

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

На правах рукописи

Перминов Андрей Игоревич

**Модифицированный байесовский классификатор 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 — 2025

## Оглавление

	Стр.
<b>Введение . . . . .</b>	<b>6</b>
<b>Глава 1. Бинарная классификация . . . . .</b>	<b>13</b>
1.1 Модифицированный байесовский классификатор . . . . .	13
1.1.1 Задача классификации и байесовский классификатор . . .	13
1.1.2 Проблематика . . . . .	14
1.1.3 Модификация байесовского классификатора . . . . .	16
1.2 Аппроксимация байесовского классификатора . . . . .	18
1.2.1 Аппроксимация классическими методами . . . . .	19
1.2.2 Аппроксимация равномерно непрерывной функцией . . .	21
1.2.3 Нейросетевая аппроксимация . . . . .	21
1.2.4 Адаптивная гистограммная аппроксимация . . . . .	23
1.3 Объясняющее двоичное дерево eXBTree . . . . .	24
1.3.1 Построение объясняющего дерева решений . . . . .	24
1.3.2 Комбинаторная сложность и практическая реализация eXBTree . . . . .	26
1.3.3 Геометрический анализ построенного дерева . . . . .	28
1.3.4 Анализ прецедентов и локальной уверенности . . . . .	29
1.4 Связь нейросетевой и гистограммной аппроксимаций. Асимптотические свойства гистограммной аппроксимации . . .	30
1.5 Случай нескольких классов . . . . .	31
1.6 Простая линейная атака на персепtron . . . . .	33
1.6.1 Существующие подходы . . . . .	33
1.6.2 Постановка задачи атаки . . . . .	34
1.6.3 Атака на однослойный персепtron . . . . .	35
1.6.4 Атака на многослойный персепtron . . . . .	37
1.6.5 Генерация произвольных входов с заданным выходом .	39
1.6.6 Экспериментальное исследование . . . . .	40
1.6.7 Выводы . . . . .	40
1.7 Применение . . . . .	41
1.7.1 Поведение вне носителя распределения . . . . .	41
1.7.2 Устойчивость . . . . .	42
1.7.3 Противодействие SLAP атаке . . . . .	43

1.7.4	Сопоставление нейросетевой и гистограммной регрессии . . . . .	44
1.7.5	Отказ от распознавания и интерпретация выходов . . . . .	45
1.7.6	Влияние порога доверия на характеристики классификатора . . . . .	45
1.7.7	Выводы . . . . .	46
<b>Глава 2. Унарная классификация</b>		<b>47</b>
2.1	Модель кластеров уровня плотности . . . . .	47
2.2	Нейросетевая регрессия (случай одного класса) . . . . .	48
2.3	Нейросетевая регрессия как оценка апостериорной вероятности класса . . . . .	50
2.4	Случай нескольких классов . . . . .	51
2.5	Преимущества унарной классификации . . . . .	52
2.6	Оценка качества унарных классификаторов . . . . .	53
2.6.1	Мощность классификатора . . . . .	53
2.6.2	Эффективность классификатора . . . . .	54
2.6.3	Мера неразделимости классов . . . . .	55
2.6.4	Визуализация метрик . . . . .	55
2.6.5	Обобщение на многоклассовый случай . . . . .	57
2.7	Иллюстрация работы на модельных примерах . . . . .	57
<b>Глава 3. Применение унарной классификации</b>		<b>60</b>
3.1	Построение репродукционных выборок . . . . .	60
3.1.1	Постановка задачи . . . . .	61
3.1.2	Обучение классификатора . . . . .	62
3.1.3	Создание репродукционных данных . . . . .	62
3.1.4	Экспериментальное исследование . . . . .	63
3.2	Обучение нейросети по некомплектным данным . . . . .	65
3.2.1	Задача классификации данных с пропущенными значениями . . . . .	66
3.2.2	Типы механизмов пропусков . . . . .	67
3.2.3	Существующие методы обработки пропусков . . . . .	67
3.2.4	Ограничения классических методов . . . . .	69
3.2.5	Метод вероятностного заполнения . . . . .	70

3.2.6	Классификация комплектного наблюдения . . . . .	72
3.2.7	Экспериментальное исследование . . . . .	72
<b>Глава 4. Интеллектуальная система машинного обучения для визуализации и исследования методов классификации</b>		<b>76</b>
4.1	Общая характеристика интеллектуальной системы машинного обучения . . . . .	76
4.2	Архитектура и интерфейс интеллектуальной системы . . . . .	77
4.2.1	Архитектура интеллектуальной системы . . . . .	78
4.2.2	Структура интерфейса . . . . .	78
4.2.3	Аппаратные и программные требования . . . . .	79
4.3	Реализованные алгоритмы и методы визуализации . . . . .	80
4.3.1	Многослойный персепtron . . . . .	80
4.3.2	Оптимационные алгоритмы . . . . .	80
4.3.3	Функции потерь . . . . .	81
4.3.4	Объясняющее двоичное дерево . . . . .	81
4.3.5	Визуализация модели и метрик . . . . .	82
4.3.6	Анализ отказов от классификации . . . . .	83
4.3.7	Генерация и модификация данных . . . . .	84
4.4	Структура и функциональные компоненты пользовательского интерфейса . . . . .	85
4.5	Алгоритмы визуализации . . . . .	86
4.6	Примеры использования . . . . .	87
4.6.1	Бинарная классификация . . . . .	87
4.6.2	Унарная классификация . . . . .	88
4.6.3	Создание синтетических данных . . . . .	89
4.6.4	Построение объясняющего дерева решений . . . . .	90
4.7	Роль интеллектуальной системы в исследовании . . . . .	90
<b>Заключение . . . . .</b>		<b>93</b>
<b>Словарь терминов . . . . .</b>		<b>94</b>
<b>Список литературы . . . . .</b>		<b>95</b>

Список рисунков . . . . .	104
Список таблиц . . . . .	106
Приложение А. Свидетельства о государственной регистрации программ и ЭВМ . . . . .	107
Приложение Б. Доказательства теорем . . . . .	112
Б.1 Доказательство теорем. 1 . . . . .	112

## Введение

Одной из ключевых задач в машинном обучении и статистическом распознавании образов является построение классификаторов, обладающих устойчивостью при ограниченных объёмах обучающих данных, а также при наличии существенного дисбаланса классов [1; 2]. Такие условия характерны для множества прикладных сценариев, в том числе в анализе табличных данных невысокой размерности, применяемых в медицинской диагностике, промышленном контроле, финансовой аналитике и других областях, где один из классов либо отсутствует в выборке частично, либо представлен незначительным числом наблюдений. При этом априорные вероятности классов могут значительно различаться, что приводит к деградации точности решений, принимаемых в пользу маломощных распределений. Невозможность обоснованно классифицировать объекты, относящиеся к слабо представленным кластерам, особенно при их близости к границам доминирующего класса, делает задачу построения классификатора в такой постановке математически и вычислительно нетривиальной [3].

Отдельную сложность представляет ситуация, когда тестовые данные выходят за пределы носителя распределения обучающей выборки. В таких случаях отсутствует статистически обоснованная возможность принимать решения с высокой степенью уверенности, и требуется формализованный механизм отказа от классификации. Это особенно важно в задачах, где цена ошибки велика, а распределения данных обладают сложной или разреженной структурой.

Известно, что при высокой априорной вероятности одного из классов даже значительное значение правдоподобия маломощного класса может оказаться недостаточным для принятия в его пользу в рамках классического байесовского подхода [4]. В то же время стандартные процедуры балансировки, такие как редукция выборки доминирующего класса либо синтетическое увеличение мощности маломощных классов (например, путём дублирования наблюдений), как правило, вносят искажения в структуру исходного распределения [5]. Это нарушает статистические предпосылки, лежащие в основе байесовского вывода, и делает полученные модели плохо интерпретируемыми и неустойчивыми при генерализации.

Дополнительным фактором, затрудняющим построение устойчивых моделей, является неполнота данных. В табличных наборах, собираемых в реальных прикладных задачах, часто присутствуют пропуски, возникающие по разным причинам – от случайных ошибок измерений до систематического отсутствия признаков у целых подмножеств объектов. Обработка таких данных требует специальных методов, обеспечивающих корректность статистического вывода и минимизацию потерь информации [6].

Актуальной задачей является построение классификатора, сохраняющего свойства состоятельности и обобщающей способности в условиях ограниченности данных, дисбаланса классов, неполноты наблюдений и возможного выхода входных данных за пределы области, охваченной обучающей выборкой. Дополнительную сложность представляет необходимость получения интерпретируемых и вычислительно эффективных правил принятия решений, особенно при высокой размерности пространства признаков и ограниченной мощности выборки [7]. Для практического применения классификатора важной становится также возможность локализации областей пространства признаков, в которых принимаемые решения обладают высокой степенью надёжности.

Особую актуальность приобретает задача получения состоятельной оценки апостериорной вероятности принадлежности к классу, заданной на компакте в пространстве признаков, без дополнительных предположений о параметрической форме плотностей [8]. В условиях отсутствия полной априорной информации, ограниченного количества обучающих примеров, неполноты данных и неоднородности структуры распределения необходимы методы, обладающие адаптивностью, устойчивостью к локальным вариациям плотности, масштабной инвариантностью и возможностью отказа от классификации в зонах высокой неопределенности. Такие методы должны быть совместимы с современными требованиями вычислительной эффективности, масштабируемости и интерпретируемости [9].

Задача построения универсальных, статистически обоснованных и практически применимых моделей классификации в подобных условиях остаётся открытой и представляет собой предмет активного теоретического и прикладного исследования.

**Целью** данной работы является построение формально обоснованных методов классификации табличных данных, обеспечивающих статистическую состоятельность, устойчивость к дисбалансу классов и некомплектности дан-

ных, а также корректную обработку объектов вне носителя обучающего распределения, на основе модифицированного байесовского классификатора с использованием многослойного персептрона.

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

1. Разработать и реализовать метод построения классификатора, обеспечивающего состоятельную аппроксимацию апостериорных вероятностей при дисбалансе классов и некомплектности данных за счёт использования модифицированного байесовского классификатора на основе многослойного персептрона.
2. Разработать методы генерации синтетических табличных данных и обработки некомплектных табличных данных на основе предложенного метода построения классификатора.
3. Провести экспериментальное исследование разработанных методов на модельных и прикладных данных для оценки устойчивости классификатора к дисбалансу классов, неполноте данных и корректности обработки объектов вне носителя обучающего распределения.
4. Разработать интеллектуальную систему машинного обучения, реализующую предложенные методы и обеспечивающую решение задач классификации табличных данных в условиях дисбаланса классов, некомплектности данных и высокой неопределённости вне носителя распределения.

#### **Основные положения, выносимые на защиту:**

1. Метод построения классификатора, обеспечивающего состоятельную аппроксимацию апостериорных вероятностей при дисбалансе классов и некомплектности данных за счёт использования модифицированного байесовского классификатора на основе многослойного персептрона.
2. Метод создания синтетических табличных данных на основе предложенного метода построения классификатора.
3. Метод обработки некомплектных табличных данных, на основе предложенного метода построения классификатора.
4. Интеллектуальная система машинного обучения, реализующая предложенные методы и обеспечивающая решение задач классификации табличных данных в условиях дисбаланса классов, некомплектности данных и высокой неопределённости вне носителя распределения.

Перечисленные положения относятся к направлениям исследований 4, 7, 8 и 9 паспорта специальности 2.3.5.

**Научная новизна:** разработан метод построения классификаторов табличных данных, обеспечивающий состоятельную аппроксимацию апостериорных вероятностей в условиях дисбаланса классов, неполноты данных и выхода объектов за пределы носителя обучающего распределения. Метод основан на аппроксимации апостериорных вероятностей с использованием многослойного персептрона с кусочно-линейными функциями активации и включает две специализированные версии: бинарную для аппроксимации вероятности между двумя классами и унарную для оценки плотности распределения одного класса с формализованной процедурой отказа от классификации и детектированием объектов вне обучающего распределения. В рамках метода предложены алгоритмы обработки данных с пропусками на основе унарной вероятностной модели и генерации синтетических табличных данных с сохранением статистических характеристик исходного распределения. Дополнительно выполнено развитие теоретических основ построения классификаторов в условиях ограниченного объёма данных и высокой неопределённости, включая формализацию процедур отказа и введение количественных показателей доверия к решениям.

Основные элементы научной новизны состоят в следующем:

1. Установлена асимптотическая эквивалентность между нейросетевой регрессией, построенной с использованием многослойного персептрона, и гистограммной оценкой апостериорной вероятности, что обеспечивает теоретическое обоснование состоятельности предложенного метода.
2. Введён формализованный механизм отказа от классификации, основанный на пороговой оценке аппроксимированной апостериорной вероятности, и предложен способ количественного измерения уровня доверия к решению классификатора в каждом наблюдении признакового пространства.
3. Разработан метод обучения по некомплектным данным, основанный на совместном восстановлении пропущенных значений и обучении унарных регрессионных моделей, что обеспечивает устойчивость к пропускам и позволяет проводить классификацию с использованием частичной информации.
4. Предложен метод создания синтетических табличных данных путём прореживания равномерного фонового распределения с использовани-

ем адаптивной плотностной аппроксимации, полученной в результате унарной классификации, с доказанной состоятельностью соответствующих оценок.

5. Показано, что предложенный классификатор обладает линейной сложностью по числу объектов выборки при применении, что обеспечивает его пригодность для обработки больших табличных наборов данных в реальном времени.

### **Теоретическая и практическая значимость**

Теоретическая значимость работы состоит в развитии математических основ байесовской классификации в условиях дисбаланса классов, ограниченного объёма выборки, отсутствующих данных и выхода за пределы носителя обучающего распределения. Установлена асимптотическая связь между нейросетевой и гистограммной оценками апостериорной вероятности, обеспечивающая обоснование состоятельности предложенного подхода. Введён формализованный критерий отказа от классификации, основанный на пороговой аппроксимации апостериорной вероятности, и предложена процедура количественной оценки уровня доверия к принимаемым классификатором решениям. Разработанный подход расширяет теоретическую базу классификации и дополняет существующие модели нелинейной регрессии в вероятностном контексте, обеспечивая строгие условия применимости и гарантии корректности работы в зонах высокой неопределенности.

Практическая значимость заключается в возможности применения разработанного метода к задачам анализа табличных данных в условиях ограниченной или неполной информации. Предложенный классификатор позволяет осуществлять устойчивую и интерпретируемую классификацию в задачах с несбалансированными классами и повышать надёжность принимаемых решений за счёт автоматического отказа от распознавания в недостоверных областях признакового пространства. Разработаны методы генерации синтетических данных, сохраняющих структуру исходного распределения, а также методы обработки пропущенных значений на основе регрессионных унарных моделей. Реализован программный комплекс, обеспечивающий воспроизводимое и наглядное применение предложенного метода в задачах предварительного анализа классификации и восстановления табличных данных, а также оценке уровня доверия обученных моделей.

**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 8 печатных изданиях, 4 из которых изданы в журналах, рекомендованных ВАК, 1 — в периодических научных журналах, индексируемых Web of Science и Scopus, 3 — в тезисах докладов. Зарегистрированы 4 программы для ЭВМ.

**Объём и структура работы.** Диссертация состоит из введения, 4 глав, заключения и 2 приложений. Полный объём диссертации составляет 112 страниц, включая 33 рисунка и 4 таблицы. Список литературы содержит 92 наименования.

## Глава 1. Бинарная классификация

### 1.1 Модифицированный байесовский классификатор

#### 1.1.1 Задача классификации и байесовский классификатор

Одной из задач, решаемых с помощью методов машинного обучения с учителем, является классификация [10]. 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  $f: [0, 1]^d \rightarrow \{\pm 1\}$ , которая значениям признаков ставит в соответствие метки классов.

Общую задачу классификации можно записать в следующем виде:

$$\mathbb{P}(Y \neq f(X)) \rightarrow \min_f, \quad (1.1)$$

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

$$\mathbb{E} (Y - f(X))^2 \rightarrow \min_f, \quad (1.2)$$

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

Решение задачи регрессии – это условное математическое ожидание

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

как следует из элементарного соотношения

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

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 [11]

$$s(x) = \begin{cases} 1, & \text{если } g(x) > 0 \text{ и } x \in \mathbb{S}, \\ \text{любое из значений } \pm 1, & \text{если } g(x) = 0 \text{ или } x \notin \mathbb{S}, \\ -1, & \text{если } g(x) < 0 \text{ и } x \in \mathbb{S}, \end{cases} \quad (1.3)$$

отвечающий  $g$  (данной версии условного математического ожидания). Последнее следует из того, что для  $f$  со значениями  $\pm 1$  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 (1.3), 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 (1.1) и (1.2) с регуляризацией [12] и различными ограничениями на классы функций  $f$ , по которым ведётся оптимизация.

### 1.1.2 Проблематика

Ключевая трудность, с которой сталкиваются методы машинного обучения [13], 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 [14]. Однако эти методы, как правило, требуют задания гиперпараметров [15] 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 [16]. Существующие подходы к детекции подобных случаев делятся на три класса: статистические методы [17], моделирование сдвигов [18] и применение вспомогательных моделей машинного обучения [19]. 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  $\mathbb{S}$ .

### 1.1.3 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 [20].

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  $\lambda$  – 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  $P_\alpha$ , а через  $\mathbb{S}$  – носитель распределения  $P_X$ .

В силу разложения Лебега и теоремы Радона–Никодима всегда найдутся неотрицательная интегрируемая функция  $\rho$  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  $g_\alpha$  задачи регрессии

$$\mathbb{E}_\alpha (Y - f(X))^2 \rightarrow \min_f \quad (1.4)$$

существует, это решение единственно  $P_X$ - и  $\lambda$ -н. и может быть задано формулой

$$g_\alpha(x) = \begin{cases} g(x), & \text{если } x \in A, \\ \frac{(1-\alpha)g(x)\rho(x)}{\alpha + (1-\alpha)\rho(x)}, & \text{если } \rho(x) > 0 \text{ и } x \in \mathbb{S} \setminus A, \\ 0, & \text{если или } \rho(x) = 0 \text{ и } x \in \mathbb{S} \setminus A, \text{ или } x \notin \mathbb{S}, \end{cases} \quad (1.5)$$

здесь минимум берется по всем (борелевским) функциям  $f$  и

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

При этом классификатор  $s_\alpha = s_\alpha(x)$ ,  $x \in [0, 1]^d$ , заданный формулой (1.3) с заменой  $g$  на любое решение  $g_\alpha$  задачи (1.4) на  $\mathbb{S}$  на  $[0, 1]^d$ , обладает следующими свойствами:

- $s_\alpha$  реализует минимум в задаче классификации

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

где минимум берется по всем (борелевским) функциям со значениями  $\pm 1$ ;

- зоной неопределённости  $s_\alpha$  является множество  $\{x \in [0, 1]^d : g_\alpha(x) = 0\}$ , которое покрывает  $\lambda$ -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 (1.4) рассматривается задача вида

$$\mathbb{E}_\alpha (Y - f(X))^2 + Pen(f) \rightarrow \min_{f \in \mathcal{F}}, \quad (1.6)$$

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 (1.6) будет эквивалентна задаче приближения  $g_\alpha$  – любого решения задачи (1.4) – функцией из класса  $\mathcal{F}$  с учётом штрафа  $Pen(f)$ :

$$(1 - \alpha) \|g_\alpha - f\|_{P_X}^2 + \alpha \|g_\alpha - f\|_\lambda^2 + Pen(f) \rightarrow \min_{f \in \mathcal{F}}, \quad (1.7)$$

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

Как было отмечено ранее (см. раздел 1.1.1), 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 (1.6), решаемый с использованием методов машинного обучения.

Поскольку в эмпирической постановке оценка функции принятия решения подвержена статистическим флюктуациям, 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 [21; 22]. Отказ от принятия решения осуществляется для тех наблюдений, по которым оценка эмпирической функции  $f$ , соответствующей приближённому решению задачи (1.6), по модулю не превосходит  $\beta$ :  $|f(x)| \leq \beta$ .

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

Параметр  $\beta$  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.

## 1.2 Аппроксимация байесовского классификатора

Для построения классификатора, приближающего оптимальное байесовское правило, предполагается наличие размеченного обучающего набора  $(X, Y) = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ , 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.

### 1.2.1 Аппроксимация классическими методами

Одним из подходов к приближению байесовского классификатора является использование классических непараметрических и параметрических методов [23; 24]. Эти методы позволяют получить приближение к оптимальной функции принятия решения, не предполагая явной формы распределения данных, и часто служат базой для анализа свойств более сложных моделей.

Наиболее простым из них выступает построение гистограммы [25]. 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:

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

где  $\mathcal{N}_k(x)$  – множество индексов  $k$  ближайших к  $x$  точек в обучающей выборке. Метод обладает асимптотической состоятельностью [26] при  $k \rightarrow \infty$  и  $k/n \rightarrow 0$ , но на практике чувствителен к выбору метрики и параметра  $k$ . В случае сложных или структурированных признаков, определение подходящей метрики может быть затруднено или неочевидно, что мотивирует переход к методам параметрической аппроксимации.

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:

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

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 [27; 28].

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 [29]. В линейном случае SVM решает задачу максимизации зазора между классами:

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

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 [30].

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

### 1.2.2 Аппроксимация равномерно непрерывной функцией

Пусть  $c(X) : [0, 1]^d \rightarrow \mathbb{R}$  – равномерно непрерывная функция на  $[0, 1]^d$ .

Рассмотрим задачу среднеквадратичной аппроксимации:

$$\mathbb{E} (c(X) - Y)^2 \rightarrow \min_{c(X)}. \quad (1.8)$$

Поскольку

$$\begin{aligned} \mathbb{E} (c(X) - Y)^2 &= \mathbb{E} (c(X) - g(X) + g(X) - Y)^2 \rightarrow \\ \mathbb{E} (c(X) - Y)^2 &= \mathbb{E} (c(X) - g(X))^2 + \mathbb{E} (g(X) - Y)^2 \end{aligned}$$

и второе слагаемое не зависит от  $c(X)$ , задача (1.8) сводится к аппроксимации функции регрессии:

$$\mathbb{E} (c(X) - g(X))^2 \rightarrow \min_{c(X)}, \quad (1.9)$$

### 1.2.3 Нейросетевая аппроксимация

Возьмём в качестве  $c(X)$  многослойный персепtron [31] (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 1.1) в каждом и выходным слоем из одного нейрона. По основной аппроксимационной теореме [32] для любого заданного  $\varepsilon > 0$  существуют такие значения параметров персептрана  $L$  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  $\varepsilon$ -приближенное решение задачи (1.9) существует.

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

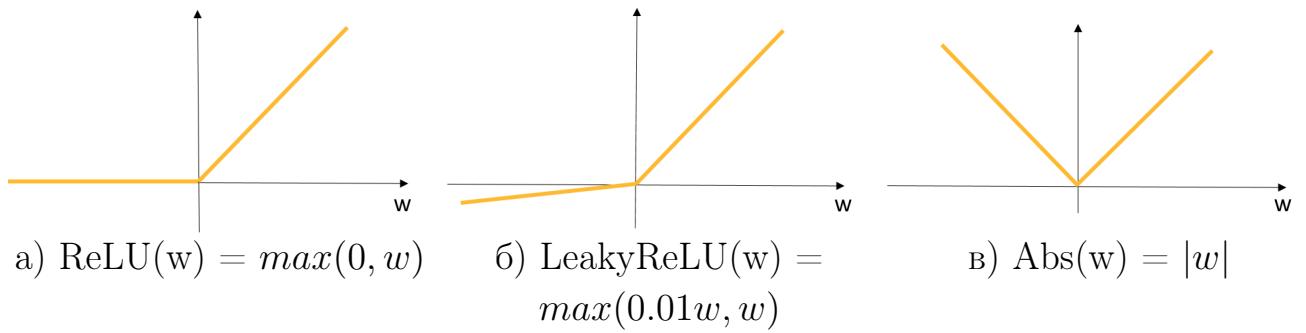


Рисунок 1.1 — Кусочно-линейные функции активации

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

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  $\{(X_{n+1}, Y_{n+1}), \dots, (X_{2n}, Y_{2n})\}$ , где векторы  $\{X_{n+1}, \dots, X_{2n}\}$  – 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 персепtronов  $c(X)$  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 [33], генетический алгоритм [34] и т.д.), построим выборочную оценку решения задачи (1.9):

$$\sum_{i=1}^{2n} (c_n(X_i) - Y_i)^2 \rightarrow \min_{c_n(X) \in C(L, k)}, \quad (1.10)$$

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

Пусть функция  $c_n^*(X)$  – решение оптимизационной задачи (1.10), которая в дальнейшем будет называться **функцией нейросетевой регрессии**. Соответствующий этому решению персепtron строит иерархическое (по слоям) разбиение компакта  $[0, 1]^d$  на  $O(k^{dL})$  непересекающихся ячеек [8] (при  $k > d$ ).

Пример такого разбиения показан на рисунке 1.2, где персепtron имеет  $L = 2$  скрытых слоя по  $k = 6$  нейронов.

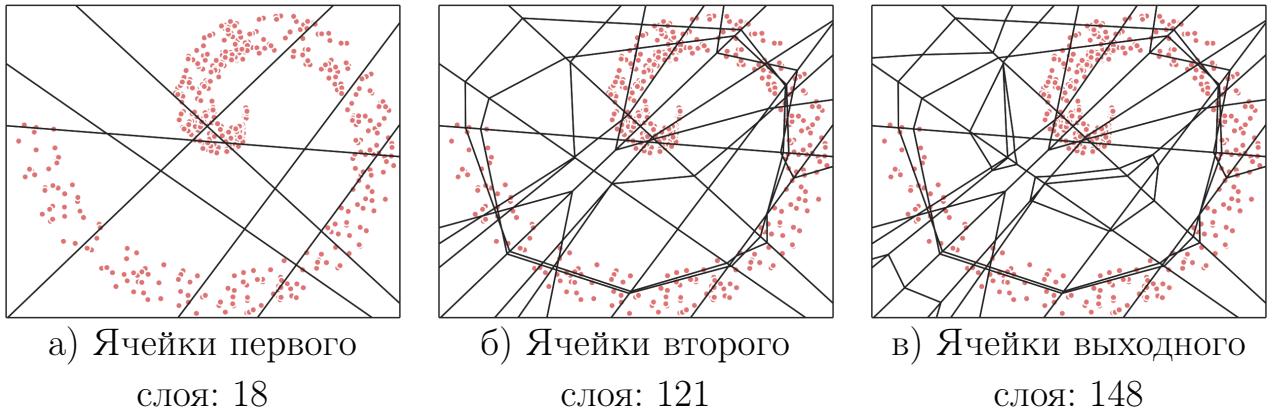


Рисунок 1.2 — Пример разбиения некоторым персепtronом с  $L = 2$ ,  $k = 6$

#### 1.2.4 Адаптивная гистограммная аппроксимация

Пусть в результате построения  $c_n^*(X)$  получено разбиение компакта  $[0, 1]^d$  на  $N$  непересекающихся ячеек  $\{K_1, K_2, \dots, K_N\}$ . 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:

$$\sum_{i=1}^{2n} (h_n(X_i) - Y_i)^2 \rightarrow \min_{h_n(X)}, \quad (1.11)$$

Пусть  $X \in K_r$ . Тогда задачу (1.11) для этой ячейки можно представить в следующем виде:

$$n_{-1}(X) \cdot (h_n(X) + 1)^2 + n_0(X) \cdot (h_n(X) - 0)^2 + n_{+1}(X) \cdot (h_n(X) - 1)^2 \rightarrow \min_{h_n(X)}, \quad (1.12)$$

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

После дифференцирования функции (1.12) по  $h_n(X)$  получаем решение задачи (1.11):

$$h_n^*(X) = \frac{n_{+1}(X) - n_{-1}(X)}{n_{-1}(X) + n_0(X) + n_{+1}(X)}. \quad (1.13)$$

Пример вычисления функции гистограммной регрессии показан на рисунке 1.3.

**ЗАМЕНИТЬ РИСУНОК**

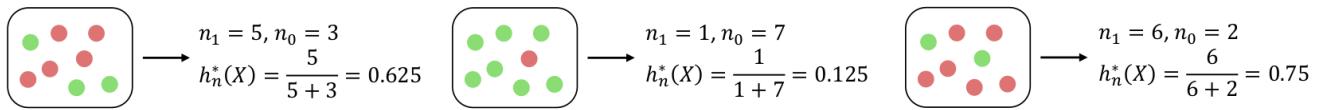


Рисунок 1.3 — Пример вычисления  $h_n^*(X)$  в некоторой ячейке  $K_r$

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.

### 1.3 Объясняющее двоичное дерево eXBTee

#### 1.3.1 Построение объясняющего дерева решений

Как было сказано в разделе 1.2.3, многослойный персепtron с кусочно-линейной функцией активации разбивает входное пространство признаков на  $N$  непересекающихся ячеек  $\{K_1, K_2, \dots, K_N\}$ . В каждой такой ячейке значение выходного нейрона определяется фиксированной линейной комбинацией признаков.

Рассмотрим полносвязный персепtron с  $L$  скрытыми слоями по  $k$  нейронов в каждом. В качестве функции активации используется модуль:

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

Обозначим выходы нейронов первого скрытого слоя через  $A_i$ , второго – через  $B_i$ , третьего – через  $C_i$ , и так далее (рисунок 1.4). Для произвольного нейрона слоя  $A$  выполняется:

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

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

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

и так далее до выходного слоя:

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

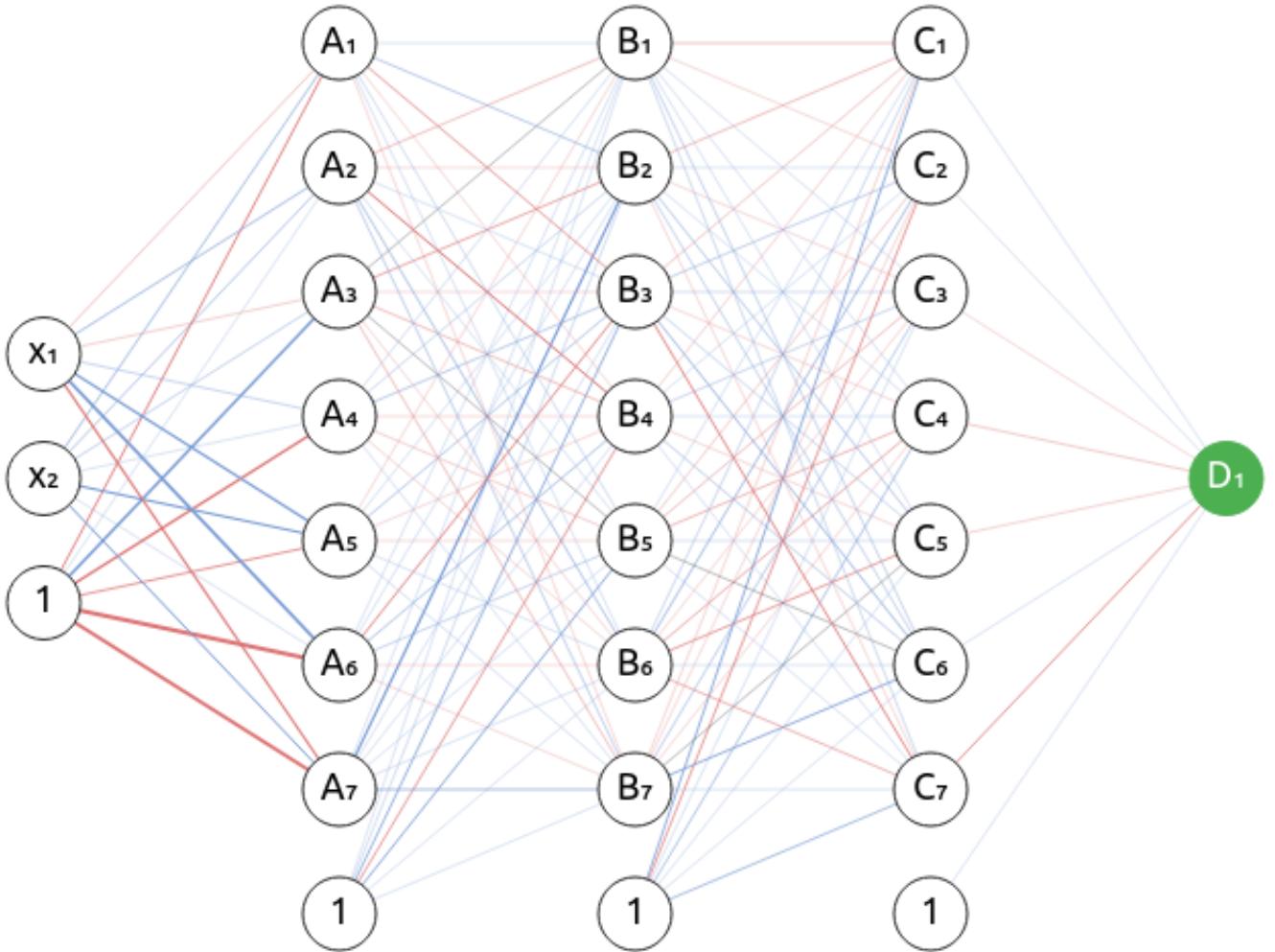


Рисунок 1.4 — Архитектура многослойного персептрона с  $d = 2$ ,  $L = 3$ ,  $k = 7$

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»). Таким образом, на каждом шаге можно заменить модуль линейным выражением с соответствующим знаком.

Пошагово раскрывая модули 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 [35], 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 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 1.5).

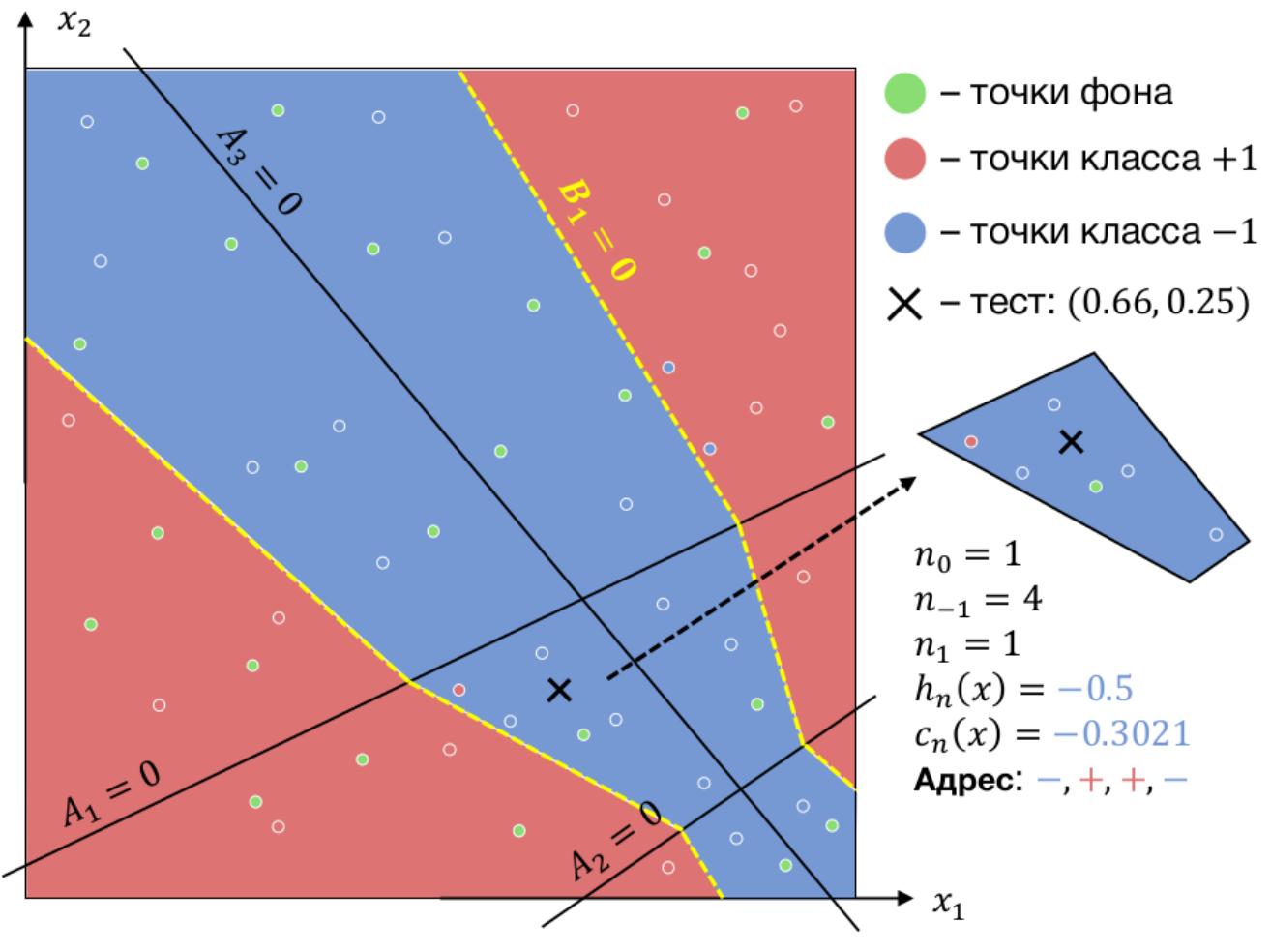
Наличие такой структуры позволяет не только интерпретировать классификацию конкретного наблюдения (путь через дерево), но и формировать иерархию разбиений пространства, объединяя соседние ветви дерева на разных уровнях. Это открывает возможности для оценки доверия и анализа прецедентов – как в пределах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

### 1.3.2 Комбинаторная сложность и практическая реализация eXBTree

**Теорема 2** (О сложности построения дерева по многослойному персептрону). *Пусть многослойный персепtron состоит из  $L$  скрытых слоёв по  $k$  нейронов в каждом ( $k > d$ , где  $d$  – размерность входного пространства). Тогда временная сложность построения двоичного дерева решений по персептрону составляет  $\mathcal{O}(k^{dL})$ .*

#### ДОКАЗАТЕЛЬСТВО

Тем не менее, в рамках задач классификации или аппроксимации нас интересуют 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.



$$\begin{aligned}A_1 &= -0.4x_1 + 0.8x_2 \\A_2 &= -1.2x_1 + 1.8x_2 + 0.8 \\A_3 &= -1.5x_1 - 1.3x_2 + 1.5\end{aligned}$$

$$\begin{aligned}A_1(x) &= -0.064 < 0 \rightarrow - \\A_2(x) &= 0.458 > 0 \rightarrow + \\A_3(x) &= 0.185 > 0 \rightarrow +\end{aligned}$$

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

Рисунок 1.5 — Пример eXBTree на основе персептрана с  $d = 2$ ,  $L = 1$ ,  $k = 3$

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.

**Теорема 3** (О временной сложности получения решения по построенному дереву). *Временная сложность получения ответа по построенному дереву решения совпадает со сложностью применения многослойного персептрана и составляет  $\mathcal{O}(kd + Lk^2)$ .*

## ДОКАЗАТЕЛЬСТВО

### 1.3.3 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 1.2.4. Эти гистограммы позволяют оценивать апостериорные вероятности классов в пределах каждой ячейки и выявлять области с высокой или низкой степенью уверенности модели.

Полученное дерево может быть также рассмотрено как дерево решений с линейными функциями разделения в узлах [36], 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 [37]. 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.

### 1.3.4 Анализ прецедентов и локальной уверенности

Для повышения интерпретируемости построенного дерева и повышения доверия пользователя к решению 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.

## 1.4 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,  $h_n^*(X)$  соответствует разности логарифмов аппроксимированных правдоподобий классов и служит приближением разности апостериорных вероятностей.

Предполагая, что плотности распределения классов равномерно непрерывны, можно показать, что гистограмма является строго состоятельной оценкой этих плотностей. Результаты работы [8] позволяют утверждать, что при росте объёма обучающей выборки  $n \rightarrow \infty$  и одновременном увеличении числа ячеек (или, эквивалентно числа нейронов  $kL \rightarrow \infty$ , отвечающего за глубину разбиения пространства) имеет место следующее соотношение:

$$\mathbb{E} (h_n^*(X) - c_n^*(X))^2 \rightarrow 0, \quad n \rightarrow \infty, \quad (1.14)$$

где  $c_n^*(x)$  – выход модифицированного персептрана, аппроксимирующего байесовскую границу принятия решения.

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 персептрана не требуется хранение всей обучающей выборки или экспоненциально большого числа ячеек.

Следствием приведённого утверждения является практический критерий принятия решения на основе выхода персептрана. Поскольку  $c_n^*(x)$  приближает  $h_n^*(x)$ , то в условиях асимптотической сходимости разумно вводить порог отказа  $\beta$  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.

## 1.5 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 класса [38]. Переход от бинарной к многоклассовой классификации существенно усложняет как построение, так и интерпретацию модели [39].

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 [40].

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 [41]: 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 остройми при наличии несбалансированных классов и сложной структуры признакового пространства. Эти соображения подводят к необходимости переосмыслиния самой постановки задачи классификации, особенно в ситуациях, когда интерес представляет лишь один или малое число целевых классов, а остальные данные играют вспомогательную роль.

## 1.6 Простая линейная атака на персептрон

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. В рамках настоящей работы предложен новый подход к формированию таких примеров, получивший название “простая линейная атака на персептрон” (SLAP, Simple Linear Attack for Perceptron). Подробности методики опубликованы в авторской статье [42].

### 1.6.1 Существующие подходы

Наиболее распространённые методы формирования атакующих примеров основываются на градиентной оптимизации. В частности, метод FGSM (Fast Gradient Sign Method) [43] и метод проецированного градиентного спуска PGD (Projected Gradient Descent) [44] 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, [45; 46]. 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 прежде всего 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.

### 1.6.2 Постановка задачи атаки

**ЗАМЕНИТЬ  $n$  НА  $d$  и  $m$  на  $C$  поправить рисунки**

Пусть имеется обученный персепtron  $c(x)$ , принимающий на вход вектор  $x \in \mathbb{R}^d$  и возвращающий вектор выходных значений  $y \in \mathbb{R}^C$ , соответствующих  $C$  ( $C < d$ ) классам. Обозначим через  $x_t$  целевой пример (рисунок 1.6а), на который должна быть «перенесена» классификация, и через  $x_a$  – пример, который подвергается атаке (рисунок 1.6б). Требуется построить новый вектор  $x$  (рисунок 1.6в), близкий к  $x_a$ , но классифицируемый так же, как  $x_t$ . Формально, задача формулируется как:

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

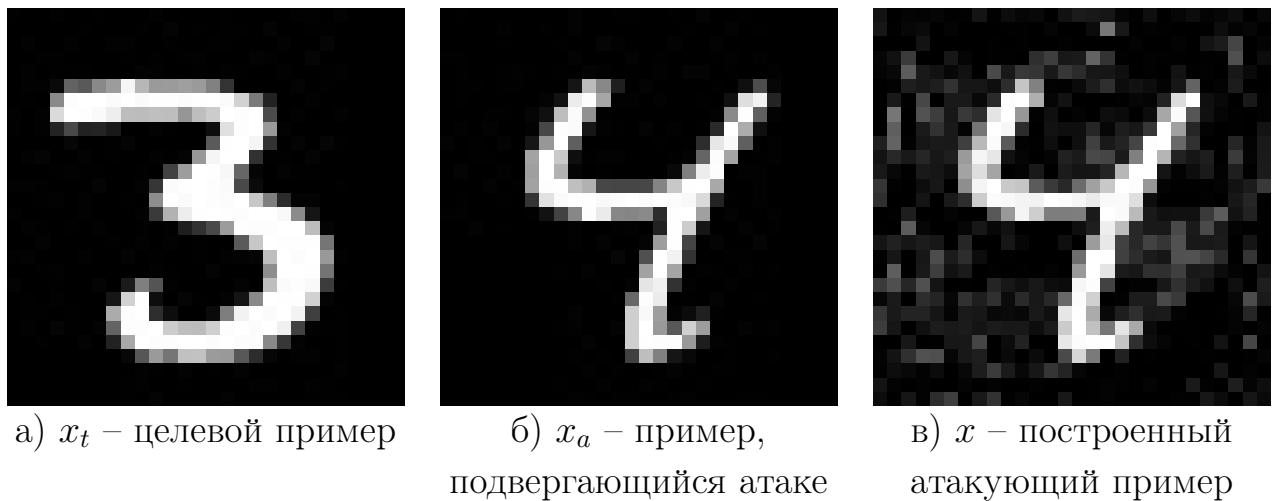


Рисунок 1.6 – Примеры, участвующие в атаке на многослойный персепtron

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

### 1.6.3 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. 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 1.7).

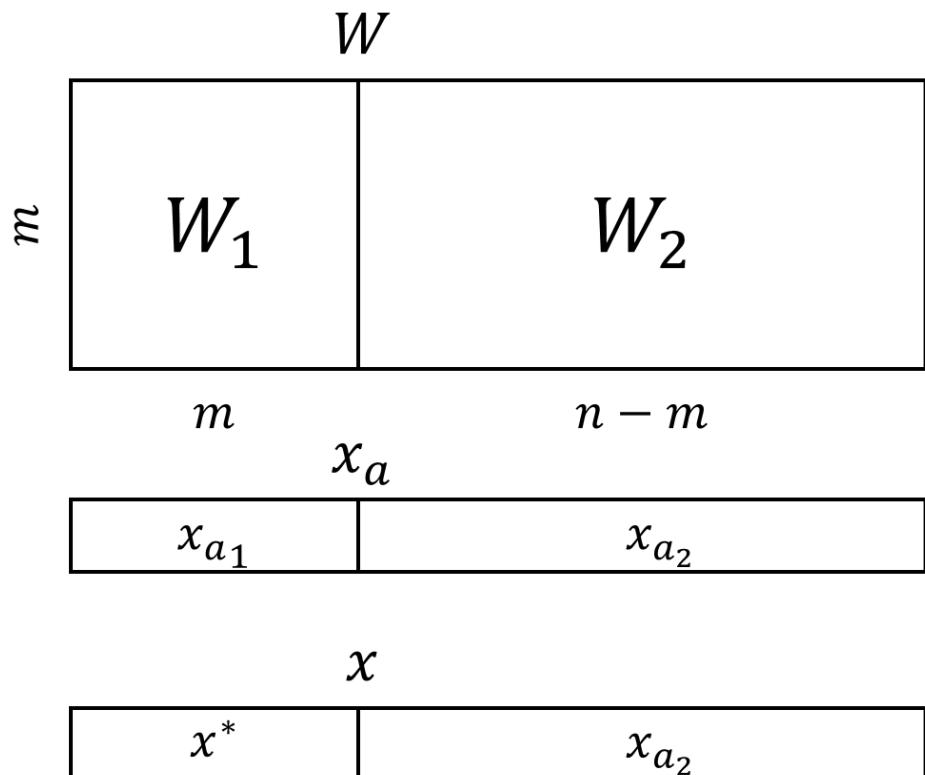


Рисунок 1.7 — Схема матричной атаки

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 1.8).



Рисунок 1.8 — Пример матричной атаки,  $x \in [-1055, 926]$

### С учётом ограничений

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

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

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

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

При невозможности точного воспроизведения  $y_t$  возможно ослабление условий за счёт введения допусков  $\varepsilon$ :

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

Эти неравенства легко переписываются в канонической форме для QP-решателей. Пример применения данного вида атаки приведён на рисунке 1.9.



Рисунок 1.9 – Пример QP атаки

#### 1.6.4 Атака на многослойный персепtron

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  $x \geq 0$  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:

$$\begin{cases} W_1x + b_1 \geq 0 \\ W_2W_1x + W_2b_1 + b_2 \leq 0 \\ y = -(W_3W_2W_1x + W_3W_2b_1 + W_3b_2) + b_3 \end{cases}$$

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:

$$\begin{cases} W_{21} = W_2W_1 \\ b_{21} = W_2b_1 + b_2 \\ W_{321} = -W_3W_2W_1 \\ b_{321} = b_3 - W_3W_2b_1 - W_3b_2 \\ W_1x + b_1 \geq 0 \\ W_{21}x + b_{21} \leq 0 \\ y = W_{321}x + b_{321} \end{cases}$$

Пример атаки на многослойный персепtron представлен на рисунке 1.10.



а)  $x_t$  – целевой пример



б)  $x_a$  – пример,  
который подвергается  
атаке



в)  $x$  – построенный  
атакующий пример

Рисунок 1.10 – Пример атаки на многослойный персепtron на датасете Cat-vs-Dog [47]

### 1.6.5 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 1.11 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.

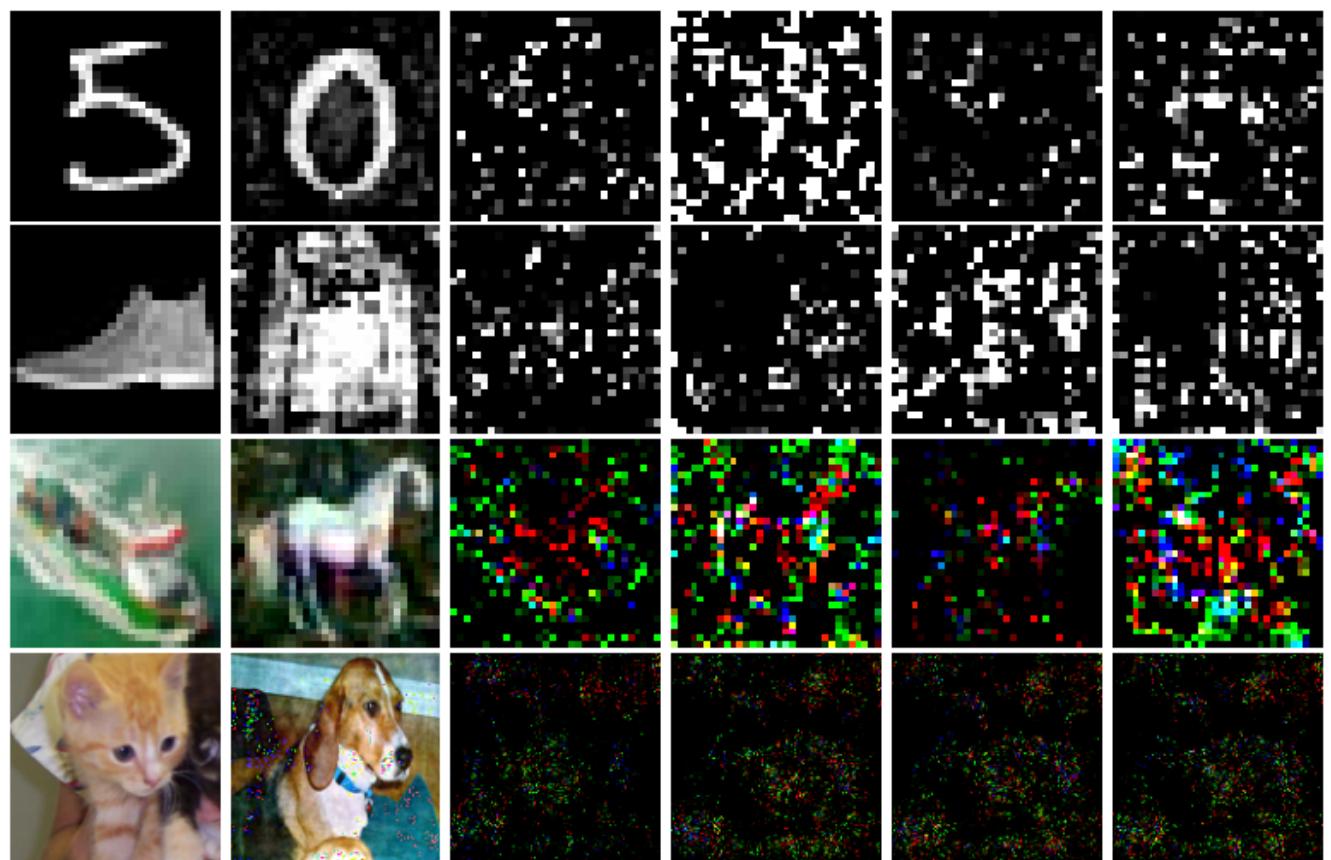


Рисунок 1.11 — Пример генерации атакующих примеров

### 1.6.6 Экспериментальное исследование

Алгоритм был реализован и протестирован на простых персептронах, обученных на датасетах MNIST [48] и CIFAR-10 [49]. 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  $\ell_\infty$ -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	
			$\ell_\infty$	Accuracy	$\ell_\infty$	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.

### 1.6.7 Выводы

Предложенный метод демонстрирует возможность построения состязательных примеров для нейросетевых моделей с кусочно-линейными активациями без использования градиентной информации. Использование методов линейной алгебры позволяет получать как точные, так и приближённые решения с учётом ограничений на значения входных признаков. Атака легко

адаптируется к многослойным моделям и может применяться не только к полносвязным сетям, но и к свёрточным архитектурам, обладающим аналогичной линейной структурой.

## 1.7 Применение

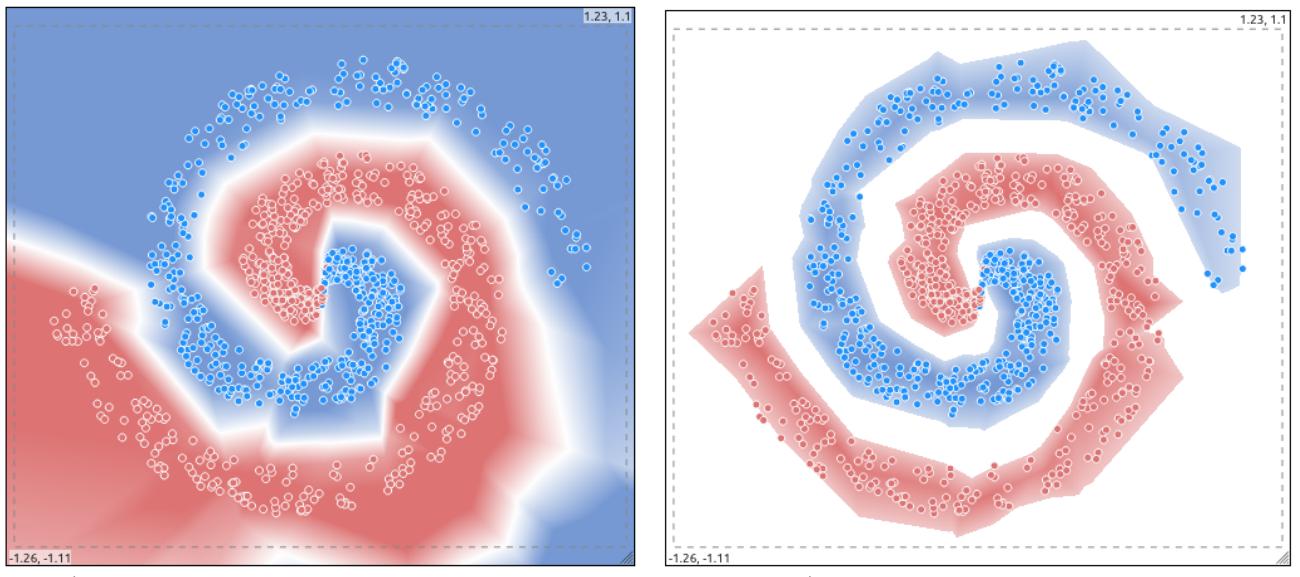
Рассмотренные в предыдущих разделах методы бинарной классификации позволяют успешно решать задачи разделения двух классов на компакте признакового пространства. В данном разделе рассматриваются примеры, иллюстрирующие особенности работы моделей, использующих описанную в разделе 1.1.3 модификацию, а также проблемы, которые такая модификация позволяет эффективно решать.

### 1.7.1 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.

Рассмотрим демонстрационный пример с двумя классами, заданными в виде спиралей на двумерной плоскости. На рисунке 1.12а 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 1.2.3 модифицированную процедуру, то классификатор начинает учитывать общую структуру распределения данных и классифицирует “внешние“ точки как фон (рисунок 1.12б). Это значительно повышает надёжность предсказаний и позволяет говорить о появлении эффекта отказа от распознавания вне носителя распределения.



а) классический классификатор

б) модифицированный  
классификатор

Рисунок 1.12 — Сравнение поведения классификаторов вне носителя

### 1.7.2 Устойчивость

Модели, обучаемые без использования фона, оказываются чрезвычайно чувствительными к отдельным аномальным точкам. Добавление даже одной точки может радикально изменить форму решающего правила (рисунок 1.13а). Это явление лежит в основе так называемых backdoor-атак [50], когда намеренно добавленные в обучающую выборку точки провоцируют нежелательное поведение модели в заранее заданной области.

Добавление фона значительно снижает эффект подобных атак (рисунок 1.13б). 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.

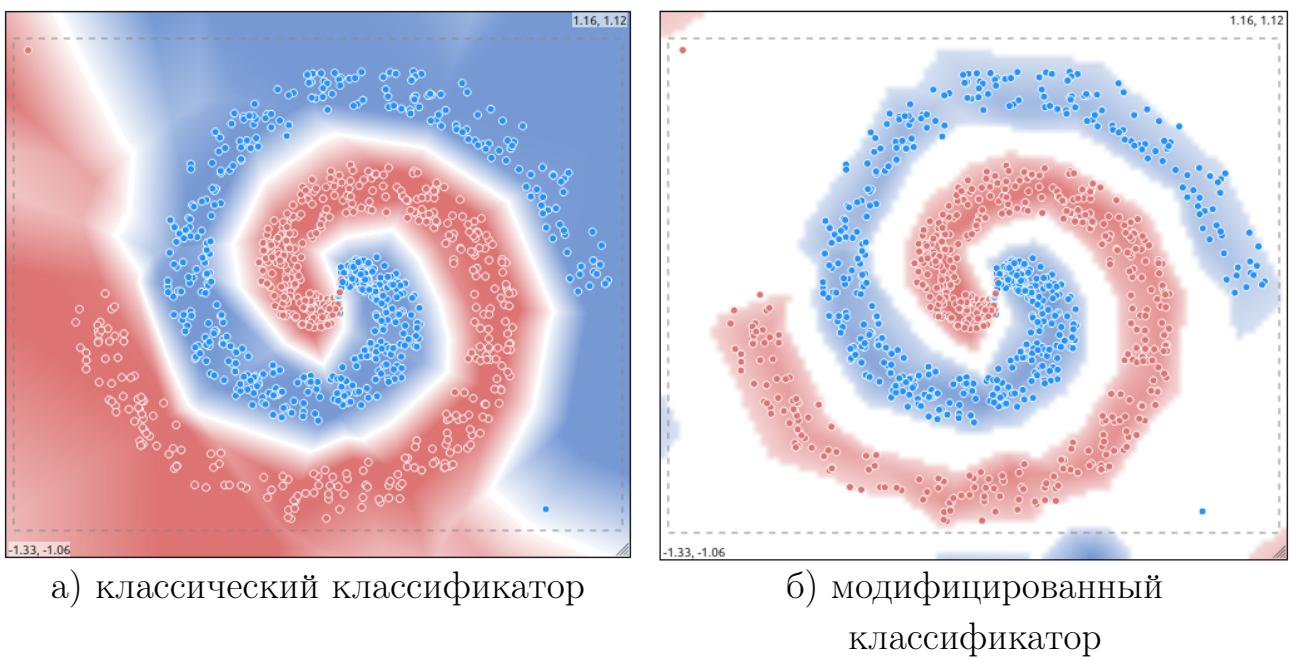


Рисунок 1.13 — Сравнение устойчивости классификаторов к backdoor-атаке

### 1.7.3 Противодействие SLAP атаке

Одним из наиболее уязвимых мест классических классификаторов является их поведение на границе принятия решений. Это свойство активно используется в рамках предложенной в разделе 1.6 атаки SLAP.

Модифицированная процедура классификации, использующая фоновый класс, существенно снижает эффективность подобного рода атак. На рисунке ?? представлены примеры атак, полученных с использованием метода SLAP как для классической модели, так и для модели с фоном. При этом использование модифицированной процедуры обучения приводит к следующим эффектам:

- во многих случаях атака не удаётся;
- при отсутствии ограничений на допустимую область признаков атакующие точки часто выходят за пределы компакта, что приводит к отказу от их распознаванию;
- при успешной атаке полученные точки, как правило, попадают в области с низким уровнем доверия, что также приводит к отказу от распознавания.

Таким образом, модификация классификатора с введением фонового класса позволяет эффективно нейтрализовать атаку SLAP, построенную без

использования градиентной информации и не зависящую от конкретной архитектуры модели.

#### 1.7.4 Сопоставление нейросетевой и гистограммной регрессии

В разделе 1.2.4 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 1.14). Ячейки гистограммы, на которые разбивает пространство персепtron, 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 мягкую версию гистограммной аппроксимации, реализуемую при помощи кусочно-линейных функций.

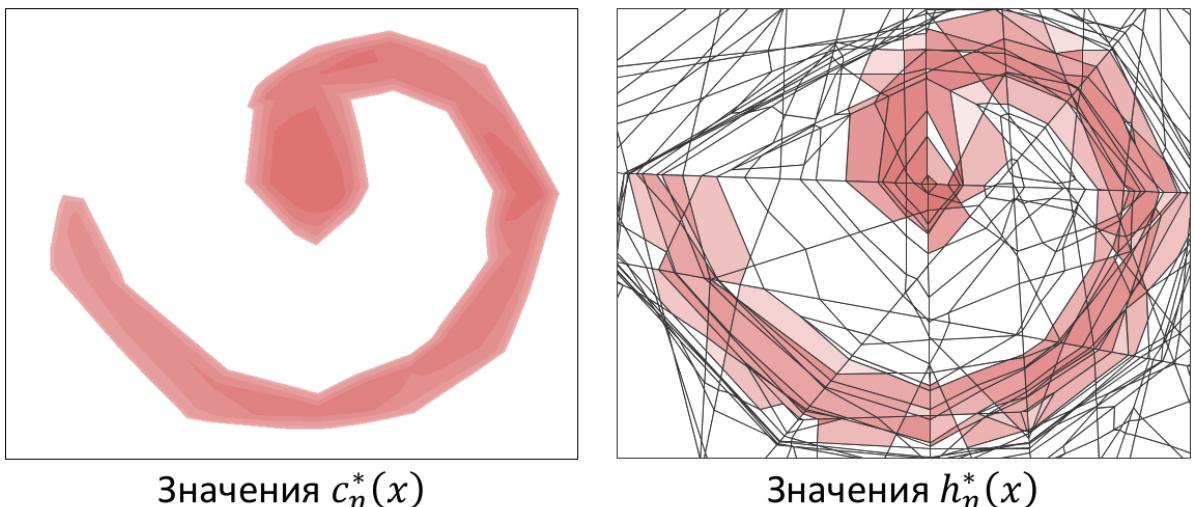


Рисунок 1.14 — Визуальное сравнение функций нейросетевой и гистограммной регрессий

### 1.7.5 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.

### 1.7.6 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  $\beta$  используется для принятия решения о классификации наблюдения: если значение выхода модели по модулю не превышает  $\beta$ , классификатор воздерживается от принятия решения, т.е. формирует отказ от распознавания.

Введение порога  $\beta$  позволяет контролировать баланс между полнотой и надёжностью классификационных решений. При низких значениях  $\beta$  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  $\beta$  ведёт к росту количества отказов от распознавания, но одновременно повышает достоверность решений по тем наблюдениям, для которых классификация всё же производится.

На рисунке 1.15 приведена визуализация результатов классификации при различных значениях порога  $\beta$ : от 0 (классификация осуществляется по всем наблюдениям) до 0.5 (классификатор выдаёт решение только в случаях высокой уверенности). Видно, что при увеличении  $\beta$  область отказов расширяется (обозначена белым цветом), что соответствует желаемому поведению системы в условиях ограниченной уверенности модели.

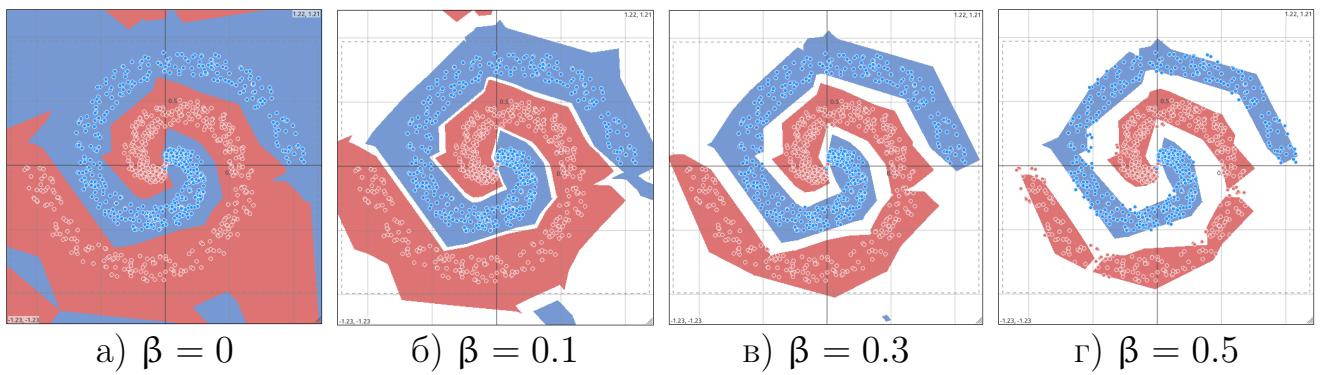


Рисунок 1.15 — Влияние порога доверия  $\beta$  на пространственное распределение классификационных решений

### 1.7.7 Выводы

Модификация бинарного классификатора позволяет существенно улучшить устойчивость и надёжность модели, основанной на многослойном персептроне. Примеры наглядно показывают, что обучение на расширенном наборе данных с добавлением фона позволяет реализовать отказ от распознавания вне носителя данных, повысить устойчивость к некоторым видам атак, а также сделать предположение о связи между функциями нейросетевой и гистограммной регрессии.

## Глава 2. Унарная классификация

### 2.1 Модель кластеров уровня плотности

Одним из фундаментальных подходов к выявлению скрытой структуры в данных является использование модели кластеров уровня плотности. Этот подход берёт своё начало в работах [51] и [52], в которых было предложено рассматривать кластеры как области пространства признаков, характеризующиеся повышенной плотностью вероятностного распределения. Интуитивно предполагается, что наблюдения, принадлежащие к одному кластеру, с большей вероятностью сосредоточены в определённой области пространства, в то время как между кластерами плотность стремится к более низким значениям.

В дальнейшем, развитие этой идеи получило формальное обоснование в рамках непараметрических методов оценивания плотности. В частности, в работах [53] и [54] были представлены строгие доказательства состоятельности таких оценок, как гистограмма, оценка по методу  $k$  ближайших соседей, а также ядерная оценка. Эти результаты впоследствии были систематизированы и обобщены в их монографии по непараметрической статистике, ставшей классической.

На основе этих теоретических результатов в [55] предложили алгоритм кластеризации, основанный на выделении областей высокой плотности, который был реализован в рамках иерархического подхода. Этот метод, получивший название вероятностного метода, позволял строить деревья кластеров на основе последовательного объединения областей с близкими характеристиками плотности. К сожалению, из-за высокой вычислительной сложности, связанной с необходимостью многократного расчёта расстояний между точками и хранения оценок плотности в памяти, его практическое применение оказалось ограниченным малыми объёмами выборок и невысокой размерностью пространства признаков.

Формально, пусть  $f(X)$  – плотность распределения случайного вектора  $X \in \mathbb{R}^d$ . Для любого порогового значения  $c > 0$  вводится **множество уровня плотности**:

$$B(c) = \{X \in \mathbb{R}^d : f(X) > c\}.$$

Это множество представляет собой объединение всех точек, в которых плотность превышает заданное значение  $c$ . Модель кластеров уровня плотности предполагает, что каждый кластер соответствует одной связной компоненте множества  $B(c)$ :

$$B(c) = \bigcup_{i=1}^M B_i(c),$$

где  $B_1(c), B_2(c), \dots, B_M(c)$  – непересекающиеся связные компоненты, каждая из которых интерпретируется как отдельный кластер.

Данный подход позволяет не задавать количество кластеров заранее, поскольку число связных компонент может изменяться в зависимости от значения  $c$ . При больших значениях  $c$  в множестве  $B(c)$  остаются лишь точки, расположенные в наиболее плотных участках пространства, а при уменьшении  $c$  связные области начинают расширяться и объединяться, формируя иерархическую структуру кластеров.

На практике множество  $B(c)$  и его компоненты  $B_i(c)$  недоступны напрямую, поскольку истинная плотность  $f(X)$  неизвестна. Однако с использованием непараметрических оценок плотности можно построить приближённое множество уровня и использовать его для выявления кластеров. Это позволяет задать концептуальную основу для методов обучения без учителя, в которых наличие плотностной структуры в данных служит основой для группировки наблюдений.

Таким образом, модель кластеров уровня плотности обеспечивает строгую вероятностную интерпретацию кластеризации как задачи геометрического разделения множества высокого уровня плотности. Это становится особенно важным в ситуациях, когда отсутствуют априорные сведения о принадлежности объектов к классам, а само разделение должно быть основано исключительно на свойствах распределения наблюдаемых данных.

## 2.2 Нейросетевая регрессия (случай одного класса)

Как отмечалось в разделе 1.2.3, многослойный персептрон с кусочно-линейной функцией активации (в частности, с активацией вида  $|x|$ ), при

наличии  $L$  скрытых слоёв, каждый из которых содержит  $k$  нейронов, способен осуществлять  $\varepsilon$ -приближенную аппроксимацию любой непрерывной функции на компакте. При этом конструкция такой нейросети задаёт иерархическое (по слоям) разбиение компакта  $[0, 1]^d$  на  $O(k^{dL})$  ячеек, внутри которых выход нейросети представляет собой линейную функцию. Вычисление значения такой нейросети в произвольной точке  $x \in [0, 1]^d$  требует только последовательного выполнения операций скалярного умножения и сравнения, что обеспечивает высокую вычислительную эффективность полученной модели.

Для дальнейшего анализа введём обобщённую задачу регрессии, сформулированную следующим образом. Пусть имеется исходный набор наблюдений  $\{X_i\}_{i=1}^n$ , представляющий собой независимые одинаково распределённые случайные величины на компакте  $[0, 1]^d$  с неизвестной ограниченной плотностью распределения  $f(X)$ . Этот набор можно интерпретировать как наблюдения некоторого целевого процесса, подлежащего моделированию. Для каждого такого наблюдения  $X_i$  положим значение метки  $Y_i = 1$ .

Дополнительно сформируем “фоновые” наблюдения  $\{X_i\}_{i=n+1}^{2n}$ , представляющие собой независимые одинаково распределённые случайные величины с равномерной плотностью распределения  $p(X)$  на том же компакте  $[0, 1]^d$ . Для этих наблюдений положим значения  $Y_i = 0$ . В результате будет получен комбинированный набор данных  $\{(X_i, Y_i)\}_{i=1}^{2n}$  мощности  $2n$ .

Рассмотрим теперь задачу построения аппроксимирующей полносвязной нейросети  $c_n(X)$ , решающей задачу регрессии в классе моделей фиксированной сложности, аналогичную задаче (1.10). Требуется найти такую нейросеть  $c_n^*(X)$ , минимизирующую среднеквадратичную ошибку на объединённом наборе данных:

$$\sum_{i=1}^{2n} (c_n(X_i) - Y_i)^2 \rightarrow \min_{c_n}, \quad (2.1)$$

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

Пусть в результате построения  $c_n^*(X)$  на компакте  $[0, 1]^d$  получено разбиение на  $N$  ячеек  $K = \{K_1, K_2, \dots, K_N\}$ . 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:

$$\sum_{i=1}^{2n} (h_n(X_i) - Y_i)^2 \rightarrow \min_{h_n} \quad (2.2)$$

Как и в (1.11), задача (2.2) может быть решена независимо в каждой ячейке  $K_r$ , при этом оптимальное значение  $h_n^*(X)$  в данной ячейке определяется соотношением:

$$h_n^*(X) = \frac{n_1(X)}{n_1(X) + n_0(X)}, \quad (2.3)$$

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 2.1.

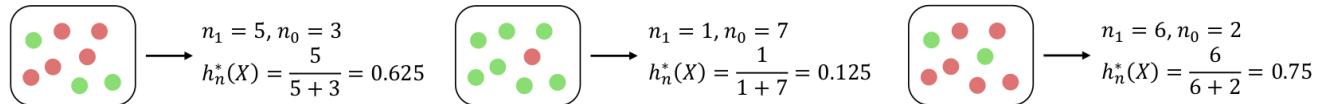


Рисунок 2.1 — Пример вычисления  $h_n^*(X)$  в некоторой ячейке  $K_r$  в унарном случае

Полученная функция  $h_n^*(X)$  представляет собой оценку апостериорной вероятности принадлежности к целевому распределению  $f(X)$ , основанную на локальной структуре данных в разбиении, индуцированном нейросетевой аппроксимацией.

## 2.3 Нейросетевая регрессия как оценка апостериорной вероятности класса

Рассмотрим гистограммные оценки плотности для двух классов:

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

где  $V(K_r)$  – мера ячейки  $K_r$ .

На основе этих оценок можно определить байесовскую аппроксимацию апостериорной вероятности класса:

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

Зададим порог  $\beta \in [0, 1]$ . Тогда неравенство  $h_n^*(X) > \beta$  позволяет выделить из множества  $K$  ячейки с высоким значением выборочной оценки апостериорной вероятности класса +1. Преобразуя выражение, получим эквивалентную форму [56]:

$$f_n(X) > \frac{\beta}{1 - \beta} \cdot p_n(X).$$

Иными словами, объект  $X$  попадает в область пространства признаков, где плотность первого класса превышает плотность фонового класса более чем в  $\frac{\beta}{1 - \beta}$  раз. Это условие можно использовать для выделения из множества ячеек  $\{K_1, K_2, \dots, K_N\}$  подмножества, соответствующего кластеру высокой плотности. Такие области могут рассматриваться как выборочные приближения к области, поддерживающей плотность распределения первого класса.

По аналогии с разделом 1.4, согласно результатам работы [8], для обеспечения статистической состоятельности гистограммных оценок плотности необходимо, чтобы с ростом объёма обучающей выборки происходило соответствующее увеличение числа нейронов в аппроксимирующей сети. Это условие отражает потребность в возрастающем разрешении пространства признаков, необходимом для точного приближения апостериорной вероятности.

В таком случае для решения о принадлежности нового наблюдения  $X$  области высокой плотности достаточно проверить выполнение неравенства  $c_n^*(X) > \beta$ , что представляет собой гораздо более простую с вычислительной точки зрения операцию, чем вычисление  $h_n^*(X)$ .

Таким образом, задача принятия решения сводится к сравнению выхода нейросети с порогом, что обеспечивает линейную по числу слоёв и нейронов вычислительную сложность и устраняет необходимость работы с явным представлением плотностей. Это делает метод особенно привлекательным в задачах, требующих масштабируемости и эффективности при обработке новых входных данных.

## 2.4 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  $c = 1, \dots, C$  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.

## 2.5 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  $(c_n^{(1)}(x), \dots, c_n^{(C)}(x))$  образует векторную оценку, позволяющую как выбрать наиболее вероятный класс (например, по максимуму), так и сформулировать стратегию отказа, если все оценки не превышают заданного порога  $\beta$ . 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.

## 2.6 Оценка качества унарных классификаторов

При оценке качества стандартных многоклассовых классификаторов традиционно используют метрики точности, полноты,  $F_1$ -score и аналогичные [57]. Однако в контексте унарной классификации такие показатели оказываются недостаточно информативными, так как каждый классификатор в унарной схеме обучается независимо и ориентирован на различие своего целевого класса от фонового распределения. В частности, стандартная точность не учитывает случаи “отказа” классификатора (когда выход нейросети не превышает порог  $\beta$ ), а  $F_1$ -score и подобные метрики не отражают взаимное влияние классификаторов при многоклассовой интерпретации.

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

### 2.6.1 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  $\beta$ . Формально для двух классов показатели вычисляются следующим образом:

$$\begin{aligned} n_1^{(1)} &= \sum_{i=1}^{n^{(1)}} \mathbb{I}_{\{c^{(1)}(x_i) \geq \beta\}}, & n_2^{(2)} &= \sum_{i=1}^{n^{(2)}} \mathbb{I}_{\{c^{(2)}(x_i) \geq \beta\}}, \\ p^{(1)} &= \frac{n_1^{(1)}}{n^{(1)}}, & p^{(2)} &= \frac{n_2^{(2)}}{n^{(2)}}, \end{aligned}$$

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.

## 2.6.2 Эффективность классификатора

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

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

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

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.

### 2.6.3 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:

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

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

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.

### 2.6.4 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 2.2а). 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 2.2б). 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  $F_{12} = 0$ . 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 2.2в). Мощность обоих классификаторов равна нулю ( $p^{(1)} = p^{(2)} = 0$ ), что даёт  $P_{12} = 0$ . Эффективность также равна нулю ( $F_{12} = 0$ ). Наплываемость обоих классификаторов максимальна ( $G_{12} = 1$ ). При этом accuracy, precision, recall и  $F_1$  принимают значение 0.5.

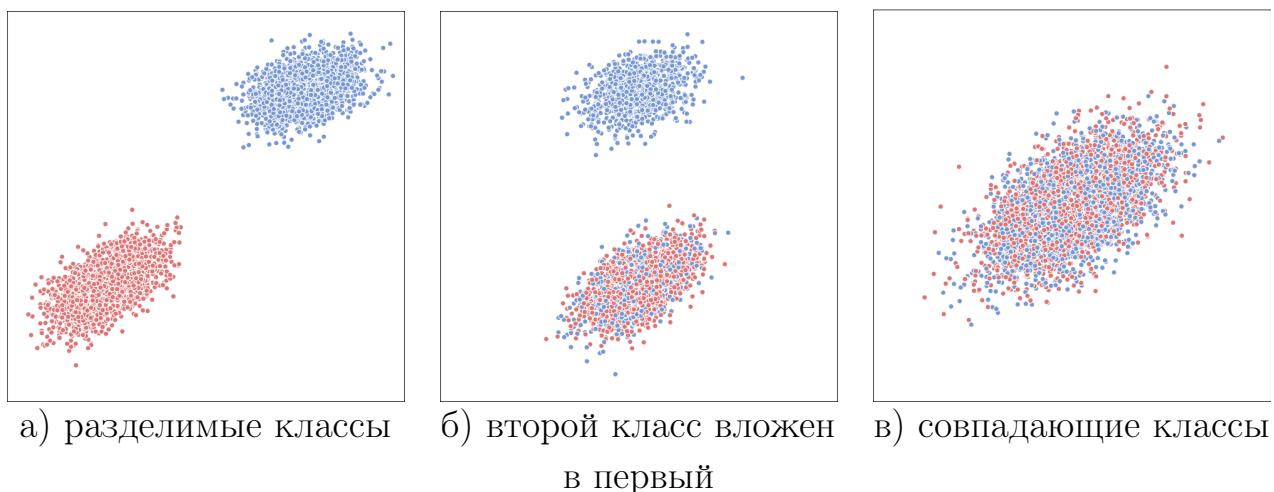


Рисунок 2.2 — Модельные ситуации для анализа метрик

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

## 2.6.5 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.

## 2.7 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 2.3 показана граница принятия решения, построенная унарным классификатором для одного класса. Видно, что модель успешно выделяет область высокой плотности положительного класса, отсекая фон.

На рисунке 2.4 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,

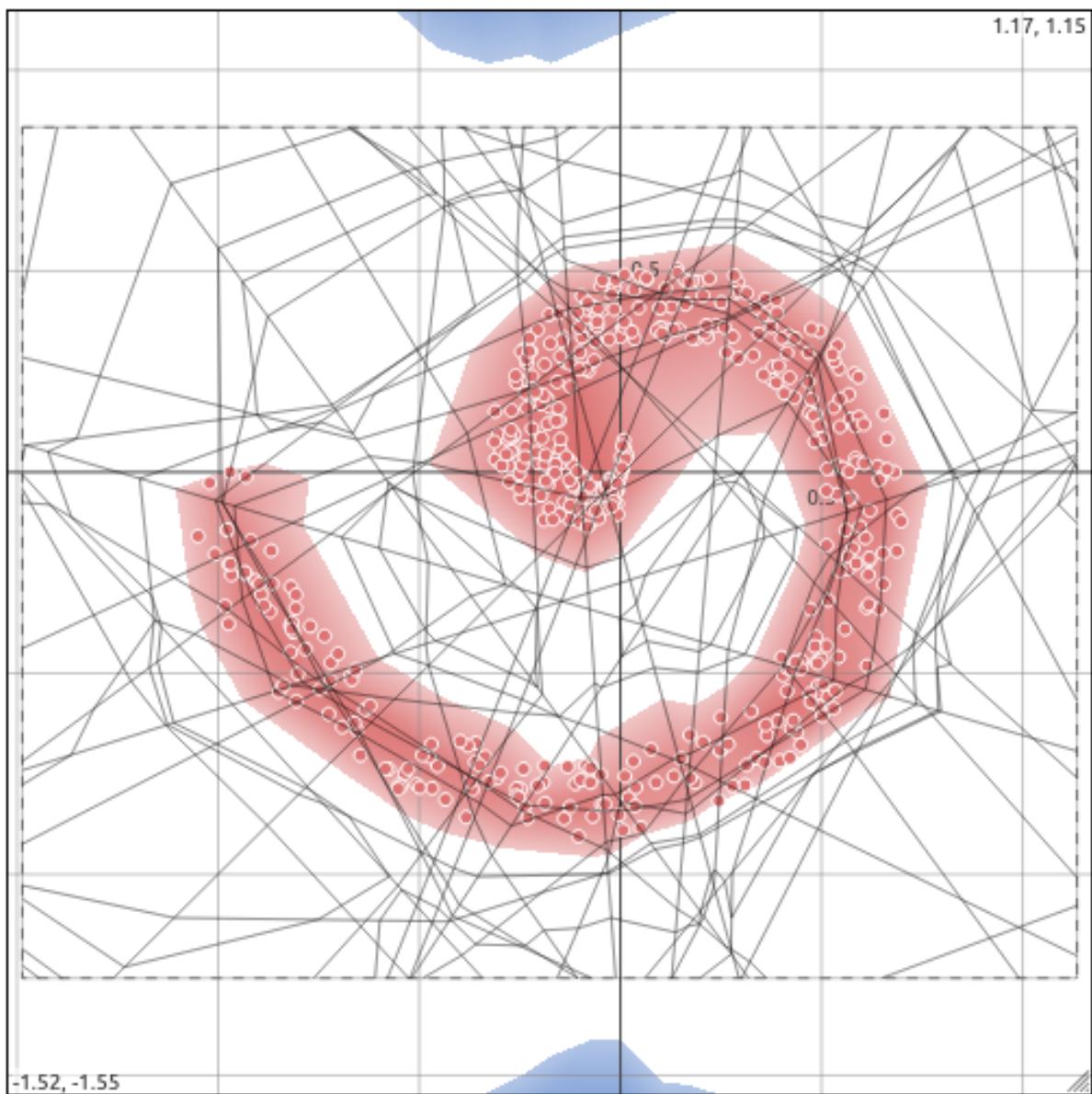


Рисунок 2.3 — Оценка плотности одного класса с использованием унарной схемы

области плотности получаются хорошо различимыми и не искаженными из-за дисбаланса. Это подтверждает устойчивость метода к нарушению пропорций классов (рисунок 2.5).

Таким образом, предложенная схема построения унарных классификаторов позволяет надёжно и интерпретируемо решать задачу многоклассовой классификации, не требуя дополнительных допущений о балансе данных или единой архитектуре модели. Векторная оценка апостериорных вероятностей предоставляет дополнительную гибкость и возможность построения сложных решений с механизмами отказа или уточнения.

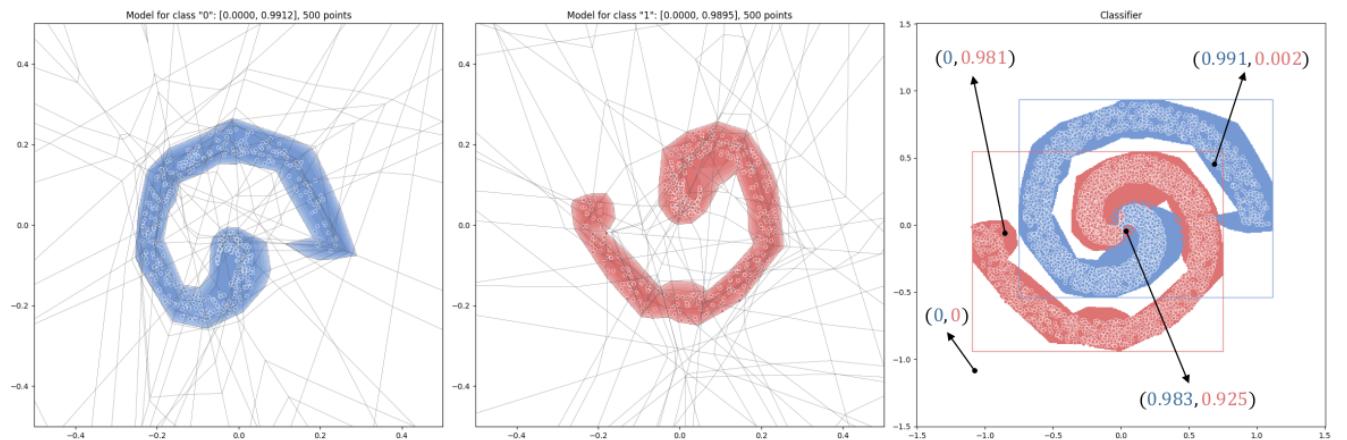


Рисунок 2.4 — Унарная классификация для двух классов

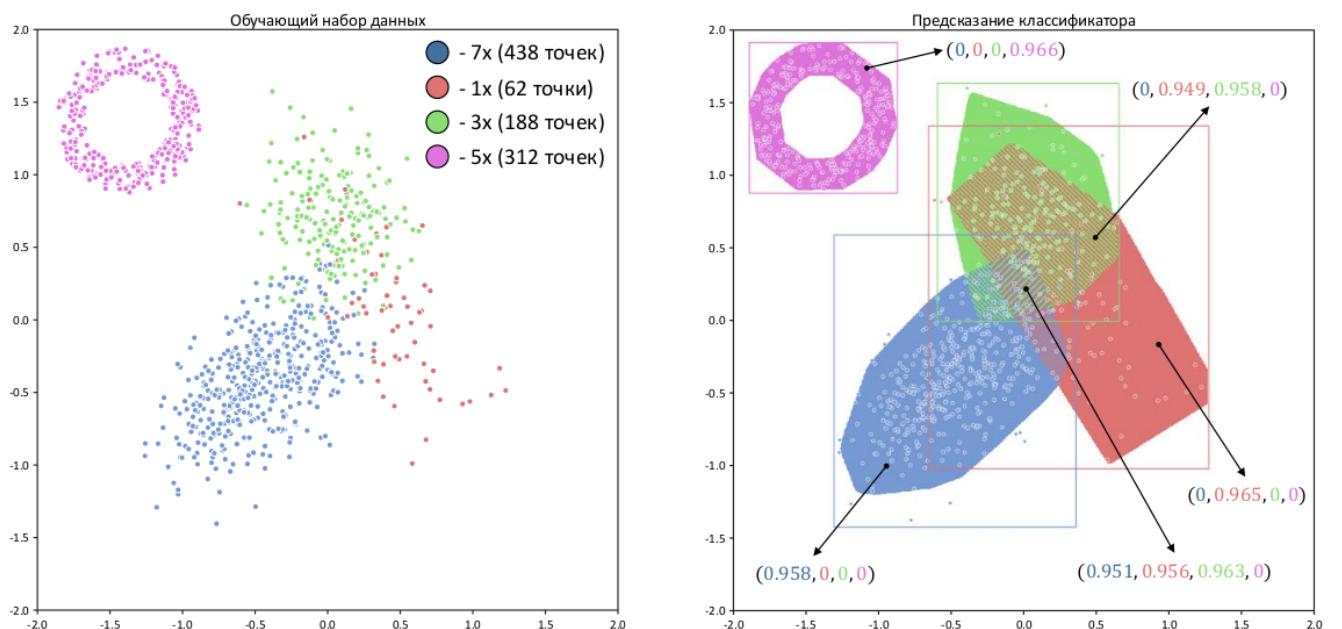


Рисунок 2.5 — Унарная классификация для четырёх классов с дисбалансом

## Глава 3. Применение унарной классификации

### 3.1 Построение репродукционных выборок

#### ДОПИСАТЬ ПРО РЕПРОДУКЦИЮ ПО ГИСТОГРАММЕ

Одним из ключевых применений унарной классификации является задача создания синтетических табличных данных, особенно актуальная в условиях ограниченного доступа к реальным данным. Такие ограничения [58] могут быть обусловлены законодательными мерами по защите персональных данных [59], коммерческой тайной или просто недостаточным объёмом исходной выборки. Синтетические данные находят применение в обучении моделей, увеличении объёма обучающих данных [60], обеспечении воспроизводимости научных исследований [61] и безопасной передаче информации между организациями.

Основное требование к синтетическим данным – сохранение статистических и структурных свойств оригинального распределения при гарантии отсутствия утечки чувствительной информации [62]. На практике это означает, что синтетические выборки должны отражать ту же плотность вероятности, что и оригинальные данные, при этом исключая прямое копирование реальных наблюдений.

Для генерации синтетических данных применяются как классические статистические методы, так и модели, основанные на машинном обучении [63] – например, вариационные автоэнкодеры (VAE) [64], генеративно-состязательные сети (GAN) [65], диффузионные модели [66] и др. [67]. Статистические методы обеспечивают согласованность оценок – то есть, при росте объёма выборки оценка приближается к истинному распределению. В то же время, нейросетевые методы, несмотря на высокую эмпирическую эффективность, зачастую не обладают теоретическими гарантиями сохранения статистических свойств.

В работе [68] предлагается альтернативный подход к генерации синтетических данных, основанный на унарной классификации. В отличие от традиционных генеративных моделей, которые непосредственно генерируют новые объекты, здесь нейросеть обучается различать реальные точки данных и точки, равномерно сэмплированные из фонового распределения на ограниченной области пространства. В дальнейшем результат классификации

используется для фильтрации вновь сгенерированных фоновых точек, формируя репродукционную выборку, приближающую плотность исходных данных.

### 3.1.1 Постановка задачи

Рассмотрим множество наблюдений  $X = \{x_1, x_2, \dots, x_n\} \in [0, 1]^d$ , представляющее собой выборку из неизвестного распределения с плотностью  $f(x)$ . Цель состоит в построении синтетической выборки  $\tilde{X} = \{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_m\}$ , которая сохраняет статистические свойства оригинального распределения (рисунок 3.1).

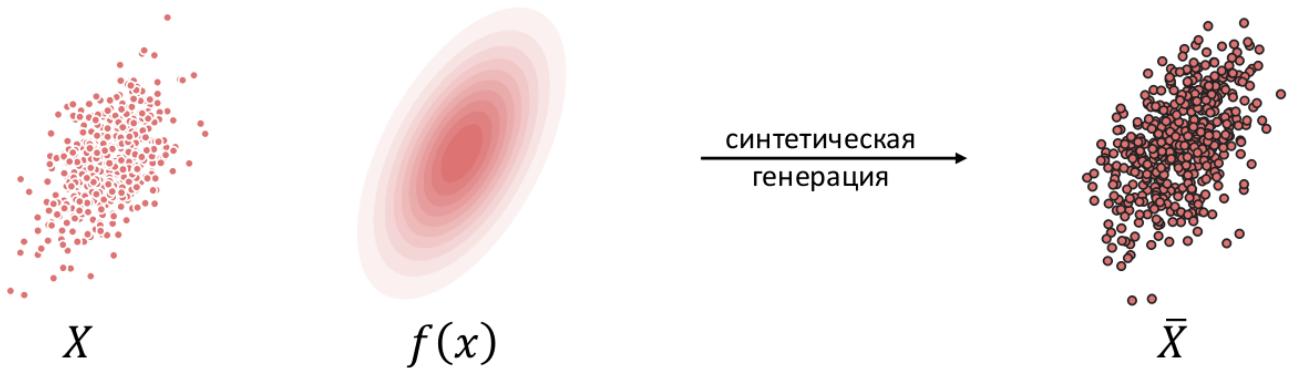


Рисунок 3.1 — Схематичное представление задачи создания синтетических данных

Традиционные методы оценки плотности распределения обычно предполагают построение непараметрической оценки  $\hat{f}(x)$  для аппроксимации  $f(x)$ . В отличие от этого, предлагается альтернативный подход, основанный на унарной классификации, где нейронная сеть обучается различать реальные точки данных и выборки, взятые из равномерного фонового распределения.

### 3.1.2 Обучение классификатора

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

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

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

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.

### 3.1.3 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 оригинальной плотности  $f(x)$ , не предполагая явной генеративной модели. При этом плотность оценивается через классификационную задачу, что позволяет интерпретировать модель как адаптивную гистограмму.

### 3.1.4 Экспериментальное исследование

Для демонстрации эффективности метода проведены эксперименты на модельных датасетах с известной структурой (рисунок 3.2). Это позволяет объективно оценить способность модели к воспроизведению статистических свойств. Рассматривались следующие выборки:

- **Сpirаль:** 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.

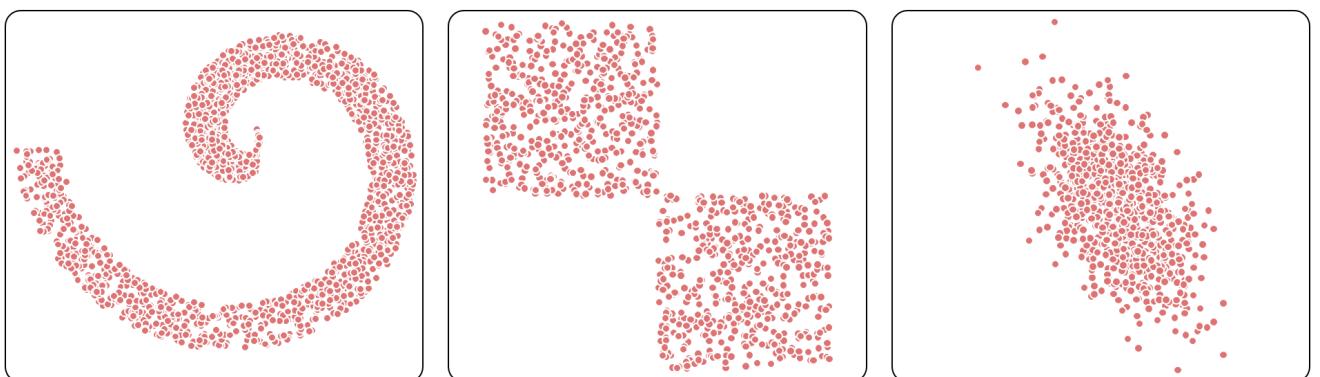


Рисунок 3.2 — Использованные наборы данных для построения репродукционных выборок (спираль, два квадрата и гауссиан)

Визуализация результатов (рисунок 3.3) демонстрирует, что сгенерированные данные точно воспроизводят форму, плотность и вариативность

оригинальных данных. В случае многомерного нормального распределения сохраняется ковариационная структура, хотя наблюдается небольшое увеличение дисперсии – эффект, обусловленный ростом размерности и разрежённостью пространства.



Рисунок 3.3 — результаты эксперимента с синтетическими данными

## Архитектура и параметры обучения

Модель обучалась на 100 эпохах, размер батча – 32. Использовались различные архитектуры сети:

- **d-10-1**: минималистичная модель с одним скрытым слоем.
- **d-10-100-1**: глубокая архитектура с повышенной ёмкостью.
- **d-10-10-10-1**: сбалансированная конфигурация.

Оптимизация осуществлялась методом Adam [69] с шагом обучения  $10^{-3}$ .

Функция активации – модульная.

## Ограничения

Преимущество предложенного подхода заключается в естественной аппроксимации плотности через сетевую структуру, эквивалентную аддитивной гистограмме. Однако с ростом размерности пространства наблюдается ухудшение качества генерации из-за разреженности, что требует более сложной архитектуры и дополнительных механизмов фильтрации. Одним из решений

является введение более высокого порога уверенности  $\beta$ , при котором в синтетическую выборку включаются только точки с  $c(x) > \beta$ , снижая шум, но уменьшая разнообразие.

Кроме того, в отличие от генеративных моделей, основанных на латентных переменных (например, CTGAN [70], TVAE [71]), рассматриваемый метод не требует обучения скрытого представления и обладает более прозрачной интерпретируемостью за счёт прямой связи с оценкой плотности.

Важно отметить, что модель может наследовать структурные и социальные смещения из обучающих данных. Поэтому при генерации синтетических выборок, особенно в задачах с социально значимой информацией, требуется дополнительный контроль справедливости.

## Выводы

Таким образом, метод построения репродукционных выборок через унарную классификацию представляет собой эффективную альтернативу генеративным моделям. Он сочетает в себе простоту реализации, теоретическую интерпретируемость и способность воспроизводить сложные структуры данных. Это делает его перспективным инструментом для синтетической генерации данных, особенно в условиях ограниченного доступа к реальным выборкам.

### 3.2 Обучение нейросети по некомплектным данным

Отсутствие значений в данных (*пропуски*) представляет собой одну из наиболее распространённых и при этом наименее формализованных проблем, с которой сталкиваются в прикладном машинном обучении и статистическом анализе. Согласно [6], в любой области, где данные собираются при помощи опросов, сенсоров или в рамках наблюдательных исследований, практически неизбежно возникает неполнота – отсутствие значений некоторых признаков у части объектов выборки. Такая ситуация наблюдается в биомедицинских

данных [72], социологических опросах, прикладных инженерных задачах, финансовом моделировании и других областях.

Пропуски могут быть обусловлены множеством факторов: техническими сбоями [73], отказами респондентов отвечать на конкретные вопросы, ограничениями ресурсоёмких измерений, фильтрацией данных, нарушениями сбора и хранения. При этом даже небольшое число пропущенных значений может существенно повлиять на результаты анализа, особенно при высокой размерности признакового пространства или в задачах, чувствительных к структуре выборки.

Корректное обращение с пропущенными значениями требует не только выбора соответствующей стратегии заполнения, но и понимания механизма возникновения пропусков. Как подчёркивается в [6; 74], пренебрежение механизмом отсутствия может привести к систематическим ошибкам, смещению оценок, снижению статистической мощности и искажению выводов. Особенно это критично в задачах обучения нейросетей, где входные данные передаются в модели в виде числовых векторов, не допускающих наличия неопределённых компонент.

### 3.2.1 Задача классификации данных с пропущенными значениями

Пусть имеется обучающая выборка из  $n$  объектов:  $\{(X_i, Y_i)\}_{i=1}^n$ , где  $X_i \in [0, 1]^d$  – вектор признаков, а  $Y_i \in \{1, 2, \dots, C\}$  – метка класса. Предполагается, что некоторые признаки могут быть пропущены. Для описания структуры отсутствующих данных вводится матрица  $M \in \{0, 1\}^{n \times d}$ , где  $m_{ij} = 1$  означает, что признак  $j$  отсутствует у объекта  $i$ , а  $m_{ij} = 0$  – признак наблюдаем.

Обозначим через  $X_0$  множество наблюдаемых признаков, а через  $X_1$  – множество отсутствующих.

### 3.2.2 Типы механизмов пропусков

Следуя формализации, приведённой в [6], различают три основных типа пропусков:

- Пропуски, отсутствующие полностью случайно (MCAR, missing completely at random): механизм пропусков не зависит ни от наблюдаемых, ни от ненаблюдаемых данных. Формально,  $P(M | X_0, X_1, Y) = P(M)$ .
- Пропуски, отсутствующие случайно (MAR, missing at random): вероятность отсутствия значения может зависеть от наблюдаемых данных, но не от пропущенных. То есть  $P(M | X_0, X_1, Y) = P(M | X_0, Y)$ .
- Пропуски, отсутствующие не случайно (NMAR, not missing at random): вероятность отсутствия значения зависит от ненаблюдаемых данных, т.е.  $P(M | X_0, X_1, Y)$  не сводится к предыдущим случаям.

При предположении MCAR и MAR допускается игнорирование механизма пропусков при построении модели. В случае NMAR необходимо явно моделировать процесс отсутствия данных, что усложняет задачу.

### 3.2.3 Существующие методы обработки пропусков

В литературе представлены различные стратегии, позволяющие бороться с неполнотой данных. Кратко охарактеризуем наиболее известные из них.

#### **Удаление некомплектных наблюдений**

Простейший метод – исключение строк с пропущенными значениями. Такой подход корректен только при MCAR и при условии, что доля удаляемых объектов незначительна. Основной недостаток – потеря потенциально полезной информации и возможное смещение выборки.

## Заполнение средним значением (mean imputation)

Отсутствующие значения заменяются средним по соответствующему признаку, вычисленным по доступным наблюдениям:

$$x_{ij} \leftarrow \frac{1}{|\mathcal{I}_j|} \sum_{i \in \mathcal{I}_j} x_{ij}, \quad \mathcal{I}_j = \{i \mid m_{ij} = 0\}.$$

Метод не учитывает зависимость между признаками, занижает дисперсию и нарушает ковариационную структуру данных.

## Заполнение медианой или модой

Применяется для категориальных признаков (мода) или числовых с выбросами (медиана). Устраняет чувствительность к экстремальным значениям, но по-прежнему игнорирует корреляции и структуру признаков.

## $k$ -ближайших соседей (kNN imputation)

Для объекта с пропущенным признаком ищется множество  $k$  ближайших (по наблюдаемым координатам) объектов, и пропущенное значение заполняется агрегатором (среднее, медиана) по этому множеству. Обозначим через  $\mathcal{N}_k(i)$  множество  $k$ -соседей объекта  $i$ . Тогда:

$$x_{ij} \leftarrow \frac{1}{k} \sum_{\ell \in \mathcal{N}_k(i)} x_{\ell j}.$$

Метод чувствителен к выбору метрики, неустойчив при высокой разреженности и требует полного набора значений для вычисления расстояний [75].

## Множественное заполнение и ЕМ-алгоритм

В рамках подхода максимального правдоподобия в [76] вводится вероятностная модель данных и итеративно оцениваются параметры и пропущенные значения. На шаге **E** оценивается распределение некомплектных данных при фиксированных параметрах, на шаге **M** – параметры по комплектным данным. Подход требует априорных предположений о распределении данных (обычно гауссовское) и высоких вычислительных затрат. В случае множественного заполнения [77] создаётся несколько возможных вариантов с последующей агрегацией.

## Методы на основе РСА и автокодировщиков

Пропуски заполняются с помощью аппроксимации данных в латентном пространстве. В методах РСА недостающие значения восстанавливаются проекцией на подпространство главных компонент [78]. В нейросетевом варианте – автокодировщики обучаются на комплектных данных и используются для реконструкции пропущенных [79].

### 3.2.4 Ограничения классических методов

Все перечисленные методы обладают рядом общих недостатков:

- не учитывают апостериорную неопределённость заполнения;
- вводят систематическое смещение в оценки параметров модели;
- теряют вариативность по заполненным признакам;
- игнорируют структуру задачи (например, наличие классов в классификации).

Таким образом, возникает необходимость в методах, которые позволяли бы обрабатывать некомплектные данные без грубых аппроксимаций, использо-

вали бы информацию о метках классов и сохраняли бы стохастический характер восстановления недостающих признаков.

В работе [80] предлагается альтернативный метод, основанный на вероятностном заполнении признаков с помощью унарной классификации и многослойного персептрона.

### 3.2.5 Метод вероятностного заполнения

Пусть  $X \in [0, 1]^{n \times d}$  – обучающая выборка с пропущенными значениями,  $Y \in \{0, 1, \dots, C\}^n$  – вектор меток классов,  $M \in \{0, 1\}^{n \times d}$  – матрица пропусков. Предполагается, что механизм пропусков является случайным (MCAR или MAR по классификации Рубина [6]).

Предлагаемый метод обучения MLP при наличии пропусков в обучающей выборке применяется последовательно к каждому из  $C$  классов и состоит из трёх шагов (схематичное изображение методики изображено на рисунке 3.4):

1. **Начальное обучение.** Для комплектной подвыборки  $\{X\}_i^n$   $j$ -го класса,  $j \in \{1, 2, \dots, C\}$ , решить задачу унарной классификации и построить персепtron, реализующий кусочно-линейную непрерывную функцию  $c_n^{(j)}(X)$ .
2. **Дообучение.** Дообучение осуществляется по всей обучающей выборке  $j$ -го класса отдельными эпохами. Перед текущей эпохой выполнить временное (для данной эпохи) заполнение некомплектных наблюдений. Для каждого некомплектного наблюдения  $X$ :
  - а) Разделить множество индексов координат вектора  $X = (x_1, x_2, \dots, x_d)$  на два подмножества  $M_0$  и  $M_1$ , включающие соответственно индексы заполненных и пропущенных координат.
  - б) Заполнить координаты  $X$  из  $M_1$  наблюдениями равномерно распределенной случайной величины на отрезке  $[0, 1]$ , в результате чего будет получен комплектный вектор  $X'$ . Вычислить  $c_n^{(j)}(X')$ . Сгенерировать наблюдение биномиальной случайной величины с вероятностью успеха  $p = c_n^{(j)}(X')$ .

- в) При успешном исходе временно заменить в обучающей выборке некомплектный вектор  $X$  на комплектный вектор  $X'$  и перейти к рассмотрению следующего некомплектного наблюдения. В противном случае повторить предыдущий шаг.
- г) Выполнить дообучение сети по “доукомплектованной” обучающей выборке.
3. Перейти к следующей эпохе дообучения, повторяя шаги а-г, до полного завершения обучения  $MLP_j$  для  $j$ -го класса с функцией нейросетевой регрессии  $c_n^{(j)}(X)$ .

Повторяя шаги 1-3 для всех классов, получим  $C$  обученных нейросетей  $MLP_j$  и соответствующих им непрерывных кусочно-линейных функций  $\{c_n^{(1)}(X), c_n^{(2)}(X), \dots, c_n^{(C)}(X)\}$ , каждая из которых есть выборочная оценка апостериорной вероятности соответствующего класса в точке  $X$ .

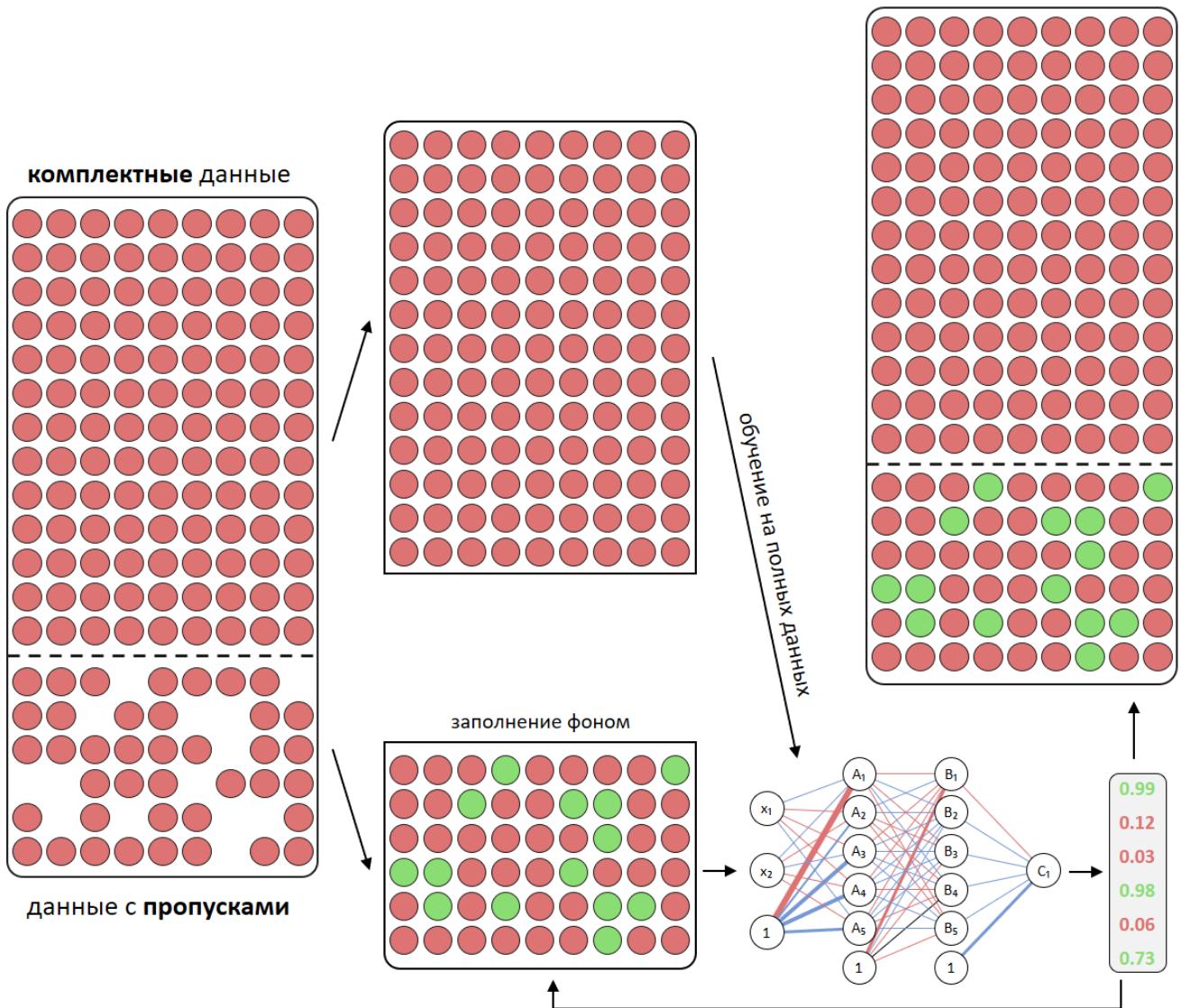


Рисунок 3.4 — Схема обучения  $c_n^{(j)}(X)$  по некомплектным данным

### 3.2.6 Классификация комплектного наблюдения

Для решения задачи классификации комплектного наблюдения  $X$  возможны различные стратегии. Простейшая состоит в выборе класса, для которого апостериорная вероятность максимальна. Другой вариант – выбрать в качестве решения все классы, значения апостериорной вероятности для которых больше некоторого заданного порога, и продолжить решение задачи классификации, например, в другом признаковом пространстве.

### 3.2.7 Экспериментальное исследование

Для экспериментального анализа были подготовлены несколько модельных наборов данных с чёткой визуальной и статистической интерпретацией классов. Это позволило обеспечить контролируемую среду и надёжную оценку устойчивости моделей к пропущенным значениям.

#### Используемые наборы данных

- **Гауссианы** – два нормально распределённых кластера с равной дисперсией и небольшим перекрытием (рисунок 3.5а).
- **Сpirали** – классы формируют витки спиралей с общей точкой начала координат, разделение классов сильно нелинейное (рисунок 3.5б).
- **Кольцо и круг** – один класс расположен внутри круга, второй образует кольцо с зазором между границами (рисунок 3.5в).

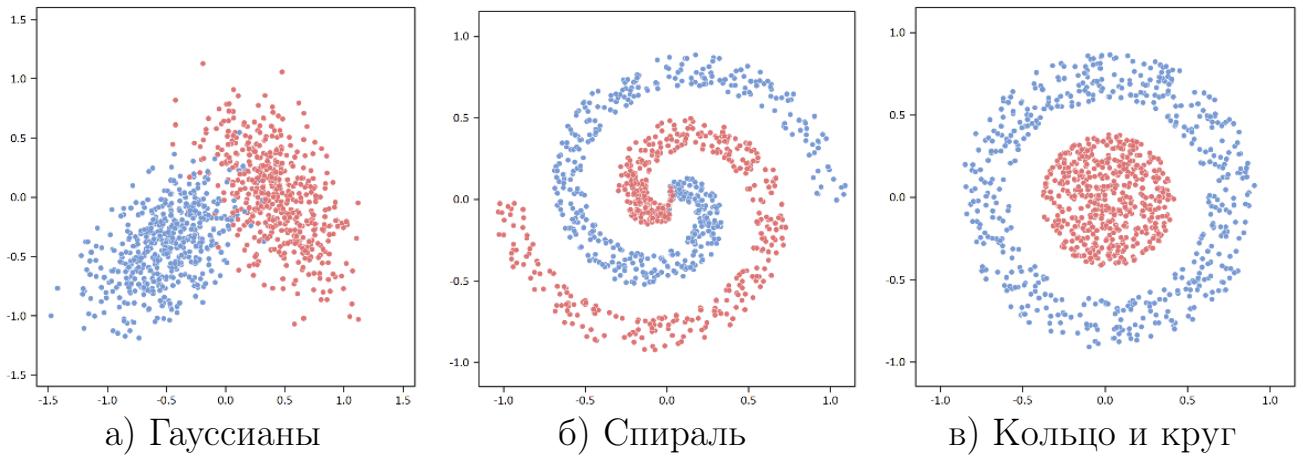


Рисунок 3.5 — Использованные наборы для обучения модели на данных с пропусками

### Обработка пропущенных значений

Во всех наборах данных искусственно вводились пропуски в признаках с уровнями 20%, 40%, 50%, 60%, 80% и 90%. Пропуски вносились случайно и только в признаках (целевые метки всегда сохранялись). Были рассмотрены следующие классические методы обработки пропусков:

- **mean** – заполнение по среднему значению признака;
- **mode** – заполнение наиболее частым значением;
- $kNN(k = 3)$  – заполнение по ближайшим трём соседям в евклидовом пространстве;
- $kNN(k = 7)$  – аналогично, но с  $k = 7$ ;
- **reproduction** (предложенный метод) – метод, основанный на унарной классификации.

### Сценарии обучения

Для каждой комбинации набора данных и уровня пропусков модель обучалась в следующих режимах:

- **full** – обучение на полном наборе без пропусков;
- **complete** – обучение только на тех примерах, где отсутствуют пропуски;

- **imputed** – обучение на наборе, где пропуски заполнялись одним из методов.

В качестве модели использовался многослойный персепtron с  $L = 2$  скрытыми слоями по  $k = 20$  нейронов в каждом и одним выходным слоем. Обучение осуществлялось на протяжении 500 эпох.

Каждая комбинация набора данных, уровня пропусков и метода заполнения запускалась 50 раз с различными начальными инициализациями весовых коэффициентов. В качестве основной метрики использовалась точность классификации (accuracy) на тестовом множестве из соответствующего набора данных из 5000 элементов. Все тестовые наборы содержали только комплектные данные.

Для метода репродукции персептрон обучался в течение 50 эпох на данных без пропусков, а затем каждую эпоху запускался процесс вероятностного заполнения пропусков и обучение продолжалось уже на обновлённых заполненных данных.

## Результаты

Сводные таблицы 2, 3 и 4 по accuracy (в среднем по 50 запускам) представлены ниже. Визуальный анализ показывает, что метод репродукции демонстрирует более высокую устойчивость при высоких уровнях пропусков, особенно на сложных наборах данных, как например "кольцо и круг". Традиционные методы заполнения (среднее, мода) показывают ожидаемое снижение качества, особенно при пропусках выше 60%. Метод  $kNN$  даёт умеренное улучшение, но чувствителен к плотности выборки.

Таблица 2 — Оценка метода репродукции для заполнения пропусков на наборе данных “Гауссианы“

Доля	full	complete	reproduce	mean	knn 7
20%		0.919±0.006	<b>0.925±0.004</b>	0.917±0.007	0.921±0.006
40%		<b>0.919±0.006</b>	0.917±0.009	0.897±0.008	0.915±0.005
50%	0.925	0.917±0.005	<b>0.918±0.008</b>	0.904±0.014	0.917±0.005
60%	±	0.910±0.013	<b>0.921±0.009</b>	0.866±0.018	0.894±0.012
80%	0.003	0.875±0.011	<b>0.912±0.008</b>	0.752±0.047	0.851±0.011
90%		0.842±0.023	<b>0.906±0.007</b>	0.593±0.092	0.806±0.015

Таблица 3 — Оценка метода репродукции для заполнения пропусков на наборе данных “Сpirаль“

Доля	full	complete	reproduce	mean	knn 7
20%		0.936±0.031	<b>0.945±0.025</b>	0.922±0.032	0.941±0.020
40%		0.924±0.023	<b>0.930±0.021</b>	0.899±0.034	0.918±0.025
50%	0.941	<b>0.926±0.016</b>	0.910±0.034	0.893±0.031	0.868±0.062
60%	±	<b>0.913±0.030</b>	0.898±0.047	0.890±0.027	0.853±0.040
80%	0.024	<b>0.869±0.039</b>	0.861±0.044	0.750±0.095	0.773±0.062
90%		<b>0.827±0.042</b>	0.812±0.067	0.545±0.082	0.695±0.040

Таблица 4 — Оценка метода репродукции для заполнения пропусков на наборе данных “Кольцо и круг“

Доля	full	complete	reproduce	mean	knn 7
20%		0.987±0.009	<b>0.989±0.006</b>	0.941±0.058	0.927±0.086
40%		0.984±0.011	<b>0.986±0.011</b>	0.865±0.103	0.846±0.123
50%	0.981	0.971±0.019	<b>0.984±0.016</b>	0.852±0.094	0.751±0.149
60%	±	0.974±0.018	<b>0.981±0.016</b>	0.763±0.083	0.705±0.103
80%	0.014	0.892±0.136	<b>0.964±0.031</b>	0.609±0.148	0.618±0.069
90%		0.852±0.080	<b>0.952±0.038</b>	0.295±0.126	0.562±0.055

## Глава 4. Интеллектуальная система машинного обучения для визуализации и исследования методов классификации

### 4.1 Общая характеристика интеллектуальной системы машинного обучения

В рамках выполненного исследования была разработана интеллектуальная система машинного обучения, предназначенная для визуального и экспериментального изучения поведения моделей классификации в условиях ограниченного объёма обучающих данных, дисбаланса классов, а также в присутствии фона и пропущенных значений. Система представляет собой автономное клиентское приложение, реализованное на языке JavaScript [81], не требующее установки, интернет-соединения или использования графического ускорителя, что обеспечивает его широкую доступность и воспроизводимость экспериментов.

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 устойчивости 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 метода с фоном;
- визуализация, построение и загрузка обучающих и тестовых множеств;
- проведение экспериментальных исследований по созданию синтетических данных, обработке данных с пропусками, а также анализ объясняющего двоичного дерева eXBTree.

Таким образом, система реализует весь цикл исследования: от генерации 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.

## 4.2 Архитектура и интерфейс интеллектуальной системы

Разработанная система реализована на JavaScript с использованием стандартных веб-технологий: HTML5 [82], CSS3 [83], SVG [84] и Canvas API [85]. Для стилизации применяется собственный CSS без привлечения сторонних фреймворков. Взаимодействие между компонентами построено на событийной модели с использованием собственного класса-эмиттера событий (EventEmitter). Основным управляющим объектом является класс Playground, который инкапсулирует логику координации работы всех компонентов и обмена данными между ними. Все вычисления производятся на стороне клиента, что исключает необходимость обращения к внешним серверам и обеспечивает полную автономность работы.

#### 4.2.1 Архитектура интеллектуальной системы

Архитектура системы выполнена по модульному принципу, включая следующие ключевые блоки:

- **Модуль данных** – отвечает за генерацию, хранение и загрузку обучающих и тестовых наборов.
- **Модуль модели** – реализует обучение персептрона и его использование для анализа.
- **Модуль обучения** – реализует методы градиентного спуска, алгоритмы оптимизации и функций потерь, а также формирование модифицированных обучающих выборок с фоновым распределением.
- **Модуль визуализации** – занимается отрисовкой данных, иерархии ячеек, карты предсказаний, структурных элементов модели, а также графиков метрик и гистограмм. Визуализация осуществляется с помощью Canvas API и SVG.
- **Модуль экспериментов** – обеспечивает выполнение преднастроенных экспериментов, таких как анализ дерева eXBTTree, создание синтетических данных и обучение модели на данных с пропусками.
- **Модуль управления** – реализует пользовательский интерфейс, включая меню, формы настройки параметров и кнопки управления, а также обработку событий от пользователя.

Все модули взаимодействуют между собой через механизм событий, что обеспечивает слабую связанность и гибкость расширения. Класс Playground выступает центральным контроллером, инициализирующим компоненты, регистрирующим сл�шатели событий и передающим данные между модулями.

#### 4.2.2 Структура интерфейса

Интерфейс интеллектуальной системы структурирован по трём основным вкладкам, каждая из которых реализована как независимый набор компонентов с собственным меню и областью отображения:

- **Вкладка “Данные“** – предоставляет средства генерации и файловой загрузки наборов данных. Отображение данных осуществляется в табличном виде и на графике с цветовой кодировкой для обучающего и тестового разбиений. Имеются инструменты нормализации и экспорта данных.
- **Вкладка “Обучение“** – главный визуально насыщенный раздел, где происходит настройка архитектуры модели (число слоёв, размер слоёв, функции активации, порог доверия), параметров обучения (скорость, функция потерь, оптимизатор, регуляризация) и параметров визуализации (отображение выходов модели и формируемых ячеек, точки обучающего, тестового и фонового множеств). Обучение может как запускаться и останавливаться по желанию, так и выполняться в виде единственного шага. Область просмотра динамически отображает состояние модели, метрики и распределение выходов персептрона на обучающих данных в виде гистограмм.
- **Вкладка “Эксперименты“** – содержит инструменты для запуска и анализа различных сценариев: анализ двоичного объясняющего дерева, создание синтетических данных, а также обучение модели на данных с пропусками. Результаты представлены в виде интерактивных таблиц, графиков и гистограмм, позволяющих детально исследовать поведение модели.

Интерфейс спроектирован с акцентом на интерактивность и прозрачность процесса: изменение параметров мгновенно отражается на визуализации, что позволяет пользователю оперативно оценивать влияние настроек.

#### **4.2.3 Аппаратные и программные требования**

Интеллектуальная система машинного обучения не требует установки дополнительных библиотек или серверной инфраструктуры. Для работы необходим любой современный браузер с поддержкой Javascript, HTML5 Canvas и SVG. Ресурсоёмкость минимальна, что позволяет запускать систему на большинстве персональных компьютеров без специальных требований.

## 4.3 Реализованные алгоритмы и методы визуализации

В интеллектуальной системе реализован широкий спектр алгоритмов и методов, необходимых для обучения, визуализации и анализа моделей нейросетевой классификации, а также для работы с неполными и синтетическими данными. Приведённые ниже компоненты составляют ядро вычислительного и аналитического функционала системы.

### 4.3.1 Многослойный персепtron

В качестве основной вычислительной модели реализован многослойный персептрон, представленный в виде набора последовательно соединённых полносвязных слоёв. Каждый слой выполняет матрично-векторное преобразование входных признаков с последующим применением нелинейной активации. Архитектура поддерживает произвольную глубину и размерность слоёв, задаваемую пользователем.

Особое внимание уделено производительности реализации. В целях обеспечения вычислительной эффективности произведено разворачивание вложенных циклов [86] и оптимизация операций умножения с использованием предварительного выделения буферов. Все операции реализованы в терминах низкоуровневых операций над типизированными массивами [87] JavaScript, без применения сторонних библиотек.

### 4.3.2 Оптимизационные алгоритмы

Для обучения нейросетевых моделей реализованы различные варианты стохастического градиентного спуска:

- SGD – базовый метод без накопления импульса;
- SGD с импульсом (momentum) – учитывает направление предыдущих градиентов;

- Adam – использует адаптивную нормализацию градиентов на основе скользящих средних;
- Adamax, Adagrad, RMSprop, Adadelta [88] — альтернативные адаптивные модификации [89], отличающиеся способами обновления весов.

Каждый из алгоритмов может быть выбран пользователем, параметры (скорость обучения, размер пакета) доступны для настройки в пользовательском интерфейсе.

### 4.3.3 Функции потерь

Система поддерживает несколько типов функций потерь, применяемых как для задач классификации, так и регрессии:

- Среднеквадратичная ошибка (MSE);
- Средняя абсолютная ошибка (MAE);
- Потеря Хубера [90] (Huber loss);
- Логарифмическая гиперболическая косинус-функция [91] (Log-Cosh).

Функции потерь реализованы вручную, с учётом числовой устойчивости и эффективности вычислений.

### 4.3.4 Объясняющее двоичное дерево

Для анализа принятия решений реализован алгоритм построения объясняющего двоичного дерева. Дерево формируется по реальным выходам модели на имеющихся данных, без рассмотрения всех гипотетических состояний пространства (что позволяет избежать экспоненциального роста сложности). Структура дерева отображается в виде таблицы ячеек, с указанием статистических характеристик.

### 4.3.5 Визуализация модели и метрик

Визуализация является ключевой частью интеллектуальной системы машинного обучения. Реализованы следующие возможности:

- Построение выходной поверхности модели (в двумерном или трёхмерном виде) с применением различных цветовых схем (рисунок 4.1), а также иерархическое разбиение пространства на ячейки (рисунок 4.2).
- Отображение архитектуры модели, включая весовые коэффициенты и их градиенты на каждом слое.
- Графики метрик (ошибки регрессии, классификации, доли отказов) на обучающих и тестовых выборках (рисунок 4.3а).
- Гистограммы распределения выходных значений модели по каждому из классов и фону (рисунок 4.3б).

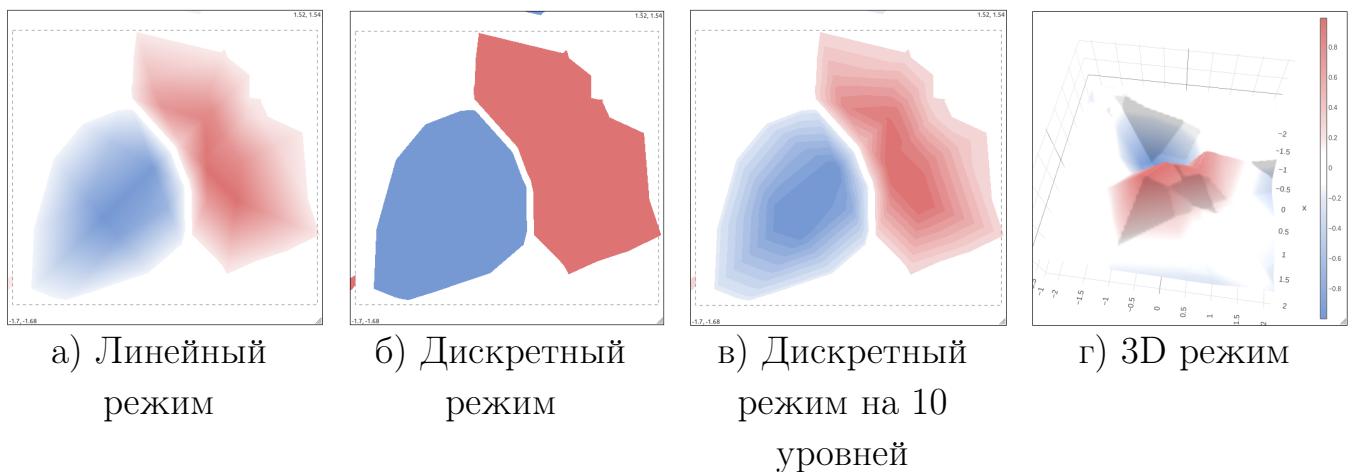
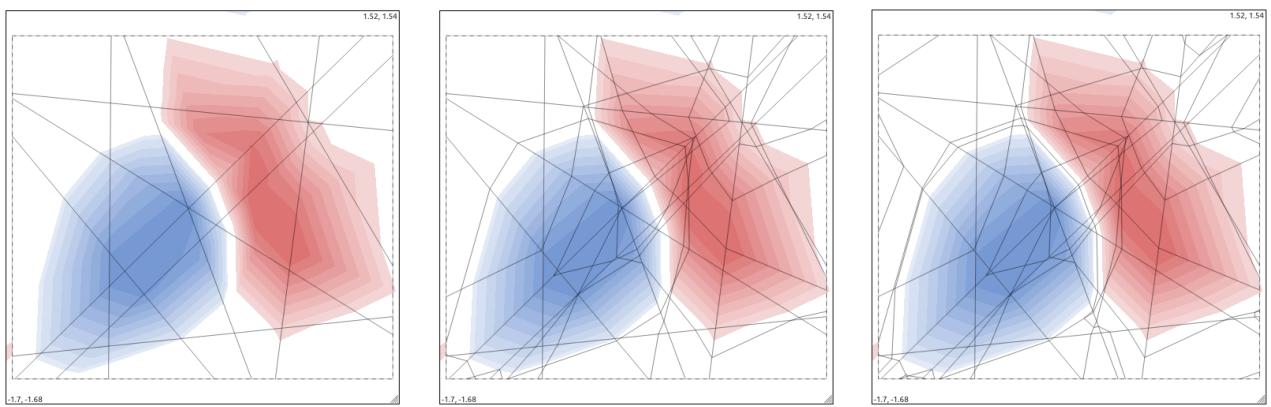


Рисунок 4.1 — Визуализация выхода модели

Для генерации графических представлений используются встроенные средства Canvas API и SVG с ручной реализацией всех визуальных компонентов. Визуализация обновляется в реальном времени по мере обучения модели. Для трёхмерного отображения выхода модели используется единственная внешняя зависимость Plotly.js [92].

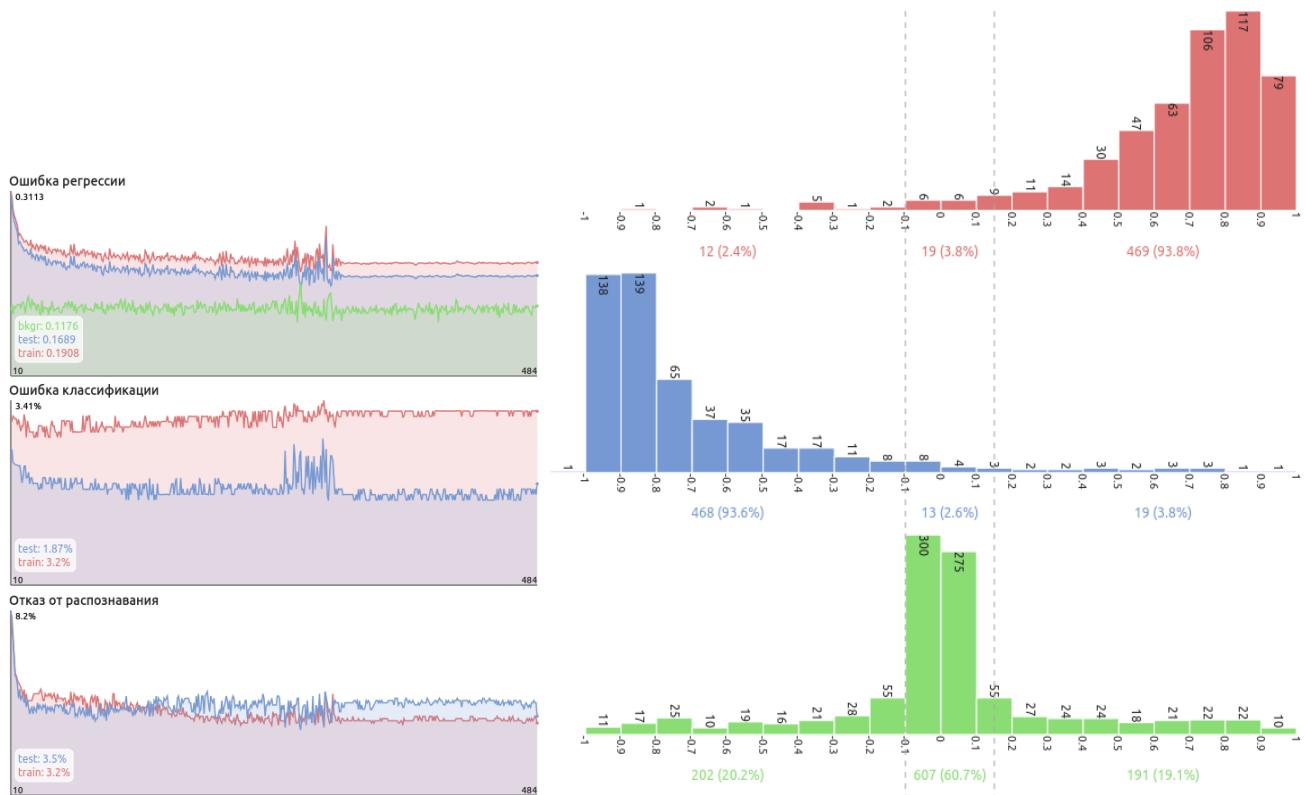


а) Ячейки первого скрытого слоя

б) Ячейки второго скрытого слоя

в) Ячейки выходного слоя

Рисунок 4.2 — Визуализация иерархического разбиения на ячейки



а) Метрики

б) Гистограммы выхода модели

Рисунок 4.3 — Визуализация метрик и гистограмм распределения выхода персептрона на обучающих данных

#### 4.3.6 Анализ отказов от классификации

Оценка надёжности классификации осуществляется с использованием порогового механизма, основанного на параметре доверия  $\beta$ . При выходном значении, не превышающем порог  $\beta$ , модель отказывается от классификации,

помечая объект как нераспознанный. Это позволяет существенно повысить доверие к решениям, принятым моделью, особенно в задачах с высокой ценой ошибки.

Реализована визуализация распределения отказов: гистограммы выходных значений и графики зависимости доли отказов и точности от выбранного порога.

#### 4.3.7 Генерация и модификация данных

Система включает инструменты генерации заранее подготовленных наборов данных для обучения и тестирования (рисунок 4.4). Пользователь может настраивать следующие параметры:

- Размер обучающей и тестовой выборки и их соотношение.
- Пропорции (баланс) классов.
- Доля ошибочно размеченных наблюдений.
- Нормализация, стандартизация и масштабирование признаков.

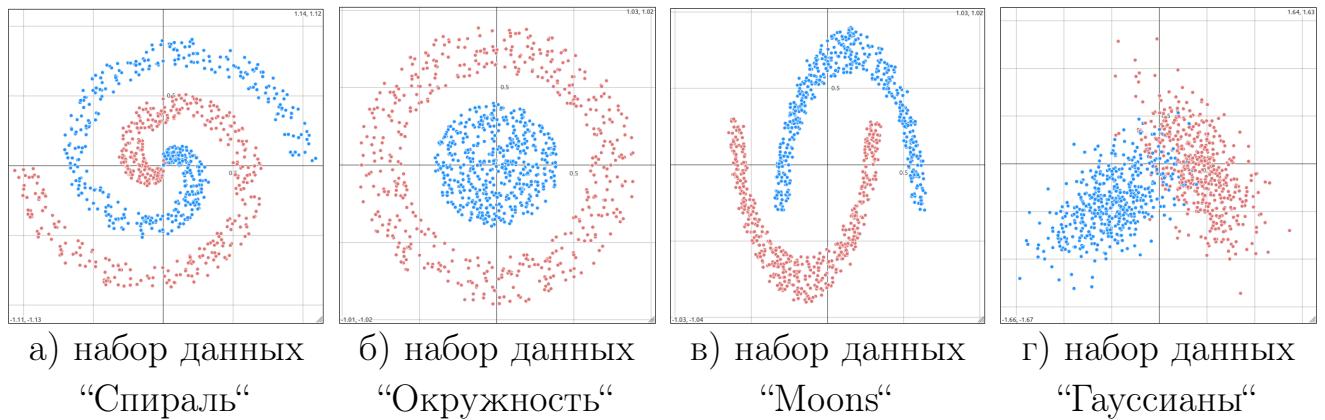


Рисунок 4.4 — Примеры наборов данных, доступных в интеллектуальной системе

#### 4.4 Структура и функциональные компоненты пользовательского интерфейса

Разработанная интеллектуальная система машинного обучения имеет модульную архитектуру пользовательского интерфейса, организованного в виде тематических вкладок. Такой подход позволяет изолировать различные этапы исследования и обеспечить пользователю интуитивно понятную навигацию между функциональными блоками. Вкладки интерфейса включают: “Данные”, “Обучение” и “Эксперименты”. Каждая из вкладок реализует отдельный аспект взаимодействия с системой.

**Вкладка “Данные“.** На данной вкладке пользователь может создавать обучающие и тестовые выборки, управляя их параметрами с помощью наглядных графических контроллеров, а также загружать данные из csv файлов. В числе доступных настроек:

- выбор числа объектов и размерности признаков;
- задание доли тестовых данных;
- задание доли объектов каждого класса;
- включение ошибок разметки;
- выполнение нормализации и стандартизации признаков.

Созданные или загруженные данные визуализируются на двумерной плоскости с использованием цветовой кодировки классов.

**Вкладка “Обучение“.** Данный раздел интерфейса предназначен для настройки архитектуры модели, параметров её обучения и визуального отображения различных элементов. Пользователь может:

- выбирать структуру многослойного персептрона (число слоёв, нейронов, функций активации);
- задавать параметры оптимизации (тип оптимизатора, скорость обучения, параметры регуляризации, функцию потерь);
- управлять отображением данных, сетки, ячеек и режимом выхода модели;
- запускать процесс обучения с возможностью пошагового анализа и прерывания.

Обучение сопровождается в реальном времени визуализацией различных характеристик: поверхности выходной функции модели, изменения функции ошибки, доли ошибок на обучающем и тестовом множествах, а также показателей отказа от классификации в зависимости от порога доверия.

**Вкладка “Эксперименты”.** Этот раздел агрегирует инструменты для проведения углублённого анализа результатов. В частности, доступны:

- визуализация объясняющего двоичного дерева;
- создание синтетических данных на основе обученной модели;
- обучение модели на данных с искусственно внесёнными пропусками.

Для каждой из опций предусмотрено графическое отображение и интуитивно понятная система управления с наглядными пояснениями каждого шага и полученных результатов. Это делает модуль эффективным инструментом для анализа моделей в условиях ограниченного количества данных, дисбаланса классов и наличия пропусков.

## 4.5 Алгоритмы визуализации

Графическая составляющая интеллектуальной системы реализована с использованием низкоуровневых возможностей браузера, таких как SVG и HTML5 Canvas API. Отказ от сторонних библиотек в пользу чистого JavaScript и CSS обусловлен требованиями к производительности, контролю над прорисовкой и необходимости точной синхронизации между визуальными компонентами.

Отрисовка данных, поверхности выхода модели, границ принятия решений, зон отказа от классификации, а также метрик качества осуществляется в режиме реального времени. Обновление изображений происходит по мере поступления новых данных или изменения параметров модели.

Основные принципы реализации визуализации включают:

- использование Canvas API для эффективной отрисовки цветных карт выхода модели;
- применение SVG для отрисовки больших массивов точек, осей, подписей, интерактивных маркеров и других элементов управления;

- разделение визуальных компонентов на независимые модули, каждый из которых регистрирует себя как подписчик событий модели и наборов данных;
- организация обмена сообщениями между компонентами через реализацию событийной модели на базе шаблона EventEmitter;
- использование аппаратного ускорения браузера при отрисовке и обновлении графических элементов;
- минимизация количества полных перерисовок за счёт дифференциального обновления слоёв.

Особое внимание уделяется синхронности всех отображаемых компонентов и их согласованности с текущим состоянием модели. Каждый визуализируемый объект автоматически обновляется при изменении состояния, что позволяет пользователю в реальном времени отслеживать последствия своих действий.

## 4.6 Примеры использования

Интеллектуальная система машинного обучения была 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.

### 4.6.1 Бинарная классификация

Для проверки способности модели к нелинейной аппроксимации границ принятия решений решается задача классификации двух переплетённых спиралей (рисунок 4.5). 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.

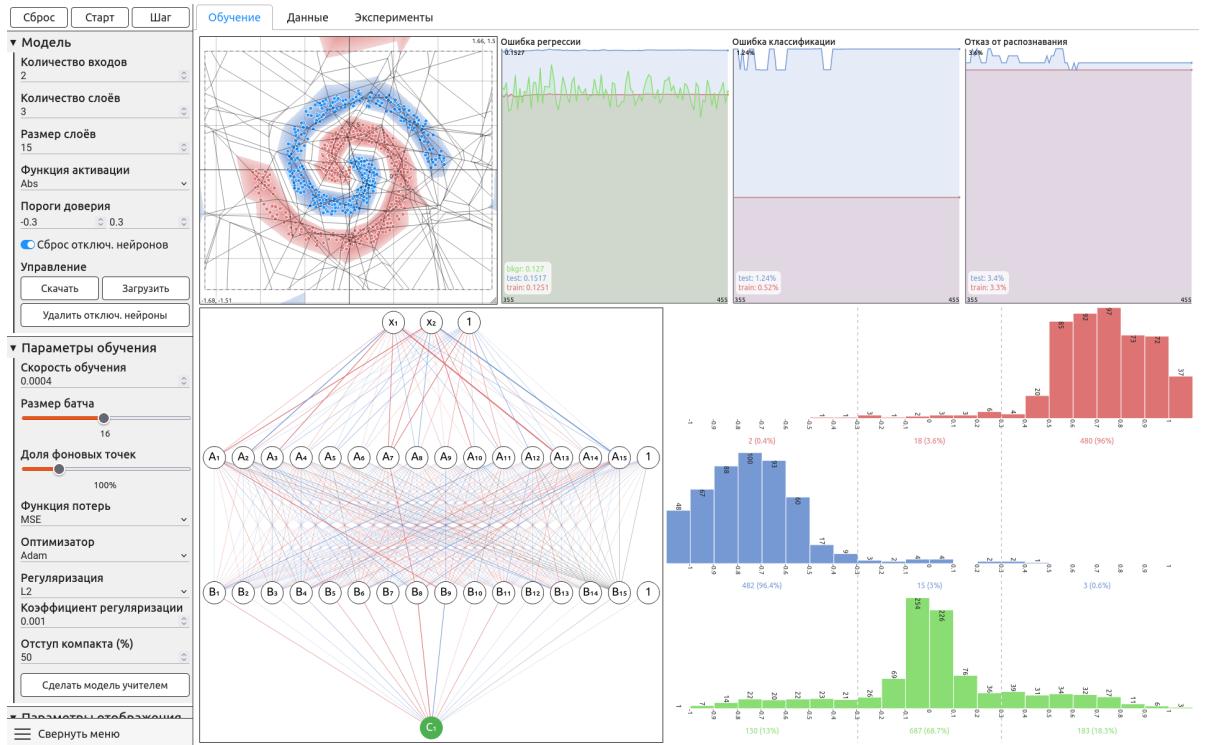


Рисунок 4.5 — Пример выполнения бинарной классификации

#### 4.6.2 Унарная классификация

Режим унарной классификации допускает обучение модели только по положительным примерам, дополненным фоновыми объектами (рисунок 4.6). В качестве примера используется одна из спиралей из предыдущего эксперимента. Пользователь может задать уровень порога  $\beta$ , 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.

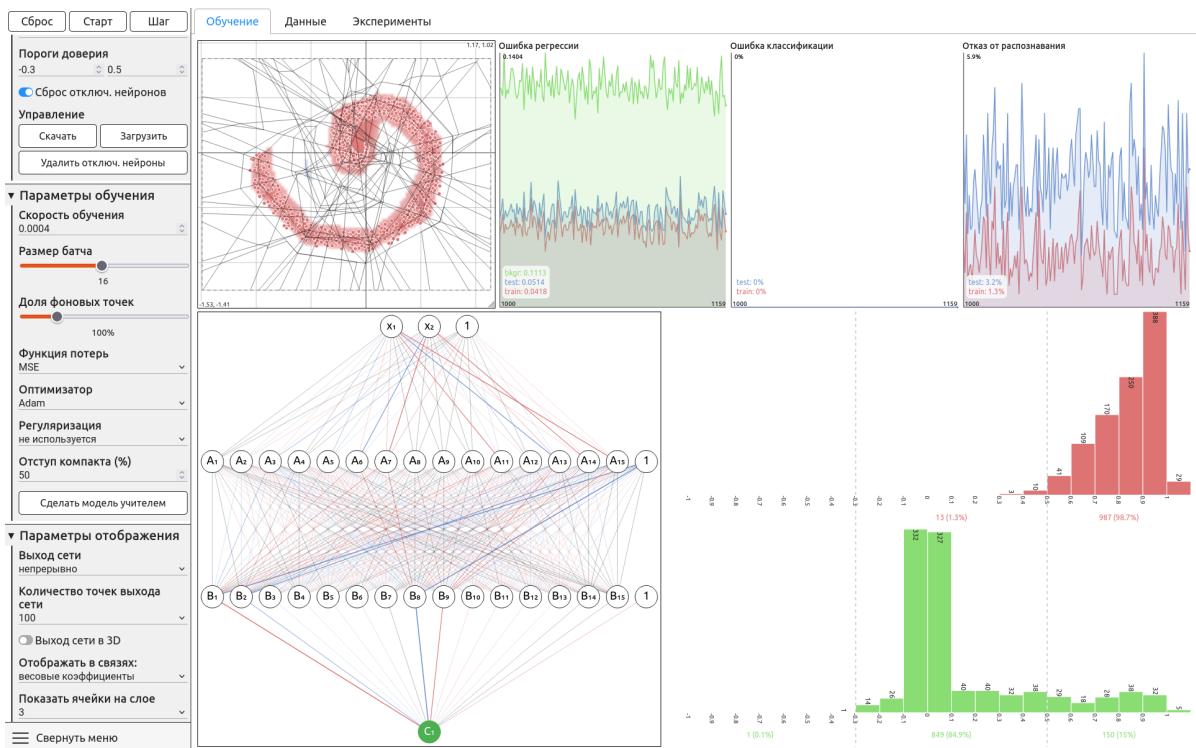


Рисунок 4.6 — Пример выполнения унарной классификации

#### 4.6.3 Создание синтетических данных

Одной из оригинальных функций системы является реализация метода синтетической генерации данных на основе репродукции, предложенного в рамках диссертационного исследования. Данный метод позволяет строить приближённую аппроксимацию распределения положительного класса в пространстве признаков, используя предварительно обученную унарную модель (рисунок 4.7).

Пользователь может интерактивно варьировать пороговое значение, наблюдать за плотностью отобранных объектов, а также визуализировать геометрию полученного множества. Данная возможность особенно важна при построении новых обучающих выборок, моделировании редких классов и оценке обобщающей способности модели на слабо покрытых областях признакового пространства.

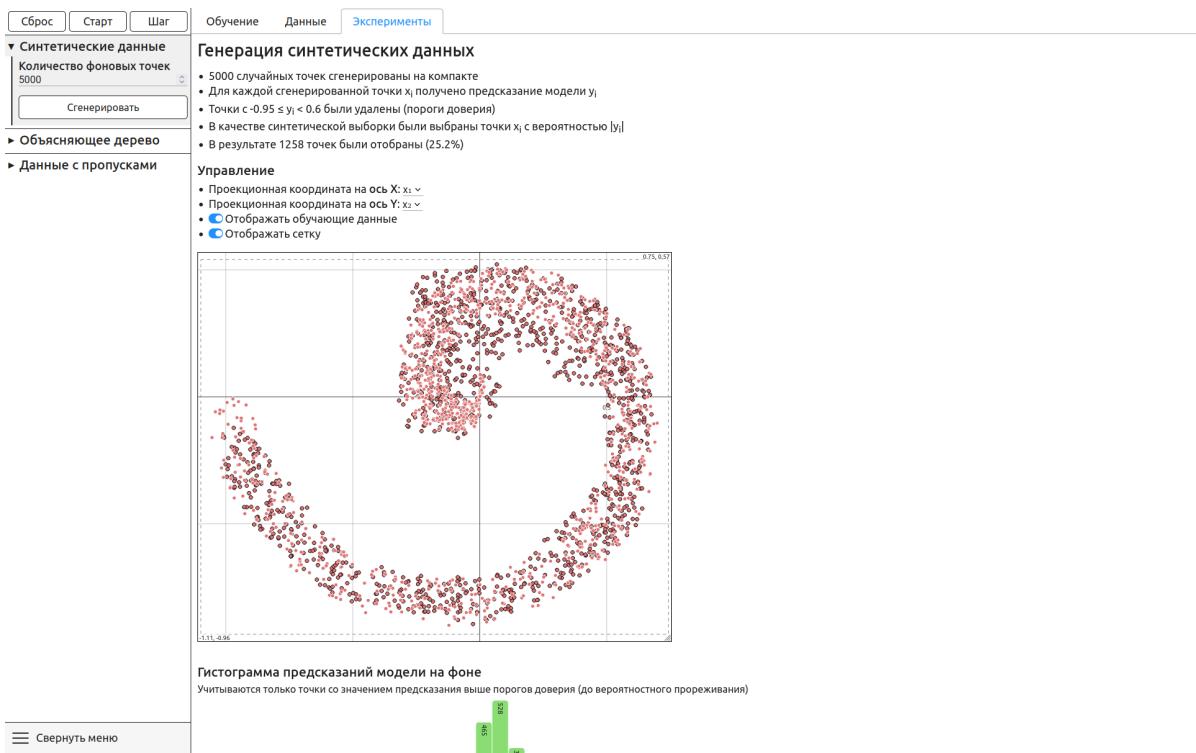


Рисунок 4.7 — Пример построения синтетических данных

#### 4.6.4 Построение объясняющего дерева решений

Одним из компонентов системы является модуль построения объясняющего дерева решений, предназначенного для геометрического описания поведения и интерпретации решения обученного персептрона (рисунок 4.8). 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.

### 4.7 Роль интеллектуальной системы в исследовании

Разработанная интеллектуальная система машинного обучения стала ключевым инструментом, обеспечивающим как реализацию предложенных в диссертации методов, так и всестороннюю экспериментальную проверку их свойств. Её архитектура ориентирована на гибкую настройку параметров моде-

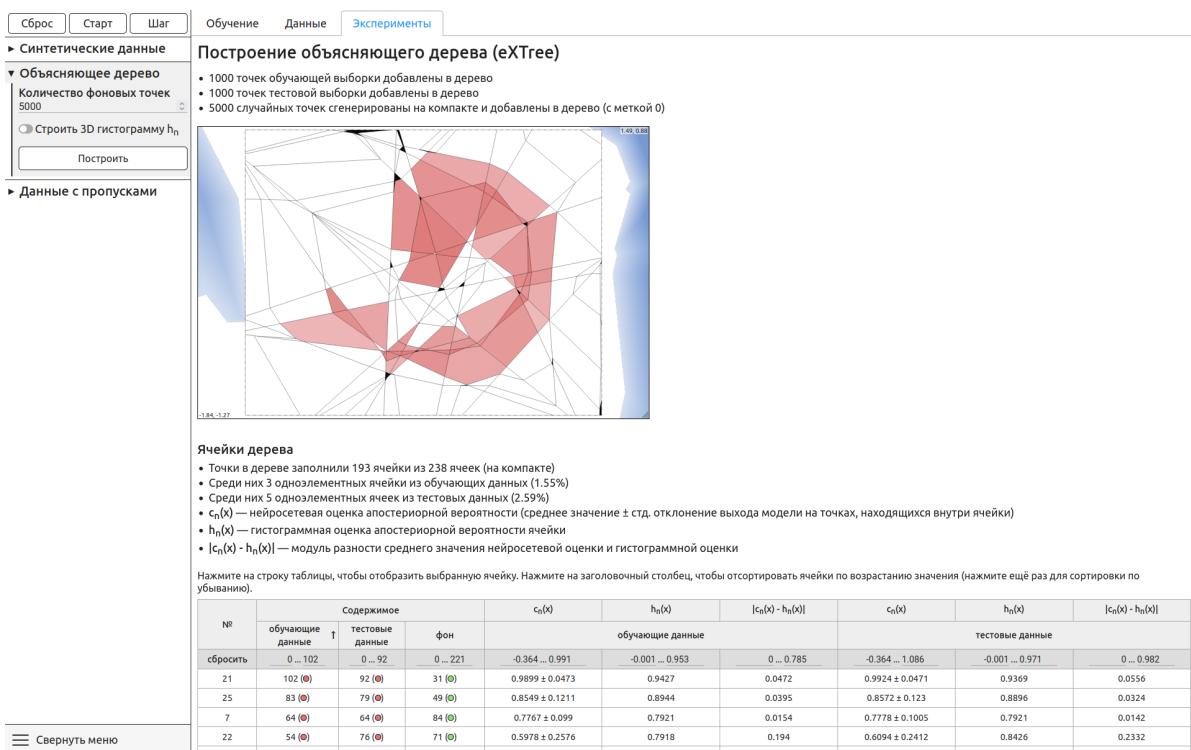


Рисунок 4.8 — Пример работы с объясняющим деревом

лей, визуальный контроль над процессом обучения, а также глубокий анализ поведения классификатора в условиях, приближённых к реальным сценариям применения.

Интеллектуальная система позволила оперативно проверять влияние архитектурных параметров нейронной сети на её аппроксимационные способности. Использование многослойного персептрона в качестве доверенного аппроксиматора вероятности принадлежности объекта к классу потребовало детальной настройки числа слоёв, функции активации и порога отказа  $\beta$ . Все эти параметры доступны для интерактивного изменения в ходе экспериментов.

Интеллектуальная система была использована для воспроизведения поведения модели на данных с перекрытием классов и в условиях неопределённости. Визуализация границ принятия решений и отказа от классификации показала, как повышение порога  $\beta$  повышает надёжность предсказаний за счёт исключения сомнительных точек. Такие наблюдения трудно формализовать численно, но они критичны для практической интерпретации поведения модели.

Кроме того, интеллектуальная система оказалась удобной платформой для отладки и тестирования предложенной модификации классификатора в условиях ограниченного объёма данных и небалансированности классов. Возможность быстрой генерации обучающих выборок и отображения результатов

классификации в реальном времени позволила провести сотни запусков, лежащих в основе статистической оценки качества.

Таким образом, интеллектуальная система машинного обучения выполнила не только вспомогательную, но и методологически значимую роль, обеспечив воспроизводимость, наглядность и полноту исследования. Её использование позволило обосновать теоретические положения диссертации эмпирически, за счёт детального анализа поведения моделей на управляемых синтетических данных.

## Заключение

Основные результаты работы заключаются в следующем.

1. Разработан и реализован метод построения классификатора, обеспечивающего состоятельную аппроксимацию апостериорных вероятностей при дисбалансе классов и некомплектности данных за счёт использования модифицированного байесовского классификатора на основе многослойного персептрона.
2. Разработаны методы генерации синтетических табличных данных и обработка некомплектных табличных данных на основе предложенного метода построения классификатора.
3. Проведено экспериментальное исследование разработанных методов на модельных и прикладных данных для оценки устойчивости классификатора к дисбалансу классов, неполноте данных и корректности обработки объектов вне носителя обучающего распределения.
4. Разработана интеллектуальная система машинного обучения, реализующая предложенные методы и обеспечивающая решение задач классификации табличных данных в условиях дисбаланса классов, некомплектности данных и высокой неопределённости вне носителя распределения.

Полученные результаты относятся к направлениям исследований 4, 7, 8 и 9 паспорта специальности 2.3.5.

## Словарь терминов

**Классификация** – задача машинного обучения, в которой требуется отнести объект к одному из заранее определённых классов.

**Унарная классификация** – подход, при котором обучение производится только на объектах одного (целевого) класса, а остальные данные считаются фоновыми или неизвестными.

**Отказ от классификации** – механизм, позволяющий модели не принимать решение о принадлежности к какому-либо классу при низком уровне уверенности.

**Порог доверия  $\beta$**  – значение, определяющее минимальный уровень уверенности модели, при котором принимается решение о классификации объекта.

**Многослойный персептрон** – класс искусственных нейронных сетей, состоящий из нескольких слоёв нейронов, каждый из которых связан с предыдущим полносвязным образом. Используется для аппроксимации сложных функций.

**Полносвязный слой** – слой нейронной сети, в котором каждый нейрон соединён со всеми выходами предыдущего слоя.

**Аппроксимация** – приближённое представление функции, заданной неявно, с помощью некоторой модели, например нейросети или гистограммы.

**Градиентный спуск** – метод оптимизации, основанный на итеративном обновлении параметров модели в направлении антиградиента функции потерь.

**Оптимизаторы градиентного спуска** – алгоритмы, используемые для настройки параметров модели в процессе обучения.

**Синтетические данные** – искусственно сгенерированные данные, используемые для проверки гипотез, обучения моделей и проведения контролируемых экспериментов при отсутствии достаточного количества реальных данных.

**Объясняющее дерево** – структура, позволяющая в интерпретируемой форме представить поведение модели путём построения дерева решений на выходных значениях модели.

**Нормализация признаков** – преобразование признаков, приводящее их к единому масштабу для повышения устойчивости и скорости обучения.

## Список литературы

1. *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.
2. *Japkowicz, N.* The class imbalance problem: A systematic study [Текст] / N. Japkowicz, S. Stephen // Intelligent data analysis. — 2002. — Т. 6, № 5. — С. 429—449.
3. *Elkan, C.* The foundations of cost-sensitive learning [Текст] / C. Elkan // International joint conference on artificial intelligence. Т. 17. — Lawrence Erlbaum Associates Ltd. 2001. — С. 973—978.
4. *Bishop, C. M.* Pattern recognition and machine learning [Текст]. Т. 4 / C. M. Bishop, N. M. Nasrabadi. — Springer, 2006.
5. SMOTE: synthetic minority over-sampling technique [Текст] / N. V. Chawla [и др.] // Journal of artificial intelligence research. — 2002. — Т. 16. — С. 321—357.
6. *Little, R. J.* Statistical analysis with missing data. [Текст] / R. J. Little, D. B. Rubin. — 1995.
7. *Rudin, C.* Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead [Текст] / C. Rudin // Nature machine intelligence. — 2019. — Т. 1, № 5. — С. 206—215.
8. *Devroye, L.* A probabilistic theory of pattern recognition [Текст]. Т. 31 / L. Devroye, L. Györfi, G. Lugosi. — Springer Science & Business Media, 2013.
9. Explainable AI (XAI): Core ideas, techniques, and solutions [Текст] / R. Dwivedi [и др.] // ACM computing surveys. — 2023. — Т. 55, № 9. — С. 1—33.
10. *Воронцов, К.* Математические методы обучения по прецедентам (теория обучения машин) [Текст] / К. Воронцов // Москва. — 2011. — С. 119—121.
11. *Воронцов, К.* Лекции по статистическим (байесовским) алгоритмам классификации [Текст] / К. Воронцов // URL: <http://www.ccas.ru/voron/download/Bayes.pdf> (20.09. 2017). — 2008.

12. *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.
13. *Muhamedyev, R.* Machine learning methods: An overview [Текст] / R. Muhamedyev // Computer modelling & new technologies. — 2015. — Т. 19, № 6. — С. 14—29.
14. *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.
15. *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.
16. Generalized out-of-distribution detection: A survey [Текст] / J. Yang [и др.] // International Journal of Computer Vision. — 2024. — Т. 132, № 12. — С. 5635—5662.
17. *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.
18. 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.
19. *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.
20. Extrapolation of the Bayesian classifier with an unknown support of the two-class mixture distribution [Текст] / K. S. Lukyanov [et al.] // Russian Mathematical Surveys. — 2024. — Vol. 79, no. 6. — P. 991—1015.
21. *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.

22. *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.
23. *Devroye, L.* Nonparametric density estimation [Текст] / L. Devroye // The L\_1 View. — 1985.
24. The elements of statistical learning [Текст] / T. Hastie, R. Tibshirani, J. Friedman [и др.]. — 2009.
25. *Devroye, L.* The Regular Histogram Rule [Текст] / L. Devroye, L. Györfi, G. Lugosi // A Probabilistic Theory of Pattern Recognition. — Springer, 1996. — C. 133—145.
26. *Devroye, L.* Consistency of the k-nearest neighbor rule [Текст] / L. Devroye, L. Györfi, G. Lugosi // A Probabilistic Theory of Pattern Recognition. — Springer, 1996. — C. 169—185.
27. *Wand, M. P.* Kernel smoothing [Текст] / M. P. Wand, M. C. Jones. — CRC press, 1994.
28. A distribution-free theory of nonparametric regression [Текст] / L. Györfi [и др.]. — Springer, 2002.
29. *Cortes, C.* Support-vector networks [Текст] / C. Cortes, V. Vapnik // Machine learning. — 1995. — Т. 20, № 3. — С. 273—297.
30. *Steinwart, I.* Support vector machines [Текст] / I. Steinwart, A. Christmann. — Springer Science & Business Media, 2008.
31. *Murtagh, F.* Multilayer perceptrons for classification and regression [Текст] / F. Murtagh // Neurocomputing. — 1991. — Т. 2, № 5/6. — С. 183—197.
32. *Cybenko, G.* Approximation by superpositions of a sigmoidal function [Текст] / G. Cybenko // Mathematics of control, signals and systems. — 1989. — Т. 2, № 4. — С. 303—314.
33. *Amari, S.-i.* Backpropagation and stochastic gradient descent method [Текст] / S.-i. Amari // Neurocomputing. — 1993. — Т. 5, № 4/5. — С. 185—196.
34. *Seiffert, U.* Multiple layer perceptron training using genetic algorithms. [Текст] / U. Seiffert // ESANN. — 2001. — С. 159—164.

35. *Song, Y.-Y.* Decision tree methods: applications for classification and prediction [Текст] / Y.-Y. Song, Y. Lu // Shanghai archives of psychiatry. — 2015.
36. *Девяткин, Д. А.* Построение ансамблей деревьев решений с использованием линейных и нелинейных разделителей [Текст] / Д. А. Девяткин. — .
37. *Salih, A. M.* Are Linear Regression Models White Box and Interpretable? [Текст] / A. M. Salih, Y. Wang // arXiv preprint arXiv:2407.12177. — 2024.
38. *Aly, M.* Survey on multiclass classification methods [Текст] / M. Aly // Neural Netw. — 2005. — Т. 19, № 1—9. — С. 2.
39. *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.
40. 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.
41. *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.
42. *Перминов, А. И.* SLAP—простая линейная атака на персепtron [Текст] / А. И. Перминов // Труды Института системного программирования РАН. — 2024. — Т. 36, № 3. — С. 83—92.
43. *Goodfellow, I. J.* Explaining and harnessing adversarial examples [Текст] / I. J. Goodfellow, J. Shlens, C. Szegedy // arXiv preprint arXiv:1412.6572. — 2014.
44. Towards deep learning models resistant to adversarial attacks [Текст] / A. Madry [и др.] // arXiv preprint arXiv:1706.06083. — 2017.
45. *Croce, F.* Sparse and imperceptible adversarial attacks [Текст] / F. Croce, M. Hein // Proceedings of the IEEE/CVF international conference on computer vision. — 2019. — С. 4724—4732.
46. *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.

47. Cats and dogs [Текст] / О. М. Parkhi [и др.] // 2012 IEEE conference on computer vision and pattern recognition. — IEEE. 2012. — С. 3498—3505.
48. *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.
49. Learning multiple layers of features from tiny images.(2009) [Текст] / A. Krizhevsky, G. Hinton [и др.]. — 2009.
50. *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.
51. *Bock, H. H.* Automatische Klassifikation. Theoretische und praktische Methoden zur Gruppierung und Strukturierung von Daten [Текст] / H. H. Bock // Studia Mathematica/Mathematische Lehrbücher. — 1974.
52. *Hartigan, J. A.* Clustering algorithms [Текст] / J. A. Hartigan. — John Wiley & Sons, Inc., 1975.
53. *Devroye, L. P.* The strong uniform consistency of kernel density estimates [Текст] / L. P. Devroye, T. J. Wagner // Multivariate Analysis V: Proceedings of the fifth International Symposium on Multivariate Analysis. Т. 5. — 1980. — С. 59—77.
54. *Devroye, L. P.* The strong uniform consistency of nearest neighbor density estimates [Текст] / L. P. Devroye, T. J. Wagner // The Annals of Statistics. — 1977. — С. 536—540.
55. *Wong, M. A.* A kth nearest neighbour clustering procedure [Текст] / M. A. Wong, T. Lane // Journal of the Royal Statistical Society: Series B (Methodological). — 1983. — Т. 45, № 3. — С. 362—368.
56. *Коваленко, А. П.* Метод унарной классификации [Текст] / А. П. Коваленко, А. И. Перминов // Материалы 34-й научно-технической конференции «Методы и технические средства обеспечения безопасности информации». — 2025.
57. *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.

58. *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.
59. 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.
60. *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.
61. *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.
62. Comprehensive exploration of synthetic data generation: A survey [Текст] / A. Bauer [и др.] // arXiv preprint arXiv:2401.02524. — 2024.
63. *Figueira, A.* Survey on synthetic data generation, evaluation methods and GANs [Текст] / A. Figueira, B. Vaz // Mathematics. — 2022. — Т. 10, № 15. — С. 2733.
64. *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.
65. *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.
66. Diffusion models for tabular data imputation and synthetic data generation [Текст] / M. Villaizán-Vallelado [и др.] // ACM Transactions on Knowledge Discovery from Data. — 2024.
67. *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.

68. *Perminov, A. I.* A Consistent Method for Generating Synthetic Tabular Data with a Fully Connected Neural Network [Текст] / A. I. Perminov, A. P. Kovalenko, D. Y. Turdakov // First Conference of Mathematics of AI. — 2025.
69. A method for stochastic optimization [Текст] / K. D. B. J. Adam [и др.] // arXiv preprint arXiv:1412.6980. — 2014. — Т. 1412, № 6.
70. *Habibi, O.* Imbalanced tabular data modelization using CTGAN and machine learning to improve IoT Botnet attacks detection [Текст] / O. Habibi, M. Chemmakha, M. Lazaar // Engineering Applications of Artificial Intelligence. — 2023. — Т. 118. — С. 105669.
71. *Ishfaq, H.* TVAE: Triplet-based variational autoencoder using metric learning [Текст] / H. Ishfaq, A. Hoogi, D. Rubin // arXiv preprint arXiv:1802.04403. — 2018.
72. Missing data in medical databases: Impute, delete or classify? [Текст] / F. Cismondi [и др.] // Artificial intelligence in medicine. — 2013. — Т. 58, № 1. — С. 63—72.
73. *Du, J.* Missing data problem in the monitoring system: A review [Текст] / J. Du, M. Hu, W. Zhang // IEEE Sensors Journal. — 2020. — Т. 20, № 23. — С. 13984—13998.
74. *Schafer, J. L.* Analysis of incomplete multivariate data [Текст] / J. L. Schafer. — CRC press, 1997.
75. 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.
76. *Dempster, A. P.* Maximum likelihood from incomplete data via the EM algorithm [Текст] / A. P. Dempster, N. M. Laird, D. B. Rubin // Journal of the royal statistical society: series B (methodological). — 1977. — Т. 39, № 1. — С. 1—22.
77. *Rubin, D. B.* An overview of multiple imputation [Текст] / D. B. Rubin // Proceedings of the survey research methods section of the American statistical association. Т. 79. — 1988. — С. 84.

78. *Moh, R. J. D.* Missing Values Imputation Using Principal Component Analysis Methods [Текст] : дис. ... канд. / Moh Rhoda Josephina Domebale. — Montana State University, 2024.
79. *Roskams-Hieter, B.* Leveraging variational autoencoders for multiple data imputation [Текст] / B. Roskams-Hieter, J. Wells, S. Wade // Joint European Conference on Machine Learning and Knowledge Discovery in Databases. — Springer. 2023. — С. 491—506.
80. *Перминов, А. И.* Метод обучения персептрана на табличных данных с пропусками [Текст] / А. И. Перминов, А. П. Коваленко, Д. Ю. Турдаков // Труды Института системного программирования РАН. — 2025. — Т. 1111, № 1111. — С. 1111—1111.
81. *Флэнаган, Д.* JavaScript [Текст] / Д. Флэнаган // Подробное руководство, Изд-во «Символ-Плюс». — 2013.
82. *Hickson, I.* Html5 [Текст] / I. Hickson, D. Hyatt // W3C working draft WD-Html5-20110525. — 2011. — Т. 53.
83. *Lunn, I.* CSS3 foundations [Текст] / I. Lunn. — John Wiley & Sons, 2012.
84. *Quint, A.* Scalable vector graphics [Текст] / A. Quint // IEEE MultiMedia. — 2003. — Т. 10, № 3. — С. 99—102.
85. *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.
86. *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.
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. *Zeiler, M. D.* Adadelta: an adaptive learning rate method [Текст] / M. D. Zeiler // arXiv preprint arXiv:1212.5701. — 2012.
89. *Ruder, S.* An overview of gradient descent optimization algorithms [Текст] / S. Ruder // arXiv preprint arXiv:1609.04747. — 2016.

90. *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. — C. 5261—5269.
91. *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.
92. Package ‘plotly’ [Текст] / C. Sievert [и др.] // R Foundation for Statistical Computing, Vienna. — 2021.

## Список рисунков

1.1	Кусочно-линейные функции активации . . . . .	22
1.2	Пример разбиения некоторым персепtronом с $L = 2, k = 6$ . . . . .	23
1.3	Пример вычисления $h_n^*(X)$ в некоторой ячейке $K_r$ . . . . .	24
1.4	Архитектура многослойного персептрана с $d = 2, L = 3, k = 7$ . . . . .	25
1.5	Пример eXBTree на основе персептрана с $d = 2, L = 1, k = 3$ . . . . .	27
1.6	Примеры, участвующие в атаке на многослойный персептран . . . . .	34
1.7	Схема матричной атаки . . . . .	35
1.8	Пример матричной атаки, $x \in [-1055, 926]$ . . . . .	36
1.9	Пример QR атаки . . . . .	37
1.10	Пример атаки на многослойный персептран на датасете Cat-vs-Dog [47] . . . . .	38
1.11	Пример генерации атакующих примеров . . . . .	39
1.12	Сравнение поведения классификаторов вне носителя . . . . .	42
1.13	Сравнение устойчивости классификаторов к backdoor-атаке . . . . .	43
1.14	Визуальное сравнение функций нейросетевой и гистограммной регрессий . . . . .	44
1.15	Влияние порога доверия $\beta$ на пространственное распределение классификационных решений . . . . .	46
2.1	Пример вычисления $h_n^*(X)$ в некоторой ячейке $K_r$ в унарном случае	50
2.2	Модельные ситуации для анализа метрик . . . . .	56
2.3	Оценка плотности одного класса с использованием унарной схемы .	58
2.4	Унарная классификация для двух классов . . . . .	59
2.5	Унарная классификация для четырёх классов с дисбалансом . . . . .	59
3.1	Схематичное представление задачи создания синтетических данных	61
3.2	Использованные наборы данных для построения репродукционных выборок (спираль, два квадрата и гауссиан) . . . . .	63
3.3	результаты эксперимента с синтетическими данными . . . . .	64
3.4	Схема обучения $c_n^{(j)}(X)$ по некомплектным данным . . . . .	71
3.5	Использованные наборы для обучения модели на данных с пропусками . . . . .	73
4.1	Визуализация выхода модели . . . . .	82
4.2	Визуализация иерархического разбиения на ячейки . . . . .	83

4.3	Визуализация метрик и гистограмм распределения выхода персептрона на обучающих данных . . . . .	83
4.4	Примеры наборов данных, доступных в интеллектуальной системе .	84
4.5	Пример выполнения бинарной классификации . . . . .	88
4.6	Пример выполнения унарной классификации . . . . .	89
4.7	Пример построения синтетических данных . . . . .	90
4.8	Пример работы с объясняющим деревом . . . . .	91

**Список таблиц**

1	Результаты применения SLAP атаки . . . . .	40
2	Оценка метода репродукции для заполнения пропусков на наборе данных “Гауссианы“ . . . . .	75
3	Оценка метода репродукции для заполнения пропусков на наборе данных “Спираль“ . . . . .	75
4	Оценка метода репродукции для заполнения пропусков на наборе данных “Кольцо и круг“ . . . . .	75

**Приложение А**

Свидетельства о государственной регистрации программ и ЭВМ



ФЕДЕРАЛЬНАЯ СЛУЖБА  
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ  
**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),  
Лукьянин 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 постфактуум 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



ФЕДЕРАЛЬНАЯ СЛУЖБА  
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ  
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



ФЕДЕРАЛЬНАЯ СЛУЖБА  
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ  
**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



ФЕДЕРАЛЬНАЯ СЛУЖБА  
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ  
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 устраниния 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 [20] совместно с Лукьяновым К.С., Яськовым П.А., Коваленко А.П. и Турдаковым Д.Ю..

**ДОПИСАТЬ ДОКАЗАТЕЛЬСТВО**