знаки активации известны (например, получены от прямого прохода по x_t). Тогда последовательное раскрытие слоёв позволяет свести сеть к линейной модели. Например, если все значения после первого слоя положительны (т.е. активация f_1 действует как тождественная функция), а после второго – отрицательны (и активация действует как умножение на константу), можно получить:

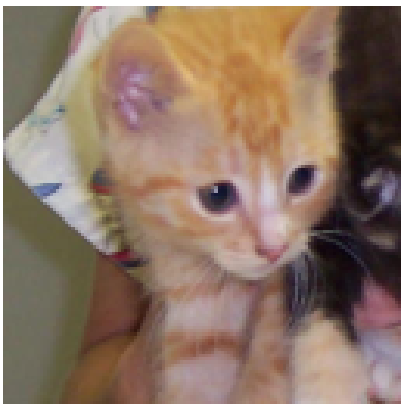
$$\begin{cases} W_1 x + b_1 \geqslant 0 \\ W_2 W_1 x + W_2 b_1 + b_2 \leqslant 0 \\ y = -(W_3 W_2 W_1 x + W_3 W_2 b_1 + W_3 b_2) + b_3 \end{cases}$$

где коэффициенты W_{321}, b_{321} выражаются через произведения матриц весов и сдвигов. Тогда атака сводится к аналогичной задаче QP, но при дополнительных ограничениях на знаки промежуточных переменных:

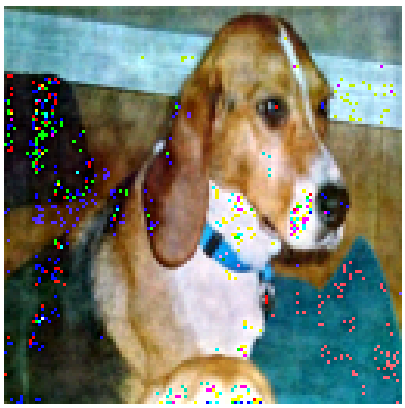
$$\begin{cases} W_{21} = W_2 W_1 \\ b_{21} = W_2 b_1 + b_2 \\ W_{321} = -W_3 W_2 W_1 \\ b_{321} = b_3 - W_3 W_2 b_1 - W_3 b_2 \\ W_1 x + b_1 \geqslant 0 \\ W_{21} x + b_{21} \leqslant 0 \\ y = W_{321} x + b_{321} \end{cases}$$

Пример атаки на многослойный персептрон представлен на рисунке 2.17.

а) x_t – целевой пример



б) x_a – пример, который подвергается атаке



в) x – построенный атакующий пример

Рисунок 2.17 — Пример атаки на многослойный персептрон на датасете Cat-vs-Dog [61]

2.9.5 Генерация произвольных входов с заданным выходом

Так как размерность входа чаще всего превышает размерность выхода, задача построения входа x, удовлетворяющего $c(x) = y_t$, имеет бесконечно много решений. В этом случае можно случайным образом зафиксировать некоторые

координаты x, оставляя другие свободными, и решать полученную переопределённую систему. Это позволяет формировать обширные множества атакующих примеров, обладающих одинаковым выходом сети.

На рисунке 2.18 приведены примеры атакующих изображений. В первом столбце расположены целевые изображения, соответствующие заданному выходу модели. Второй столбец содержит атакующие примеры, полученные путём минимального возмущения других исходных изображений с целью приведения их к тому же выходу. Остальные столбцы демонстрируют изображения, сгенерированные методом случайного поиска при условии воспроизведения целевого выхода. Все изображения в пределах одной строки имеют идентичный выходной вектор персептрона, несмотря на различия в визуальном представлении.

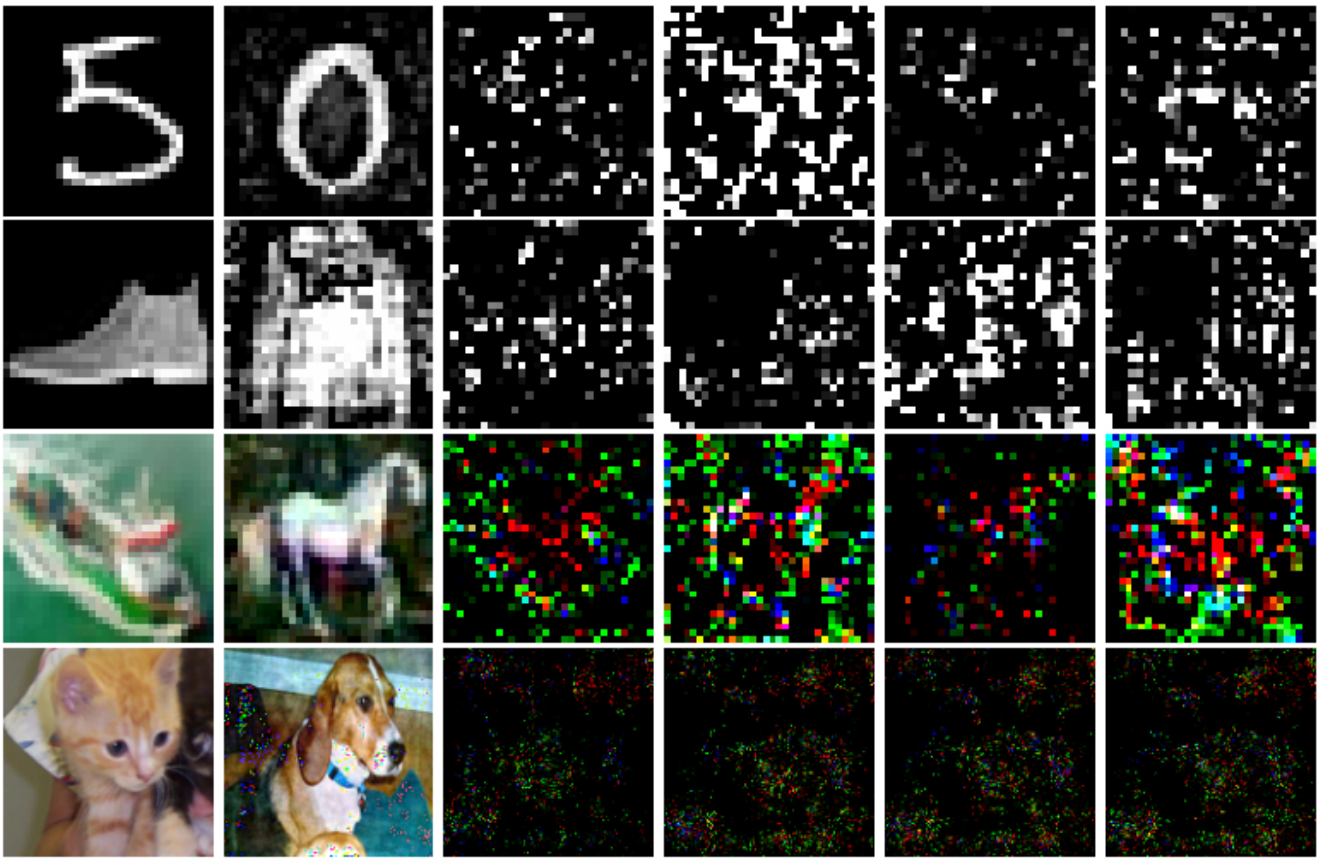


Рисунок 2.18 — Пример генерации атакующих примеров

2.9.6 Экспериментальное исследование

Алгоритм был реализован и протестирован на простых персептронах, обученных на датасетах MNIST [62] и CIFAR-10 [63]. Для каждого изображения из тестовой выборки выбиралось случайное изображение другого класса, и применялась атака, направленная на минимальное изменение первого изображения с целью получения выхода второго. В эксперименте оценивалась ℓ_∞-норма между оригиналом и атакующим примером. Результаты приведены в таблице 1.

Таблица 1 — Результаты применения SLAP атаки

Модель	Набор	Accuracy	Атака на значения		Атака на класс	
			ℓ_∞	Accuracy	ℓ_∞	Accuracy
10	MNIST	0.9288	0.019	0.003	0.019	0.002
10-10		0.9326	0.021	0.007	0.022	0.001
100-10		0.9805	0.052	0.009	0.051	0.005
1000-10		0.9849	0.091	0.012	0.092	0.009
160-80-40-20-10		0.9792	0.117	0.000	0.114	0.000
10	CIFAR10	0.3989	0.027	0.014	0.024	0.011
100-10		0.4853	0.054	0.032	0.055	0.018
1000-10		0.5236	0.095	0.041	0.096	0.023
320-160-80-40-10		0.5353	0.121	0.049	0.119	0.037

Результаты показывают, что при использовании простых архитектур удаётся достигать атакующих примеров с минимальными отклонениями, зачастую визуально незаметными. При переходе к более глубоким моделям число необходимых изменений возрастает, что объясняется более сложной геометрией границ принятия решений.

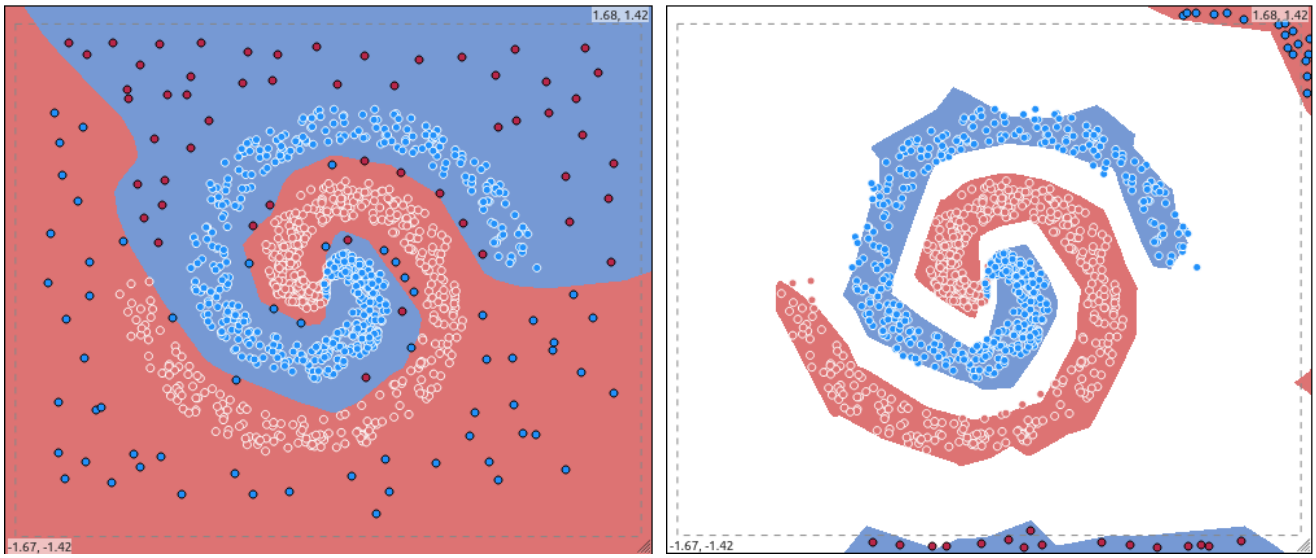
2.9.7 Устойчивость модифицированного классификатора к данной атаке

Модифицированный (доверенный) классификатор предназначен для работы с данными малой размерности, поэтому экспериментальная проверка устойчивости проводилась на модельном двумерном наборе данных, представляющем собой два витка спирали.

Для стандартного многослойного персептрона SLAP-атака успешно строит атакующие примеры. Найденные точки часто оказываются в областях пространства, вне визуально определяемого носителя данных (спирали), что, однако, не препятствует их успешной классификации моделью (рисунок 2.19а). Обучающие данные представлены в виде точек с белой обводкой. Атакующие точки представлены с чёрной обводкой (цвет точек инвертирован для большей контрастности). Компакт, внутри которого принимается решение обозначен пунктирной линией.

В случае модифицированного классификатора результаты принципиально иные (рисунок 2.19б):

– **Без учёта ограничений**: при решении системы линейных уравнений алгоритм формально находит атакующую точку, но она практически всегда оказывается вне заданного компакта (допустимой области определения модели), что делает такой пример легко детектируемым.
– **С учётом ограничений**: при формулировке задачи как квадратичной оптимизации с условиями $x \in [x_{min}, x_{max}]$ в подавляющем большинстве случаев допустимое решение не существует. Это означает, что не удаётся найти точку, одновременно удовлетворяющую условию смены класса и остающуюся в доверенной области модели. В тех же случаях, когда решение удаётся найти, оно оказывается близко к границе компакта в отдалении от носителя распределения и также может быть легко обнаружено путём анализа выбросов.



а) обычный классификатор б) доверенный классификатор

Рисунок 2.19 — Устойчивость доверенного классификатора к SLAP атаке

Данные наблюдения свидетельствуют о высокой устойчивости модифицированного классификатора к SLAP-атаке. Наличие фоновых точек эффективно сужает пространство возможных атак, требуя от злоумышленника создания либо тривиально обнаруживаемых (выходящих за границы), либо вычислительно труднодостижимых возмущений. Таким образом, модифицированный классификатор демонстрирует качественно более высокий уровень доверия с точки зрения устойчивости к целенаправленным состязательным воздействиям.

2.10 Выводы

Представленная в данном разделе математическая модель модифицированного байесовского классификатора, предназначена для создания доверенных решающих систем при малой размерности данных. Доказанная теорема гарантирует корректный отказ от классификации вне носителя распределения, что обеспечивает предложенной модели статистическую обоснованность, отсутствующую у стандартных нейросетевых классификаторов. Ключевой практической реализацией этой теоретической конструкции является аппроксимация байесовского решающего правила с помощью многослойного персептрона, дополненного механизмом явного ограничения области принятия решений.

Важным аспектом разработанного подхода является обеспечение статистической интерпретируемости модели. Предложенное объясняющее двоичное дерево eXBTree позволяет анализировать локальную уверенность классификатора и предоставляет инструмент для понятного объяснения его решений, что критически важно для доверенных систем.

Практическим следствием такого подхода является достижение двух взаимосвязанных целей: статистически корректного поведения модели на фоновых областях и существенного повышения устойчивости к состязательным атакам. Экспериментальная проверка с использованием метода SLAP подтвердила, что модифицированный классификатор успешно противостоит целенаправленным возмущениям - атакующие примеры либо оказываются за границами допустимого компакта и становятся легко детектируемыми, либо не могут быть построены без нарушения базовых ограничений модели. Таким образом, работа демонстрирует возможность построения доверенных классификаторов, которые сочетают

эффективность нейросетевой аппроксимации со статистической строгостью байесовского подхода в условиях малой размерности данных.

Глава 3. Унарная классификация

Описанный в предыдущем разделе доверенный бинарный классификатор, несмотря на свои теоретические преимущества, сталкивается с существенной практической проблемой: его устойчивость и качество предсказаний критически зависят от сбалансированности обучающих данных. В реальных сценариях данные часто обладают выраженным дисбалансом классов, что приводит к смещённым оценкам плотности и, как следствие, к снижению надёжности модели. Это противоречит самой цели создания доверенной системы, так как её решения становятся статистически необоснованными для слабо представленного класса.

В данной главе предлагается новый подход, называемый унарной классификацией. Вместо одновременного моделирования двух классов в фокусе оказывается только целевой класс, а данные противостоящего класса исключаются из процесса обучения персептрона. Такой подход позволяет полностью устранить влияние дисбаланса, сфокусировать ресурсы модели на точном описании целевого распределения и, как результат, построить более устойчивую и предсказуемую систему, соответствующую требованиям доверенного искусственного интеллекта в условиях несбалансированных данных.

3.1 Нейросетевая регрессия для единственного класса

Как отмечалось в разделе 2.4.3, многослойный персептрон с кусочно-линейной функцией активации способен осуществлять ε-приближённую аппроксимацию любой непрерывной функции на компакте. При наличии L скрытых слоёв с k нейронами в каждом, структура такой сети задаёт иерархическое разбиение компакта $[0,1]^d$ на $O(k^{dL})$ ячеек, внутри которых выход модели является линейной функцией. Вычисление значения сети в произвольной точке $x \in [0,1]^d$ требует лишь последовательных операций скалярного произведения и сравнения, что обеспечивает высокую вычислительную эффективность.

Для построения унарного классификатора вводится обобщённая задача регрессии. Пусть имеется выборка наблюдений $\{X_i\}_{i=1}^n$ - независимые одинаково распределённые случайные величины на компакте $[0,1]^d$ с неизвестной

ограниченной плотностью $f(X)$. Эта выборка интерпретируется как наблюдения целевого процесса, каждому из которых сопоставляется метка $Y_i = 1$. Дополнительно формируется фоновый набор $\{X_i\}_{i=n+1}^{2n}$ - независимые равномерно распределённые на $[0,1]^d$ случайные величины с метками $Y_i = 0$. В результате получается сбалансированный комбинированный набор $\{(X_i, Y_i)\}_{i=1}^{2n}$ мощности $2n$.

Рассмотрим теперь задачу построения аппроксимирующей полносвязной нейросети $c_n(X)$, решающей задачу регрессии в классе моделей фиксированной сложности, аналогичную задаче (2.10). Требуется найти такую нейросеть $c_n^*(X)$, минимизирующую среднеквадратичную ошибку на объединённом наборе данных:

$$\sum_{i=1}^{2n} (c_n(X_i) - Y_i)^2 \to \min_{c_n}, \tag{3.1}$$

где минимум берётся по всем полносвязным нейросетям, общее число нейронов в которых не превышает заданного порогового значения $kL + 1$.

Полученную в результате оптимизации модель, будем называть **нейросетевым унарным классификатором**. Принятие решения о принадлежности наблюдения классу будет осуществляться при превышении выходного значения персептрона порога доверия $\beta \in [0,1)$:

$$c_n^*(X) > \beta.$$

3.2 Гистограммная регрессия для единственного класса

Пусть в результате построения $c_n^*(X)$ на компакте $[0,1]^d$ получено разбиение на N ячеек $K = \{K_1, K_2, \ldots, K_N\}$. Введём далее кусочно-постоянную функцию $h_n(X)$, принимающую постоянные значения внутри каждой ячейки K_r, и сформулируем задачу приближённой оценки вероятности принадлежности наблюдения классу $Y = 1$ в виде:

$$\sum_{i=1}^{2n} (h_n(X_i) - Y_i)^2 \to \min_{h_n}. \tag{3.2}$$

Как и в (2.11), задача (3.2) может быть решена независимо в каждой ячейке K_r, при этом оптимальное значение $h_n^*(X)$ в данной ячейке определяется соотношением:

$$h_n^*(X) = \frac{n_1(X)}{n_1(X) + n_0(X)}, \tag{3.3}$$

где $n_1(X)$ – количество наблюдений с меткой $Y = 1$ в ячейке, содержащей точку X, а $n_0(X)$ – количество фоновых наблюдений (с меткой $Y = 0$) в той же ячейке.

Пример вычисления функции гистограммной регрессии показан на рисунке 3.1.

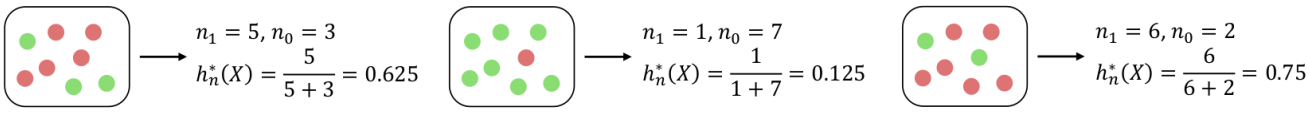


Рисунок 3.1 — Пример вычисления $h_n^*(X)$ в некоторой ячейке K_r в унарном случае

Полученная функция $h_n^*(X)$ представляет собой гистограммную оценку апостериорной вероятности принадлежности наблюдения X к целевому распределению, основанную на разбиении, полученном с помощью нейросетевой аппроксимации.

Теорема 4 (О сходимости нейросетевой и гистограммной регрессий)**.** *Пусть задана последовательность многослойных персептронов, обученных на выборке из n наблюдений из распределения с ограниченной плотностью на $[0,1]^d$, с модульной функцией активации, архитектура которых состоит из первого слоя ширины r_n, L_n слоёв ширины k_n и одноэлементного последнего слоя с линейной активацией, при условии, что весовые коэффициенты инициализируются независимо из непрерывного распределения, а параметры первого слоя заморожены при обучении, если целевая функция является кусочно-гладкой и выполнены ограничения:*

(i) Число ненулевых параметров:

$$S_{\mathrm{nnz},n} \;=\; c_1' \cdot \max\left\{ n^{\frac{d}{2\beta+d}},\, n^{\frac{d-1}{\alpha+d-1}} \right\}. \tag{3.4}$$

(ii) Ограничение на величины параметров:

$$B_n \;\leqslant\; c_1\, n^s. \tag{3.5}$$

(iii) Количество слоёв после первого:

$$L_n \leqslant c_1\Big(1 + \max\big\{\tfrac{\beta}{d}, \tfrac{\alpha}{2(d-1)}\big\}\Big). \tag{3.6}$$

(iv) Число нейронов в первом слое:

$$r_n \geqslant 2d. \tag{3.7}$$

(v) Ограничение на скорость роста архитектуры:

$$k_n^{L_n \min(d,k_n)}\, r_n^d = o(n), \tag{3.8}$$

$$k_n^{L_n \min(d,k_n)} r_n^d\, d\,(k_n L_n + r_n)\,\log(k_n L_n + r_n)\,\frac{\log n}{n} \to 0. \tag{3.9}$$

(vi) Ограничения на ширину первого слоя (для произвольного $\gamma > 0$):

$$\sum_{n=1}^{\infty} e^{\frac{-c\gamma^4 r_n}{d^2}} < \infty, \tag{3.10}$$

$$r_n \geqslant C\,\gamma^{-12}\,\frac{d^7}{2\pi}. \tag{3.11}$$

Тогда

$$(c_n(X) - h_n(X)) \xrightarrow{P} 0. \tag{3.12}$$

Доказательство теорем. 4 приведено в приложении Б.2. Данная теорема обосновывает статистическую состоятельность классификатора на основе многослойного персептрона с кусочно-линейными функциями активации скрытых слоёв.

3.3 Вероятностная интерпретация унарной классификации

Построенный унарный классификатор допускает естественную вероятностную интерпретацию, связывающую его выход с оценкой плотности целевого распределения. Для этого рассмотрим гистограммные оценки плотностей, соответствующие двум компонентам обучающей выборки:

$$f_n(X) = \frac{n_1(X)}{n \cdot V(K_r)}, \quad p_n(X) = \frac{n_0(X)}{n \cdot V(K_r)},$$

где $V(K_r)$ - мера ячейки K_r, $n_1(X)$ - число целевых наблюдений в этой ячейке, $n_0(X)$ - число фоновых наблюдений.

Апостериорная вероятность, оцениваемая гистограммным методом, может быть выражена через эти плотности:

$$h_n^*(X) = \frac{f_n(X)}{f_n(X) + p_n(X)}.$$

Учитывая, что фоновые данные распределены равномерно ($p(X) \equiv 1$ на компакте $[0,1]^d$), и принимая $p_n(X)$ в качестве оценки этой постоянной плотности, можно выразить оценку плотности целевого распределения непосредственно через выход обученного нейросетевого классификатора $c_n^*(X)$:

$$f_n(X) \approx \frac{c_n^*(X)}{1 - c_n^*(X)}.$$

Таким образом, унарный классификатор на основе персептрона представляет собой не только решающее правило, но и потенциально эффективный непараметрический метод оценки плотности распределения. В отличие от классических непараметрических методов, унарному классификатору не требуется хранить всю обучающую выборку целиком - вся информация о распределении инкапсулируется в параметрах обученной нейронной сети. Более того, вычисление оценки плотности в новой точке сводится к одному прямому проходу через сеть, что обеспечивает существенно более высокую вычислительную эффективность по сравнению с методами, требующими обращения ко всем обучающим данным. Это делает предложенный подход перспективным для задач, где важны как точность оценки плотности, так и скорость работы модели в режиме реального времени.

3.4 Случай нескольких классов

В случае многоклассовой классификации ($C > 2$) предлагаемая конструкция унарных классификаторов сохраняет свою применимость и обладает рядом существенных преимуществ по сравнению с классическим подходом, основанным на многоклассовой нейронной сети или на парных классификаторах “один против одного”. Прежде всего, при использовании унарной схемы для каждого

класса $c = 1, \ldots, C$ строится собственный унарный классификатор, обученный различать носитель класса c от фонового равномерного распределения.

Таким образом, требуется построить C независимых классификаторов, каждый из которых решает задачу бинарной классификации в формате "объекты данного класса против фона". В отличие от схемы "один против одного", где количество классификаторов составляет $\frac{C(C-1)}{2}$, унарная схема масштабируется линейно по числу классов и не требует сложных стратегий агрегации результатов голосования.

3.5 Преимущества унарной классификации

Ключевым достоинством унарного подхода является полная устойчивость к проблеме дисбаланса классов. Каждый классификатор обучается только на положительных объектах своего класса и на независимом фоновом множестве, совпадающим по размеру. Таким образом, влияние других, возможно многочисленных, классов исключается на этапе обучения, и несбалансированность исходного обучающего множества не приводит к смещению в сторону более представленных классов.

Кроме того, каждый классификатор формирует свою собственную аппроксимацию апостериорной вероятности $c_n^{(i)}(x)$, оценивая степень принадлежности точки x классу i. Совокупность таких значений $(c_n^{(1)}(x), \ldots, c_n^{(C)}(x))$ образует векторную оценку, позволяющую как выбрать наиболее вероятный класс (например, по максимуму), так и сформулировать стратегию отказа, если все оценки не превышают заданного порога β. Последнее обеспечивает возможность построения отказоустойчивой классификационной системы, способной помечать сомнительные случаи как требующие дополнительного рассмотрения.

Ещё одним немаловажным преимуществом является модульность архитектуры: поскольку все классификаторы независимы, допускается использование различной архитектуры (в том числе различной глубины и сложности) для различных классов. Это даёт возможность адаптировать модель под особенности каждого из классов, делая систему более гибкой и устойчивой к неоднородности обучающих данных.

3.6 Оценка качества унарных классификаторов

При оценке качества стандартных многоклассовых классификаторов традиционно используют метрики точности, полноты, F_1-score и аналогичные [64]. Однако в контексте унарной классификации такие показатели оказываются недостаточно информативными, так как каждый классификатор в унарной схеме обучается независимо и ориентирован на различение своего целевого класса от фонового распределения. В частности, стандартная точность не учитывает случаи "отказа" классификатора (когда выход нейросети не превышает порог β), а F_1-score и подобные метрики не отражают взаимное влияние классификаторов при многоклассовой интерпретации.

Для более детальной оценки работы унарного классификатора предлагается рассматривать три дополнительных свойства: мощность, эффективность и меру неразделимости классов.

3.6.1 Мощность классификатора

Мощность классификатора $c^{(i)}(x)$ определяется как доля точек целевого класса i, принимаемых классификатором, то есть для которых выходная аппроксимация апостериорной вероятности превышает заданный порог β. Формально для двух классов показатели вычисляются следующим образом:

$$n_1^{(1)} = \sum_{i=1}^{n^{(1)}} \mathbb{I}_{\{c^{(1)}(x_i) \geqslant \beta\}}, \quad n_2^{(2)} = \sum_{i=1}^{n^{(2)}} \mathbb{I}_{\{c^{(2)}(x_i) \geqslant \beta\}},$$

$$p^{(1)} = \frac{n_1^{(1)}}{n^{(1)}}, \quad p^{(2)} = \frac{n_2^{(2)}}{n^{(2)}},$$

где $n^{(1)}$ и $n^{(2)}$ – количество наблюдений классов 1 и 2 соответственно. Мощность позволяет оценить долю объектов класса, корректно распознанных классификатором без отказа, и является базовой характеристикой "чувствительности" модели к своему классу.

Для получения интегральной характеристики мощности всей пары классификаторов можно использовать гармоническое среднее:

$$P_{12} = \frac{2p^{(1)}p^{(2)}}{p^{(1)} + p^{(2)}}.$$

Метрика P_{12} отражает общую способность пары классификаторов корректно распознавать свои классы. При этом, если один из классификаторов имеет низкую мощность, интегральная метрика также будет снижена, что интуитивно соответствует снижению общей чувствительности системы.

3.6.2 Эффективность классификатора

Эффективность характеризует способность классификатора корректно выделять объекты своего класса относительно других классификаторов. Для двух классов вводятся следующие показатели:

$$n_{10}^{(1)} = \sum_{i=1}^{n^{(1)}} \mathbb{I}_{\{c^{(1)}(x_i) \geqslant \beta \wedge c^{(2)}(x_i) < \beta\}}, \quad n_{02}^{(2)} = \sum_{i=1}^{n^{(2)}} \mathbb{I}_{\{c^{(2)}(x_i) \geqslant \beta \wedge c^{(1)}(x_i) < \beta\}},$$

$$e^{(1)} = \frac{n_{10}^{(1)}}{n^{(1)}}, \quad e^{(2)} = \frac{n_{02}^{(2)}}{n^{(2)}}.$$

Показатели качества классификатора $e^{(i)}$ отражают долю объектов, корректно распознанных своим классификатором и отвергнутых чужим(и). На их основе определяется интегральная метрика эффективности:

$$E_{12} = \frac{2e^{(1)}e^{(2)}}{e^{(1)} + e^{(2)}}.$$

Метрика E_{12} аналогична гармоническому среднему и позволяет количественно оценить согласованность работы классификаторов при минимизации взаимных ошибок.

3.6.3 Мера неразделимости классов

Для количественной оценки степени пересечения областей, признанных обоими классификаторами, вводится понятие меры неразделимости классов.

Внутренние показатели, характеризующие долю объектов, которые одновременно принимаются обоими классификаторами, интерпретируются как свойство наплываемости классов:

$$n_{12}^{(1)} = \sum_{i=1}^{n^{(1)}} \mathbb{I}_{\{c^{(1)}(x_i)\geqslant\beta \wedge c^{(2)}(x_i)\geqslant\beta\}}, \quad n_{12}^{(2)} = \sum_{i=1}^{n^{(2)}} \mathbb{I}_{\{c^{(2)}(x_i)\geqslant\beta \wedge c^{(1)}(x_i)\geqslant\beta\}},$$

$$g^{(1)} = \frac{n_{12}^{(1)}}{n^{(1)}}, \quad g^{(2)} = \frac{n_{12}^{(2)}}{n^{(2)}},$$

На основе этих показателей определяется интегральная мера неразделимости классов:

$$G_{12} = \frac{2g^{(1)}g^{(2)}}{g^{(1)} + g^{(2)}}.$$

Метрика G_{12} отражает, насколько сильно области, распознаваемые различными классификаторами, перекрываются. Высокое значение G_{12} свидетельствует о значительном наплывании классов друг на друга и, следовательно, о потенциальной сложности их разделения в пространстве признаков.

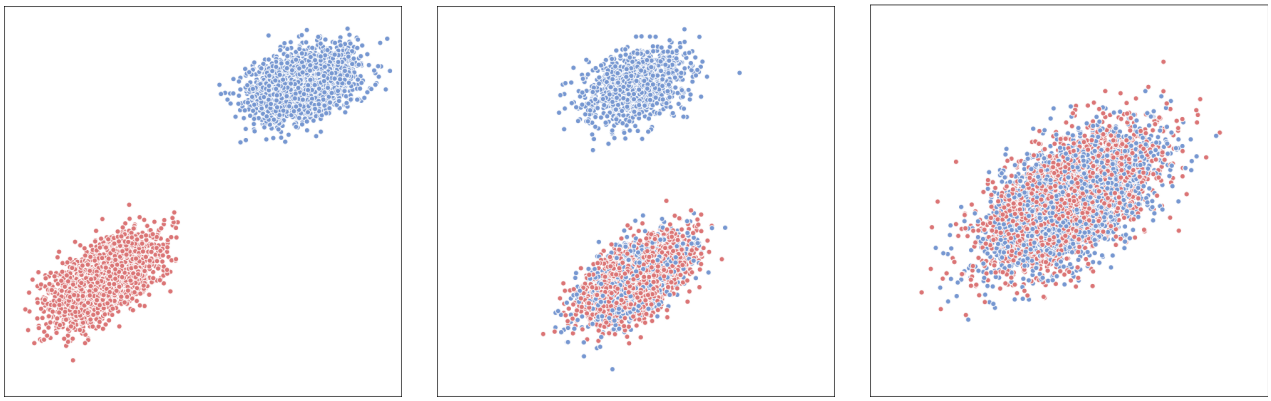
3.6.4 Визуализация метрик

Для иллюстрации поведения предложенных метрик рассмотрены три модельные ситуации и описаны значения интегральных показателей мощности, эффективности и меры неразделимости классов. Для сопоставления приведены также значения стандартных метрик бинарной классификации (accuracy, precision, recall, F_1).

1. **Два разнесённых гауссиана**. Классы линейно разделимы (рисунок 3.2а). Мощности обоих классификаторов равны единице ($p^{(1)} = p^{(2)} = 1$), эффективность также равна единице ($E_{12} = 1$), мера неразделимости равна нулю ($G_{12} = 0$). Классические метрики accuracy, precision, recall и F_1 также принимают значение 1.
2. **Три гауссиана с вложением одного класса в другой**. Второй класс полностью лежит внутри первого (рисунок 3.2б). Мощность первого классификатора равна 1, второго – равна 0 ($p^{(1)} = 1$, $p^{(2)} = 0$),

интегральный показатель мощности $P_{12} = 0$. Эффективность первого классификатора равна 0.5, второго — 0 ($e^{(1)} = 0.5$, $e^{(2)} = 0$), что даёт $E_{12} = 0$. Наплываемость первого классификатора равна 0.5, второго – 1, интегральная мера неразделимости $G_{12} = 0.75$. Для стандартных метрик accuracy, precision, recall и F_1 равны $2/3$.

3. **Два полностью совпадающих гауссиана**. Классы неразделимы (рисунок 3.2в). Мощность обоих классификаторов равна нулю ($p^{(1)} = p^{(2)} = 0$), что даёт $P_{12} = 0$. Эффективность также равна нулю ($E_{12} = 0$). Наплываемость обоих классификаторов максимальна ($G_{12} = 1$). При этом accuracy, precision, recall и F_1 принимают значение 0.5.



а) разделимые классы б) второй класс вложен в первый в) совпадающие классы

Рисунок 3.2 — Модельные ситуации для анализа метрик

Таким образом, видно, что предложенные показатели позволяют различать случаи, в которых стандартные метрики дают одинаковые значения, но интерпретация существенно различается.

3.6.5 Обобщение на многоклассовый случай

Для системы из $C > 2$ унарных классификаторов аналогичные метрики могут быть вычислены попарно для каждой пары классификаторов, что позволяет получить полную картину взаимодействия классов. В то же время для оценки качества отдельных классификаторов достаточно использовать показатели мощности, а для анализа пересечений и эффективности – соответствующие обобщённые гармонические средние по всем парам. Такой подход обеспечивает

более информативную и детализированную оценку по сравнению с традиционными многоклассовыми метриками и учитывает особенности работы унарной схемы: независимость классификаторов, возможность отказа и линейную масштабируемость по числу классов.

3.7 Иллюстрация работы на модельных примерах

Для наглядной демонстрации описанного подхода были построены унарные классификаторы для одного, двух и четырёх классов на модельных данных. В каждом случае в качестве фона использовались равномерно распределённые точки на единичном квадрате $[0, 1]^2$, а положительные объекты представляли собой выборки из компактных, хорошо различимых распределений.

На рисунке 3.3 показана граница принятия решения, построенная унарным классификатором для одного класса. Видно, что модель успешно выделяет область высокой плотности положительного класса, отсекая фон.

На рисунке 3.4 приведены результаты построения двух независимых унарных классификаторов для двух классов. Каждый классификатор определяет свою область плотности, и итоговая классификация осуществляется по наибольшей из двух аппроксимаций.

Наиболее показательный случай – построение унарных классификаторов для четырёх классов с искусственно созданным дисбалансом. Один из классов содержит в семь раз больше наблюдений, чем другой, ещё один – в пять раз больше и ещё один в три раза больше. Тем не менее, благодаря независимому обучению каждого классификатора на своём классе и фоновом множестве, области приятия решения получаются хорошо различимыми и не искаженными из-за дисбаланса. Это подтверждает устойчивость метода к нарушению пропорций классов (рисунок 3.5).

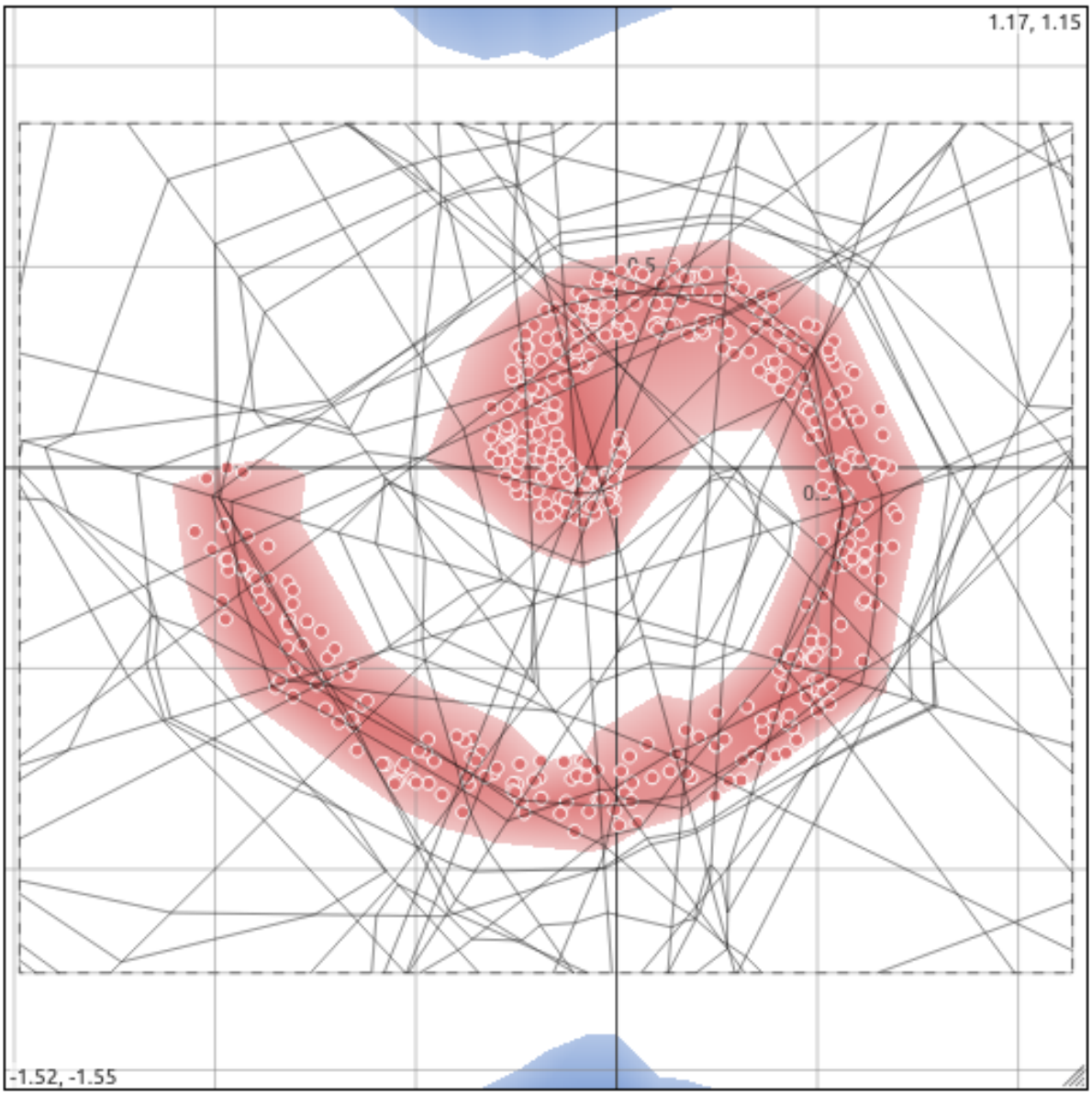


Рисунок 3.3 — Классификация данных одного класса с использованием унарной схемы

3.8 Работа на реальных данных

Практическая значимость любого алгоритма машинного обучения подтверждается его работоспособностью на реальных данных. Для оценки предложенного метода унарной классификации был выбран ряд общедоступных табличных наборов из репозитория UC Irvine Machine Learning Repository [65], типичных для задач анализа данных малой и средней размерности. Наборы включают Iris ($d = 4$), tic-tac-toe ($d = 9$), liver disease ($d = 10$), wine quality ($d = 11$) и heart disease ($d = 13$). Они различаются как по размерности, так

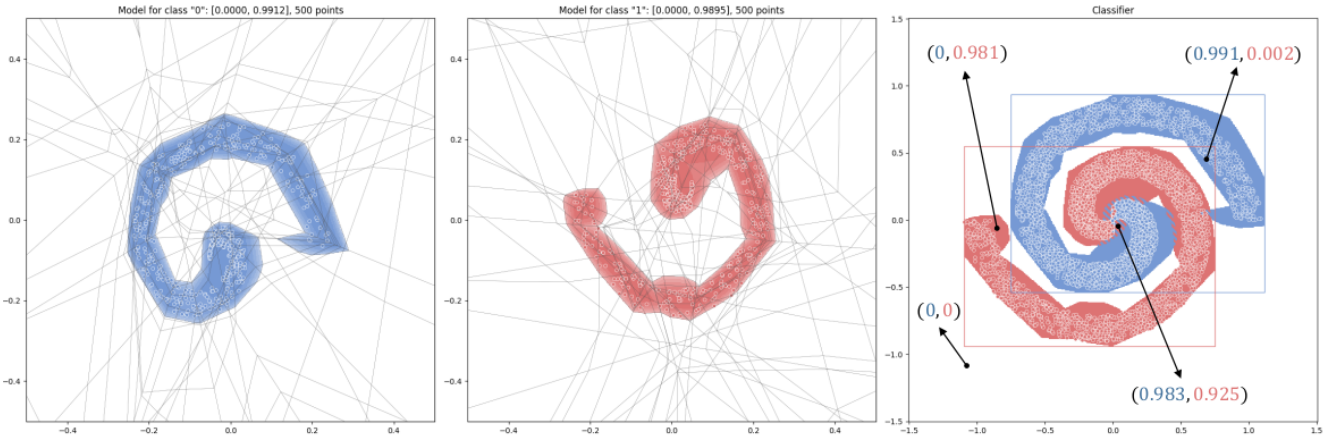


Рисунок 3.4 — Унарная классификация для двух классов

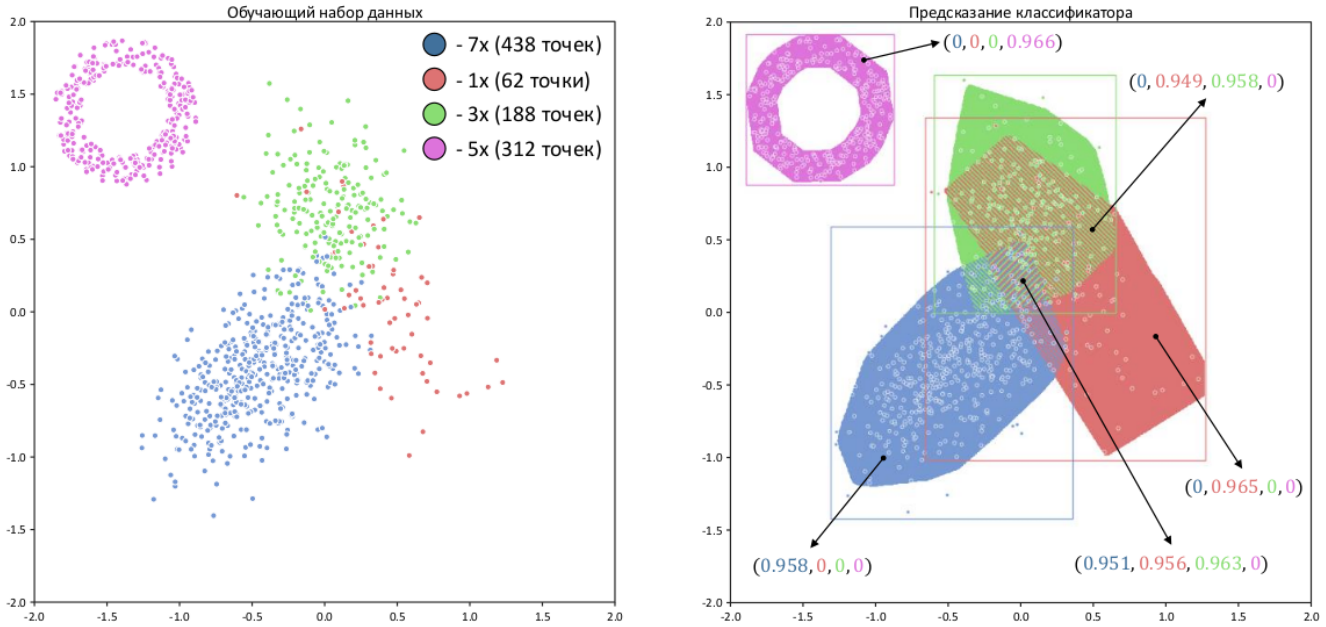


Рисунок 3.5 — Унарная классификация для четырёх классов с дисбалансом

и по степени выраженности дисбаланса классов: от сбалансированной выборки Iris (соотношение 1:1:1) до наборов с заметным преобладанием одного класса, таких как tic-tac-toe (1:1.7) и wine quality (1:2). Такой подбор позволяет проверить метод в условиях, приближенных к реальным задачам классификации, где дисбаланс является частым явлением.

Для сравнения в качестве базового практического решения использовался классификатор XGBoost – популярный и эффективный алгоритм, хорошо зарекомендовавший себя для работы с табличными данными. Цель сравнения – показать, что предложенный метод, обладая формальными теоретическими гарантиями, демонстрирует предсказательную способность, сопоставимую с широко применяемым на практике инструментом.

Результаты эксперимента, представленные в таблице 2, показывают, что метод унарной классификации демонстрирует конкурентоспособные результаты. На сбалансированном наборе Iris с малой размерностью метод показывает превосходство, что согласуется с теоретическим ожиданием его эффективности в подобных условиях. Важным является результат на наборе tic-tac-toe, где при наличии заметного дисбаланса (1:1.7) и умеренной размерности ($d = 9$) предложенный алгоритм также показывает более высокое качество. На наборах большей размерности (liver disease, wine quality) результаты методов довольно близки, а на heart disease ($d = 13$) унарный классификатор вновь демонстрирует небольшое преимущество. Эти результаты свидетельствуют о том, что подход не только сохраняет высокую предсказательную силу на реальных данных, но и проявляет ожидаемую устойчивость в условиях дисбаланса, эффективно работая как в малой, так и в средней размерности.

Таблица 2 — f_1 мера на реальных наборах данных

Набор данных	d	**Unary**	**XGBoost**
Iris	4	**0.941**	0.933
tic tac toe	9	**0.992**	0.967
Liver disease	10	0.815	**0.824**
Wine quality	11	0.789	**0.797**
Heart disease	13	**0.798**	0.788

3.9 Использование унарной классификации для обработки некомплектных данных

Одной из распространённых проблем прикладного машинного обучения является наличие некомплектных данных - наборов с пропущенными значениями некоторых признаков [66]. Классические методы заполнения (среднее, мода, k-ближайших соседей [67]) зачастую вносят систематическое смещение и не учитывают структуру задачи классификации. В качестве альтернативы в работе [7] был предложен метод, использующий унарную классификацию для вероятностного заполнения пропусков непосредственно в процессе обучения модели.

Суть метода заключается в итеративном дообучении персептрона на данных с пропусками. На каждой эпохе для объектов с некомплектными признаками пропуски временно заполняются случайными значениями. Полученный объект включается в обучающую выборку текущей эпохи с вероятностью, равной выходу обучаемой модели (функции $c_n^{(j)}(X)$ для класса j). Это позволяет адаптивно заполнять пропуски, ориентируясь на уверенность модели, и обучать классификатор без фиксированного искажения данных.

Метод демонстрирует практическую применимость унарной классификации для решения задачи обучения на неполных данных малой размерности, сохраняя стохастическую природу заполнения и избегая детерминированного смещения, характерного для классических методов [68]. Подробное описание алгоритма и экспериментов приведено в авторской работе [7].

3.10 Связь с современными архитектурами и направления развития

Несмотря на использование простой базовой модели – многослойного персептрона, предложенный подход унарной классификации концептуально согласуется с современными нейросетевыми архитектурами и может быть обобщён на более сложные классы моделей.

3.10.1 Свёрточные нейронные сети

Свёрточные нейронные сети являются одной из наиболее распространённых архитектур для обработки изображений и других пространственно организованных данных, благодаря способности эффективно выделять локальные признаки и сохранять инвариантность к сдвигам. Нетрудно показать, что операция свёртки может быть представлена как частный случай полносвязного линейного преобразования. Рассмотрим пример: пусть входное изображение X, ядро свёртки K и результат применения ядра на изображении Y заданы

матрицами

$$X = \begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{bmatrix}, \quad K = \begin{bmatrix} k_1 & k_2 \\ k_3 & k_4 \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 & y_2 \\ y_3 & y_4 \end{bmatrix},$$

где

$$Y = \begin{bmatrix} k_1x_1 + k_2x_2 + k_3x_4 + k_4x_5 & k_1x_2 + k_2x_3 + k_3x_5 + k_4x_6 \\ k_1x_4 + k_2x_5 + k_3x_7 + k_4x_8 & k_1x_5 + k_2x_6 + k_3x_8 + k_4x_9 \end{bmatrix}.$$

В векторной форме это преобразование можно записать как

$$x = \mathrm{vec}(X) \in \mathbb{R}^9, \quad y = \mathrm{vec}(Y) \in \mathbb{R}^4, \quad y = W \cdot x, \quad W \in \mathbb{R}^{4\times 9},$$

где матрица W имеет разреженную структуру, отражающую локальность взаимодействий и многократное использование одних и тех же весовых коэффициентов ядра свёртки:

$$W = \begin{bmatrix} k_1 & k_2 & 0 & k_3 & k_4 & 0 & 0 & 0 & 0 \\ 0 & k_1 & k_2 & 0 & k_3 & k_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 & 0 \\ 0 & 0 & 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix}.$$

Этот пример иллюстрирует, что свёрточный слой легко интерпретируется как полносвязное линейное преобразование с особой структурой матрицы весов. Это позволяет рассматривать перенос механизма оценки носителя распределения и процедуры отказа на свёрточные нейросети как естественное направление развития подхода, обеспечивая его применимость к задачам анализа изображений и других пространственно организованных данных.

3.10.2 Генеративно-состязательные сети

Можно продемонстрировать связь между предложенным подходом унарной классификации и принципами, лежащими в основе обучения генеративно-состязательных сетей (GAN), что позволяет подчеркнуть универсальность

подхода в задачах разделения распределений и оценки доверия к предсказаниям. В обоих случаях решается задача выделения эмпирического распределения реальных данных из шумового распределения.

Пусть P_{data} – распределение реальных данных, а P_{noise} – шумовое распределение. Тогда обучение классификатора сводится к минимизации функционала вида

$$\mathcal{L}(c) = \mathbb{E}_{x \sim P_{\text{data}}}[\ell(c(x), 1)] + \mathbb{E}_{x \sim P_{\text{noise}}}[\ell(c(x), 0)],$$

где c — классификатор, а ℓ – функция потерь бинарной классификации. По своей форме этот функционал совпадает с обучением дискриминатора в генеративно-состязательных сетях, за исключением характера фонового распределения: в GAN оно обычно задаётся нормальным распределением, тогда как в унарной классификации используется равномерное распределение на компакте.

Применение предложенного подхода может способствовать построению более надёжных дискриминаторов и, как следствие, теоретически улучшать качество генеративных моделей за счёт использования предсказаний с формализованной оценкой доверия и контроля области применимости.

3.10.3 Дальнейшее развитие

Указанная связь с современными архитектурами демонстрирует потенциал расширения предложенного подхода на более сложные модели, включая свёрточные, рекуррентные и трансформерные архитектуры. Такое развитие открывает перспективы построения нейросетевых моделей, сочетающих высокую выразительную способность с формализованной оценкой области применимости и уровня доверия к предсказаниям, что является ключевым аспектом доверенного искусственного интеллекта.

3.11 Выводы

Представленный в данной главе подход к построению унарных классификаторов на основе доверенного персептрона позволяет решить ключевую

проблему бинарной модели – чувствительность к дисбалансу классов. Поскольку каждый класс моделируется независимо, исключается систематическое смещение оценок, вызванное дисбалансом в данных, что повышает статистическую обоснованность и, как следствие, доверие к системе в условиях реальных задач с неравномерным распределением объектов.

Метод обеспечивает высокую степень интерпретируемости и гибкости: независимые модели для каждого класса могут использовать различные архитектуры и параметры, адаптированные под специфику соответствующих распределений. Векторная оценка апостериорных вероятностей, получаемая на выходе ансамбля унарных классификаторов, позволяет не только выполнять надёжную многоклассовую классификацию, но и реализовывать сложные сценарии принятия решений – такие как отказ от классификации при недостаточной уверенности или уточняющий запрос в активном обучении.

Таким образом, предложенная схема формирует основу для построения доверенных классификаторов, которые сочетают в себе устойчивость к дисбалансу, статистическую строгость подхода и практическую гибкость, необходимую для использования в ответственных приложениях.

Глава 4. Применение унарной классификации для генерации синтетических табличных данных

В контексте доверенного искусственного интеллекта важной задачей становится обеспечение конфиденциальности данных, используемых при обучении моделей. Прямая передача обученных моделей в регулируемых областях (медицина, финансы) может приводить к рискам обратного восстановления чувствительных обучающих выборок по параметрам модели [69], что ставит под угрозу приватность исходных данных. Одним из ключевых решений этой проблемы является генерация синтетических табличных данных, особенно актуальная в условиях ограниченного доступа к реальным данным [70]. Такие ограничения могут быть обусловлены законодательными мерами по защите персональных данных [71], коммерческой тайной или просто недостаточным объёмом исходной выборки.

Синтетические данные находят применение в нескольких критически важных сценариях доверенного ИИ: в безопасной передаче информации между организациями, обучении моделей без доступа к оригиналам, увеличении объёма обучающих данных [6] и обеспечении воспроизводимости научных исследований [72]. Основное требование к таким данным — сохранение статистических и/или структурных свойств оригинального распределения при гарантии отсутствия утечки чувствительной информации [73], что исключает прямое копирование реальных наблюдений.

Для генерации синтетических данных применяются как классические статистические методы, так и модели, основанные на машинном обучении [74] – например, вариационные автоэнкодеры (VAE) [75], генеративно-состязательные сети (GAN) [76], диффузионные модели [77] и др. [78]. Принципиальное различие между ними заключается в наличии формальных гарантий: статистические подходы зачастую обладают свойством состоятельности, обеспечивая сходимость оценок к истинному распределению, тогда как для нейросетевых методов такие строгие теоретические обоснования, как правило, отсутствуют, несмотря на их высокую эмпирическую эффективность.

4.1 Постановка задачи

Рассмотрим множество наблюдений $X = \{x_1, x_2, \ldots, x_n\} \in [0,1]^d$, представляющее собой выборку из неизвестного распределения. Цель состоит в построении синтетической выборки $\tilde{X} = \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_m\}$, которая сохраняет геометрические и статистические свойства оригинального распределения (рисунок 4.1).

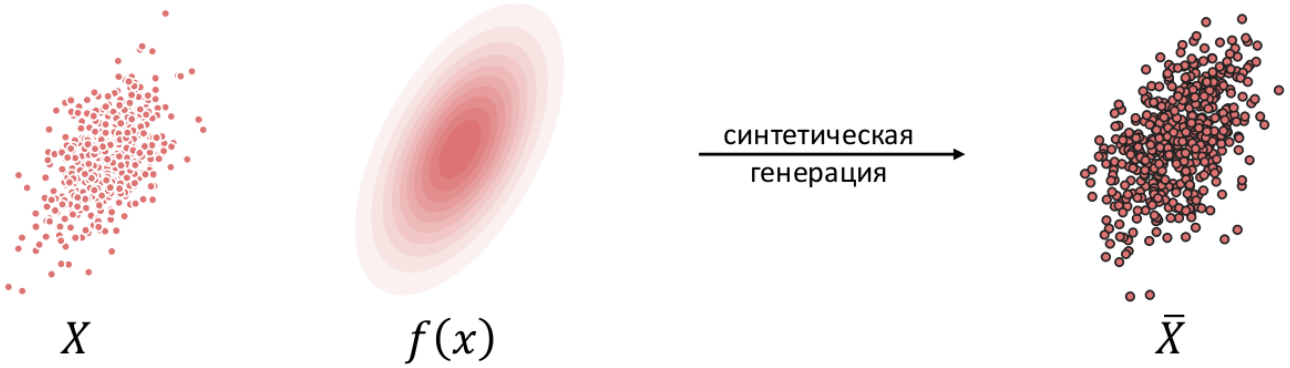


Рисунок 4.1 — Схематичное представление задачи создания синтетических данных

В отличие от традиционных генеративных подходов, стремящихся к точному восстановлению многомерной плотности распределения, предлагаемый метод фокусируется на сохранении геометрической структуры данных и их статистических характеристик. Это делает его особенно подходящим для задач синтетического расширения данных в условиях, где важнее сохранение топологии множества, чем точное соответствие плотности распределения.

4.2 Метод создания синтетических (репродукционных) данных

Для решения задачи генерации синтетических данных, сохраняющих геометрические и статистические свойства исходной выборки, предлагается метод, основанный на задаче унарной классификации и описанный в авторской работе [5]. Процесс включает два последовательных этапа: обучение классификатора и генерацию синтетической выборки, формальное описание которых

представлено в алгоритме 1. Данный подход не требует предположений о параметрической форме распределения, адаптивен к сложным зависимостям и позволяет контролировать баланс между точностью воспроизведения геометрии и разнообразием генерируемых точек.

4.2.1 Обучение классификатора

В рамках предложенного подхода рассматривается фоновое распределение – равномерное на компакте $[0,1]^d$. Из него отбирается множество точек $B = \{b_1, b_2, \ldots, b_n\}$, равное по мощности множеству X.

На объединённой выборке $X \cup B$ унарно обучается многослойный персептрон $c_n(x) : [0,1]^d \to [0,1]$. Метки классов задаются следующим образом:

$$\begin{cases} c_n(x) \to 1, & \text{если } x \in X, \\ c_n(b) \to 0, & \text{если } b \in B, \end{cases}$$

Для обучения модели используется функция потерь среднеквадратичной ошибки (MSE):

$$L = \sum_{x \in X} (1 - c_n(x))^2 + \sum_{b \in B} (0 - c_n(b))^2.$$

Выбор MSE вместо кросс-энтропии обусловлен желанием получить гладкую аппроксимацию функции плотности. В отличие от кросс-энтропии, которая стремится к резкому разделению классов, MSE интерпретируется как регрессионная функция, позволяющая трактовать выход сети как сглаженную аппроксимацию плотности без необходимости нормировки.

Особенность метода – генерация новых фоновых точек на каждом обучающем шаге (эпохе), а не фиксированное множество B, заданное в начале. Это обеспечивает более полное покрытие области и снижает переобучение на конкретных фоновых примерах.

4.2.2 Создание репродукционных данных

После завершения обучения классификатора, синтетические данные получаются путём фильтрации новых фоновых точек. Из равномерного распределения на $[0,1]^d$ сэмплируется множество $\tilde{B}$, и каждая точка $\tilde{b} \in \tilde{B}$ включается в итоговую выборку с вероятностью $c_n(\tilde{b})$. То есть:

$$\tilde{X} = \{\tilde{b} \in \tilde{B} \mid \xi < c_n(\tilde{b})\}, \quad \xi \sim \text{Uniform(0,1)}.$$

Такой подход позволяет строить выборку, приближенную к оригинальному носителю данных. Это достигается за счёт того, что точки сэмплируются пропорционально вероятности $P(Y = 1|X)$ бинарного классификатора $c_n(X)$, обученного отличать реальные данные от фонового шума.

Дополнительно возможен вариант репродукции по гистограмме выходов модели. Для этого после вероятностного прореживания строится гистограмма значений $c_n(x)$ на обучающей выборке. Далее синтетические точки $\tilde{X}$ отбираются случайным образом так, чтобы распределение значений $c_n(\tilde{x})$ совпадало с исходной гистограммой. В случае генерации выборки той же мощности, что и обучающая, совпадение достигается в абсолютных числах; при построении выборки произвольного размера обеспечивается совпадение относительных частот. Такой механизм позволяет контролировать форму распределения, повышает согласованность синтетических данных с оригинальной структурой и снижает дисперсию репродукционных данных. Особенно полезен данный вариант в задачах, где требуется корректная передача хвостов распределения, так как он снижает риск их недопредставленности в сгенерированной выборке.

4.3 Экспериментальное исследование

4.3.1 Эксперименты на модельных данных

Для наглядной демонстрации эффективности метода проведены эксперименты на модельных наборах данных с известной структурой (рисунок 4.2).

Метод 1: Создание синтетических табличных данных

Вход : исходная выборка $X = \{x_1, \ldots, x_n\} \in [0,1]^d$
архитектура модели $(L,\ k)$
мощность синтетической выборки m

Выход : синтетическая выборка $\tilde{X}$

Инициализация: создать случайный персептрон $c_n(X)$

for *epoch* $\leftarrow 1$ **to** E **do**

Сгенерировать фоновую выборку $B = \{b_1, \ldots, b_n\} \sim U[0,1]^d$

Обучить $c_n(x)$ на $X \cup B$ с функцией потерь L:
$L = \sum_{x \in X}(1 - c_n(x))^2 + \sum_{b \in B}(0 - c_n(b))^2$

$\tilde{X} \leftarrow \varnothing$

while $|\tilde{X}| < m$ **do**

Сгенерировать точку $\tilde{b} \sim U[0,1]^d$

Сгенерировать $\xi \sim U(0,1)$

if $\xi < c_n(\tilde{b})$ **then**

$\tilde{X} \leftarrow \tilde{X} \cup \{\tilde{b}\}$

Вернуть $\tilde{X}$

Это позволяет объективно оценить способность модели к воспроизведению статистических свойств. Рассматривались следующие выборки:

- **Спираль**: двумерная выборка, где точки образуют спираль. Проверяется способность метода к моделированию нелинейной кластеризации.
- **Два квадрата**: два раздельных кластера квадратной формы. Оценивается сохранение пространственной структуры и разделимости.
- **Нормальное распределение**: двумерное распределение с известными параметрами. Проверяется соответствие ковариационной структуры.
- **Многомерное нормальное распределение**: 10-мерный аналог предыдущего случая, оценивающий качество генерации в высокоразмерном пространстве.

Визуализация результатов (рисунок 4.3) демонстрирует, что сгенерированные данные точно воспроизводят форму, плотность и вариативность оригинальных данных. В случае многомерного нормального распределения со-

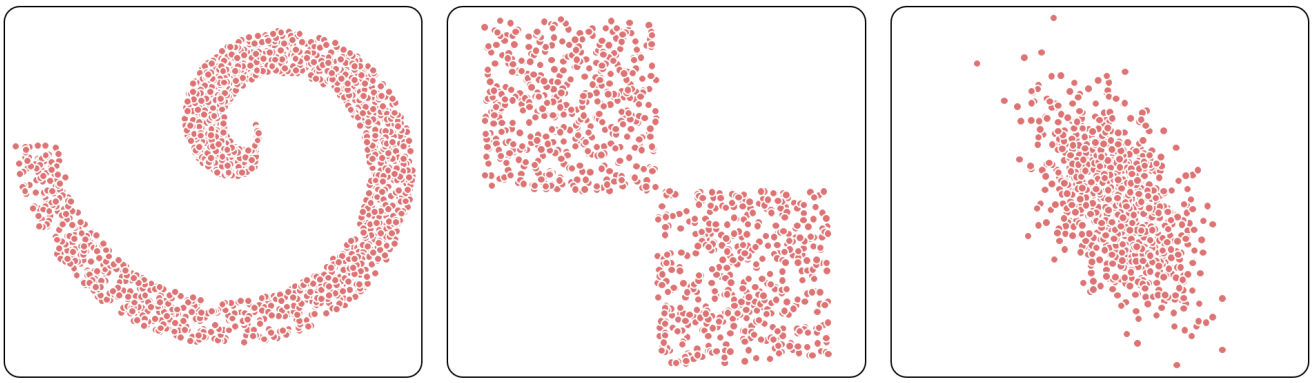
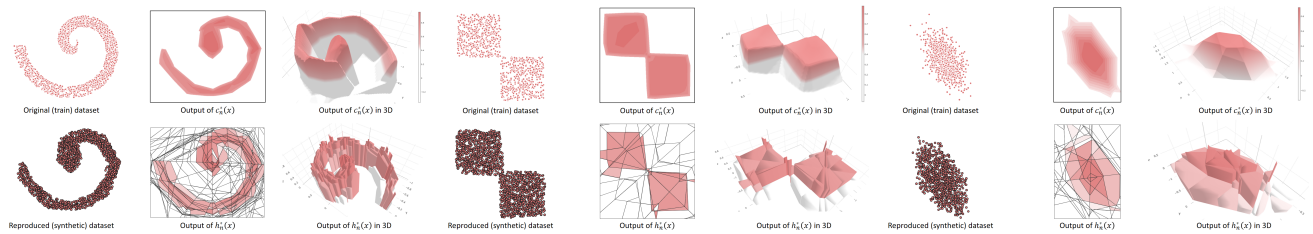


Рисунок 4.2 — Использованные наборы данных для построения репродукционных выборок (спираль, два квадрата и гауссиан)

храняется ковариационная структура, хотя наблюдается небольшое увеличение дисперсии – эффект, обусловленный ростом размерности и разрежённостью пространства.



а) Спираль б) Два квадрата в) Смесь нормальных распределений

Рисунок 4.3 — результаты эксперимента с синтетическими данными

4.3.2 Сравнение методов генерации на модельных данных

Для оценки качества предложенного подхода был проведён сравнительный анализ с нейросетевыми генеративными моделями: вариационным автоэнкодером (VAE) и генеративно-состязательной сетью (GAN). В качестве тестового примера использовался двумерный спиральный набор данных, позволяющий наглядно проверить способность моделей к воспроизведению сложной нелинейной структуры распределения.

Вариационный автоэнкодер (VAE) продемонстрировал ограниченную способность к воспроизведению глобальной геометрической структуры данных

(рисунок 4.4). Генерация точек концентрировалась преимущественно в области среднего распределения, что приводило к размытию формы спирали. Такой эффект объясняется свойственной VAE тенденцией усреднять латентное пространство, что особенно заметно при генерации данных с выраженной многообразной структурой.

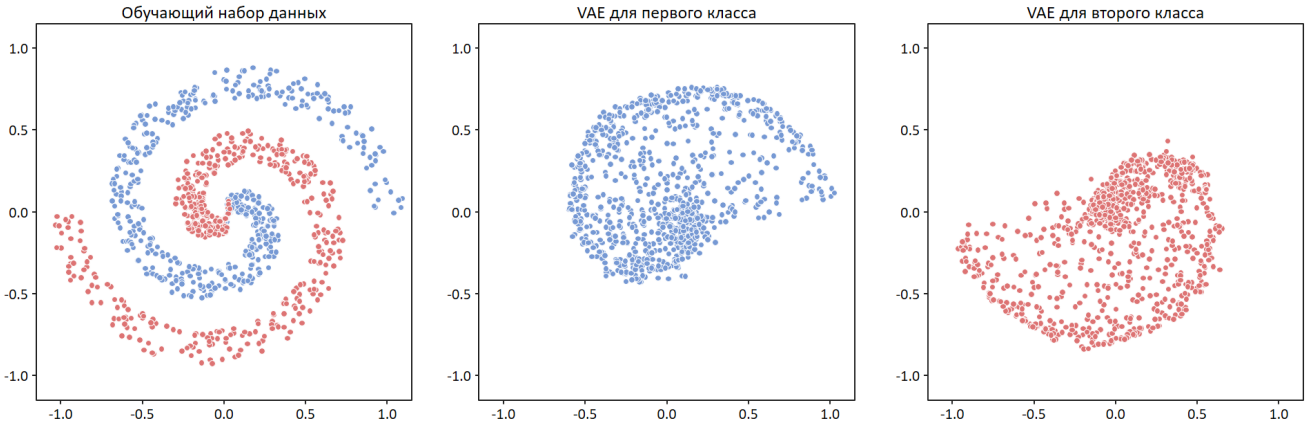


Рисунок 4.4 — VAE модель

Генеративно-состязательная сеть (GAN) показала более высокую способность к сохранению формы спирали по сравнению с VAE (рисунок 4.5). Однако наблюдается искажение геометрии в виде сужения спиральных ветвей. Модель стремится усиливать локальные плотности, что приводит к чрезмерному уплотнению точек вдоль траектории спирали и потере равномерности распределения. Этот эффект характерен для GAN в условиях ограниченного объёма обучающих данных и высокой сложности целевого распределения.

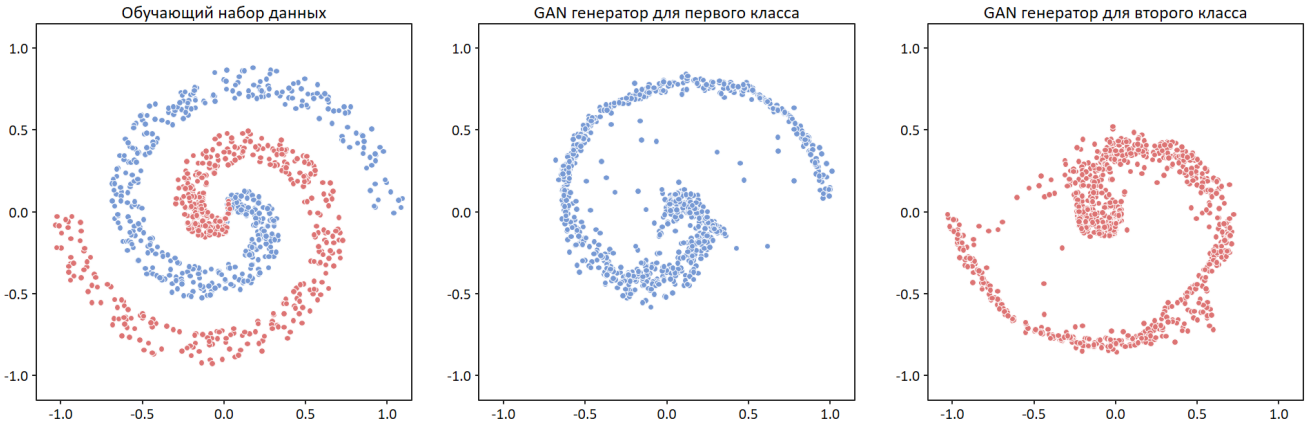


Рисунок 4.5 — GAN модель

Предложенный метод, основанный на унарной классификации, в отличие от VAE и GAN, продемонстрировал способность воспроизводить как глобальную форму, так и локальную плотность данных (рисунок 4.6). Предлагаемый

метод сохраняет равномерность распределения вдоль спирали и при этом избегает чрезмерного сглаживания или концентрации точек, что обеспечивает более устойчивое воспроизведение сложных структур.

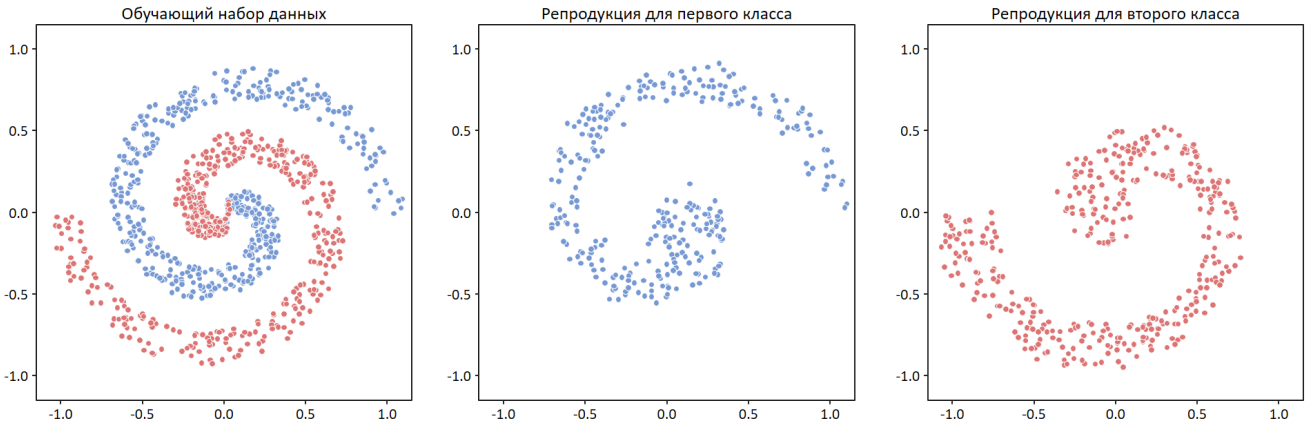


Рисунок 4.6 — Предложенный метод

Таким образом, в задаче генерации структурированных данных низкой размерности метод на основе унарной классификации демонстрирует более высокую устойчивость и точность воспроизведения геометрической формы по сравнению с GAN и VAE.

4.3.3 Эксперименты на реальных данных

Для верификации практической применимости предложенного метода были проведены эксперименты на реальных наборах данных с последующей оценкой метрик полезности (utility) и верности (fidelity).

Метрики оценки

– *Utility* оценивает сохранение полезности данных для последующих задач машинного обучения. В качестве модели выбран градиентный бустинг на деревьях (XGBoost [79]), демонстрирующий высокую эффективность классификации табличных данных. Оцениваются значения F_1 мер моделей, обученных на синтетических и оригинальных обучающих

данных соответственно, и протестированных на одном и том же реальном тестовом множестве.

– *Fidelity* измеряет статистическую близость синтетической выборки к исходной. Для её оценки использовался пакет SDMetrics [80], а именно – усреднённое значение статистики Колмогорова–Смирнова по всем столбцам. Значение метрики лежит в диапазоне $[0, 1]$, где 1 соответствует полному статистическому совпадению.

Базовые модели сравнения

В качестве базового генеративного метода выбран Conditional Tabular GAN (CTGAN) [81] – генеративно-состязательная сеть, специально разработанная для табличных данных, эффективно моделирующая смешанные распределения. Данный метод является популярным бенчмарком в области генерации табличных данных, не требующим при этом большого числа вычислительных ресурсов в отличие от современных диффузионных моделей для табличных данных.

4.3.4 Наборы данных

Тестирование проводилось на пяти наборах данных из репозитория UC Irvine Machine Learning Repository [65]:

– **Iris** ($d = 4$): 150 образцов, 3 класса, все признаки числовые.
– **Tic-Tac-Toe** ($d = 9$): 958 образцов, категориальные признаки.
– **Liver disease** ($d = 10$): 1700 образцов, смешанные признаки (числовые и бинарные).
– **Wine quality** ($d = 11$): 4898 образцов, все признаки числовые.
– **Heart Disease** ($d = 13$): 303 образца, смешанные признаки (числовые и бинарные).

Результаты

В таблице 3 представлены результаты оценки метрики полезности (*utility*). Значения F_1-score демонстрируют, что классификатор, обученный на синтетических данных, сохраняет конкурентоспособное качество на реальных тестовых данных, особенно на наборах малой размерности (Iris). С ростом размерности полезность сгенерированных данных с помощью предложенного метода постепенно снижается, однако на наборе Heart disease качество классификатора, обученного на синтетических данных, наблюдается заметный прирост качества по сравнению с обучением на реальных тренировочных данных.

В таблице 4 приведено сравнение статистической верности синтетических данных, сгенерированных предложенным методом и CTGAN. На низкоразмерном наборе Iris метод демонстрирует лучший результат, что согласуется с его концепцией сохранения геометрической структуры данных. С увеличением размерности преимущество CTGAN становится заметным, что объясняется его более сложной архитектурой, ориентированной на точное моделирование многомерных распределений.

Таблица 3 — Оценка полезности (utility) методов генерации синтетических данных

Набор данных	d	F_1 мера на тестовом наборе		
		train	**Unary**	**CTGAN**
Iris	4	0.933	0.973	0.920
tic tac toe	9	0.967	0.957	0.667
Liver disease	10	0.824	0.731	0.682
Wine quality	11	0.797	0.698	0.718
Heart disease	13	0.788	**0.838**	0.727

Таблица 4 — Оценка верности (fidelity) методов генерации синтетических данных

Набор данных	d	**Unary**	**CTGAN**
Iris	4	**0.901**	0.880
tic tac toe	9	0.858	**0.935**
Liver disease	10	0.875	**0.922**
Wine quality	11	0.857	**0.887**
Heart disease	13	0.852	**0.882**

Выводы по экспериментам

– Предложенный метод демонстрирует высокую *utility* на данных малой и средней размерности ($d \leqslant 13$), что подтверждает его способность сохранять репрезентативность данных для последующих задач классификации.
– В области *fidelity* наблюдается ожидаемый эффект “проклятия размерности”: с ростом d точность статистического воспроизведения снижается. Однако на низкоразмерных данных (Iris) метод превосходит CTGAN, что можно объяснить его фокусом на сохранении геометрической структуры (support) распределения, а не плотности.
– Полученные результаты определяют область эффективного применения метода – генерация синтетических данных для задач, где критически важна сохранность геометрии исходного множества, а не точное воспроизведение многомерной плотности. Для задач, требующих высокой статистической точности на данных высокой размерности, предпочтительнее использование более сложных моделей, таких как CTGAN или диффузионные модели.

4.4 Выводы

Предложенный метод построения репродукционных выборок с помощью метода унарной классификации представляет собой эффективный подход для задач генерации синтетических данных с акцентом на сохранение геометрической структуры исходного распределения. Ключевыми преимуществами метода являются его концептуальная простота, теоретическая интерпретируемость и способность точно воспроизводить топологию низкоразмерных многообразий данных.

Полученные результаты определяют чёткую область эффективного применения метода – генерация синтетических данных малой размерности, где его производительность сопоставима с более сложными генеративными моделями.

При этом метод демонстрирует заметное преимущество в вычислительной эффективности и устойчивости при работе с ограниченными выборками.

Основное ограничение метода связано с эффектом «проклятия размерности»: с увеличением количества признаков точность воспроизведения статистических характеристик закономерно снижается, что делает его менее подходящим для задач, требующих точного моделирования высокоразмерных плотностей распределений. В таких сценариях более уместно применение специализированных моделей, таких как CTGAN или диффузионные модели.

Глава 5. Интеллектуальная система машинного обучения для визуализации и исследования методов классификации

Для формулирования исследовательских гипотез и последующей их экспериментальной проверки разработана интеллектуальная система машинного обучения, отвечающая совокупности требований:

1. **Автономность и кроссплатформенность**.Работоспособность на устройствах без графического процессора и доступа к сети, не требующая установки дополнительного программного обеспечения, для обеспечения воспроизводимости в изолированных средах
2. **Интерактивная визуализация**. Возможность визуализации обучающих данных, аппроксимации носителя распределения, архитектуры модели, её выходных значений и разбиения компакта, а также динамики метрик обучения без многократной перерисовки интерфейса.
3. **Динамическая модификация параметров**. Возможность динамической модификации используемых данных и архитектуры сети в реальном времени без прерывания процесса обучения.
4. **Численная корректность**. Численная эквивалентность вычислительного ядра системы эталонной реализации на платформе PyTorch.
5. **Реализация разработанных методов** (модифицированная бинарная классификация, унарная классификация, генерация синтетических данных на основе метода унарной классификации, объясняющее дерево eXBTree) и инструментов для проведения экспериментов с ними.

В данной главе описывается архитектура и программная реализация системы, разработанной в соответствии с приведёнными требованиями.

5.1 Общая характеристика интеллектуальной системы машинного обучения

Разработанная система представляет собой автономное клиентское WEB-приложение, реализованное на языке JavaScript [82], не требующее установки, интернет-соединения или использования графического ускорителя, что обеспе-

чивает его широкую доступность и воспроизводимость экспериментов, а также удовлетворяет требованию автономности и кроссплатформенности.

Интеллектуальная система предназначена для комплексной демонстрации, отладки и тестирования алгоритмов, описанных в теоретических разделах настоящей работы. Предоставляется интуитивно понятный графический интерфейс с возможностью гибкой настройки параметров моделей, наборов данных и условий обучения. Благодаря использованию визуальных компонентов пользователь может в интерактивном режиме наблюдать за процессом формирования разделяющих поверхностей, анализировать выходы моделей, а также проводить тестирование работы классификаторов.

Разработка велась с учётом необходимости масштабируемости архитектуры: структура системы разделена на независимые функциональные блоки, что обеспечивает возможность расширения и модификации без необходимости переписывания всего кода. Интерфейс системы логически организован по вкладкам, каждая из которых отвечает за определённую группу задач: генерация и загрузка данных, обучение модели, проведение экспериментов, визуализация и анализ результатов.

На момент завершения работы интеллектуальная система машинного обучения включает в себя следующие ключевые функциональные возможности:

- настройка параметров архитектуры многослойного персептрона, включая размеры и количество слоёв, выбор функции активации, установку порогов доверия;
- управление параметрами обучения (функция потерь, оптимизатор, регуляризация, и т.д.);
- пошаговая визуализация процесса обучения, включая изменение выходов модели, метрик и формирование ячеек;
- реализация как классических методов бинарной классификации, так и модифицированного и унарного методов;
- визуализация, построение и загрузка обучающих и тестовых множеств;
- проведение экспериментальных исследований по созданию синтетических данных и анализ объясняющего двоичного дерева eXBTree.

Таким образом, система реализует весь цикл исследования: от создания обучающего множества до визуализации результатов и анализа поведения модели в различных условиях. Её применение позволяет не только демонстрировать основные методы, описанные в главах 1–3, но и проводить дополнительный

количественный и качественный анализ, направленный на верификацию теоретических положений.

5.2 Архитектура системы

5.2.1 Модульная организация и паттерн проектирования EventEmitter

Архитектура системы построена на модульном принципе, где каждый компонент имеет чётко определенную ответственность и интерфейс взаимодействия. Основные модули включают:

- Вычислительное ядро (`model`) – реализация нейросетевых алгоритмов;
- Управление данными (`data`) – генерация, загрузка, хранение и обработка наборов данных;
- Система визуализации (`view`) – отрисовка всех графических компонентов;
- Модули экспериментов (`experiments`) – проведение исследований разработанных методов.

Для организации взаимодействия между модулями применён паттерн `Observer`, реализующий событийно-ориентированную архитектуру (рис. 5.1). Каждый модуль, требующий реакции на изменения состояния, наследуется от базового класса `EventEmitter` и может генерировать события, на которые подписываются другие компоненты (листинг 5.1).

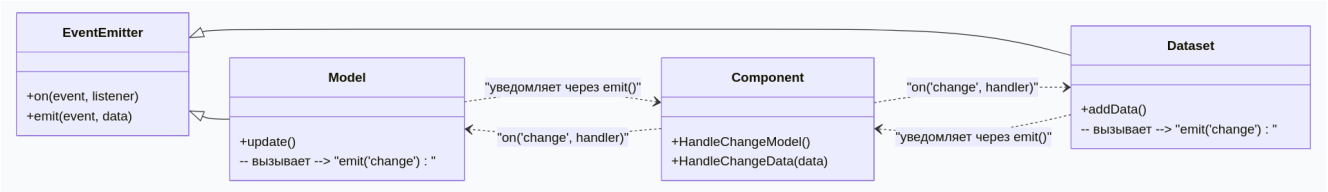


Рисунок 5.1 — Архитектура на основе EventEmitter

Листинг 5.1: Пример использования EventEmitter для реакции на изменения

```
this.model.on("change", () => this.HandleChangeModel())
this.dataset.on("change", (data) => this.HandleChangeData(data))
```

Этот подход обеспечивает дифференциальное обновление интерфейса: при изменении набора данных перерисовываются только точки на графике, а при обновлении весов модели – только её визуализация, что минимизирует накладные расходы на рендеринг по сравнению с полной перерисовкой всего интерфейса с некоторым интервалом.

5.2.2 Вычислительное ядро и интерфейс

Система имеет чёткое разделение между логикой вычислений и пользовательским представлением:

- Класс `Visualizer` – центральный компонент, управляющий всеми вычислительными процессами. Может использоваться программно без графического интерфейса в виде независимой библиотеки.
- Класс `Playground` – адаптер, связывающий HTML-компоненты с методами класса `Visualizer`. Такое разделение обеспечивает лёгкую расширяемость и возможность интеграции системы в другие приложения.

5.2.3 Система событий для минимизации перерисовки интерфейса

Механизм событий позволяет компонентам системы реагировать на изменения состояния только тех элементов, от которых они зависят. Каждый компонент может генерировать события при изменении своего внутреннего состояния и подписываться на события других компонентов, которые влияют на его отображение или вычисления.

Ключевые типы событий

Система использует следующие основные категории событий:

- `change` – общее событие изменения состояния (используется моделью, наборами данных, метриками);
- `change-architecture` – изменение архитектуры нейронной сети;
- `change-prediction` – обновление предсказаний модели для конкретного набора данных;
- `change-dimension` – изменение размерности входных данных;
- `change-view` – изменение области просмотра визуализации;
- `click` – пользовательское взаимодействие с элементами интерфейса с помощью мыши.

Динамика взаимодействия через события

Взаимодействие компонентов организовано по принципу публикации-подписки:

- При изменении внутреннего состояния компонент генерирует соответствующее событие.
- Все компоненты, подписанные на это событие, получают уведомление и выполняют необходимые действия.
- Эти действия могут включать обновление собственного состояния и генерацию новых событий, что приводит к каскадному распространению изменений по системе.

Пример последовательности событий

Типичный сценарий взаимодействия после шага обучения модели представлен на рис. 5.2 и включает следующие этапы:

1. Менеджер модели изменяет веса сети и генерирует событие `change`.
2. Визуализатор модели, подписанный на это событие, обновляет отображение архитектуры и выходной поверхности.

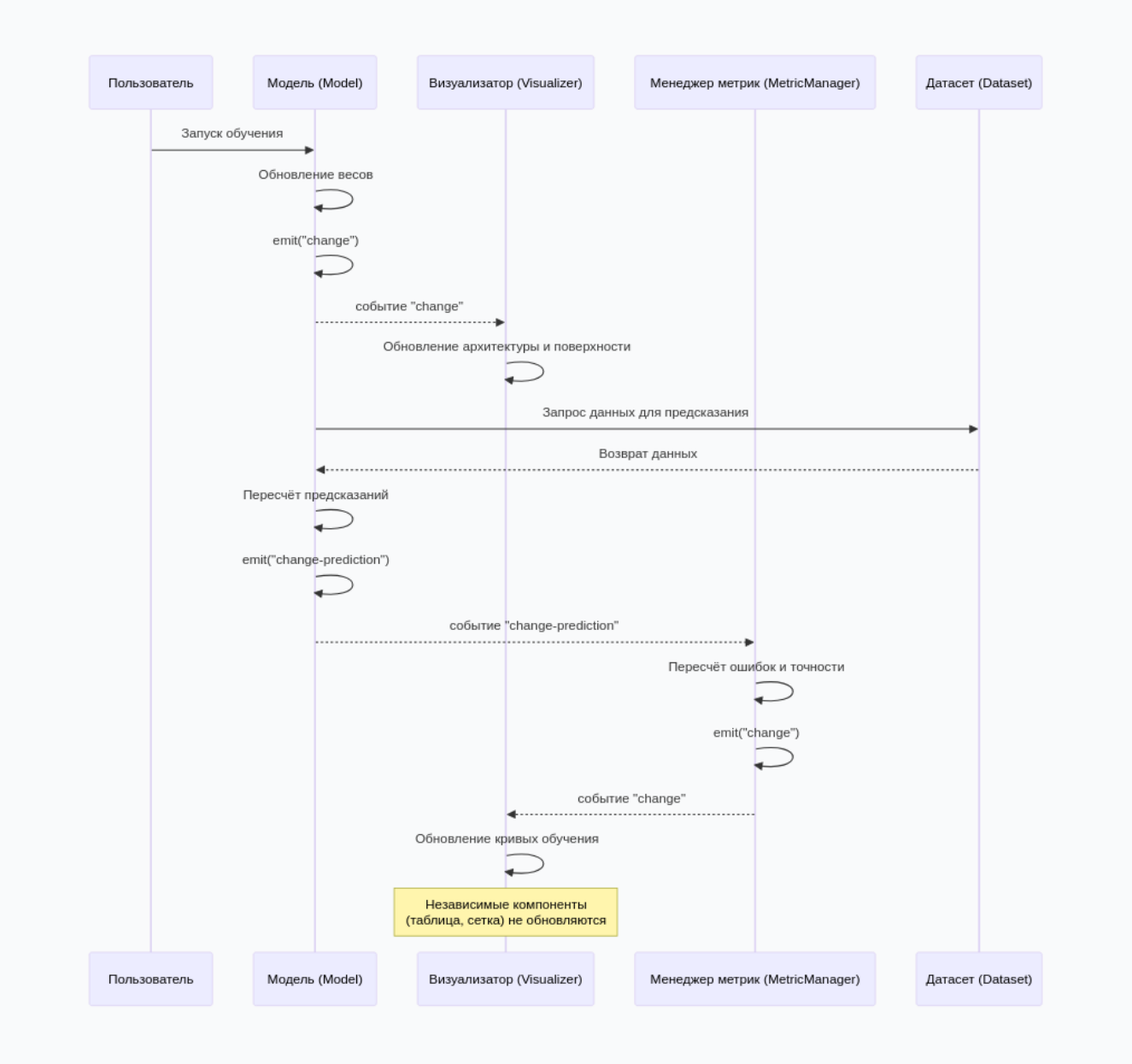


Рисунок 5.2 — Последовательность событий после шага обучения модели

3. Сам менеджер модели запускает пересчёт предсказаний для всех активных наборов данных, после чего генерируется событие `change-prediction`.
4. Модуль метрик, подписанный на `change-prediction`, пересчитывает значения ошибок и точности, затем генерирует событие `"change"`.
5. График метрик, подписанный на `"change"` от модуля метрик, обновляет кривые обучения.

При этом независимые компоненты (таблица данных, координатная сетка и др.) не получают уведомлений и не перерисовываются, если они не зависят от изменённых параметров. Такая избирательная обработка событий минимизиру-

ет избыточные вычисления и обеспечивает высокую отзывчивость интерфейса даже при работе с большими объёмами данных.

5.3 Реализация вычислительного ядра машинного обучения

5.3.1 Основные компоненты нейросетевой подсистемы

Вычислительное ядро реализует полный набор компонентов для работы с нейросетевыми моделями (рис. 5.3):

– Полносвязный слой (FullyConnectedLayer) – поддерживает различные функции активации (ReLU, LeakyReLU, Abs, линейная), механизм отключения нейронов, методы прямого и обратного распространения.
– Модель нейронной сети (NeuralNetwork) – содержит список слоёв и методы обучения, предсказания и сериализации. Поддерживает динамическое изменение архитектуры во время работы.
– Функции потерь (Loss) – реализованы MSE, MAE, Huber [83] и LogCosh [84].
– Оптимизаторы [85] градиентного спуска (Optimizer) – включают SGD, Momentum SGD, Adam, Adamax, Adadelta [86], Adagrad, RMSprop с поддержкой L1/L2 регуляризации.

5.3.2 Оптимизация работы с памятью

Для минимизации накладных расходов на выделение памяти применены следующие стратегии:

– Типизированные массивы (`Float64Array`, `Int32Array`) – все вычисления выполняются над типизированными массивами [87], что обеспечивает непрерывное расположение данных в памяти и ускоряет доступ к элементам.

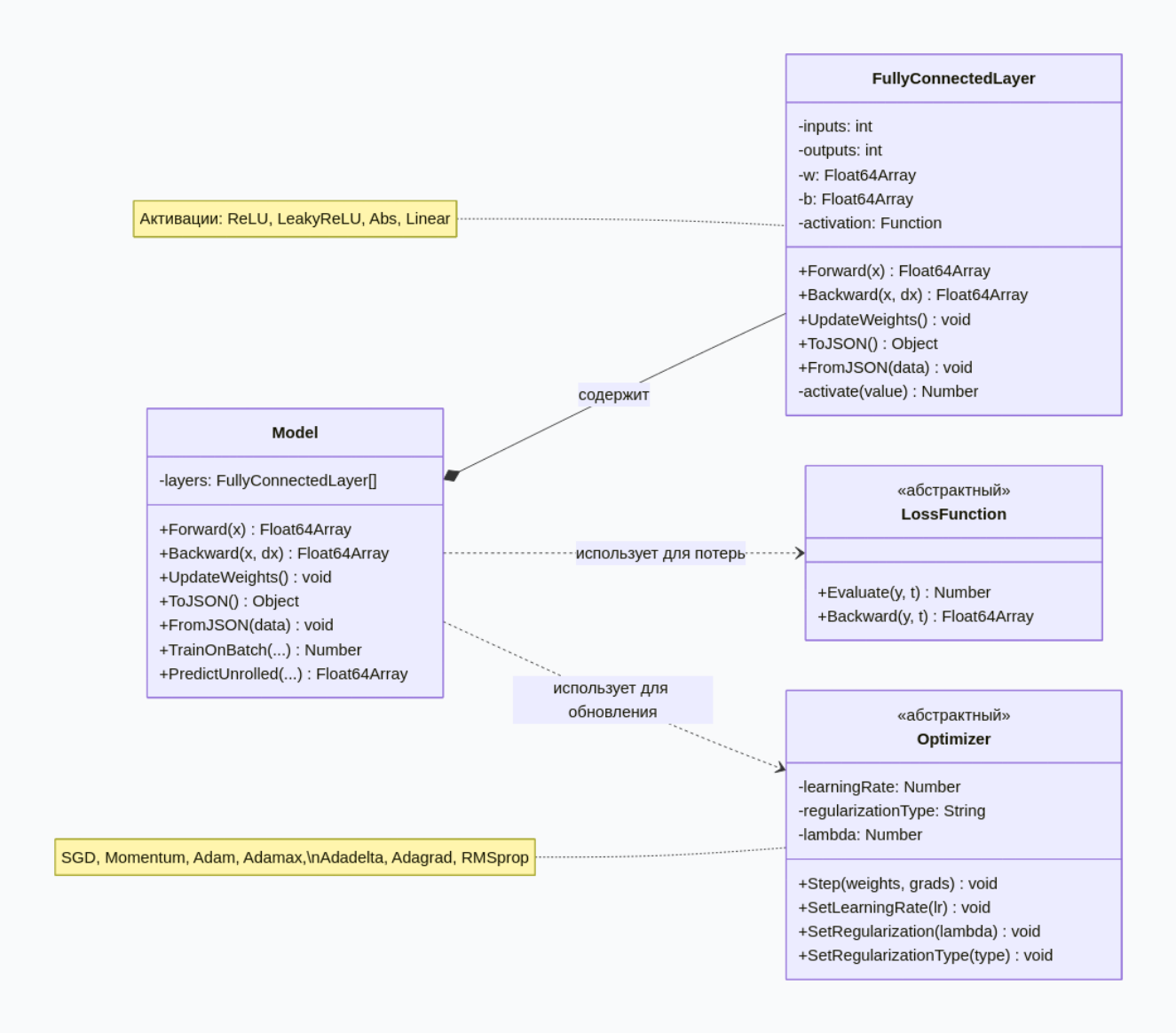


Рисунок 5.3 — Архитектура вычислительного ядра

– Преаллокация буферов – все рабочие массивы (выходы, активации, градиенты) выделяются один раз при создании слоя с учётом максимального размера пакета (см. листинг 5.2).
– Стратегия перевыделения – при изменении размеров модели массивы расширяются с сохранением существующих данных, но никогда не уменьшаются.

Листинг 5.2: Преаллокация буферов в полносвязном слое

```
class FullyConnectedLayer {
    constructor(inputs, outputs, activation, MAX_BATCH_SIZE) {
        this.value = new Float64Array(outputs * MAX_BATCH_SIZE)
        this.output = new Float64Array(outputs * MAX_BATCH_SIZE)
        this.df = new Float64Array(outputs * MAX_BATCH_SIZE)
        this.dx = new Float64Array(inputs * MAX_BATCH_SIZE)
```

```
            // ...
        }
    }
```

5.3.3 Разворачивание циклов для ускорения вычислений на CPU

Исполнение кода в однопоточной среде JavaScript накладывает фундаментальное ограничение: все вычисления выполняются последовательно в рамках одного потока. Для преодоления этого ограничения необходимо максимально эффективно использовать аппаратные возможности современного центрального процессора. Ключевой особенностью CPU является конвейерная архитектура, где выполнение каждой машинной инструкции разбито на последовательные стадии: выборку (fetch), декодирование (decode), исполнение (execute) и запись результата (write).

Зачастую в потоке команд следующая инструкция зависит от результата, который должна записать предыдущая (”зависимость «чтение после записи”). В этом случае процессор вынужден вводить в конвейер пустые такты и простаивать в ожидании завершения нужной стадии, что приводит к снижению производительности.

Однако, если в потоке исполнения появляются независимые инструкции, не связанные подобными зависимостями, ситуация меняется. Стадии конвейера могут быть заполнены непрерывно: пока одна независимая инструкция исполняется, следующая может декодироваться, а третья – выбираться. Это создаёт эффект параллельного исполнения на уровне инструкций (Instruction-Level Parallelism, ILP) внутри одного потока, позволяя процессору выполнять полезную работу на каждом такте и тем самым существенно ускоряя вычисления (рисунок 5.4).

Для генерации таких независимых инструкций в вычислительно нагруженных участках кода применяется метод разворачивания циклов (loop unrolling), который является ключевым аспектом оптимизации производительности в разработанной системе [88]. Наиболее частой и критичной с точки зрения производительности операцией в ядре нейронной сети является матричное умножение, лежащее в основе как прямого, так и обратного

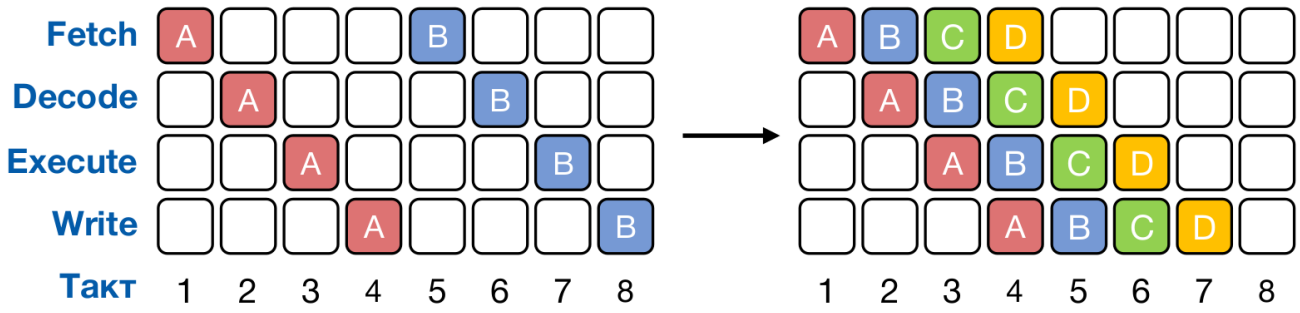


Рисунок 5.4 — Визуализация эффекта параллелизма на уровне инструкций (ILP)

распространения сигнала. В стандартной реализации операции прямого распространения имеют вид, представленный в листинге 5.3. В оптимизированной версии (листинг 5.4) цикл разворачивается на 4 итерации, что позволяет интерпретатору JavaScript лучше использовать конвейер процессора и регистры за счёт парралелизма на уровне инструкций.

Листинг 5.3: Базовая реализация прямого распространения

```
for (let i = 0; i < outputs; i++) {
    let value = this.b[i]
    for (let j = 0; j < inputs; j++)
        value += this.w[i * inputs + j] * x[batch * inputs + j]
    this.activate(index, value)
}
```

Листинг 5.4: Оптимизированная реализация прямого распространения

```
const end = (this.outputs >> 2) << 2
for (let i = 0; i < end; i += 4) {
    let value1 = this.b[i]
    let value2 = this.b[i + 1]
    let value3 = this.b[i + 2]
    let value4 = this.b[i + 3]
    let wOffset = i * this.inputs

    for (let j = 0; j < this.inputs; j++) {
        const xj = x[j]
        const wIdx = wOffset + j
        value1 += this.w[wIdx] * xj
        value2 += this.w[wIdx + this.inputs] * xj
        value3 += this.w[wIdx + this.inputs * 2] * xj
        value4 += this.w[wIdx + this.inputs * 3] * xj
    }
```

```
    // Активация для всех четырёх нейронов
}
```

Аналогичное разворачивание применено для операций обратного распространения ошибки и вычисления градиентов по входному вектору. Оптимальный коэффициент разворачивания (4 операции) был определён эмпирически: разворачивание на 2 итерации не обеспечивало достаточной степени параллелизма для эффективной загрузки конвейера процессора, в то время как увеличение коэффициента до 8 или 16 итераций не приводило к статистически значимому приросту производительности по сравнению с разворачиванием на 4 итерации.

5.3.4 Цикл обучения в системе

В системе реализован метод, выполняющий одну полную эпоху обучения на предоставленном обучающем наборе данных. Алгоритм его работы заключается в последовательной мини-пакетной обработке данных: выборка разбивается на батчи, для каждого из которых выполняются прямой проход модели, расчёт функции потерь, обратное распространение ошибки и шаг оптимизатора для обновления весовых коэффициентов. После завершения обработки всех пакетов генерируются события изменения модели, инициирующие каскадное обновление всех зависимых визуализаций и метрик.

Управление процессом обучения осуществляется через цикл анимации, построенный на основе функции `requestAnimationFrame`. На каждой итерации цикла проверяется состояние флага `isTraining`. Если обучение активно, вызывается метод выполнения эпохи `TrainStep`, что обеспечивает непрерывное итеративное обучение до явной остановки пользователем. Данный подход также предоставляет возможность пошагового режима, когда пользователь может самостоятельно инициировать каждый шаг обучения через единичный вызов `TrainStep`.

Ключевым преимуществом реализации является её высокая скорость выполнения одной эпохи, обусловленная малой размерностью решаемых в системе задач. Это позволяет в реальном времени изменять используемые данные и

архитектуру модели (количество слоёв, нейронов), не прерывая процесс обучения. Следующая же итерация цикла обучения автоматически использует уже обновлённую конфигурацию, обеспечивая немедленную реакцию системы на модификации.

Использование цикла анимации создаёт эффект параллельного выполнения вычислений, сохраняя при этом отзывчивость интерфейса для интерактивных операций, что удовлетворяет требованию динамической модификации параметров.

5.3.5 Верификация корректности: модульные тесты и сравнение с PyTorch

Для всесторонней проверки корректности реализации оптимизированных алгоритмов была разработана комплексная система тестирования, включающая модульное тестирование и сравнительный анализ с эталонной реализацией на PyTorch.

Модульное тестирование оптимизированных версий

Для каждого оптимизированного метода (с развёрнутыми циклами) проводилось сравнение с его базовой версией. На случайно сгенерированных входных данных выполнялись обе реализации, после чего вычислялась максимальная абсолютная разность результатов. Во всех случаях различие не превышало 1×10^{-16}, что соответствует машинной точности операций с плавающей запятой двойной точности и подтверждает идентичность логики оптимизированных и базовых алгоритмов.

Сравнительное тестирование с PyTorch

Для удовлетворения требованию численной корректности проведено сравнительное тестирование всех компонентов машинного обучения с эталонной реализацией на PyTorch (версия 1.16). Тестирование выполнялось на детерминированных наборах данных, обеспечивающих воспроизводимость результатов:

- **Функции потерь**: протестированы MSE, MAE, Huber и LogCosh loss на детерминированных парах «предсказание–цель», с проверкой как значений функций, так и их градиентов.
- **Оптимизаторы**: для каждого оптимизатора (SGD, MomentumSGD, Adam, Adamax, Adadelta, Adagrad, RMSprop) проверялись значения вычисленных градиентов после каждого шага оптимизации функции Huber в течение 50 итераций. Сравнение проводилось с соответствующими оптимизаторами PyTorch с идентичными гиперпараметрами (регуляризация, скорость сходимости и специфические для оптимизаторов).
- **Полносвязная сеть**: проведено комплексное тестирование полного цикла обучения для сетей различной глубины – двухслойной (3-5-1), трёхслойной (3-7-9-1) и пятислойной (3-6-12-24-1). Каждая сеть инициализировалась заранее определёнными весами, обучение выполнялось на фиксированном наборе из 12 элементов (размер пакета 4) в течение 100 эпох. Контрольные значения функции потерь, выходов модели и весовых коэффициентов сравнивались с PyTorch после 1-й, 10-й, 42-й и 100-й эпохами.

Во всех тестах максимальная разность между результатами реализованной системы и PyTorch не превышала 1×10^{-15}. Такой уровень погрешности соответствует машинной точности операций с числами двойной точности и подтверждает корректность реализации всех численных алгоритмов, включая вычисление градиентов и обновление весов.

Разработанный набор детерминированных тестов обеспечивает полное покрытие модуля машинного обучения и позволяет проводить регрессионное тестирование при дальнейшем развитии системы.

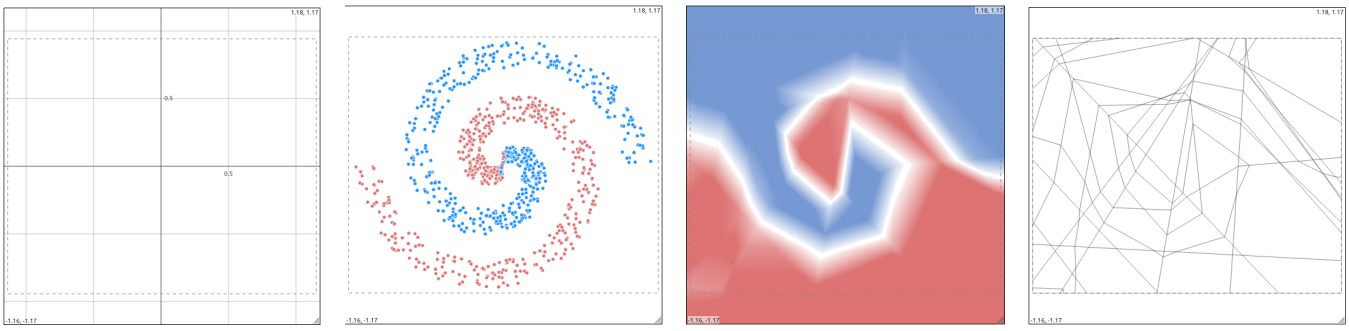
5.4 Система визуализации и интерактивности

Для удовлетворения требования интерактивной визуализации были созданы независимые компоненты, использующие механизм событий для дифференцированного обновления интерфейса, позволяющего перерисовывать только необходимые части графического интерфейса.

5.4.1 Архитектура подсистемы визуализации

Визуализация построена на принципе многослойного рендеринга, где каждый слой отвечает за отрисовку определенного типа информации:

- Слой сетки (`GridLayer`) – координатная сетка и оси (рисунок 5.5 а)
- Слой данных (`DataLayer`) – отрисовка точек наборов данных через SVG с цветовой кодировкой классов(рисунок 5.5 б);
- Слой модели (`ModelOutputLayer`) – визуализация выхода нейросети через HTML5 Canvas(рисунок 5.5 в);
- Слой ячеек (`CellsPlot`) – границы иерархического разбиения компакта персептроном на ячейки(рисунок 5.5 г).



а) Слой сетки б) Слой данных в) Слой модели г) Слой ячеек

Рисунок 5.5 — Визуализация слоёв отрисовки

Все слои синхронизированы через общий объект `ViewBox`, обеспечивающий преобразование координат и обработку масштабирования/перемещения на координатной плоскости.

5.4.2 Алгоритмы отрисовки многомерных данных

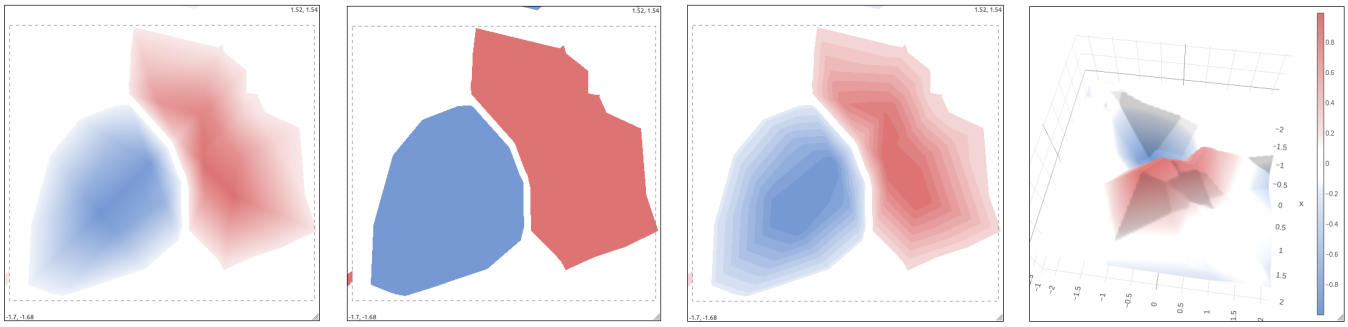
Для визуализации данных произвольной размерности $d > 2$ реализован механизм проекций на выбранные координаты. Пользователь выбирает оси x_i и x_j для отображения, остальные координаты $x_k, k \neq i,j$ фиксируются текущим значением точки в пространстве, которая задаётся пользователем.

Это позволяет исследовать поведение модели в произвольных двумерных сечениях многомерного пространства.

5.4.3 Визуализация структуры нейросети и её выхода

Для отображения выхода модели реализованы несколько режимов (рисунок 5.6):

– Линейный – плавный градиент от синего (-1) через белый (0) к красному (+1);
– Дискретный (2, 4, 10 уровней) – квантизация выхода для анализа пороговых эффектов;
– 3D – отображение в виде 3D поверхности.



а) Линейный режим | б) Дискретный режим | в) Дискретный режим на 10 уровней | г) 3D режим

Рисунок 5.6 — Визуализация выхода модели

Визуализация архитектуры сети (ModelArchitectureLayer) отображает (рисунок 5.7):

– Веса связей (цветом и толщиной линии);

– Исследуемый нейрон (использует зелёную заливку в отличие от остальных серых).
– Отключенные нейроны (используют чёрную заливку).

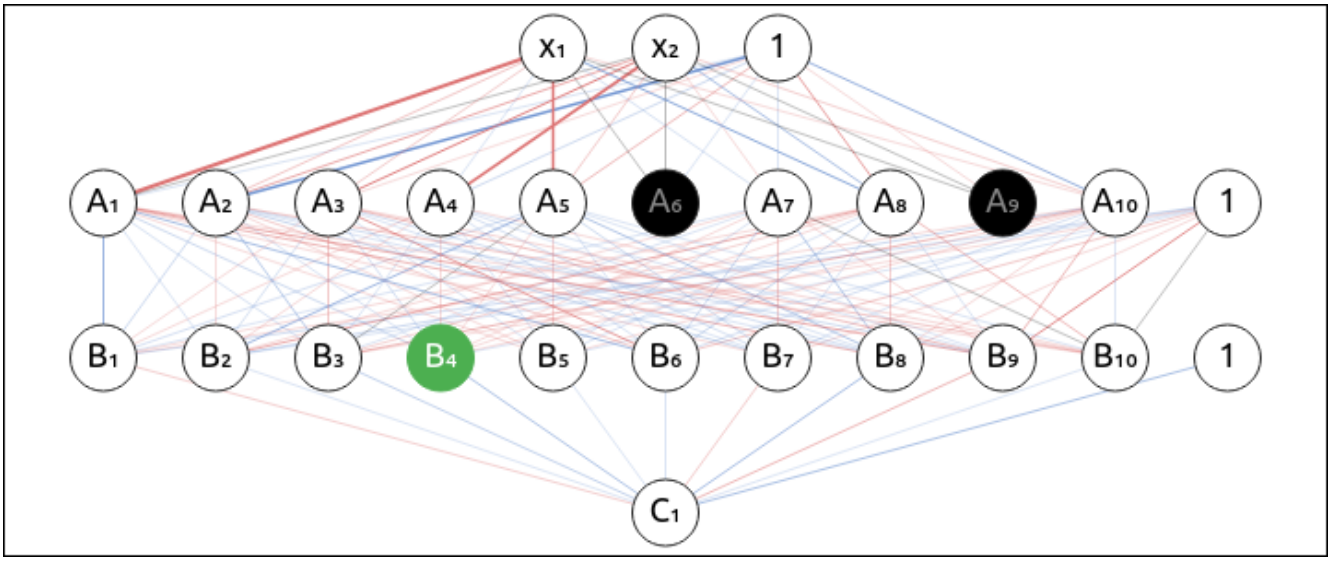


Рисунок 5.7 — Визуализация архитектуры нейросети (исследуется нейрон B_4, нейроны A_6 и A_9 отключены

5.5 Анализ производительности и системные характеристики

5.5.1 Сравнение производительности оптимизированных и базовых версий

Для количественной оценки эффективности применённых оптимизаций проведено сравнительное тестирование производительности базовой и оптимизированной (с развёрнутыми циклами) версий критических компонентов системы. Тестирование выполнено для различных архитектур моделей и размеров пакетов данных, что позволяет оценить влияние оптимизаций в различных сценариях использования.

Методология тестирования

Для каждого теста предварительно выполнялось 100 итераций прогрева для стабилизации производительности JIT-компилятора JavaScript, затем 1000 итераций на заранее сгенерированных случайных данных для точного замера времени выполнения. Тестирование проводилось для различных размеров пакетов (4, 16, 32 элемента) и различных конфигураций слоёв и моделей. Тестирование производилось в браузере Google Chrome 136.

Тестирование полносвязного слоя

Протестированы операции прямого и обратного распространения для полносвязных слоёв различной размерности:

– Прямое распространение (`forward`);
– Обратное распространение без вычисления градиента входа (`backward`;
– Обратное распространение с вычислением градиента входа (`backward(dx)`).

Таблица 5 — Производительность полносвязного слоя

Параметры слоя	Операция	Ускорение оптимизированной версии		
		$bs = 4$	$bs = 16$	$bs = 32$
2×10	forward	1.74	1.35	1.37
	backward	2.34	1.25	1.96
	backward(dx)	1.66	1.63	1.20
10×10	forward	1.24	1.10	1.32
	backward	2.45	2.23	1.34
	backward(dx)	1.79	1.25	1.59
10×100	forward	1.99	1.52	1.83
	backward	1.48	2.85	1.85
	backward(dx)	1.86	1.95	1.76
100×10	forward	1.52	1.31	1.13
	backward	1.23	1.46	1.39
	backward(dx)	1.64	1.97	1.42

Конфигурации слоёв включали комбинации входных и выходных размерностей (2x10, 10x10, 10x100, 100x10). Результаты значений ускорения для различных размерностей и размеров пакетов представлены в таблице 5.

Тестирование полносвязной сети

Протестированы комплексные операции для полносвязных сетей различной архитектуры:

– Предсказание (`predict`);
– Прямое распространение по всей сети (`forward`);
– Обратное распространение (`backward`);
– Полный цикл обучения (`train_on_batch`).

Таблица 6 — Производительность полносвязной сети

Параметры сети	Операция	Ускорение оптимизированной версии		
		$bs = 4$	$bs = 16$	$bs = 32$
$2 - 10$	predict	1.09	2.87	1.27
	forward	1.12	2.23	1.74
	backward	1.36	1.52	4.07
	train on batch	2.10	0.74	1.54
$10 - 10 - 1$	predict	1.77	1.42	1.03
	forward	1.05	1.02	1.08
	backward	1.41	1.94	1.23
	train on batch	1.01	1.46	1.32
$25 - 25 - 25 - 25 - 1$	predict	1.51	1.23	2.03
	forward	1.36	1.56	1.05
	backward	2.53	2.07	1.82
	train on batch	1.38	1.69	1.75

Архитектуры сетей включали 1-5 слоёв с различным количеством нейронов. Результаты значений ускорения представлены в таблице 6.

Анализ результатов

Анализ результатов показывает, что применение развёртывания циклов позволяет достичь ускорения до 2.8 раз в зависимости от конкретной операции, архитектуры модели и размера пакета. Наибольший выигрыш в производительности наблюдается для операций обратного распространения с вычислением градиентов – наиболее вычислительно интенсивных частей алгоритма обучения. Для операций прямого распространения ускорение составляет 1.1–2 раза, что также существенно для интерактивной работы системы.

5.5.2 Кроссплатформенность

Разработанная система реализована как веб-приложение, что обеспечивает полную кроссплатформенность без необходимости установки дополнительного программного обеспечения. Для работы достаточно любого современного браузера (Chrome 80+, Firefox 75+, Safari 14+, Edge 80+) с поддержкой JavaScript ES2020, HTML5 [89], Canvas [90] и SVG [91]. Это позволяет системе функционировать идентично на всех основных операционных системах (Windows, Linux, macOS) и архитектурах процессоров (x86-64, ARM), а также на мобильных платформах (iOS, Android).

Для адаптации к мобильным устройствам реализованы адаптивный CSS3-интерфейс [92], обработка жестов (масштабирование, перемещение) и отложенный рендеринг ресурсоёмких графических компонентов. Использование стандартизированных браузерных API гарантирует согласованное поведение системы независимо от платформы, обеспечивая доступность инструмента визуализации и исследования персептронов на любом устройстве с веб-браузером.

5.5.3 Масштабируемость и практические ограничения

Реализованная система спроектирована таким образом, что формальных ограничений на размерность данных, количество нейронов или объём обучающей выборки не существует – они определяются исключительно доступными ресурсами оперативной памяти и вычислительными возможностями целевой платформы. Однако для типичного компьютера с 16 ГБ оперативной памяти и современным процессором (например, Intel Core i7 или аналог) определены следующие эмпирические границы комфортного использования системы:

- **Размер модели**: система эффективно работает с полносвязными сетями, содержащими до 200 нейронов суммарно по всем слоям. При этом сохраняется возможность интерактивного изменения архитектуры, визуализации весов и отрисовки выходной поверхности в реальном времени.
- **Объём данных**: в 20-мерном пространстве система позволяет загружать и визуализировать до 10^6 точек (обучающих и тестовых) без существенных задержек при масштабировании и перемещении. Основным узким местом становится не вычисление предсказаний (оптимизированное развёрткой циклов), а отрисовка большого числа SVG-элементов.
- **Параметры обучения**: максимальный размер пакета (batch size) зафиксирован на уровне $2^7 = 128$. Это значение выбрано исходя из компромисса между эффективностью градиентного спуска и объёмом заранее выделенной памяти под промежуточные буферы (активации и градиенты). Увеличение этого параметра линейно повышает потребление памяти, но не всегда приводит к ускорению обучения из-за роста времени обновления весов.

5.6 Примеры использования

Интеллектуальная система машинного обучения разработана с целью поддержки полного цикла исследования поведения нейросетевых моделей, включая

этапы создания данных, обучения классификатора, анализа и интерпретации результатов. Ниже приведены ключевые сценарии использования системы, иллюстрирующие её функциональные возможности.

5.6.1 Бинарная классификация

Для проверки способности модели к нелинейной аппроксимации границ принятия решений решается задача классификации двух переплетённых спиралей (рисунок 5.8). В системе предусмотрена генерация соответствующего набора данных и обучение модели с возможностью пошагового отображения изменения решения по мере выполнения эпохи градиентного спуска. Интеллектуальная система машинного обучения позволяет наблюдать как локальные ошибки, так и итоговую зону классификации, что особенно полезно при выборе архитектуры сети и прочих гиперпараметров.

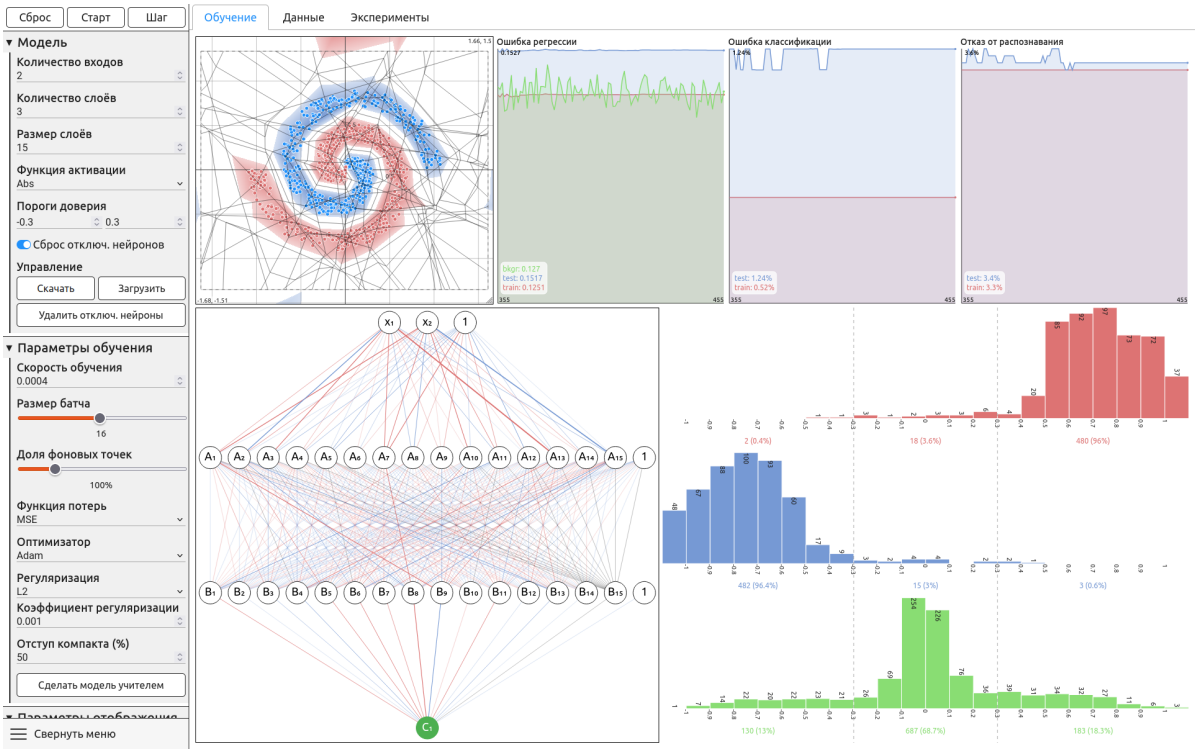


Рисунок 5.8 — Пример выполнения бинарной классификации

5.6.2 Унарная классификация

Режим унарной классификации допускает обучение модели только по положительным примерам (рисунок 5.9). В качестве примера используется одна из спиралей из предыдущего эксперимента. Пользователь может задать уровень порога β, визуализировать полученную область принятия положительного класса, а также проследить, каким образом меняется зона отказа при варьировании параметров. Данный сценарий позволяет исследовать свойство доверия, характерное для унарных моделей.

Рисунок 5.9 — Пример выполнения унарной классификации

5.6.3 Создание синтетических данных

Одной из оригинальных функций системы является реализация метода синтетической генерации данных на основе предложенного в рамках диссертационного исследования метода, используя предварительно обученную унарную модель (рисунок 5.10).

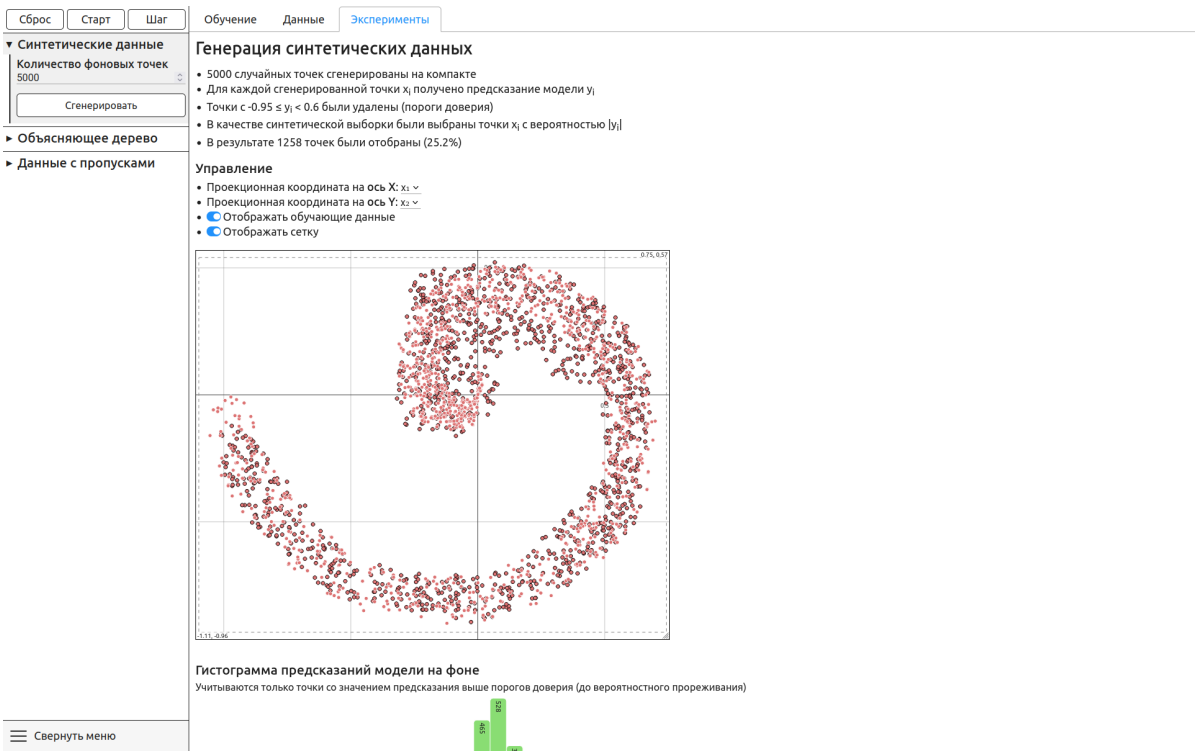


Рисунок 5.10 — Пример построения синтетических данных

Пользователь может интерактивно варьировать пороговое значение, наблюдать за статистическими характеристиками (средним, минимумом, максимумом, среднеквадратичным отклонением и ковариационной матрицей) отобранных объектов, а также визуализировать геометрию полученного множества в виде 2D проекций.

5.6.4 Построение объясняющего дерева решений

Одним из компонентов системы является модуль построения объясняющего дерева решений, предназначенного для статистической интерпретации решения обученного персептрона (рисунок 5.11). Пользователь может выбрать интересующую ячейку и подробно изучить как её содержимое, так и геометрию пространства. Это позволяет проводить интерпретацию решения в выбранной области и служит средством повышения доверия к результатам классификации.

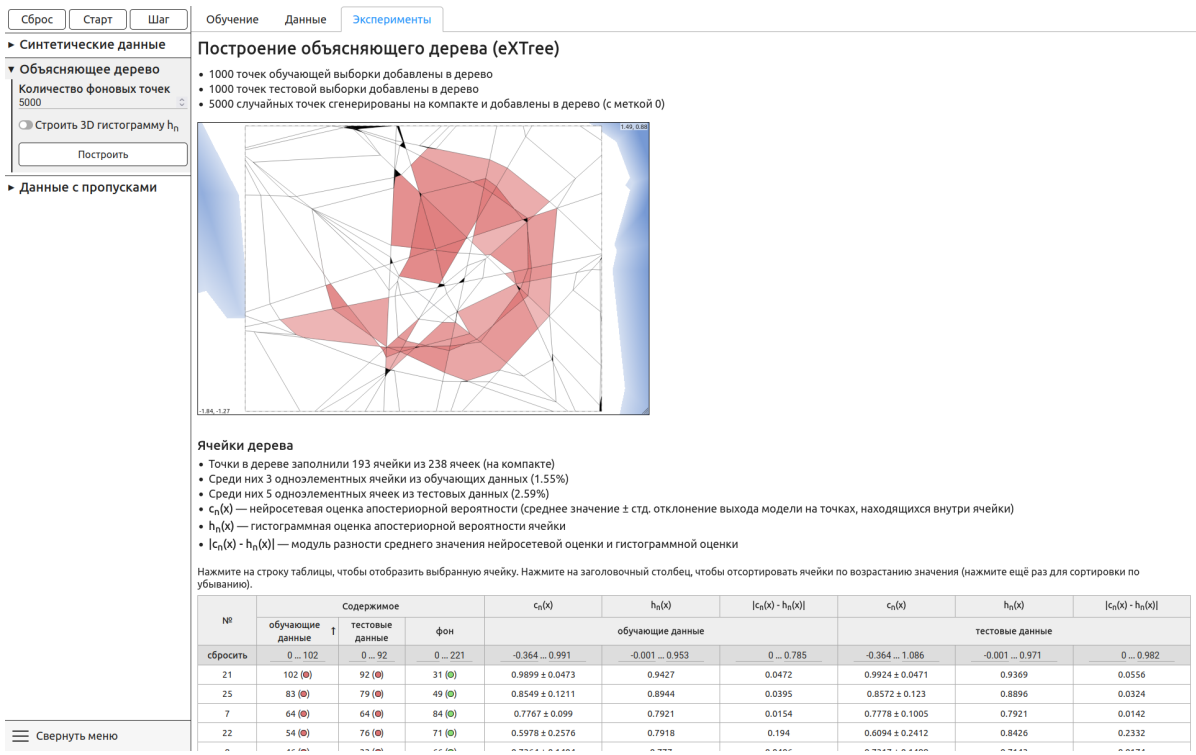


Рисунок 5.11 — Пример работы с объясняющим деревом

5.7 Выводы

Разработанная в рамках диссертационного исследования интеллектуальная система машинного обучения представляет собой законченное программное решение, которое полностью удовлетворяет всем сформулированным требованиям.

Система реализует разработанные методы – модифицированную бинарную классификацию, унарную классификацию, генерацию синтетических данных на её основе и интерпретирующую модель eXBTree – и предоставляет инструментарий для их экспериментального исследования.

Обеспечена автономность и кроссплатформенность: система работает как автономное веб-приложение, не требующее установки дополнительного программного обеспечения, специальных драйверов, графического процессора или доступа к сети. Это гарантирует её работоспособность и воспроизводимость результатов на любом устройстве, включая компьютеры с ограниченными ресурсами.

Важной особенностью является интерактивная визуализация в реальном времени. Система отображает используемые данные, аппроксимацию носителя распределения, архитектуру модели, выходные активации нейронов, формируе-

мое разбиение пространства решений и динамику метрик обучения, обеспечивая целостное восприятие процесса без артефактов производительности.

Ключевым достижением стала поддержка динамической модификации параметров. Пользователь может в реальном времени изменять параметры данных (дисбаланс классов, уровень шума) и архитектуру сети (количество слоёв, нейронов, функции активации) без прерывания процесса обучения. Любое изменение немедленно отражается на всех визуализациях, обеспечивая прямую связь между конфигурацией модели и её поведением.

Обеспечена численная корректность вычислительного ядра. Операции прямого и обратного распространения, а также расчёт градиентов функционально эквивалентны эталонной реализации на PyTorch в пределах машинной точности, что подтверждается разработанным пакетом модульных тестов.

Для достижения высокой производительности в условиях однопоточного JavaScript применена оптимизация разворачивания циклов, что за счёт параллелизма на уровне инструкций (ILP) обеспечило ускорение вычислений на CPU в 1.5–2.5 раза по сравнению с базовой реализацией. Система опубликована как проект с открытым исходным кодом на платформе GitHub, что способствует верификации, повторному использованию и дальнейшему развитию.

Таким образом, система успешно интегрирует предложенные методы в единую интерактивную исследовательскую платформу. Она служит инструментом не только для экспериментального подтверждения теоретических результатов диссертации, но и для интерактивного анализа, визуализации и отладки доверенных персептронных моделей в средах с ограниченными вычислительными ресурсами.

Заключение

В представленной работе решена задача построения статистически обоснованного классификатора для пространств малой размерности в рамках парадигмы доверенного искусственного интеллекта. Исследование направлено на преодоление разрыва между эмпирическими нейросетевыми методами и формальным аппаратом математической статистики.

Основные результаты работы заключаются в следующем:

1. Разработана теоретическая база непараметрического оценивания в условиях дисбаланса классов и малой размерности. Сформулированы и доказаны теоремы, обосновывающие асимптотическую связь между нейросетевой и гистограммной оценками апостериорной вероятности. Данный результат обеспечивает формальное статистическое обоснование для предложенного подхода.
2. Разработан метод построения статистически обоснованного объяснимого байесовского классификатора на основе многослойного персептрона и дерева решений. Метод обеспечивает оценку апостериорных вероятностей с теоретическими гарантиями и формирование интерпретируемых правил классификации.
3. Разработан метод построения унарного классификатора, устойчивого к дисбалансу классов и позволяющего генерировать синтетические данные, сохраняющие геометрические и статистические свойства исходного распределения.
4. Создана интеллектуальная система машинного обучения, реализующая предложенные методы и обеспечивающая решение задач классификации данных малой размерности в условиях дисбаланса классов и высокой неопределённости вне носителя распределения.

Результаты экспериментального исследования подтвердили устойчивость классификатора к дисбалансу классов, корректность работы в условиях выхода за носитель распределения и адекватность генерируемых синтетических данных.

В работе предложен формально обоснованный подход к созданию доверенных классификаторов, сочетающий выразительность нейросетевых моделей со строгостью статистических методов.

Список литературы

1. *Кобринский, Б. А.* Доверие к технологиям искусственного интеллекта [Текст] / Б. А. Кобринский // Искусственный интеллект и принятие решений. — 2024. — № 3. — С. 3—17.
2. *Perminov, A.* SLAP—Simple Linear Attack against Perceptron (SLAP) [Текст] / A. Perminov // Programming and Computer Software. — 2025. — Т. 51, № 6. — С. 446—452.
3. Extrapolation of the Bayesian classifier with an unknown support of the two-class mixture distribution [Текст] / K. S. Lukianov [и др.] // Uspekhi Matematicheskikh Nauk. — 2024. — Т. 79, № 6. — С. 57—82.
4. *Eliseev, N. A.* Convergence of a multilayer perceptron to histogram Bayesian regression [Текст] / N. A. Eliseev, A. I. Perminov, D. Y. Turdakov // Uspekhi Matematicheskikh Nauk. — 2025. — Т. 80, № 6. — С. 45—72.
5. *Perminov, A.* CONSISTENT METHOD FOR SYNTHETIC TABULAR DATA OBTAINING USING A MULTILAYER PERCEPTRON [Текст] / A. Perminov, A. Kovalenko, D. Turdakov // Journal of Mathematical Sciences. — 2026. — С. 1—12.
6. *Belyaeva, O. V.* Synthetic data usage for document segmentation models fine-tuning [Текст] / O. V. Belyaeva, A. I. Perminov, I. S. Kozlov // Proceedings of the Institute for System Programming of the RAS (Proceedings of ISP RAS). — 2020. — Т. 32, № 4. — С. 189—202.
7. *Perminov, A. I.* Method for training perceptron on tabular data with missing values [Текст] / A. I. Perminov, A. P. Kovalenko, D. Y. Turdakov // Proceedings of the Institute for System Programming of the RAS. — 2025. — Т. 37, № 62. — С. 93—106.
8. *Воронцов, К.* Математические методы обучения по прецедентам (теория обучения машин) [Текст] / К. Воронцов // Москва. — 2011. — С. 119—121.
9. *Devroye, L.* A probabilistic theory of pattern recognition [Текст]. Т. 31 / L. Devroye, L. Györfi, G. Lugosi. — Springer Science & Business Media, 2013.
10. *Cortes, C.* Support-vector networks [Текст] / C. Cortes, V. Vapnik // Machine learning. — 1995. — Т. 20, № 3. — С. 273—297.

11. Classification and regression trees. Wadsworth Int [Текст] / L. Breiman [и др.] // Group. — 1984. — Т. 37, № 15. — С. 237—251.

12. *Breiman, L.* Random forests [Текст] / L. Breiman // Machine learning. — 2001. — Т. 45, № 1. — С. 5—32.

13. *Friedman, J. H.* Greedy function approximation: a gradient boosting machine [Текст] / J. H. Friedman // Annals of statistics. — 2001. — С. 1189—1232.

14. *Goodfellow, I.* Deep feedforward networks [Текст] / I. Goodfellow, Y. Bengio, A. Courville // Deep learning. — 2016. — Т. 1. — С. 161—217.

15. *Yarotsky, D.* Optimal approximation of continuous functions by very deep ReLU networks [Текст] / D. Yarotsky // Conference on learning theory. — PMLR. 2018. — С. 639—649.

16. *Kendall, A.* What uncertainties do we need in bayesian deep learning for computer vision? [Текст] / A. Kendall, Y. Gal // Advances in neural information processing systems. — 2017. — Т. 30.

17. Uncertainty in deep learning [Текст] / Y. Gal [и др.]. — 2016.

18. *Der Kiureghian, A.* Aleatory or epistemic? Does it matter? [Текст] / A. Der Kiureghian, O. Ditlevsen // Structural safety. — 2009. — Т. 31, № 2. — С. 105—112.

19. A review of uncertainty quantification in deep learning: Techniques, applications and challenges [Текст] / M. Abdar [и др.] // Information fusion. — 2021. — Т. 76. — С. 243—297.

20. *Hendrycks, D.* A baseline for detecting misclassified and out-of-distribution examples in neural networks [Текст] / D. Hendrycks, K. Gimpel // arXiv preprint arXiv:1610.02136. — 2016.

21. A simple unified framework for detecting out-of-distribution samples and adversarial attacks [Текст] / K. Lee [и др.] // Advances in neural information processing systems. — 2018. — Т. 31.

22. *Hendrycks, D.* Deep anomaly detection with outlier exposure [Текст] / D. Hendrycks, M. Mazeika, T. Dietterich // arXiv preprint arXiv:1812.04606. — 2018.

23. *He, H.* Learning from imbalanced data [Текст] / H. He, E. A. Garcia // IEEE Transactions on knowledge and data engineering. — 2009. — Т. 21, № 9. — С. 1263—1284.

24. SMOTE: synthetic minority over-sampling technique [Текст] / N. V. Chawla [и др.] // Journal of artificial intelligence research. — 2002. — Т. 16. — С. 321—357.

25. *Воронцов, К.* Лекции по статистическим (байесовским) алгоритмам классификации [Текст] / К. Воронцов // URL: http://www. ccas. ru/voron/download/Bayes. pdf (20.09. 2017). — 2008.

26. *Obi, J. C.* A review of techniques for regularization [Текст] / J. C. Obi, I. C. Jecinta // International Journal of Research in Engineering and Science. — 2023. — Т. 11, № 1. — С. 360—367.

27. *Muhamedyev, R.* Machine learning methods: An overview [Текст] / R. Muhamedyev // Computer modelling & new technologies. — 2015. — Т. 19, № 6. — С. 14—29.

28. *Boukerche, A.* Outlier detection: Methods, models, and classification [Текст] / A. Boukerche, L. Zheng, O. Alfandi // ACM Computing Surveys (CSUR). — 2020. — Т. 53, № 3. — С. 1—37.

29. *Ding, X.* Hyperparameter sensitivity in deep outlier detection: Analysis and a scalable hyper-ensemble solution [Текст] / X. Ding, L. Zhao, L. Akoglu // Advances in Neural Information Processing Systems. — 2022. — Т. 35. — С. 9603—9616.

30. Generalized out-of-distribution detection: A survey [Текст] / J. Yang [и др.] // International Journal of Computer Vision. — 2024. — Т. 132, № 12. — С. 5635—5662.

31. *Caron, A.* A view on out-of-distribution identification from a statistical testing theory perspective [Текст] / A. Caron, C. Hicks, V. Mavroudis // arXiv preprint arXiv:2405.03052. — 2024.

32. BOOST: Out-of-Distribution-Informed Adaptive Sampling for Bias Mitigation in Stylistic Convolutional Neural Networks [Текст] / M. Vijendran [и др.] // Expert Systems with Applications. — 2025. — С. 128905.

33. *Shmuel, A.* Machine and deep learning performance in out-of-distribution regressions [Текст] / A. Shmuel, O. Glickman, T. Lazebnik // Machine Learning: Science and Technology. — 2025. — Т. 5, № 4. — С. 045078.
34. *Коваленко, А. П.* Подход к решению «проблемы экстраполяции» нейросетевого классификатора [Текст] / А. П. Коваленко, А. И. Перминов // Материалы 32-й научно-технической конференции «Методы и технические средства обеспечения безопасности информации». — 2023.
35. *Коваленко, А. П.* Доверять... или не доверять? Лемма об экстраполяции байесовского классификатора [Текст] / А. П. Коваленко, А. И. Перминов, П. А. Яськов // Материалы 33-й научно-технической конференции «Методы и технические средства обеспечения безопасности информации». — 2024.
36. *Devroye, L.* Nonparametric density estimation [Текст] / L. Devroye // The L_1 View. — 1985.
37. The elements of statistical learning [Текст] / T. Hastie, R. Tibshirani, J. Friedman [и др.]. — 2009.
38. *Devroye, L.* The Regular Histogram Rule [Текст] / L. Devroye, L. Györfi, G. Lugosi // A Probabilistic Theory of Pattern Recognition. — Springer, 1996. — С. 133—145.
39. *Devroye, L.* Consistency of the k-nearest neighbor rule [Текст] / L. Devroye, L. Györfi, G. Lugosi // A Probabilistic Theory of Pattern Recognition. — Springer, 1996. — С. 169—185.
40. *Wand, M. P.* Kernel smoothing [Текст] / M. P. Wand, M. C. Jones. — CRC press, 1994.
41. A distribution-free theory of nonparametric regression [Текст] / L. Györfi [и др.]. — Springer, 2002.
42. *Steinwart, I.* Support vector machines [Текст] / I. Steinwart, A. Christmann. — Springer Science & Business Media, 2008.
43. *Murtagh, F.* Multilayer perceptrons for classification and regression [Текст] / F. Murtagh // Neurocomputing. — 1991. — Т. 2, № 5/6. — С. 183—197.

44. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function [Текст] / M. Leshno [и др.] // Neural Networks. — 1993. — Т. 6, № 6. — С. 861—867. — URL: https://www.sciencedirect.com/science/article/pii/S0893608005801315.

45. *Amari, S.-i.* Backpropagation and stochastic gradient descent method [Текст] / S.-i. Amari // Neurocomputing. — 1993. — Т. 5, № 4/5. — С. 185—196.

46. *Seiffert, U.* Multiple layer perceptron training using genetic algorithms. [Текст] / U. Seiffert // ESANN. — 2001. — С. 159—164.

47. *Song, Y.-Y.* Decision tree methods: applications for classification and prediction [Текст] / Y.-Y. Song, Y. Lu // Shanghai archives of psychiatry. — 2015.

48. On the number of linear regions of deep neural networks [Текст] / G. Montúfar [и др.] // Advances in neural information processing systems. — 2014. — Т. 27.

49. *Cochran, W. G.* Some methods for strengthening the common χ 2 tests [Текст] / W. G. Cochran // Biometrics. — 1954. — Т. 10, № 4. — С. 417—451.

50. *Девяткин, Д.* Система распределенного построения случайных лесов деревьев решений с линейными и нелинейными разделителями [Текст] / Д. Девяткин // Системы высокой доступности. — 2022. — Т. 3, № 18. — С. 59—68.

51. *Salih, A. M.* Are Linear Regression Models White Box and Interpretable? [Текст] / A. M. Salih, Y. Wang // arXiv preprint arXiv:2407.12177. — 2024.

52. *Aly, M.* Survey on multiclass classification methods [Текст] / M. Aly // Neural Netw. — 2005. — Т. 19, № 1—9. — С. 2.

53. *Lorena, A. C.* A review on the combination of binary classifiers in multiclass problems [Текст] / A. C. Lorena, A. C. De Carvalho, J. M. Gama // Artificial Intelligence Review. — 2008. — Т. 30, № 1. — С. 19.

54. An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes [Текст] / M. Galar [и др.] // Pattern Recognition. — 2011. — Т. 44, № 8. — С. 1761—1776.

55. *Kang, S.* Constructing a multi-class classifier using one-against-one approach with different binary classifiers [Текст] / S. Kang, S. Cho, P. Kang // Neurocomputing. — 2015. — Т. 149. — С. 677—682.

56. *Guo, W.* An overview of backdoor attacks against deep neural networks and possible defences [Текст] / W. Guo, B. Tondi, M. Barni // IEEE Open Journal of Signal Processing. — 2022. — Т. 3. — С. 261—287.

57. *Goodfellow, I. J.* Explaining and harnessing adversarial examples [Текст] / I. J. Goodfellow, J. Shlens, C. Szegedy // arXiv preprint arXiv:1412.6572. — 2014.

58. Towards deep learning models resistant to adversarial attacks [Текст] / A. Madry [и др.] // arXiv preprint arXiv:1706.06083. — 2017.

59. *Croce, F.* Sparse and imperceivable adversarial attacks [Текст] / F. Croce, M. Hein // Proceedings of the IEEE/CVF international conference on computer vision. — 2019. — С. 4724—4732.

60. *Wong, E.* Provable defenses against adversarial examples via the convex outer adversarial polytope [Текст] / E. Wong, Z. Kolter // International conference on machine learning. — PMLR. 2018. — С. 5286—5295.

61. Cats and dogs [Текст] / O. M. Parkhi [и др.] // 2012 IEEE conference on computer vision and pattern recognition. — IEEE. 2012. — С. 3498—3505.

62. *Deng, L.* The mnist database of handwritten digit images for machine learning research [best of the web] [Текст] / L. Deng // IEEE signal processing magazine. — 2012. — Т. 29, № 6. — С. 141—142.

63. Learning multiple layers of features from tiny images.(2009) [Текст] / A. Krizhevsky, G. Hinton [и др.]. — 2009.

64. *Obi, J. C.* A comparative study of several classification metrics and their performances on data [Текст] / J. C. Obi // World Journal of Advanced Engineering Technology and Sciences. — 2023. — Т. 8, № 1. — С. 308—314.

65. *Dua, D.* UCI Machine Learning Repository [Текст] / D. Dua, C. Graff. — 2017. — URL: http://archive.ics.uci.edu/ml.

66. *Little, R. J.* Statistical analysis with missing data. [Текст] / R. J. Little, D. B. Rubin. — 1995.

67. K-nearest neighbor (k-NN) based missing data imputation [Текст] / U. Pujianto, A. P. Wibawa, M. I. Akbar [и др.] // 2019 5th International Conference on Science in Information Technology (ICSITech). — IEEE. 2019. — С. 83—88.

68. *Schafer, J. L.* Analysis of incomplete multivariate data [Текст] / J. L. Schafer. — CRC press, 1997.

69. Reconstructing training data from trained neural networks [Текст] / N. Haim [и др.] // Advances in Neural Information Processing Systems. — 2022. — Т. 35. — С. 22911—22924.

70. *Gal, M. S.* Synthetic data: legal implications of the data-generation revolution [Текст] / M. S. Gal, O. Lynskey // IoWa L. rev. — 2023. — Т. 109. — С. 1087.

71. Synthetic data in human analysis: A survey [Текст] / I. Joshi [и др.] // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2024. — Т. 46, № 7. — С. 4957—4976.

72. *Grund, S.* Using synthetic data to improve the reproducibility of statistical results in psychological research. [Текст] / S. Grund, O. Lüdtke, A. Robitzsch // Psychological Methods. — 2022.

73. Comprehensive exploration of synthetic data generation: A survey [Текст] / A. Bauer [и др.] // arXiv preprint arXiv:2401.02524. — 2024.

74. *Figueira, A.* Survey on synthetic data generation, evaluation methods and GANs [Текст] / A. Figueira, B. Vaz // Mathematics. — 2022. — Т. 10, № 15. — С. 2733.

75. *Wan, Z.* Variational autoencoder based synthetic data generation for imbalanced learning [Текст] / Z. Wan, Y. Zhang, H. He // 2017 IEEE symposium series on computational intelligence (SSCI). — IEEE. 2017. — С. 1—7.

76. *Jordon, J.* PATE-GAN: Generating synthetic data with differential privacy guarantees [Текст] / J. Jordon, J. Yoon, M. Van Der Schaar // International conference on learning representations. — 2018.

77. Diffusion models for tabular data imputation and synthetic data generation [Текст] / M. Villaizán-Vallelado [и др.] // ACM Transactions on Knowledge Discovery from Data. — 2024.

78. *Akkem, Y.* A comprehensive review of synthetic data generation in smart farming by using variational autoencoder and generative adversarial network [Текст] / Y. Akkem, S. K. Biswas, A. Varanasi // Engineering Applications of Artificial Intelligence. — 2024. — Т. 131. — С. 107881.

79. *Chen, T.* XGBoost: A Scalable Tree Boosting System [Текст] / T. Chen // Cornell University. — 2016.

80. Synthetic Data Metrics [Текст] / DataCebo, Inc. — 10.2023. — URL: https://docs.sdv.dev/sdmetrics/ ; Version 0.12.0.

81. Modeling tabular data using conditional gan [Текст] / L. Xu [и др.] // Advances in neural information processing systems. — 2019. — Т. 32.

82. *Флэнаган, Д.* JavaScript [Текст] / Д. Флэнаган // Подробное руководство, Изд-во «Символ-Плюс. — 2013.

83. *Meyer, G. P.* An alternative probabilistic interpretation of the huber loss [Текст] / G. P. Meyer // Proceedings of the ieee/cvf conference on computer vision and pattern recognition. — 2021. — С. 5261—5269.

84. *Saleh, R. A.* Statistical properties of the log-cosh loss function used in machine learning [Текст] / R. A. Saleh, A. Saleh // arXiv preprint arXiv:2208.04564. — 2022.

85. *Ruder, S.* An overview of gradient descent optimization algorithms [Текст] / S. Ruder // arXiv preprint arXiv:1609.04747. — 2016.

86. *Zeiler, M. D.* Adadelta: an adaptive learning rate method [Текст] / M. D. Zeiler // arXiv preprint arXiv:1212.5701. — 2012.

87. *Matsakis, N. D.* Typed objects in javascript [Текст] / N. D. Matsakis, D. Herman, D. Lomov // ACM SIGPLAN Notices. — 2014. — Т. 50, № 2. — С. 125—134.

88. *Huang, J.-C.* Generalized loop-unrolling: a method for program speedup [Текст] / J.-C. Huang, T. Leng // Proceedings 1999 IEEE Symposium on Application-Specific Systems and Software Engineering and Technology. ASSET'99 (Cat. No. PR00122). — IEEE. 1999. — С. 244—248.

89. *Hickson, I.* Html5 [Текст] / I. Hickson, D. Hyatt // W3C working draft WD-Html5-20110525. — 2011. — Т. 53.

90. *Lubbers, P.* Using the html5 canvas api [Текст] / P. Lubbers, B. Albers, F. Salim // Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development. — Springer, 2010. — С. 25—63.

91. *Quint, A.* Scalable vector graphics [Текст] / A. Quint // IEEE MultiMedia. — 2003. — Т. 10, № 3. — С. 99—102.

92. *Lunn, I.* CSS3 foundations [Текст] / I. Lunn. — John Wiley & Sons, 2012.

93. *Imaizumi, M.* Deep neural networks learn non-smooth functions effectively [Текст] / M. Imaizumi, K. Fukumizu // Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics. — PMLR, 2019. — С. 869—878.

94. Phase retrieval from the magnitudes of affine linear measurements [Текст] / B. Gao [и др.] // Advances in Applied Mathematics. — 2018. — Т. 93. — С. 121—141.

95. *Guo, F.* On continuous selections of polynomial functions [Текст] / F. Guo, L. Jiao, D. Sang Kim // Optimization. — 2024. — Т. 73, № 2. — С. 295—328.

96. *Scholtes, S.* Piecewise affine functions [Текст] / S. Scholtes // Introduction to Piecewise Differentiable Equations. — New York, NY : Springer New York, 2012. — С. 13—63.

97. *Nobel, A.* Histogram regression estimation using data-dependent partitions [Текст] / A. Nobel // The Annals of Statistics. — 1996. — Т. 24, № 3. — С. 1084—1105.

98. *Goujon, A.* On the number of regions of piecewise linear neural networks [Текст] / A. Goujon, A. Etemadi, M. Unser // Journal of Computational and Applied Mathematics. — 2024. — Т. 441. — С. 115667.

99. *Csikós, M.* Tight lower bounds on the VC-dimension of geometric set systems [Текст] / M. Csikós, N. H. Mustafa, A. Kupavskii // Journal of Machine Learning Research. — 2019. — Т. 20, № 81. — С. 1—8.

100. *Plan, Y.* Dimension reduction by random hyperplane tessellations [Текст] / Y. Plan, R. Vershynin // Discrete & Computational Geometry. — 2014. — Т. 51, № 2. — С. 438—461.

101. *Mityagin, B.* The zero set of a real analytic function [Текст] / B. Mityagin // arXiv preprint arXiv:1512.07276. — 2015. — arXiv: 1512.07276.

102. *Giryes, R.* Deep neural networks with random Gaussian weights: A universal classification strategy? [Текст] / R. Giryes, G. Sapiro, A. M. Bronstein // IEEE Transactions on Signal Processing. — 2016. — Т. 64, № 13. — С. 3444—3457.

Список рисунков

Список таблиц

Приложение А

Свидетельства о государственной регистрации программ и ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

RU2023689161

ГОСУДАРСТВЕННАЯ РЕГИСТРАЦИЯ ПРОГРАММЫ ДЛЯ ЭВМ

Номер регистрации (свидетельства):
2023689161
Дата регистрации: **26.12.2023**
Номер и дата поступления заявки:
2023688363 15.12.2023
Дата публикации и номер бюллетеня:
26.12.2023 Бюл. № 1

Контактные реквизиты:
+7-903-700-79-86, m.kalugin@ispras.ru

Автор(ы):
Перминов Андрей Игоревич (RU),
Коваленко Андрей Петрович (RU),
Дробышевский Михаил Дмитриевич (RU),
Лукьянов Кирилл Сергеевич (RU)

Правообладатель(и):
Федеральное государственное бюджетное учреждение науки Институт системного программирования им. В.П. Иванникова Российской академии наук (RU)

Название программы для ЭВМ:
«DenseNetworkVisualizer: программное обеспечение для геометрической и вероятностной интерпретации и визуализации многослойного персептрона»

Реферат:
Программа предназначена для исследования работы многослойного персептрона и его геометрической и вероятностной интерпретаций. Может использоваться в качестве стенда для изучения особенностей работы многослойного персептрона. Является системой, реализующей основные возможности для управления многослойным персептроном и данными. Для этого предоставляется функциональность: управление параметрами модели; выбор данных; настройка параметров обучения; объясняющее дерево. Программа разработана ИСП РАН в рамках мероприятия «Разработка программного обеспечения, реализующего исследованные методы объяснения постфактум и методы встраивания интерпретируемости, в соответствии с разработанным ТЗ» Программы центра ИИ «Разработка методов и технологий создания систем доверенного искусственного интеллекта» по направлению доверенный искусственный интеллект. Тип ЭВМ: IBM PC-совмест. ПК; ОС: Linux, Windows, MacOS.

Язык программирования: JavaScript

Объем программы для ЭВМ: 4,4 МБ

РОССИЙСКАЯ ФЕДЕРАЦИЯ

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

RU2022682843

ГОСУДАРСТВЕННАЯ РЕГИСТРАЦИЯ ПРОГРАММЫ ДЛЯ ЭВМ

Номер регистрации (свидетельства):
2022682843
Дата регистрации: **28.11.2022**
Номер и дата поступления заявки:
2022681967 18.11.2022
Дата публикации и номер бюллетеня:
28.11.2022 Бюл. № 12

Контактные реквизиты:
+7-903-700-79-86, m.kalugin@ispras.ru

Автор(ы):
Булгакова Мария Ивановна (RU),
Гетьман Александр Игоревич (RU),
Горюнов Максим Николаевич (RU),
Мацкевич Андрей Георгиевич (RU),
Перминов Андрей Игоревич (RU),
Рыболовлев Дмитрий Александрович (RU)

Правообладатель(и):
Федеральное государственное бюджетное учреждение науки Институт системного программирования им. В.П. Иванникова Российской академии наук (RU)

Название программы для ЭВМ:
«Программа реализации атаки уклонения в отношении модели обнаружения вторжений»

Реферат:
Программа предназначена для реализации атаки уклонения в отношении модели машинного обучения, применяемой в системе обнаружения компьютерных атак. Поиск состязательных примеров ведётся при наличии знания о модели и обучающей выборке. Для каждой сетевой сессии тестовой выборки применяется метод перебора значений выбранного признака с проверкой сохранения метки «атака» у модифицированной сессии и изменения ответа модели. В рамках подхода учитывается невозможность прямого произвольного изменения значений отдельных признаков сессий сетевого трафика со стороны атакующего. Программа разработана ИСП РАН в рамках мероприятия «Методы обнаружения и противодействия атакам с внедрением закладок и зловредного кода в модели машинного обучения» Программы центра ИИ «Разработка методов и технологий создания систем доверенного искусственного интеллекта» по направлению доверенный искусственный интеллект. IBM-совместимые ПК Linux.

Язык программирования: Python (Jupyter Notebook)

Объем программы для ЭВМ: 39 КБ

РОССИЙСКАЯ ФЕДЕРАЦИЯ

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

RU2022685576

ГОСУДАРСТВЕННАЯ РЕГИСТРАЦИЯ ПРОГРАММЫ ДЛЯ ЭВМ

Номер регистрации (свидетельства):
2022685576
Дата регистрации: **26.12.2022**
Номер и дата поступления заявки:
2022684362 12.12.2022
Дата публикации и номер бюллетеня:
26.12.2022 Бюл. № 1
Контактные реквизиты:
+7-903-700-79-86, m.kalugin@ispras.ru

Автор(ы):
Булгакова Мария Ивановна (RU),
Гетьман Александр Игоревич (RU),
Горюнов Максим Николаевич (RU),
Мацкевич Андрей Георгиевич (RU),
Перминов Андрей Игоревич (RU),
Рыболовлев Дмитрий Александрович (RU)
Правообладатель(и):
Федеральное государственное бюджетное учреждение науки Институт системного программирования им. В.П. Иванникова Российской академии наук (RU)

Название программы для ЭВМ:
«Программа защиты от атаки уклонения в системе обнаружения вторжений»

Реферат:
Программа предназначена для защиты от атак уклонения в отношении модели машинного обучения в системе обнаружения компьютерных атак. Состязательные примеры генерируются перебором значений одного из признаков классификации для каждой сессии тестовой выборки с меткой "атака". При изменении ответа модели, пример считается состязательным. Для защиты в обучающую выборку добавляются найденные примеры с корректной разметкой. После обучения на них модель верно классифицирует состязательные примеры, то есть обеспечивается устойчивость классификатора к состязательным атакам. Программа разработана ИСП РАН в рамках мероприятия «Методы обнаружения и противодействия атакам с внедрением закладок и зловредного кода в модели машинного обучения» Программы центра ИИ «Разработка методов и технологий создания систем доверенного искусственного интеллекта» по направлению доверенный искусственный интеллект. Тип ЭВМ: IBM PC - совмест. ПК. ОС: Linux.

Язык программирования: Python (Jupyter Notebook)

Объем программы для ЭВМ: 46 КБ

РОССИЙСКАЯ ФЕДЕРАЦИЯ

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

RU2024692147

ГОСУДАРСТВЕННАЯ РЕГИСТРАЦИЯ ПРОГРАММЫ ДЛЯ ЭВМ

Номер регистрации (свидетельства):
2024692147
Дата регистрации: **26.12.2024**
Номер и дата поступления заявки:
2024691617 12.12.2024
Дата публикации и номер бюллетеня:
26.12.2024 Бюл. № 1

Контактные реквизиты:
m.kalugin@ispras.ru

Автор(ы):
Алексеевская Ирина Сергеевна (RU),
Архипенко Константин Владимирович (RU),
Прилепская Дарья Дмитриевна (RU),
Перминов Андрей Игоревич (RU),
Лобастова Екатерина Олеговна (RU),
Голодков Александр Олегович (RU)

Правообладатель(и):
Федеральное государственное бюджетное учреждение науки Институт системного программирования им. В.П. Иванникова Российской академии наук (RU)

Название программы для ЭВМ:
«Программное обеспечение для выявления и устранения предвзятости моделей машинного обучения»

Реферат:
Программное обеспечение представляет собой библиотеку, содержащую методы устранения предвзятости для распространённых генеративных моделей: обычных и мультимодальных языковых моделей, диффузионных моделей. Программа разработана ИСП РАН в рамках мероприятия «Разработка программного обеспечения для выявления и устранения предвзятости моделей машинного обучения» Программы центра ИИ «Разработка методов и технологий создания систем доверенного искусственного интеллекта» по направлению доверенный искусственный интеллект. Тип ЭВМ: IBM PC-совмест. ПК; ОС: Linux.

Язык программирования: Python

Объем программы для ЭВМ: 912 КБ

Приложение Б

Доказательства теорем

Б.1 Доказательство теорем. 1

Доказательство опубликовано в работе [3] совместно с Лукьяновым К.С., Яськовым П.А., Коваленко А.П. и Турдаковым Д.Ю.. Авторство доказательства теоремы принадлежит Яськову П. А..

Как показано в п. 2.1 решение задачи 2.4 существует и совпадает с условным математическим ожиданием

$$g_\alpha(x) = E_\alpha(Y|X = x), \quad x \in [0,1]^d,$$

которое определено однозначно λ_α-п. н., иначе говоря, одновременно P_X-п. н. и λ-п. н.

Чтобы вывести явную формулу для g_α, заметим следующее. Прежде всего, $E_\alpha g_\alpha^2(X) < \infty$, поскольку значения Y ограничены. Тем самым, в 2.4 из рассмотрения можно исключить все (борелевские) функции f такие, что $E_\alpha f^2(X) = \infty$, и считать последний интеграл конечным. Тогда по определению P_α имеем

$$E_\alpha(Y - f(X))^2 = (1-\alpha)E(Y - f(X))^2 + \alpha \int_{[0,1]^d} f^2(x)dx$$

$$= (1-\alpha)E(Y - g(X))^2 + L(f),$$

где

$$L(f) = (1-\alpha)E(g(X) - f(X))^2 + \alpha \int_{[0,1]^d} f^2(x)dx.$$

Перепишем $L(f)$ в следующем виде:

$$L(f) = (1-\alpha)\int_A (g(x)-f(x))^2 P_X(dx)$$

$$+\int_{\{x\in\mathbb{S}\setminus A:\rho(x)>0\}} \left[(1-\alpha)(g(x)-f(x))^2\rho(x)+\alpha f^2(x)\right]dx$$

$$+\alpha\int_{[0,1]^d\setminus\mathbb{S}\cup\{x\in\mathbb{S}\setminus A:\rho(x)=0\}} f^2(x)dx.$$

Поскольку минимум выражения

$$(1-\alpha)(a-z)^2 b+\alpha z^2$$

по z при $b>0$ достигается в точке

$$z_* = \frac{(1-\alpha)ab}{\alpha+(1-\alpha)b},$$

то минимум $L(f)$ очевидным образом достигается на g_α, заданном по формуле 2.5. Перепишем равенство выше в виде

$$(1-\alpha)(a-z)^2 b+\alpha z^2 = (\alpha+(1-\alpha)b)(z-z_*)^2+\frac{\alpha(1-\alpha)a^2 b}{\alpha+(1-\alpha)b}.$$

Следовательно, с точностью до слагаемого R_α, не зависящего от f, для g_α из (2.5) имеет место соотношение

$$L(f) = (1-\alpha)\|g_\alpha - f\|^2_{P_X}+\alpha\|g_a-f\|^2_\lambda + R_\alpha,$$

где $\|\cdot\|_\mu$ – это L_2-норма относительно меры $\mu = P_X$ или $\mu=\lambda$, и учитываем, что g_α равно нулю вне $\mathbb{S}$ или при $\rho(x)=0$ и $g_\alpha = g$ на A. Поскольку все решения задачи 2.4 определены P_X- и λ-п. н. однозначно, это соотношение с $L(f)$ будет справедливо и для любого другого решения g_α, не обязательно заданного по формуле 2.5.

Пусть теперь s_α — классификатор из условия теоремы. Для проверки свойства (i) достаточно показать, что $P(s(X)=s_\alpha(X))=1$, где s – байесовский классификатор из (2.3) для $g(x)=E(Y|X=x)$ на $[0,1]^d$. Последнее эквивалентно тому, что

$$P(g(X)g_\alpha(X) > 0 \quad \text{или} \quad g(X) = g_\alpha(X) = 0) = 1.$$

Поскольку g_α в определении s_α определено однозначно P_X-п. н. и может быть задано формулой 2.5, можно считать, что g_α всюду задано этой формулой. Осталось заметить, что условие

$$g(x)g_\alpha(x) > 0 \quad \text{или} \quad g(x) = g_\alpha(x) = 0$$

выполнено по определению в тех случаях, когда либо $x \in A$, либо $\rho(x) > 0$ и $x \in \mathbb{S} \setminus A$; при этом остальные случаи имеют нулевую вероятность:

$$P(\rho(X) = 0, X \in \mathbb{S} \setminus A) = P(X \notin \mathbb{S}) = 0.$$

Доказательство свойства (i) завершено.

Свойство (ii) следует из (2.5) и того, что g_α в определении s_α определено однозначно λ-п. н.

Теорема доказана.

Б.2 Доказательство теорем. 4

Доказательство опубликовано в работе [4].

Б.3 Используемые теоремы и леммы

В приложении приведены все используемые теоремы, определения и леммы, не требующие дополнений.

Определение 1. Сходимости

Сходимость по вероятности.

$$X_n \xrightarrow{P} X \quad \Longleftrightarrow \quad \forall\, \varepsilon > 0 : \ \lim_{n\to\infty} \mathbb{P}(|X_n - X| > \varepsilon) = 0.$$

Сходимость почти наверное.

$$X_n \xrightarrow{a.s.} X \quad \Longleftrightarrow \quad \mathbb{P}\Big(\lim_{n\to\infty} X_n = X\Big) = 1.$$

Сходимость в $L^2(P_X)$.

$$X_n \xrightarrow{L^2(P_X)} X \quad \Longleftrightarrow \quad \mathbb{E}[(X_n - X)^2] \to 0.$$

Норма в пространстве $L^2(P_X)$:

$$\|f\|_{L^2(P_X)} = \Big(\int |f(x)|^2 \, dP_X(x)\Big)^{1/2}.$$

Лемма 1. ***Бореля–Кантелли***

$$\limsup_{n\to\infty} \mathsf{E}_n = \bigcap_{m=1}^{\infty} \bigcup_{n\geqslant m} \mathsf{E}_n.$$

Если $\sum_{n=1}^{\infty} \mathbb{P}(\mathsf{E}_n) < \infty$, *то* $\mathbb{P}(\limsup \mathsf{E}_n) = 0$.

Теорема 5. ***О сходимости персептрона к целевой функции****. Согласно [93].*

Пусть $Y = g(X) + \xi$ *на компакте* $K = [0,1]^d$, $(X_i, Y_i)_{i=1}^n$ — *i.i.d.*, p_X *ограничена на* K, $\mathbb{E}[\xi \mid X] = 0$, $\mathrm{Var}(\xi) \leqslant \sigma_\xi^2$, *активация* $\eta(t) = \mathrm{ReLU}(t)$. *Обозначим*

через $\mathcal{N}_\eta(S_{\mathrm{nnz},n},B_n,L_n)$ *класс полносвязных ReLU-сетей с* L_n *слоями, не более* $S_{\mathrm{nnz},n}$ *ненулевых параметров и ограничением* $|w|\leqslant B_n$*;* $\hat g_{L_n}$ *— ERM-оценка по MSE, с клиппингом выхода* $[-T_{\max},T_{\max}]$.

Пусть

$$\mathcal{F}_{M_F,J,\alpha,\beta}=\Big\{\,f=\sum_{m=1}^{M_F} f_m\,\mathbf{1}_{R_m}\ :\ f_m\in\mathcal{C}^\beta([0,1]^d),\ R_m\in\mathcal{R}_{\alpha,J}\Big\},$$

где $\beta=q+s$, $s\in(0,1]$, *а* $\mathcal{R}_{\alpha,J}$ *— пересечения областей вида*

$$\mathcal{R}_{\alpha,J}=\Big\{R=\bigcap_{k=1}^{J} S_{i_k,u_k}^{\zeta_k}:\ u_k\in\mathcal{C}^\alpha([0,1]^{d-1}),\ \zeta_k\in\{\leqslant,\geqslant\},\ i_k\in\{1,\dots,d\}\Big\},$$

$$S_{i,u}^{\leqslant}=\{\,x\in[0,1]^d:\ x_i\leqslant u(x_1,\dots,x_{i-1},x_{i+1},\dots,x_d)\,\}.$$

Если $g\in\mathcal{F}_{M_F,J,\alpha,\beta}$, *то существуют* $c_{\mathrm{IF}},c'_{\mathrm{IF}},C_{\mathrm{IF}}>0$, *целое* $s\geqslant 2$, $T_{\max}\geqslant\|g\|_\infty$ *и архитектура* $(S_{\mathrm{nnz},n},B_n,L_n)$ *такая, что*

$$S_{\mathrm{nnz},n}=c'_{\mathrm{IF}}\max\Big\{n^{\frac{d}{2\beta+d}},\,n^{\frac{d-1}{\alpha+d-1}}\Big\},\qquad B_n\leqslant c_{\mathrm{IF}}\,n^s,\qquad L_n\leqslant c_{\mathrm{IF}}\Big(1+\max\{\tfrac{\beta}{d},\tfrac{\alpha}{2(d-1)}\}\Big),$$

и с вероятностью $\geqslant 1-c_{\mathrm{IF}}n^{-2}$ *выполняется*

$$\|\hat g_{L_n}-g\|^2_{L^2(P_X)}\ \leqslant\ C_{\mathrm{IF}}\max\Big\{n^{-\frac{2\beta}{2\beta+d}},\,n^{-\frac{\alpha}{\alpha+d-1}}\Big\}(\log n)^2.$$

Теорема 6. ***Об инъективности*** $x\mapsto|Ax+b|$. *Согласно [94].*

Для $A\in\mathbb{R}^{m\times d}$, $b\in\mathbb{R}^m$ *положим*

$$M_{A,b}(x)=\big(\,|a_1^\top x+b_1|,\dots,|a_m^\top x+b_m|\,\big).$$

Если $m\geqslant 2d$ *и выполняется условие generic-типа:*

$$\forall I\subseteq\{1,\dots,m\}:\quad \big(b_I\in\operatorname{span}(A_I)\big)\ \Rightarrow\ \operatorname{span}(A_{I^c})=\mathbb{R}^d,$$

то $M_{A,b}$ *инъективно.*

Подразумевается что $\operatorname{span}(A_I)$ *– линейная оболочка по столбцам матрицы, а* $\operatorname{span}(A_{I^c})$ *– линейная оболочка по строкам матрицы.*

Теорема 7. ***О билипшицевости*** $x \mapsto |Ax + b|$. *Согласно [94].*

Если отображение из теорем. 6 инъективно и $\Omega \subset \mathbb{R}^d$ *— компакт, то существуют константы* $C_{\text{Lip}}, c_{\text{Lip}} > 0$, *зависящие от* (A,b) *и* Ω, *такие что для всех* $x,y \in \Omega$

$$\frac{c_{\text{Lip}}}{1 + \|x\| + \|y\|} \|x - y\| \;\leqslant\; \|M_{A,b}(x) - M_{A,b}(y)\| \;\leqslant\; C_{\text{Lip}} \|x - y\|.$$

Теорема 8. ***О сохранении кусочной гладкости композиции функций***. *Согласно [95].*

Если F *и* φ *— кусочно-гладкие функции, то* $F \circ \varphi$ *также кусочно-гладкая.*

Теорема 9. ***О кусочной аффинности обратного отображения***. *Согласно [96].*

Если φ *— кусочно-аффинное инъективное отображение с невырожденными областями линейности, то* φ^{-1} *также кусочно-аффинна.*

Теорема 10. ***О состоятельности гистограмной регрессии***. *Согласно [97].*

Пусть по выборке $T_n = \{(X_i,Y_i)\}_{i=1}^n$ *строится разбиение* $\Pi_n = \psi_n(T_n)$. *Для компакта* V *обозначим* $N_{\text{cells}}(\Pi : V)$ *— максимум числа ячеек, пересекающих* V; *многоклассовая функция роста* $\Delta_n^*(\Pi) = \max_{x_1,\ldots,x_n} \Delta((x_1,\ldots,x_n),\Pi)$. *Обозначим ячейку с* x *через* $\mathcal{A}_n(x)$ *и диаметр множества через* $\operatorname{diam}(\cdot)$.

Если при $n \to \infty$ *выполняются:*

$$(a)\quad \frac{N_{\text{cells}}(\Pi_n : V)}{n} \to 0 \;\; \forall\, V \subset \mathbb{R}^d; \qquad (b)\quad \frac{\log \Delta_n^*(\Pi_n)}{n} \to 0;$$

$$(c)\quad \forall\, \gamma > 0,\, \delta \in (0,1) : \inf_{\mathsf{S}:\, P(\mathsf{S}) \geqslant 1-\delta} P\big(x :\; \operatorname{diam}(\mathcal{A}_n(x) \cap \mathsf{S}) > \gamma\big) \xrightarrow{a.s.} 0,$$

то гистограммная регрессия

$$\hat{r}_n(x) = \frac{\sum_{i=1}^n Y_i \, \mathbf{1}\{X_i \in \mathcal{A}_n(x)\}}{\sum_{i=1}^n \mathbf{1}\{X_i \in \mathcal{A}_n(x)\}}$$

состоятельна: $\int |\hat{r}_n(x) - g(x)|^2 \, dP_X(x) \to 0$, *то есть* $\hat{r}_n \xrightarrow{L^2(P_X)} g$.

Теорема 11. ***О числе ячеек ReLU-MLP***. *Согласно [98].*

Пусть L_n — число скрытых слоёв, n_l — число нейронов в слое l, d — размерность входа. Тогда для ReLU-сети

$$R_{\max}(n) \;\leqslant\; \prod_{l=1}^{L_n} \sum_{j=0}^{\min(d,n_l)} \binom{n_l}{j}, \qquad \sum_{j=0}^{\min(d,n_l)} \binom{n_l}{j} = \Theta\big(n_l^{\min(d,n_l)}\big).$$

Определение 2. Функция роста.

Для класса бинарных гипотез H

$$\Pi_H(m) = \max_{\{x_1,\dots,x_m\}\subset X} \big|\{(h(x_1),\dots,h(x_m)) :\; h \in H\}\big|.$$

Определение 3. VC-размерность.

Это максимальное число точек, которое класс функций способен разделить всеми возможными способами.

Лемма 2. *Сауэра–Шелаха*

Если $\mathrm{VCdim}(H) = Z < \infty$, *то для всех* $m \in \mathbb{N}$:

$$\Pi_H(m) \;\leqslant\; \sum_{i=0}^{Z} \binom{m}{i}, \qquad \textit{в частности, при } m \geqslant Z :\; \Pi_H(m) \;\leqslant\; \Big(\frac{e\,m}{Z}\Big)^{Z}.$$

Теорема 12. *О VC-размерности пересечений полупространств*. *Согласно [99].*

Для класса всех пересечений не более чем M_{hs} аффинных полупространств в $\mathbb{R}^d$:

$$\mathrm{VCdim} = \Theta\big(d\, M_{\mathrm{hs}} \log M_{\mathrm{hs}}\big).$$

Теорема 13. *О разбиении случайными гиперплоскостями*. *Согласно [100].*

Пусть H_{hp} — число случайных аффинных гиперплоскостей (из непрерывного распределения), $K \subset \mathbb{R}^d$ — компакт, $D = \mathrm{diam}(K)$, а $\{C_j\}$ — ячейки разбиения K и $\gamma > 0$. Если

$$H_{\mathrm{hp}} \;\gtrsim\; \gamma^{-12}\, \omega(K)^2 D^{10},$$

то

$$\mathbb{P}(\forall j :\; \mathrm{diam}(C_j) \leqslant \gamma) \;\geqslant\; 1 - 2\exp\Big(\frac{-c\,\varepsilon^4 H_{\mathrm{hp}}}{D^4}\Big).$$

Б.3.1 Необходимые леммы

Лемма 3. *Из сходимости* $X_n \xrightarrow{L^2(P_X)} X$ *следует* $X_n \xrightarrow{P} X$.

Доказательство. Согласно неравенству Маркова $\forall \varepsilon > 0$:

$$\mathbb{P}(|X_n - X| \geqslant \varepsilon) = \mathbb{P}((X_n - X)^2 \geqslant \varepsilon^2) \leqslant \frac{\mathbb{E}[(X_n - X)^2]}{\varepsilon^2}. \tag{Б.1}$$

Так как $X_n \xrightarrow{L^2(P_X)} X$, по определению

$$\mathbb{E}[(X_n - X)^2] \to 0. \tag{Б.2}$$

В неравенстве Маркова $\frac{\mathbb{E}[(X_n - X)^2]}{\varepsilon^2} \to 0$, следовательно

$$\mathbb{P}(|X_n - X| \geqslant \varepsilon) \to 0, \tag{Б.3}$$

откуда по определению сходимости по вероятности следует $X_n \xrightarrow{P} X$. Лемма 3 доказана.

Лемма 4. *Мера Лебега* $\lambda^{m(d+1)}$ *generic-множества из теорем. 6, теорем. 7 равна* 1.

Доказательство. Рассмотрим дополнение к generic-множеству. Как видно по условию инъективности, неинъективны такие A и b, что

$$\exists I: \quad b_I \in \operatorname{span}(A_I), \quad \operatorname{span}(A_{I^c}) \neq \mathbb{R}^d. \tag{Б.4}$$

Случай 1: $|I| \leqslant d$ *(тогда* $|I^c| \geqslant m - d \geqslant d$*).* Тогда A_{I^c} имеет хотя бы d строк, и попадание в дополнение generic-множества возможно только если $\operatorname{rank}(A_{I^c}) < d$. Множество $Z_I = \{A : \operatorname{rank}(A_{I^c}) < d\}$ задаётся обращением в ноль всех миноров $d \times d$ матрицы A_{I^c}. Это множество является множеством корней полинома, а согласно [101] множество корней аналитической функции в $\mathbb{R}^d$ имеет меру Лебега 0. Следовательно, и λ^{md}, и $\lambda^{m(d+1)}$ меры дополнения равны 0.

Случай 2: $|I| > d$ *(тогда* $|I^c| \leqslant d - 1$*).* Здесь $\operatorname{rank}(A_{I^c}) < d$ всегда, поэтому элемент не generic тогда и только тогда, когда $b_I \in \operatorname{span}(A_I)$. Полученное множество

$$B_I(A) = \{b \in \mathbb{R}^m : b_I \in \operatorname{span}(A_I)\} \tag{Б.5}$$

является линейным подпространством в $\mathbb{R}^m$ меньшей размерности. Для любого такого подпространства мера Лебега $\lambda^m = 0$. Следовательно, и $\lambda^{m(d+1)}$ этого множества также равна 0.

Так как дополнение generic-множества есть конечное объединение множеств меры 0, его $\lambda^{m(d+1)}$ мера также равна 0. Следовательно, мера Лебега generic-множества равна 1. Лемма 4 доказана.

Введём следующие обозначения: $\mathcal{A}_n$ — класс всех возможных ячеек.
$X_{1:n} = \{x_1, \ldots, x_n\} \subset V$ — фиксированный набор точек.
Π — семейство разбиений пространства.
$N_{\text{cells}}(\Pi : V)$ — максимум числа ячеек, пересекающих компакт V для Π.
$\Delta^*(X_{1:n}, \Pi)$ — многоклассовая функция роста (каждой точке присваивается индекс ячейки, к которой она принадлежит).
$\Delta(X_{1:n}, \mathcal{A}_n)$ — бинарная функция роста класса ячеек $\mathcal{A}_n$ (каждая ячейка рассматривается отдельно).

Лемма 5.

$$\Delta^*(X_{1:n}, \Pi) \leqslant \sum_{t=1}^{N_{\max}} \big(\Delta(X_{1:n}, \mathcal{A}_n)\big)^t, \tag{Б.6}$$

где $N_{\max} := N_{\text{cells}}(\Pi : V)$.

Доказательство. Зафиксируем число ячеек t. Все разбиения $P \in \Pi_t$ задаются набором ячеек $C_1, \ldots, C_t \in \mathcal{A}_n$. Каждое разбиение индуцирует t-классовую принадлежность точек к ячейкам. Так как бинарный паттерн принадлежности по каждой ячейке принимает не более чем $\Delta(X_{1:n}, \mathcal{A}_n)$ значений, то композиция t таких паттернов принимает не более чем $(\Delta(X_{1:n}, \mathcal{A}_n))^t$ значений. Следовательно, для фиксированного t

$$\Delta^*(X_{1:n}, \Pi_t) \leqslant \big(\Delta(X_{1:n}, \mathcal{A}_n)\big)^t. \tag{Б.7}$$

Если число ячеек произвольно от 1 до верхней границы $N_{\max}$, то имеет место неравенство

$$\Delta^*(X_{1:n}, \Pi) \leqslant \sum_{t=1}^{N_{\max}} \big(\Delta(X_{1:n}, \mathcal{A}_n)\big)^t, \tag{Б.8}$$

откуда следует грубая оценка

$$\Delta^*(X_{1:n}, \Pi) \leqslant N_{\max} \cdot \big(\Delta(X_{1:n}, \mathcal{A}_n)\big)^{N_{\max}}. \tag{Б.9}$$

Лемма 5 доказана.

Б.3.2 Сходимость разности нейросетевой и гистограммной регрессии

Требуемый факт доказывается через сходимость разности, используя целевую функцию $g(x)$:

$$g(x) = \mathbb{E}[Y \mid X = x], \qquad x \in K. \tag{Б.10}$$

Если $c_n(X) \xrightarrow{P} g(X)$ и $h_n(X) \xrightarrow{P} g(X)$, то в таком случае:

$$c_n(X) - h_n(X) \xrightarrow{P} 0. \tag{Б.11}$$

И, следовательно, по определению сходимости по вероятности

$$\lim_{n\to\infty} \mathbb{P}(|c_n(x) - h_n(x))| > \varepsilon) = 0. \tag{Б.12}$$

Таким образом, в пределе вероятность события, состоящего в том, что выход персептрона не совпадает с выходом гистограммной регрессии, равна нулю. Это позволяет говорить об асимптотической эквивалентности персептрона и гистограммной регрессии, а также возможности оценки гистограммной регрессии при помощи выздного значения персептрона.

Б.3.3 Сходимость персептрона к целевой функции

В данном пункте воспользуемся теорем. 5. Так как теорема напрямую не предусматривает заморозку первого слоя персептрона, представим фиксированный первый слой в виде отображения из исходного пространства в новое пространство признаков:

$$z = |W_1 x + b_1| =: \varphi(x), \qquad x = \varphi^{-1}(z). \tag{Б.13}$$

Тогда выход персептрона можно записать в виде

$$c_n(x) = f_{\theta,n}(\varphi(x)) = f_{\theta,n}(z), \tag{Б.14}$$

где $f_{\theta,n}$ — обучаемая часть персептрона. Целевая функция в новых координатах:

$$\tilde{g}(z) = g(\varphi^{-1}(z)). \tag{Б.15}$$

Задача сходимости переформулируется как

$$f_{\theta,n}(Z) \xrightarrow{P} \tilde{g}(Z). \tag{Б.16}$$

Так как это равносильно

$$f_{\theta,n}(\varphi(X)) \xrightarrow{P} g(\varphi^{-1}(\varphi(X))), \tag{Б.17}$$

то получаем

$$c_n(X) \xrightarrow{P} g(X). \tag{Б.18}$$

Чтобы выкладки были корректны, необходимо потребовать от отображения, порождаемого замороженным слоем:

– *инъективности*, чтобы существовало обратное отображение;
– *сохранения кусочной гладкости* $\tilde{g}(z)$, чтобы теорема теорем. 5 была применима;
– *билипшицевости* и ограниченности липшицевых констант, чтобы константы гладкости оставались контролируемыми, в том числе для получения корректных скоростей сходимости.

Покажем каждый из необходимых пунктов ниже.

Инъективность.

Согласно теорем. 6 и доказанной лемм. 4 и так как W_1 и b_1 взяты независимо из непрерывного распределения, при условии на число нейронов $r_n \geqslant 2d$, можно утверждать, что

$$\mathbb{P}(\varphi \text{ инъективно}) = 1.$$

Билипшицевость.

Согласно теорем. 7 и лемм. 4, а также с учётом того, что изначальная задача рассматривается на компакте K, можно утверждать, что с вероятностью 1 отображение φ билипшицево и имеет аналитические оценки c_{Lip} и C_{Lip}.

Сохранение кусочной гладкости.

Покажем, что при кусочно-гладкой g и замене переменных при помощи φ^{-1}, кусочная гладкость сохраняется. Согласно теорем. 8, если F и φ^{-1} кусочно-гладкие, то $F \circ \varphi^{-1}$ тоже кусочно-гладкая функция. Согласно теореме теорем. 9,

если φ — кусочно-линейное отображение, для которого с вероятностью 1 (при попадении в generic множество теорем. 6) ранг всех рассматриваемых подматриц, формирующих ячейки, является полным, и следовательно области линейности описаны невырожденными матрицами, то φ^{-1} также является кусочно-линейной функцией. Таким образом, так как инъективное отображение, заданное замороженным первым слоем, является кусочно-линейной функцией, можно утверждать, что $\tilde{g}(z) = g(\varphi^{-1}(z))$ как композиция кусочно-гладких функций является кусочно-гладкой функцией. Для применения теорем. 5 в пространстве признаков $z = \varphi(x)$ необходимо, чтобы распределение случайного вектора z обладало ограниченной плотностью на образе компакта. Это выполняется, если исходный x имеет ограниченную плотность на компакте K, а φ является билипшицевым отображением: тогда распределение z абсолютно непрерывно, и его плотность ограничена константами, зависящими от липшицевых констант.

Сходимость по вероятности.

Продемонстрируем, что из результатов теоремы теорем. 5 следует сходимость по вероятности к целевой функции.

Согласно теорем. 5:

$$\mathcal{R}_n = \mathbb{E}\big((c_n - g)^2\big), \tag{Б.19}$$

$$\mathbb{P}(\mathcal{R}_n \leqslant C\rho_n) \geqslant 1 - c\,n^{-2}, \tag{Б.20}$$

где $\rho_n \to 0$ при $n \to \infty$.

Обозначим событие

$$\mathsf{E}_n = \{\mathcal{R}_n \leqslant C\rho_n\}, \qquad \mathsf{E}_n^c = \{\mathcal{R}_n \geqslant C\rho_n\}. \tag{Б.21}$$

Тогда

$$\sum_{n=1}^{\infty} \mathbb{P}(\mathsf{E}_n^c) \;\leqslant\; \sum_{n=1}^{\infty} c\,n^{-2} \;< \infty. \tag{Б.22}$$

По лемме Бореля–Кантелли получаем

$$\mathbb{P}(\mathsf{E}_n^c \text{ бесконечно часто}) = 0, \tag{Б.23}$$

то есть

$$\mathbb{P}\big(\mathsf{E}_n \text{ для почти всех } n\big) = 1. \tag{Б.24}$$

Следовательно,

$$\mathcal{R}_n \to 0, \qquad \mathbb{E}\big((c_n - g)^2\big) \to 0, \tag{Б.25}$$

что можно записать как:

$$c_n(x) \xrightarrow{L^2(P_X)} g(x). \tag{Б.26}$$

Откуда, согласно лемм. 3, следует:

$$c_n(X) \xrightarrow{P} g(X) \tag{Б.27}$$

Таким образом, согласно теорем. 5, на компакте K при ограниченной плотности и подходящей целевой функции $g(x)$, для последовательности персептронов с соответствующей архитектурой и кусочно-линейными активациями, обучаемых по MSE, при $n \to \infty$ выполняется

$$c_n(X) \xrightarrow{P} g(X). \tag{Б.28}$$

Так как

$$|x| = \mathrm{ReLU}(x) + \mathrm{ReLU}(-x), \tag{Б.29}$$

то нейрон с модульной функцией активации можно представить как два нейрона с кусочно линейными активациями. Следовательно, все изложенные выкладки корректны и для моделей с модульной функцией активации (с точностью до констант).

Б.3.4 Сходимость гистограммной регрессии к целевой функции

Воспользуемся теорем. 10. Для её применения необходимо выполнение трёх условий. Рассмотрим каждое из них в контексте гистограммы, построенной по ячейкам многослойного персептрона.

Условие (a).

В данном пункте необходимо оценить рост числа ячеек персептрона. Применим оценку максимального числа ячеек из теоремы теорем. 11. Для случая, когда имеется один скрытый слой ширины r_n и L_n скрытых слоёв ширины k_n, получаем:

$$R_{\max}(n) \;=\; O\Big(k_n^{L_n \min(d,k_n)}\, r_n^{\,d}\Big). \tag{Б.30}$$

Так как активация в виде модуля, аналогично ReLU, разбивает пространство на два полупространства, действуя в каждом линейно, указанная оценка сохраняется.

Для выполнения условия (а) необходимо требовать:

$$k_n^{L_n \min(d,k_n)}\, r_n^{\,d} \;=\; o(n). \tag{Б.31}$$

В случае фиксированной размерности входа d и числа нейронов в скрытых слоях не меньше d, условие принимает вид:

$$k_n^{L_n d}\, r_n^{\,d} \;=\; o(n). \tag{Б.32}$$

Таким образом, имеется чёткое ограничение на рост ширины и глубины сети с ростом числа данных.

Условие (б).

В данном пункте необходимо показать ограничение на скорость роста богатства возможных разбиений пространства. По лемм. 5 верхняя оценка многоклассовой функции роста выражается через бинарную:

$$\Delta_n^*(\Pi_n,V) \;\leqslant\; N_{\text{cells}}(\Pi_n : V)\, \Delta_n(\mathcal{A}_n,V)^{N_{\text{cells}}(\Pi_n:V)}. \tag{Б.33}$$

В результате математических преобразований и разделив на n, получаем:

$$\frac{1}{n}\log \Delta_n^*(\Pi_n,V) \;\leqslant\; \frac{\log N_{\text{cells}}(\Pi_n : V)}{n} + \frac{N_{\text{cells}}(\Pi_n : V)}{n}\log\big(\Delta_n(\mathcal{A}_n,V)\big). \tag{Б.34}$$

Применяя лемму Сауэра–Шелаха лемм. 2, обозначим

$$Z_n = \text{VCdim}(\mathcal{A}_n). \tag{Б.35}$$

По лемме Сауэра–Шелаха, при $Z_n \leqslant n$:

$$\Delta_n(\mathcal{A}_n,V) \;\leqslant\; \left(\frac{e\,n}{Z_n}\right)^{Z_n}. \tag{Б.36}$$

Следовательно:

$$\log \Delta_n(\mathcal{A}_n, V) = O\big(Z_n \log n\big), \tag{Б.37}$$

и

$$\frac{1}{n} \log \Delta_n^*(\Pi_n, V) \;\leqslant\; \frac{\log N_{\text{cells}}(\Pi_n : V)}{n} + \frac{N_{\text{cells}}(\Pi_n : V)}{n} Z_n \log n. \tag{Б.38}$$

Первое слагаемое стремится к нулю по условию (а). Следовательно, необходимо требовать стремления к нулю второго слагаемого. Подставим оценку на число ячеек из условия (а):

$$\frac{k_n^{L_n \min(d, k_n)} r_n^d}{n} Z_n \log n \;\to\; 0. \tag{Б.39}$$

Так как Z_n — VC-размерность класса ячеек многослойного персептрона, где каждая ячейка задаётся пересечением конечного числа аффинных полупространств, то по теорем. 12:

$$Z_n \;\leqslant\; d\, M_n \log M_n, \tag{Б.40}$$

где M_n — число нейронов в сети.

Для сети с одним скрытым слоем ширины r_n и L_n скрытыми слоями одинаковой ширины k_n:

$$Z_n \;\leqslant\; d\,(k_n L_n + r_n)\, \log(k_n L_n + r_n). \tag{Б.41}$$

Таким образом, условие (б) принимает вид:

$$k_n^{L_n \min(d, k_n)} r_n^d\, d\,(k_n L_n + r_n)\, \log(k_n L_n + r_n)\, \frac{\log n}{n} \;\to\; 0. \tag{Б.42}$$

Условие (в).

В данном пункте необходимо показать, что при росте числа ячеек персептрона диаметры ячеек стремятся к нулю.

В статье [102] авторы, опираясь на теорем. 13 о разбиении пространства случайными гиперплоскостями, утверждают, что для слоя нейронов с кусочно-линейной функцией активации и случайными гауссовыми весами разбиение входного пространства на ячейки индуцирует разбиение случайными гиперплоскостями.

Так как для глубокой сети с кусочно-линейными активациями разбиение входа, индуцированное композицией до слоя l, является уточнением разбиения, индуцированного до слоя $l-1$ (каждый следующий слой дробит регионы на более мелкие части). Поэтому на любом фиксированном компакте верхняя оценка на диаметры ячеек, заданная после первого слоя, сохраняется (не ухудшается) для всей глубокой сети. Это утверждение верно на generic-множествах параметров, то есть с вероятностью 1.

Введём событие:

$$\mathsf{E}_{H_{\text{hp}}}(\varepsilon) \;=\; \{\; \forall\, \mathcal{A}_{H_{\text{hp}}} \subset K :\; \operatorname{diam}(\mathcal{A}_{H_{\text{hp}}}) \leqslant \varepsilon D \;\}, \tag{Б.43}$$

где $\mathcal{A}_{H_{\text{hp}}}$ — ячейка, полученная в результате разбиения компакта K H_{hp} случайными гиперплоскостями, а $\mathcal{A}_{H_{\text{hp}}}(x)$ — ячейка, содержащая точку x.

Для произвольного $\gamma > 0$ и множества S такого, что $P_X(\mathsf{S}) \geqslant 1-\delta$, и при событии $\mathsf{E}_{H_{\text{hp}}}(\varepsilon D)$ с $\varepsilon D \leqslant \gamma$, выполняется:

$$\operatorname{diam}(\mathcal{A}_{H_{\text{hp}}}(x) \cap \mathsf{S}) \;\leqslant\; \operatorname{diam}(\mathcal{A}_{H_{\text{hp}}}(x)) \;\leqslant\; \varepsilon D \;\leqslant\; \gamma, \tag{Б.44}$$

и

$$\mathbf{1}\{\operatorname{diam}(\mathcal{A}_{H_{\text{hp}}}(x) \cap \mathsf{S}) > \gamma\} \;=\; 0. \tag{Б.45}$$

Следовательно:

$$\inf_{\mathsf{S}:P_X(\mathsf{S})\geqslant 1-\delta} \mathbb{P}_X\big(\operatorname{diam}(\mathcal{A}_{H_{\text{hp}}}(x)\cap \mathsf{S}) > \gamma\big) \;\leqslant\; \mathbb{P}_X\big(\operatorname{diam}(\mathcal{A}_{H_{\text{hp}}}(x)\cap K) > \gamma\big) \;\leqslant\; \mathbf{1}\{\mathsf{E}_{H_{\text{hp}}}(\gamma)^{\,c}\} \tag{Б.46}$$

Так как по теорем. 13

$$\mathbb{P}\big(\mathsf{E}_{H_{\text{hp}}}(\gamma)\big) \;\geqslant\; 1 - 2\, e^{\frac{-c\,\gamma^4 H_{\text{hp}}}{D^4}}, \tag{Б.47}$$

то математическое ожидание по всем разбиениям удовлетворяет:

$$\mathbb{E}\left[\inf_{\mathsf{S}:P_X(\mathsf{S})\geqslant 1-\delta} \mathbb{P}_X\big(\operatorname{diam}(\mathcal{A}_{H_{\text{hp}}}(x) \cap \mathsf{S}) > \gamma\big)\right] \;\leqslant\; \mathbb{P}\big(\mathsf{E}_{H_{\text{hp}}}(\gamma)^{\,c}\big) \;\leqslant\; 2\, e^{\frac{-c\,\gamma^4 H_{\text{hp}}}{D^4}}. \tag{Б.48}$$

Покажем сходимость почти наверное при $n \to \infty$ и числе гиперплоскостей (нейронов первого слоя) r_n, если

$$r_n \;\geqslant\; C\,\gamma^{-12}\,\omega^2(K) D^{10}, \tag{Б.49}$$

и

$$\sum_{n=1}^{\infty} e^{\frac{-c\gamma^4 r_n}{D^4}} \;<\; \infty, \tag{Б.50}$$

где D – диаметр компакта. Тогда по лемме Бореля–Кантелли:

$$\inf_{\mathsf{S}:P_X(\mathsf{S})\geqslant 1-\delta} \mathbb{P}_X\big(\operatorname{diam}(\mathcal{A}_{r_n}(x)\cap \mathsf{S}) > \gamma\big) \;\leqslant\; \mathbf{1}\{\mathsf{E}_{r_n}(\gamma)^c\} \xrightarrow{a.s.} 0. \tag{Б.51}$$

Таким образом, ограничения на r_n для выполнения условия (в) имеют вид:

$$\sum_{n=1}^{\infty} e^{\frac{-c\gamma^4 r_n}{D^4}} \;<\; \infty, \qquad r_n \;\geqslant\; C\,\gamma^{-12}\,\omega^2(K) D^{10}. \tag{Б.52}$$

Следовательно, при случайной инициализации параметров из непрерывного распределения условие (в) выполняется почти наверное при выполнении введённых ограничений на ширину первого слоя.

Так как последующие слои не увеличивают диаметры ячеек, а нейроны первого слоя заморожены, то при верном подборе ширины r_n и инициализации всех параметров из непрерывного распределения условие (в) выполняется почти наверное.

Таким образом, при выполнении ограничений на архитектуру, наложенных в процессе доказательства пунктов (а), (б), (в), теорема Нобеля выполняется, и, следовательно,

$$h_n(x) \xrightarrow{L^2(P_X)} g(x). \tag{Б.53}$$

Откуда, согласно лемм. 3, следует:

$$h_n(X) \xrightarrow{P} g(X). \tag{Б.54}$$

Так как

$$c_n(X) \xrightarrow{P} g(X) \tag{Б.55}$$

и

$$h_n(X) \xrightarrow{P} g(X), \tag{Б.56}$$

имеет место сходимость разности выхода персептрона и гистограммной регрессии:

$$c_n(X) - h_n(X) \xrightarrow{P} 0. \tag{Б.57}$$

Теорема доказана.