

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет компьютерных технологий и прикладной математики
Центр искусственного интеллекта

УТВЕРЖДАЮ

Проректор по учебной работе,
качеству образования – первый
проректор

Хагуров Т.А.

подпись

«02» июня 2026 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.04 Математические модели нейронных сетей

Направление подготовки 02.03.03 Математическое обеспечение и
администрирование информационных систем

Направленность (профиль) Искусственный интеллект и аналитика данных

Форма обучения очная

Квалификация бакалавр

Краснодар 2026

Рабочая программа дисциплины «Математические модели нейронных сетей» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем.

Программу составил(и):

М.Х. Урtenов, профессор КПМ, д-р.физ.-мат.н.,
профессор



подпись

Рабочая программа дисциплины утверждена на заседании центра
искусственного интеллекта

протокол № 04 «14» мая 2026 г.

Руководитель центра ИИ Коваленко А.В.

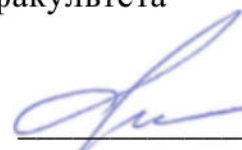


подпись

Утверждена на заседании учебно-методической комиссии факультета
компьютерных технологий и прикладной математики

протокол № 03 «15» мая 2026 г.

Председатель УМК факультета Добровольская Н.Ю.



подпись

Рецензенты:

Мостовой Евгений Викторович, генеральный директор ООО «Портал-Юг»,
e-mail: mostovoy@portal-yug.ru

Луценко Евгений Вениаминович, доктор экономических наук, кандидат технических наук, профессор кафедры компьютерных технологий и систем Федерального государственного бюджетное образовательное учреждение высшего образования «Кубанский государственный аграрный университет имени И.Т. Трубилина», e-mail: prof.lutsenko@gmail.com

1 Цели и задачи изучения дисциплины (модуля)

1.1 Цель освоения дисциплины

Формирование у студентов глубокого понимания принципов функционирования различных типов нейронных сетей, освоение базовых алгоритмов обучения и оптимизации нейронных сетей, а также развитие навыков анализа и интерпретации результатов моделирования нейронных сетей, необходимого для профессиональной деятельности аналитика данных, AI-инженера, MLOps-специалиста и менеджера ИИ-проектов.

1.2 Задачи дисциплины

1. Изучение основных понятий теории нейронных сетей: искусственные нейроны, топологии сетей, виды функций активации, оптимизаторы.
2. Анализ архитектур нейронных сетей (персептрон, MLP, самоорганизующиеся сети, сети Кохонена, сети Хопфилда, спайковые сети, радиально-базисные сети и т.д.).
3. Овладение методами обратного распространения ошибок и стохастического градиентного спуска.
4. Практическое освоение инструментов реализации несложных нейросетевых архитектур на языках программирования Python, MATLAB / C++ без использования фреймворков глубокого обучения.

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Математические модели нейронных сетей» относится к части, формируемой участниками образовательных отношений Блока 1 «Дисциплины (модули)» учебного плана. Данная дисциплина в качестве прerreквизитов требует освоения следующих дисциплин: «Математический анализ» (основы дифференциального исчисления, интегральное исчисление и ряды, теория функций нескольких переменных), «Введение в специальность» (поиск ошибок и рефакторинг кода через LLM), «Обработка данных на Python» (работа с данными для подготовки обучающей выборки), «Алгебра и геометрия» (операции с векторами и матрицами, системы линейных уравнений, проекции, ортогональность, преобразования координат), «Инструментальные средства разработки ИИ» (основы сред Matlab и Jupyter notebook), «Основы программирования», «Алгоритмы и структуры данных», «Объектно-ориентированное программирование» (освоение языков программирования C++, Python, освоение ООП). Постреквизитами выступают дисциплины, где продолжается изучение нейросетевых архитектур, принципов их обучения и их применения на практике: «Нейросетевые технологии» (применение на практике изученных принципов обучения, освоение новых архитектур и реализация их через фреймворки tensorflow и keras и библиотеку Spektral), «Глубокое обучение» (применение на практике изученных принципов обучения, освоение новых архитектур и техник глубокого обучения, реализация через pytorch, tensorflow, keras), «Управление жизненным циклом ИИ-продукта» (применение изученных архитектур), «Методы искусственного интеллекта в задачах классификации» (применение на практике изученных принципов обучения), «Технологии интеллектуальной поддержки принятия решений и управления» (внедрение нейросетей в процессы управления), «Искусственный интеллект в оценке рисков и разработке страховых продуктов» (внедрение нейросетей в процессы разработки страховых продуктов), «Методы обучения с подкреплением» (продолжение изучения и применение на практике обучения с обратной связью от человека (RLHF, RLAIIF), модели с наградой, PPO/DPO, смешанного обучения, PPO/DPO), «Генеративный искусственный интеллект» (применение на практике изученных принципов обучения, освоение новых архитектур для генерации, продолжение изучения и применение на практике дистилляции), «Технологии обработки языка, звуковых данных, включая распознавание и синтез речи» (применение на практике изученных принципов обучения, освоение новых архитектур для работы с текстами и аудио). Смежными дисциплинами, изучаемыми параллельно или логически близко, являются «Интеллектуальные методы оптимизации» (классические

методы оптимизации, условия экстремума функций нескольких переменных), «Курс теории вероятностей» (распределения, матожидание, дисперсия и др. важные понятия).

Связи разделов дисциплины с пререквизитами и постреквизитами

Связь	Дисциплина	Раздел(ы) текущей РПД, зависящие от данной дисциплины	Характер зависимости
Пререквизит	Математический анализ	Разделы 1-4	Сильный
Пререквизит	Введение в специальность	Разделы 2,4,5	Слабый
Пререквизит	Обработка данных на Python	Разделы 2, 4, 5	Средний
Пререквизит	Алгебра и геометрия	Разделы 1-4	Сильный
Пререквизит	Инструментальные средства разработки ИИ	Разделы 2,4,5	Сильный
Пререквизит	Основы программирования	Разделы 2,4,5	Сильный
Пререквизит	Алгоритмы и структуры данных	Разделы 2,4,5	Средний
Пререквизит	Объектно-ориентированное программирование	Разделы 2,4,5	Сильный
Постреквизит	Нейросетевые технологии	Разделы 1-5	Критический
Постреквизит	Глубокое обучение	Раздел 1-5	Критический
Постреквизит	Управление жизненным циклом ИИ-продукта	Разделы 2,4,5	Сильный
Постреквизит	Методы искусственного интеллекта в задачах классификации	Разделы 2-5	Средний
Постреквизит	Технологии интеллектуальной поддержки принятия решений и управления	Разделы 3-5	Слабый
Постреквизит	Искусственный интеллект в оценке рисков и разработке страховых продуктов	Разделы 2-5	Средний
Постреквизит	Методы обучения с подкреплением	Раздел 3	Средний
Постреквизит	Генеративный искусственный интеллект	Раздел 3-5	Сильный
Постреквизит	Технологии обработки языка,	Разделы 3-5	Сильный

	звуковых данных, включая распознавание и синтез речи		
--	--	--	--

1.4 Профессиональные роли в структуре образовательной программы

Роль 1: Data Analyst (Аналитик данных)

Задачи:

1. Статистический анализ, визуализация данных, предварительная обработка.
2. Создание прогнозных моделей
3. Построение аналитических моделей для поддержки бизнес-решений.

Роль 2: MLOps (Специалист по эксплуатации ИИ)

Задачи:

1. DevOps для ML.
2. Автоматизация, мониторинг ML-систем.
3. Операционное управление жизненным циклом ML-моделей.

Роль 3: AI PM (Менеджер проектов ИИ)

Задачи:

1. Управление ИИ-проектами от идеи до внедрения
2. Анализ бизнес-требований и постановка задач
3. Оценка эффективности и ROI ИИ-решений

1.5 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций:

Код, уровень и формулировка компетенции	Индикаторы	Уровни освоения индикаторов компетенции
DL-1 Способен применять и (или) разрабатывать архитектуры глубоких нейронных сетей	DL-1.1 Способен объяснять и применять математические основы нейронных сетей, включая расчет градиентов, методы оптимизации и алгоритм обратного распространения ошибки (backpropagation), для эффективного обучения моделей	(Б) Выбирает функцию активации и степень связности нейронов в зависимости от задачи; применяет обратное распространение ошибки для обновления весов нейронов; выбирает способ формирования начальных значений весов нейронов
	DL-1.2 Способен проектировать и реализовывать неглубокие нейронные сети (перцептроны, MLP), выбирать appropriate функции активации и регуляризации для решения задач классификации и регрессии	(Б) Способен реализовывать многослойный перцептрон

LLM-2 Способен дообучать, адаптировать и оптимизировать генеративные модели под специфические задачи и условия применения	LLM-2.4 Понимает обучение с обратной связью	(Б) Оценивает разницу между классическим обучением с учителем и RLHF
	LLM-2.5 Применяет дистилляцию моделей	(Б) Понимает идею уменьшения модели через дистилляцию
FC-1 Способен проводить фронтальные исследования в области архитектур, алгоритмов МО, оптимизации и математики	FC-1.1 Разрабатывает фундаментальные основы и новые алгоритмы машинного обучения	(Б) Знает основной математический аппарат для теоретического обоснования свойств моделей глубокого обучения. Использует способы эффективного обучения при заданных условиях для часто встречающихся задач.

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 3 зач. ед. (108 часов), их распределение по видам работ представлено в таблице

Вид учебной работы	Всего часов	Семестры (часы)					
		4					
Контактная работа, в том числе:	52.3	52.3					
Аудиторные занятия (всего):	48	48					
Занятия лекционного типа	32	32					
Лабораторные занятия	16	16					
Занятия семинарского типа (семинары, практические занятия)							
Иная контактная работа:	4.3	4.3					
Контроль самостоятельной работы (КСР)	4	4					
Промежуточная аттестация (ИКР)	0.3	0.3					
Самостоятельная работа, в том числе:	20	20					
Курсовая работа	-	-					
Проработка учебного (теоретического) материала	14	14					
Выполнение индивидуальных заданий (подготовка сообщений, презентаций)	-	-					
Реферат	-	-					
Подготовка к текущему контролю	6	6					
Контроль:	35,7	35,7					
Подготовка к экзамену	35,7	35,7					
Общая трудоемкость	час.	108	108				

	в том числе контактная работа	52.3	52.3					
	зач. Ед	3	3					

2.2 Структура дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

Разделы (темы) дисциплины, изучаемые в 4 семестре

№	Наименование разделов (тем)	Количество часов				
		Все го	Аудиторная работа			Внеа удито рная работ а
			Л	ПЗ	ЛР	СРС
1	2	3	4	5	6	7
1.	Введение в нейронные сети	10	8			2
2.	Архитектуры нейронных сетей	24	8		10	6
3.	Алгоритмы и методы обучения нейронных сетей	14	8			6
4.	Оптимизаторы	10	4		4	2
5.	Интерпретация результатов обучения	10	4		2	4
ИТОГО по разделам дисциплины		68	32		16	20
Контроль самостоятельной работы (КСР)		4				
Промежуточная аттестация (ИКР)		0.3				
Подготовка к экзамену		35,7				
Общая трудоемкость по дисциплине		108				

Примечание: Л – лекции, ПЗ – практические занятия/семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

2.3 Содержание разделов (тем) дисциплины

2.3.1 Занятия лекционного типа

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущ его контро ля
1	2	3	4
1.	Введение в нейронные сети	Введение в нейронные сети. Параллели из биологии. История развития искусственных нейронных сетей. Краткий обзор ключевых достижений. Базовая искусственная модель. Применение нейронных сетей. Теоремы Колмогорова, Арнольда и Хехт-Нильсена. Основные концепции нейронных сетей. Нейрокомпьютеры. Основные типы активационных функций (ступенчатая, сигмоидная, ReLU, EeLU,	Т

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		LeakyReLU, гиперболический тангенс, softmax и др.). Понятие архитектуры нейронной сети и её влияние на производительность модели. Программное обеспечение для НС.	
2.	Архитектуры нейронных сетей	Однослойные и многослойные перцептроны (MLP). Вероятностные нейросети (PNN). Байесовский нейронные сети. Машина Больцмана. Введение в радиально-базисные нейронные сети (RBF), самоорганизующиеся карты Кохонена (SOM) и сети Хопфилда. Спайковые нейронные сети (SNN).	Т
3.	Алгоритмы и методы обучения нейронных сетей	Постановка задачи обучения нейронной сети: обучение с учителем и без учителя. Классический метод градиентного спуска (Gradient Descent). Обзор алгоритма обратного распространения ошибки (backpropagation). Метод наименьших квадратов и регуляризация. Использование стохастического градиентного спуска (SGD) и его модификации (mini-batch SGD, batch gradient descent). Методы инициализации весов нейронных сетей (случайными числами, нулями, Xavier initialization, Kaiming, He-et-al). Техника предварительной подготовки данных (preprocessing data techniques). Обучение с обратной связью от человека (RLHF, RLAIIF), модели с наградой, PPO/DPO, смешанное обучение (многозадачное дообучение). Дистилляция (учитель-ученик, мягкие цели), PPO/DPO, модели с наградой, регуляризация, смешанное обучение, оптимизация сигналов вознаграждения и штрафов.	Т
4.	Оптимизаторы	Метод Левенберга–Марквардта.	Т

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		Адаптивные оптимизирующие алгоритмы (SGDM, Adam, RMSProp, AdaGrad). Гибридные методы оптимизации и их преимущества. Оценка эффективности оптимизаций и выбор подходящего метода. Построение вычислительных графов и автоматическое дифференцирование. Шаговые скорости обучения (learning rate scheduling). Мета-эвристические методы оптимизации.	
5.	Интерпретация результатов обучения	Выбор и модификация моделей. Моделирование на основе статистического анализа. Формулировка нестандартных гипотез и статистических критериев на основе приближенных генеративных методов. Определение переобучения (overfitting) и недостаточного обучения (underfitting). Понимание процесса переноса знания между слоями нейронной сети. Статистическая значимость результатов и кросс-валидация. Визуализация распределения классов и кластеров в пространстве признаков. Ошибки и аномалии при обучении нейронных сетей (примеры случаев неверного поведения модели). Применение LIME/SHAP для интерпретации нейросетей	Т

Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.2 Занятия семинарского типа

Не предусмотрены

Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.3 Лабораторные занятия

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
2.	Архитектуры нейронных сетей	Лабораторная работа №1. Однослойный и многослойный персептрон. Функции активации. Лабораторная работа №2. Вероятностные и радиально-базисные нейронные сети. Лабораторная работа №3. Самоорганизующиеся карты Кохонена. Лабораторная работа №4. Сеть Хопфилда. Лабораторная работа №5. Спайковые нейронные сети.	ЛР
4.	Оптимизаторы	Лабораторная работа №6. Оптимизационные алгоритмы в нейронных сетях	ЛР
5.	Интерпретация результатов обучения	Лабораторная работа №7. Диагностика и устранение проблем переобучения и недостаточного обучения нейронных сетей. Интерпретация результатов обучения	ЛР

Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.4 Примерная тематика курсовых работ (проектов)

Не предусмотрены

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1	Изучение теоретического материала	Методические указания по организации самостоятельной работы студентов, утвержденные УСФ, протокол №1 от 30.06.2025
2	Решение задач	Методические указания по организации самостоятельной работы студентов, утвержденные УСФ, протокол №1 от 30.06.2025

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа,
- в форме аудиофайла,

– в печатной форме на языке Брайля.

Для лиц с нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа,

– в форме аудиофайла.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. Образовательные технологии

В соответствии с требованиями ФГОС в программа дисциплины предусматривает использование в учебном процессе следующих образовательные технологии: чтение лекций с использованием мультимедийных технологий; метод малых групп, разбор практических задач и кейсов.

При обучении используются следующие образовательные технологии:

- Технология коммуникативного обучения – направлена на формирование коммуникативной компетентности студентов, которая является базовой, необходимой для адаптации к современным условиям межкультурной коммуникации.

- Технология разноуровневого (дифференцированного) обучения – предполагает осуществление познавательной деятельности студентов с учётом их индивидуальных способностей, возможностей и интересов, поощряя их реализовывать свой творческий потенциал. Создание и использование диагностических тестов является неотъемлемой частью данной технологии.

- Технология модульного обучения – предусматривает деление содержания дисциплины на достаточно автономные разделы (модули), интегрированные в общий курс.

- Информационно-коммуникационные технологии (ИКТ) - расширяют рамки образовательного процесса, повышая его практическую направленность, способствуют интенсификации самостоятельной работы учащихся и повышению познавательной активности. В рамках ИКТ выделяются 2 вида технологий:

- Технология использования компьютерных программ – позволяет эффективно дополнить процесс обучения языку на всех уровнях.

- Интернет-технологии – предоставляют широкие возможности для поиска информации, разработки научных проектов, ведения научных исследований.

- Технология индивидуализации обучения – помогает реализовывать личностно-ориентированный подход, учитывая индивидуальные особенности и потребности учащихся.

- Проектная технология – ориентирована на моделирование социального взаимодействия учащихся с целью решения задачи, которая определяется в рамках профессиональной подготовки, выделяя ту или иную предметную область.

- Технология обучения в сотрудничестве – реализует идею взаимного обучения, осуществляя как индивидуальную, так и коллективную ответственность за решение учебных задач.

- Игровая технология – позволяет развивать навыки рассмотрения ряда возможных способов решения проблем, активизируя мышление студентов и раскрывая личностный потенциал каждого учащегося.

- Технология развития критического мышления – способствует формированию разносторонней личности, способной критически относиться к информации, умению отбирать информацию для решения поставленной задачи.

Комплексное использование в учебном процессе всех вышеперечисленных технологий стимулируют личностную, интеллектуальную активность, развивают познавательные процессы, способствуют формированию компетенций, которыми должен обладать будущий специалист.

Основные виды интерактивных образовательных технологий включают в себя:

- работа в малых группах (команде) - совместная деятельность студентов в группе под руководством лидера, направленная на решение общей задачи путём творческого сложения результатов индивидуальной работы членов команды с делением полномочий и ответственности;

- проектная технология - индивидуальная или коллективная деятельность по отбору, распределению и систематизации материала по определенной теме, в результате которой составляется проект;

- анализ конкретных ситуаций - анализ реальных проблемных ситуаций, имевших место в соответствующей области профессиональной деятельности, и поиск вариантов лучших решений;

- развитие критического мышления – образовательная деятельность, направленная на развитие у студентов разумного, рефлексивного мышления, способного выдвинуть новые идеи и увидеть новые возможности.

Подход разбора конкретных задач и ситуаций широко используется как преподавателем, так и студентами во время лекций, лабораторных занятий и анализа результатов самостоятельной работы. Это обусловлено тем, что при исследовании и решении каждой конкретной задачи имеется, как правило, несколько методов, а это требует разбора и оценки целой совокупности конкретных ситуаций.

Семестр	Вид занятия	Используемые интерактивные образовательные технологии	количество интерактивных часов
4	ЛР	Практические занятия в режимах взаимодействия «преподаватель – студент» и «студент – студент»	8
Итого			8

Примечание: Л – лекции, ПЗ – практические занятия/семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

Темы, задания и вопросы для самостоятельной работы призваны сформировать навыки поиска информации, умения самостоятельно расширять и углублять знания, полученные в ходе лекционных и практических занятий.

Подход разбора конкретных ситуаций широко используется как преподавателем, так и студентами при проведении анализа результатов самостоятельной работы.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

4. Оценочные и методические материалы

4.1 Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «Математические модели нейронных сетей».

Оценочные средства включает контрольные материалы для проведения **текущего контроля** в форме тестовых заданий, лабораторных работ и **промежуточной аттестации** в форме вопросов и заданий к **экзамену**.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

Структура оценочных средств для текущей и промежуточной аттестации

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
1	Введение в нейронные сети	FC-1.1 (базовый); DL-1.1 (базовый);	Тестирование	Вопросы к экзамену 1-7

		DL-1.2 (базовый)		
2	Архитектуры нейронных сетей	FC-1.1 (базовый); DL-1.1 (базовый); DL-1.2 (базовый)	Тестирование, Лабораторные работы №1-5	Вопросы к экзамену 8-13
3	Алгоритмы и методы обучения нейронных сетей	FC-1.1 (базовый); DL-1.1 (базовый); DL-1.2 (базовый); LLM-2.4 (базовый); LLM-2.5 (базовый)	Тестирование	Вопросы к экзамену 14-20, 33, 34
4	Оптимизаторы	FC-1.1 (базовый)	Тестирование, Лабораторная работа №6	Вопросы к экзамену 21-26
5	Интерпретация результатов обучения	FC-1.1 (базовый); DL-1.1 (базовый); DL-1.2 (базовый)	Тестирование, Лабораторная работа №7	Вопросы к экзамену 27-32

Показатели, критерии и шкала оценки сформированных компетенций

Соответствие **пороговому уровню** освоения компетенций планируемым результатам обучения и критериям их оценивания (оценка: **удовлетворительно**):

DL-1 *Способен применять и (или) разрабатывать архитектуры глубоких нейронных сетей*

Выбирает функцию активации и степень связности нейронов в зависимости от задачи; применяет обратное распространение ошибки для обновления весов нейронов; выбирает способ формирования начальных значений весов нейронов

Способен реализовывать многослойный персептрон

LLM-2 *Способен дообучать, адаптировать и оптимизировать генеративные модели под специфические задачи и условия применения*

Оценивает разницу между классическим обучением с учителем и RLHF

Понимает идею уменьшения модели через дистилляцию

FC-1 *Способен проводить фронтальные исследования в области архитектур, алгоритмов МО, оптимизации и математики*

Знает основной математический аппарат для теоретического обоснования свойств моделей глубокого обучения. Использует способы эффективного обучения при заданных условиях для часто встречающихся задач.

Соответствие **базовому уровню** освоения компетенций планируемым результатам обучения и критериям их оценивания (оценка: **хорошо**):

DL-1 *Способен применять и (или) разрабатывать архитектуры глубоких нейронных сетей*

Задаёт скорость обучения в зависимости от задачи и набора данных; выбирает функцию потерь в зависимости от задачи и набора данных;

- способен применять регуляризацию и прореживание; выбирает размер пакета для стохастического градиентного спуска; понимает принцип градиентного спуска
- Способен разрабатывать и/или применять самоорганизующиеся карты Кохонена; Способен разрабатывать RBF-сети (сети регуляризации, обобщенные RBF-сети)
- LLM-2** ***Способен дообучать, адаптировать и оптимизировать генеративные модели под специфические задачи и условия применения***
 Настраивает пайплайны RLHF с использованием человекоориентированных меток
 Применяет дистилляцию для адаптации модели под периферийные устройства
- FC-1** ***Способен проводить фронтальные исследования в области архитектур, алгоритмов МО, оптимизации и математики***
 Знает продвинутый математический аппарат приводящий к более разностороннему пониманию процессов обучения. Использует математический аппарат для эффективного обучения моделей, для создания моделей меньшего размера. Создаёт гибридные модели.

Соответствие **продвинутому уровню** освоения компетенций планируемым результатам обучения и критериям их оценивания (оценка: **отлично**):

- DL-1** ***Способен применять и (или) разрабатывать архитектуры глубоких нейронных сетей***
 Выбирает функцию активации и степень связности нейронов в зависимости от задачи; применяет обратное распространение ошибки для обновления весов нейронов; выбирает способ формирования начальных значений весов нейронов
 Способен реализовывать многослойный персептрон
 Задаёт скорость обучения в зависимости от задачи и набора данных; выбирает функцию потерь в зависимости от задачи и набора данных; способен применять регуляризацию и прореживание; выбирает размер пакета для стохастического градиентного спуска; понимает принцип градиентного спуска
 Способен разрабатывать и/или применять самоорганизующиеся карты Кохонена; Способен разрабатывать RBF-сети (сети регуляризации, обобщенные RBF-сети)
 применяет для обучения нейронных сетей методы оптимизации второго порядка (Левенберга-Марквардта);
 Владеет способами борьбы с переключением в сетях SOM; Способен разрабатывать ограниченные машины Больцмана и глубокие сети доверия
- LLM-2** ***Способен дообучать, адаптировать и оптимизировать генеративные модели под специфические задачи и условия применения***
 Оценивает разницу между классическим обучением с учителем и RLHF
 Понимает идею уменьшения модели через дистилляцию
 Настраивает пайплайны RLHF с использованием человекоориентированных меток
 Применяет дистилляцию для адаптации модели под периферийные устройства
- FC-1** ***Способен проводить фронтальные исследования в области архитектур, алгоритмов МО, оптимизации и математики***

Знает основной математический аппарат для теоретического обоснования свойств моделей глубокого обучения. Использует способы эффективного обучения при заданных условиях для часто встречающихся задач. Знает продвинутый математический аппарат приводящий к более разностороннему пониманию процессов обучения. Использует математический аппарат для эффективного обучения моделей, для создания моделей меньшего размера. Создаёт гибридные модели.

Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Пример тестирования

1. Что такое искусственная нейронная сеть (ИНС)?

- A) Программа, выполняющая логические операции
- B) Система, копирующая мозг человека
- C) Математическая модель, вдохновленная устройством нервной системы живых организмов
- D) Алгоритм шифрования данных

2. Кто считается автором первого математического описания нейрона?

- A) Дональд Хебб
- B) Уоррен Маккаллоу и Уолтер Питтс
- C) Йозеф Шеррингтон
- D) Герман фон Гельмгольц

3. Назовите ключевое событие в развитии нейронных сетей конца XX века:

- A) Появление компьютера ENIAC
- B) Разработка первой рабочей модели нейрона
- C) Открытие правила Хебба
- D) Появление алгоритма обратного распространения ошибки (backpropagation)

4. Что означает аббревиатура MLP?

- A) Многослойный персептрон
- B) Минимальная линейная программа
- C) Матрица логических операций
- D) Максимально лёгкая парадигма

5. Что обозначает функция активации ReLU?

- A) Линейная единица прямоугольного наклона
- B) Логарифмическая экспоненциальная функция
- C) Синусоидальная активация
- D) Гиперболический тангенс

6. Какая функция активации возвращает диапазон значений от 0 до 1?

- A) Hyperbolic Tangent (TanH)
- B) Rectified Linear Unit (ReLU)
- C) Exponential Linear Unit (ELU)
- D) Sigmoid

7. Какой метод называется градиентным спуском?

- A) Постепенное обновление весов нейронной сети путём уменьшения ошибки
- B) Случайный перебор вариантов весов
- C) Копирование архитектуры биологического нейрона
- D) Распределённая система параллельных вычислений

8. Что такое backpropagation?

- A) Метод обратного распространения ошибки для обучения нейронных сетей
- B) Метод борьбы с переобучением

C) Метод аугментации данных

D) Стандарт защиты данных

9. Как называется процедура коррекции весов с учётом среднего квадрата ошибки?

A) Stochastic Gradient Descent (SGD)

B) Mean Squared Error Minimization (MSE)

C) Batch Normalization

D) Regularization

10. В каком типе нейронной сети используется концепция самоорганизации?

A) Convolutional Neural Networks (CNN)

B) Recurrent Neural Networks (RNN)

C) Self-Organizing Map (SOM)

D) Long Short Term Memory (LSTM)

11. Что характерно для сети Хопфилда?

A) Возможность реконструкции зашумленных образов

B) Применение исключительно в задачах классификации

C) Высокоскоростной режим обучения

D) Обязательное использование GPU

12. Как называются нейронные сети, работающие с импульсами?

A) Spiking Neural Networks (SNN)

B) Binary Neurons

C) Decision Trees

D) Support Vector Machines (SVM)

13. Как называют ситуацию, когда модель слишком сильно подгоняется под обучающие данные?

A) Underfitting

B) Overfitting

C) Dropout

D) Bagging

14. Какая проблема возникает, если модель недостаточно сложна для задачи?

A) Vanishing Gradients

B) Exploding Gradients

C) Underfitting

D) Overfitting

15. Какая методика помогает уменьшить риск переобучения?

A) Cross-Validation

B) Grid Search

C) Forward Propagation

D) Weight Initialization

16. Какая библиотека часто используется для визуализации графов нейронных сетей?

A) Matplotlib

B) TensorBoard

C) Seaborn

D) Plotly

17. Какой алгоритм используется для инициализации весов нейронной сети, разработанный Глоротом и Бенджио?

A) Random Weights

B) Zeros Initialization

C) Xavier/Glorot Initialization

D) Ones Initialization

18. Что означает сокращение SGD?

A) Simple Gradient Decrease

B) Stochastic Gradient Descent

- C) Symmetric Graph Distribution
- D) Systematic Global Design

19. Что такое кластеризация в контексте нейронных сетей?

- A) Метод группировки схожих данных
- B) Вид атаки на безопасность сети
- C) Режим быстрой загрузки
- D) Тип архитектуры нейронной сети

20. Какое свойство характеризует качественное распределение классов в пространстве признаков?

- A) Равномерное распределение по оси координат
- B) Четкое разделение кластеров
- C) Отсутствие корреляции между переменными
- D) Высокий показатель коллинеарности

21. Что значит "Transfer Learning"?

- A) Перемещение данных между серверами
- B) Повторное использование заранее натренированной модели
- C) Трансляция новостей в социальных сетях
- D) Технология резервного копирования

Вопрос 22. Что такое "Learning Rate Schedule"?

- A) Постоянная величина шага обучения
- B) Периодичность сбора статистики
- C) Варианты изменения скорости обучения в течение цикла обучения
- D) Автоматическое переключение режимов работы сети

23. Какой оптимизатор сочетает в себе моменты и коррекцию ускорения?

- A) RMSProp
- B) Adam
- C) AdaGrad
- D) SGD

24. Какая операция в процессе обучения нейронных сетей снижает влияние масштаба входных данных?

- A) Normalization
- B) Regularization
- C) Pruning
- D) Quantization

25. Какая функция активации не страдает проблемой насыщения?

- A) Sigmoid
- B) TanH
- C) ReLU
- D) Softmax

26. Что такое регуляризация в контексте нейронных сетей?

- A) Процедура увеличения сложности модели
- B) Метод контроля за сложностью модели для предотвращения переобучения
- C) Алгоритм для начальной инициализации весов
- D) Техника быстрого обучения сети

27. Что показывает статистика кросс-валидации?

- A) Надежность и воспроизводимость модели
- B) Количество классов в наборе данных
- C) Степень свободы модели
- D) Размер тренировочного набора

28. Что подразумевается под понятием "statistical significance" в контексте нейронных сетей?

- A) Число скрытых слоев

- В) Важность результата обучения с точки зрения статистики
- С) Масштаб активаций
- Д) Величина скачка весов

Время выполнения: 30-40 минут.

Критерии оценки:

- оценку «отлично» получает студент, правильно ответивший на 82 и более процентов вопросов.
- оценку «хорошо» получает студент, правильно ответивший на 71-81 процентов вопросов.
- оценку «удовлетворительно» получает студент, правильно ответивший на 60-70 процентов вопросов.
- оценку «неудовлетворительно» получает студент, правильно ответивший менее, чем на 60 процентов вопросов.

Перечень компетенций (части компетенции), проверяемых оценочным средством:
FC-1.1; DL-1.1; DL-1.2, LLM-2.4, LLM-2.5.

Практические кейсы по тематике лабораторных работ

1. Однослойный и многослойный персептрон. Функции активации.

Кейс: Автоматизация сортировки писем по рубрикам.

Постановка задачи: Ваш отдел получил задачу по созданию автоматической системы рассылки писем клиентам. Каждый клиент получает письма разного содержания (реклама, уведомления, техническая поддержка). Вам нужно разработать нейронную сеть, которая будет классифицировать входящие письма по трем категориям ("реклама", "уведомления", "техническая поддержка").

Требования:

- Создать и обучить однослойный персептрон с функцией активации ReLU.
- Далее модернизировать модель, превратив её в многослойный персептрон с двумя скрытыми слоями и сигмоидой на выходе.
- Сделать вывод о том, какая модель показала лучшие результаты и почему.

2. Вероятностные и радиально-базисные нейронные сети.

Кейс: Сортировка продуктов по вкусовым предпочтениям покупателей.

Постановка задачи: Вам дан массив данных о продуктах питания и рейтингах, выставленных покупателями. Нужно обучить вероятностную нейронную сеть (PNN) и радиально-базисную сеть (RBF), чтобы классифицировать продукты на категории: "полезные", "средние", "вредные".

Требования:

- Реализовать и обучить PNN и RBF.
- Сравнить результаты работы обеих моделей.
- Определить, какая модель эффективнее справляется с задачей классификации и почему.

3. Самоорганизующиеся карты Кохонена.

Кейс: Анализ демографической структуры населения города.

Постановка задачи: Имеется массив данных о жителях города, содержащих возраст, доход, образование и предпочтения в отдыхе. Необходимо использовать самоорганизующиеся карты Кохонена (SOM) для визуализации и сегментирования жителей по поведенческим признакам.

Требования:

- Обучить SOM для выделения сегментов населения.
- Интерпретировать полученный результат, выделив характерные черты каждого сегмента.
- Представить итоговый результат в виде тепловой карты или кластерного дерева.

4. Сеть Хопфилда.

Кейс: Восстановление искривленных логотипов компании.

Постановка задачи: Некоторые логотипы компании стали деформироваться вследствие некачественного сканирования или плохого хранения. Необходимо восстановить оригинальные формы логотипов, используя сеть Хопфилда.

Требования:

- Построить и обучить сеть Хопфилда для восстановления оригиналов логотипов.
- Оценить эффективность восстановления с помощью показателя точности совпадения восстановленных изображений с оригинальными.

5. Спайковые нейронные сети.

Кейс: Распознавание аварийных сигналов автомобиля.

Постановка задачи: Автомобиль оснащен системой безопасности, выдающей световые сигналы тревоги (красный, желтый, зеленый). Однако некоторые датчики начали сбоить, выдавая ложные сигналы. Нужно разработать спайковую нейронную сеть (SNN), которая бы корректировала сигнал датчика.

Требования:

- Реализовать SNN для классификации сигналов.
- Измерить процент правильных распознаваний и частоту ложных срабатываний.

6. Оптимизационные алгоритмы в нейронных сетях.

Кейс: Обучение нейронной сети для игры в шахматы.

Постановка задачи: Требуется создать нейронную сеть, играющую в шахматы на уровне новичка. Но ваша первая модель обучалась долго и неэффективно. Теперь предстоит оптимизировать процесс обучения с помощью адаптивных алгоритмов (Adam, RMSProp, AdaGrad).

Требования:

- Провести обучение сети с каждым из трёх алгоритмов оптимизации.
- Сравнить время обучения и качество результатов.
- Выбрать самый эффективный алгоритм для дальнейшего использования.

7. Диагностика и устранение проблем переобучения и недостаточного обучения нейронных сетей.

Кейс: Система рекомендаций фильмов перестала выдавать точные советы.

Постановка задачи: Система рекомендаций начала демонстрировать низкое качество работы: одни фильмы предлагаются практически всем пользователям, другие совсем игнорируются. Вероятно, модель либо переобучилась, либо недостаточна.

Требования:

- Определить, присутствует ли переобучение или недостаточная мощность модели.
- Исправить проблему, скорректировав архитектуру или введя регуляризацию.
- Проверить улучшение качества рекомендаций на тестовом наборе данных.
- Объяснить решение нейросети через LIME/SHAP

Пример лабораторной работы

Лабораторная работа №1. Однослойный и многослойный персептрон. Функции активации

Цель: познакомиться с концепциями однослойных и многослойных персептронов, запрограммировать алгоритм обратного распространения ошибки, изучить влияние различных функций активации на результат обучения и сравнить оба типа моделей без использования специальных библиотек и фреймворков вроде TensorFlow, Keras, PyTorch и scikit-learn.

Задачи:

- Разработать нейронные сети в виде однослойного и двухслойного персептрона;
- Изучить алгоритм обратного распространения ошибки с последующей его реализацией;
- Запрограммировать различные функции активации;
- визуализировать результаты обучения;
- проанализировать разработанные модели, сделать выводы о их качестве и предложить улучшения.

Ожидаемые результаты:

- Умеет программировать алгоритм обратного распространения ошибки;
- Разрабатывает однослойные и многослойные персептроны;
- Знает наиболее популярные функции активации и умеет их правильно применять на практике;
- Умеет интерпретировать результаты обучения нейронной сети.

Инструменты и библиотеки:

- Язык программирования Python
- Jupyter Notebook / Google Colab
- Библиотеки numpy, matplotlib/seaborn.

Исходные данные: для наглядности воспользуемся синтетическим датасетом XOR (исключающее ИЛИ), который требует наличия не-линейной границы разделения.

Ход работы:

1. Установка библиотек

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

2. Генерация данных XOR

```
np.random.seed(42)
```

```
X = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])
```

```
Y = np.array([0, 1, 1, 0])[:, None]
```

3. Функция для визуализации данных

```
def visualize_data():
```

```
    colors = ['blue' if i == 0 else 'green' for i in Y.ravel()]
```

```

plt.scatter(X[:, 0], X[:, 1], c=colors)
plt.title('XOR Data')
plt.show()
visualize_data()
4. Реализация однослойного персептрона с сигмоидной функцией активации
class OneLayerPerceptron:
    def __init__(self, input_dim, output_dim):
    ...
    def predict(self, x):
    ...
    @staticmethod
    def sigmoid(x):
    ...
def train(self, X, Y, lr=0.1, epochs=1000):
    ...
5. Обучение однослойного персептрона
perceptron = OneLayerPerceptron(input_dim=X.shape[1], output_dim=Y.shape[1])
mse_history_one_layer = perceptron.train(X, Y)
6. Построение прогнозов с помощью однослойного персептрона
predictions = perceptron.predict(X)
print("Predictions (One-Layer): ", predictions.round())
7. Реализация многослойного персептрона с функциями ReLU и sigmoid
class TwoLayerPerceptron:
    def __init__(self, input_dim, hidden_dim, output_dim):
    ...
    def forward(self, x):
    ...
    @staticmethod
    def relu(x):
    ...
    @staticmethod
    def sigmoid(x):
    ...
    def backward(self, X, Y, o, lr=0.1):
    ...
    def train(self, X, Y, lr=0.1, epochs=1000):
    ...
8. Обучение многослойного персептрона
ml_perceptron = TwoLayerPerceptron(input_dim=X.shape[1], hidden_dim=4,
output_dim=Y.shape[1])
mse_history_two_layers = ml_perceptron.train(X, Y)
9. Построение прогнозов с помощью многослойного персептрона
..
10. Сравнение результатов
plt.figure(figsize=(10, 5))
plt.plot(mse_history_one_layer, label="One-Layer Perceptron")
plt.plot(mse_history_two_layers, label="Two-Layer Perceptron")
plt.title("MSE Loss Over Time")
plt.xlabel("Epochs")
plt.ylabel("Mean Square Error")
plt.legend()
plt.show()

```

Требования к отчету

1. Титульный лист

Название работы, ФИО студента, группа, дата.

2. Введение

Цель и задачи работы.

Описание датасета (объем, типы ошибок).

3. Математическая модель нейросети

Описание алгоритма обучения.

Описание архитектуры.

Описание оптимизатора и функций активации.

4. Реализация

Код на Python с комментариями.

Примеры до/после обработки.

5. Результаты

Визуализация кривых обучения.

6. Выводы

Проблемы, возникшие при обработке.

Предложения по улучшению.

7. Приложения

Исходный код.

Словарь сленга.

Критерии оценки:

Отлично: Полное выполнение всех шагов, анализ полученных результатов с построением функции потерь.

Хорошо: Корректный код, но без анализа полученных результатов.

Удовлетворительно: Частичное выполнение с ошибками на этапах визуализации модели и сравнения результатов.

Неудовлетворительно: Невыполнение ключевых этапов.

Лабораторная работа №2. Вероятностные и радиально-базисные нейронные сети

Цель: познакомиться с концепциями вероятностных нейронных сетей и радиально-базисных нейронных сетей и сравнить эти сети на одном датасете.

Задачи:

- Реализовать и обучить вероятностную нейронную сеть (PNN).
- Реализовать и обучить радиально-базисную нейронную сеть (RBF).
- Сравнить результаты работы обеих моделей.
- Определить, какая модель эффективнее справляется с задачей классификации и почему.

Ожидаемые результаты:

- Понимает архитектуру вероятностной нейронной сети
- Понимает архитектуру радиально-базисной нейронной сети;
- Знает наиболее популярные функции активации и умеет их правильно применять на практике;

- Умеет интерпретировать результаты обучения нейронных сетей классификации и сравнивать модели между собой.

Инструменты и библиотеки:

- Язык программирования Python / Matlab
- Jupyter Notebook / Google Colab / Matlab
- Библиотеки numpy, matplotlib/seaborn, pandas, scikit-learn, если выбрана реализация на python / nntool при выборе среды Matlab

Исходные данные: для наглядности воспользуемся синтетическим датасетом Ирисы Фишера для задачи классификации.

Ход выполнения работы на Matlab:

1. Подготовка данных и среды

Сначала необходимо загрузить данные и разделить их на обучающую и тестовую выборки.

% Очистка рабочей области

```
clear; clc; close all;
```

% Загрузка встроенного набора данных Fisher's Iris

```
load fisheriris
```

% Используем только первые два признака для 2D-визуализации

```
X = meas(:,1:2); % Входные данные (длина и ширина чашелистника)
```

```
y = species; % Целевые значения (классы)
```

% Преобразуем строковые метки в категориальный тип (для удобства)

```
y = categorical(y);
```

% Разделение данных на обучающую (70%) и тестовую (30%) выборки

```
cv = cvpartition(height(y), 'HoldOut', 0.3);
```

```
Xtrain = X(training(cv), :);
```

```
ytrain = y(training(cv));
```

```
Xtest = X(test(cv), :);
```

```
ytest = y(test(cv));
```

2. Реализация и обучение PNN (Вероятностная нейронная сеть)

В MATLAB для этого используется функция `newpnn`. Эта сеть относится к сетям без обучения в классическом смысле: она просто запоминает все обучающие примеры.

% Создание и обучение PNN

```
net_pnn = newpnn(Xtrain', dummyvar(categories(ytrain)));
```

% Прогнозирование на тестовых данных

```
...
```

% Определение предсказанных классов (находим индекс максимального выхода)

```
[~, predicted_labels_pnn] = max(outputs_pnn);
```

```
predicted_labels_pnn = categories(ytrain)(predicted_labels_pnn);
```

% Вычисление точности (accuracy)

```
...
```

3. Реализация и обучение RBF (Радиально-базисная сеть)

Для RBF в MATLAB используется функция `newrb`. Она также является сетью с "нулевой ошибкой" на обучающей выборке, но работает иначе, чем PNN.

% Создание и обучение RBF

```
net_rbf = newrb(Xtrain', dummyvar(categories(ytrain))');
```

% Прогнозирование на тестовых данных

...

% Определение предсказанных классов

```
[~, predicted_labels_rbf] = max(outputs_rbf);
```

```
predicted_labels_rbf = categories(ytrain)(predicted_labels_rbf);
```

% Вычисление точности (accuracy)

...

Примечание: `dummyvar` создает матрицу векторов-целей, где каждый столбец соответствует классу.

4. Визуализация результатов

Чтобы понять, как работают модели, построим график, где цветом обозначены истинные классы, а маркерами — предсказанные.

```
figure('Name', 'Сравнение моделей', 'Units', 'normalized', 'Position', [0.2 0.2 0.6 0.6]);
```

```
 tiledlayout(1, 3, 'Padding', 'compact', 'TileSpacing', 'compact');
```

% 1. Исходные данные (Обучающая выборка)

...

% 2. Результаты PNN на тесте

```
nexttile;
```

```
gscatter(Xtest(:,1), Xtest(:,2), predicted_labels_pnn, 'rgb', 'x');
```

```
title(['PNN | Точность: ', num2str(accuracy_pnn*100, '%.1f'), '%']);
```

```
xlabel('Длина чашелистника'); ylabel('Ширина чашелистника');
```

```
legend('setosa', 'versicolor', 'virginica', 'Location','northwest');
```

```
hold on;
```

```
gscatter(Xtest(:,1), Xtest(:,2), ytest, 'k', '.', 8); % Истинные метки черным цветом
```

% 3. Результаты RBF на тесте

```
nexttile;
```

```
gscatter(Xtest(:,1), Xtest(:,2), predicted_labels_rbf, 'rgb', '+');
```

```
title(['RBF | Точность: ', num2str(accuracy_rbf*100, '%.1f'), '%']);
```

```
xlabel('Длина чашелистника'); ylabel('Ширина чашелистника');
```

```
legend('setosa', 'versicolor', 'virginica', 'Location','northwest');
```

```
hold on;
```

```
gscatter(Xtest(:,1), Xtest(:,2), ytest, 'k', '.', 8); % Истинные метки черным цветом
```

```
sgtitle('Сравнение PNN и RBF на наборе данных Iris');
```

Ход выполнения работы на python с использованием библиотеки `scikit-learn`:

Прим.: В scikit-learn нет прямого класса PNN или RBF для классификации, но их функциональность можно реализовать с помощью готовых инструментов:

1. Вероятностная нейронная сеть (PNN) реализуется через класс KernelDensity (ядерная оценка плотности). Мы будем оценивать плотность вероятности для каждого класса в точках тестовой выборки и выбирать класс с максимальной плотностью.
2. Радиально-базисная сеть (RBF) реализуется через стандартный пайплайн: StandardScaler (для нормализации данных) + SGDClassifier с ядром 'rbf'.

1. Подготовка данных и окружения

Мы будем использовать тот же набор данных Iris. Для работы понадобятся библиотеки numpy, matplotlib и scikit-learn.

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.neighbors import KernelDensity
from sklearn.linear_model import SGDClassifier
from sklearn.pipeline import make_pipeline
from sklearn.metrics import accuracy_score
```

Загрузка данных

```
iris = load_iris()
X = iris.data[:, :2] # Используем только первые два признака для 2D-визуализации
y = iris.target
```

Разделение на обучающую и тестовую выборки

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42, stratify=y)
```

2. Реализация PNN (Вероятностная нейронная сеть)

Идея метода: для каждого класса мы строим оценку плотности распределения его признаков. Для нового объекта мы вычисляем плотность вероятности для каждого класса и относим объект к тому классу, где плотность оказалась выше.

Кодирование метки классов (хотя для KDE это не обязательно, но удобно)

```
le = LabelEncoder()
y_train_enc = le.fit_transform(y_train)
classes = np.unique(y_train_enc)
```

Создаем словарь для хранения KDE-моделей для каждого класса

```
kde_models = {}
log_densities = {}
```

Обучаем модель (на самом деле, просто "запоминаем" данные для оценки плотности)

```
for c in classes:
```

```
    # Выбираем данные только текущего класса
    X_class = X_train[y_train_enc == c]
```

```
    # Создаем и подгоняем модель KDE.
```

```
    # bandwidth (ширина окна) подобрана эмпирически для лучшей визуализации.
```

```
    # metric='euclidean' - стандартная евклидова метрика.
```

```

kde = KernelDensity(bandwidth=0.3, metric='euclidean', kernel='gaussian')
...

# Предсказание на тестовой выборке
...

# Класс с максимальной плотностью вероятности - наш ответ
y_pred_pnn = np.argmax(log_dens_test, axis=1)

# Вычисление точности
...

```

3. Реализация RBF (Радиально-базисная сеть)

Здесь мы используем стандартный подход scikit-learn. Пайплайн сначала нормализует данные (это критически важно для RBF-ядер), а затем обучает линейный классификатор (SGDClassifier) в пространстве признаков, преобразованных RBF-ядром.

```

# Создаем пайплайн: Масштабирование + Классификатор с RBF ядром
rbf_pipeline = make_pipeline(
    StandardScaler(), # Нормализация данных (очень важно!)
    SGDClassifier(loss='hinge', alpha=0.001, max_iter=1000, tol=1e-3, random_state=42)
)

```

```

# Обучение модели
...

```

```

# Предсказание на тестовой выборке
...

```

```

# Вычисление точности
...

```

Примечание: SGDClassifier с потерей hinge обучает SVM (метод опорных векторов), который по своей сути является линейным классификатором в пространстве признаков, заданном ядром. Это стандартный способ реализации нелинейных разделяющих границ в sklearn.

4. Визуализация результатов

Теперь построим графики, чтобы увидеть, как модели разделили пространство признаков.

```

# Создаем сетку точек для визуализации границ принятия решения
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, 0.02),
                     np.arange(y_min, y_max, 0.02))
grid_points = np.c_[xx.ravel(), yy.ravel()]

```

```

# Функция для построения графика модели
def plot_decision_boundary(X_train, y_train, X_test, y_test, y_pred, title, accuracy):
    plt.figure(figsize=(6, 5))
    # Цвета для классов
    cmap_light = plt.cm.Pastel1
    cmap_bold = plt.cm.Dark2

```

```

# Предсказание на сетке для отрисовки границ
if title == 'PNN':
    log_dens_grid = np.zeros((len(grid_points), len(classes)))
    for idx, c in enumerate(classes):
        log_dens_grid[:, idx] = kde_models[c].score_samples(grid_points)
    Z = np.argmax(log_dens_grid, axis=1)
else:
    Z = rbf_pipeline.predict(grid_points)

Z = Z.reshape(xx.shape)
plt.contourf(xx, yy, Z, cmap=cmap_light, alpha=0.8)

# Наносим на график обучающие данные (крестики) и тестовые (точки)
...

plt.xlim(xx.min(), xx.max())
plt.ylim(yy.min(), yy.max())
plt.title(f'{title} | Accuracy: {accuracy:.2%}')
plt.xlabel(iris.feature_names[0])
plt.ylabel(iris.feature_names[1])
plt.legend()
plt.show()

print("\nВизуализация PNN:")
plot_decision_boundary(X_train, y_train, X_test, y_test, y_pred_pnn, 'PNN', accuracy_pnn)

print("\nВизуализация RBF:")
plot_decision_boundary(X_train, y_train, X_test, y_test, y_pred_rbf, 'RBF', accuracy_rbf)

```

Требования к отчету

1. Титульный лист

Название работы, ФИО студента, группа, дата.

2. Введение

Цель и задачи работы.

Описание датасета (объем, типы ошибок).

3. Математическая модель нейросети

Описание алгоритма обучения.

Описание архитектуры.

Описание оптимизаторов и функций активации.

4. Реализация

Код на Python с комментариями или Matlab.

Примеры до/после обработки.

5. Результаты

Визуализация кривых обучения.

Сравнение результатов обучения для разных архитектур.

6. Выводы

Проблемы, возникшие при обработке.

Предложения по улучшению.

7. Приложения

Исходный код.

Словарь сленга.

Критерии оценки:

Отлично: Полное выполнение всех шагов, анализ полученных результатов с построением функции потерь.

Хорошо: Корректный код, но без анализа полученных результатов.

Удовлетворительно: Частичное выполнение с ошибками на этапах визуализации модели и сравнения результатов.

Неудовлетворительно: Невыполнение ключевых этапов.

Проверяемые компетенции комплексом практических заданий: FC-1.1; DL-1.1; DL-1.2

Зачетно-экзаменационные материалы для промежуточной аттестации (экзамен)

Вопросы для подготовки к экзамену

1. Строение биологического нейрона. Что такое искусственная нейронная сеть (ИНС)?
2. Назовите ключевые этапы истории развития искусственных нейронных сетей.
3. Приведите примеры важнейших открытий и изобретений в области нейронных сетей.
4. В чём состоят теоремы Колмогорова-Арнольда-Хехт-Нильсена и как они связаны с функционированием нейронных сетей?
5. Какие существуют основные типы активационных функций? Приведите примеры (ступенчатая, сигмоида, ReLU, ELU, LeakyReLU, tanh, SoftMax).
6. Что такое архитектура нейронной сети? Как она влияет на производительность модели?
7. Приведите примеры программного обеспечения для работы с нейронными сетями.
8. Объясните разницу между однослойными и многослойными персептронами (MLP).
9. Что такое радиально-базисные нейронные сети (RBF)? Где они применяются?
10. Охарактеризуйте вероятностные нейронные сети (PNN) и приведите примеры их использования.
11. Опишите особенности самоорганизующихся карт Кохонена (SOM). Для каких задач они чаще всего применяются?
12. В чём суть сети Хопфилда? Чем она полезна?
13. Чем принципиально отличаются спайковые нейронные сети (SNN)?
14. В чём разница между обучением с учителем и без учителя?
15. Опишите классический метод градиентного спуска (Gradient Descent) и его модификации.
16. Объясните алгоритм обратного распространения ошибки (Backpropagation).
17. Что такое регуляризация и почему она необходима при обучении нейронных сетей?
18. Что такое стохастический градиентный спуск (SGD)? Чем различаются mini-batch SGD и full-batch GD?

19. Опишите технику Xavier initialization и He initialization. Зачем нужны методы инициализации весов?
20. Что такое техника предварительной подготовки данных (preprocessing)? Приведите примеры методов нормализации.
21. В чём сущность метода Левенберга–Марквардта? Когда его целесообразно использовать?
22. Какие адаптивные оптимизирующие алгоритмы существуют (SGDM, Adam, RMSProp, AdaGrad)? Как они улучшают процесс обучения?
23. Мета-эвристические методы оптимизации.
24. Как оценивают эффективность выбранных методов оптимизации и выбирают подходящий оптимизатор?
25. Что такое вычислительный граф и зачем нужно автоматическое дифференцирование?
26. Как влияют настройка learning rate schedule (шагов скоростей обучения) на успех обучения?
27. Как диагностируются ситуации переобучения (overfitting) и недостаточного обучения (underfitting)? Приведите примеры мер борьбы с этими проблемами.
28. Объясните процесс переноса знаний между слоями нейронной сети. Приведите конкретные примеры.
29. Какие инструменты и методы позволяют визуально представлять классы и кластеры в пространстве признаков?
30. Что означает термин «статистическая значимость» в контексте обучения нейронных сетей? Какие методы оценки статистической значимости существуют?
31. В чём смысл кросс-валидации (cross-validation)? Приведите примеры схем кросс-валидации.
32. Приведите примеры ошибочного поведения нейронных сетей (ошибочные предсказания, нестабильность и др.). Как предотвратить такие ситуации?
33. Обучение с обратной связью от человека (RLHF, RLAIIF), модели с наградой, PPO/DPO, смешанное обучение (многозадачное дообучение).
34. Дистилляция (учитель-ученик, мягкие цели), PPO/DPO, модели с наградой, регуляризация, смешанное обучение.

Перечень компетенций (части компетенции), проверяемых оценочным средством
FC-1.1; DL-1.1; DL-1.2, LLM-2.4, LLM-2.5.

Практические задания к экзамену

Задание №1: Создание и обучение нейронной сети с учителем

Задание: Напишите программу на Python или C++/Matlab, реализовав однослойный перцептрон для классификации датасета Iris с учителем.

Задание №2. Создание и обучение нейронной сети без учителя

Напишите программу на Python или C++/Matlab для кластеризации датасета Iris в виде сети без учителя (например, Self-organizing map).

Задание №3: Применение различных активационных функций

Задание: Возьмите одну из нейросетей, разработанных в рамках лабораторных работ, реализуйте с различными функциями активации (ступенчатая, сигмоидная, гиперболический тангенс, softmax) и обучите сеть. Подготовьте отчёт с результатами и сравнительным анализом.

Задание №4: Применение различных активационных функций

Задание: Возьмите одну из нейросетей, разработанных в рамках лабораторных работ, реализуйте с различными функциями активации (ReLU, LeakyReLU, eeLU) и обучите сеть. Подготовьте отчет с результатами и сравнительным анализом.

Задание №5: Переобучение и недостаточное обучение

Задание: Возьмите небольшой датасет (например, Boston Housing dataset) и спроектируйте простую нейронную сеть. Добейтесь эффекта переобучения и недостаточного обучения, варьируя количество эпох, размеры сети и степень регуляризации. Докажите ваши наблюдения соответствующими графиками (loss curve, training vs test error).

Задание №6: Визуализация кластеров в пространстве признаков

Задание: Используйте датасет Wine Quality и обучите нейронную сеть для классификации вин. После обучения визуализируйте распределение классов и выделенные кластеры в пространстве признаков (используя PCA или t-SNE).

Задание №7: Кросс-валидация и статистическая значимость

Задание: Примените k-fold кросс-валидацию на датасете MNIST. Сравнивая средние показатели точности и стандартного отклонения, докажите статистическую значимость выбранного вами метода обучения.

Задание №8: Сравнение оптимизаторов SGDM и RMSProp

Задание: Обучите простую нейронную сеть (например, на датасете Fashion-MNIST) с разными оптимизаторами (SGDM, RMSProp). Проведите анализ их производительности, отметив скорость обучения и достигаемые результаты. Объяснить решение нейросети через LIME/SHAP.

Задание №9: Сравнение оптимизаторов Adam и AdaGrad

Задание: Обучите простую нейронную сеть (например, на датасете Fashion-MNIST) с разными оптимизаторами (Adam, AdaGrad). Проведите анализ их производительности, отметив скорость обучения и достигаемые результаты.

Перечень компетенций (части компетенции), проверяемых оценочным средством
FC-1.1; DL-1.1; DL-1.2

4.2 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Методические рекомендации, определяющие процедуры оценивания на экзамене:

Процедура промежуточной аттестации проходит в соответствии с Положением о текущем контроле и промежуточной аттестации обучающихся ФГБОУ ВО «КубГУ».

Итоговой формой контроля сформированности компетенций у обучающихся по дисциплине является экзамен. Студенты обязаны сдать экзамен в соответствии с расписанием и учебным планом.

ФОС промежуточной аттестации состоит из вопросов к экзамену и результатов текущего контроля.

Экзамен по дисциплине преследует цель оценить работу студента за курс, получение теоретических знаний, их прочность, развитие творческого мышления, приобретение

навыков самостоятельной работы, умение применять полученные знания для решения практических задач.

Форма проведения экзамена: устно.

Экзаменатору предоставляется право задавать студентам дополнительные вопросы по всей учебной программе дисциплины.

Результат сдачи экзамена заноситься преподавателем в экзаменационную ведомость и зачетную книжку.

Оценивание уровня освоения дисциплины основывается на качестве выполнения студентом заданий текущего контроля (выполнения лабораторных работ, тестирования и проектного задания) и ответов на вопросы экзамена.

Критерии оценки:

1. Оценка ответов на экзаменационные вопросы (40% итоговой оценки)

Отлично (5)

Полные, развернутые ответы с демонстрацией глубокого понимания темы.

Использование примеров, формул, корректных терминов.

Умение анализировать и сравнивать (например, функции активации или оптимизаторы).

% выполнения: 90–100% (допускаются незначительные неточности).

Хорошо (4)

Ответы содержат основные идеи, но без углубленного анализа.

Возможны небольшие ошибки в деталях или формулировках.

% выполнения: 75–89%.

Удовлетворительно (3)

Ответы поверхностные, с существенными пробелами.

Отсутствие примеров или некорректное применение терминов.

% выполнения: 60–74%.

Неудовлетворительно (2)

Отсутствие понимания ключевых концепций.

Грубые ошибки или неспособность ответить на большую часть вопросов.

% выполнения: <60%.

2. Оценка выполнения проектной работы (практических кейсов) (15% итоговой оценки)

Отлично (5)

Полное выполнение всех этапов кейса с инновационными решениями.

Достижение целевых показателей обучения.

Четкая документация кода и качественный доклад (полнота описания хода работы, анализ ошибок, визуализация).

% выполнения: 90–100%.

Хорошо (4)

Выполнены основные задачи, но без дополнительной оптимизации.

Незначительные отклонения от целевых показателей обучения (например, ошибка больше желаемой на 3-5%) или низкое качество доклада (не полностью описан ход работы, нет визуализаций)

% выполнения: 75–89%.

Удовлетворительно (3)

Решены базовые задачи, но с критическими ошибками.

Низкое качество кода или доклада (не полностью описан ход работы, нет визуализаций)

% выполнения: 60–74%.

Неудовлетворительно (2)

Невыполнение ключевых этапов.

Код нерабочий или отсутствует или не выполнен доклад о результатах выполнения проектной работы
% выполнения: <60%.

3. Оценка тестовых вопросов (20% итоговой оценки)

Отлично (5)

24–28 правильных ответов (82–100%).

Демонстрация уверенного владения терминологией и методами.

Хорошо (4)

20–23 правильных ответов (71–81%).

Незначительные ошибки в сложных вопросах.

Удовлетворительно (3)

16–19 правильных ответов (60–70%).

Путаница в базовых концепциях (например, путает назначение архитектур или не понимает разницу в оптимизаторах).

Неудовлетворительно (2)

Менее 16 правильных ответов (<60%).

Неспособность отличить архитектуры (например, многослойный персептрон от спайковой сети).

4. Оценка лабораторных работ (25% итоговой оценки)

Отлично (5)

Средний балл за лабораторные работы не ниже 4,5

% выполнения: 91–100%.

Хорошо (4)

Средний балл за лабораторные работы не ниже 3,8

% выполнения: 76–90%.

Удовлетворительно (3)

Средний балл за лабораторные работы не ниже 3

% выполнения: 60–75%.

Неудовлетворительно (2)

Средний балл за лабораторные работы ниже 3

% выполнения: <60%.

Методические рекомендации, определяющие процедуры оценивания лабораторных работ:

Процедура оценивания лабораторных работ проходит в соответствии с Положением о текущем контроле и промежуточной аттестации обучающихся ФГБОУ ВО «КубГУ».

По каждой лабораторной работе оформляется отчет. Отчеты сдаются на проверку руководителю в течение курса по мере их выполнения, и защищаются студентами в установленном порядке.

При защите отчета студенту могут быть заданы вопросы и дополнительные задания по сути лабораторной работы, в том числе из списка контрольных вопросов к данной лабораторной работе. При неудовлетворительной оценке знаний студента по теме данного отчета, студент возвращается к повторному изучению соответствующих материалов, после чего допускается к повторной защите. Неудовлетворительно выполненный отчет также возвращается на доработку.

Отчет должен содержать заголовок, тему лабораторной работы, цель, задание, индивидуальную тему, описание хода выполнения работы, необходимые прикладные материалы (схемы, макеты документов и т.п.), в соответствии с требованиями к содержанию, и выводы по работе.

Итоговая оценка (суммарно)

Оценка	Экзамен (40%)	Проект (15%)	Тест (20%)	ЛР (25%)	Общий %
Отлично (5)	90–100%	90–100%	82–100%	90-100%	≥85%
Хорошо (4)	75–89%	75–89%	71–81%	76-89%	72–85%
Удовлетворительно (3)	60–74%	60–74%	60–70%	60-75%	60–71%
Неудовлетворительно (2)	<60%	<60%	<60%	<60%	<60%

Для допуска к экзамену необходимо выполнить **все лабораторные работы на минимум "удовлетворительно"**.

"Отлично" требует высоких результатов во всех компонентах (особенно в практической части).

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

4.3. Методические указания по организации вычислительной инфраструктуры

Условия применения:

- Курс рассчитан на студентов 2-го года обучения.
- Наличие доступа к вычислительным ресурсам (GitLab, Google Colab или Yandex DataSphere, JupyterHub, VS, среда Matlab).

- Разработаны лабораторные работы;
- Инфраструктура для приёма задач (gitlab, CI/CD) согласована с лабораторными работами.

Цели, задачи и ожидаемые результаты

Цели организации вычислительной инфраструктуры:

- закрепить навыки работы в IT инфраструктуре (приучить пользоваться гитом, jupyter-ноутбуками).

Задачи преподавателя:

- Организация регистрации студентов в Google Colab и Yandex DataSphere
- Создание учетных записей студентов в gitlab вуза;
- GitLab Runner;
- Работа над шаблонным репозиторием лабораторных работ;
- Написание инструкции для студентов.

Ожидаемые результаты студентов:

- начальное представление о работе в IT инфраструктуре (гит, нейминг).

Порядок реализации

Задача №1: Организация регистрации студентов в Google Colab и Yandex DataSphere

Задача №2: Создание учетных записей студентов в gitlab вуза

Задача №3: GitLab Runner.

Для выполнения CI/CD пайплайна был настроен GitLab Runner на удалённой виртуальной машине с ОС Ubuntu 24.04.

Последовательность настройки включала следующие шаги:

- Настройка системы – установка необходимых компонентов, таких, как Docker.
- Установка GitLab Runner по официальной инструкции.
- Регистрация Runner для частного сервера GitLab.

Задача №4: Работа над шаблонным репозиторием лабораторных работ

- Создать шаблон GitLab-репозитория с автотестами (pytest) для каждой ЛР. Этот репозиторий служит основой для всех новых лабораторных заданий и содержит преднастроенную структуру проекта и CI-конфигурации.
- Студенты делают fork, коммитают код, CI запускает тесты на CPU (GPU – по желанию).

Ключевые файлы и их назначение:

.gitlab-ci.yml – основной конфигурационный файл CI/CD. Определяет среду выполнения (образ Docker), команды для сборки.

get_path.py – скрипт для определения того, какие тесты использовать, по названию коммита.

Dockerfile – файл для запуска докер-образа с тестами.

README.md – инструкции к использованию шаблонного репозитория и тестов.

Также была реализована генерация отчётов о результатах тестирования кода. Вся структура максимально адаптирована для копирования студентами и минимизации порога входа при выполнении лабораторных

Порядок проверки корректности

Чек-лист для проверки лабораторных работ:

- гит инфраструктура с учетными записями студентов;

- шаблон гит репозитория для клонирования и работы студента с подключенными автотестами;
- инструкция по работе с гитом с подробным описанием именования методов и коммитов в ReadMe файле.

Задача №5: Написание инструкции для студентов

README.md – инструкции к использованию шаблонного репозитория.

Вся структура максимально адаптирована для копирования студентами и минимизации порога входа при выполнении лабораторных

4.4. Методические указания по организации лабораторных работ

Условия применения:

- Курс рассчитан на студентов 2-го года обучения.
- Наличие доступа к вычислительным ресурсам (GitLab) и к GPU/CPU.
- Разработана инфраструктура для приёма задач (Gitlab, CI/CD) и согласована с лабораторными работами и настроена на всех студентов образовательной программы;
- Использование открытых датасетов и библиотек.

Цели, задачи и ожидаемые результаты

Цели организации лабораторных работ:

- Закрепление теоретических знаний на практике.
- Развитие навыков разработки нейронных сетей.
- Подготовка к решению реальных задач в индустрии.

Задачи преподавателя:

- Обеспечить студентов структурированными лабораторными работами.
- Предоставить доступ к необходимым вычислительным ресурсам.
- Организовать проверку и обратную связь по выполненным работам.

Ожидаемые результаты студентов:

- Навыки разработки нейронных сетей на python, matlab / C++.
- Умение подбирать архитектуру нейросети и оптимизаторы в соответствии с задачами.
- Работа с библиотеками (numpy, matplotlib, seaborn, pandas, scikit learn, nntool).
- Опыт решения задач классификации, кластеризации и регрессии.
- Умение интерпретировать результаты обучения нейросети.

Порядок реализации

Задача №1: Подготовка лабораторных работ (в соответствии с п. 2.3.3 РПД)

1) Определение тем:

- Однослойный и многослойный персептрон. Функции активации.
- Вероятностные и радиально-базисные нейронные сети.
- Самоорганизующиеся карты Кохонена.
- Сеть Хопфилда.

- Спайковые нейронные сети.
- Оптимизационные алгоритмы в нейронных сетях
- Диагностика и устранение проблем переобучения и недостаточного обучения нейронных сетей. Интерпретация результатов обучения

Наименование Лабораторной работы	Содержание Лабораторной работы	Распределение часов
2	3	
Однослойный и многослойный персептрон. Функции активации.	Разработка нейронных сетей в виде однослойного и двухслойного персептрона. Изучение алгоритма обратного распространения ошибки с последующей его реализацией. Запрограммировать различные функции активации. Визуализировать результаты обучения. Проанализировать разработанные модели, сделать выводы о их качестве и предложить улучшения.	4 часа (2 часа ЛР, 2 часа СР)
Вероятностные и радиально-базисные нейронные сети.	Реализовать и обучить вероятностную нейронную сеть (PNN). Реализовать и обучить радиально-базисную нейронную сеть (RBF). Сравнить результаты работы обеих моделей. Определить, какая модель эффективнее справляется с задачей классификации и почему.	4 часа (2 часа ЛР, 2 часа СР)
Самоорганизующиеся карты Кохонена	Реализовать и обучить самоорганизующуюся карту Кохонена (SOM). Использовать SOM для визуализации и сегментирования данных. Интерпретировать полученный результат, выделив характерные черты каждого сегмента. Представить итоговый результат в виде тепловой карты или кластерного дерева.	4 часа (2 часа ЛР, 2 часа СР)
Сеть Хопфилда	Реализовать и обучить сеть Хопфилда. Использовать сеть Хопфилда для восстановления зашумленных образов. Оценить эффективность восстановления с помощью показателя точности совпадения восстановленных изображений с оригинальными.	4 часа (2 часа ЛР, 2 часа СР)
Спайковые нейронные сети	Реализовать и обучить спайковую нейронную сеть (SNN). Использовать SNN для классификации сигналов. Измерить процент правильных распознаваний и частоту ложных срабатываний.	4 часа (2 часа ЛР, 2 часа СР)
Оптимизационные алгоритмы в нейронных сетях	Реализовать и обучить нейронную сеть с различными оптимизаторами (SGDM, Adam, RMSProp, AdaGrad). Сравнить время обучения и качество результатов. Выбрать самый эффективный алгоритм для дальнейшего использования.	10 часов (4 часа ЛР, 6 часов СР)
Диагностика и устранение проблем	Определить, присутствует ли переобучение или недостаточная мощность модели.	8 часов (2 часа ЛР, 6 часов СР)

Наименование Лабораторной работы	Содержание Лабораторной работы	Распределение часов
2	3	
переобучения и недостаточно го обучения нейронных сетей. Интерпретаци я результатов обучения	Исправить проблему, скорректировав архитектуру или введя регуляризацию. Проверить улучшение качества рекомендаций на тестовом наборе данных. Объяснить решение нейросети через LIME/SHAP	

2) Разработка заданий:

- Пошаговые инструкции.
- Примеры кода.

Контрольные вопросы.

Разработка заданий для лабораторной работы «Однослойный и многослойный перцептрон. Функции активации»

Цель: познакомиться с концепциями однослойных и многослойных перцептронов, запрограммировать алгоритм обратного распространения ошибки, изучить влияние различных функций активации на результат обучения и сравнить оба типа моделей без использования специальных библиотек и фреймворков вроде TensorFlow, Keras, PyTorch и scikit-learn.

Технологии: Python, библиотеки numpy и matplotlib.

Длительность: 4 часа (2 часа лабораторных работ, 2 часа самостоятельной работы).

Результат: Код на GitHub + ответы на вопросы

2. Пошаговые инструкции

Для наглядности воспользуемся синтетическим датасетом XOR (исключающее ИЛИ), который требует наличия не-линейной границы разделения.

Шаг 1: Установка библиотек

Шаг 2: Генерация данных XOR

Шаг 3: Функция для визуализации данных

Шаг 4: Реализация однослойного перцептрона с сигмоидной функцией активации

Шаг 5: Обучение однослойного перцептрона

Шаг 6: Построение прогнозов с помощью однослойного перцептрона

Шаг 7: Реализация многослойного перцептрона с функциями ReLU и sigmoid

Шаг 8: Обучение многослойного перцептрона

Шаг 9: Построение прогнозов с помощью многослойного перцептрона

Шаг 10: Сравнение результатов

3. Примеры кода на Python

Полный скрипт

#Установка библиотек

```
import numpy as np
import matplotlib.pyplot as plt
```

#Генерируем данные XOR

```
np.random.seed(42)
```

```
X = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])
```

```
Y = np.array([0, 1, 1, 0])[:, None]
```

#Функция для визуализации

```
def visualize_data():
    colors = ['blue' if i == 0 else 'green' for i in Y.ravel()]
    plt.scatter(X[:, 0], X[:, 1], c=colors)
    plt.title('XOR Data')
    plt.show()
visualize_data()
```

#Реализация однослойного персептрона с сигмоидной функцией активации

```
class OneLayerPerceptron:
    def __init__(self, input_dim, output_dim):
        self.weights = np.random.rand(input_dim + 1, output_dim) # Включаем смещение
```

```
    def predict(self, x):
        z = np.dot(np.insert(x, 0, 1), self.weights) # Добавляем смещение
        return self.sigmoid(z)
```

```
    @staticmethod
    def sigmoid(x):
        return 1 / (1 + np.exp(-x))
```

```
    def train(self, X, Y, lr=0.1, epochs=1000):
        mse_history = []
```

```
        for _ in range(epochs):
            predictions = self.predict(X)
```

```
            # MSE loss
            error = Y - predictions
            mse = np.mean(error ** 2)
            mse_history.append(mse)
```

```
            delta_weights = np.dot(np.insert(X.T, 0, 1, axis=0), error.reshape((len(Y), 1)))
            self.weights += lr * delta_weights
```

```
        return mse_history
```

#Обучение однослойного персептрона

```
perceptron = OneLayerPerceptron(input_dim=X.shape[1], output_dim=Y.shape[1])
mse_history_one_layer = perceptron.train(X, Y)
```

#Прогнозы

```
predictions = perceptron.predict(X)
print("Predictions (One-Layer): ", predictions.round())
```

#Реализация многослойного персептрона:

```
class TwoLayerPerceptron:
    def __init__(self, input_dim, hidden_dim, output_dim):
        self.W1 = np.random.rand(input_dim + 1, hidden_dim) # Входной слой + смещение
        self.W2 = np.random.rand(hidden_dim + 1, output_dim) # Выходной слой + смещение
```

```

def forward(self, x):
    # Первый слой
    h = self.relu(np.dot(np.insert(x, 0, 1), self.W1)) # relu(h1)
    # Второй слой
    o = self.sigmoid(np.dot(np.insert(h, 0, 1), self.W2)) # sigmoid(o)
    return o

@staticmethod
def relu(x):
    return np.maximum(0, x)

@staticmethod
def sigmoid(x):
    return 1 / (1 + np.exp(-x))

def backward(self, X, Y, o, lr=0.1):
    # Backward pass
    error_out = Y - o
    grad_o = error_out * o * (1 - o) # Сигмоидная производная

    h = self.forward(X[:-1])[..., :-1] # Output без последнего элемента
    grad_h = np.dot(grad_o, self.W2[1:, :].T) * (h > 0) # Производная ReLU

    # Обновляем W2
    delta_W2 = np.outer(np.insert(h, 0, 1), grad_o)
    self.W2 += lr * delta_W2

    # Обновляем W1
    delta_W1 = np.outer(np.insert(X, 0, 1), grad_h)
    self.W1 += lr * delta_W1

def train(self, X, Y, lr=0.1, epochs=1000):
    mse_history = []

    for _ in range(epochs):
        outputs = self.forward(X)
        error = Y - outputs
        mse = np.mean(error ** 2)
        mse_history.append(mse)

        self.backward(X, Y, outputs, lr)

    return mse_history

#Обучение многослойного персептрона
ml_perceptron = TwoLayerPerceptron(input_dim=X.shape[1], hidden_dim=4,
output_dim=Y.shape[1])
mse_history_two_layers = ml_perceptron.train(X, Y)

#Прогнозы
predictions_ml = ml_perceptron.forward(X)
print("Predictions (Two-Layer): ", predictions_ml.round())

```

```
#Сравнение результатов
plt.figure(figsize=(10, 5))
plt.plot(mse_history_one_layer, label="One-Layer Perceptron")
plt.plot(mse_history_two_layers, label="Two-Layer Perceptron")
plt.title("MSE Loss Over Time")
plt.xlabel("Epochs")
plt.ylabel("Mean Square Error")
plt.legend()
plt.show()
```

4. Контрольные вопросы

1. Теоретические:

- Какую роль играют функции активации?
- Какие функции активации можно использовать в этой задаче?
- Что происходит на этапе обратного прохода по сети?

2. Практические:

- Попробуйте изменить количество нейронов в скрытом слое.
- Попробуйте изменить количество скрытых слоёв.
- Попробуйте использовать другие функции активации (например, Tanh или LeakyReLU), а также увеличить количество эпох обучения. Обратите внимание на изменения в качестве модели.

3. Аналитические:

- Сделайте выводы о том, как многослойная модель решает проблему XOR.
- Как можно улучшить результаты решения данной задачи?

Критерии оценки

Отлично: Полное выполнение всех шагов, анализ полученных результатов с построением функции потерь.

Хорошо: Корректный код, но без анализа полученных результатов.

Удовлетворительно: Частичное выполнение с ошибками на этапах визуализации модели и сравнения результатов.

Неудовлетворительно: Невыполнение ключевых этапов.

3) Подготовка датасетов:

- Подбор открытых данных с Kaggle и HuggingFace
- Генерация синтетических данных при необходимости.

Задача №2: доступ к необходимым вычислительным ресурсам (в п.4.3 РПД)

Задача №3: Организация проверки и обратной связи по выполненным работам.

Порядок проверки корректности

Чек-лист для проверки лабораторных работ:

1. Выполнение заданий:
 - Код запускается без ошибок.
 - Достигнуто целевое качество модели.
2. Качество кода:
 - Соблюдение PEP-8.
 - Наличие комментариев.

3. Отчет:

инструкция по работе с гитом с подробным описанием именования методов и коммитов;
Описание хода работы.
Анализ результатов.

4. Своевременность:

Работа сдана в установленный срок.

Критерии оценки:

Отлично: Полное выполнение всех заданий, качественный код и отчет.

Хорошо: Незначительные недочеты в коде или отчете.

Удовлетворительно: Выполнены базовые задания, но с ошибками.

Неудовлетворительно: Критические ошибки или невыполнение работы.

4.5. Методические указания по организации проектной деятельности студентов

Условия применения:

- Курс рассчитан на студентов 2-го года обучения,
- Общее время на проект – не более 16 часов на каждого студента.
- Имеется доступ к кейсам промышленных партнеров; есть возможность адаптации кейсов для студентов второго курса.

Цели, задачи и ожидаемые результаты

Цели организации вычислительной инфраструктуры:

- дать начальное представление о реальных задачах, решаемых с помощью нейронных сетей и возникающих проблемах.

Задачи преподавателя:

- сбор кейсов промышленных партнеров;
- сбор кейсов преподавателей практиков и лабораторий в вузе;
- формирование ТЗ на экзаменационный проект на основе кейсов;
- разработка системы учёта результатов проекта в итоговой экзаменационной оценке

Ожидаемые результаты студентов:

- начальное представление о реальных задачах, решаемых с помощью нейросетей и возникающих проблемах.

Порядок реализации

Задача №1: сбор кейсов промышленных партнеров (подробнее кейсы приведены в п. 6 РПД)

1. Оптимизация кредитного скоринга с использованием нейронных сетей

Описание: Сбербанк активно развивает кредитные продукты и стремится улучшить процесс оценки кредитоспособности клиентов. Задача – разработать нейронную сеть, которая будет анализировать широкий спектр данных (финансовые показатели, поведенческие данные, социальные сети и т.д.) для более точной оценки риска.

Цель: Создать модель, которая сможет предсказывать вероятность дефолта с высокой точностью, используя различные типы данных.

Ожидаемый результат: Нейронная сеть, способная обрабатывать разнородные данные и выдавать точные прогнозы по кредитоспособности клиентов.

Задача № 2: кейсов преподавателей практиков и лабораторий в вузе.

1. Сортировка продуктов по вкусовым предпочтениям покупателей.

Постановка задачи: Вам дан массив данных о продуктах питания и рейтингах, выставленных покупателями. Нужно обучить вероятностную нейронную сеть (PNN) и радиально-базисную сеть (RBF), чтобы классифицировать продукты на категории: "полезные", "средние", "вредные".

Требования:

- Реализовать и обучить PNN и RBF.
- Сравнить результаты работы обеих моделей.
- Определить, какая модель эффективнее справляется с задачей классификации и почему.

2. Анализ демографической структуры населения города.

Постановка задачи: Имеется массив данных о жителях города, содержащих возраст, доход, образование и предпочтения в отдыхе. Необходимо использовать самоорганизующиеся карты Кохонена (SOM) для визуализации и сегментирования жителей по поведенческим признакам.

Требования:

- Обучить SOM для выделения сегментов населения.
- Интерпретировать полученный результат, выделив характерные черты каждого сегмента.
- Представить итоговый результат в виде тепловой карты или кластерного дерева.

Задача №3: формирование ТЗ на экзаменационный проект на основе кейсов

1. Применение различных активационных функций

Задание: Возьмите одну из нейросетей, разработанных в рамках лабораторных работ, реализуйте с различными функциями активации (ReLU, LeakyReLU, eeLU) и обучите сеть. Подготовьте отчет с результатами и сравнительным анализом.

Проект выполняется в командах от 1 до 3 человек. Оценивается вся команда одной оценкой.

Индивидуальное задание состоит в анализе качества разработанных нейросетей. Для выполнения задания необходимо выполнить несколько задач:

- подготовить набор данных для проведения тестирования;
- протестировать разработанные нейросети,
- посчитать функцию потерь
- построить ошибки на обучающих и тестовых данных.
- посчитать метрики
- сделать выводы о качестве нейросети

В зависимости от качества теоретического ответа и количества реализованного самостоятельно кода преподаватель выставляет оценку от 3 до 5.

Критерии оценки:

Отлично: Полное выполнение всех шагов, анализ полученных результатов с построением функции потерь и необходимых метрик, развернутый ответ на теоретические вопросы.

Хорошо: Корректный код, построена функция потерь, сделан правильный вывод о качестве работы модели. Ответил не на все теоретические вопросы.

Удовлетворительно: Выполнение кода с ошибками на этапах визуализации модели и сравнения результатов. Ответил не на все теоретические вопросы.

Неудовлетворительно: Невыполнение ключевых этапов. Не ответил на теоретические вопросы.

Оценку можно повысить, реализовав требуемый функционал или ответив дополнительно или заново на необходимые вопросы.

Требования для повышения оценки и итоговую оценку формирует ПРЕПОДАВАТЕЛЬ.
Задача №4: разработка системы учёта результатов проекта в итоговой оценке за экзамен
Выполнено в РПД, п 4.2

Порядок проверки корректности

Чек-лист для проверки лабораторных работ:

- Набор кейсов индустриальных партнеров – 13 шт;
- Набор кейсов преподавателей практиков и лабораторий ВУЗа – 7 шт;
- Набор ТЗ в количестве 9 штук.

5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

5.1 Основная литература:

1. Хливненко, Л. В. Практика нейросетевого моделирования : учебное пособие для вузов / Л. В. Хливненко, Ф. А. Пятакович. – 4-е изд., стер. – Санкт-Петербург : Лань, 2024. – 200 с. – ISBN 978-5-507-47590-2. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/393482> (дата обращения: 21.07.2025). – Режим доступа: для авториз. пользователей.

2. Кохонен, Т. Самоорганизующиеся карты : учебное пособие / Т. Кохонен ; под редакцией Ю. В. Тюменцева ; перевод с английского В. Н. Агеева. – 4-е эл.изд. – Москва : Лаборатория знаний, 2025. – 660 с. – ISBN 978-5-00101-179-8. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/495335> (дата обращения: 21.07.2025). – Режим доступа: для авториз. пользователей.

3. Ростовцев, В. С. Искусственные нейронные сети : учебник для вузов / В. С. Ростовцев. – 5-е изд., стер. – Санкт-Петербург : Лань, 2025. – 216 с. – ISBN 978-5-507-50568-5. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/447392> (дата обращения: 21.07.2025). – Режим доступа: для авториз. пользователей.

4. Рабчевский, А. Н. Синтетические данные и развитие нейросетевых технологий : учебное пособие для вузов / А. Н. Рабчевский. – Москва : Издательство Юрайт, 2024. – 187 с. – (Высшее образование). – ISBN 978-5-534-17716-9. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/545036> (дата обращения: 19.07.2025).

5.2 Дополнительная литература:

1. Sun, X., Li, J., Kovalenko, A.V., Feng, W., Ou, Y. Integrating Reinforcement Learning and Learning From Demonstrations to Learn Nonprehensile Manipulation //IEEE Transactions on Automation Science and Engineering, 2023, 20(3), 1735–1744, DOI: 10.1109/TASE.2022.3185071, Q1

2. Petukhova, A.V.; Kovalenko, A.V.; Ovsyannikova, A.V. Algorithm for Optimization of Inverse Problem Modeling in Fuzzy Cognitive Maps. Mathematics 2022, 10, 3452. DOI: 10.3390/math10193452, Q1

3. Kirillova, E.; Kovalenko, A.; Urtenov, M. Study of the Current–Voltage Characteristics of Membrane Systems Using Neural Networks. AppliedMath 2025, 5, 10. <https://doi.org/10.3390/appliedmath5010010>.

4. Kadurin, Artur, et al. "The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology." Oncotarget 8.7 (2016): 10883.

5. Kadurin, Artur, et al. "druGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico." *Molecular pharmaceutics* 14.9 (2017): 3098-3104.
6. Polykovskiy, Daniil, et al. "Molecular sets (MOSES): a benchmarking platform for molecular generation models." *Frontiers in pharmacology* 11 (2020): 565644.
7. Khrabrov, Kuzma, et al. " ∇^2 DFT: A Universal Quantum Chemistry Dataset of Drug-Like Molecules and a Benchmark for Neural Network Potentials." *Advances in Neural Information Processing Systems* 37 (2024): 36869-36889.
8. Polykovskiy, Daniil, et al. "Entangled conditional adversarial autoencoder for de novo drug discovery." *Molecular pharmaceutics* 15.10 (2018): 4398-4405.
9. Николенко, Сергей, Кадурин, Артур и Архангельская Екатерина. Глубокое обучение. "Издательский дом" Питер"", 2017.
10. Митяков, Е. С. Искусственный интеллект и машинное обучение : учебное пособие для вузов / Е. С. Митяков, А. Г. Шмелева, А. И. Ладынин. – Санкт-Петербург : Лань, 2025. – 252 с. – ISBN 978-5-507-51465-6. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/450827> (дата обращения: 21.07.2025). – Режим доступа: для авториз. пользователей.
11. Елисеев А. И., Минин Ю. В. Разработка программных интерфейсов веб-приложений с использованием фреймворка FastAPI : учебное пособие. Тамбов: ТГТУ, 2024. 81 с. <https://e.lanbook.com/book/472310> (дата обращения: 19.07.2025).
12. Чернышев, С. А. Основы программирования на Python : учебное пособие для вузов / С. А. Чернышев. – Москва : Издательство Юрайт, 2022. – 286 с. – (Высшее образование). – ISBN 978-5-534-14350-8. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/496893>. (дата обращения: 19.07.2025).
13. Златопольский Д. М. Основы программирования на языке Python. 2-е изд. Москва: ДМК Пресс, 2018.

5.3. Периодические издания и конференции:

1. Базы данных компании «Ист Вью» <http://dlib.eastview.com>
2. Электронная библиотека GREBENNIKON.RU <https://grebennikon.ru/>

Конференции А*:

3. <https://openreview.net/forum?id=FMMF1a9ifL>
4. <https://openreview.net/forum?id=ElUrNM9U8c#discussion>
5. <https://openreview.net/forum?id=JoO6mtCLHD>
6. <https://aclanthology.org/2024.findings-emnlp.760/>
7. <https://aclanthology.org/2020.coling-main.588/>
8. https://link.springer.com/chapter/10.1007/978-3-030-72113-8_30
9. https://link.springer.com/chapter/10.1007/978-3-031-42448-9_10
10. <https://aclanthology.org/2024.findings-naacl.288/>

5.4. Интернет-ресурсы, в том числе современные профессиональные базы данных и информационные справочные системы

Электронно-библиотечные системы (ЭБС):

1. ЭБС «ЮРАЙТ» <https://urait.ru/>
2. ЭБС «УНИВЕРСИТЕТСКАЯ БИБЛИОТЕКА ОНЛАЙН» <http://www.biblioclub.ru/>
3. ЭБС «BOOK.ru» <https://www.book.ru>
4. ЭБС «ZNANIUM.COM» www.znanium.com
5. ЭБС «ЛАНЬ» <https://e.lanbook.com>

Профессиональные базы данных

1. Scopus <http://www.scopus.com/>
2. ScienceDirect <https://www.sciencedirect.com/>

3. Журналы издательства Wiley <https://onlinelibrary.wiley.com/>
4. Научная электронная библиотека (НЭБ) <http://www.elibrary.ru/>
5. Полнотекстовые архивы ведущих западных научных журналов на Российской платформе научных журналов НЭИКОН <http://archive.neicon.ru>
6. Национальная электронная библиотека (доступ к Электронной библиотеке диссертаций Российской государственной библиотеки (РГБ) <https://rusneb.ru/>
7. Президентская библиотека им. Б.Н. Ельцина <https://www.prilib.ru/>
8. База данных CSD Кембриджского центра кристаллографических данных (CCDC) <https://www.ccdc.cam.ac.uk/structures/>
9. Springer Journals: <https://link.springer.com/>
10. Springer Journals Archive: <https://link.springer.com/>
11. Nature Journals: <https://www.nature.com/>
12. Springer Nature Protocols and Methods: <https://experiments.springernature.com/sources/springer-protocols>
13. Springer Materials: <http://materials.springer.com/>
14. Nano Database: <https://nano.nature.com/>
15. Springer eBooks (i.e. 2020 eBook collections): <https://link.springer.com/>
16. "Лекториум ТВ" <http://www.lektorium.tv/>
17. Университетская информационная система РОССИЯ <http://uisrussia.msu.ru>

Бесплатные образовательные ресурсы

1. Jupyter Notebook – интерактивные вычисления
2. Visual Studio Code – редактор кода с поддержкой Python и C++
3. Google Scholar/arXiv – доступ к научным публикациям

Ресурсы свободного доступа

1. Официальная документация по библиотеке numpy <https://numpy.org/doc/>
2. Официальная документация по библиотеке pandas <https://pandas.pydata.org/docs/>
3. Официальная документация по библиотеке matplotlib <https://matplotlib.org/stable/index.html>.
4. Официальная документация по библиотеке scikit-learn <https://scikit-learn.org/stable/>
5. Официальная документация по библиотеке seaborn <https://seaborn.pydata.org/>
6. КиберЛенинка <http://cyberleninka.ru/>;
7. Американская патентная база данных <http://www.uspto.gov/patft/>
8. Министерство науки и высшего образования Российской Федерации <https://www.minobrnauki.gov.ru/>;
9. Федеральный портал "Российское образование" <http://www.edu.ru/>;
10. Информационная система "Единое окно доступа к образовательным ресурсам" <http://window.edu.ru/>;
11. Единая коллекция цифровых образовательных ресурсов <http://school-collection.edu.ru/>.
12. Проект Государственного института русского языка имени А.С. Пушкина "Образование на русском" <https://pushkininstitute.ru/>;
13. Справочно-информационный портал "Русский язык" <http://gramota.ru/>;
14. Служба тематических толковых словарей <http://www.glossary.ru/>;
15. Словари и энциклопедии <http://dic.academic.ru/>;
16. Образовательный портал "Учеба" <http://www.ucheba.com/>;
17. Законопроект "Об образовании в Российской Федерации". Вопросы и ответы http://xn--273--84d1f.xn--p1ai/voprosy_i_otvety

Собственные электронные образовательные и информационные ресурсы КубГУ

1. Электронный каталог Научной библиотеки КубГУ
<http://megapro.kubsu.ru/MegaPro/Web>
2. Электронная библиотека трудов ученых КубГУ
<http://megapro.kubsu.ru/MegaPro/UserEntry?Action=ToDb&idb=6>
3. Среда модульного динамического обучения <http://moodle.kubsu.ru>
4. База учебных планов, учебно-методических комплексов, публикаций и конференций
<http://infoneeds.kubsu.ru/>
5. Библиотека информационных ресурсов кафедры информационных образовательных технологий <http://mschool.kubsu.ru;>
6. Электронный архив документов КубГУ <http://docspace.kubsu.ru/>
7. Электронные образовательные ресурсы кафедры информационных систем и технологий в образовании КубГУ и научно-методического журнала "ШКОЛЬНЫЕ ГОДЫ"
<http://icdau.kubsu.ru/>

6. Методические указания для обучающихся по освоению дисциплины (модуля)

По дисциплине «Математические модели нейронных сетей» предусмотрено проведение лекционных занятий, на которых даётся систематизированное представление об основах нейронных сетей. Курс сосредоточен на теоретическом освоении математического аппарата нейросетей и самостоятельном программировании архитектуры простых нейросетей и методов их обучения, обзорно даётся также библиотека scikit learn (использование персептронов, разбиение выборки, метрики) и пакет nntool среды Matlab. В ходе лекций студенты знакомятся с историей возникновения и устройством нейронных сетей, изучают архитектуры (персептрон, MLP, RBF, PNN, SNN, SOM и т.д.), методами и алгоритмами обучения. Особое внимание уделяется функциям активации и правильному их подбору под тип задачи. Подробно изучаются различные оптимизаторы и их влияние на работу нейросети. Демонстрируются практические примеры реализации нейронных сетей в задачах классификации, регрессии и кластеризации. Важным разделом также является интерпретация результатов работы нейросети, где студенты учатся выявлять недостаточно обученные и переобученные сети.

Лабораторные занятия курса посвящены практическому освоению разработки нейронных сетей различной архитектуры, с различными функциями активации и оптимизаторами. На занятиях студенты сначала учатся самостоятельно разрабатывать нейронную сеть с нуля (программировать алгоритм обратного распространения ошибки, функции активации, оптимизаторы), а затем используют готовые решения из библиотеки scikit learn и пакета nntool среды Matlab. Студенты работают с реальными датасетами, имеющимися в открытом доступе, используя библиотеки numpy, pandas, matplotlib, scikit learn, seaborn. Кроме того, помимо работы на ЯП Python, предусмотрено также использование библиотеки nntool среды Matlab или ЯП C++, с последующим сравнением, разработанных различными инструментами, нейросетей. После каждого лабораторного занятия предлагаются задания для самостоятельного закрепления материала - реализация отдельных этапов разработки нейросети или модификация рассмотренных алгоритмов.

При самостоятельной работе студентам необходимо изучать рекомендованную литературу (учебники, научные статьи, документацию библиотек) для глубокого понимания теоретических основ нейросетей. Выполняя проектные задания, студент должен уметь: формулировать задачу (классификация, регрессия, кластеризация); подбирать и преобразовывать данные; выбирать и реализовывать подходящую архитектуру; интерпретировать результаты работы нейросети. Особое внимание уделяется навыкам подбора функций активации и оптимизатора.

Важнейшим компонентом курса является самостоятельная проектная работа, в ходе которой студент разрабатывает законченное приложение (например, автоматизация сортировки, анализ демографической обстановки и т.д.). Такой проект позволяет закрепить

навыки проектирования и реализации комплексных решений в области нейросетевого моделирования.

Для студентов с ограниченными возможностями здоровья предусмотрены индивидуальные консультации и адаптированные материалы. Преподаватель помогает осваивать интерфейсы взаимодействия с ИИ, объясняет ключевые понятия в доступной форме, предоставляет инструкции с альтернативным форматированием. При необходимости используются голосовые интерфейсы, увеличенный масштаб экрана, сопровождение при выполнении заданий. Индивидуальный подход обеспечивает равные условия участия в образовательном процессе и достижения запланированных результатов обучения.

Подход, определяющий установление соответствия кейсов ИП и УГТ (5-7), позволяет четко соотносить этапы развития технологии с вовлеченностью партнера и снижать риски при переходе от лабораторных испытаний к промышленному внедрению.

Кейсы ПАО «Сбербанк»

1. Оптимизация кредитного скоринга с использованием нейронных сетей

Описание: Сбербанк активно развивает кредитные продукты и стремится улучшить процесс оценки кредитоспособности клиентов. Задача – разработать нейронную сеть, которая будет анализировать широкий спектр данных (финансовые показатели, поведенческие данные, социальные сети и т.д.) для более точной оценки риска.

Цель: Создать модель, которая сможет предсказывать вероятность дефолта с высокой точностью, используя различные типы данных.

Ожидаемый результат: Нейронная сеть, способная обрабатывать разнородные данные и выдавать точные прогнозы по кредитоспособности клиентов.

2. Обнаружение мошеннических действий

Описание: В Сбербанке ежедневно обрабатывается огромное количество транзакций, и важно выявлять аномалии и подозрительные операции для предотвращения мошенничества. Задача – разработать модель, которая будет анализировать транзакционные данные и выявлять аномалии.

Цель: Создать модель, которая сможет автоматически выявлять аномальные транзакции и предупреждать о возможных мошеннических действиях.

Ожидаемый результат: Нейронная сеть, способная анализировать транзакционные данные и выявлять аномалии с высокой точностью.

3. Оптимизация работы отделений банка

Описание: Сбербанк имеет множество отделений, и важно оптимизировать их работу, улучшая распределение ресурсов и сокращая время ожидания клиентов. Задача – разработать модель, которая будет анализировать данные о работе отделений и улучшать их взаимодействие с клиентами.

Цель: Создать модель, которая сможет обучаться на основе данных о работе отделений и улучшать их взаимодействие с клиентами.

Ожидаемый результат: Нейронная сеть, способная оптимизировать работу отделений банка, за счет кластеризации (например, с помощью самоорганизующейся карты Кохонена), что повысит удовлетворенность клиентов и снизит нагрузку на сотрудников.

4. Модель анализа инвестиционной привлекательности малого бизнеса

Описание: Банк активно развивает кредитование и инвестиционные инструменты для малого и среднего предпринимательства (МСП). Требуется создать модель, которая на основе открытых и банковских данных (выручка, расходы, тип деятельности, отзывы, онлайн-активность) оценивает инвестиционную привлекательность МСП.

Цель: Разработать систему рейтинговой оценки компаний малого бизнеса с возможностью визуализации факторов и динамики показателей.

Ожидаемый результат: Модель, присваивающая компании инвестиционный рейтинг (например, А–Е), объясняющая ключевые параметры и дающая рекомендации для инвестора. Объяснить решение нейросети через LIME/SHAP

5. Индивидуальная оценка кредитоспособности клиента на основе поведенческих данных

Описание: Современный кредитный скоринг выходит за рамки финансовых данных. Необходимо исследовать, как поведенческие и цифровые следы (частота входа в мобильный банк, способы оплаты, география, время отклика) влияют на персональную оценку риска.

Цель: Разработать нейросеть, оценивающую вероятность дефолта по нестандартным поведенческим признакам.

Ожидаемый результат: Прототип скоринговой модели, которая, помимо стандартных данных, учитывает цифровой профиль клиента и объясняет решения (SHAP, LIME и др.).

6. Оптимизация маршрута доставки

Описание: Сбербанк предоставляет услуги по доставке документов и других материалов. Задача – разработать модель, которая будет оптимизировать маршруты доставки, учитывая различные факторы, такие как время, расстояние и загруженность дорог.

Цель: Создать модель, которая сможет находить оптимальные маршруты для доставки, что сократит время и затраты на логистику.

Ожидаемый результат: Спайковая нейронная сеть, способная анализировать данные и находить оптимальные маршруты доставки.

Кейсы от «АВАЛАБ»

1. Модель прогнозирования сроков сдачи объектов на основе табличных данных

Описание: Девелоперская компания ведёт аналитический архив по срокам строительства. С помощью нейросетевой модели можно прогнозировать вероятность отклонения от графика сдачи.

Цель: Разработать модель, которая по текущему статусу объекта (отчёт СМР) оценивает риски задержек.

Ожидаемый результат: Прототип, который показывает вероятность отклонений и даёт текстовые пояснения о том, на каком этапе произошла задержка сроков.

2. Прогнозирование сроков строительства с использованием нейронных сетей

Описание: Строительная компания сталкивается с необходимостью точного прогнозирования сроков завершения строительных проектов. Задача – разработать модель, которая будет анализировать данные о предыдущих проектах и прогнозировать сроки завершения текущих проектов.

Цель: Создать модель, которая сможет предсказывать сроки завершения проектов с высокой точностью, учитывая различные факторы, такие как погодные условия, доступность ресурсов и другие.

Ожидаемый результат: Нейронная сеть, способная анализировать данные и выдавать точные прогнозы по срокам завершения проектов.

3. Анализ качества строительных материалов с использованием нейронных сетей

Описание: Строительная компания использует различные строительные материалы, и важно контролировать их качество. Задача – разработать модель, которая будет анализировать данные о материалах и прогнозировать их качество.

Цель: Создать модель, которая сможет анализировать данные о материалах и предсказывать их качество, что поможет избежать использования некачественных материалов.

Ожидаемый результат: Нейронная сеть, способная анализировать данные и выдавать прогнозы по качеству строительных материалов.

4. Оптимизация маршрутов доставки материалов

Описание: Девелоперская компания предоставляет услуги по доставке материалов и оборудования на стройплощадки. Задача – разработать модель, которая будет оптимизировать маршруты доставки, учитывая различные факторы, такие как время, расстояние и загруженность дорог.

Цель: Создать модель, которая сможет находить оптимальные маршруты для доставки, что сократит время и затраты на логистику.

Ожидаемый результат: Спайковая нейронная сеть, способная анализировать данные и находить оптимальные маршруты доставки.

5. Оптимизация работы с клиентами в онлайн-сервисах

Описание: Девелоперская компания активно развивает свои онлайн-сервисы для взаимодействия с клиентами. Задача – разработать модель, которая будет оптимизировать работу с клиентами в онлайн-сервисах, улучшая их опыт и удовлетворенность.

Цель: Создать модель, которая сможет обучаться на основе данных о поведении клиентов в онлайн-сервисах и улучшать их взаимодействие с компанией.

Ожидаемый результат: Сеть Хопфилда, способная оптимизировать работу с клиентами в онлайн-сервисах, что повысит их удовлетворенность и лояльность.

6. Оценка инвестиционной привлекательности проектов

Описание: Девелоперская компания активно развивает свои проекты и стремится улучшить процесс оценки инвестиционной привлекательности объектов. Задача – разработать модель, которая будет анализировать данные о проектах и прогнозировать их инвестиционную привлекательность.

Цель: Создать модель, которая сможет предсказывать инвестиционную привлекательность проектов с высокой точностью, используя различные типы данных.

Ожидаемый результат: Вероятностная или радиально-базисная нейронная сеть, способная анализировать данные и выдавать точные прогнозы по инвестиционной привлекательности проектов. Объяснить решение нейросети через LIME/SHAP

7. Оптимизация управления ресурсами на стройплощадке

Описание: Девелоперская компания управляет множеством ресурсов на стройплощадке, включая материалы, оборудование и персонал. Задача – разработать модель, которая будет оптимизировать управление ресурсами, улучшая их использование и сокращая простои.

Цель: Создать модель, которая сможет обучаться на основе данных о ресурсах и улучшать их распределение и использование.

Ожидаемый результат: Нейросеть, способная оптимизировать управление ресурсами на стройплощадке за счет кластеризации (например, с помощью самоорганизующейся карты Кохонена), что повысит эффективность работы компании и снизит затраты.

7. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю)

7.1 Перечень информационно-коммуникационных технологий

1. Облачные платформы и сервисы

Google Colab – облачная среда для выполнения кода на Python с GPU/TPU

Kaggle – платформа для работы с датасетами и соревнований по ML

cloud.ru, YandexCloud, AWS/GCP/Azure – облачные вычисления

2. Системы управления версиями и коллаборации
 Git/GitHub/GitLab – контроль версий кода и совместная разработка
 Notion/Trello – организация проектной деятельности

3. Система управления обучением
 Moodle – сдача работ

7.2 Перечень лицензионного и свободно распространяемого программного обеспечения

1. Лицензионное ПО
 Visual Studio (VS) – IDE для Python, C++
 LibreOffice – оформление отчетов (свободно распространяемое)
 Matlab R2020 и выше – разработка нейросетей

2. Свободное ПО (Open Source)
 JupyterHub – для работы с jupyter notebook

Библиотеки:

numpy – работа с матрицами
 pandas – работа с датасетами (загрузка из файлов, предобработка)
 scikit-learn – разработка простых нейросетей
 nntool (Matlab) – разработка нейросетей

Инструменты для визуализации:

Matplotlib/Seaborn – графики и анализ данных

СУБД:

SQLite/PostgreSQL – хранение структурированных данных
 FAISS/Annoy – векторный поиск

8. Материально-техническое обеспечение по дисциплине (модулю)

Виртуальные машины и ресурсы GPU в облаке предоставляется промышленным партнером ПАО «Сбербанк»:

№	Продукт	Параметры продукта	Кол-во	Кол-во конфигураций	Ед. изм.
1	Виртуальная машина	Виртуальная машина 10% vCPU 2 vCPU 4 RAM	1	60	Шт
		ОС Ubuntu 22.04	1		Шт
		Системный диск SSD	1		Шт
			10		Гб
		Аренда публичного IP	1		Шт
2	K8S	Master node 8 vCPU 16 RAM	1	1	Шт
		Worker node 10% доля 4 vCPU 32 RAM	5		Шт

		Worker node SSD-NVME	64	Гб
		Аренда публичного IP	1	Шт
3	LLM	Токены GigaChat 2 Max	50	Млн. Шт
		Токены Embeddings	400	Млн. Шт

Дополнительные облачные ресурсы предоставляются технологическим партнером Yandex Cloud.

№	Вид работ	Наименование учебной аудитории, ее оснащенность оборудованием и техническими средствами обучения
1.	Лекционные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения
2.	Лабораторные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, проектором, программным обеспечением
3.	Практические занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения
4.	Групповые (индивидуальные) консультации	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
5.	Текущий контроль, промежуточная аттестация	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
6.	Самостоятельная работа	Кабинет для самостоятельной работы, оснащенный компьютерной техникой с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета.

Примечание: Конкретизация аудиторий и их оснащение определяется ОПОП.