МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ СВЕРДЛОВСКОЙ ОБЛАСТИ «ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

Рабочая программа учебной дисциплины ОП.07 «ВВЕДЕНИЕ В НЕЙРОННЫЕ СЕТИ» для специальности 09.02.09 «Веб-разработка»

Екатеринбург
2026

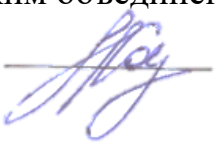
Чистяков Валерий Николаевич
я подтверждаю этот документ своей удостоверяющей подписью
место подписания
2026.03.30 12:06:03+05'00'

Рабочая программа учебной дисциплины ОП.07 «Введение в нейронные сети» нацелена на решение задач, разработана на основе Федерального государственного образовательного стандарта среднего профессионального образования (далее – ФГОС СПО) по специальности 09.02.09 Веб-разработка, примерной рабочей программы учебной дисциплины, учебного плана специальности.

Организация-разработчик: **ГАПОУ СО «Екатеринбургский монтажный колледж»**

Разработчики:
Грибова А. В. – преподаватель ГАПОУ СО «Екатеринбургский монтажный колледж», ВКК

Рассмотрена методическим объединением Информационных технологий

Руководитель МО  Софьина Н.А.

УТВЕРЖДАЮ

Заместитель директора по учебной работе

 Л. С. Хоринова

«13» марта 2026

СОДЕРЖАНИЕ

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ УЧЕБНОЙ ДИСЦИПЛИНЫ ОП.04 «ВВЕДЕНИЕ В НЕЙРОННЫЕ СЕТИ» 4

1.1. Область применения программы 4

1.2. Место учебной дисциплины в структуре программы подготовки специалистов среднего звена 4

1.3. Цели и планируемые результаты освоения учебной дисциплины 4

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ 6

2.1. Объем учебной дисциплины и виды учебной работы 6

2.2. Тематический план и содержание учебной дисциплины 7

3. УСЛОВИЯ РЕАЛИЗАЦИИ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ 9

3.1. Требования к минимальному материально-техническому обеспечению 9

3.2. Информационное обеспечение реализации программы 9

3.3. Общие требования к организации образовательного процесса 9

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ 10

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ 12

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ УЧЕБНОЙ ДИСЦИПЛИНЫ ОП.04 «ВВЕДЕНИЕ В НЕЙРОННЫЕ СЕТИ»

1.1. Область применения программы

Рабочая программа учебной дисциплины «Введение в нейронные сети» является частью основной образовательной программы в соответствии с ФГОС по специальности СПО **09.02.09 «Веб-разработка»** квалификация выпускника **«Разработчик Веб-**приложений» в части подготовки студентов к освоению профессионального модуля ПМ.01 «Проектирование и разработка информационных ресурсов» и овладению профессиональных компетенций.

Учебная дисциплина направлена на изучение самых популярных методов создания систем искусственного интеллекта – глубоких нейронных сетей. Курс рассматривает архитектуры нейронных сетей, вопросы применения нейросетей для решения прикладных задач компьютерного зрения и регрессии.

1.2. Место учебной дисциплины в структуре программы подготовки специалистов среднего звена

Учебная дисциплина входит в общепрофессиональный цикл.

Особое значение дисциплина имеет при формировании и развитии общих компетенций ОК 01, ОК 02, ОК 05, ОК 9

1.3. Цели и планируемые результаты освоения учебной дисциплины

В рамках программы учебной дисциплины обучающимися осваиваются умения и знания, представленные в таблице 1.

Таблица 1. Умения и знания, осваиваемые в рамках учебной дисциплины

Код ПК, ОК	Умения	Знания
ОК1 ОК 2 ОК 4 ОК 5 ОК 9 ПК 1.1 ПК 2.1 ПК 2.2 ПК 2.3	Использования методов обучения искусственного нейрона и искусственной нейронной сети; Использования методов построения полносвязных и сверточных нейронных сетей; Применения нейронных сетей для решения различных прикладных задач искусственного интеллекта.	устройство искусственного нейрона; методы обучения искусственного нейрона и искусственной нейронной сети; методы построения полносвязных и сверточных нейронных сетей; особенности применения нейронных сетей для решения различных прикладных задач искусственного интеллекта

1.4 Цели и планируемые результаты освоения учебной дисциплины

Результатом освоения учебной дисциплины является овладение обучающимися общими (ОК) и профессиональными (ПК) компетенциями, представленными в Таблице 2.

Таблица 2. Результаты освоения учебной дисциплины

Код	**Наименование результата обучения**
ОК 01.	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02.	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности
ОК 03.	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях.
ОК 04.	Эффективно взаимодействовать и работать в коллективе и команде
ОК 05.	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09.	Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 1.1	Проектировать информационные ресурсы
ПК 1.3	Интегрировать программный код в соответствующую инфраструктуру
ПК 1.4	Использовать систему контроля версий в процессе коллективной (параллельной) разработки

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Объем учебной дисциплины и виды учебной деятельности представлены в таблице 3.

Таблица 3. Структура и содержание учебной дисциплины

Вид учебной работы	**Объем часов**
Суммарная учебная нагрузка во взаимодействии с преподавателем	76
Самостоятельная работа	12
Объем образовательной программы	76
в том числе:	
теоретическое обучение	24
практические занятия	40
Самостоятельная работа	12
Итоговая аттестация проводится в форме дифференцированного зачета	

2.2. Тематический план и содержание учебной дисциплины

Наименование разделов и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная учебная работа обучающихся, курсовая работа (проект) *(если предусмотрены)*	Объем часов	Коды компетенций, формированию которых способствует элемент программы
1	2	3	4
Тема 1.1. Введение в тематику искусственных нейронных сетей	**Содержание**	**6**	ОК 01. ОК 02. ОК 03. ОК 04. ОК 05. ОК 09. ПК 1.1 ПК 1.3 ПК 1.4
	Модель искусственного нейрона и архитектура глубоких сетей	2	
	Инструментальный стек и библиотеки для обучения нейронных сетей	2	
	Полный цикл обучения нейронной сети: от данных до выбора архитектуры нейронной сети	2	
	Тематика практических занятий и лабораторных работ	**12**	
	1. Математика нейрона	2	
	2. Настройка окружения и работа с тензорами	2	
	3. Предобработка данных и EDA	2	
	4. Построение первой MLP-сети	2	
	5. Борьба с переобучением	2	
	6. Векторизация и оценка качества	2	
Тема 1.2. Нейронные сети для задачи анализа изображений	**Содержание**	**10**	ОК 01. ОК 02. ОК 03. ОК 04. ОК 05. ОК 09. ПК 1.1
	Основы сверточных нейронных сетей (CNN)	2	
	Распознавание предметов (Image Classification) и современные архитектуры	2	
	Распознавание объектов на изображении (Object Detection)	2	
	Семантическая и контекстная сегментация (Semantic Segmentation).	2	
	Введение в генерацию изображений и автокодировщики	2	
	Тематика практических занятий и лабораторных работ	**12**	

	1. Основы CNN: Свертка и Пулинг	2	ПК 1.3 ПК 1.4
	2. Глубокие архитектуры и Transfer Learning	2	
	3. Распознавание объектов (Object Detection) с YOLO	2	
	4. Семантическая сегментация (U-Net)	2	
	5. Автокодировщики (Autoencoders) и сжатие данных	2	
	6. Введение в генеративные модели (VAE/GAN)	2	
Тема 1.3. Нейронные сети для задачи анализа естественного языка	**Содержание**	**8**	ОК 01. ОК 02. ОК 03. ОК 04. ОК 05. ОК 09. ПК 1.1 ПК 1.3 ПК 1.4
	Представление текста и одномерные сверточные сети	2	
	Рекуррентные нейронные сети для задач обработки естественного языка	2	
	Архитектура Трансформер и механизм внимания	2	
	Большие языковые модели (LLM) и практическое применение	2	
	Тематика практических занятий и лабораторных работ	**16**	
	1. Векторизация текста и 1D-CNN	2	
	2. Рекуррентные сети (RNN, LSTM, GRU)	2	
	3. Рекуррентные сети (RNN, LSTM, GRU)	2	
	4. Работа с предобученными моделями (Hugging Face)	2	
	5. Дообучение (Fine-tuning) BERT	2	
	6. Применение LLM и Prompt Engineering	2	
	7. RAG (Retrieval-Augmented Generation) на практике	2	
	8. RAG (Retrieval-Augmented Generation) на практике	2	
Самостоятельная работа: Применить CNN, трансферное обучение и сегментацию к реальным задачам: – Классификация дефектов материалов (Найти датасет с фотографиями поверхности (металл, ткань, пластик) с дефектами и без. Обучить CNN отличать бракованные изделия от годных. – Распознавание эмоций по лицам (Использовать датасет FER-2013. Обучить сеть определять 7 эмоций (радость, гнев, грусть и т.д.) по черно-белым снимкам лиц)		12	
Итоговая аттестация в форме дифференцированного зачёта			

3. УСЛОВИЯ РЕАЛИЗАЦИИ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1. Требования к минимальному материально-техническому обеспечению

Реализация программы предполагает наличие стандартного учебного кабинета и лаборатории программирования.

Оборудование лаборатории и рабочих мест лаборатории:

- Учебная доска;
- Рабочие места по количеству обучающихся;
- Рабочее место преподавателя;
- Персональный компьютер с лицензионным программным обеспечением;
- Проектор и мультимедийная доска;
- Наглядные пособия;
- Программное обеспечение общего и профессионального назначения.

3.2. Информационное обеспечение реализации программы

Для реализации программы библиотечный фонд ГАПОУ СО «ЕМК» обладает следующим перечнем используемых учебных изданий, Интернет- ресурсов, дополнительной литературы

Основные источники

Основные электронные издания

1. Руководство по Python – Документация по Python 3 . [Электронный ресурс]. Режим доступа https://www.python.org/doc/ (дата обращения 22.03.2026)

Дополнительные источники

1. Ginsburg Ilya Пишем нейросеть на Python с нуля. [Электронный ресурс]. - Режим доступа: https://proglib.io/p/pishem-neyroset-na-python-s-nulya-2020-10-07 (дата обращения 22.03.2026)

2. Первая модель: набор данных Fashion MNIST[Электронный ресурс]. - Режим доступа: https://habr.com/ru/articles/454034/ (дата обращения 22.03.2026)

3.3. Общие требования к организации образовательного процесса

Освоению данной дисциплины предшествует изучение следующих учебных дисциплин:

– ОП 03. Основы алгоритмизации и программирования (весь практический блок курса строится на умении работать с языком программирования, понимании структуру данных, использовании библиотек

– ОП 01. Элементы высшей математики (курс строится на понимании математического аппарата: производные и градиенты, функции, матрицы и векторы)

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Результаты обучения	Критерии оценки	Формы и методы оценки
Перечень осваиваемых компетенций в рамках дисциплины: ОК 01.,ОК 02., ОК 03., ОК 04. ОК 05., ОК 09., ПК 1.1, ПК 1.3 ПК 1.4	Оценка «отлично» - Обучающийся показывает полные и глубокие знания программного материала, логично и аргументированно отвечает на поставленный вопрос, а также на дополнительные вопросы, демонстрируя понимание связей между архитектурой, математикой и результатом. Показывает высокий уровень теоретической подготовки. Оценка «хорошо» - Обучающийся показывает глубокие знания материала, грамотно его излагает, достаточно полно отвечает на основной и дополнительные вопросы, умело формулирует выводы. При ответе допускает несущественные погрешности (неточности в терминах, мелкие ошибки в формулах), не влияющие на общую логическую цепочку. Оценка «удовлетворительно» - Обучающийся показывает достаточные, но не глубокие знания; при ответе не допускает грубых	**Текущий контроль в форме:** – практических заданий; – задач для самостоятельного выполнения. **Итоговая аттестация:** дифференцированный зачет
Перечень умений, осваиваемых в рамках дисциплины: – Разрабатывать архитектуры нейронных сетей (MLP, CNN, RNN, Transformer) для конкретных прикладных задач (классификация, детекция, генерация текста). – Использовать специализированные библиотеки и фреймворки для глубокого обучения (Keras, PyTorch, Hugging Face) для построения и обучения моделей. – Определять сложность и вычислительные требования обучения нейронных сетей (подбор гиперпараметров,(batch size, learning rate), оценка времени обучения). – Работать в среде программирования для реализации пайплайнов машинного обучения. – Реализовывать построенные архитектуры в виде программных модулей с применением векторизованных операций и автоматического дифференцирования. – Оформлять код программы в соответствии со стандартами кодирования (PEP 8, модульность, документация). – Выполнять проверку, отладку кода программы, диагностику проблем обучения (переобучение, затухание градиента) и проводить коррекцию модели.		
Перечень знаний, осваиваемых в рамках дисциплины: – Понятие искусственного нейрона, математическую модель, функции активации (ReLU, Sigmoid, Softmax), принцип работы		

прямого и обратного распространения ошибки. – Основные элементы архитектуры глубоких сетей: слои (Dense, Conv, Pooling, LSTM, Attention), механизмы регуляризации (Dropout, BatchNorm), функции потерь. – Структуру программы обучения нейросети: инициализация весов, оптимизаторы (SGD, Adam), циклы обучения, валидация, сохранение и загрузка моделей. – Особенности работы с данными: предобработка, нормализация, аугментация (для изображений), токенизация и векторизация (для текста), понятие дисбаланса классов. – Принципы работы современных архитектур: сверточных сетей (CNN), рекуррентных сетей (RNN/LSTM), трансформеров (Self-Attention) и больших языковых моделей (LLM). – Методы оценки качества моделей: метрики (Accuracy, Precision, Recall, F1, IoU), матрица ошибок, анализ кривых обучения. – Основы трансферного обучения, дообучения (Fine-tuning) и использования предобученных моделей (Hugging Face, YOLO).	ошибок, однако отсутствует должная связь между анализом, аргументацией и выводами. Для получения правильного ответа требуются уточняющие вопросы преподавателя. Знания формальные, фрагментарные. Оценка «неудовлетворительно» - Обучающийся показывает недостаточные знания, не способен аргументированно и последовательно излагать материал. Допускает грубые ошибки в ответах (путает понятия, неверно описывает принцип работы), неправильно отвечает на основной вопрос или затрудняется с ответом даже после наводящих вопросов.	

МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ СВЕРДЛОВСКОЙ ОБЛАСТИ
«ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ОП.07 «ВВЕДЕНИЕ В НЕЙРОННЫЕ СЕТИ»

для специальности

09.02.09 «Веб-разработка»

Екатеринбург

2026

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1. Назначение

Данный фонд оценочных средств разработан для сопровождения учебной дисциплины ОП 07.«Введение в нейронные сети», входящей в вариативную часть профессионального цикла основной образовательной программы по специальности 09.02.09 «Веб-разработка».

Фонд ориентирован на формирование у обучающихся фундаментальных знаний о принципах работы глубоких нейронных сетей, понимание факторов, сделавших их практическое применение возможным, а также на выработку практических навыков проектирования, обучения, настройки и развертывания нейросетевых моделей для решения прикладных задач в области компьютерного зрения, обработки естественного языка (NLP). Особое внимание уделено развитию профессиональных компетенций, способствующих успешной профессиональной деятельности в сфере разработки современных программных решений с элементами ИИ.

Фонд призван помочь обучающимся:

- усвоить ключевые понятия нейронных сетей;
- научиться применять знания на практике;
- освоить инструментарий;
- развить навыки самостоятельного исследования.

Основной целью фонда оценочных средств является обеспечение системного контроля уровня освоения дисциплины через комплекс теоретических вопросов, практических заданий и критериев оценки, соответствующих требованиям ФГОС СПО.

Формы контроля:

- Текущий контроль: выполнение 20 практических занятий (лабораторных работ), охватывающих полный цикл от построения простейшего перцептрона до работы с большими языковыми моделями (LLM).
- Итоговая аттестация: выполнение индивидуального проекта (самостоятельной работы) по выборочной теме (например, «Система распознавания дефектов», «Анализ тональности отзывов», «Генерация текстов»), защита которого демонстрирует способность решать нетиповые прикладные задачи.

Результатом освоения дисциплины является способность выпускника самостоятельно создавать, обучать и внедрять прототипы нейросетевых моделей для автоматизации задач, ранее решавшихся только человеческим интеллектом, что соответствует современным требованиям рынка труда в сфере разработки программного обеспечения и искусственного интеллекта.

2. Структура фонда оценочных средств

Материал фонда оценочных средств структурирован в соответствии с разделами рабочей программы и включает:

- Теоретический материал, который направлен на проверку знаний по модели искусственного нейрона, архитектуре глубоких сетей (MLP, CNN, RNN, Transformer), принципам работы алгоритмов обратного распространения

ошибки, факторам, обеспечившим практическую реализацию глубокого обучения (аппаратное ускорение, Big Data, алгоритмические улучшения), а также ключевым понятиям современного инструментария (библиотеки Keras/TensorFlow, PyTorch, Hugging Face) и методам оценки качества нейросетевых моделей.

– Практические задания, которые направлены на комплексное освоение основных принципов работы с искусственными нейронными сетями: настройку вычислительного окружения, работу с тензорами, предобработку данных и EDA (разведочный анализ), построение первых полносвязных сетей (MLP), проведение экспериментов по борьбе с переобучением (Dropout, регуляризация), реализацию основ сверточных сетей (свертка, пулинг), построение глубоких архитектур (VGG-style) и применение трансферного обучения (Transfer Learning), решение задач распознавания объектов (Object Detection) с использованием YOLO, реализацию семантической сегментации (U-Net), работу с автокодировщиками для сжатия данных и введение в генеративные модели (VAE/GAN), векторизацию текста и применение одномерных сверток (1D-CNN), построение рекуррентных сетей (RNN, LSTM, GRU) для работы с последовательностями, использование предобученных моделей через Hugging Face, дообучение (Fine-tuning) моделей семейства BERT, применение больших языковых моделей (LLM) с использованием Prompt Engineering и реализацию RAG-систем (Retrieval-Augmented Generation) на практике.

Критерии оценки – обеспечивают объективную систему измерения усвоения материала, выполнения практических умений (выбор архитектуры, подбор гиперпараметров, борьба с переобучением) и развития профессиональных компетенций в области проектирования, обучения,-debugging и внедрения нейросетевых моделей для решения прикладных задач в сфере компьютерного зрения, обработки естественного языка и генерации контента.

3. Краткое изложение планов хода лекционных занятий

Тема «Модель искусственного нейрона и архитектура глубоких сетей»

Рассмотрение биологической аналогии и математической модели нейрона (вес, смещение, суммирование). Введение понятий функции активации (ReLU, Sigmoid, Tanh), прямого распространения сигнала и многослойной структуры. Анализ эволюции от перцептрона к глубоким сетям, проблема затухания градиента и роль «глубины» в иерархическом извлечении признаков.

Тема «Инструментальный стек и библиотеки для обучения нейронных сетей»

Обзор экосистемы глубокого обучения: сравнение основных фреймворков (Keras/TensorFlow, PyTorch). Принцип работы с тензорами, автоматическое вычисление градиентов (autograd). Структура проекта, организация данных (даталоадеры), выбор оптимизаторов и функций потерь. Базовый код создания и компиляции модели.

Тема «Полный цикл обучения нейронной сети: от данных до выбора архитектуры»

Этапы решения задачи: предобработка и нормализация данных, аугментация, разделение на обучающую, валидационную и тестовую выборки. Методология построения архитектуры (MLP), понятие переобучения (overfitting) и недообучения. Стратегии регуляризации (Dropout, L2), мониторинг метрик (loss, accuracy) и выбор лучшей модели.

Тема «Основы сверточных нейронных сетей (CNN) »

Ограничения полносвязных сетей для изображений. Принцип работы сверточного слоя (локальные связи, общие веса), типы свертков, операции пулинга (max, average) и параметры (stride, padding). Визуализация того, как сеть сохраняет пространственную структуру данных.

Тема «Распознавание предметов (Image Classification) и современные архитектуры»

Эволюция архитектур: от LeNet и AlexNet к VGG, ResNet (остаточные связи) и EfficientNet. Концепция трансферного обучения (Transfer Learning): использование предобученных весов, заморозка слоев, дообучение (fine-tuning) «головы» сети. Практические аспекты оценки качества классификации.

Тема «Распознавание объектов на изображении (Object Detection) »

Переход от классификации всего изображения к локализации объектов. Понятие ограничивающей рамки (bounding box), метрика IoU. Обзоры семейств детекторов: двухэтапные (R-CNN) и одноэтапные (YOLO, SSD). Принцип работы современных детекторов в реальном времени.

Тема «Семантическая и контекстная сегментация (Semantic Segmentation)»

Задача классификации каждого пикселя изображения. Архитектура типа «кодировщик- декодировщик» (Encoder-Decoder), пример U-Net. Роль пропускных связей (skip connections) для сохранения деталей. Области применения: медицина, автономное вождение.

Тема «Введение в генерацию изображений и автокодировщики»

Автокодировщики (Autoencoders) для сжатия данных и обучения представлений. Вариационные автокодировщики (VAE) и генеративно-состязательные сети (GAN): принцип «генератор против дискриминатора». Краткий обзор диффузионных моделей как основы современных генераторов (Stable Diffusion).

Тема «Представление текста и одномерные сверточные сети»

Проблемы представления текста: от мешка слов (Bag-of-Words) к векторным вложению (Word2Vec, GloVe). Принцип работы одномерных сверток (1D-CNN) для поиска локальных паттернов (n-gram) в последовательностях.

Применение для классификации настроений и тем.

Тема «Рекуррентные нейронные сети для задач обработки естественного языка»

Обработка последовательностей с памятью: архитектура RNN. Ограничения простых RNN (проблема долгосрочных зависимостей). Решения: ячейки LSTM (долгая краткосрочная память) и GRU. Би-направленные сети (Bi-RNN) для учета контекста с обеих сторон.

Тема «Архитектура Трансформер и механизм внимания»

Отказ от рекуррентности в пользу механизма самовнимания (Self-Attention). Математическая интуиция Q, K, V (Query, Key, Value). Структура трансформера (Encoder-Decoder), позиционное кодирование. Революция BERT и его вариаций в понимании контекста.

Тема «Большие языковые модели (LLM) и практическое применение»

Принципы предобучения (Pre-training) и дообучения (Fine-tuning) огромных моделей. Инженерия промптов (Prompt Engineering) как метод управления моделью. Архитектура RAG (Retrieval-Augmented Generation) для работы с внешними знаниями. Этика, ограничения и сценарии внедрения LLM в прикладные системы.

4. Краткое изложение планов хода лекционных занятий

Практическое занятие «Математика нейрона»

Ход занятия:

- Изучение биологической аналогии и математической модели искусственного нейрона.
- Реализация функции прямого распространения (sum of weighted inputs + bias) на базовых массивах.
- Вручную (без авто-градиентов) реализация функций активации (ReLU, Sigmoid) и их производных.
- Реализация алгоритма обратного распространения ошибки (backpropagation) для одного слоя: вычисление градиента функции потерь по весам и смещениям.
- Написание простого цикла обучения (градиентный спуск) для решения задачи XOR или классификации простых точек.

Задание:

Написать скрипт, который обучает однослойный перцептрон на синтетических данных без использования готовых слоев нейронных сетей. Зафиксировать изменение весов на каждой итерации и построить график сходимости ошибки.

Практическое занятие «Настройка окружения и работа с тензорами»

Ход занятия:

- Обзор экосистемы инструментов для глубокого обучения, создание изолированного окружения.
- Создание многомерных массивов (тензоров), проверка их свойств (размерность, тип данных, устройство).
- Практика операций Broadcasting (векторизованные операции над тензорами разной формы).
- Изучение механизма автоматического дифференцирования: построение вычислительного графа, замыкание градиентов.
- Сравнение производительности вычислений на CPU и GPU.

Задание:

Реализовать векторизованную операцию масштабирования и сдвига для большого набора данных (1 млн строк). Построить искусственный вычислительный граф и вручную вызвать обращение градиентов, проверив совпадение с численным дифференцированием.

Практическое занятие «Предобработка данных и EDA

Ход занятия:

- Загрузка стандартного набора данных (например, Fashion MNIST), анализ структуры файлов.
- Визуальный разведочный анализ: отображение случайных примеров, гистограммы распределения яркости пикселей.
- Анализ дисбаланса классов (построение диаграммы распределения меток).
- Нормализация данных (приведение к диапазону или стандартизация).[1]
- Реализация аугментации данных: геометрические преобразования (поворот, отражение, сдвиг) и их применение «на лету» или предварительно.

Задание:

Подготовить датасет для обучения: разделить на обучающую, валидационную и тестовую выборки (например, 60/20/20). Применить аугментацию к обучающей выборке (не менее 3 видов преобразований) и визуально сравнить оригиналы с полученными примерами.

Практическое занятие «Построение первой MLP-сети

Ход занятия:

- Концепция последовательной (стекловой) архитектуры модели.
- Определение слоев: входной (соответствие размерности данных), скрытый (выбор количества нейронов), выходной (соответствие количеству классов).
- Выбор функции активации для скрытых слоев (ReLU) и выходного (Softmax для классификации).
- Компиляция модели: выбор функции потерь (категориальная перекрестная энтропия), оптимизатора (Adam) и метрик.
- Запуск процесса обучения с разделением части данных для валидации.

Задание:

Собрать полносвязную нейронную сеть (MLP) для распознавания одежды. Обучить сеть на 25 эпох, записав историю обучения (значения потери и точности на обучающей и валидационной выборках).

Практическое занятие «Борьба с переобучением (Overfitting)

Ход занятия:

- Анализ графиков обучения: выявление расхождения между ошибками на тренировке и валидации.
- Внедрение регуляризации: Dropout (случайное отключение нейронов) и L2-регуляризация (штраф за большие веса).
- Настройка Early Stopping (остановка обучения при ухудшении метрики на валидации).
- Практический подбор гиперпараметров: размер мини-выборки (batch_size), количество эпох, скорость обучения (learning rate).
- Сравнение результатов «чистой» сети и сети с регуляризацией.

Задание:

Создать 3 версии одной и той же архитектуры: без регуляризации, с Dropout, с L2. Обнаружить переобучение в первой версии и устранить его во второй и третьей. Подобрать такие гиперпараметры, чтобы достичь максимума точности на валидации без падения качества.

Практическое занятие «Векторизация и оценка качества

Ход занятия:

- Преобразование вероятностного выхода сети (тензор) в дискретные метки классов (функция argmax).
- Расчет метрик качества: Accuracy, Precision, Recall, F1-score.
- Построение матрицы ошибок (Confusion Matrix) и интерпретация ложных срабатываний/пропусков.
- Сохранение обученной модели (архитектура + веса) в файл.
- Загрузка модели из файла и проверка её работоспособности на одном примере (инференс).

Задание:

Оценить итоговую модель на тестовой выборке. Сгенерировать и зафиксировать матрицу ошибок, выделив классы, которые модель путает чаще всего. Сохранить модель и записать скрипт, который загружает её и предсказывает класс для пользовательского изображения.

Практическое занятие «Основы CNN: Свертка и Пулинг

Ход занятия:

- Объяснение принципа локальности и общности весов в свертке.
- Реализация операции свертки (с нулевым дополнением и шагом) «с нуля» или через базовые примитивы.
- Построение архитектуры, чередующей слои свертки и максимального пулинга.

- Создание простой сети (типа LeNet) для классификации более сложного датасета (CIFAR-10).
- Визуализация карт признаков (feature maps) после первого и второго сверточного слоя.

Задание:

Обучить сверточную сеть на CIFAR-10. Сравнить скорость схождения и итоговую точность с полносвязной сетью (MLP) той же численности параметров. Визуализировать 4 карты признаков, чтобы понять, какие паттерны (края, углы) выучил первый слой.

Практическое занятие «Глубокие архитектуры и Transfer Learning

Ход занятия:

- Обзор современных архитектур (ResNet, EfficientNet, VGG) и их ключевых идей (остаточные связи).
- Загрузка предобученной модели (веса ImageNet) без выходного слоя.
- «Замораживание» (freeze) базовых слоев для сохранения выученных признаков.
- Добавление своей «головы» (классификатора) и дообучение (fine-tuning) только новых слоев.
- Сравнение качества обучения с нуля и использования трансферного обучения на малом датасете.

Задание:

Взять датасет из 100–500 изображений (свои фото кошек/собак или другой темы). Обучить классификатор с нуля (плохой результат) и с использованием Transfer Learning (ResNet50). Показать, что предобученная модель дает точность >80% даже на малых данных.

Практическое занятие «Распознавание объектов (Object Detection) с YOLO

Ход занятия:

- Разница между классификацией (что?) и детекцией (что и где?).
- Изучение метрики IoU (Intersection over Union) и формата аннотаций (bounding boxes).
- Запуск готовой предобученной модели YOLO (v8/v10) на тестовых изображениях и видео.
- Процесс сбора и разметки собственных данных (инструменты аннотирования).
- Дообучение детектора на специфическом классе объектов.

Задание:

Разметить 20–30 своих изображений с объектом (например, «шлем»). Дообучить модель YOLO на этих данных. Протестировать детектор на новых видео, зафиксировать количество ложных срабатываний и пропущенных объектов.

Практическое занятие «Семантическая сегментация (U-Net)

Ход занятия:

- Понятие сегментации (пиксельная классификация) и метрики IoU для сегментации.
- Архитектура U-Net: энкодер (сжатие), декодер (восстановление), пропускные связи (skip connections).
- Подготовка датасета для сегментации (изображение + маска).
- Сборка модели и выбор функции потерь (Dice Loss, CrossEntropy).
- Визуализация предсказанной маски поверх оригинала.

Задание:

Обучить U-Net на датасете медицинских снимков (сегментация опухолей) или спутниковых снимков (дороги). Вывести результат: исходное изображение, предсказанная маска и их наложение. Рассчитать метрику IoU на валидационной выборке.

Практическое занятие «Автокодировщики (Autoencoders) и сжатие данных

Ход занятия:

- Принцип работы автокодировщика: сжатие в «бутылочное горлышко» (bottleneck) и восстановление.
- Построение архитектуры с kendalaемым скрытым слоем.
- Обучение на задаче восстановления исходного изображения из зашумленного (Denoising).
- Визуализация скрытого представления (если размерность = 2).
- Сравнение качества восстановления (PSNR, MSSIM).

Задание:

Создать автокодировщик для Fashion MNIST. Обучить его восстанавливать изображения. Затем добавить шум к входным данным (до 50% пикселей) и показать, что сеть успешно удаляет шум и восстанавливает четкое изображение.

Практическое занятие «Введение в генеративные модели (VAE/GAN)

Ход занятия:

- Различие между дискриминативными и генеративными моделями.
- Архитектура Вариационного автокодировщика (VAE): кодирование в среднее и дисперсию, репараметризация.
- Обучение VAE и генерация новых образцов из случайного вектора в латентном пространстве.
- Обзор принципа работы GAN (генератор vs дискриминатор).
- Запуск готового примера GAN для генерации лиц.

Задание:

Обучить VAE на датасете одежды. Сгенерировать 10 новых, никогда не виденных сетью, изображений одежды. Визуализировать интерполяцию между двумя известными классами (например, плавно переходить от футболки к свитеру в латентном пространстве).

Практическое занятие «Векторизация текста и 1D-CNN

Ход занятия:

- Препроцессинг текста: токенизация, очистка от стоп-слов, стемминг/лемматизация.
- Методы векторизации: Bag of Words, TF-IDF и предобученные эмбеддинги (Word2Vec, GloVe).
- Построение архитектуры 1D-свертки для последовательностей (Conv1D, GlobalMaxPooling).
- Обучение модели для классификации тональности (позитив/негатив).
- Оценка качества и анализ ошибочных предсказаний.

Задание:

Подготовить датасет отзывов. Реализовать две модели: на основе TF-IDF + полносвязной сети и 1D-CNN на эмбеддингах. Сравнить скорость обучения и точность.

Практическое занятие «Рекуррентные сети (RNN, LSTM, GRU)

Ход занятия:

- Проблема последовательных данных и ограниченность MLP/CNN для контекста.
- Архитектура RNN, проблема затухающего градиента.
- Усовершенствованные ячейки: LSTM (долгая краткосрочная память) и GRU.
- Построение двунаправленной сети (Bidirectional) для учета контекста с обеих сторон.
- Задача генерации текста (предсказание следующего символа/слова).

Задание:

Обучить Bi-LSTM сеть для классификации тональности отзывов. Затем обучить простую рекуррентную сеть генерировать текст (например, строки стихов или код) на примере корпуса текстов и показать сгенерированную последовательность.

Практическое занятие «Механизм внимания (Attention) и Трансформер

Ход занятия:

- Ограничения рекуррентных сетей и идея механизма Self-Attention.
- Математика внимания: Query, Key, Value и softmax-взвешивание.
- Архитектура Transformer: позиционное кодирование, Multi-Head Attention, слои нормализации.
- Визуализация матриц внимания (какие слова «старались» друг друга).
- Простая задача машинного перевода или классификации на мини-трансформере.

Задание:

Реализовать (или собрать из готовых блоков) слой Multi-Head Attention. Визуализировать, на какие слова предложения модель обращает внимание при предсказании следующего слова. Выполнить простую задачу классификации текста, используя только механизм внимания.

Практическое занятие «Работа с предобученными моделями (Hugging Face)

Ход занятия:

- Концепция предобучения (Pre-training) и дообучения (Fine-tuning).
- Работа с токенизатором: превращение текста в IDs, Padding, Truncation.
- Загрузка предобученной модели (например, BERT/RuBERT) через высокоуровневый API.
- Использование готового pipeline для NER (извлечение имен, дат) и классификации.
- Анализ результатов работы модели на реальных текстах.

Задание:

Загрузить модель RuBERT. Написать скрипт, который извлекает все имена собственными и даты из произвольного новостного текста (NER) и определяет тональность.

Практическое занятие «Дообучение (Fine-tuning) BERT

Ход занятия:

- Подготовка датасета для конкретной узкой задачи (например, спам в чатах).
- Настройка «головы» классификатора поверх предобученного трансформера.
- Стратегии обучения: заморозка всех слоев кроме головы, затем постепенное размораживание.
- Использование специализированных оптимизаторов (AdamW) и расписание обучения (Linear with Warmup).
- Анализ влияния размера выборки на качество (pochi примеров vs много).

Задание:

Дообучить предобученную модель под задачу классификации спама. Разбить эксперимент: обучение на 50 примерах, 500, 5000. Показать, как растет качество и когда наступает насыщение.

Практическое занятие «Применение LLM и Prompt Engineering

Ход занятия:

- Обзор современных больших языковых моделей (LLM) и их возможностей.
- Основы промпт-инженерии: конкретика, примеры (few-shot), ограничения.
- Локальный запуск LLM или работа через API.
- Задачи: суммаризация, перевод, извлечение фактов, ролевая игра.

- Выявление и оценка галлюцинаций (выдуманных фактов).

Задание:

Составить 3 разных промпта для одной задачи (суммаризация длинной статьи). Запустить их в LLM, сравнить результаты, выделить ошибки и галлюцинации. Сформулировать «идеальный» промпт.

Практическое занятие «RAG (Retrieval-Augmented Generation) на практике

Ход занятия:

- Архитектура RAG: поиск + генерация.
- Векторизация документов (эмбеддинги) и хранение в векторной базе (FAISS/Chroma).
- Поиска релевантных фрагментов по запросу пользователя.
- Формирование контекста: «Ответь на вопрос, опираясь только на этот текст».
- Отправка контекста и вопроса в LLM получение ответа.

Задание:

Создать чат-бота, который отвечает на вопросы по учебному конспекту (текстовому файлу). Реализоватьзагрузку документа, его разбиение на части, создание векторного индекса и цикл: вопрос -> поиск -> контекст -> ответ LLM.

Практическое занятие «Эффективная донастройка LLM (LoRA/QLoRA)

Ход занятия:

- Проблема полной донастройки больших моделей (требует огромных ресурсов).
- Принцип LoRA (Low-Rank Adaptation): обучение только малых матриц при заморозке основной модели.
- Квантование (QLoRA): сжатие весов модели до 4 бит для работы на видеокарте потребителя.
- Подготовка датасета в формате «ввод-вывод» для настройки на новый стиль.
- Сохранение только весов адаптера (adapter) и их применение к базовой модели.

Задание:

Взять маленькую открытую LLM. Дообучить её с использованием LoRA на датасете из 50–100 примеров (например, научить отвечать в стиле конкретного персонажа). Сохранить адаптер, загрузить его к базовой модели и продемонстрировать изменение стиля ответа.

Критерии оценивания практических работ:

«Отлично»

- Код полностью работает, без ошибок, запускается «из коробки».
- Задание выполнено в полном объеме + присутствует расширенная часть (доп.

эксперимент, сравнение методов, визуализация).
- Графики/метрики построены корректно, интерпретация результатов глубока и аргументирована.
- Код чистый, структурированный, есть комментарии, имена переменных осмысленные.

«Хорошо»
- Код работает, но есть мелкие замечания (неоптимальная векторизация, лишние предупреждения).
- Задание выполнено полностью, но без расширенной части.
- Графики есть, но интерпретация поверхностна (описано «что», но не «почему»).
- Форматирование кода нарушено в отдельных местах.

«Удовлетворительно»
- Код работает с ошибками (нужна правка преподавателем) или дает результат, но с низким качеством.
- Выполнено только базовое ядро задания, пропущены ключевые этапы (например, нет валидации, нет сохранения модели).
- Метрики не рассчитаны или рассчитаны неверно.
- Отсутствие аналитического вывода (резюме) в отчете.

«Неудовлетворительно»
- Код не работает, синтаксические ошибки, не загружаются данные/модели.
- Задание выполнено менее чем на 50%.
- Плагиат (копирование чужого кода без понимания и адаптации).

5. Варианты тем индивидуального проекта

– Система обнаружения дефектов на производстве (Задача: Различать годные и бракованные изделия (металл, ткань, пластик) по фотографиям. Данные: Датасеты с поверхностными дефектами (например, MVTec AD, Kaggle). Архитектура: CNN с трансферным обучением (ResNet/EfficientNet) или U-Net для сегментации дефекта.

– Распознавание эмоций или возраст/пол по лицу (Задача: Классификация эмоций (радость, гнев, грусть, удивление) на фото лиц. Данные: FER-2013, AffectNet. Архитектура: CNN с аугментацией данных.)

– Распознавание жестов для управления интерфейсом (Задача: Классификация жестов рук (кому, стоп, палец вверх) в реальном времени.Данные: Custom dataset (свои видео) или Google Sign Language.Архитектура.: CNN + видео-аугментация.)

- Анализ тональности отзывов о товарах/услугах (Русский язык) (Задача: Классификация отзывов на «позитив», «негатив», «нейтрал».Данные: RuReviews, Ozon/Яндекс.Маркет (парсинг или open datasets).Архитектура.: Bi-LSTM или Fine-tuning RuBERT)

- Классификация новостных статей по темам (Задача: Автоматическое распределение новостей по категориям (Спорт, Политика, Технологии, Криминал).Данные: 20 Newsgroups (English) или русскоязычные новостные корпуса.Архитектура.: 1D-CNN,fastText, BERT)

- Система извлечения именованных сущностей (NER) из юридических текстов (Задача: Находить в договорах/статьях имена, организации, суммы, даты.Данные: ruNER, custom-разметка. Архитектура.: Fine-tuning RuBERT (BIO-разметка).

- Генератор коротких новостных заголовков (Задача: По тексту статьи сгенерировать краткий заголовок. Данные: Пара «текст-заголовок». Архитектура.: Sequence-to-Sequence (LSTM/Transformers), T5-small)

- Генерация стилизованных изображений (Задача: Перенос стиля картины (Ван Гог) на фото студента.Архитектура.: VGG-based loss minimization.)

- Сжатие изображений с помощью автокодировщиков (Задача: Сжать изображение в 50–100 раз и восстановить его с качеством, сравнимым с JPEG.Архитектура: Convolutional Autoencoder.)

Критерии оценивания индивидуального проекта:

- Отлично (90–100 баллов): Проект полностью соответствует требованиям, результат качественный, проведен глубокий анализ, код оптимизирован. Возможна работа с нестандартными данными или усложненная архитектура.

- Хорошо (75–89 баллов): Проект решен успешно, но есть мелкие недочеты в анализе, коде или визуализации. Точность модели хороша, но не идеальна.

- Удовлетворительно (50–74 балла): Проект решен, но с существенными упрощениями. Модель имеет низкую точность (но выше случайной), анализ поверхностный, код «сырой» (рабочий, но неструктурированный).

- Неудовлетворительно (<50 баллов): Проект не завершен, модель не обучается, код не запускается, результаты недостоверны, или обнаружен плагиат.