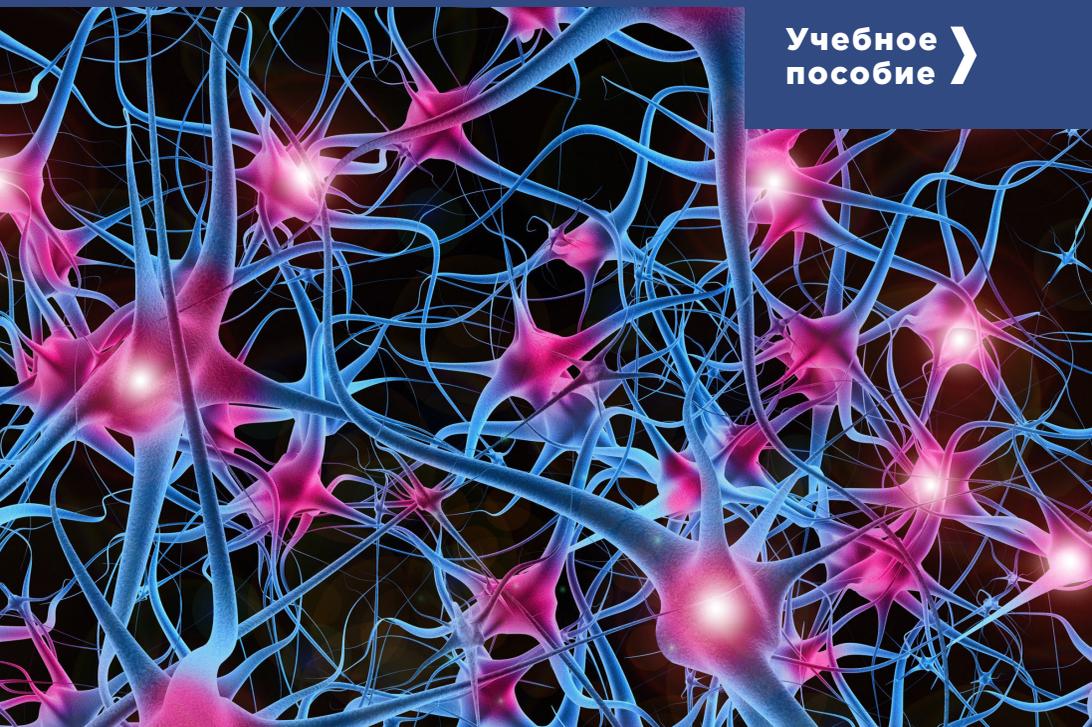


МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ
НЕКОММЕРЧЕСКОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
УЧЕБНО-НАУЧНО-ПРОИЗВОДСТВЕННЫЙ КОМПЛЕКС
«МЕЖДУНАРОДНЫЙ УНИВЕРСИТЕТ КЫРГЫЗСТАНА»

Е. Ю. Савченко

Основы и применение НЕЙРОННЫХ СЕТЕЙ: теория, практика, инновации

Учебное
пособие >



Бишкек • 2024

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ
НЕКОММЕРЧЕСКОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
УЧЕБНО-НАУЧНО-ПРОИЗВОДСТВЕННЫЙ КОМПЛЕКС
«МЕЖДУНАРОДНЫЙ УНИВЕРСИТЕТ КЫРГЫЗСТАНА»

Е. Ю. Савченко

Основы и применение НЕЙРОННЫХ СЕТЕЙ: теория, практика, инновации

**Учебное
пособие** 

Бишкек • 2024

УДК 004.6
ББК 16.6
С 13

Рекомендовано к изданию ученым советом УНПК МУК

Допущено Министерством образования и науки
Кыргызской Республики в качестве учебного пособия
для студентов высших учебных заведений.

Приказ МОиН КР № 670/1, от 26.04.2024 г.

Рецензенты:

Каплина Т. Ю. — канд. техн. наук, доцент кафедры «Не-традиционные и возобновляемые источники энергии» КРСУ им. Б. Ельцина.

Мусакулова Ж. А. — канд. техн. наук, доцент кафедры «Компьютерных информационных систем и управления» УНПК МУК.

Савченко Е. Ю.

С 13 **Основы и применение нейронных сетей: теория, практика, инновации.** Учеб. пособие / Е. Ю. Савченко. – Б.: Нео Принт, 2024. – 116 с.: ил.

ISBN 978-9967-9517-3-0 (print)

ISBN 978-9967-9517-4-7 (online)

Учебное пособие «Основы и применение нейронных сетей: Теория, практика, инновации» является руководством по изучению и применению нейронных сетей в различных сферах искусственного интеллекта. Оно охватывает широкий диапазон тем, начиная от основ и истории развития нейронных сетей, до современных технологий и инновационных применений, таких как генеративные сети (GAN) и обработка естественного языка (NLP). Пособие содержит теоретические основы, практические задания «Нейро-практикум» для закрепления знаний, а также обзор последних тенденций и будущего развития нейронных сетей. Это руководство станет ценным ресурсом для студентов, преподавателей и профессионалов в области ИИ, желающих глубже изучить эту быстро развивающуюся область.

УДК 004.6

ББК 16.6

ISBN 978-9967-9517-3-0 (print)

ISBN 978-9967-9517-4-7 (online)

© Савченко Е. Ю., 2024.

Оглавление

ВВЕДЕНИЕ. Нейронные сети	7
ГЛАВА 1. Основы нейронных сетей	10
1.1. История развития нейронных сетей	10
Современность — Глубокое обучение и нейронные сети	11
1.2. Основные проблемы, решаемые искусственными нейронными сетями	11
Классификация образов	11
Кластеризация/категоризация	12
Аппроксимация функций	12
Предсказание/прогноз	12
Оптимизация	12
1.3. Биологический нейрон и его математическая модель	12
Искусственный нейрон	14
Типы функций активации	17
1.4. Классификация нейронных сетей и их свойства	19
Представление знаний в нейронных сетях	21
ГЛАВА 2. Многослойные нейронные сети прямого распространения	23
2.1. Обучение нейронных сетей. Парадигмы обучения	24
Рассмотрим парадигму обучения нейронной сети с учителем	25
2.2. Обучение, основанное на коррекции ошибок	27
Многослойная сеть прямого распространения	28
2.3. Алгоритм обратного распространения ошибки (<i>back propagation</i>)	30
Алгоритм обучения сети обратного распространения	35

2.4. Последовательный и пакетный режимы обучения	
алгоритма обратного распространения ошибки	38
Последовательный режим	38
Пакетный режим	39
Последовательный (онлайн) режим обучения	40
Пакетный режим обучения	40
Мини-пакетное обучение (<i>Mini-Batch Training</i>)	41
ГЛАВА 3. Самоорганизующиеся нейронные сети	42
Рассмотрим парадигму обучения нейронной сети	
без учителя	42
3.1. Принципы работы	43
Важность карты признаков	46
3.2. Примеры и применение	47
Вывод	47
ГЛАВА 4. Генеративные нейронные сети (GAN)	48
4.1. Принцип работы генеративных нейронных сетей	48
4.2. Приложения GAN: синтез изображений,	
структур и другие возможности	49
4.3. Приложение GAN для синтеза изображений	
на примере DALL·E	51
ГЛАВА 5. Нейронные сети для обработки	
естественного языка (NLP)	53
5.1. Типы нейронных сетей для обработки естественного	
языка NLP	53
5.2. Трансформеры	54
5.3. Применение NLP	56
5.4. Приложение ChatGPT	57
5.5. ChatGPT для решения задач по программированию ...	58
ГЛАВА 6. Практические аспекты обучения	
нейронных сетей	61
6.1. Подготовка данных и предобработка	61
6.2. Выбор функций ошибок и метрик качества	62

Различие с функцией ошибки	65
ГЛАВА 7. Приложения и последние тенденции	66
7.1. Нейронные сети в медицине	66
7.2. Нейронные сети в экономике	68
7.3. Автоматическое управление и умные системы	69
7.4. Развитие нейросетей в будущем	70
ЗАКЛЮЧЕНИЕ. Подведение итогов	72
Рекомендации по дополнительному изучению	73
Основные книги, статьи и учебные пособия	74
Практический опыт и проекты	76
Постоянное обучение и развитие	76
ЛИТЕРАТУРА. Список источников	77
ПРИЛОЖЕНИЕ 1. Тренировочные задания «Нейро-практикум»	79
ПРИЛОЖЕНИЕ 2. Зачем изучать нейронные сети	112
ПРИЛОЖЕНИЕ 3. Ключевые термины и определения	114

ВВЕДЕНИЕ

Нейронные сети

В современном информационном обществе нейронные сети стали одной из самых актуальных и мощных технологий, преобразовавших множество сфер жизни. Они стали двигателем инноваций в области искусственного интеллекта и машинного обучения, совершив невероятный скачок в развитии за последние десятилетия. Нейронные сети нашли свое применение в медицине, финансах, компьютерном зрении, обработке естественного языка, управлении роботами и многих других областях, что делает их ключевым инструментом для исследователей, инженеров и разработчиков.

Это учебное пособие посвящено изучению нейронных сетей и их применению. Здесь мы представим вам обширный и информативный материал, который поможет вам овладеть основами нейронных сетей и развить навыки в создании и применении различных видов нейронных архитектур.

Цели и задачи учебного пособия по нейронным сетям:

ЦЕЛИ:

Обеспечение понимания основ нейронных сетей

Пособие направлено на предоставление читателю глубокого понимания основных концепций, принципов и математических аспектов, лежащих в основе работы нейронных сетей

Обучение практическим навыкам

Пособие предоставляет практические рекомендации и примеры для реализации, обучения и оценки нейронных сетей на практике

Освоение различных архитектур и приложений	Читатели узнают о разнообразных архитектурах нейронных сетей и их приложениях в различных областях, таких как компьютерное зрение, обработка текста и звука, автономная навигация и другие
Поддержка обучения и исследования	Пособие может использоваться студентами, преподавателями, исследователями и всеми, кто интересуется нейронными сетями, в качестве основного ресурса для изучения и проведения исследований.
ЗАДАЧИ:	
Объяснение ключевых понятий	Пособие объясняет базовые и продвинутое понятия в области нейронных сетей, начиная с принципов работы одного нейрона и заканчивая глубокими архитектурами
Предоставление практических инструкций	Пособие предоставляет практические шаги и рекомендации по созданию, обучению и тестированию нейронных сетей
Разработка навыков решения задач	Читатели учатся применять знания и навыки, приобретенные из пособия, для решения конкретных задач, таких как классификация, аппроксимация, сегментация и др.
Изучение актуальных тенденций и исследований	Пособие включает информацию о последних тенденциях, исследованиях и применениях нейронных сетей, что позволяет читателям оставаться в курсе современных разработок в этой области
Содействие образованию и научным исследованиям	Пособие служит инструментом для образования и научных исследований, способствуя распространению знаний и внедрению нейронных сетей в различные области деятельности

Цели и задачи учебного пособия по нейронным сетям нацелены на то, чтобы сделать эту сложную область доступной и понятной для широкой аудитории, а также вдохновить читателей на изучение и применение нейронных сетей в своей работе и исследованиях.

Мы призываем вас начать этот увлекательный путь в мир нейронных сетей и искусственного интеллекта с нами. Независимо от вашего уровня подготовки и опыта, это пособие предоставит вам необходимые знания и инструменты для успешного освоения этой захватывающей области. Начнем с основ и постепенно продвинемся к более сложным концепциям и практическим применениям. Добро пожаловать в мир нейронных сетей!

ГЛАВА 1

Основы нейронных сетей

1.1. История развития нейронных сетей

История развития нейронных сетей представляет собой увлекательный путь от первых попыток создания искусственных нейронных систем до современных глубоких нейронных сетей.

Вот краткий обзор ключевых этапов в развитии этой области:

1940-е годы Рождение нейронных сетей	В начале 1940-х годов Уоррен Маккаллок и Уолтер Питц создали первую математическую модель биологического нейрона, что послужило отправной точкой для исследований в области искусственных нейронных сетей
1950-е годы Перцептрон и период оптимизма	Фрэнк Розенблатт представил модель перцептрона, однослойной нейронной сети, способной решать линейно разделимые задачи. Это привело к оптимизму относительно способности нейронных сетей решать сложные проблемы
1960-е годы Ограничения и забвение	Мин Мински и Сеймур Пейперт опубликовали книгу «Перцептроны», в которой указали на ограничения перцептрона, включая неспособность решать задачи, нелинейно разделимые входные данные. Это привело к временному забвению нейронных сетей в те годы.
1980-е годы Возвращение с обратным распространением ошибки	В 1986 году Дэвид Румельхарт, Джеффри Хинтон и Рональд Уильямс представили метод обратного распространения ошибки (Backpropagation), который позволил обучать многослойные нейронные сети. Это возродило интерес к нейронным сетям

1990-е годы Возникновение сверточных нейронных сетей	Ян ЛеКун разработал сверточные нейронные сети (CNN), которые успешно применяются в обработке изображений и распознавании образов. CNN стали ключевым элементом в сфере компьютерного зрения
2000-е годы Развитие рекуррентных нейронных сетей	Рекуррентные нейронные сети (RNN) получили широкое применение в обработке последовательных данных, таких как естественный язык и временные ряды. Методы обучения LSTM и GRU улучшили эффективность RNN.

Современность — Глубокое обучение и нейронные сети

С развитием вычислительных ресурсов и появлением глубоких архитектур нейронных сетей, таких как глубокие сверточные нейронные сети (*Deep CNN*) и рекуррентные нейронные сети с долгой краткосрочной памятью (*LSTM*), нейронные сети стали неотъемлемой частью машинного обучения и искусственного интеллекта.

История нейронных сетей отражает их постоянное развитие и эволюцию от простых моделей к сложным, способным решать разнообразные задачи. Сегодня нейронные сети являются одной из ключевых технологий в области искусственного интеллекта и машинного обучения.

1.2. Основные проблемы, решаемые искусственными нейронными сетями

Классификация образов

Задача состоит в указании принадлежности входного образа, представленного вектором признаков, одному или нескольким предварительно определенным классам. К известным приложениям относятся распознавание букв, распознавание речи, классификация сигнала электрокардиограммы, классификация клеток крови.

Кластеризация / категоризация

При решении задачи кластеризации, которая известна также как классификация образов без учителя, отсутствует обучающая выборка с образцами классов. Алгоритм кластеризации основан на подобии образов и размещает близкие образы в один кластер. Известны случаи применения кластеризации для извлечения знаний, сжатия данных и исследования свойств данных.

Аппроксимация функций

Предположим, что имеется обучающая выборка $((x_1, y_2), (x_2, y_2), \dots, (x_n, y_m))$, которая генерируется неизвестной функцией, искаженной шумом. Задача аппроксимации состоит в нахождении оценки этой функции.

Предсказание / прогноз

Пусть заданы N дискретных отсчетов $\{y(t_1), y(t_2), \dots, y(t_k)\}$ в последовательные моменты времени t_1, t_2, \dots, t_k . Задача состоит в предсказании значения $y(t_{k+1})$ в момент t_{k+1} . Прогнозы имеют значительное влияние на принятие решений в бизнесе, науке и технике.

Оптимизация

Многочисленные проблемы в математике, статистике, технике, науке, медицине и экономике могут рассматриваться как проблемы оптимизации. Задачей оптимизации является нахождение решения, которое удовлетворяет системе ограничений и максимизирует или минимизирует целевую функцию.

1.3. Биологический нейрон и его математическая модель

Нервная система и мозг человека состоят из нейронов, соединенных между собой нервными волокнами. Нервные волокна способны передавать электрические импульсы между нейронами. Все процессы передачи раздражений от кожи, ушей

и глаз к мозгу, процессы мышления и управления действиями — все это реализовано в живом организме как передача электрических импульсов между нейронами. *Нейрон* (нервная клетка) является особой биологической клеткой, которая обрабатывает информацию (рис. 1.1.). Он состоит из *тела*, или *сомы*, и отростков нервных волокон двух типов — *дендритов*, по которым принимаются импульсы, и единственного *аксона*, по которому нейрон может передавать импульс. Тело нейрона включает *ядро*, которое содержит информацию о наследственных свойствах, и *плазму*, обладающую молекулярными средствами для производства необходимых нейрону материалов. Нейрон получает сигналы (импульсы) от аксонов других нейронов через дендриты (приемники) и передает сигналы, сгенерированные телом клетки, вдоль своего аксона (передатчика), который в конце разветвляется на волокна. На окончаниях этих волокон находятся специальные образования — *синапсы*, которые влияют на величину импульсов.

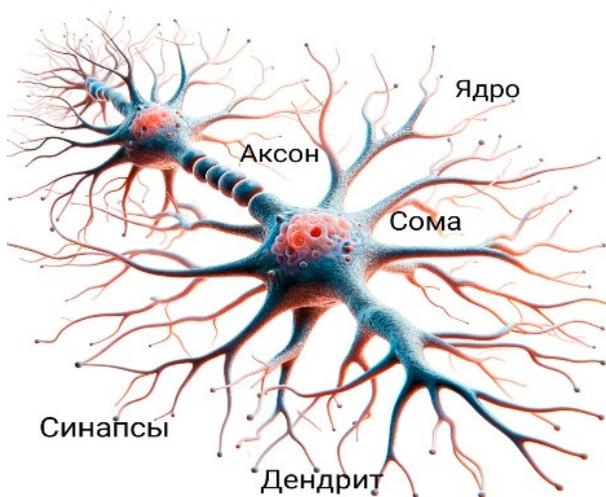


Рис. 1.1. Взаимосвязь биологических нейронов

Искусственный нейрон

Каждый нейрон характеризуется своим текущим состоянием по аналогии с нервными клетками головного мозга, которые могут быть возбуждены или заторможены. Он обладает группой *синапсов* — однонаправленных входных связей, соединенных с выходами других нейронов, а также имеет *аксон* — выходную связь данного нейрона, с которой сигнал (возбуждения или торможения) поступает на синапсы следующих нейронов. Общий вид искусственного нейрона приведен на *рисунке 1.2*.

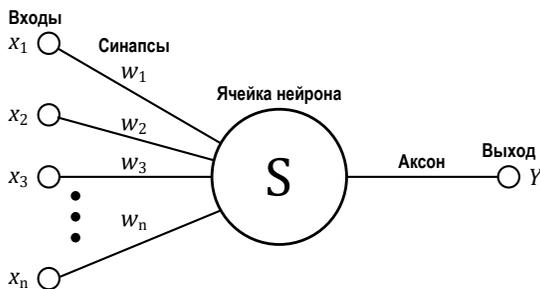


Рис. 1.2. Искусственный нейрон

На *рисунке 1.3*, представлена модель нейрона, лежащего в основе искусственных нейронных сетей.

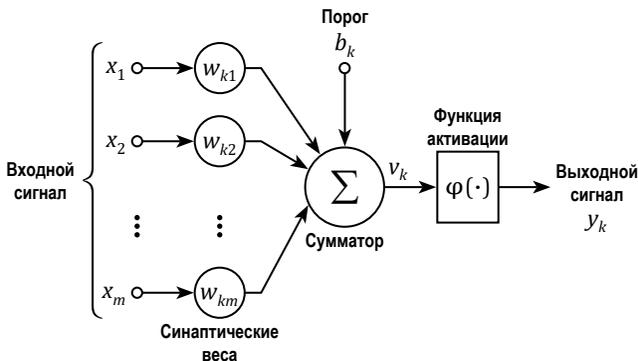


Рис. 1.3. Нелинейная модель нейрона

В этой модели можно выделить три основных элемента.

1. Набор синапсов или связей, каждый из которых характеризуется своим весом или силой. В частности, сигнал x_j на входе синапса j , связанного с нейроном k , умножается на вес w_{kj} . Важно обратить внимание на то, в каком порядке указаны индексы синаптического веса w_{kj} . Первый индекс относится к рассматриваемому нейрону, второй — ко входному окончанию синапса, с которым связан данный вес. В отличие от синапсов головного мозга, синапсы в искусственных нейронных сетях могут принимать как положительные, так и отрицательные значения.
2. Сумматор складывает входные сигналы, взвешенные относительно соответствующих синапсов нейрона. Эту операцию можно описать как линейную комбинацию.
3. Функция активации ограничивает амплитуду выходного сигнала нейрона. Эта функция также называется функцией сжатия. Обычно нормированный диапазон амплитуд выхода нейрона лежит в интервале $[0,1]$, $[-1,1]$. В модель нейрона на *рис. 1.3*. включен пороговый элемент, который обозначен символом b_k . Эта величина отражает увеличение или уменьшение входного сигнала, подаваемого на функцию активации.

В математическом представлении функционирование нейрона k можно описать следующей парой уравнений:

$$u_k = \sum_{j=0}^m w_{kj}x_j, \tag{1}$$

$$y_k = \varphi(u_k + b_k). \tag{2}$$

Где x_1, x_2, \dots, x_m — входные сигналы;

$w_{k1}, w_{k2}, \dots, w_{km}$ — синаптические веса нейрона k ;

u_k — линейная комбинация входных воздействий;

b_k — порог;
 $\varphi(\bullet)$ — функция активации;
 y_k — выходной сигнал нейрона.

В модели, показанной на *рис. 1.3.*, постсинаптический потенциал вычисляется следующим образом:

$$v_k = u_k + b_k \quad (3)$$

В частности, в зависимости от того, какое значение принимает порог b_k , положительное или отрицательное, индуцированное локальное поле v_k нейрона v_k изменяется так, как показано на *рисунке 1.4.*

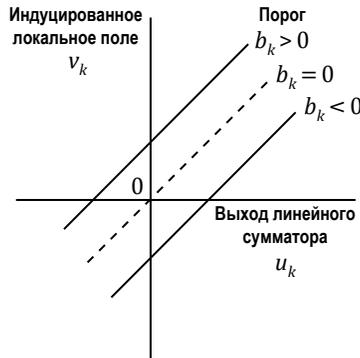


Рис. 1.4. Аффинное преобразование, вызванное наличием порога

Обратите внимание, что в точке, где $u_k = 0$, $v_k = b_k$. Порог b_k является внешним параметром искусственного нейрона k . Тогда формулу (1) можно преобразовать к следующему виду:

$$u_k = \sum_{j=0}^m w_{kj} x_j, \quad (4)$$

$$y_k = \varphi(u_k). \quad (5)$$

В выражении 4 добавился новый синапс. Его входной сигнал равен:

$$x_0 = +1, \tag{6}$$

а его вес:

$$w_{k0} = b_k. \tag{7}$$

Тогда модель нейрона будет выглядеть следующим образом *рисунок 1.5*.

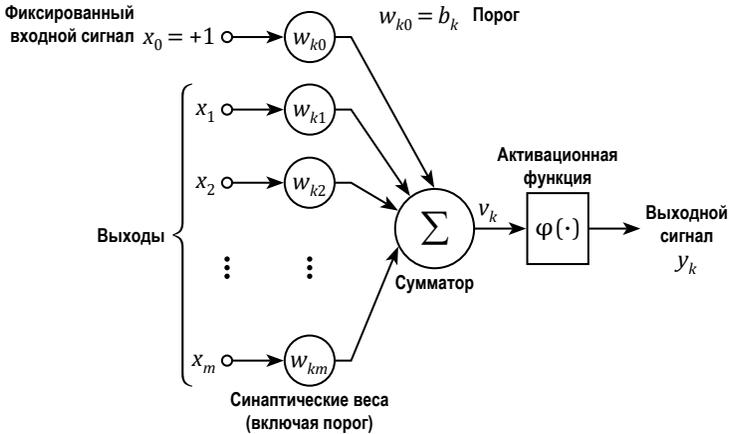


Рис. 1.5 Нелинейная модель нейрона

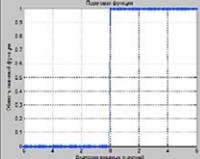
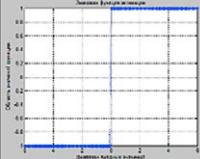
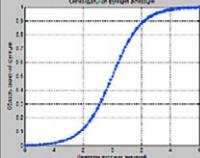
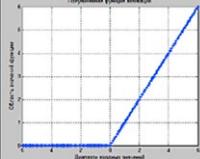
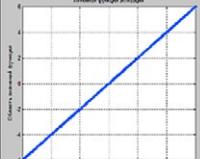
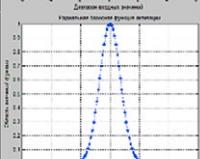
Хотя внешне модели, показанные на *рис. 1.5* и *рис. 1.3*, внешне не схожи, математически они эквивалентны.

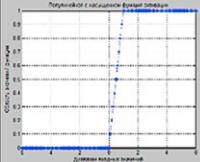
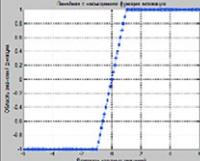
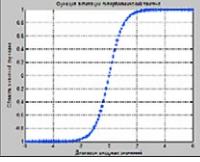
Типы функций активации

Функции активации, представленные в формулах как $\varphi(\cdot)$, определяют выходной сигнал нейрона в зависимости от индуцировано локального поля v . В таблице приведены примеры активационных функций и их в *Таблице 1.1*.

Таблица 1.1.

Виды активационных функций

№	Название	График функций	Формула	Область значений
1	Пороговая		$\varphi(v) = \begin{cases} 0, v < 0 \\ 1, v \geq 0 \end{cases}$	0, 1
2	Знаковая		$\varphi(v) = \begin{cases} 1, v > 0 \\ -1, v \leq 0 \end{cases}$	-1, 1
3	Сигмоидальная		$\varphi(v) = \frac{1}{1 + e^{-v}}$	(0, 1)
4	Полулинейная		$\varphi(v) = \begin{cases} v, v > 0 \\ 0, v \leq 0 \end{cases}$	(0, ∞)
5	Линейная		$\varphi(v) = v$	($-\infty$, ∞)
6	Радиальная базисная		$\varphi(v) = e^{-v^2}$	(0, 1)

№	Название	График функций	Формула	Область значений
7	Полулинейная с насыщением		$\varphi(v) = \begin{cases} 0, & v \leq 0 \\ v, & 0 < v < 1 \\ 1, & v \geq 1 \end{cases}$	(0, 1)
8	Линейная с насыщением		$\varphi(v) = \begin{cases} -1, & v \leq -1 \\ v, & -1 < v < 1 \\ 1, & v \geq 1 \end{cases}$	(-1, 1)
9	Гиперболический тангенс		$\varphi(v) = \frac{e^v - e^{-v}}{e^v + e^{-v}}$	(-1, 1)

1.4. Классификация нейронных сетей и их свойства

Нейронные сети (*Neural Networks*) — это модели биологических нейронных сетей мозга, в которых нейроны имитируются относительно простыми, часто однотипными, элементами (искусственными нейронами), топология соединений которых зависит от типа сети. В зависимости от функций, выполняемых нейронами в сети, можно выделить три их типа:

- **входные нейроны** — это нейроны, на которые подается входной вектор, кодирующий входное воздействие или образ внешней среды; в них обычно не осуществляется вычислительных процедур, информация передается с входа на выход нейрона путем изменения его активации;
- **выходные нейроны** — это нейроны, выходные значения которых представляют выход сети;
- **промежуточные нейроны** — эти нейроны составляют основу искусственных нейронных сетей.

Классифицируя нейронные сети по топологии, можно выделить три основных типа таких сетей *рис. 1.6.*:

1. Полносвязные сети;
2. Многослойные или слоистые сети;
3. Слабосвязные сети.

Полносвязные сети представляют собой Искусственные нейронные сети (ИНС), каждый нейрон которой передает свой выходной сигнал остальным нейронам, в том числе и самому себе. Все входные сигналы подаются всем нейронам. Выходными сигналами сети могут быть все или некоторые выходные сигналы нейронов после нескольких тактов функционирования сети.

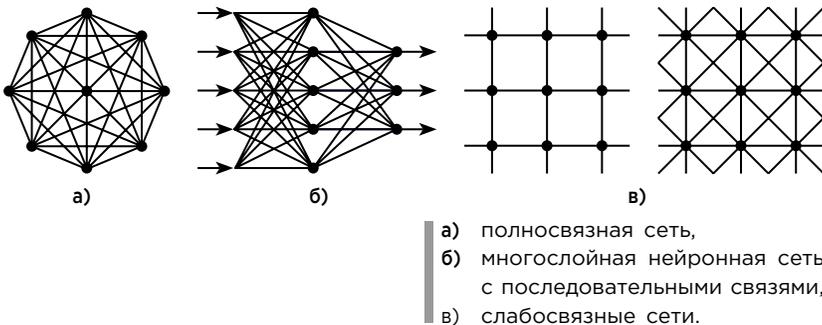


Рис. 1.6. Архитектура нейронных сетей:

В многослойных сетях нейроны объединяются в слои. Слой содержит совокупность нейронов с едиными входными сигналами. Число нейронов в каждом слое может быть любым и никак заранее не связано с количеством нейронов в других слоях. Внешние входные сигналы подаются на входы нейронов первого слоя, а выходами сети являются выходные сигналы последнего слоя. Кроме входного и выходного слоев в многослойной нейронной сети есть один или несколько промежуточных скрытых слоев.

В свою очередь, среди многослойных сетей выделяют следующие типы:

- **сети без обратных связей** (классический вариант — многослойная сеть прямого распространения) *рис.1.7.*;

где x_1, \dots, x_n — входные данные;
 y_1, \dots, y_m — выход нейронной сети.

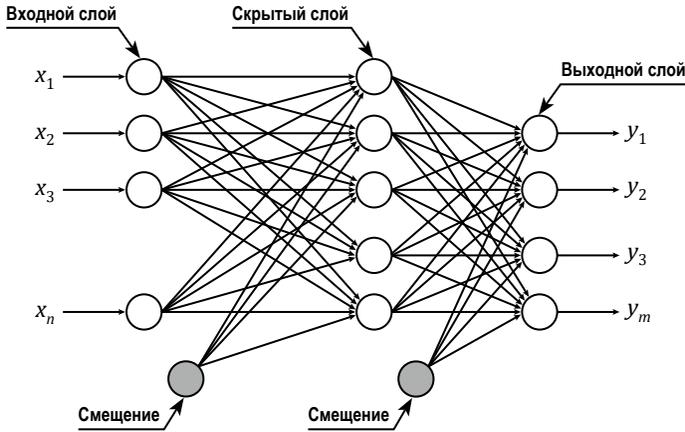


Рисунок 1.7. Многослойный перцептрон прямого распространения

- **сети с обратными связями** — это сети, у которых информация с последующих слоев передается на предыдущие;

Представление знаний в нейронных сетях

Под знаниями понимается хранимая информация или модели, используемые человеком или машиной для интерпретации, предсказания и реакции на внешние события.

К вопросам *представления знаний (knowledge representation)* относятся следующие: какую информацию необходимо хранить и как эту информацию представить физически для ее последующего использования. Таким образом, исходя из самой природы знаний, способ их представления определяется поставленной целью. Относительно реальных приложений «интеллектуальных»

систем можно утверждать, что успех решения зависит от хорошего представления знаний. Это касается и нейронных сетей, представляющих собой отдельный класс интеллектуальных систем. Форма представления входных сигналов может быть самой разной. Это приводит к тому, что разработка приемлемых нейросетевых решений становится творческим процессом.

Основной задачей нейронной сети является наилучшее обучение модели окружающего мира для решения поставленной задачи. Знания о мире включают два типа информации.

1. Известное состояние окружающего мира, представленное имеющимися в наличии достоверными фактами. Такая информация называется *априорной* (prior).
2. Наблюдения за окружающим миром (измерения), полученные с помощью сенсоров, адаптированных для конкретных условий, в которых должна функционировать данная нейронная сеть. Обычно такие измерения в значительной мере зашумлены, что потенциально может стать источником ошибок. В любом случае измерения, полученные таким способом, формируют множество информации, *примеры* из которого используются для обучения нейронной сети.

Примеры могут быть *маркированными* (labeled) и *немаркированными* (unlabeled). В маркированных примерах *входному сигналу* (input signal) соответствует *желаемый отклик* (desired response). Немаркированные примеры состоят из нескольких различных реализаций одного входного сигнала. В любом случае набор примеров, будь то маркированных или нет, представляет собой знания об интересующей предметной области, на основании которых и проводится обучение нейронной сети.

Множество пар сигналов вход-выход, каждая из которых состоит из входного сигнала и соответствующего ему желаемого выхода, называют *обучающими данными* (training data) или *обучающей выборкой* (training sample).

ГЛАВА 2

Многослойные нейронные сети прямого распространения

Многослойные нейронные сети прямого распространения (*feedforward neural networks*) являются одним из основных типов нейронных сетей, используемых в машинном обучении и искусственном интеллекте.

Многослойные нейронные сети прямого распространения — это тип искусственных нейронных сетей, характеризующийся направленным потоком данных от входного слоя к выходному, без каких-либо обратных связей или циклов. Они состоят из трех основных типов слоев: входного, одного или нескольких скрытых слоев и выходного слоя. Определение можно разделить на несколько ключевых аспектов:

Многослойность	Сеть состоит из нескольких слоев, где каждый слой содержит нейроны или узлы. Многослойность позволяет сети изучать и моделировать сложные и абстрактные шаблоны в данных.
Прямое распространение (feedforward)	В таких сетях информация движется строго в одном направлении — вперед. От входного слоя, через один или несколько скрытых слоев, к выходному слою. Не существует циклов или петель в данных, в отличие от рекуррентных нейронных сетей.
Входной слой	Первый слой, который принимает исходные данные. Каждый нейрон в этом слое представляет одну характеристику входного набора данных.
Скрытые слои	Один или несколько слоев, которые находятся между входным и выходным слоями. Нейроны в этих слоях выполняют сложные преобразования входных данных, используя веса и функции активации.

Выходной слой:	Слой, который выдает конечный результат сети. В задачах классификации, например, каждый нейрон в выходном слое может представлять определенный класс.
веса и функции активации:	В процессе обучения сети веса между нейронами адаптируются для оптимизации производительности сети.

Функции активации используются для внесения нелинейности, что позволяет сети учиться более сложным шаблонам и зависимостям в данных.

Эти сети широко используются в машинном обучении для решения разнообразных задач, включая классификацию, регрессию и распознавание образов.

2.1. Обучение нейронных сетей. Парадигмы обучения

Самым важным свойством нейронных сетей является их способность обучаться на основе данных окружающей среды. Процесс обучения — это процесс, в котором свободные параметры нейронной сети настраиваются посредством моделирования среды, в которую сеть встроена. Тип обучения определяется способом надстройки этих параметров. Это определение процесса обучения предполагает следующую последовательность событий:

1. В нейронную сеть поступают стимулы из внешней среды.
2. В результате этого изменяются свободные параметры нейронной сети.
3. После изменения внутренней структуры нейронная сеть отвечает на возбуждения уже иным образом.

Данный список четких правил решения проблемы обучения называется алгоритмом обучения. В теории нейронных сетей выделяют две фундаментальные парадигмы обучения нейронных сетей: обучение с учителем и без учителя. И пять основных моделей обучения:

- на основе коррекции ошибок;
- с использованием памяти;
- Хеббовское обучение;
- конкурентное обучение;
- обучение Больцмана.

Для реализации перечисленных моделей обучения могут применяться различные алгоритмы, приведем некоторые из них:

- алгоритмы локальной оптимизации с вычислением частных производных первого порядка;
- градиентный алгоритм;
- метод Ньютона;
- методы оптимизации с разреженными матрицами Гессе;
- метод Левенберга—Марквардта;
- поиск в случайном направлении;
- алгоритм обратного распространения ошибки.

Рассмотрим парадигму обучения нейронной сети с учителем

Обобщенная схема обучения представлена на *рисунке 2.1*. Процесс обучения нейронной сети заключается в следующем. Формируется база данных примеров на основе сведений окружающей среды, из базы данных примеров выбирается один пример, который состоит из входного вектора данных и эталонного вектора. На вход нейронной сети подается входной вектор. В результате вычислений нейронной сети получаем отклик нейронной сети на данный пример, сравниваем выходное значение нейронной сети с эталоном, если ошибка мала, нейронная сеть считается обученной, иначе необходимо настраивать весовые коэффициенты сети (синапсы) согласно алгоритмам обучения. После окончания обучения учителя можно отключить и позволить нейронной сети работать со средой самостоятельно.

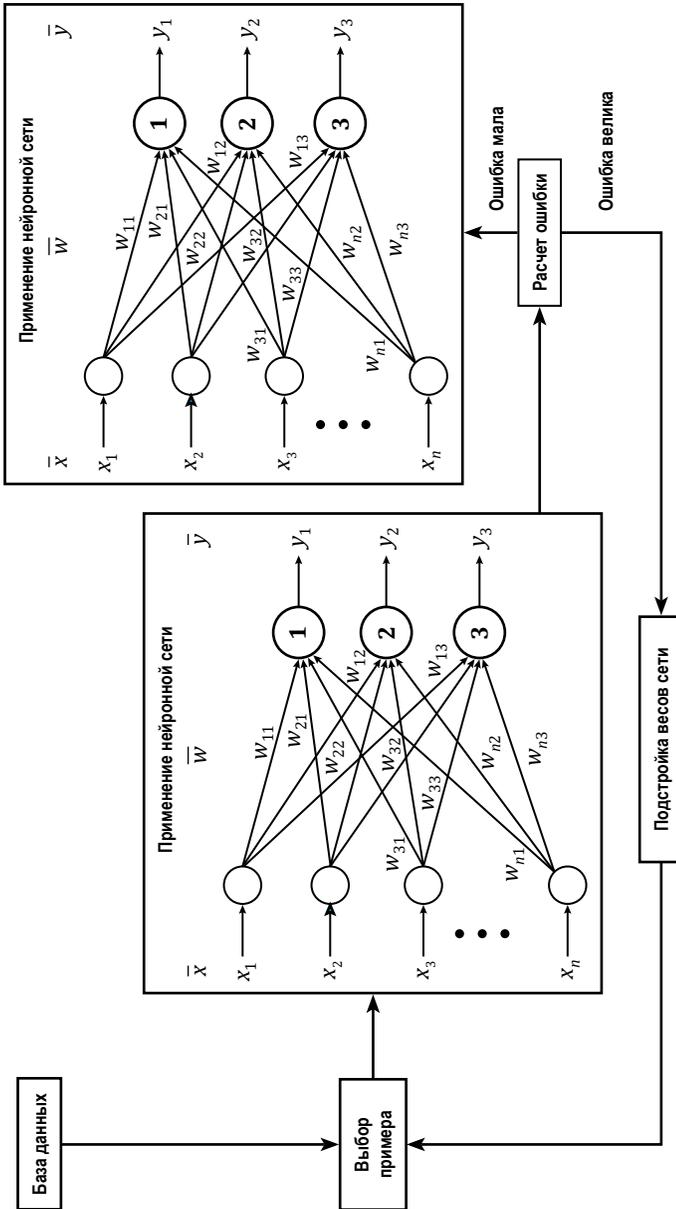


Рис. 2.1. Процесс обучения нейронной сети

2.2. Обучение, основанное на коррекции ошибок

Для того чтобы проиллюстрировать правило обучения, рассмотрим простейший случай нейрона k единственного вычислительного узла выходного слоя нейронной сети прямого распространения рисунке 2.2. а). Нейрон k работает под управлением *вектора сигнала* $x(n)$, производимого одним или несколькими скрытыми слоями нейронов, которые, в свою очередь, получают информацию из входного вектора (возбуждения), передаваемого начальным узлам (входному слою) нейронной сети. Под n подразумевается дискретное время, или, более конкретно, — номер шага итеративного процесса настройки синаптических весов нейрона k . *Выходной сигнал* нейрона k обозначается $y_k(n)$. Этот сигнал является единственным выходом нейронной сети. Он будет сравниваться с *желаемым выходом*, обозначенным $d_k(n)$. В результате получим *сигнал ошибки* $e_k(n)$. По определению

$$e_k(n) = d_k(n) - y_k(n). \quad (2.1)$$

Сигнал ошибки инициализирует *механизм управления* (control mechanism), цель которого заключается в применении последовательности корректировок к синаптическим весам нейрона k . Эти изменения нацелены на пошаговое приближение выходного сигнала $y_k(n)$ к желаемому $d_k(n)$. Эта цель достигается за счет минимизации *функции стоимости* или *индекса производительности* $E(n)$, определяемой в терминах сигнала ошибки следующим образом:

$$E(n) = \frac{1}{2} e_k^2(n). \quad (2.2)$$

Где $E(n)$ — *текущее значение энергии ошибки*. Пошаговая корректировка синаптических весов нейрона k продолжается до тех пор, пока система не достигнет *устойчивого состояния* (т. е. такого, при котором синаптические веса практически стабилизируются). В этой точке процесс обучения останавливается.

Процесс, описанный выше, называется *обучением, основанном на коррекции ошибок (error-correction learning)*. Минимизация функции стоимости $E(n)$ выполняется по так называемому дельта-правилу, или правилу Видроу-Хоффа, названному так в честь его создателей. Обозначим $w_{kj}(n)$ текущее значение синаптического веса w_{kj} нейрона k , соответствующего элементу $x_k(n)$ вектора $x(n)$ на шаге дискретизации n . В соответствии с дельта-правилом изменение $\Delta w_{kj}(n)$, применяемое к синаптическому весу w_{kj} на этом шаге дискретизации, задается выражением

$$\Delta w_{kj}(n) = \eta e_k(n) x_j(n) \quad (2.3)$$

где η — некоторая положительная константа, определяющая *скорость обучения* и используемая при переходе от одного шага процесса к другому. Из формулы (2.3) видно, что эту константу, естественно, именовать *параметром скорости обучения*. Вербально дельта-правило можно определить следующим образом.

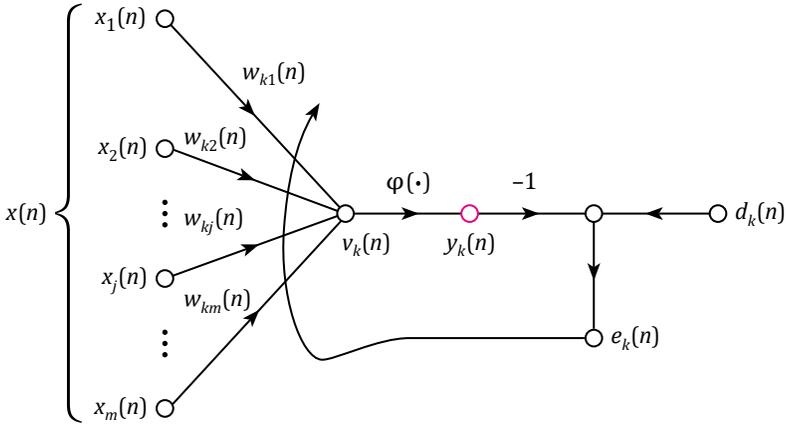
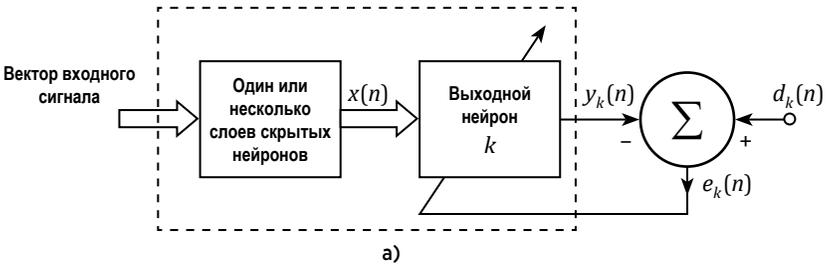
Многослойная сеть прямого распространения

Корректировка, применяемая к синаптическому весу нейрона, пропорциональна произведению сигнала ошибки на входной сигнал, его вызвавший.

Определенное таким образом дельта-правило предполагает возможность *прямого измерения* сигнала ошибки. Для обеспечения такого измерения требуется поступление желаемого отклика от некоторого внешнего источника, непосредственно доступного для нейрона k . Другими словами, нейрон k должен быть *видимым* для внешнего мира (рис. 2.2., а). На этом рисунке видно, что обучение на основе коррекции ошибки по своей природе является *локальным*. Это прямо указывает на то, что корректировка синаптических весов по дельта-правилу может быть локализована в отдельном нейроне k . Вычислив величину изменения синаптического веса $\Delta w_{kj}(n)$ можно определить его новое значение для следующего шага дискретизации:

$$\Delta w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n) \quad (2.4)$$

Таким образом, $\Delta w_{kj}(n+1)$ и $w_{kj}(n)$ можно рассматривать как старое и *новое* значения синаптического веса $w_{kj}(n)$.



- а) Блочная диаграмма нейронной сети; показаны только нейроны выходного слоя,
- б) Граф передачи сигнала выходного нейрона.

Рис. 2.2 Обучение, основанное на коррекции ошибок

2.3. Алгоритм обратного распространения ошибки (back propagation)

Рассмотрим идею одного из самых распространенных алгоритмов обучения алгоритма обратного распространения ошибки (back propagation). Предложенный в середине 1980-х годов *алгоритм обратного распространения ошибки* стал одним из ведущих факторов, породивших современный интерес к нейронным сетям, т. к. являлся эффективным способом обучения нейронной сети достаточно произвольной топологии.

Данный алгоритм применяется для обучения такого класса нейронных сетей, как многослойные сети прямого распространения. Обычно сеть состоит из множества сенсорных элементов (входных узлов и узлов источника), которые образуют входной слой, одного или нескольких скрытых слоев вычислительных нейронов и одного выходного слоя нейронов *рис. 2.3*.

где x_1, \dots, x_n — входные данные;

(y_1, \dots, y_m) — выход нейронной сети.

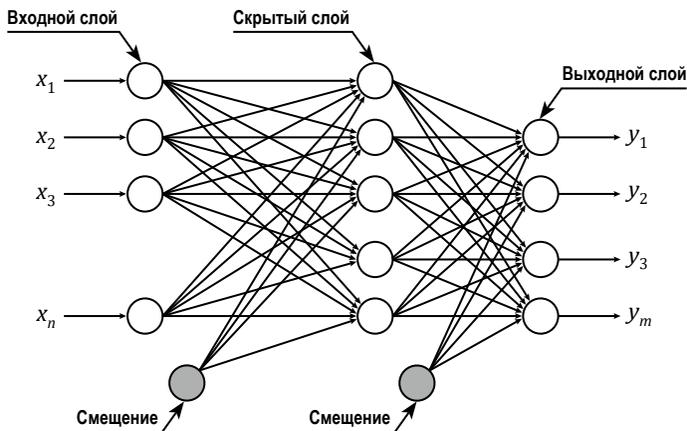


Рис. 2.3. Многослойная нейронная сеть прямого распространения

Входной сигнал распространяется по сети в прямом направлении, от слоя к слою. Многослойные НС успешно применяются

для решения разнообразных сложных задач. При этом обучение с учителем выполняется с помощью алгоритма обратного распространения ошибки. Этот алгоритм основывается на коррекции ошибки. Обучение методом обратного распространения ошибки предполагает два прохода по слоям: прямого и обратного *рис. 2.4*.

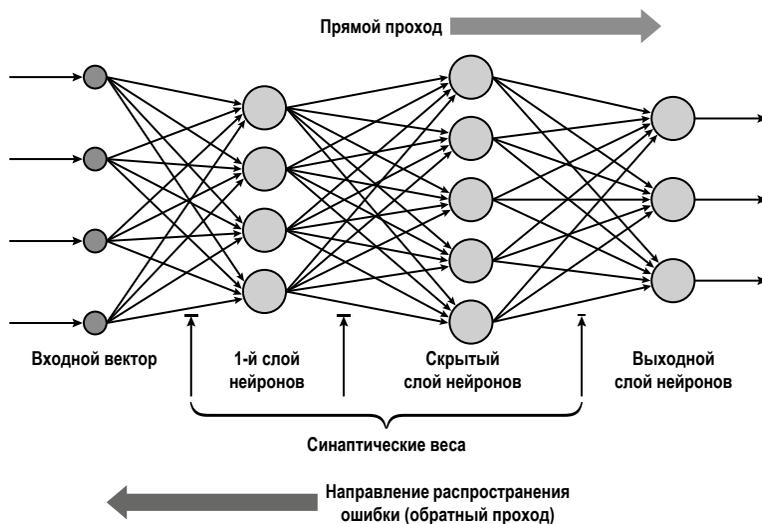


Рис. 2.4. Два прохода обучения в алгоритме обратного распространения

При прямом проходе входной вектор подается на сенсорные узлы сети, после чего распространяется по сети от слоя к слою. В результате генерируется набор выходных сигналов, который является фактической реакцией сети на данный входной образ.

Во время прямого прохода все синаптические веса сети фиксированы. Во время обратного прохода все синаптические веса сети настраиваются в соответствии с правилом коррекции ошибок (*рис. 2.5*), тем самым формируется сигнал ошибки.

Этот сигнал впоследствии распространяется по сети в направлении, обратном направлению синаптических связей. Синаптические веса настраиваются с целью максимального

приближения выходного сигнала к желаемому. Классический метод обратного распространения относится к алгоритмам с линейной сходимостью.

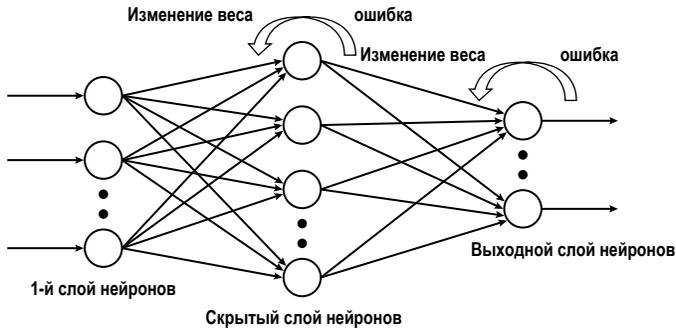


Рис. 2.5. Обратный проход в алгоритме обратного распространения ошибки

Рассмотрим алгоритм обратного распространения ошибки на *рис. 2.6.* изображен нейрон j , на который поступает поток сигналов.

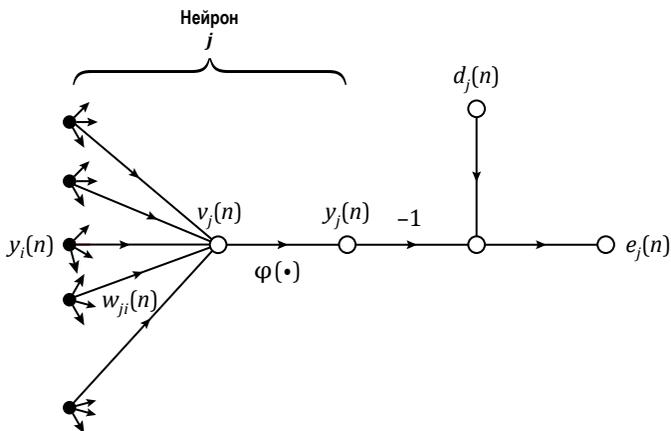


Рис. 2.6. Граф передачи сигнала в пределах некоторого нейрона j .

Коррекция $\Delta w_{ji}(n)$, применяемая к синаптическому весу, соединяющему нейроны i и j , определяется следующим дельта-правилом:

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n), \quad (2.5)$$

где $\Delta w_{ji}(n)$ — коррекция веса,
 η — параметр скорости обучения,
 $\eta \delta_j(n)$ — локальный градиент,
 $y_i(n)$ — входной сигнал нейрона i .

Значение локального градиента $\delta_j(n)$ зависит от положения нейрона в сети:

- если нейрон j — выходной, то градиент $\delta_j(n)$ равен произведению производной $f'_j(v_j(n))$ на сигнал ошибки $e_j(n)$ для нейрона j ;
- если нейрон j — скрытый, то градиент $\delta_j(n)$ равен произведению производной $f'_j(v_j(n))$ на взвешенную сумму градиентов, вычисленных для нейронов следующего скрытого или выходного, которые непосредственно связаны с данным нейроном j .

Вычисление локального градиента δ для каждого нейрона многослойного персептрона требует знания производной функции активации $f(\bullet)$, связанной с этим нейроном. Примером непрерывно дифференцируемой нелинейной функции активации, которая часто используется в многослойных персептронах, является сигмоидальная логистическая функция:

$$f_j(v_j(n)) = \frac{1}{1 + \exp(-\alpha v_j(n))}, \quad (2.6)$$

где $v_j(n)$ — индуцированное локальное поле нейрона j . Дифференцируя (2.7) по $v_j(n)$ получим:

$$f'_j(v_j(n)) = \frac{\alpha \exp(-\alpha v_j(n))}{[1 + \exp(-\alpha v_j(n))]^2}, 1 \quad (2.7)$$

так как $y_j(n) = f_j(v_j(n))$, то выражение (2.8) можно преобразовать к следующему виду:

$$f'_j(v_j(n)) = \alpha y_j(n)[1 - y_j(n)], \quad (2.8)$$

для нейрона j , расположенного в выходном слое, $y_j(n) = o_j(n)$. Отсюда локальный градиент нейрона j можно выразить следующим образом:

$$\delta_j(n) = e_j(n) f'_j(v_j(n)) = \alpha [d_j(n) - o_j(n)] o_j(n) [1 - o_j(n)], \quad (2.9)$$

где $o_j(n)$ — функциональный сигнал на выходе нейрона j ,
 $d_j(n)$ — его желаемый сигнал.

На рис. 2.7 показана передача сигнала, детально отражающий связь выходного нейрона k со скрытым нейроном j .

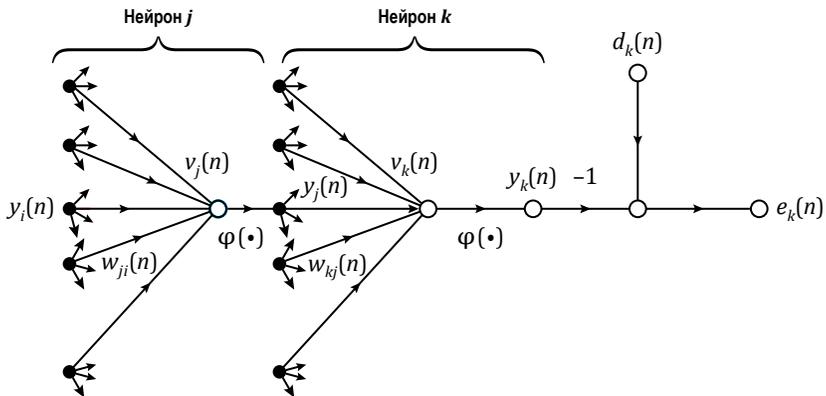


Рис. 2.7. Граф передачи сигнала от одного нейрона к другому нейрону

Для скрытого нейрона j локальный градиент можно выразить:

$$\delta_j(n) = f'_j(v_j(n)) \sum_{k=1}^M \delta_k(n) w_{kj}(n) = \alpha y_j(n) [1 - y_j(n)] \sum_{k=1}^M \delta_k(n) w_{kj}(n), \quad (2.10)$$

где M — количество нейронов в выходном слое.

Алгоритм обучения сети обратного распространения

Рассмотрим теперь полный алгоритм обучения нейросети.

Шаг 1. *Инициализация сети.* Весовым коэффициентам присваиваются малые случайные значения, например из диапазона $(-0.3, 0.3)$; задаются ε — параметр точности обучения, α — параметр скорости обучения (как правило, ≈ 0.1 и может еще уменьшаться в процессе обучения), N_{\max} — максимально допустимое число итераций.

Шаг 2. *Вычисление текущего выходного сигнала.* На вход сети подается один из образов обучающей выборки и определяются значения выходов всех нейронов нейросети.

Шаг 3. *Настройка синаптических весов.* Рассчитать изменение весов для скрытого слоя и выходного слоя по формулам

$$\delta_j(n) = f'_j(v_j(n)) \sum_{k=1}^M \delta_k(n) w_{kj}(n) = \alpha y_j(n) [1 - y_j(n)] \sum_{k=1}^M \delta_k(n) w_{kj}(n), \quad (2.11)$$

$$\delta_j(n) = e_j(n) f'_j(v_j(n)) = \alpha [d_j(n) - o_j(n)] o_j(n) [1 - o_j(n)], \quad (2.12)$$

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n), \quad (2.13)$$

$$w_{ji}(n+1) = w_{ji}(n) + \eta \delta_j(n) y_i(n). \quad (2.14)$$

Шаг 4. Шаги 2–3 повторяются для всех обучающих векторов. Обучение завершается по достижении для каждого из обучающих образов значения функции ошибки, не превосходящего ε или после максимально допустимого числа итераций.

Замечание 1. На шаге 2 векторы из обучающей последовательности лучше предъявлять на вход в случайном порядке.

Замечание 2. Во многих случаях желательно наделять каждый нейрон обучаемым смещением. Это позволяет сдвигать начало отсчета логистической функции, давая эффект, аналогичный подстройке порога персептронного нейрона, и приводит к ускорению процесса обучения. Эта возможность может быть легко введена в обучающий алгоритм с помощью добавляемого к каждому нейрону веса, присоединенного $k+1$. Этот вес обучается так же, как и все остальные веса, за исключением того, что подаваемый на него сигнал всегда равен $+1$, а не выходу нейрона предыдущего слоя.

Замечание 3. Количество входов и выходов сети, как правило, диктуется условиями задачи, а размер скрытого слоя находят экспериментально. Обычно число нейронов в нем составляет 30–50% от числа входов. Слишком большое количество нейронов скрытого слоя приводит к тому, что сеть теряет способность к обобщению (она просто досконально запоминает элементы обучающей выборки и не реагирует на схожие образцы, что неприемлемо для задач распознавания). Если число нейронов в скрытом слое слишком мало, сеть оказывается просто не в состоянии обучиться.

Замечание 4. Выходы каждого нейрона сети лежат в диапазоне $(0,1)$ — области значений логистической функции — это надо учитывать при формировании обучающей выборки. Если необходимо получить от сети бинарный выход, то, как правило, вместо 0 используют 0,1, а вместо 1—0,9, так как границы интервала недостижимы.

Модификации алгоритма обратного распространения ошибки, часто используемого в обучении нейронных сетей, преследуют несколько важных целей:

- 1. Ускорение сходимости:** некоторые модификации направлены на ускорение обучения нейронных сетей. Это может достигаться за счёт более эффективного обновления весов или более точного выбора направления для градиентного спуска.
- 2. Избегание локальных минимумов:** традиционный алгоритм обратного распространения может застревать в локальных минимумах функции потерь. Модификации могут помочь найти более оптимальные точки минимума.
- 3. Предотвращение переобучения:** переобучение — это когда модель слишком хорошо адаптируется к тренировочным данным и теряет способность к обобщению. Модификации, такие как ранняя остановка или регуляризация, могут помочь контролировать этот процесс.
- 4. Обработка проблемы исчезающего или взрывающегося градиента:** в глубоких сетях градиенты, необходимые для обучения, могут становиться очень маленькими или, наоборот, чрезмерно большими, что затрудняет эффективное обучение. Модификации могут помочь стабилизировать градиенты.
- 5. Адаптация к специфическим типам данных:** некоторые модификации разработаны для работы с особыми типами данных, такими как временные ряды или изображения, и могут включать изменения в архитектуре сети или функции потерь.
- 6. Улучшение общей производительности:** это может включать уменьшение времени обучения, повышение точности модели или улучшение её способности к обобщению на новых данных.

Примеры популярных модификаций включают алгоритмы, такие как стохастический градиентный спуск (*SGD*), метод момента, *Adagrad*, *RMSprop*, *Adam* и другие. Эти методы различаются по способам корректировки скорости обучения и обновления весов в процессе обучения.

2.4. Последовательный и пакетный режимы обучения алгоритма обратного распространения ошибки

В практических приложениях алгоритма обратного распространения в процессе обучения многослойного персептрона ему многократно предъявляется предопределенное множество обучающих примеров. Как уже отмечалось, один полный цикл предъявления полного набора примеров обучения называют *эпохой*. Процесс обучения проводится от эпохи к эпохе, пока синаптические веса и уровни порога не стабилизируются, а среднеквадратическая ошибка на всем обучающем множестве не сойдется к некоторому минимальному значению. Целесообразно *случайным образом изменять порядок представления примеров обучения* для разных эпох. Такой принцип предъявления образов делает поиск в пространстве весов стохастическим, предотвращая потенциальную возможность появления замкнутых циклов в процессе эволюции синаптических весов.

Для данного обучающего множества алгоритм обратного распространения можно реализовать двумя способами.

Последовательный режим

Последовательный режим обучения по методу обратного распространения также иногда называют стохастическим или интерактивным. В этом режиме корректировка весов проводится после подачи каждого примера. Для примера рассмотрим эпоху, состоящую из N обучающих примеров, упорядоченных следующим образом: $(x(1), d(1)), \dots, (x_i(v), d_i(v))$. Сети предъявляется первый пример $(x(1), d(1))$ этой эпохи, после чего выполняются описанные выше прямые и обратные вычисления. В результате проводится корректировка синаптических весов и уровней порогов в сети. После этого сети предъявляется вторая пара $(x(2), d(2))$ в эпохе, повторяются прямой и обратный проходы, приводящие к следующей коррекции синаптических весов и уровня порога. Этот процесс повторяется, пока сеть не завершит обработку последнего примера (пары) данной эпохи — $(x_i(v), d_i(v))$.

Пакетный режим

В *пакетном режиме (batch mode)* обучения по методу обратного распространения корректировка весов проводится после подачи в сеть примеров обучения (эпохи). Для конкретной эпохи функция стоимости определяется как среднеквадратическая ошибка (2.15) и (2.16), представленная в составной форме:

$$E_{av}(n) = \frac{1}{2N} \sum_{n=1}^N \sum_{j=1}^N e_j^2(n), \quad (2.15)$$

где сигнал ошибки $e_j(n)$ соответствует нейрону j для примера обучения n и определяется формулой (2.15). Ошибка $e_j(n)$ равна разности между $d_j(n)$ и $y_j(n)$ для j -го элемента вектора желаемых откликов $d_j(n)$ и соответствующего выходного нейрона сети. В выражении (2.15) внутреннее суммирование по j выполняется по всем нейронам выходного слоя сети, в то время как внешнее суммирование по n выполняется по всем образам данной эпохи. При заданном параметре скорости обучения η корректировка, применяемая к синаптическому весу w_{ij} , связывающему нейроны i и j , определяется следующим дельта-правилом:

$$\Delta w_{ij}(n) = -\eta \frac{dE_{av}}{dw_{ij}} = -\frac{\eta}{2N} \sum_{n=1}^N e_j(n) \frac{de_j(n)}{dw_{ij}}, \quad (2.16)$$

Для вычисления частной производной $\frac{de_j(n)}{dw_{ij}}$ нужно проделывать тот же путь, что и ранее. Согласно (2.16), в пакетном режиме корректировка веса Δw_{ij} выполняется только после прохождения по сети всего множества примеров.

Последовательный (онлайн) и пакетный режимы обучения нейронных сетей — это два различных подхода к обновлению весов в процессе обучения. Они имеют свои особенности и применения:

Последовательный (онлайн) режим обучения

1. **Определение:** в последовательном режиме обучения обновление весов происходит после каждого обучающего примера.
2. **Преимущества:**
 - быстрое обновление весов может привести к более быстрому обучению на больших наборах данных;
 - хорошо подходит для обучения в режиме реального времени или, когда данные поступают последовательно (например, в онлайн-системах);
 - может лучше адаптироваться к изменяющимся данным (например, в задачах, где данные меняются со временем).
3. **Недостатки:**
 - может быть менее стабильным, так как каждый обучающий пример может значительно повлиять на веса;
 - чаще подвержен застреванию в локальных минимумах.
 - может потребовать больше итераций для сходимости.

Пакетный режим обучения

1. **Определение:** в пакетном режиме обучения обновление весов происходит после обработки всего набора обучающих данных (пакета).
2. **Преимущества:**
 - обеспечивает более стабильное и надёжное обновление весов, так как учитывается весь набор данных;
 - часто сходится быстрее, чем последовательное обучение (в терминах эпох);
 - меньше подвержен влиянию отдельных выбросов в данных.
3. **Недостатки:**
 - требует больше памяти, так как необходимо хранить весь набор данных в памяти для обновления весов;
 - может быть медленнее на больших наборах данных из-за необходимости обработки всего набора перед обновлением весов;

- не подходит для онлайн-обучения или когда данные постоянно меняются.

Мини-пакетное обучение (Mini-Batch Training)

Существует также промежуточный подход, называемый мини-пакетным обучением, который сочетает преимущества обоих методов. В этом подходе обновление весов происходит после обработки небольшого подмножества обучающих данных (мини-пакета). Это позволяет более эффективно использовать память и ускорять обучение, сохраняя при этом достаточную стабильность процесса обучения.

ГЛАВА 3

Самоорганизующиеся нейронные сети

Самоорганизующиеся нейронные сети — это особый класс нейросетей, которые способны автоматически организовать свою структуру и настроить свои параметры в процессе обучения без явного внешнего вмешательства. Эти сети основываются на принципах самоорганизации, которые вдохновлены биологическими процессами в мозге. Ниже представлены ключевые аспекты и примеры самоорганизующихся нейронных сетей:

Рассмотрим парадигму обучения нейронной сети без учителя

Обучение без учителя (или *обучение на основе самоорганизации*) осуществляется без вмешательства внешнего учителя или корректора, контролирующего процесс обучения. Существует лишь *независимая от задачи мера качества (task-independent measure)* представления, которому должна научиться нейронная сеть, и свободные параметры сети оптимизируются по отношению к этой мере. После обучения сети на статистические закономерности входного сигнала она способна формировать внутреннее представление кодируемых признаков входных данных и, таким образом, автоматически создавать новые классы.

Для обучения без учителя можно воспользоваться правилом конкурентного обучения. Например, можно использовать нейронную сеть, состоящую из двух слоев — входного и выходного. Входной слой получает доступные данные. Выходной слой состоит из нейронов, конкурирующих друг с другом за право отклика на признаки, содержащиеся во входных данных.

В простейшем случае нейронная сеть действует по принципу «победитель получает все».

Эпоха обучения — это один просмотр всех примеров обучающей выборки с одновременной коррекцией весов сети (на этих примерах, в зависимости от правильности их решения сетью). Чтобы сеть обучилась, может потребоваться несколько (и даже несколько десятков, и даже несколько сотен,) эпох. В течение эпохи не на каждом обучающем примере может происходить коррекция весов сети, т. к. если сеть уже обучилась правильно решать данный конкретный пример, то можно перестать требовать постоянных «улучшений» качества решения этого примера (поскольку эти улучшения уже будут несущественными) и позволить сети направить её внимание на другие примеры, которые пока ещё решаются с недостаточной точностью.

3.1. Принципы работы

1. Адаптация без учителя: в отличие от традиционных нейросетей, которые обучаются на основе предоставленных обучающих данных с явными метками, самоорганизующиеся сети адаптируются к входным данным автономно.

Адаптация без учителя в контексте самоорганизующихся нейронных сетей означает, что сеть способна адаптироваться и обучаться из входных данных без явного вмешательства или руководства со стороны человека. Это ключевой принцип работы таких сетей, и вот его основные аспекты:

ОСНОВНЫЕ АСПЕКТЫ АДАПТАЦИИ БЕЗ УЧИТЕЛЯ

Автоматическое обнаружение структур и шаблонов	Сеть автоматически выявляет взаимосвязи, структуры и шаблоны в данных, анализируя их внутреннюю структуру без каких-либо предварительно заданных инструкций или меток
-------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------

Кластеризация данных	Один из основных механизмов адаптации без учителя — кластеризация, где данные группируются
-----------------------------	--------------------------------------------------------------------------------------------

	на основе сходства их характеристик. Это позволяет идентифицировать скрытые паттерны и связи в данных
Обучение на основе плотности входных данных	В отличие от обучения с учителем, где сеть обучается на основе явно заданных примеров и меток, обучение без учителя опирается на плотность и распределение входных данных
Динамическая адаптация к изменениям	Сети, обучающиеся без учителя, способны динамически адаптироваться к новым данным, изменяя свою структуру или
Минимизация внутренних критериев	Вместо оптимизации внешних показателей, таких как точность классификации, самоорганизующиеся сети стремятся минимизировать внутренние критерии, такие как ошибка восстановления или внутрикластерное расстояние

Адаптация без учителя делает самоорганизующиеся нейронные сети мощным инструментом для анализа неструктурированных и сложных данных, где традиционные подходы могут быть неэффективны. Это особенно важно в областях, где явные метки отсутствуют или где нужно исследовать скрытые структуры данных.

- 2. Самоорганизация:** сети автоматически формируют структуры или паттерны в данных, идентифицируя сходства и различия без заранее заданных критериев.

Самоорганизация в нейронных сетях относится к их способности автоматически обнаруживать и формировать структуры или паттерны в данных. Этот процесс происходит без заранее определенных критериев или меток, исключительно на основе входных данных. Рассмотрим подробнее, как это работает и какие особенности он имеет.

КЛЮЧЕВЫЕ ОСОБЕННОСТИ САМООРГАНИЗАЦИИ

Обнаружение скрытых структур	Самоорганизующиеся сети анализируют входные данные и находят в них скрытые структуры, такие как группы или кластеры похожих элементов.
Формирование внутренних представлений	Сеть развивает внутренние представления данных, которые отражают основные характеристики и закономерности в этих данных.
Обучение на основе плотности входных данных	В отличие от обучения с учителем, где сеть обучается на основе явно заданных примеров и меток, обучение без учителя опирается на плотность и распределение входных данных.
Адаптация к новым данным	Самоорганизующиеся сети способны адаптироваться к новым или изменяющимся данным, постоянно обновляя свои структуры в ответ на новую информацию.
Работа без меток	В отличие от контролируемых методов обучения, самоорганизующиеся сети не требуют меток или классификаций для обучения, они работают с необработанными, неструктурированными данными.

3. Карта признаков: они часто используют механизм карты признаков для визуализации и интерпретации структуры данных.

Карта признаков в контексте самоорганизующихся нейронных сетей, таких как Карты Кохонена (*Self-Organizing Maps, SOMs*), представляет собой способ визуализации и интерпретации внутренней структуры данных. Эти карты создаются путем проекции многомерных данных на обычно двумерное пространство, сохраняя при этом ключевые топологические и метрические свойства исходного пространства данных. Рассмотрим основные аспекты карты признаков:

ОСНОВНЫЕ АСПЕКТЫ КАРТЫ ПРИЗНАКОВ

Визуализация многомерных данных	Карты признаков позволяют визуализировать сложные многомерные данные в удобном для понимания
----------------------------------------	----------------------------------------------------------------------------------------------

	двумерном формате. Это делает их мощным инструментом для анализа данных
Сохранение топологических свойств	Важной особенностью карты признаков является сохранение топологических свойств исходных данных. Это означает, что похожие входные векторы будут проецироваться на близкие участки карты
Кластеризация и группировка	Карта признаков может использоваться для кластеризации данных, где схожие элементы группируются вместе, образуя легко идентифицируемые кластеры на карте
Интерпретация и анализ	Карты признаков обеспечивают интуитивно понятный способ для анализа и интерпретации сложных наборов данных, выявления закономерностей, аномалий и взаимосвязей между различными признаками

ПРИМЕРЫ ПРИМЕНЕНИЯ

Анализ данных	Используется для визуального анализа сложных данных в таких областях, как финансы, биоинформатика, маркетинг и социология
Распознавание образов	В области машинного обучения и распознавания образов для визуализации и понимания, как сеть классифицирует и группирует различные образцы
Робототехника и искусственный интеллект	Для картографирования и навигации в пространстве, где сеть может учиться распознавать и реагировать на различные среды и условия.

Важность карты признаков

Карта признаков является ценным инструментом в машинном обучении и искусственном интеллекте, поскольку она предоставляет наглядное и интерпретируемое представление сложных данных. Это помогает исследователям и аналитикам лучше понимать структуру и свойства данных, находить скрытые шаблоны и отношения, а также упрощать принятие решений на основе анализа данных.

3.2. Примеры и применение

1. **Карты Кохонена (*Self-Organizing Maps, SOMs*):** это один из самых известных примеров самоорганизующихся сетей. SOMs преобразуют сложные, многомерные входные данные в одномерные или двумерные карты, сохраняя при этом топологические свойства исходного пространства данных.
2. **Адаптивные резонансные теории (*ART*):** эти сети способны обучаться в реальном времени, адаптируясь к новым данным без забывания ранее обученной информации (проблема стабильности-пластичности).
3. **Нейронные сети Гроссберга:** они используются для моделирования внимания, обучения и памяти, имитируя процессы, происходящие в мозге.

ПРЕИМУЩЕСТВА И НЕДОСТАТКИ

Преимущества	<ul style="list-style-type: none">▪ Способность обрабатывать неструктурированные и сложные данные▪ Высокая адаптивность к изменениям в данных▪ Полезны для визуализации и кластеризации данных
Недостатки	<ul style="list-style-type: none">▪ Сложность интерпретации результата▪ Ограниченная способность к обобщению за пределы наблюдаемых данных▪ Зависимость от качества и разнообразия входных данных

Вывод

Самоорганизующиеся нейронные сети предлагают уникальный подход к обработке и анализу данных, особенно когда нет ясных меток или когда данные слишком сложны для традиционных методов. Они находят применение в таких областях, как биоинформатика, финансовый анализ, обработка изображений и многое другое. Однако их использование требует тщательного понимания их работы и ограничений.

ГЛАВА 4

Генеративные нейронные сети (GAN)

4.1. Принцип работы генеративных нейронных сетей

Генеративные нейронные сети (*Generative Neural Networks*) — это класс нейронных сетей, которые используются для создания новых данных, обычно из обучающего набора данных. Они работают по-разному в зависимости от своей архитектуры, но основной принцип работы остается примерно одинаковым: они учатся моделировать вероятностное распределение данных и затем могут генерировать новые примеры данных, которые максимально похожи на те, что были в обучающем наборе.

Принцип работы генеративных нейронных сетей включает в себя следующие основные шаги:

1. **Загрузка и предобработка данных:** сначала необходимо загрузить обучающий набор данных и выполнить предобработку, чтобы привести данные к форме, пригодной для обучения нейронной сети.
2. **Выбор архитектуры сети:** выбирается архитектура генеративной нейронной сети, которая может включать в себя различные типы слоев, такие как сверточные, рекуррентные, полно связанные и др., в зависимости от задачи.
3. **Обучение:** Сеть обучается на обучающем наборе данных, где она пытается моделировать статистические закономерности и распределения данных. В процессе обучения сеть корректирует свои веса и параметры, чтобы минимизировать ошибку между сгенерированными данными и данными из обучающего набора.

4. Генерация данных: после завершения обучения сеть может быть использована для генерации новых данных. Обычно это делается путем передачи случайного входного шума или некоторого начального состояния в сеть, которая затем генерирует новые примеры данных, пытаясь следовать тем статистическим закономерностям, которые она узнала во время обучения.
5. Оценка качества: сгенерированные данные могут быть оценены с помощью различных метрик, чтобы определить, насколько они похожи на реальные данные.

4.2. Приложения GAN: синтез изображений, структур и другие возможности

Примерами генеративных нейронных сетей являются Генеративные Состязательные Сети (GANs), Вариационные Автокодировщики (VAEs), и Рекуррентные Нейронные Сети (RNNs) с использованием LSTM или GRU слоев для генерации последовательных данных, таких как текст или музыка. Каждая из этих архитектур имеет свои особенности и применения, но все они работают на основе общего принципа моделирования данных и их последующей генерации.

Генеративные нейронные сети (GAN) и другие генеративные модели могут быть использованы для создания разнообразных контента, включая изображения, текст, звук и многое другое.

Вот некоторые примеры генеративных нейронных сетей и их приложений:

1. **DeepDream** — это генеративная модель, разработанная Google, которая преобразует существующие изображения в художественные произведения искусства, используя нейронные сети для анализа исходных данных и создания фантастических визуальных эффектов.
2. **StyleGAN** и его последующие версии, такие как **StyleGAN2** и **StyleGAN3**, позволяют создавать реалистичные портреты

лиц, которые выглядят, как настоящие люди, но при этом они являются вымышленными персонажами. Эта технология часто используется в генерации фотографий людей, которых на самом деле не существует.

3. **OpenAI GPT** (*Generative Pre-trained Transformer*): GPT-3 и его последующие версии — это генеративные модели для текста, которые способны генерировать человекоподобные тексты, отвечать на вопросы, создавать статьи и даже помогать в чат-ботах.
4. **CycleGAN** — это модель, которая способна преобразовывать изображения из одного домена в изображения другого домена без парных данных для обучения. Например, она может превращать фотографии лошадей в фотографии зебр.
5. **MuseNet** — это генеративная модель, способная создавать музыку в разных стилях и жанрах. Она может генерировать оригинальные музыкальные композиции, исходя из заданных параметров.
6. **WaveGAN** — это генеративная модель, используемая для генерации аудио, включая речь и музыку. Она может создавать звуковые эффекты, голоса и музыкальные фрагменты.
7. **DALL·E** — это генеративная модель, разработанная OpenAI, которая способна создавать изображения, основанные на текстовых описаниях. Например, вы можете описать «жирафа, сделанного из жележных мишек», и DALL·E сгенерирует соответствующее изображение.



Эти примеры генеративных нейронных сетей демонстрируют разнообразие применений и потенциал создания контента в различных областях, используя технологии генеративного искусства и генеративного моделирования.

4.3. Приложение GAN для синтеза изображений на примере DALL·E

DALL·E — это искусственный интеллект, разработанный компанией OpenAI, который создаёт изображения на основе текстовых описаний, что является формой генерации изображений по тексту. Название «DALL·E» представляет собой сочетание имени художника Сальвадора Дали и анимационного персонажа WALL·E.

ОПИСАНИЕ ПРИНЦИПА РАБОТЫ DALL·E:

Определение изображения:

Цель: определите, что именно вы хотите изобразить. Это может быть конкретный объект, сцена, абстрактная идея или комбинация элементов.

Детали: подумайте о деталях, таких как цвета, стиль (например, реалистичный, карикатурный, футуристический), настроение (спокойное, энергичное, мрачное), и окружение (внутреннее пространство, природа, городской пейзаж).

Особые желания: если у вас есть особые пожелания касаясь композиции, персонажей или элементов, включите их в ваше описание.

Описание запроса:

Точность: чем точнее и детальнее ваше описание, тем лучше результат. Включите все важные элементы, которые вы хотите видеть на изображении.

Стиль: укажите, если вы предпочитаете изображение в определенном художественном стиле или имитирующее творчество конкретного художника (с учетом того, что работы должны быть созданы до 1912 года).

Контекст: если изображение имеет специфический контекст или предназначение (например, иллюстрация к статье, подарок), это также стоит упомянуть.

Отправка запроса:

Формулировка: сформулируйте ваше описание в виде четкого текстового сообщения.

Отправка: отправьте мне ваш запрос. Я обработаю его и создам подробное описание для DALL·E.

Ревизия и модификация:	Просмотр: как только изображение будет создано, вы сможете его просмотреть. Обратная связь: если вам что-то не нравится или вы хотите внести изменения, сообщите мне. Вы можете попросить изменить стиль, цвета, добавить или убрать элементы и так далее.
Итоговое изображение:	Получение результата: после утверждения финальной версии, вы получите итоговое изображение. Использование: изображение можно использовать в соответствии с вашими потребностями, с учетом правил и ограничений использования, связанных с DALL·E.

Модель DALL·E тренируется на большом наборе данных, состоящем из изображений и сопутствующих текстовых описаний, что позволяет ей учиться на связях между словами и визуальными концепциями. Это обучение позволяет DALL·E творчески реагировать на запросы, создавая уникальные изображения, которые ранее не существовали.

Важно помнить, что DALL·E не может создавать изображения нарушающие авторские права, содержащие изображения реальных людей без их согласия, или нарушающие этические нормы.

ГЛАВА 5

Нейронные сети для обработки естественного языка (NLP)

5.1. Типы нейронных сетей для обработки естественного языка NLP

Для обработки естественного языка (*Natural Language Processing, NLP*) используются различные типы нейронных сетей. Каждая из них обладает своими особенностями, которые делают её подходящей для определенных задач NLP. Вот некоторые из наиболее часто используемых нейронных сетей в NLP:

Рекуррентные нейронные сети (RNN):

- способны обрабатывать последовательности данных (например, слова в предложении).
- имеют «память» о предыдущих входных данных, что позволяет им учитывать контекст в последовательности.
- примеры включают Long Short-Term Memory (LSTM) и Gated Recurrent Units (GRU), которые предназначены для решения проблемы исчезающего градиента, свойственной базовым RNN.

Свёрточные нейронные сети (CNN):

- хотя чаще используются в области компьютерного зрения, CNN также применяются в NLP для распознавания паттернов в данных.
- могут эффективно обнаруживать локальные и иерархические структуры в данных (например, фразы или словосочетания в тексте).

Трансформеры:

- основаны на механизмах внимания, позволяющих модели фокусироваться на разных частях входных данных для более качественной обработки.
 - параллельная обработка данных делает их более эффективными по сравнению с RNN и LSTM.
-

	<ul style="list-style-type: none"> ■ трансформеры лежат в основе многих передовых моделей NLP, таких как BERT, GPT, RoBERTa и других.
Модели на основе графов	включают Graph Neural Networks (GNN), которые используются для работы с данными, имеющими структуру графа, например, синтаксические деревья в NLP.
Автоэнкодеры	<ul style="list-style-type: none"> ■ используются для уменьшения размерности и предварительного обучения на размеченных данных. ■ в NLP могут помогать в извлечении семантических представлений из больших объемов текста.

Выбор конкретного типа нейронной сети зависит от задачи NLP, которую необходимо решить, а также от размера и характеристик используемого набора данных. Например, для генерации текста часто используются трансформеры, в то время как для классификации текста могут быть использованы как CNN, так и RNN.

5.2. Трансформеры

Нейронные сети «Трансформеры» — это современный класс архитектур искусственного интеллекта, который стал основой для большинства передовых моделей обработки естественного языка (NLP). Впервые они были предложены в статье “Attention is All You Need” в 2017 году исследователями из Google. Основная идея трансформеров заключается в использовании механизмов внимания (attention mechanisms), которые позволяют моделям динамически фокусироваться на различных частях входных данных для более эффективного их представления и обработки.

ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ТРАНСФОРМЕРОВ

Механизмы внимания	Трансформеры используют так называемое «самовнимание» (self-attention), чтобы вычислить веса входных данных. Это позволяет каждому элементу
---------------------------	---------------------------------------------------------------------------------------------------------------------------------------------

	последовательности взаимодействовать с каждым другим элементом, улучшая понимание контекста и зависимостей.
Отсутствие рекуррентности	В отличие от рекуррентных нейронных сетей (RNN), трансформеры не обрабатывают данные последовательно. Это позволяет им обучаться намного быстрее, поскольку обработка может быть распараллелена.
Позиционное кодирование	Так как в трансформерах нет последовательной обработки, для сохранения информации о порядке элементов в последовательности используется позиционное кодирование.
Слои энкодера и декодера	<ul style="list-style-type: none">■ В полной архитектуре трансформера (как в оригинальной модели) используются слои энкодера для обработки входных данных и слои декодера для генерации выходных данных. Энкодеры и декодеры содержат множество одинаковых слоев с механизмами внимания и полносвязными нейронными сетями.■ Сквозное соединение (residual connections):■ Сквозное соединение позволяет избежать проблемы исчезающего градиента в глубоких сетях, направляя выходы предыдущих слоёв непосредственно к последующим слоям.
Нормализация слоёв (layer normalization)	Нормализация применяется перед каждым подслоем в энкодерах и декодерах и после сквозного соединения, что помогает стабилизировать обучение.

Трансформеры стали основой для многих последующих моделей, включая BERT, GPT, RoBERTa, T5 и другие, каждая из которых адаптирована для выполнения различных задач NLP, таких как перевод, классификация текстов, вопросно-ответные системы и генерация текста. Эти модели тренируются на больших наборах текстовых данных и способны генерировать удивительно коэрентный и релевантный текст.

5.3. Применение NLP

NLP, или обработка естественного языка (*Natural Language Processing*), это область информатики и искусственного интеллекта, которая занимается взаимодействием между компьютерами и человеческим (естественным) языком. В частности, NLP позволяет компьютерам анализировать, понимать и генерировать язык, который естественен для людей.

NLP применяется во многих аспектах повседневной жизни, включая:

Распознавание речи	Преобразование устной речи в текст, используется в голосовых помощниках и для транскрибирования аудиозаписей.
Машинный перевод	Автоматический перевод текста с одного языка на другой.
Анализ настроений	Определение эмоциональной окраски текста для понимания мнений в отзывах и социальных сетях.
Чат-боты и виртуальные помощники	Ведение диалога с пользователями в естественной форме.
Извлечение информации	Автоматическое извлечение конкретной информации из текстов, такой как имена, даты и местоположения.
Резюмирование текста	Создание кратких изложений длинных документов.
Классификация текста	Распределение текстов по категориям, например, фильтрация спама в электронной почте.

Основные задачи NLP часто включают синтаксический анализ, семантический анализ, генерацию языка и понимание речи. Современные достижения в области NLP в значительной степени связаны с использованием машинного обучения и, в частности, с глубоким обучением, где нейронные сети, такие как трансформеры, играют ключевую роль.

5.4. Приложение ChatGPT

ChatGPT — это большая языковая модель, созданная OpenAI, основанная на архитектуре GPT-4. Это программа искусственного интеллекта, способная генерировать текстовые ответы на вопросы и запросы, представленные пользователем. ChatGPT обучается на огромном объеме текстовых данных, что позволяет ей имитировать человеческий язык и предоставлять информацию или разговорные ответы на различные темы. Модель способна на ведение диалога, генерацию идей, предоставление образовательной информации и даже творческую писательскую деятельность.

Использование ChatGPT довольно просто и интуитивно понятно. Вот основные шаги и советы по использованию этой системы:

Задайте ваш вопрос или сделайте запрос	<ul style="list-style-type: none">■ начните с формулирования вашего вопроса или запроса. Это может быть что угодно: вопрос по какой-либо теме, просьба о помощи в написании текста, создании идей и т. д.■ будьте как можно более конкретными в своих запросах для получения наиболее релевантных ответов.
Четкость и контекст	<ul style="list-style-type: none">■ если ваш вопрос требует определенного контекста, убедитесь, что вы включили всю необходимую информацию.■ если вы обсуждаете сложную или многоаспектную тему, разбейте свои вопросы на части, чтобы облегчить понимание и ответы.
Проверка ответов	ChatGPT стремится предоставить точную и полезную информацию, но всегда полезно перепроверять полученные данные, особенно если они касаются специализированных или очень актуальных тем.
Использование дополнительных функций	ChatGPT может выполнять различные задачи, такие как перевод текста, создание стихов, программирование и многое другое. Не стесняйтесь просить о разнообразных услугах.

Повторная формулировка вопросов	Если ответ не удовлетворяет ваш запрос, попробуйте переформулировать или уточнить ваш вопрос.
Обратная связь	Ваши отзывы помогают улучшить chatgpt, поэтому не стесняйтесь делиться своими мыслями о качестве и полезности полученных ответов.
Соблюдение норм и правил	Важно помнить, что есть определенные ограничения и нормы, которых придерживается chatgpt, включая отказ от создания оскорбительного контента или соблюдение авторских прав.
Терпение и эксперименты	Иногда может потребоваться несколько попыток, чтобы добиться желаемого ответа. Экспериментируйте с разными формулировками и подходами.

Следуя этим простым рекомендациям, вы сможете максимально эффективно использовать возможности ChatGPT.

5.5. ChatGPT для решения задач по программированию

Использование ChatGPT для решения задач по программированию требует некоторых специфических подходов. Вот подробная инструкция, как эффективно использовать ChatGPT в этом контексте:

Определите проблему	Четко опишите задачу программирования, с которой вы столкнулись. Включите конкретные детали, такие как язык программирования, тип задачи (например, алгоритмическая задача, ошибка в коде, разработка функции и т. д.).
Предоставьте контекст и код	<ul style="list-style-type: none"> ■ если у вас есть уже написанный код, связанный с вашим вопросом, предоставьте его. Это поможет ChatGPT лучше понять контекст и предложить более точное решение. ■ указывайте версию языка программирования или используемые библиотеки, если это важно для задачи.

Формулируйте вопросы ясно	Будьте конкретными в ваших вопросах. Например, вместо «почему мой код не работает?» лучше спросить: «Почему эта функция возвращает ошибку “NullReferenceException” в моем C# коде?».
Используйте ChatGPT для отладки	Если у вас есть проблема с кодом, попробуйте описать ошибку и спросить о возможных причинах и решениях.
Запросы на написание кода	Вы можете попросить chatgpt написать небольшие фрагменты кода или псевдокод для конкретной задачи.
Учебные материалы и концепции	Если вам нужно понять определенную концепцию программирования, попросите ChatGPT объяснить ее вам или предоставить учебные ресурсы.
Проверка и тестирование кода	После получения кода от chatgpt всегда проверяйте и тестируйте его в вашей среде разработки. ChatGPT может дать общее представление или направление, но код может требовать доработки или адаптации.
Улучшение навыков	Используйте chatgpt для задавания вопросов по лучшим практикам программирования, паттернам проектирования и способам оптимизации кода.
Обратная связь и уточнения	Если предложенное решение не совсем подходит, не стесняйтесь задавать дополнительные вопросы или уточнять ваш запрос.
Этика и честное использование	Помните об этических аспектах использования кода, сгенерированного chatgpt, особенно при использовании его в учебных или профессиональных целях.

Соблюдение этих рекомендаций поможет вам максимально эффективно использовать ChatGPT для решения задач по программированию.

Приведем пример написания кода на языке программирования PHP для нахождения минимального значения в матрице заданного размера, который выдает ChatGPT по запросу. Для начала нужно создать эту матрицу, а затем найти в ней минимальное значение.

Вот пример кода, который делает это: в этом коде есть две функции: **“generateMatrix”**, которая создает матрицу заданного размера с случайными значениями, и **“findMinInMatrix”**, которая ищет минимальное значение в этой матрице. Вы можете изменять размеры матрицы, а также минимальные и максимальные значения элементов, генерируемых в **“generateMatrix”**.

ГЛАВА 6

Практические аспекты обучения нейронных сетей

6.1. Подготовка данных и предобработка

Эта тема охватывает ряд важных шагов и методов, которые необходимы для эффективного обучения нейронных сетей. Вот ключевые моменты:

Подготовка данных	Важный начальный шаг, который включает сбор, оценку и организацию данных, которые будут использоваться для обучения сети. Это может включать действия, такие как сбор больших объемов данных, их очистка от ошибок или неполных записей, и разделение на наборы для обучения, валидации и тестирования.
Предобработка данных	Прежде чем данные можно использовать для обучения, их часто нужно предварительно обработать. Это может включать масштабирование или нормализацию данных (например, приведение всех значений к диапазону от 0 до 1), кодирование категориальных данных, обработку пропущенных значений и т. д. Предобработка помогает улучшить производительность и стабильность процесса обучения.
Выбор и архитектура модели	Выбор правильной архитектуры нейронной сети в зависимости от задачи (например, сверточные нейронные сети для обработки изображений, рекуррентные нейронные сети для работы с последовательностями и т. д.).
Обучение модели	Под этим подразумевается настройка весов сети на основе данных обучения. Это включает в себя использование алгоритмов обратного распространения ошибки

	и оптимизаторов (например, SGD, Adam), настройку гиперпараметров (скорость обучения, размер партии и т. д.), а также мониторинг за переобучением и использование техник, таких как ранняя остановка или регуляризация для его предотвращения.
Валидация и тестирование	После обучения модели она тестируется на отдельном наборе данных, чтобы оценить её производительность. Это помогает понять, насколько хорошо модель обобщает данные, которые она ранее не видела.
Оптимизация и настройка	После первоначального обучения и тестирования, процесс обучения может быть дополнительно оптимизирован путем настройки гиперпараметров, использования более сложных архитектур сетей или внедрения дополнительных техник обработки данных.
Развертывание	После обучения и валидации модель готова к развертыванию в реальном приложении или системе.

Каждый из этих шагов критически важен для успешного применения нейронных сетей в различных областях, от распознавания образов до обработки естественного языка.

6.2. Выбор функций ошибок и метрик качества

Функция ошибки в контексте обучения нейронных сетей — это основной компонент, который используется для оценки «ошибки» или «расхождения» между предсказаниями модели и фактическими значениями. Эта функция является ключевой в процессе обучения нейронных сетей. Вот основные аспекты функций ошибки:

НАЗНАЧЕНИЕ ФУНКЦИЙ ОШИБКИ

Оценка ошибки	Функция ошибки вычисляет разницу между выходными данными модели и истинными значениями. Меньшие значения ошибки указывают на лучшую производительность модели.
----------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------

Обучение модели	В процессе обучения нейронной сети функция ошибки используется для определения того, насколько хорошо модель работает, и для корректировки её весов и параметров с целью минимизации этих ошибки.
Обратное распространение (Backpropagation)	функция ошибки играет центральную роль в алгоритме обратного распространения, который используется для эффективного обучения глубоких нейронных сетей. Градиенты функции ошибки распространяются обратно по сети, обновляя веса и параметры.

ТИПЫ ФУНКЦИЙ ОШИБКИ

Средняя квадратичная ошибка (MSE)	Часто используется в задачах регрессии. Она измеряет средний квадрат различий между предсказанными и истинными значениями.
Перекры́стная энтропия (Cross-Entropy)	Широко применяется в задачах классификации. Она измеряет разницу между двумя вероятностными распределениями - предсказанным распределением и истинным распределением.
Функции ошибки для конкретных задач	Существуют и более специализированные функции ошибки, разработанные для конкретных приложений, например, функции ошибки для задач сегментации изображений или рекомендательных систем.
Значение в обучении модели	<ul style="list-style-type: none">■ Функция ошибки напрямую влияет на то, как модель «учится» на данных. Она определяет, какие аспекты производительности модели считаются важными и как модель должна корректировать свои предсказания, чтобы улучшить эти аспекты.■ Выбор правильной функции ошибки для конкретной задачи является критически важным шагом, поскольку неподходящая функция ошибки может привести к неэффективному обучению или даже к невозможности модели достичь приемлемых результатов.

В общем, функция ошибки является основополагающим элементом в процессе обучения нейронных сетей, определяющим, как модель извлекает информацию из данных и как она адаптируется для выполнения заданной задачи.

Метрики качества в контексте обучения машинного обучения и нейронных сетей — это стандарты или меры, используемые для оценки и интерпретации производительности модели. Эти метрики помогают понять, насколько хорошо модель выполняет заданную задачу, и часто используются для сравнения различных моделей или для настройки гиперпараметров. Вот несколько ключевых аспектов метрик качества:

ОБЩИЕ ВИДЫ МЕТРИК КАЧЕСТВА

Точность (Accuracy)	Доля правильных предсказаний из общего числа случаев. Часто используется в задачах классификации.
Полнота (Recall) и точность (Precision)	Важны в ситуациях, где важны ложноположительные и ложноотрицательные результаты. Полнота измеряет, какую долю реальных положительных случаев модель определила правильно, в то время как точность отражает, какая доля предсказанных положительных случаев была действительно правильной.
F1-Score	Гармоническое среднее между точностью и полнотой. Особенно полезна в случаях, где баланс между полнотой и точностью критичен.
ROC-AUC	Площадь под кривой рабочих характеристик приемника (Receiver Operating Characteristic). Оценивает способность модели различать классы при разных пороговых значениях.

Средняя квадратичная ошибка (MSE) и средняя абсолютная ошибка (MAE): часто используются в задачах регрессии для измерения разницы между предсказанными значениями и фактическими.

ЗНАЧЕНИЕ МЕТРИК КАЧЕСТВА

Оценка производительности	Метрики дают количественное представление о производительности модели, позволяя оценить, насколько хорошо модель работает на задаче.
----------------------------------	--------------------------------------------------------------------------------------------------------------------------------------

Выбор и сравнение моделей	Метрики качества используются для сравнения разных моделей или подходов, чтобы выбрать наилучший вариант.
Руководство для настройки	Они могут указывать на области, которые нуждаются в улучшении, и могут направлять процесс настройки гиперпараметров.
Адаптация к специфике задачи	Важно выбирать метрики, которые наилучшим образом отражают цели и ограничения конкретной задачи или приложения.

Различие с функцией ошибки

В то время как функция потерь используется в процессе обучения модели для оптимизации её параметров, метрики качества чаще используются после обучения для оценки производительности модели и не влияют непосредственно на процесс обучения.

В целом, метрики качества являются важным инструментом в области машинного обучения и искусственного интеллекта, предоставляя необходимые средства для объективной оценки и сравнения различных моделей и подходов.

ГЛАВА 7

Приложения и последние тенденции

7.1. Нейронные сети в медицине

Нейронные сети находят широкое применение в медицине, благодаря своей способности анализировать большие объемы данных и выявлять сложные закономерности. Вот некоторые ключевые области их использования:

ДИАГНОСТИКА И АНАЛИЗ ИЗОБРАЖЕНИЙ

Радиология	Нейронные сети, особенно сверточные (CNN), используются для анализа медицинских изображений (например, рентгеновских снимков, МРТ, КТ), помогая в диагностике заболеваний, таких как рак, пневмония и др.
-------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Патология	Автоматизированный анализ образцов тканей для выявления признаков заболеваний.
------------------	--------------------------------------------------------------------------------

ПРОГНОЗИРОВАНИЕ И ЛИЧНАЯ МЕДИЦИНА

Прогнозирование заболеваний	Используя данные электронных медицинских записей, нейронные сети могут предсказывать риск развития определенных заболеваний, например, диабета или сердечно-сосудистых заболеваний.
------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Анализ генетической информации	Для определения риска наследственных заболеваний или реакции на лекарственные препараты.
---------------------------------------	------------------------------------------------------------------------------------------

ЛЕЧЕНИЕ И УПРАВЛЕНИЕ ЗАБОЛЕВАНИЯМИ

Разработка лекарств	Использование нейронных сетей для предсказания взаимодействия молекул и помощи в создании новых лекарственных средств.
----------------------------	------------------------------------------------------------------------------------------------------------------------

Персонализированное лечение	Анализ медицинских данных пациента для разработки индивидуализированных планов лечения.
------------------------------------	-----------------------------------------------------------------------------------------

ОПЕРАЦИОННЫЕ ТЕХНОЛОГИИ

Роботизированная хирургия	Нейронные сети могут анализировать хирургические процедуры и помогать в управлении роботизированными хирургическими системами для повышения точности и безопасности операций.
----------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

МОНИТОРИНГ И ПОДДЕРЖКА ПРИНЯТИЯ РЕШЕНИЙ

Мониторинг пациентов	Анализ данных, полученных от медицинских устройств (например, ЭКГ), для раннего выявления проблем.
-----------------------------	----------------------------------------------------------------------------------------------------

Поддержка клинических решений	Системы, основанные на нейронных сетях, могут предоставлять рекомендации врачам на основе медицинских данных и исследований.
--------------------------------------	------------------------------------------------------------------------------------------------------------------------------

ПРЕИМУЩЕСТВА И ВЫЗОВЫ

Преимущества	Нейронные сети могут обрабатывать и анализировать гораздо больше данных, чем человек, и иногда находить закономерности, которые не видны врачам.
---------------------	--------------------------------------------------------------------------------------------------------------------------------------------------

Вызовы:	Вопросы конфиденциальности данных, необходимость больших объемов обучающих данных, риск ошибок и вопросы этики являются ключевыми вызовами в этой области.
----------------	------------------------------------------------------------------------------------------------------------------------------------------------------------

В целом, использование нейронных сетей в медицине открывает множество возможностей для улучшения диагностики, лечения и управления здравоохранением, хотя это также сопряжено с определенными вызовами и требует тщательного учета этических и практических соображений.

7.2. Нейронные сети в экономике

Нейронные сети находят широкое применение в экономике, где они используются для анализа, прогнозирования и оптимизации различных экономических процессов. Их способность обрабатывать большие объемы данных и выявлять сложные закономерности делает их ценным инструментом в этой области. Вот некоторые из ключевых способов использования нейронных сетей в экономике:

Прогнозирование финансовых рынков	Нейронные сети используются для анализа и прогнозирования движения цен на акции, валютные курсы и другие финансовые показатели. Они могут обучаться на исторических данных для выявления трендов и закономерностей, которые могут быть неочевидны для традиционных аналитических методов.
Риск-менеджмент и кредитный скоринг	В банковском секторе нейронные сети применяются для оценки кредитоспособности клиентов и управления кредитными рисками. Они помогают банкам и финансовым институтам точнее определять вероятность невыплаты по кредитам, а также находить оптимальные стратегии кредитования.
Эконометрика и анализ экономических данных	Нейронные сети используются для анализа и моделирования экономических данных, таких как ВВП, инфляция, уровень безработицы и т.д. Они способны выявлять сложные взаимосвязи между различными экономическими показателями.
Оптимизация цепочек поставок и производства	Применяя нейронные сети для анализа и прогнозирования спроса, предприятия могут оптимизировать свои производственные процессы и управление запасами, уменьшая издержки и повышая эффективность.
Маркетинговый анализ	В сфере маркетинга нейронные сети помогают анализировать потребительское поведение, прогнозировать отклик на маркетинговые кампании и оптимизировать стратегии продвижения продуктов и услуг.
Автоматизация принятия решений	В корпоративной среде нейронные сети могут анализировать большие объемы данных для поддержки принятия решений, обеспечивая предприятиям ценные

	инсайты для разработки стратегий и управления ресурсами.
Прогнозирование макроэкономических показателей	Использование нейронных сетей для анализа и прогнозирования макроэкономических тенденций помогает правительствам и политикам в принятии обоснованных решений, касающихся экономической политики.

В экономике нейронные сети предлагают новые возможности для анализа и интерпретации данных, что позволяет более точно и эффективно принимать решения, основанные на данных. Однако их применение также требует внимания к качеству и представительности данных, а также к возможным проблемам, связанным с их интерпретацией и этическими аспектами.

7.3. Автоматическое управление и умные системы

Применение нейронных сетей в автоматическом управлении и в разработке умных систем охватывает широкий спектр областей и приложений. Нейронные сети способны обучаться, адаптироваться и принимать сложные решения, что делает их идеальными для следующих задач:

Робототехника и автоматизация	Нейронные сети используются для управления роботами, позволяя им адаптироваться к изменяющимся условиям и выполнять сложные задачи, такие как сборка, сортировка и перемещение объектов.
Автономное вождение	В автомобилях с автономным управлением нейронные сети анализируют данные с датчиков и камер для обеспечения безопасного вождения, распознавания объектов, принятия решений и предсказания действий других участников дорожного движения.
Умные дома и автоматизация зданий: нейронные сети могут управлять системами освещения	Отопления, вентиляции и кондиционирования, адаптируясь к предпочтениям пользователей и изменяющимся условиям окружающей среды.

Промышленное управление и мониторинг	В промышленности нейронные сети используются для оптимизации производственных процессов, обнаружения аномалий, прогнозирования неисправностей и управления качеством продукции.
Системы безопасности и наблюдения	Используя распознавание образов и анализ видео, нейронные сети способны обнаруживать подозрительные действия, идентифицировать лица и следить за безопасностью объектов.
Управление энергосистемами	В энергетике нейронные сети применяются для прогнозирования потребления энергии, управления распределением нагрузки и интеграции возобновляемых источников энергии.
Автоматизация в сельском хозяйстве	Нейронные сети помогают в управлении сельскохозяйственной техникой, анализе состояния урожая, определении необходимости полива и удобрений.
Медицинские и здравоохранительные системы	Нейронные сети используются для анализа медицинских данных, диагностики заболеваний, управления больничными системами и создания персонализированных планов лечения.

Эти примеры показывают, как нейронные сети могут значительно улучшить автоматизацию и умные системы, делая их более адаптивными, эффективными и способными к самообучению.

7.4. Развитие нейросетей в будущем

Развитие нейросетей в будущем, вероятно, будет сосредоточено на нескольких ключевых областях, которые значительно расширят их возможности и области применения. Вот несколько основных направлений, которые могут определить будущее нейросетей:

Улучшение алгоритмов обучения	Исследования будут сосредоточены на создании более эффективных алгоритмов обучения, которые могут обучаться быстрее, требуют меньше данных и более устойчивы к ошибкам.
--------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Развитие искусственного интеллекта общего назначения (AGI): хотя современные нейросети способны выполнять конкретные задачи, будущие исследования могут сосредоточиться на разработке AGI, способного выполнять широкий спектр задач с человеческим уровнем интеллекта.

Энергоэффективность и устойчивость	Развитие более энергоэффективных нейросетей, которые требуют меньше ресурсов для обучения и работы, будет критически важным для устойчивого развития технологий ИИ.
Интеграция с квантовыми технологиями	Квантовые компьютеры могут революционизировать область машинного обучения, предоставляя новые способы обработки информации и решения сложных задач.
Этические и правовые аспекты	По мере развития нейросетей возникнет необходимость в разработке этических норм и законодательных рамок для регулирования использования искусственного интеллекта.
Развитие интерфейсов мозг-компьютер	Улучшение технологий, позволяющих напрямую связывать человеческий мозг с компьютерными системами, может привести к новым способам взаимодействия с технологиями и улучшению когнитивных способностей.
Объединение нейросетей с другими технологиями	Интеграция нейросетей с другими передовыми технологиями, такими как биотехнологии, нанотехнологии и робототехника, может привести к созданию совершенно новых систем и устройств.
Личная кастомизация и адаптация	нейросети будущего могут быть способны адаптироваться к индивидуальным пользователям, предлагая более персонализированные и эффективные решения.

Эти направления подчеркивают потенциал нейросетей в будущем, который может радикально изменить многие аспекты жизни, от науки и технологий до образования и здравоохранения.

ЗАКЛЮЧЕНИЕ

Подведение итогов

Изучение нейронных сетей — это многообещающее и динамично развивающееся направление в сфере искусственного интеллекта и машинного обучения. Подводя итоги, можно выделить несколько ключевых выводов:

Многообразие ресурсов	Существует множество качественных ресурсов для изучения нейронных сетей — от специализированных книг и онлайн-курсов до интерактивных платформ и открытых исследовательских проектов.
Практический опыт	Важным аспектом обучения является практический опыт. Работа над реальными проектами и участие в соревнованиях по машинному обучению помогают углубить понимание теории и развить практические навыки.
Непрерывное обучение	Область нейронных сетей постоянно развивается, поэтому важно постоянно обновлять свои знания, следить за новыми исследованиями и тенденциями в этой области.
Широкий спектр применения	Нейронные сети находят применение в различных сферах, от автономных транспортных средств и медицинской диагностики до финансового анализа и создания искусственного интеллекта, что открывает широкие возможности для профессионального применения полученных знаний.
Комбинация теории и практики	изучение нейронных сетей требует сочетания теоретических знаний о математических и алгоритмических основах с практическим применением этих знаний для решения реальных задач.
Важность сообщества	участие в сообществах, посвященных ИИ и машинному обучению, способствует обмену знаниями, идеями и опытом, что является важным фактором для профессионального роста в этой области.

В заключение, изучение нейронных сетей открывает перспективные возможности для карьеры и научных исследований, требует постоянного обучения и экспериментирования, и предлагает широкий спектр практических применений своих знаний.

Рекомендации по дополнительному изучению

Вот несколько популярных онлайн-курсов по нейронным сетям на русском языке:

Нетология Data Scientist	Этот курс считается лучшим по отзывам и предлагает глубокое погружение в Data Science и нейронные сети. (https://netology.ru/programs/data-scientist).
Skillbox Профессия Data Scientist	Этот курс является одним из самых популярных и охватывает широкий спектр тем, связанных с Data Science, включая нейронные сети. (https://skillbox.ru/course/profession-data-scientist-trial)
SkillFactory Курс по нейронным сетям	Этот курс специально сосредоточен на нейронных сетях и глубоком обучении. (https://skillfactory.ru/neironnye-seti-deep-learning)
Специалист.ру Нейронные сети. Компьютерное зрение и чтение (NLP)	Курс охватывает конкретные области применения нейронных сетей, такие как компьютерное зрение и обработка естественного языка. (https://www.specialist.ru/course/pyml2)
SkillFactory AI разработчик	Этот курс предлагает обучение по искусственному интеллекту и нейронным сетям для тех, кто хочет стать специалистом в этой области. (https://skillfactory.ru/specialist-po-iskusstvennomu-intellektu).

Эти курсы предоставляют обширные знания и практические навыки в области нейронных сетей и их применения в различных сферах.

Для дополнительного изучения нейронных сетей вам могут быть полезны следующие рекомендации.

ОНЛАЙН КУРСЫ И ОБУЧАЮЩИЕ ПЛАТФОРМЫ

Coursera и edX	Предлагают курсы от ведущих университетов и институтов по темам искусственного интеллекта, машинного обучения и глубокого обучения.
Udacity и Udemy	Предоставляют специализированные наностепени и курсы, сфокусированные на практических аспектах нейронных сетей.
DeepLearning.AI	Серия курсов от Эндрю Н., Одного из ведущих экспертов в области машинного обучения.

Основные книги, статьи и учебные пособия

Вот несколько ключевых книг по нейронным сетям, которые рекомендуются для изучения и понимания глубокого обучения и искусственных нейронных сетей:

“Deep Learning”, авторы Ian Goodfellow, Yoshua Bengio и Aaron Courville	Основополагающая книга в области глубокого обучения
“Python Deep Learning”, авторы Ivan Vasilev и Daniel Slater	Практическое руководство по использованию Python в глубоком обучении
“Neural Networks for Pattern Recognition”, автор Christopher M. Bishop	
“Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks”, авторы Russell Reed & Robert J MarksII	Эта книга полезна для практических приложений алгоритмов и содержит код и графики, которые являются её основными элементами
“Deep Learning” (серия Adaptive Computation and Machine Learning), авторы Ian, Yoshua и Aaron	Эта книга соединяет современные и классические концепции глубокого обучения.
“Neural Networks and Deep Learning: A Textbook”, автор Charu C. Aggarwal	Эта книга ориентирована на теорию и включает кодирование и сценарии на Python, обсуждающие основные техники машинного обучения и глубокого обучения
“Deep Learning with Python”, автор François Chollet	Книга использует практический подход к теории и содержит множество примеров

	для глубокого обучения с контекстом компьютерного зрения, текста и последовательностей
“Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems”, автор Aurélien Géron	Эта книга охватывает как основы машинного обучения, так и теоретические аспекты глубокого обучения через TensorFlow.
“TensorFlow 1.x Deep Learning Cookbook: Over 90 unique recipes to solve artificial-intelligence driven problems with Python”, авторы Antonio Gulli & Amita Kapoor	Эта книга фокусируется на кодировании и решении задач искусственного интеллекта с помощью Python.
“Deep Learning: A Practitioner’s Approach”, авторы Josh Patterson & Adam Gibson	Книга использует уникальный подход к обучению кодированию, используя Java и библиотеки DL4J.
“Deep Learning for Computer Vision: Expert techniques to train advanced neural networks using TensorFlow and Keras”, автор Rajalingappa Shanmugamani	Эта книга сосредоточена на компьютерном зрении и современных методологиях
“Deep Learning in Computer Vision: Principles and Applications (Digital Imaging and Computer Vision)”, авторы Mahmoud Hassaballah & Ali Ismail Awad	Эта книга представляет нетрадиционные подходы к решению проблем, связанных с обработкой изображений.
«Нейронные сети, генетические алгоритмы и нечеткие системы», автор Д. Рутковская	(2023)
«Искусственные нейронные сети», автор В. С. Ростовцев	Учебник для вузов, 3-е изд. (2023)
«Нейронные сети», автор Р. Тадеусевич	Толковый словарь (2023)
«Создаем нейронную сеть», автор Рашид Тарик	(2018)
«Нейронные сети», автор Хайкин Саймон	Полный курс (2019)

«Как учится машина: Революция (2021)
в области нейронных сетей и
глубокого обучения», автор Леун Ян

Эти книги представляют собой как базовые учебники для тех, кто только начинает знакомиться с нейронными сетями, так и более продвинутые ресурсы для глубокого понимания темы.

ИНТЕРНЕТ-РЕСУРСЫ И ФОРУМЫ

GitHub	Исследуйте открытые проекты и библиотеки, связанные с нейронными сетями.
Stack Overflow и Reddit	Активные сообщества, где можно задавать вопросы и обсуждать проблемы, связанные с нейронными сетями.
ArXiv и Google Scholar	Чтение актуальных исследовательских работ и статей по нейронным сетям.

Практический опыт и проекты

Работайте над собственными проектами или участвуйте в соревнованиях по машинному обучению на платформах, таких как Kaggle.

Используйте данные с открытых источников, таких как UCI Machine Learning Repository или Google Dataset Search, для практики и экспериментов.

Постоянное обучение и развитие

Подписывайтесь на блоги и каналы, посвященные искусственному интеллекту и машинному обучению.

Посещайте конференции и вебинары, чтобы быть в курсе последних трендов и достижений в области нейронных сетей.

Помните, что ключ к успешному освоению нейронных сетей — это сочетание теоретических знаний и практического опыта. Экспериментируйте с разными подходами и технологиями, чтобы найти то, что лучше всего соответствует вашим интересам и целям обучения.

ЛИТЕРАТУРА

СПИСОК ИСТОЧНИКОВ

1. *Bishop C. M.* Neural Networks for Pattern Recognition. — Oxford University Press, 1995.
2. *Reed R., Marks II R. J.* Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks. — MIT Press, 1999.
3. *Goodfellow I., Bengio Y., Courville A.* Deep Learning. (Adaptive Computation and Machine Learning series). — MIT Press, 2016.
4. *Aggarwal C. C.* Neural Networks and Deep Learning: A Textbook. — Springer, 2018.
5. *Chollet F.* Deep Learning with Python. — Manning Publications, 2017.
6. *Géron A.* Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. — O'Reilly Media, 2019.
7. *Gulli A., Kapoor A.* TensorFlow 1.x Deep Learning Cookbook. — Packt Publishing, 2017.
8. *Patterson J., Gibson A.* Deep Learning: A Practitioner's Approach. — O'Reilly Media, 2017.
9. *Shanmugamani R.* Deep Learning for Computer Vision: Expert techniques to train advanced neural networks using TensorFlow and Keras. — Packt Publishing, 2018.
10. *Hassaballah M., Awad A. I.* Deep Learning in Computer Vision: Principles and Applications (Digital Imaging and Computer Vision). — CRC Press, 2020.
11. *Рутковская Д.* Нейронные сети, генетические алгоритмы и нечеткие системы. — М.: Горячая линия — Телеком, 2023.

12. *Ростовцев В. С.* Искусственные нейронные сети. Учебник для вузов. 3-е изд. — М.: Высшая школа, 2023.
13. *Тадеусевич Р.* Нейронные сети. Толковый словарь. — М.: Радио и связь, 2023.
14. *Рашид Т.* Создаем нейронную сеть. — Питер, 2018.
15. *Хайкин С.* Нейронные сети. Полный курс. — М.: Вильямс, 2019.
16. *Лекун Я.* Как учится машина: Революция в области нейронных сетей и глубокого обучения. — М.: Альпина Паблишер, 2021.
17. Нетология. Data Scientist [Электронный ресурс]. — Режим доступа: <https://netology.ru/programs/data-scientist>.
18. Skillbox. Профессия Data Scientist [Электронный ресурс]. — Режим доступа: <https://skillbox.ru/course/profession-data-scientist-trial>.
19. SkillFactory. Курс по нейронным сетям [Электронный ресурс]. — Режим доступа: <https://skillfactory.ru/nejronnye-seti-deep-learning>.
20. Специалист.ру. Нейронные сети. Компьютерное зрение и чтение (NLP) [Электронный ресурс]. — Режим доступа: <https://www.specialist.ru/course/pyml2>.
21. SkillFactory. AI разработчик [Электронный ресурс]. — Режим доступа: <https://skillfactory.ru/specialist-po-iskusstvennomu-intellektu>.

ПРИЛОЖЕНИЕ 1

Тренировочные задания «Нейро-практикум»

Мы рады предложить вам уникальную возможность погрузиться в мир искусственного интеллекта и машинного обучения с помощью нашего учебного пособия, и сопутствующих материалов. Для того чтобы ваше обучение было максимально эффективным и практическим, мы подготовили специальные датасеты, которые будут незаменимыми в ходе выполнения нейро-практикумов.

Доступ к этим датасетам организован через нашу группу в Telegram (ссылка <https://t.me/+ZkL2a-WQW4lmZWYy>).

Мы создали QR-код, который легко и быстро перенаправит вас прямо в группу, где вы сможете скачать необходимые данные. Этот подход позволит вам без труда получать актуальные датасеты и делиться своими достижениями и опытом с единомышленниками.

Чтобы воспользоваться этой возможностью, вам лишь нужно отсканировать QR-код, расположенный в нашем пособии. Для этого используйте камеру вашего смартфона или любое приложение для сканирования QR-кодов. После сканирования вы будете автоматически переадресованы в группу Telegram, где найдете все необходимые материалы.

QR-код



Мы уверены, что эти датасеты станут весомым вкладом в ваше обучение и позволят глубже понять принципы работы нейросетей на практике. Не упустите шанс расширить свои знания и навыки, используя актуальные и качественные данные для тренировки искусственного интеллекта.

Нейро-практикум № 1 Исследование функций активаций нейронных сетей

Цель:

Исследование функций активаций нейронных сетей

Задачи:

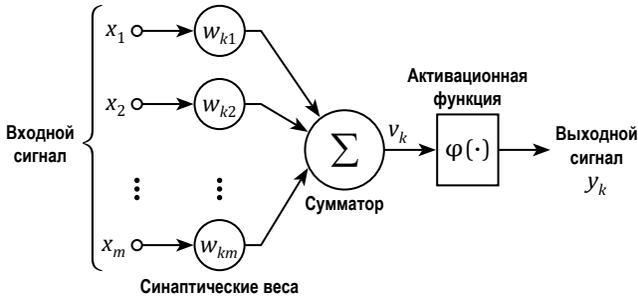
1. Построить графики функций активации (Глава 1, таблица 1) в заданных диапазонах значений в соответствии с вариантом (таблица 2).
2. Составить и распечатать отчет, который должен содержать:
 - титульный лист;
 - цель лабораторной работы;
 - название, описание функции активации, график функции.

Таблица 2

№	Диапазон входных значений	№	Диапазон входных значений	№	Диапазон входных значений
1	-3...+3	11	-4...+4	21	-2...+2
2	-1...+1	12	-5...+5	22	-8...+8
3	-4...+4	13	-2...+2	23	-7...+7
4	-5...+5	14	-8...+8	24	-10...+10
5	-2...+2	15	-7...+7	25	-3...+3
6	-8...+8	16	-10...+10	26	-1...+1
7	-7...+7	17	-3...+3	27	-4...+4
8	-10...+10	18	-1...+1	28	-1...+1
9	-3...+3	19	-4...+4	29	-4...+4
10	-6...+6	20	-5...+5	30	-5...+5

Нейро-практикум № 2 Математическая модель нейрона
Цель:

Изучение структурных схем модели нейрона с представлением их в виде блочных диаграмм. Решение задач.

Общие сведения. Описание условия задачи:

Нейрон получает входной сигнал x_m , где $m = 4$, блочная диаграмма данной модели представлена на рисунке

$$x_1 = 10, x_2 = -20, x_3 = 4, x_4 = -2.$$

Соответствующие весовые коэффициенты нейрона равны

$$w_{11} = 0.8, w_{12} = 0.2, w_{13} = 0.4, w_{14} = -0.9.$$

Вычислить выходное значение нейрона, модель которого описывается сигмоидальной функцией активации. Предполагается, что порог отсутствует.

Пример решения:

Сигмоидальная функция активации имеет вид:

$$\varphi(v) = \frac{1}{1 + e^{-v}}.$$

Сделаем вычисления согласно следующим формулам:

$$u_k = \sum_{j=0}^m w_{kj} x_j,$$

$$v_k = u_k + b_k,$$

$$y_k = \varphi(u_k).$$

Тогда,

$$u_1 = x_1 w_{11} + x_2 w_{12} + x_3 w_{13} + x_4 w_{14}.$$

$$u_1 = 10 \cdot 0.8 + (-20 \cdot 0.2) + 4 \cdot 0.4 + (-2 \cdot -0.0) = 7.4$$

$$v_1 = u_1 + b_1$$

$b_1 = 0$, т. к. по условию порог отсутствует

$$v_1 = 7.4$$

$$y_1 = \frac{1}{1 + e^{-7.4}} = 0.99$$

Таким образом, выход нейронной сети на данное входное множества равен $y_1 = 0.99$

Задача:

Нейрон получает входной сигнал x_m и значения весовых коэффициентов и соответствующие им весовые коэффициенты нейрона (см. таблицу № 3).

1. Вычислить выходное значение нейрона, модель которого описывается сигмоидальной функцией активации. Предполагается, что порог равен +1.
2. Нарисовать блочную диаграмму модель нейронной сети.
3. Составить и распечатать отчет, который должен содержать:
 - титульный лист;
 - расчеты;

Таблица №3

№	m	Входные значения	Весовые коэффициенты	№	m	Входные значения	Весовые коэффициенты
1	3	[2,5,4]	[1,0.5,2]	16	3	[2,5,4]	[1,0.5,2]
2	4	[1,8,3,9]	[0.2,-1, 2,-0.3]	17	4	[1,8,3,9]	[0.2,-1, 2,-0.3]
3	5	[3,4,8,4,1]	[-0.1,-3.5,-0.7,0.6]	18	5	[3,4,8,4,1]	[-0.1,-3.5,-0.7,0.6]

№	m	Входные значения	Весовые коэффициенты	№	m	Входные значения	Весовые коэффициенты
4	2	[3,5]	[-3,5]	19	2	[3,5]	[-3,5]
5	4	[1,8,3,4]	[-3,-0.7,0,6]	20	4	[1,8,3,4]	[-3,-0.7,0,6]
6	3	[4,8,4]	[1,0.5,2]	21	3	[4,8,4]	[1,0.5,2]
7	4	[5,4,3,2]	[0.2,-1, 2,-0.3]	22	4	[5,4,3,2]	[0.2,-1, 2,-0.3]
8	5	[1,3,5,7,9]	[-0.1,-3,5,-0.7,0.5]	23	5	[1,3,5,7,9]	[-0.1,-3,5,-0.7,0.5]
9	2	[2,6]	[-3,5]	24	2	[2,6]	[-3,5]
10	4	[7,6,5,4]	[-3,-0.7,0,6]	25	4	[1,8,3,4]	[-3,-0.7,0,6]
11	3	[2,5,4]	[1,0.5,2]	26	3	[2,5,4]	[1,0.5,2]
12	4	[1,8,3,9]	[0.2,-1, 2,-0.3]	27	4	[1,8,3,9]	[0.2,-1, 2,-0.3]
13	5	[3,4,8,4,1]	[-0.1,-3,5,-0.7,0.6]	28	5	[3,4,8,4,1]	[-0.1,-3,5,-0.7,0.6]
14	2	[3,5]	[-3,5]	29	2	[3,5]	[-3,5]
15	4	[1,8,3,4]	[-3,-0.7,0,6]	30	4	[1,8,3,4]	[-3,-0.7,0,6]

Нейро-практикум № 3 Многослойные нейронные сети

Цель:

Изучение структурных схемы модели многослойной нейронной сети с представлением их в виде ориентированных графов. Решение задач.

Общие сведения. Описание условия задачи:

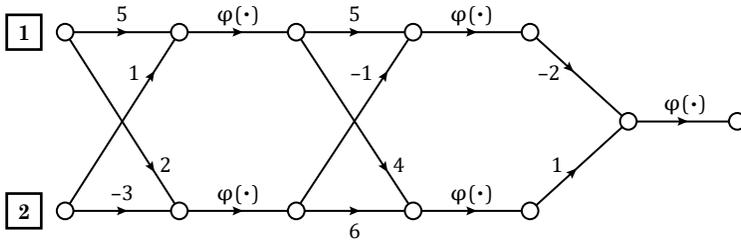
На рисунке представлен граф прохождения сигнала по сети прямого распространения вида 2-2-2-1. Функция активации является логистической. Описать отображение вход-выход. Для входа равного [1, 2].

Сигмоидальная функция активации имеет вид:

$$\varphi(v) = \frac{1}{1 + e^{-v}}$$

Прочитав граф передачи сигнала, можно сказать следующие:

1. Нейронная сеть имеет 1 входной слой, состоящий из двух элементов; 1 входной слой нейронов, состоящий из двух нейронов, один скрытый слой, состоящий из двух нейронов и один выходной слой, состоящий из одного нейрона. (2-2-2-1).
2. Весовые коэффициенты первого слоя равны 5, 1, 2, -3.
3. Весовые коэффициенты скрытого слоя равны 3, -1, 4, 6.
4. Весовые коэффициенты выходного слоя равны -2, 1.



Вычисления согласно следующим формулам

Тогда

Вычислим выходной сигнал первого слоя

$$u_1 = 1 \cdot 5 + 2 \cdot 1 = 7$$

$$u_2 = 1 \cdot 2 + 2 \cdot -3 = -4$$

$$v_1 = u_1 + b_1$$

$b_1 = 0$, $b_2 = 0$, т. к. по условию порог отсутствует

$$v_1 = 7, v_2 = -4$$

$$y_1 = \frac{1}{1 + e^{-7}} = 0.9991, y_2 = \frac{1}{1 + e^4} = 0.0180$$

Вычислим выходной сигнал второго слоя

$$u_1 = 0.9991 \cdot 3 + 0.0180 \cdot -1 = 2.99 - 0.0180 = 2.972$$

$$u_2 = 0.9991 \cdot 4 + 0.0180 \cdot 6 = 3.964 + 0.108 = 4.072$$

$$v_1 = u_1 + b_1, v_2 = u_2 + b_2$$

$b_1 = 0, b_2 = 0$, т. к. по условию порог отсутствует

$$v_1 = 2.972, v_2 = 4.072$$

$$y_1 = \frac{1}{1 + e^{-2.972}} = 0.9513, y_2 = \frac{1}{1 + e^{-4.0724}} = 0.9832$$

Вычислим выходной сигнал выходного слоя

$$u_1 = 0.9513 \cdot -2 + 0.9832 \cdot 1 = -0.9194$$

$$v_1 = u_1 + b_1$$

$b_1 = 0, b_2 = 0$, т. к. по условию порог отсутствует

$$v_1 = -0.9194$$

$$y_1 = \frac{1}{1 + e^{0.9194}} = 0.2851$$

Таким образом, выход нейронной сети на данное входное множества равен $y_1 = 0.2851$

Задача:

1. Нарисовать граф прохождения сигнала по многослойной нейронной сети прямого распространения вида, заданного вариантом (см. таблицу № 4). Описать отображение вход-выход.
2. Нарисовать блочную диаграмму модель нейронной сети.
3. Составить и распечатать отчет, который должен содержать:
 - титульный лист;
 - расчеты.

Таблица №4

№	Вид сети	Функции активации	Входное значение	Весовые коэф-ты 1 слоя	Весовые коэф-ты 2 слоя	Весовые коэф-ты 3 слоя
1	2-3-3-1	Пороговая Сигмоидальная	[1,1]	[1,2,1,3,-5,2]	[6,3,-2,1,-1,-0,5,-1,3,7,]	[2,1,3]
2	3-3-2-1	Сигмоидальная Радиальная базисная	[1,2,3]	[6,3,-2,1,-1,-0,5,-1,3,7,]	[1,2,1,3,-5,2]	[1,1]
3	2-3-2-1	Линейная Сигмоидальная	[4,6]	[2,1,3,-0,3,-4,3]	[1,2,1,3,-5,2]	[-2,1]
4	2-2-3-1	Радиальная базисная Сигмоидальная	[-2,1]	[-4,2]	[1,2,1,3,-5,2]	[2,1,3]
5	3-2-4-1	Линейная с насыщением Сигмоидальная	[0,-1,5]	[2,1,3,-0,3,-4,3]	[6,3,-2,1,-1,-0,5,-1,3]	[2,1,3,1]
6	2-3-4-1	Полулинейная Сигмоидальная	[-4,2]	[1,2,1,3,-5,2]	[6,3,-2,1,-1,-0,5,-1,3,7,1,2,4]	[2,1,3,1]
7	3-4-2-1	Гиперболический тангенс Сигмоидальная	[3,-3,1]	[6,3,-2,1,-1,-0,5,-1,3,7,1,2,4]	[6,3,-2,1,-1,-0,5,-1,3,7,]	[-2,1]
8	2-2-3-1	Полулинейная с насыщением Сигмоидальная	[1,-9]	[-4,2]	[1,2,1,3,-5,2]	[2,1,3]

№	Вид сети	Функции активации	Входное значение	Весовые коэф-ты 1 слоя	Весовые коэф-ты 2 слоя	Весовые коэф-ты 3 слоя
9	3-3-4-1	Знаковая Сигмоидальная	[-3,-2,4]	[6,3,-2,1,-1,-0,5,-1,3,7,]	[6,3,-2,1,-1,-0,5,-1,3,7,1,2,4]	[2,1,3,1]
10	2-4-3-1	Полулинейная с насыщением Сигмоидальная	[5,1]	[6,3,-2,1,-1,-0,5,-1,3]	[6,3,-2,1,-1,-0,5,-1,3,7,1,2,4]	[2,1,3]
11	2-3-3-1	Пороговая Сигмоидальная	[1,1]	[2,1,3,-0,3,-4,3]	[6,3,-2,1,-1,-0,5,-1,3,7,]	[2,1,3]
12	3-3-2-1	Сигмоидальная Радиальная базисная	[1,2,3]	[6,3,-2,1,-1,-0,5,-1,3,7,]	[2,1,3,-0,3,-4,3]	[-2,1]
13	2-3-2-1	Линейная Сигмоидальная	[4,6]	[2,1,3,-0,3,-4,3]	[2,1,3,-0,3,-4,3]	[-2,1]
14	2-2-3-1	Радиальная базисная Сигмоидальная	[-2,1]	[2,1,3,1]	[2,1,3,-0,3,-4,3]	[2,1,3]
15	3-2-4-1	Линейная с насыщением Сигмоидальная	[0,-1,5]	[2,1,3,-0,3,-4,3]	[6,3,-2,1,-1,-0,5,-1,3]	[2,1,3,1]
16	2-3-4-1	Полулинейная Сигмоидальная	[-4,2]	[1,2,1,3,-5,2]	[6,3,-2,1,-1,-0,5,-1,3,7,1,2,4]	[2,1,3,1]
17	3-4-2-1	Гиперболический тангенс Сигмоидальная	[3,-3,1]	[6,3,-2,1,-1,-0,5,-1,3,7,1,2,4]	[6,3,-2,1,-1,-0,5,-1,3,7,]	[-2,1]

№	Вид сети	Функции активации	Входное значение	Весовые коэф-ты 1 слоя	Весовые коэф-ты 2 слоя	Весовые коэф-ты 3 слоя
18	2-2-3-1	Полулинейная с насыщением Сигмоидальная	[1,-9]	[-4,2]	[1,2,1,3,-5,2]	[2,1,3]
19	3-3-4-1	Знаковая Сигмоидальная	[-3,-2,4]	[6,3,-2,1,-1,-0.5,-1,3,7]	[6,3,-2,1,-1,-0.5,-1,3,7,1,2,4]	[2,1,3,1]
20	2-4-3-1	Полулинейная с насыщением Сигмоидальная	[5,1]	[6,3,-2,1,-1,-0.5,-1,3]	[6,3,-2,1,-1,-0.5,-1,3,7,1,2,4]	[2,1,3]
21	2-3-3-1	Пороговая Сигмоидальная	[1,1]	[2,1,3,-0.3,-4,3]	[6,3,-2,1,-1,-0.5,-1,3,7]	[2,1,3]
22	3-3-2-1	Сигмоидальная Радиальная базисная	[1,2,3]	[6,3,-2,1,-1, -0.5,-1,3,7]	[2,1,3,-0.3,-4,3]	[-2,1]
23	2-3-2-1	Линейная Сигмоидальная	[4,6]	[2,1,3,-0.3,-4,3]	[2,1,3,-0.3,-4,3]	[-2,1]
24	2-2-3-1	Радиальная базисная Сигмоидальная	[-2,1]	[2,1,3,1]	[2,1,3,-0.3,-4,3]	[2,1,3]
25	3-2-4-1	Линейная с насыщением Сигмоидальная	[0,-1,5]	[2,1,3,-0.3,-4,3]	[6,3,-2,1,-1,-0.5,-1,3]	[2,1,3,1]

№	Вид сети	Функции активации	Входное значение	Весовые коэф-ты 1 слоя	Весовые коэф-ты 2 слоя	Весовые коэф-ты 3 слоя
26	2-3-4-1	Полулинейная Сигмоидальная	[-4,2]	[1,2,1,3,-5,2]	[6,3,-2,1,-1,-0,5, -1,3,7,1,2,4]	[2,1,3,1]
27	3-4-2-1	Гиперболический тангенс Сигмоидальная	[3,-3,1]	[6,3,-2,1,-1,-0,5, -1,3,7,1,2,4]	[6,3,-2,1,-1,-0,5,-1,3,7,]	[-2,1]
28	2-3-4-1	Полулинейная Сигмоидальная	[-4,2]	[1,2,1,3,-5,2]	[6,3,-2,1,-1,-0,5, -1,3,7,1,2,4]	[2,1,3,1]
29	3-4-2-1	Гиперболический тангенс Сигмоидальная	[3,-3,1]	[6,3,-2,1,-1,-0,5,- 1,3,7,1,2,4]	[6,3,-2,1,-1,-0,5,-1,3,7,]	[-2,1]
30	2-2-3-1	Полулинейная с насыщением Сигмоидальная	[1,-9]	[-4,2]	[1,2,1,3,-5,2]	[2,1,3]

Цель работы:

Изучение нейронных сетей для задач аппроксимации функций.

Общие сведения

Аппроксимация функций. Предположим, что имеется обучающая выборка $((x_1, y_2), (x_2, y_2), \dots, (x_n, y_n))$, которая генерируется неизвестной функцией, искаженной шумом. Задача аппроксимации состоит в нахождении оценки этой функции.

Порядок выполнения работы:

- Для заданного преподавателем варианта (таблица) аппроксимировать функцию.
- Разработать и реализовать алгоритм создания и моделирования нейронной сети.
- Определить параметры созданной нейронной сети (веса и смещение) и проверить правильность работы сети для последовательности входных векторов.
- Построить график, аналогичный представленному на рисунке 1, для своих исходных данных.

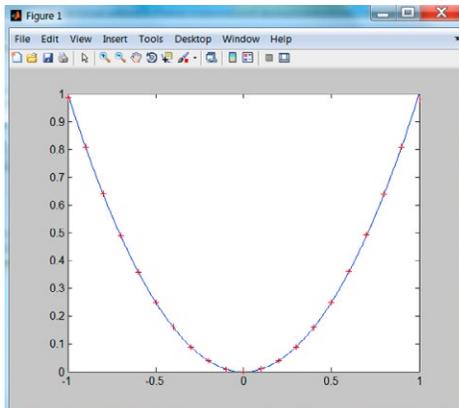


Рис. 1. Пример аппроксимации функций

Составить отчет

Вариант	Функция	Вариант	Функция
1	$Y = x \cdot x^3$	16	$Y = 0.1 \cdot x \cdot x^4 + 0.3 \cdot x$
2	$Y = 0.3 \cdot x \cdot x^3 + 0.4$	17	$Y = x \cdot x^3 + 0.2x$
3	$Y = 0.1 \cdot x \cdot x^3 + 0.2$	18	$Y = 0.3 \cdot x \cdot x^3 + x \cdot x^2 - 0.2 \cdot x$
4	$Y = 0.3 \cdot x \cdot x^3 + 0.3 \cdot x \cdot x^2$	19	$Y = \sin(2 \cdot \pi \cdot x)$
5	$Y = 0.4 \cdot x \cdot x^2 + 0.3$	20	$Y = \cos(2 \cdot \pi \cdot x) + x \cdot x^3 + 0.2 \cdot x$
6	$Y = 0.4 \cdot x \cdot x^2 + 0.4 \cdot x \cdot x^3$	21	$Y = \cos(2 \cdot \pi \cdot x) + x \cdot x^3 + 0.2$
7	$Y = 0.1 \cdot x \cdot x^2 + 0.5$	22	$Y = 0.4 \cdot x \cdot x^2 + \cos(3 \cdot \pi \cdot x)$
8	$Y = 0.4 \cdot x \cdot x^5 + 0.4 \cdot x$	23	$Y = 0.5 \cdot \pi \cdot x \cdot x^2 + 0.1$
9	$Y = 0.1 \cdot x \cdot x^3 + 0.2 \cdot x \cdot x^2 + \sin(2 \cdot \pi \cdot x)$	24	$Y = 0.4 \cdot x \cdot x^5 + 0.4 \cdot x - \sin(2 \cdot \pi \cdot x)$
10	$Y = 0.1 \cdot x \cdot x^3 - 0.2 \cdot x - \cos(0.2 \cdot \pi \cdot x)$	25	$Y = 0.1 \cdot x \cdot x^3 + 0.2 \cdot x \cdot x^2 - \sin(2 \cdot \pi \cdot x)$
11	$Y = 0.2 \cdot x \cdot x^3 - 0.1$	26	$Y = 0.1 \cdot x \cdot x^3 - 0.2 \cdot x + \sin(2 \cdot \pi \cdot x)$
12	$Y = 0.5 \cdot x \cdot x^3 - 0.2 \cdot x \cdot x^2$	27	$Y = \cos(2 \cdot \pi \cdot x) + 0.2 \cdot x \cdot x^3 - 0.1$
13	$Y = 0.4 \cdot x \cdot x^2 - 0.6$	28	$Y = \cos(2 \cdot \pi \cdot x) + 0.5 \cdot x \cdot x^3 - 0.2 \cdot x \cdot x^2$
14	$Y = 0.3 \cdot x \cdot x^2 - 0.4 \cdot x \cdot x^3$	29	$Y = \cos(0.4 \cdot \pi \cdot x) + 0.4 \cdot x \cdot x^2 - 0.6$
15	$Y = 0.2 \cdot x \cdot x^2 + 0.4$	30	$Y = 0.3 \cdot x \cdot x^2 - 0.4 \cdot x \cdot x^3$

Нейро-практикум № 5 Классификация Ирисов Фишера**Цель:**

На основании набора данных требуется построить правило классификации, определяющее вид Ириса.

Задачи:

- изучение данных;
- создание архитектуры нейронной сети;
- моделирование нейронной сети;
- обучение и тестирование нейронной сети;
- интерпретация результатов.

Описание данных:

Набор данных Ирисов Фишера является классическим примером в области статистики и машинного обучения, часто используемым для демонстрации алгоритмов классификации.

Фишер использовал этот набор данных для демонстрации работы метода дискриминантного анализа, который позволяет определить, к какому виду относится конкретный ирис на основе этих четырёх измерений. Метод дискриминантного анализа — это статистический метод, используемый для моделирования различий и классификации. Он строит комбинацию атрибутов (признаков), которые наилучшим образом разделяют категории, и использует эти комбинации для классификации новых наблюдений.

С момента своего создания набор данных Ирисов Фишера стал одним из наиболее известных примеров в мире машинного обучения, часто используемым как первый пример для обучения на курсах по машинному обучению и статистике, а также в качестве тестового «полигона» для алгоритмов классификации.

Этот набор данных содержит 150 примеров ирисов, разделённых на три вида: Ирис щетинистый (*Iris setosa*), Ирис разноцветный (*Iris versicolor*), и Ирис виргинский (*Iris virginica*). Каждый пример описывается четырьмя атрибутами: длиной и шириной чашелистиков, а также длиной и шириной лепестков.

1. Длина чашелистика (англ. *sepal length*)
2. Ширина чашелистика (англ. *sepal width*);
3. Длина лепестка (англ. *petal length*);
4. Ширина лепестка (англ. *petal width*).

На основании этого набора данных требуется построить правило классификации, определяющее вид растения по данным измерений. Это задача многоклассовой классификации, так как имеется три класса — три вида ириса.

Один из классов (*Iris setosa*) линейно-разделим от двух остальных.



Iris virginica



Iris setosa



Iris versicolor

Нейро-практикум № 6 Классификация зерен риса «Каммео и Османчик»**Цель:**

На основании набора данных требуется построить правило классификации, определяющее тип зерна риса «Каммео и Османчик».

Задачи:

- изучение данных;
- создание архитектуры нейронной сети;
- моделирование нейронной сети;
- обучение и тестирование нейронной сети;
- интерпретация результатов.

Описание данных:

Рис Осман (*Osmancik*).

Происхождение:

Турция. Используется в восточной кухне.

Характеристики:

Имеет короткую зерновую форму и хорошую способность впитывать в себя влагу, что делает его хорошим выбором для приготовления плова.

Описание данных:

Рис Каммолино (*Cammeo*).

Происхождение:

Италия. Используется в европейской кухне.

Характеристики:

Имеет короткую и толстую зерновую форму, богат крахмалом, что придаёт ему кремовую текстуру. Этот сорт часто используется для приготовления итальянских блюд, таких как ризотто.

Для классификации используемых сортов риса (*Cammeo* и *Osmancik*) была проведена предварительная обработка изображений, полученных с помощью системы компьютерного зрения,

в результате чего было получено 3810 рисовых зерен. Кроме того, для каждого зерна было выведено 7 морфологических признаков. Для полученных свойств был создан набор данных.

1. **Площадь:** количество пикселей рисового зерна.
2. **Периметр:** вычисляет окружность путем вычисления расстояния между пикселями по границам рисового зерна.
3. **Длина главной оси:** дает самую длинную линию, которую можно провести на рисовом зерне, т. е. расстояние до главной оси.
4. **Длина малой оси:** Самая короткая линия, которую можно провести на рисовом зерне, т. е. расстояние между малыми осями.
5. **Эксцентриситет:** измеряет, насколько круглым является эллипс.
6. **Выпуклая область:** измеряет количество пикселей наименьшей выпуклой оболочки области, образованной рисовым зерном.
7. **Extent:** Возвращает отношение пикселей области, образованной рисовым зерном, к пикселям ограничительного поля.

На основании этого набора данных требуется построить правило классификации, определяющее вид зерна по данным измерений.

Предположим, что Каммео — 0.1, Османчик — 0.2 для обучения нейронной сети.

База данных:

<https://archive.ics.uci.edu/datasets?skip=0&take=10&sort=desc&orderBy=NumHits&search=Rice+%28Cammeo+and+Osmancik%29>

Нейро-практикум № 7 **Определение возраста морского ушка**

Цель:

На основании набора данных требуется построить правило классификации, определяющее возраст морского ушка на основе физических измерений.

Задачи:

- изучение данных;
- создание архитектуры нейронной сети;
- моделирование нейронной сети;
- обучение и тестирование нейронной сети;
- интерпретация результатов.

Описание данных:

Набор данных состоит из 4177 наблюдений в возрастном диапазоне от 4 до 32 лет.

Входные параметры — 9 величин (данные о физических измерениях):

1. Sex (Пол — Мужской (M), Женский (F), и Инфантильный (I — это не взрослые).
2. Length (Длина — самое длинное измерение раковины, мм.)
3. Diameter (Диаметр — перпендикулярно к длине, мм).
4. Height (Высота — с мясом внутри, мм.).
5. Whole weight (Общий вес — вес целиком, грамм).
6. Shucked weight (Вес без панциря — вес мяса, грамм.).
7. Viscera weight (Вес внутренностей после вырезания крови, грамм.).
8. Shell weight (Вес раковины после высушивания, грамм.).
9. Rings (Количество колец) +1,5 дает возраст в годах.

Для определения возраста моллюска требуется добавить 1,5 к значениям величины «кольца».

Выходной параметр — возраст в диапазоне от 4 до 32 лет. Исходные данные записаны в файле.

Имеется 4177 примеров, которые разбиваются на две группы — первые N примеров используются для обучения многослойного персептрона, остальные для тестирования.

База данных:

<http://archive.ics.uci.edu/dataset/1/abalone>

Нейро-практикум № 8**Классификационный анализ диагностики
хронического заболевания почек****Цель:**

Провести классификационный анализ для диагностики хронического заболевания почек (ХЗП) у пациентов.

Задачи:

- изучение данных;
- создание архитектуры нейронной сети;
- моделирование нейронной сети;
- обучение и тестирование нейронной сети;
- интерпретация результатов.

Описание данных:

Выборка состоит из 300 наблюдений.

Входные переменные – 14 признаков:

1. age: Возраст.
2. bp: Артериальное давление.
3. sg: Плотность мочи.
4. al: Протеин в моче (альбумин).
5. su: Сахар в моче.
6. rsc: Патологические клетки в моче.
7. ba: Бактерии.
8. psv: Объем упакованных клеток.
9. htn: Гипертония.
10. dm: Сахарный диабет.
11. cad: Ишемическая болезнь сердца.
12. appet: Аппетит.
13. pe: Отеки.
14. ane: Анемия.

Выходная переменная:

- 1 — имеет заболевание.
- 2 — нет заболевания.

База данных:

<https://archive.ics.uci.edu/dataset/336/chronic+kidney+disease>

Нейро-практикум № 9**Диагностика сердечного приступа****Цель:**

Классификация данных с помощью нейронной сети.

Задачи:

- изучение данных;
- создание архитектуры нейронной сети;
- моделирование нейронной сети;
- обучение и тестирование нейронной сети;
- интерпретация результатов.

Описание работы:

База данных содержит результаты медицинских наблюдений за 303 людьми старше 29 лет.

Входные параметры включают:

1. age: Возраст пациента.
2. sex: Пол пациента.
3. exang: стенокардия, вызванная физической нагрузкой (1 = да; 0 = нет).
4. ca: количество крупных судов (0–3).
5. cp: Тип боли в груди.
 - a. Значение 1: типичная стенокардия.
 - b. Значение 2: атипичная стенокардия.
 - c. Значение 3: неангинальная боль.
 - d. Значение 4: бессимптомное течение.
6. trtbps: артериальное давление в состоянии покоя (в мм рт. ст.).
7. chol: холестерин в мг/дл, полученный с помощью датчика ИМТ.
8. fbs: (уровень сахара в крови натощак > 120 мг/дл) (1 = верно; 0 = неверно).

9. `rest_ecg`: результаты электрокардиографии в состоянии покоя.
 - а. Значение 0: нормальное.
 - б. Значение 1: наличие аномалии ST-T (инверсия зубца T и/или элевация или депрессия $ST > 0,05$ мВ).
 - с. Значение 2: показывает вероятную или определенную гипертрофию левого желудочка по критериям Эстеса.
10. `thalach`: достигается максимальная частота сердечных сокращений.
11. `target`: 0 = меньше шансов на сердечный приступ 1 = больше шансов на сердечный приступ.

Выходная переменная

Наличие сердечного приступа (0 — да, 1 — нет)

База данных:

<https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset/>

Нейро-практикум № 10**Выявления сердечных заболеваний****Цель:**

Классификация сердечных заболеваний с помощью нейронных сетей.

Задачи:

1. Изучение данных.
2. Создание архитектуры нейронной сети.
3. Моделирование нейронной сети.
4. Обучение и тестирование нейронной сети.
5. Интерпретация результатов.

Описание работы:

Данные содержат 918 наблюдений в каждой переменной. Всего переменных 12:

1. Age — возраст.
2. Gender — пол.
3. Chest_Pain_Type — это тип боли в груди, вызванный снижением притока крови к сердцу.
4. Resting_BP — давление в состоянии покоя.
5. Cholesterol — холестерин.
6. Fasting_BS — уровень сахара в крови натощак.
7. Resting_ECG — ЭКГ в состоянии покоя.
8. Max_HR — максимальная частота сердечных сокращений.
9. Exercise_Angina — стенокардия.
10. Old_peak — депрессия ST (сегмент), вызванная физической нагрузкой по сравнению с отдыхом.
11. ST_Slope.
12. Heart_Disease — болезнь сердца.

На основании этого набора данных требуется построить правило классификации, определяющее наличие у человека сердечного заболевания.

Выход должен быть равен:

- 0 — нет сердечных заболеваний
- 1 — есть сердечные заболевания

База данных:

<https://archive.ics.uci.edu/dataset/45/heart+disease>

Цель:

Классификация сортов вина с помощью нейронных сетей.

Задачи:

- изучение данных;
- создание архитектуры нейронной сети;
- моделирование нейронной сети;

- обучение и тестирование нейронной сети;
- интерпретация результатов.

описание работы:

Дается набор данных, который состоит из 178 экземпляров вин. Эти данные являются результатом химического анализа вин, выращенных в одном и том же регионе Италии, но полученных из трех разных сортов. В результате анализа определено количество 13 компонентов, обнаруженных в каждом из трех видов вин.

Определены 13 компонент, входящих в каждое из 3 видов вин:

1. Спирт.
2. Яблочная кислота.
3. Зола.
4. Щелочность золы.
5. Магний.
6. Общее количество фенолов.
7. Флавоноиды.
8. Нефлаваноидные фенолы.
9. Проантоцианы.
10. Интенсивность окраски.
11. Цветовой тон.
12. ОП 280/ОП 315 разбавленных вин.
13. Пролин.

На основании имеющегося набора данных требуется построить алгоритм классификации, определяющий сорт вина по исходной выборке.

Первая переменная является результирующей и обозначает сорт вина, которая принимает значения: 1 — первый сорт вина, 2 — второй сорт вина, 3 — 3 сорт вина.

База данных:

<https://www.kaggle.com/datasets/aarontanjaya/uci-wine-dataset>

Нейро-практикум № 12 Семена трех сортов пшеницы**Цель:**

Классификация семян трех сортов пшеницы с помощью нейронных сетей.

Задачи:

- изучение данных;
- создание архитектуры нейронной сети;
- моделирование нейронной сети;
- обучение и тестирование нейронной сети;
- интерпретация результатов.

Описание работы:

В этом исследовании использовались три вида семян пшеницы с учетом таких особенностей, как форма, тип и структура. Набор данных, состоит из 210 экземпляров.

Выходная переменная:

1 — первый сорт пшеницы; 2 — второй сорт пшеницы; 3 — третий сорт пшеницы.

Входные переменные — 7 компонентов:

1. Площадь A .
2. Периметр P .
3. Компактность.
4. Длина ядра.
5. Ширина ядра.
6. Коэффициент асимметрии.
7. Длина канавки ядра.

На основании этого набора данных требуется построить правило классификации, определяющее вид семечка по данным измерений. Это задача многоклассовой классификации, так как имеется 3 класса.

Предположим, что к первому классу относиться — 1, ко второму — 2 и к третьему — 3.

База данных:

<https://archive.ics.uci.edu/dataset/236/seeds>

Нейро-практикум № 13**Определение типа стекла****Цель:**

Классификация типа стекла с помощью нейронных сетей.

Задачи:

- изучение данных;
- создание архитектуры нейронной сети;
- моделирование нейронной сети;
- обучение и тестирование нейронной сети;
- интерпретация результатов.

Описание работы:

Изучение классификации типов стекла было мотивировано криминологическим расследованием. Оставленное на месте преступления стекло может быть использовано в качестве вещественного доказательства, если оно правильно идентифицировано.

На основании данных требуется обучить нейросеть, определяющую тип стекла на основе химических элементов, содержащихся в нем. 6 различных типа стекла:

- оконные стекла зданий, обработанные методом плавления;
- оконные стекла зданий, не обработанные методом плавления;
- автомобильные оконные стекла, обработанные методом плавления;
- стекла, используемые для контейнеров;
- стекло, используемое для посуды;
- стекло фар.

Данные взяты с сайта Kaggle.com и состоят из 169 наблюдений и 9 переменных: показатель преломления, Натрий, Магний, Алюминий, Кремний, Калий, Кальций, Барий, Железо; и типа стекла соответственно.

База данных:

<https://www.kaggle.com/datasets/uciml/glass>

Нейро-практикум № 14 **Классификация сушеных бобов****Цель:**

Классификация сушеных бобов с помощью нейронных сетей.

Задачи:

- изучение данных;
- создание архитектуры нейронной сети;
- моделирование нейронной сети;
- обучение и тестирование нейронной сети;
- интерпретация результатов.

Описание работы:

В этом исследовании использовались семь различных типов сушеных бобов с учетом таких особенностей, как форма, тип и структура. Для модели классификации с помощью камеры с высоким разрешением были сделаны изображения 13611 зерен 7 различных зарегистрированных сушеных бобов: *Seker*, *Barbunya*, *Bombay*, *Cali*, *Dermosan*, *Horoz* and *Sira*. Изображения бобов, полученные системой компьютерного зрения, были подвергнуты этапам сегментации и выделения признаков, всего 16 признаков (12 переменных по размерам бобов и 4 переменных по форме бобов):

1. **Площадь (A):** площадь зоны боба и количество пикселей в пределах ее границ.
2. **Периметр (P):** Окружность боба определяется как длина его границы.
3. **Длина большой оси (L):** расстояние между концами самой длинной линии, которую можно провести из боба.
4. **Длина малой оси (l):** самая длинная линия, которую можно провести от боба, стоя перпендикулярно главной оси.
5. **Соотношение сторон (K):** определяет соотношение между L и l.

6. **Эксцентриситет (Ес):** эксцентриситет эллипса, имеющий те же моменты, что и область.
7. **Выпуклая область (С):** количество пикселей в наименьшем выпуклом многоугольнике, который может содержать площадь семени фасоли.
8. **Эквивалентный диаметр (Ed):** Диаметр круга, площадь которого равна площади семени боба.
9. **Экстент (Ех):** Отношение пикселей в ограничивающей рамке к площади боба.
10. **Твердость (S):** Также известна как выпуклость. Отношение пикселей в выпуклой оболочке к тем, которые находятся в бобах.
11. **Округлость (R):** рассчитывается по следующей формуле: $(4\pi A)/(P^2)$.
12. **Компактность (CO):** Измеряет округлость объекта: Ed/L .
13. Коэффициент формы 1 (SF1).
14. Коэффициент формы 2 (SF2).
15. Коэффициент формы 3 (SF3).
16. Коэффициент формы 4 (SF4).

На основании этого набора данных требуется построить правило классификации, определяющее вид боба по данным измерений. Это задача многоклассовой классификации, так как имеется 7 классов — 7 видов бобов.

Предположим, что Seker — 0.1, Barbunya — 0.2, Bombay — 0.3, Cali — 0.4, Horoz — 0.5, Sira — 0.6, Dermosan — 0.7.

База данных:

<https://archive.ics.uci.edu/dataset/602/dry+bean+dataset>

Цель:

Классификация качества воды с помощью нейронных сетей.

Задачи:

- изучение данных;
- создание архитектуры нейронной сети;
- моделирование нейронной сети;
- обучение и тестирование нейронной сети;
- интерпретация результатов.

Описание работы:

На основании имеющегося набора данных требуется построить правило классификации, определяющее качество воды.

1. **Значение pH:** pH является важным параметром при оценке кислотно-щелочного баланса воды. Это также индикатор кислотного или щелочного состояния воды. ВОЗ рекомендует максимально допустимый предел pH от 6,5 до 8,5. Текущие диапазоны исследований составляли 6,52–6,83, что находится в диапазоне стандартов ВОЗ.
2. **Твердость (*Hardness*):** жесткость в основном обусловлена солями кальция и магния. Эти соли растворяются в геологических отложениях, через которые проходит вода. Продолжительность контакта воды с материалом, создающим жесткость, помогает определить степень жесткости сырой воды. Первоначально жесткость определяли как способность воды осаждать мыло, вызванное кальцием и магнием.
3. **Твердые вещества (Общее количество растворенных твердых веществ — TDS) (*Solids*):** вода обладает способностью растворять широкий спектр неорганических и некоторых органических минералов или солей, таких как калий, кальций, натрий, бикарбонаты, хлориды, магний, сульфаты и т. д. Эти минералы придают воде нежелательный вкус и обесцвеченный цвет. Это важный параметр для использования воды. Вода с высоким значением TDS указывает на высокую минерализацию воды. Желаемый предел для TDS составляет 500 мг/л, а максимальный предел составляет 1000 мг/л, что предписано для питьевых целей.

4. **Хлорамины (*Chloramines*):** хлор и хлорамин являются основными дезинфицирующими средствами, используемыми в системах общественного водоснабжения. Хлорамины чаще всего образуются при добавлении аммиака к хлору для обработки питьевой воды. Уровни хлора до 4 миллиграммов на литр (мг/л или 4 частей на миллион (ppm)) считаются безопасными для питьевой воды.
5. **Сульфат (*Sulfate*):** сульфаты — это природные вещества, которые содержатся в минералах, почве и горных породах. Они присутствуют в атмосферном воздухе, грунтовых водах, растениях и пищевых продуктах. Основное коммерческое использование сульфата находится в химической промышленности. Концентрация сульфатов в морской воде составляет около 2700 миллиграммов на литр (мг/л). В большинстве источников пресной воды он колеблется от 3 до 30 мг/л, хотя в некоторых географических точках встречаются гораздо более высокие концентрации (1000 мг/л).
6. **Проводимость (*Conductivity*):** чистая вода не является хорошим проводником электрического тока, а хорошим изолятором. Увеличение концентрации ионов увеличивает электропроводность воды. Как правило, количество растворенных в воде твердых веществ определяет электрическую проводимость. Электропроводность (ЕС) фактически измеряет ионный процесс раствора, который позволяет ему передавать ток. По стандартам ВОЗ значение ЕС не должно превышать 400 мкСм/см.
7. **Органический углерод (*Organic carbon*):** общий органический углерод (ТОС) в исходных водах поступает из разлагающихся природных органических веществ (НОМ), а также из синтетических источников. ТОС является мерой общего количества углерода в органических соединениях в чистой воде. По данным Агентства по охране окружающей среды США 2 мг/л общего органического углерода в очищенной/питьевой воде и 4 мг/л в исходной воде, используемой для очистки.

8. **Тригалометаны (*Trihalomethanes*):** ТГМ — это химические вещества, которые можно найти в воде, обработанной хлором. Концентрация ТГМ в питьевой воде варьируется в зависимости от уровня органических веществ в воде, количества хлора, необходимого для обработки воды, и температуры очищаемой воды. Уровни ТГМ до 80 частей на миллион считаются безопасными для питьевой воды.
9. **Мутность (*Turbidity*):** мутность воды зависит от количества твердого вещества, находящегося во взвешенном состоянии. Это мера светоизлучающих свойств воды, и тест используется для определения качества сброса сточных вод по отношению к коллоидным веществам. Среднее значение мутности, полученное для кампуса Wondo Genet (0,98 NTU), ниже рекомендуемого ВОЗ значения 5,00 NTU.
10. **Пригодность (*Potability*):** указывает, безопасна ли вода для потребления человеком, где 1 означает Пригодна для питья, а 0 означает Непригодна для питья.

База данных:

<https://www.kaggle.com/datasets/adityakadiwal/water-potability>

Нейро-практикум № 16**Диагностика заболеваний диабетом у женщин из индейских племен Pima****Цель:**

Диагностика заболеваний диабетом у женщин из индейских племен Pima с помощью нейронных сетей.

Задачи:

- изучение данных;
- создание архитектуры нейронной сети;
- моделирование нейронной сети;
- обучение и тестирование нейронной сети;
- интерпретация результатов.

Описание работы:

Требуется построить нейросетевую медико-диагностическую систему с минимальной ошибкой тестирования. Для обучения системы используются первые 450 примеров, остальные — для тестирования.

База данных содержит результаты медицинских наблюдений за 532 женщинами старше 21 года, кровей индейцев племени Pima, проживающих около Финикса (штат Аризона). Наблюдения проведены по критериям Всемирной Организации Здравья (World Health Organization) и выполнены американским Институтом диабета и заболеваний почечно-пищеварительного тракта (Smith et al, 1988).

Входные параметры включают:

1. *preg* — Число беременностей.
2. *glu* — Концентрация плазма-глюкозы в стоматическом тесте.
3. *bp* — Диастолическое давление крови (mm Hg).
4. *skin* — Толщина кожи на складке трехглавой мышцы (mm).
5. *ins* — Содержание сыворотки инсулина (микро-U/ml).
6. *bmi* — Индекс веса тела (вес в килограммах/(рост в метрах)²).
7. *ped* — Наличие диабета в роду.
8. *age* — Возраст (в годах).

Выходная переменная

Наличие диабета (0 — да, 1 — нет)

База данных:

(UCI machine-learning database collection (Murphy & Aha, 1995))
<https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database>

Нейро-практикум № 17 Классификация сортов семечек тыквы»**Цель:**

Классификация сортов семечек тыквы с помощью нейронных сетей.

Задачи:

- изучение данных;
- создание архитектуры нейронной сети;
- моделирование нейронной сети;
- обучение и тестирование нейронной сети;
- интерпретация результатов.

Описание работы:

В этом исследовании использовались два вида тыквенных семечек с учетом таких особенностей, как форма, тип и структура. Семена тыквы часто употребляют в качестве кондитерских изделий во всем мире из-за достаточного содержания в них белков, жиров, углеводов и минералов. Это исследование было проведено на двух наиболее важных и качественных видах семян тыквы: «Ургюп Сивриси» и «Черчевелик», обычно выращиваемых в регионах Ургюп и Караджаорен в Турции. Всего 12 признаков:

1. **Площадь (A):** площадь зоны семечка и количество пикселей в пределах ее границ.
2. **Периметр (P):** окружность семечка определяется как длина его границы.
3. **Длина большой оси (L):** расстояние между концами самой длинной линии, которую можно провести из боба.
4. **Длина малой оси (l):** самая длинная линия, которую можно провести от семечка, стоя перпендикулярно главной оси.
5. **Соотношение сторон (K):** определяет соотношение между L и l.
6. **Эксцентриситет (Ec):** эксцентриситет эллипса, имеющий те же моменты, что и область.

7. **Выпуклая область (C):** количество пикселей в наименьшем выпуклом многоугольнике, который может содержать площадь семечка.
8. **Эквивалентный диаметр (Ed):** Диаметр круга, площадь которого равна площади семечка.
9. **Экстент (Ex):** Отношение пикселей в ограничивающей рамке к площади семечка.
10. **Твердость (S):** Также известна как выпуклость. Отношение пикселей в выпуклой оболочке к тем, которые находятся в семечке.
11. **Округлость (R):** рассчитывается по следующей формуле: $(4\pi A)/(P^2)$.
12. **Компактность (CO):** Измеряет округлость объекта: Ed/L .

На основании этого набора данных требуется построить правило классификации, определяющее вид семечка по данным измерений. Это задача многоклассовой классификации, так как имеется 2 класса — 2 вида семечек тыквы.

Предположим, что Çerçevelik — 0.1, Ürgüp Sivrisi — 0.2 для обучения нейронной сети.

База данных:

<https://www.muratkoklu.com/datasets>

ПРИЛОЖЕНИЕ 2

Зачем изучать нейронные сети

Изучение нейронных сетей имеет большое значение как для научного сообщества, так и для широкого круга промышленных и коммерческих приложений.

Вот несколько ключевых причин, почему стоит изучать нейронные сети:

- 1. Искусственный интеллект и машинное обучение:** нейронные сети являются основой современного искусственного интеллекта (ИИ) и машинного обучения. Их понимание критически важно для разработки и улучшения алгоритмов ИИ.
- 2. Решение сложных проблем:** нейронные сети способны обрабатывать и анализировать большие объемы данных, находить в них закономерности и решать задачи, которые традиционными алгоритмическими методами решить сложно или невозможно.
- 3. Широкий спектр приложений:** нейросети находят применение во множестве областей — от распознавания образов и обработки естественного языка до автономных транспортных средств, медицины, финансов и многих других.
- 4. Инновации и научные исследования:** понимание нейронных сетей способствует инновациям и новым открытиям в науке и технологиях, открывая новые горизонты в исследованиях и разработках.
- 5. Карьерные возможности:** спрос на специалистов в области ИИ и машинного обучения постоянно растет. Знание нейронных сетей увеличивает карьерные перспективы в высокооплачиваемых и востребованных отраслях.

- 6. Понимание человеческого мозга:** изучение нейронных сетей также способствует лучшему пониманию работы человеческого мозга, так как многие концепции в нейросетях вдохновлены биологическими нейронными сетями.
- 7. Развитие критического мышления и проблемного анализа:** работа с нейросетями требует глубокого понимания и анализа данных, что способствует развитию навыков критического мышления и решения сложных задач.
- 8. Персонализация и адаптация:** нейросети позволяют создавать адаптивные и персонализированные системы, которые могут улучшать пользовательский опыт в различных приложениях, от рекомендательных систем до персонализированной медицины.

В целом, изучение нейронных сетей открывает двери в мир передовых технологий, способствует развитию научных исследований и создает основу для будущих технологических инноваций.

ПРИЛОЖЕНИЕ 3

Ключевые термины и определения

В области нейронных сетей существует ряд ключевых терминов и определений, которые необходимо понимать для глубокого осмысления предмета.

Вот некоторые из них:

Нейронная сеть: система, вдохновленная структурой и функциональностью биологического мозга, состоящая из взаимосвязанных узлов, известных как искусственные нейроны, которые обрабатывают информацию с использованием своей динамической состояний ответа на внешние входные данные.

Искусственный нейрон: основная вычислительная единица в нейронной сети, которая имитирует функции биологического нейрона. Она получает один или несколько входных сигналов и генерирует выходной сигнал.

Весы (*weights*): параметры в нейронной сети, которые регулируют силу связи между нейронами. Они обновляются в процессе обучения для минимизации ошибки между актуальным и предсказанным выходом сети.

Функция активации: математическая функция, применяемая к выходу нейрона, которая определяет, будет ли и как сильно активирован данный нейрон. Примеры включают ReLU, сигмоидную и гиперболический тангенс.

Обучение с учителем (*supervised learning*): метод обучения, при котором нейронная сеть обучается на наборе данных, содержащем входные данные и соответствующие им правильные ответы (метки).

Обучение без учителя (*unsupervised learning*): метод обучения, при котором нейронная сеть работает с входными данными без каких-либо явных ответов или меток, обучаясь находить закономерности и структуры в данных.

Обратное распространение (*backpropagation*): метод обучения, используемый в многослойных нейронных сетях, который включает в себя пропускание данных через сеть, вычисление потерь (разницы между предсказанными и фактическими результатами), и затем проход назад по сети для обновления весов для минимизации этих потерь.

Градиентный спуск: метод оптимизации, используемый для настройки параметров (весов) нейронной сети путем минимизации функции потерь.

Переобучение (*overfitting*): ситуация, когда нейронная сеть слишком точно обучается на тренировочных данных, включая шум и исключения, что снижает её способность к обобщению на новых данных.

Регуляризация: техника, используемая для предотвращения переобучения, путем добавления штрафа за сложность модели к функции потерь.

Эти термины и определения составляют основу для понимания и работы с нейронными сетями и их различными применениями.

Е. Ю. Савченко

**ОСНОВЫ И ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ:
ТЕОРИЯ, ПРАКТИКА, ИННОВАЦИИ**

Учебное пособие

Дизайн и компьютерная верстка: *В. Горнушкин*

Подписано к печати 16.10.2024.

Заказ № 64.

Формат бумаги $64 \times 80 \frac{1}{16}$. Объем 7,25 п. л.

Тираж 100 экз.

ОсОО «НЕО Принт»