

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.10 Глубокое обучение

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 «Глубокое обучение» 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):

Е.В. Казаковцева, доцент кафедры анализа данных и искусственного интеллекта, к. ф.-м. н.

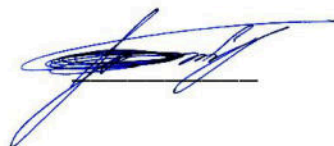


А.А. Кадури́н, доцент кафедры анализа данных и искусственного интеллекта



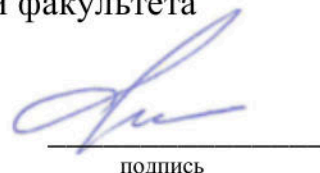
Рабочая программа дисциплины Глубокое обучение утверждена на заседании центра искусственного интеллекта протокол № 04 «14» мая 2026 г.

Руководитель центра ИИ Коваленко А.В.



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

Цели изучения дисциплины определены государственным образовательным стандартом высшего образования и соотнесены с общими целями ООП ВО по направлениям подготовки, в рамках которых преподается дисциплина.

Дисциплина посвящена овладению современными архитектурами и алгоритмами глубокого обучения, обеспечивающими эффективное решение широкого спектра задач компьютерного зрения, обработки естественного языка и синтеза/моделирования данных.

1.2 Задачи дисциплины

- Ознакомить студентов с основами глубокого обучения (архитектуры, типы задач, борьба с переобучением, техника transfer learning, fine-tuning нейронной сети и т.д.).
- Освоить создание, настройку и оценку моделей глубокого обучения на практике.
- Научить работать с библиотеками и фреймворками Keras, PyTorch и Tensorflow для целей глубокого обучения.
- Подготовить специалистов, способных применить глубокое обучение в профессиональных проектах и исследованиях.

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Глубокое обучение» относится к части, формируемой участниками образовательных отношений Блока 1 «Дисциплины (модули)» учебного плана. Данная дисциплина в качестве прerreквизитов требует освоения следующих дисциплин: «Математические модели нейронных сетей» (основные алгоритмы обучения, функции активации, оптимизаторы, начальные знания об интерпретируемости моделей, работа в jupyter notebook и google colab, библиотека scikit-learn), «Нейросетевые технологии» (аугментация, ансамблевые методы, автокодировщики, простые архитектуры для работы с текстами и изображениями, метрики качества, начальные сведения о работе tensorflow и keras), «Технологии компьютерного зрения» (начальные сведения о работе с изображениями, фильтрации и задачах в CV, работа с gitlab, библиотека opencv), «Математический анализ» (основы дифференциального исчисления, интегральное исчисление и ряды, теория функций нескольких переменных), «Обработка данных на Python» (основы Python, библиотеки numpy, pandas, matplotlib, seaborn), «Курс теории вероятностей» (распределения, матожидание, дисперсия и др. важные понятия), «Алгебра и геометрия» (операции с векторами и матрицами). Постреквизитами выступают дисциплины, где продолжается применение принципов глубокого обучения: «Технологии обработки языка, звуковых данных, включая распознавание и синтез речи» (архитектуры для работы с текстами и аудио через SpeechRecognition, torchaudio, librosa, Hugging Face Transformers, FastSpeech), «Генеративный искусственный интеллект» (более глубокое освоение трансформеров и GAN архитектур, работа с несколькими модальностями). Смежными дисциплинами, изучаемыми параллельно или логически близко, являются «MLOps&DevOps» (внедрение нейросетей), «Технологии обработки больших данных» (работа с большими датасетами), «Рекомендательные системы» (нейросети в задачах предоставления рекомендаций).

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-1.3 Способен применять современные архитектуры глубоких сетей для решения различных задач, понимая их внутреннюю структуру и особенности обучения.	(С) Применяет принцип построения вычислительного блока Google Inception; Применяет принцип работы остаточного блока в ResNet; разрабатывает решения с применением базовых (backbone) сетей; знает отличия и способен применять нейронные сети для отслеживания объектов (семейство R-CNN, YOLO)
	DL-1.5 Разрабатывает и оптимизирует специализированные архитектуры для работы с последовательностями, учитывая их уникальные	(С) Понимает принцип работы блока единичной задержки в рекуррентных нейронных сетях; применяет блоки RNN, GRU, LSTM для решения задач
	DL-1.6 Разрабатывает, адаптирует и внедряет генеративные нейронные сети для решения практических задач, включая создание новых архитектур, оптимизацию обучения и промышленное развертывание моделей	(С) Модифицирует архитектуры под специфические требования; разрабатывает комплексные пайплайны обучения; применяет продвинутое техники - перенос обучения, генерация по нескольким примерам

	DL-1.7 Разрабатывает, оптимизирует и применяет автоэнкодеры (AE) и вариационные автоэнкодеры (VAE) для решения задач снижения размерности, генерации данных и обнаружения аномалий, включая создание архитектур, обучение моделей и их внедрение в продуктивную среду	(C) Применяет различные автокодировщики для решения задач; настраивает сложные архитектуры (глубокие AE, условные VAE); оптимизирует функции потерь, такие как комбинация СКО + KL-дивергенция. Работает с разными типами данных (изображения, временные ряды, графы).
	DL-1.9 Разрабатывает, адаптирует и внедряет трансформерные архитектуры для решения задач обработки последовательностей, включая создание новых моделей, оптимизацию обучения и промышленное развертывание	(C) Понимает и применяет специфические архитектурные варианты для задач (например, вариант "энкодер-декодер" для перевода/суммаризации, вариант "только энкодер" для генерации); использует для работы с длинными последовательностями эффективные схемы внимания (Linformer, Performer, Longformer, Sparse Transformers). Проводит детальный анализ ошибок модели, понимает типичные проблемы (смещение в данных/модели, переобучение) и применяет методы борьбы с ними (регуляризация, аугментация данных)
DL-2 Способен применять и (или) разрабатывать современные архитектуры генеративных глубоких сетей	DL-2.1 Применяет известные архитектуры генеративных глубоких нейронных сетей для решения прикладной задачи (генерация текста, генерация изображений по тексту, синтез речи и т.д.), при необходимости проводя дообучение на наборах данных	(C) Настраивает параметры генерации под конкретную задачу. Реализует техники контролируемой генерации (CFG Scale, промпт-инжиниринг). Комбинирует несколько моделей в пайплайн. Проводит доменную адаптацию для специфических данных. Использует продвинутое методы эффективного обучения (QLoRA, DreamBooth). Оптимизирует процесс обучения (подбор скорости обучения и пакетов).
	DL-2.2 Имплементирует известные архитектуры генеративных сетей (VAE, GAN, диффузионные модели)	(C) Кастомизирует архитектуры под задачи (изменение латентного пространства в GAN). Реализует сложные пайплайны обучения (многостадийное

		обучение). Применяет продвинутое техники дообучения (LoRA, DreamBooth). Настраивает распределенное обучение (DDP, DeepSpeed). Оптимизирует память и скорость обучения (контрольные точки градиента).
DL-3 Способен применять и (или) разрабатывать алгоритмы, методы и технологии компьютерного зрения	DL-3.1 Применяет (проводя выбор и эксперименты) известные алгоритмы и библиотеки компьютерного зрения, предобученные глубокие нейросетевые модели для прикладных задач анализа изображений и видеопотока, при необходимости дообучая и валидируя на собственных наборах данных	(С) Сравнивает разные предобученные модели под конкретную задачу. Проводит перенос обучения на своих данных. Оптимизирует гиперпараметры для улучшения качества. Создает сложные пайплайны аугментации (с использованием библиотеки Albumentations). Умеет работать с видео: извлечение кадров, обработка временных последовательностей путём применения CNN+RNN, 3D CNN.
LLM-2 Способен дообучать, адаптировать и оптимизировать генеративные модели под специфические задачи и условия применения	LLM-2.1 Понимает принципы дообучения	(С) Применяет дообучение к предобученным моделям на новых датасетах
	LLM-2.2 Создаёт обучающие наборы данных	(С) Адаптирует и валидирует датасеты под разные типы задач

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 2 зач. ед. (72 часов), их распределение по видам работ представлено в таблице

Вид учебной работы	Всего часов	Семестры (часы)					
		6					
Контактная работа, в том числе:	50.2	50.2					
Аудиторные занятия (всего):	48	48					
Занятия лекционного типа	16	16					
Лабораторные занятия	32	32					
Занятия семинарского типа (семинары, практические занятия)							
Иная контактная работа:	0.2	0.2					
Контроль самостоятельной работы (КСР)	2	2					
Промежуточная аттестация (ИКР)							
Самостоятельная работа, в том числе:	21.8	21.8					
Курсовая работа	-	-					
Проработка учебного (теоретического) материала							

Выполнение индивидуальных заданий (подготовка сообщений, презентаций)							
Реферат							
Подготовка к текущему контролю							
Контроль:							
Подготовка к экзамену		-	-				
Общая трудоемкость	час.	72	72				
	в том числе контактная работа	50.2	50.2				
	зач. Ед	2	2				

2.2 Структура дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

Разделы (темы) дисциплины, изучаемые в 6 семестре

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	СРС
1	2	3	4	5	6	7
1.	Введение в глубокое обучение	3	2			1
2.	Архитектуры глубоких нейронных сетей	11	4		4	3
3.	Алгоритмы обучения глубоких сетей	12,8	4		6	2,8
4.	Глубокие сети в компьютерном зрении	20	2		12	6
5.	Глубокие сети в обработке естественного языка	9	2		4	3
6.	Оптимизация и повышение производительности	14	2		6	6
ИТОГО по разделам дисциплины		69,8	16		32	21.8
Контроль самостоятельной работы (КСР)		2				
Промежуточная аттестация (ИКР)		0.2				
Подготовка к текущему контролю						
Общая трудоемкость по дисциплине		72				

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 текущего контроля
1	2	3	4
1.	Введение в глубокое обучение	История и предпосылки появления глубокого обучения. Сравнение с традиционным машинным обучением.	УО
2.	Архитектуры глубоких нейронных сетей	Введение в многослойные нейронные сети. Особенности глубоких сетей. Свёрточные нейронные сети (CNN): архитектура, слои свёртки и пулинга.	ЛР

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		Рекуррентные нейронные сети (RNN): долгосрочная память, GRU, LSTM, модификация LSTM со смотровыми глазками. Обратное распространение ошибки в рекуррентных сетях. Сети трансформеров (Transformer Architecture): само-внимание, энкодеры и декодеры. Капсульные нейронные сети (CN). Сеть Хопфилда.	
3.	Алгоритмы обучения глубоких сетей	Подбор оптимизатора для глубокой нейросети. Выбор коэффициента скорости обучения (Learning Rate). Проблема исчезающего и взрывающегося градиента. Способы решения. Пакетное обучение (Mini-batch training) и его влияние на качество модели. Гиперпараметры: подбор оптимальной архитектуры и настройка. Валидация и cross validation: стратегия оценки качества моделей в глубоких сетях. Проблемы переобучения и недообучения. Методы их устранения (L1/L2 регуляризация, dropout, batch-normalization, early stopping). Переобучение и сдвиг данных (overfitting, domain shift), регуляризация, настройка скорости обучения и планировщика (learning rate, scheduler), дообучение и предварительное обучение, смешанное обучение Трансферное обучение и fine-tuning моделей. Задача уменьшения размерности. Дивергенция Кульбака-Лейблера в качестве функции потерь. Дообучение vs предварительное обучение, формирование обучающих выборок под сдвиги домена, регуляризация и оценка устойчивости, оценка метрик до и после обучения (потери, перплексия, BLEU, ROUGE и т.д.)	ЛР
4.	Глубокие сети в компьютерном зрении	Задачи классификации изображений: архитектура и примеры (VGG, ResNet, Xception, DenseNet, MobileNet). Остаточные блоки Задачи детекции изображений: архитектуры и примеры (Region Proposal Networks (RPNs), SSD, YOLO).	ЛР

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		Задачи сегментации: UNet, SegNet, Mask RCNN. Генеративные нейронные сети. Генеративно-состязательные архитектуры. Вариационные автокодировщики. Капсульные архитектуры в задачах генерации. Проблема коллапса моделей Современное состояние компьютерного зрения: перспективные подходы и будущее. Визуальные трансформеры. Генерация изображений по тексту и текста по изображению.	
5.	Глубокие сети в обработке естественного языка	Классификация текстов с помощью глубоких нейронных сетей. Трансформеры. Механизм внимания, многоголовое внимание. Оптимизация и модификация архитектуры трансформера. Адаптация и тонкая настройка. Предобученные модели на базе трансформера. Генерация изображений по тексту. Генерация изображений по тексту. Визуальное распознавание с генерацией языка, мультимодальное обучение. Архитектура энкодер-декодер, внимание "изображение-текст" Генерация текста по изображениям. Модели ГНС-трансформер, архитектуры с механизмом внимания. Контроль стиля и тона. Метрики качества описаний: BLEU, ROUGE.	ЛР
6.	Оптимизация и повышение производительности	Параллельные вычисления и распределенное обучение глубоких нейронных сетей. Оптимизация вычислений: Quantization, Pruning, Knowledge Distillation (Мастер-класс от Кадурина А.А.) Специализированные аппаратные решения для глубокого обучения (GPUs, TPUs, ASICs). Производительность нейронных сетей: оптимизация памяти и задержек. Инновации в hardware-accelerated computing для deep learning (Мастер-класс от Кадурина А.А.)	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, УО – Устный опрос.

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 глубоких нейронных сетей	Лабораторная работа №1: Реализация архитектуры LeNet Лабораторная работа №2: Построение свёрточной нейронной сети для классификации изображений Fashion-MNIST	ЛР
3.	Алгоритмы обучения глубоких сетей	Лабораторная работа №3: Подбор оптимизатора для глубокой нейросети Лабораторная работа №4: Исследование методов борьбы с переобучением. Лабораторная работа №5: Метод grid-search и ранняя остановка модели	ЛР
4.	Глубокие сети в компьютерном зрении	Лабораторная работа №6: Разработка нейросети VGG-16 для классификации изображений Лабораторная работа №7: Детекция объектов с помощью YOLO. Лабораторная работа №8: Сегментация сети с помощью Unet. Лабораторная работа №9: Перенос обучения и fine-tuning моделей Лабораторная работа №10 Визуальный трансформер	ЛР
5.	Глубокие сети в обработке естественного языка	Лабораторная работа №11: Применение трансформера для анализа тональности текстов	ЛР
6.	Оптимизация и повышение производительности	Лабораторная работа №12: Параллельное обучение глубоких нейросетей	ЛР

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
		Лабораторная работа №13: Исследование методов прунинга и дистилляции	

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:

- Мастер-класс – предполагает разбор тем от ведущих специалистов в области (Кадуриной А.А.).

- 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
	ЛР	Практические занятия в режимах взаимодействия «преподаватель – студент» и «студент – студент»	12
Итого			12

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 «Глубокое обучение».

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

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 зачете;
- 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 глубокое обучение	DL-1.3 (Средний)	Тестирование	Вопросы к зачету 1-5
2	Архитектуры глубоких нейронных сетей	DL-1.3 (Средний), DL-1.5 (Средний), DL-2.1 (Средний)	Тестирование, Лабораторные работы №1,2	Вопросы к зачету 6-9, 30
3	Алгоритмы обучения глубоких сетей	DL-2.1 (Средний), DL-1.7 (Средний), LLM-2.1 (Средний), LLM-2.2 (Средний)	Тестирование, Лабораторная работа №3-5	Вопросы к зачету 10-14, 31-36
4	Глубокие сети в компьютерном зрении	DL-1.3 (Средний), DL-1.6 (Средний), DL-1.7 (Средний), DL-2.1 (Средний), DL-3.1 (Средний)	Тестирование, Лабораторные работы №6-10	Вопросы к зачету 15-19
5	Глубокие сети в обработке естественного языка	DL-1.9 (Средний), DL-2.1 (Средний), DL-2.2 (Средний)	Тестирование, Лабораторная работа №11	Вопросы к зачету 20-24, 37-40
6	Оптимизация и повышение производительности	DL-3.1 (Средний)	Тестирование, Лабораторные работы №12,13	Вопросы к зачету 25-29

Показатели, критерии и шкала оценки сформированных компетенций

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

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

Применяет принцип построения вычислительного блока Google Inception; Применяет принцип работы остаточного блока в ResNet; разрабатывает решения с применением базовых (backbone) сетей; знает отличия и способен применять нейронные сети для отслеживания объектов (семейство R-CNN, YOLO)

Понимает принцип работы блока единичной задержки в рекуррентных нейронных сетях; применяет блоки RNN, GRU, LSTM для решения задач

Модифицирует архитектуры под специфические требования; разрабатывает комплексные пайплайны обучения; применяет продвинутое техники - перенос обучения, генерация по нескольким примерам

Применяет различные автокодировщики для решения задач; настраивает сложные архитектуры (глубокие AE, условные VAE); оптимизирует функции потерь, такие как комбинация СКО + KL-дивергенция. Работает с разными типами данных (изображения, временные ряды, графы).

Понимает и применяет специфические архитектурные варианты для задач (например, вариант "энкодер-декодер" для перевода/суммаризации, вариант "только энкодер" для генерации); использует для работы с длинными последовательностями эффективные схемы внимания (Linformer, Performer, Longformer, Sparse Transformers). Проводит детальный анализ ошибок модели, понимает типичные проблемы (смещение в данных/модели, переобучение) и применяет методы борьбы с ними (регуляризация, аугментация данных)

DL-2 *Способен применять и (или) разрабатывать современные архитектуры генеративных глубоких сетей*

Настраивает параметры генерации под конкретную задачу. Реализует техники контролируемой генерации (CFG Scale, промпт-инжиниринг). Комбинирует несколько моделей в пайплайн. Проводит доменную адаптацию для специфических данных. Использует продвинутое методы эффективного обучения (QLoRA, DreamBooth). Оптимизирует процесс обучения (подбор скорости обучения и пакетов).

Кастомизирует архитектуры под задачи (изменение латентного пространства в GAN). Реализует сложные пайплайны обучения (многостадийное обучение). Применяет продвинутое техники дообучения (LoRA, DreamBooth). Настраивает распределенное обучение (DDP, DeepSpeed). Оптимизирует память и скорость обучения (контрольные точки градиента).

DL-3 *Способен применять и (или) разрабатывать алгоритмы, методы и технологии компьютерного зрения*

Сравнивает разные Сравнивает разные предобученные модели под конкретную задачу. Проводит перенос обучения на своих данных. Оптимизирует гиперпараметры для улучшения качества. Создает сложные пайплайны аугментации (с использованием библиотеки Albumentations). Умеет работать с видео: извлечение кадров, обработка временных последовательностей путём применения CNN+RNN, 3D CNN.

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

Применяет дообучение к предобученным моделям на новых датасетах

Адаптирует и валидирует датасеты под разные типы задач

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

Практические кейсы по тематике лабораторных работ

Лабораторная работа №1. Реализация архитектуры LeNet

Кейс: Распознавание рукописных цифр на почте.

Постановка задачи: Почтовая служба получает огромное количество писем с адресами, написанными от руки. Необходимо разработать нейросеть, способную автоматически считывать адреса, распознавая цифры почтового индекса.

Требования:

- Реализовать архитектуру LeNet с точным соблюдением параметров, описанных в оригинальной статье.
- Обучить модель на датасете MNIST и достигнуть точности не ниже 98%.
- Оценить влияние изменения количества слоёв и фильтров на производительность модели.

Лабораторная работа №2. Построение свёрточной нейронной сети для классификации изображений Fashion-MNIST

Кейс: Система автоматической сортировки гардероба.

Постановка задачи: Компания производит интерактивные зеркала для магазинов одежды, позволяющие покупателям получать персональные рекомендации по стилю. Необходимо обучить CNN-классификатор, который сможет определить категорию предмета одежды (рубашка, туфли, брюки и т.д.) по изображению.

Требования:

- Создать и обучить CNN для классификации Fashion-MNIST.
- Достичь точности классификации не ниже 90%.
- Провести анализ влияния добавления новых слоёв и изменения размера фильтра на производительность.

Лабораторная работа №3. Подбор оптимизатора для глубокой нейросети

Кейс: Улучшение распознавания автомобильных номеров.

Постановка задачи: Городская администрация внедряет систему видеонаблюдения для регистрации транспортных средств. Необходимо подобрать оптимальный оптимизатор для нейросети, распознающей автомобильные номера, чтобы сократить время обучения и повысить точность распознавания.

Требования:

- Обучить нейросеть с тремя разными оптимизаторами (Adam, SGD, RMSProp).
- Сравнить производительность по скорости обучения и качеству результата.
- Сделать вывод о самом эффективном оптимизаторе для данной задачи.

Лабораторная работа №4. Исследование методов борьбы с переобучением

Кейс: Автоматическая классификация пищевых продуктов.

Постановка задачи: Производитель продуктов питания создаёт автоматизированную систему маркировки упаковок. Необходимо исследовать методы

борьбы с переобучением, чтобы повысить точность классификации продуктов по изображениям.

Требования:

- Обучить CNN на датасете Food-101 без методов борьбы с переобучением.
- Применить Dropout, BatchNormalization и регуляризацию.
- Сравнить результаты до и после внедрения методов.

Лабораторная работа №5. Метод grid-search и ранняя остановка модели

Кейс: Оптимизация параметров нейросети для прогнозирования спроса на товары.

Постановка задачи: Магазин электроники хочет прогнозировать спрос на товары с помощью нейросети. Необходимо оптимизировать гиперпараметры (learning rate, batch size, dropout ratio) и включить early-stopping для повышения точности прогнозов.

Требования:

- Использовать grid-search для поиска оптимальных гиперпараметров.
- Реализовать early-stopping для предотвращения переобучения.
- Оценить влияние каждого гиперпараметра на производительность модели.

Лабораторная работа №6. Разработка нейросети VGG-16 для классификации изображений

Кейс: Анализ ассортимента мебельного магазина.

Постановка задачи: Онлайн-магазин мебели реализует систему автоматического каталога, которая должна автоматически классифицировать предметы интерьера по стилям (классический, современный, хай-тек и т.д.).

Требования:

- Реализовать и обучить нейросеть VGG-16 на своём собственном датасете мебели.
- Обеспечить точность классификации не ниже 90%.
- Оценить эффективность VGG-16 по сравнению с другими CNN-архитектурами.

Лабораторная работа №7. Детекция объектов с помощью YOLO

Кейс: Безопасность дорожного движения.

Постановка задачи: Местные власти хотят повысить безопасность на дорогах, устанавливая камеры, способные автоматически регистрировать нарушения правил дорожного движения. Необходимо обучить YOLO для детекции пешеходов и велосипедистов на дороге.

Требования:

- Подготовить собственный датасет изображений дорог.
- Обучить YOLO v3/v5 для детекции объектов.
- Оценить точность детекции с помощью метрик AP и IoU.

Лабораторная работа №8. Сегментация сети с помощью UNet

Кейс: Медицинская диагностика.

Постановка задачи: Врачи нуждаются в инструментах для точного выделения опухолей головного мозга на МРТ-изображениях. Необходимо разработать UNet-сеть для сегментации опухоли на МРТ.

Требования:

- Собрать и разметить датасет MRI-изображений.
- Реализовать и обучить UNet.
- Оценить качество сегментации с помощью Dice Score и IoU.

Лабораторная работа №9. Перенос обучения и fine-tuning моделей

Кейс: Автоматизация ветеринарной клиники.

Постановка задачи: Ветклиника хочет внедрить систему идентификации пород собак и кошек по изображениям. Необходимо адаптировать готовую модель ResNet-50 для этой задачи.

Требования:

- Произвести transfer-learning и fine-tune модели ResNet-50 на новом датасете.
- Оценить точность классификации.
- Сравнить производительность до и после fine-tuning.

Лабораторная работа №10. Визуальные трансформеры

Кейс: Идентификация редких растений в ботаническом саду

Постановка задачи: Крупнейший ботанический сад страны столкнулся с необходимостью систематизации огромного массива фотоснимков редких растений. Специалисты сада собирают фотоматериалы редких представителей флоры, однако многие растения трудно отличить друг от друга визуально. Учреждению необходимо реализовать систему идентификации растений на основе фотографий, которая будет автоматически определять принадлежность растения к соответствующему виду.

Требования:

- Произведите transfer-learning предварительно обученной Vision Transformer (ViT) модели на собственном наборе изображений редких растений.
- Реализуйте fine-tuning ViT на собранном датасете.
- Оцените точность классификации на тестовом наборе данных.
- Сравните производительность модели до и после fine-tuning.
- Подготовьте отчёт с рекомендациями по дальнейшему развитию проекта.

Лабораторная работа №11. Применение трансформера для анализа тональности текстов

Кейс: Мониторинг общественного мнения в социальных сетях.

Постановка задачи: Компания хочет отслеживать общественное мнение о своих продуктах в соцсетях. Необходимо реализовать трансформерную модель для анализа тональности русскоязычных сообщений.

Требования:

- Обучить трансформер BERT/RUBERT на датасете тональности.
- Оценить точность модели на независимых тестах.
- Проверить способность модели воспринимать нюансы русского языка.

Лабораторная работа №12. Параллельное обучение глубоких нейросетей

Кейс: Повышение скорости обучения нейросети для анализа спутниковых снимков.

Постановка задачи: Геологическая компания проводит анализ земельных участков с помощью спутниковых снимков. Необходимо организовать параллельное обучение нейросети на нескольких GPU для ускорения процесса.

Требования:

- Реализовать distributed training на нескольких GPU с использованием PyTorch/DDP.
- Оценить выигрыш в скорости обучения и экономичности ресурса.
- Проанализировать влияние увеличения количества GPU на качество обучения.

Лабораторная работа №13. Исследование методов прунинга и дистилляции

Кейс: Оптимизация мобильного приложения для распознавания растений.

Постановка задачи: Ботаническое мобильное приложение, предназначенное для распознавания растений, стало занимать слишком много памяти смартфона. Необходимо

применить методы прунинга и дистилляцию для оптимизации модели, уменьшив её размер без значительной потери точности.

Требования:

- Применить прунинг для уменьшения размера модели MobileNet.
- Использовать knowledge distillation для переноса знаний от полной модели к лёгкой.
- Оценить разницу в размере модели и сохранить падение точности не более 3%.

Пример лабораторной работы

Лабораторная работа №1: Реализация архитектуры LeNet

Цель: Разработать классическую архитектуру LeNet-5 для задачи классификации рукописных цифр (MNIST).

Задачи:

- Изучить задачу классификации изображений и способы её решения;
- Загрузить датасет MNIST;
- Разработать свёрточную нейросеть с помощью фреймворка pyTorch.

Ожидаемые результаты:

- Знает принципы разработки свёрточной архитектуры;
- Умеет разрабатывать свёрточные нейросети с помощью pyTorch;
- Умеет оценивать качество разработанной нейросети.

Инструменты и библиотеки:

- Язык программирования Python
- Jupyter Notebook / Google Colab
- фреймворк pyTorch.

Исходные данные: датасет рукописных цифр (MNIST).

Ход работы:

1. Установка библиотек

```
pip install torch torchvision
import torch
import torchvision
from torchvision import datasets, transforms
```

2. Загрузка и подготовка данных:

```
transform = transforms.Compose([transforms.ToTensor()])
trainset = datasets.MNIST('./data', train=True, download=True, transform=transform)
testset = datasets.MNIST('./data', train=False, download=True, transform=transform)
trainloader = torch.utils.data.DataLoader(trainset, batch_size=64, shuffle=True)
testloader = torch.utils.data.DataLoader(testset, batch_size=64, shuffle=False)
```

3. Реализация LeNet-5:

```
import torch.nn as nn
import torch.nn.functional as F
class LeNet(nn.Module):
    def __init__(self):
        ...
    def forward(self, x):
        ...
```

4. Обучение и оценка модели:

```
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
```

```

model = LeNet().to(device)
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.SGD(model.parameters(), lr=0.001, momentum=0.9)

for epoch in range(10):
    running_loss = 0.0
    for i, data in enumerate(trainloader, 0):
        inputs, labels = data[0].to(device), data[1].to(device)
        optimizer.zero_grad()
        outputs = model(inputs)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
        running_loss += loss.item()
    print(f"[{epoch + 1}/10] Loss: {running_loss / len(trainloader)}")

```

5. Тестирование модели

```

correct = 0
total = 0
with torch.no_grad():
    for data in testloader:
        images, labels = data[0].to(device), data[1].to(device)
        outputs = model(images)
        _, predicted = torch.max(outputs.data, 1)
        total += labels.size(0)
        correct += (predicted == labels).sum().item()

print(f"Точность на тестовом наборе: {100 * correct / total:.2f}%")

```

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, функции потерь и функций активации.

4. Реализация

Код на Python с комментариями.

Примеры до/после обработки.

5. Результаты

Метрики

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 подсчетом метрик и функции потерь.

Хорошо: Корректный код, но без анализа полученных результатов.

Удовлетворительно: Частичное выполнение с ошибками на этапах оценки и тестирования модели.

Неудовлетворительно: Невыполнение ключевых этапов.

Проверяемые компетенции комплексом практических заданий: DL-1.3

Зачетно-экзаменационные материалы для промежуточной аттестации (зачет)

Вопросы для подготовки к зачету

1. Чем глубокое обучение принципиально отличается от обычного машинного обучения?
2. Какие основные компоненты входят в архитектуру нейронной сети?
3. Какова роль функций активации в нейронных сетях?
4. Что такое прямой и обратный проход в обучении нейронных сетей?
5. В чем заключаются этапы обучения нейронных сетей?
6. Что такое сверточные нейронные сети (CNN), и какие задачи они решают?
7. В чем главное отличие рекуррентных нейронных сетей (RNN) от CNN?
8. Какие существуют разновидности RNN (GRU, LSTM), и как они улучшают запоминание длинных последовательностей?
9. Чем характеризуются сети трансформеров (Transformers), и почему они стали популярны в NLP?
10. Какие оптимизаторы чаще всего используются при обучении глубоких нейронных сетей?
11. Что такое проблема "взрыва" и "исчезновения" градиента, и как с ними бороться?
12. В чем заключается регуляризация, и какие методы регуляризации существуют?
13. В чем польза пакетного обучения (mini-batch training) и как его правильно настроить?
14. Как подбирать оптимальное значение скорости обучения (learning rate)?
15. Какие общие задачи решаются в области компьютерного зрения?
16. Как устроены свёрточные нейронные сети (CNN), и какие у них слои?
17. Что такое object detection и какие существуют методы (YOLO, SSD)?
18. В чем отличие semantic segmentation от instance segmentation?
19. Какие современные архитектуры используются для задач классификации изображений (DenseNet, VGG, ResNet)?
20. Что такое обработка естественного языка (NLP), и какие задачи она решает?
21. Какие возможности открывают архитектуры на основе трансформеров?
22. Почему внимание (attention mechanisms) стало ключевой частью современных моделей NLP?
23. Популярные нейросети для распознавания и генерации речи.
24. Современные тренды в NLP: мультимодальность, персонализация, новые архитектуры.
25. Какие аппаратные устройства ускоряют обучение нейронных сетей (GPU, TPU)?

26. В чем заключается quantization, и как оно сокращает потребление памяти и увеличивает производительность?
27. Какие подходы позволяют снижать сложность модели без значительного ухудшения качества (pruning, distillation)?
28. Как распределяют нагрузку при параллельном обучении нейронных сетей?
29. Какие специальные приемы повышают эффективность работы нейронных сетей на мобильных устройствах?
30. Капсульные нейронные сети (CN) и сети Хопфилда.
31. Валидация и cross validation: стратегия оценки качества моделей в глубоких сетях.
32. Проблемы переобучения и недо-обучения. Методы их устранения (L1/L2 регуляризация, dropout, batch-normalization, early stopping).
33. Переобучение и сдвиг данных (overfitting, domain shift), регуляризация, настройка скорости обучения и планировщика (learning rate, scheduler), дообучение и предварительное обучение, смешанное обучение
34. Трансферное обучение и fine-tuning моделей.
35. Задача уменьшения размерности. Дивергенция Кульбака-Лейблера в качестве функции потерь.
36. Дообучение vs предварительное обучение, формирование обучающих выборок под сдвиги домена, регуляризация и оценка устойчивости, оценка метрик до и после обучения (потери, перплексия, BLEU, ROUGE и т.д.)
37. Генеративные нейронные сети. Генеративно-состязательные архитектуры. Вариационные автокодировщики. Капсульные архитектуры в задачах генерации. Проблема коллапса моделей
38. Визуальные трансформеры. Генерация изображений по тексту и текста по изображению.
39. Генерация изображений по тексту. Генерация изображений по тексту. Визуальное распознавание с генерацией языка, мультимодальное обучение. Архитектура энкодер-декодер, внимание "изображение-текст"
40. Генерация текста по изображениям. Модели ГНС-трансформер, архитектуры с механизмом внимания. Контроль стиля и тона. Метрики качества описаний: BLEU, ROUGE.

Перечень компетенций (части компетенции), проверяемых оценочным средством
DL-1.3; DL-1.5; DL-1.6; DL-1.7; DL-1.9; DL-2.1; DL-2.2; DL-3.1; LLM-2.1; LLM-2.2.

Практические задания к зачету

1. Сравнение CNN и обычного ML
Сравните производительность простой свёрточной нейронной сети (CNN) и традиционного классификатора SVM на датасете MNIST. Измерьте точность и время обучения.
2. Forward и Backward Pass
Пройдите вручную один шаг прямого прохода (forward pass) и обратного прохода (backward pass) для нейронной сети с одним скрытым слоем (10 нейронов). Данные берём из первых записей датасета MNIST.
3. Optimiser Choice
Реализуйте оптимизатор Adam для обучения линейной регрессии на датасете Boston Housing. Сравните его с SGD по скорости сходимости и точности.

4. Проблема vanishing gradients

Объясните, почему исчезает градиент в глубокой нейронной сети и как это проявляется на примере простого RNN. Предложите и реализуйте решение (например, заменить на LSTM).

5. Object Detection

Используя готовый обученный YOLO, прогоните его на одном изображении из датасета Pascal VOC и отметьте обнаруженные объекты рамками прямо на картинке.

6. Semantic Segmentation

Используя UNet, сегментировать клетки на изображении из маленького датасета (например, MICCAI Cells). Показать результат наложением маски на картинку.

7. Transfer Learning

Обучите последнюю полностью подключённую (fully connected) часть модели ResNet-50 на датасете CIFAR-10. Измерьте точность до и после обучения.

8. Attention Mechanisms

Используя трансформер (например, DistilBERT), напишите простой код для классификации положительных и отрицательных отзывов (датасет IMDB). Измерьте точность.

9. GPU Acceleration

Обучите простую нейросеть на датасете MNIST сначала на CPU, потом на GPU (через CUDA). Измерьте разницу во времени обучения.

10. Quantization

Проведите квантификацию (quantization) готовой модели MNIST и оцените её производительность на маленьком мобильном устройстве (например, Android Emulator).

11. Pruning нейросети

Удалите лишние соединения в нейросети MNIST, проведя manual pruning, и измерьте влияние на точность и размер модели.

12. Мини-батч обучение (Mini-batch training)

Проведите эксперимент с размером батчей (batch sizes) на датасете MNIST (например, 16, 32, 64, 128). Оцените влияние на скорость обучения и точность.

13. Настройка скорости обучения (Learning Rate)

Протестируйте постоянную и экспоненциально убывающую скорость обучения (learning rates) на простой нейросети MNIST. Оцените, как это влияет на качество и скорость обучения.

14. NLP Task

Используя fastText, проведите анализ тональности текста (Sentiment Analysis) на датасете TinyTwitter (меньше 1000 записей). Измерьте точность.

15. Параллельное обучение (Parallel Training)

Создайте простую нейросеть MNIST и запустите её обучение на двух ядрах CPU параллельно. Оцените разницу во времени обучения по сравнению с одиночным потоком.

Перечень компетенций (части компетенции), проверяемых оценочным средством
DL-1.3; DL-1.5; DL-1.6; DL-1.7; DL-1.9; DL-2.1; DL-2.2; DL-3.1; LLM-2.1; LLM-2.2.

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

Методические рекомендации, определяющие процедуры оценивания на зачете:

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 зачет. Студенты обязаны сдать зачет в соответствии с расписанием и учебным планом.

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

Зачет 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 зачета: устно.

Результат сдачи зачета заносится преподавателем в зачетную ведомость и зачетную книжку.

Оценивание уровня освоения дисциплины основывается на качестве выполнения студентом заданий текущего контроля и ответов на вопросы зачета.

Критерии оценки:

1. Оценка ответов на вопросы к зачету (50% итоговой оценки)

Зачет

Полные, развернутые ответы с демонстрацией глубокого понимания темы.

Ответы содержат основные идеи, но без углубленного анализа.

Использование примеров, формул, корректных терминов.

Возможны небольшие ошибки в деталях или формулировках.

Умение анализировать и сравнивать методы.

% выполнения: 60–100% (допускаются незначительные неточности).

Незачет

Отсутствие понимания ключевых концепций.

Грубые ошибки или неспособность ответить на большую часть вопросов.

% выполнения: <60%.

2. Оценка выполнения практических кейсов и лабораторных работ (50% итоговой оценки)

Зачет

Полное выполнение всех этапов кейса с инновационными решениями.

Достижение целевых метрик (например, $F1 > 0.9$).

Четкая документация кода и анализ результатов.

% выполнения: 60–100%.

или

Выполнены основные задачи, но без дополнительной оптимизации.

Незначительные отклонения от целевых метрик (например, $F1 = 0.85$).

или

Решены базовые задачи, но с критическими ошибками.

Низкое качество кода или отсутствие анализа.

Незачет

Невыполнение ключевых этапов.

Код нерабочий или отсутствует.

% 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.

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 зачете;

– 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 3-го года обучения.
- Наличие доступа к вычислительным ресурсам (GitLab, Google Colab или Yandex DataSphere, JupyterHub, Hugging Face).
- Разработаны лабораторные работы;

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

Цели, задачи и ожидаемые результаты

Цели организации вычислительной инфраструктуры:

- дать начальное представление о работе 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;
- Настройка GitLab Runner для автоматического тестирования кода.
- Разработка шаблонного репозитория для лабораторных работ с предустановленными зависимостями (PyTorch, Keras, TensorFlow, Hugging Face Transformers).
- Написание автотестов для проверки корректности выполнения заданий (например, fine-tuning моделей).
- Визуализация результатов тестирования через HTML-отчеты.
- Подготовка инструкций по работе с Git и облачными ресурсами.

Ожидаемые результаты студентов:

- начальное представление о работе в IT инфраструктуре (гит, нейминг).
- Навыки запуска и тестирования глубоких нейросетей в облачных средах.
- Понимание CI/CD-процессов в контексте разработки глубоких нейросетей.

Порядок реализации

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

Задача №2: Создание учетных записей студентов в gitlab вуза

Задача №3: Настройка GitLab Runner:

Для автоматического тестирования кода используется Docker-образ с предустановленными библиотеками (PyTorch, Keras, TensorFlow, Hugging Face Transformers).

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: Шаблонный репозиторий:

Включает:

.gitlab-ci.yml для CI/CD.

Скрипты для предобработки текста и обучения моделей.

Примеры кода для работы с BERT, GPT и другими архитектурами.

Задача №5: Автотесты:

Проверяют корректность fine-tuning моделей (например, ассигасу на тестовом датасете).

Задача №6: Визуализация результатов:

Генерация HTML-отчетов с результатами тестирования, включая метрики качества моделей.

Порядок проверки корректности:

- Наличие Git-репозитория у всех студентов.
- Шаблонный репозиторий с подключенными автотестами.
- Инструкция по работе с Git и CI/CD в формате README.md.

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 3-го года обучения.
- Наличие доступа к вычислительным ресурсам (GitLab) и к GPU/CPU (Kaggle, локальные серверы).
- 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 глубоких 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:

- Умение применять глубокие нейронные сети на практике.
- Владение библиотеками и фреймворками (PyTorch, TensorFlow, Keras, Hugging Face Transformers).
- Опыт решения задач классификации, сегментации и детекции объектов.
- Опыт применения трансферного обучения и fine-tuning моделей.

Порядок реализации**Задача №1: Подготовка лабораторных работ (в соответствии с п. 2.3.3 РПД)****1) Определение тем:**

- Реализация архитектуры LeNet
- Построение свёрточной нейронной сети для классификации изображений Fashion-MNIST
- Подбор оптимизатора для глубокой нейросети
- Исследование методов борьбы с переобучением.
- Метод grid-search и ранняя остановка модели
- Разработка нейросети VGG-16 для классификации изображений
- Детекция объектов с помощью YOLO.
- Сегментация сети с помощью Unet.
- Перенос обучения и fine-tuning моделей
- Визуальные трансформеры
- Применение трансформера для анализа тональности текстов
- Параллельное обучение глубоких нейросетей
- Исследование методов прунинга и дистилляции

Наименование Лабораторной работы	Содержание Лабораторной работы	Распределени е часов
2	3	
Реализация архитектуры LeNet	Реализовать архитектуру LeNet с использованием библиотеки TensorFlow или PyTorch. Обучить модель на датасете MNIST. Оценить точность модели на тестовом наборе данных. Проанализировать результаты и сделать выводы о производительности модели.	3 часа (2 часа ЛР, 1 час СР)

Наименование Лабораторной работы	Содержание Лабораторной работы	Распределени е часов
2	3	
Построение свёрточной нейронной сети для классификации и изображений Fashion-MNIST	Реализовать архитектуру CNN с использованием библиотеки TensorFlow или PyTorch. Обучить модель на датасете Fashion-MNIST. Оценить точность модели на тестовом наборе данных. Проанализировать результаты и сделать выводы о производительности модели.	3 часа (2 часа ЛР, 1 час СР)
Подбор оптимизатора для глубокой нейросети	Реализовать глубокую нейронную сеть с использованием библиотеки TensorFlow или PyTorch. Обучить модель с различными оптимизаторами (SGDM, Adam, RMSProp, AdaGrad). Сравнить время обучения и качество результатов. Выбрать самый эффективный алгоритм для дальнейшего использования.	3 часа (2 часа ЛР, 1 час СР)
Исследование методов борьбы с переобучением.	Реализовать глубокую нейронную сеть с использованием библиотеки TensorFlow или PyTorch. Применить методы борьбы с переобучением (регуляризация, dropout, пакетная нормализация). Сравнить результаты работы модели с и без методов борьбы с переобучением. Сделать выводы о производительности модели.	3 часа (2 часа ЛР, 1 час СР)
Метод grid-search и ранняя остановка модели	Реализовать глубокую нейронную сеть с использованием библиотеки TensorFlow или PyTorch. Применить метод grid-search для подбора оптимальных гиперпараметров. Использовать раннюю остановку для предотвращения переобучения. Оценить точность модели на тестовом наборе данных. Проанализировать результаты и сделать выводы о производительности модели.	3 часа (2 часа ЛР, 1 час СР)
Разработка нейросети VGG-16 для классификации и изображений	Реализовать архитектуру VGG-16 с использованием библиотеки TensorFlow или PyTorch. Обучить модель на датасете CIFAR-10. Оценить точность модели на тестовом наборе данных. Проанализировать результаты и сделать выводы о производительности модели.	3 часа (2 часа ЛР, 1 час СР)
Детекция объектов с помощью YOLO.	Реализовать архитектуру YOLO с использованием библиотеки TensorFlow или PyTorch. Обучить модель на датасете COCO. Оценить точность модели на тестовом наборе данных.	3 часа (2 часа ЛР, 1 час СР)

Наименование Лабораторной работы	Содержание Лабораторной работы	Распределе ние часов
2	3	
	Проанализировать результаты и сделать выводы о производительности модели.	
Сегментация сети с помощью Unet.	Реализовать архитектуру U-Net с использованием библиотеки TensorFlow или PyTorch. Обучить модель на датасете для сегментации. Оценить точность модели на тестовом наборе данных. Проанализировать результаты и сделать выводы о производительности модели.	3 часа (2 часа ЛР, 1 час СР)
Перенос обучения и fine-tuning моделей	Реализовать архитектуру нейронной сети с использованием библиотеки TensorFlow/Keras или PyTorch. Применить метод переноса обучения, используя предварительно обученную модель. Выполнить fine-tuning модели на новом датасете. Оценить точность модели на тестовом наборе данных. Проанализировать результаты и сделать выводы о производительности модели.	3 часа (2 часа ЛР, 1 час СР)
Визуальные трансформер ы	Реализовать архитектуру ViT с использованием библиотеки Hugging Face Transformers. Обучить модель на датасете CIFAR-10. Оценить точность модели на тестовом наборе данных. Проанализировать результаты и сделать выводы о производительности модели.	5 часов (4 часа ЛР, 1 час СР)
Применение трансформера для анализа тональности текстов	Реализовать архитектуру трансформера с использованием библиотеки Hugging Face Transformers. Обучить модель на датасете для анализа тональности. Оценить точность модели на тестовом наборе данных. Проанализировать результаты и сделать выводы о производительности модели.	5 часов (4 часа ЛР, 1 час СР)
Параллельное обучение глубоких нейросетей	Реализовать архитектуру глубокой нейронной сети с использованием библиотеки TensorFlow или PyTorch. Применить методы параллельного обучения (например, распределенное обучение на нескольких GPU). Оценить точность модели на тестовом наборе данных. Проанализировать результаты и сделать выводы о производительности модели.	6 часов (3 часа ЛР, 3 часа СР)
Исследование методов прунинга и дистилляции	Реализовать архитектуру глубокой нейронной сети с использованием библиотеки TensorFlow, Keras или PyTorch.	6 часов (3 часа ЛР, 3 часа СР)

Наименование Лабораторной работы	Содержание Лабораторной работы	Распределени е часов
2	3	
	Применить методы прунинга для уменьшения размера модели. Применить метод дистилляции для передачи знаний от большой модели к меньшей. Оценить точность модели на тестовом наборе данных. Проанализировать результаты и сделать выводы о производительности модели.	

2) Разработка заданий:

- Пошаговые инструкции.
- Примеры кода.

Контрольные вопросы.

Разработка заданий для лабораторной работы «Реализация архитектуры LeNet»

Цель: Освоить архитектуру LeNet-5 и ее применение для задачи классификации изображений.

2. Пошаговые инструкции

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

Шаг 1: Установка библиотек

Шаг 2: Загрузка и подготовка данных:

Шаг 3: Реализация LeNet-5:

Шаг 4: Обучение и оценка модели:

Шаг 5. Тестирование модели

3. Примеры кода на Python

Полный скрипт

#1. Установка библиотек

```
pip install torch torchvision
import torch
import torchvision
from torchvision import datasets, transforms
```

#2. Загрузка и подготовка данных:

```
transform = transforms.Compose([transforms.ToTensor()])
trainset = datasets.MNIST('./data', train=True, download=True, transform=transform)
testset = datasets.MNIST('./data', train=False, download=True, transform=transform)
trainloader = torch.utils.data.DataLoader(trainset, batch_size=64, shuffle=True)
testloader = torch.utils.data.DataLoader(testset, batch_size=64, shuffle=False)
```

#3. Реализация LeNet-5:

```
import torch.nn as nn
import torch.nn.functional as F
class LeNet(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(1, 6, 5)
```

```

self.conv2 = nn.Conv2d(6, 16, 5)
self.fc1 = nn.Linear(16*4*4, 120)
self.fc2 = nn.Linear(120, 84)
self.fc3 = nn.Linear(84, 10)
def forward(self, x):
    x = F.relu(self.conv1(x)) # 28x28 -> 24x24
    x = F.avg_pool2d(x, 2) # 24x24 -> 12x12
    x = F.relu(self.conv2(x)) # 12x12 -> 8x8
    x = F.avg_pool2d(x, 2) # 8x8 -> 4x4
    x = x.view(-1, 16*4*4)
    x = F.relu(self.fc1(x))
    x = F.relu(self.fc2(x))
    x = self.fc3(x)
    return x

```

#4. Обучение и оценка модели:

```

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = LeNet().to(device)
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.SGD(model.parameters(), lr=0.001, momentum=0.9)

```

```

for epoch in range(10):
    running_loss = 0.0
    for i, data in enumerate(trainloader, 0):
        inputs, labels = data[0].to(device), data[1].to(device)
        optimizer.zero_grad()
        outputs = model(inputs)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
        running_loss += loss.item()
    print(f"[{epoch + 1}/10] Loss: {running_loss / len(trainloader)}")

```

#5. Тестирование модели

```

correct = 0
total = 0
with torch.no_grad():
    for data in testloader:
        images, labels = data[0].to(device), data[1].to(device)
        outputs = model(images)
        _, predicted = torch.max(outputs.data, 1)
        total += labels.size(0)
        correct += (predicted == labels).sum().item()

```

```

print(f"Точность на тестовом наборе: {100 * correct / total:.2f}%")

```

4. Контрольные вопросы

1. Теоретические:

- В чем отличие CNN от обычной нейронной сети?
- Что такое фильтр свертки и пулинг?
- Каково назначение dropout и batch-norm в нейронных сетях?

2. Практические:

- Напишите код для загрузки данных Fashion-MNIST и визуализации изображений.

- Реализуйте простую CNN для Fashion-MNIST с использованием PyTorch/TensorFlow.
- 3. Аналитические:
 - Почему CNN лучше подходят для работы с изображениями, чем обычные нейронные сети?
 - Какие ошибки могут возникать при чрезмерном усложнении архитектуры нейронной сети?
 - Как можно улучшить результаты решения данной задачи?

Критерии оценки

Отлично: Полное выполнение всех шагов, анализ полученных результатов с выводом метрик и функции потерь.

Хорошо: Корректный код, но без анализа полученных результатов.

Удовлетворительно: Частичное выполнение с ошибками на этапах оценки и тестирования модели.

Неудовлетворительно: Невыполнение ключевых этапов.

3) Подготовка датасетов:

- Использование открытых данных Kaggle, paperswithcode и Hugging Face.
- 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 3-го года обучения,

Общее время на проект – не более 16 часов на каждого студента.

Имеется доступ к кейсам промышленных партнеров.

Цели, задачи и ожидаемые результаты

Цели организации вычислительной инфраструктуры:

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

Задачи преподавателя:

- сбор кейсов индустриальных партнеров;
- сбор кейсов преподавателей практиков и лабораторий в вузе;
- формирование ТЗ на зачетный проект на основе кейсов;
- разработка системы учёта результатов проекта в итоговой оценке зачета

Ожидаемые результаты студентов:

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

Порядок реализации

Задача №1: сбор кейсов индустриальных партнеров

1. Распознавание дефектов строительных конструкций (Construction Defect Detection)

Описание: Необходимо создать модель автоматического анализа фотографий строящихся зданий и выявлять дефекты на ранней стадии строительства. Это позволит минимизировать риски аварийных ситуаций и снизить затраты на ремонт.

Цель: Автоматизировать процесс контроля качества строительства, снизив потребность в ручном осмотре объектов и повысив безопасность эксплуатации сооружений.

Ожидаемые результаты:

Готовая модель распознавания дефектов на фотографиях.

Автоматизированный отчет о состоянии строительной площадки с указанием проблемных зон.

Сокращение затрат на инспекции и ремонты.

2. Автоматическое определение мошеннических операций (Anti-Fraud Detection)

Описание: Разработать алгоритм обнаружения подозрительных финансовых операций на основе анализа больших объемов транзакционных данных. Алгоритм должен уметь выделять аномалии среди множества легитимных платежей и переводов.

Цель: Минимизировать потери от фродовых действий, одновременно сокращая количество ложноположительных срабатываний (false positives), чтобы сохранить доверие клиентов.

Ожидаемые результаты:

Реализовать эффективную архитектуру нейронной сети для выявления мошенничества.

Оценить точность и скорость работы предложенной модели.

Определить факторы, наиболее влияющие на успешность обнаружений.

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

1. Безопасность дорожного движения.

Постановка задачи: Местные власти хотят повысить безопасность на дорогах, устанавливая камеры, способные автоматически регистрировать нарушения правил дорожного движения. Необходимо обучить YOLO для детекции пешеходов и велосипедистов на дороге.

Требования:

- Подготовить собственный датасет изображений дорог.
- Обучить YOLO v3/v5 для детекции объектов.
- Оценить точность детекции с помощью метрик AP и IoU.

2. Система автоматической сортировки гардероба.

Постановка задачи: Компания производит интерактивные зеркала для магазинов одежды, позволяющие покупателям получать персональные рекомендации по стилю. Необходимо обучить CNN-классификатор, который сможет определить категорию предмета одежды (рубашка, туфли, брюки и т.д.) по изображению.

Требования:

Создать и обучить CNN для классификации Fashion-MNIST.

Достичь точности классификации не ниже 90%.

Провести анализ влияния добавления новых слоёв и изменения размера фильтра на производительность.

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

1. Attention Mechanisms

Используя трансформер (например, DistilBERT), напишите простой код для классификации положительных и отрицательных отзывов (датасет IMDB). Измерьте точность.

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 зачтено или незачтено.

Критерии оценки:

Зачтено: Полное выполнение всех шагов, анализ полученных результатов с построением функции потерь и необходимых метрик, ответил практически на все теоретические вопросы.

Незачтено: 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 отметке за зачет

Выполнено в РПД, п 4.2

Порядок проверки корректности

Чек-лист для проверки лабораторных работ:

Набор кейсов промышленных партнеров – 21 шт;

Набор кейсов преподавателей практиков и лабораторий ВУЗа – 13 шт;

Набор ТЗ в количестве 15 штук.

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

5.1 Основная литература:

1. Шапиро, Л. Компьютерное зрение : учебное пособие / Л. Шапиро, Д. Стокман ; перевод с английского А. А. Богуславского под редакцией С. М. Соколова. — 5-е изд. (эл.). — Москва : Лаборатория знаний, 2024. — 763 с. — ISBN 978-5-93208-725-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/417998> (дата обращения: 22.07.2025). — Режим доступа: для авториз. пользователей.

2. Селянкин, В. В. Компьютерное зрение. Анализ и обработка изображений / В. В. Селянкин. — 3-е изд., стер. — Санкт-Петербург : Лань, 2023. — 152 с. — ISBN 978-5-507-

45583-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/276455> (дата обращения: 22.07.2025). — Режим доступа: для авториз. пользователей.

3. 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.

4. 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.

5. 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.

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, Кадури, Артур и Архангельская Екатерина. Глубокое обучение. "Издательский дом"" Питер""", 2017.

10. 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).

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
3. Google Scholar/arXiv – доступ к научным публикациям

Ресурсы свободного доступа

1. Официальная документация по TensorFlow <https://www.tensorflow.org/?hl=ru>
2. Официальная документация по Keras <https://www.tensorflow.org/guide/keras?hl=ru>
3. Официальная документация по pyTorch <https://docs.pytorch.org/docs/stable/index.html>
4. КиберЛенинка <http://cyberleninka.ru/>;
5. Американская патентная база данных <http://www.uspto.gov/patft/>
6. Министерство науки и высшего образования Российской Федерации <https://www.minobrnauki.gov.ru/>;
7. Федеральный портал "Российское образование" <http://www.edu.ru/>;
8. Информационная система "Единое окно доступа к образовательным ресурсам" <http://window.edu.ru/>;
9. Единая коллекция цифровых образовательных ресурсов <http://school-collection.edu.ru/> .
10. Проект Государственного института русского языка имени А.С. Пушкина "Образование на русском" <https://pushkininstitute.ru/>;
11. Справочно-информационный портал "Русский язык" <http://gramota.ru/>;
12. Служба тематических толковых словарей <http://www.glossary.ru/>;
13. Словари и энциклопедии <http://dic.academic.ru/>;
14. Образовательный портал "Учеба" <http://www.ucheba.com/>;
15. 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/](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 «Глубокое обучение» обеспечено разнообразие форматов и методов обучения (лекции, мастер-классы (от Кадурина А.А.), командные проектные задания и др.).

На лекционных занятиях даётся систематизированное представление о методах обучения глубоких нейронных сетей. В ходе лекций студенты знакомятся с базовыми принципами обучения глубоких нейронных сетей (борьба с переобучением, трансферное обучение и fine-tuning моделей, подбор параметров обучения). Особое внимание уделяется изучению известных глубоких нейросетей в области компьютерного зрения для задач классификации, детекции и сегментации изображений, а также в области компьютерного зрения для классификации текстов, распознавания и генерации речи. Демонстрируются практические примеры использования дообученных глубоких нейронных сетей в разных областях. Кроме того, в рамках дисциплины рассматривается оптимизация и повышение производительности глубоких нейронных сетей.

В ходе мастер-классов крупнейшими специалистами в области на практических примерах разбираются темы:

1. Параллельные вычисления и распределенное обучение глубоких нейронных сетей. Оптимизация вычислений: Quantization, Pruning, Knowledge Distillation (Мастер-класс от Кадурина А.А.)

2. Производительность нейронных сетей: оптимизация памяти и задержек. Инновации в hardware-accelerated computing для deep learning (Мастер-класс от Кадурина А.А.)

Лабораторные занятия посвящены формированию практических навыков работы с глубокими нейросетями. Студенты занимаются подбором оптимизатора для глубокой сети, отрабатывают техники борьбы с переобучением (dropout, batchnorm, регуляризация), создают глубокие свёрточные архитектуры с нуля, а также дообучают существующие нейросети (VGG-16, YOLO, Unet и т.д.), используют предобученный трансформер для работы с текстом и изображениями. Важной задачей также является отработка на практике навыка параллельного обучения нейросетей, а также прунинга и дистилляции.

При самостоятельной работе студентам необходимо изучать рекомендованную литературу (учебники, научные статьи, документацию библиотек) для полного понимания теоретических основ и современных подходов в области глубокого обучения нейросетей. Выполняя проектные задания, студент должен уметь: определять с помощью какой архитектуры можно решить поставленную задачу, как дообучать и настраивать глубокую архитектуру под свою задачу, как оптимизировать процесс обучения, а также как оценивать качество работы модели с помощью соответствующих метрик.

Важнейшим компонентом курса является самостоятельная проектная работа, в ходе которой студент разрабатывает законченное CV-приложение (например, система детекции объектов на фото или сегментации изображений). Такой проект позволяет закрепить навыки проектирования и реализации комплексных решений в области компьютерного зрения.

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. Предсказание оттока клиентов (Churn Prediction)

Описание: Используя исторические данные о клиентах Сбербанка, ваша задача заключается в создании модели предсказания вероятности ухода клиентов из банка. Данные включают такие признаки, как история транзакций, уровень дохода, демографические характеристики и активность пользователей банковских продуктов.

Цель: Создать систему раннего предупреждения, позволяющую Сбербанку своевременно реагировать на потенциальных уходящих клиентов и предложить им специальные условия, акции или продукты для удержания.

Ожидаемые результаты:

Построение эффективной модели предсказания риска оттока клиентов.

Анализ влияния различных факторов на вероятность ухода клиента.

Создание рекомендаций по улучшению клиентского опыта и повышению лояльности клиентов.

2. Персонализированная рекомендационная система продуктов (Product Recommendations)

Описание: Реализуйте рекомендательную систему, способную предлагать клиентам индивидуальные банковские продукты и услуги на основе их истории покупок, поведения и предпочтений. Используйте матрицу взаимодействий клиентов с услугами банка и оценивайте метрики точности рекомендации.

Цель: Увеличить продажи услуг и повысить удовлетворенность клиентов путем персонального подхода и предложений нужных продуктов в нужное время.

Ожидаемые результаты:

Моделируйте поведение клиентов с применением техники Deep Learning.

Создайте интерфейс API для интеграции рекомендательной системы в веб-интерфейсы и мобильные приложения.

Проведите A/B-тестирование разработанной системы.

3. Прогнозирование спроса на кредитные продукты с использованием глубоких нейронных сетей

Описание: Сбербанк предлагает широкий спектр кредитных продуктов, и важно предсказывать спрос на них для эффективного планирования и управления ресурсами. Задача — разработать модель, которая будет предсказывать спрос на кредитные продукты на основе исторических данных и внешних факторов (например, экономические показатели, сезонность).

Цель: Создать модель, которая сможет предсказывать спрос на кредитные продукты с высокой точностью, что позволит банку оптимизировать свои ресурсы и предложения.

Ожидаемый результат: Глубокая нейронная сеть, способная анализировать сложные взаимосвязи между различными факторами и предсказывать спрос на кредитные продукты.

4. NLP-анализ жалоб клиентов в свободной форме

Описание: В рамках клиентского сервиса Сбербанк обрабатывает обращения из чатов, мобильного приложения и жалобной формы. Требуется построить модель семантического анализа, выделяющую суть обращения, определяющую тональность и потенциальную серьёзность инцидента.

Цель: Автоматизировать классификацию обращений для ускорения маршрутизации и выявления повторяющихся болевых точек в продуктах и процессах.

Ожидаемый результат: Прототип модели, автоматически выделяющей темы жалоб (например, «ошибка в приложении», «двойное списание»), их эмоциональную окраску и критичность.

5. Объяснимость и контроль генеративных моделей в банковском ИИ

Описание: Банк активно использует LLM и NLP-сервисы (в чат-ботах, генерации шаблонов ответов, автоответах на e-mail), однако встает вопрос: как объяснять и контролировать поведение таких моделей, особенно в юридически значимых коммуникациях?

Цель: Исследовать подходы к трассировке решений LLM (например, через логирование reasoning chain, пост-фильтрацию ответов, встроенные правила).

Ожидаемый результат: Концепция системы explainability + compliance-модуля, обеспечивающего соответствие генерации стандартам банка и регулятора.

6. 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.

7. Голосовой помощник

Описание: Сбербанк разрабатывает голосовые ассистенты и системы синтеза речи. Задача – реализовать модель Tacotron и обучить её на датасете для генерации речи.

Цель: Исследовать и применить модель Tacotron для генерации речи.

Ожидаемые результаты: Модель, способная генерировать речь на основе текстовых входных данных.

8. Автоматическое определение мошеннических операций (Anti-Fraud Detection)

Описание: Разработать алгоритм обнаружения подозрительных финансовых операций на основе анализа больших объемов транзакционных данных. Алгоритм должен уметь выделять аномалии среди множества легитимных платежей и переводов.

Цель: Минимизировать потери от фродовых действий, одновременно сокращая количество ложноположительных срабатываний (false positives), чтобы сохранить доверие клиентов.

Ожидаемые результаты:

Реализовать эффективную архитектуру нейронной сети для выявления мошенничества.

Оценить точность и скорость работы предложенной модели.

Определить факторы, наиболее влияющие на успешность обнаружений.

9. Чат-бот поддержки клиентов (Customer Support Chatbot)

Описание: Разработайте чат-бота, использующего глубокие нейронные сети, который сможет автоматически отвечать на запросы клиентов, обрабатывая тексты сообщений и предоставляя точные и полезные ответы.

Цель: Повышение эффективности службы поддержки клиентов, снижение нагрузки на операторов кол-центра и повышение уровня обслуживания клиентов.

Ожидаемые результаты:

Запустить прототип бота с возможностью общения на естественном языке.

Интеграция чата с существующими каналами коммуникации банка (сайтом, мобильным приложением).

Измерить точность понимания запросов и удовлетворение клиентов качеством предоставляемых ответов.

10. Система идентификации лица для входа в мобильный банк (Face ID Authentication)

Описание: Необходимо разработать систему биометрической аутентификации, основанную на сверточных нейронных сетях, которая позволит пользователям мобильного приложения Сбербанк входить в аккаунт, используя технологию распознавания лиц.

Цель: Обеспечить высокий уровень безопасности и удобства пользования мобильным банком посредством надежного метода биометрии.

Ожидаемые результаты:

Повышение надежности защиты аккаунта клиента.

Уменьшение количества попыток несанкционированного доступа.

Увеличение числа пользователей мобильной версии банковского сервиса.

Кейсы от «АВАЛАБ»

1. Генерация документации и шаблонов договоров

Описание: Юридический департамент регулярно работает с договорами долевого участия, актами приёма-передачи и другими документами. Использование LLM может значительно сократить время на подготовку черновиков — достаточно ввести параметры сделки.

Цель: Создать систему, которая генерирует адаптированные тексты документов по вводным данным (тип объекта, этаж, площадь, ФИО, сроки и пр.).

Ожидаемый результат: Генератор документов в формате Word или PDF с автоматической подстановкой параметров и соблюдением юридического стиля.

2. Модель прогнозирования сроков сдачи объектов на основе текстовых и визуальных данных

Описание: Девелоперская компания ведёт аналитический архив по срокам строительства. С помощью мультимодальных моделей (текстовые отчёты + фото стройки) можно прогнозировать вероятность отклонения от графика сдачи.

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

Ожидаемый результат: Прототип, который показывает вероятность отклонений и даёт текстовые пояснения (основанные на распознанных признаках — «не завершены фасадные работы», «монтаж инженерии не начат»).

3. Анализ обращений клиентов и CRM-переписки

Описание: В службе клиентского сервиса застройщика ежедневно обрабатываются десятки обращений (e-mail, звонки, мессенджеры). Требуется реализовать систему семантического анализа и классификации NLU: выявлять суть обращений, уровень удовлетворенности, отслеживать повторяющиеся запросы.

Цель: Автоматизировать первичный разбор и маршрутизацию запросов по тематике (сдача объекта, отделка, документы, жалоба и т.д.).

Ожидаемый результат: Прототип, который выделяет суть обращений и формирует дашборд по текущим «болям» клиентов.

4. Обратная генерация — ИИ-помощник для покупателей квартир

Описание: Будущие покупатели часто задают типовые вопросы о квартирах, планировках, ипотеке, акциях, сроках. Вместо call-центра предлагается реализовать LLM-бота, который обрабатывает текстовые и голосовые запросы, показывает планировки, ссылается на PDF-документы и может «объяснять» информацию простым языком.

Цель: Упростить коммуникацию с клиентами на этапе выбора квартиры и повысить качество первичного контакта.

Ожидаемый результат: Демо-бот, способный отвечать на вопросы о жилом комплексе, ориентируясь в его характеристиках и маркетинговых документах.

5. Прогнозирование спроса на жилые комплексы

Описание: Девелоперская компания хочет предсказывать спрос на жилые комплексы в различных районах города. Задача — разработать модель, которая будет анализировать данные о рынке недвижимости, демографические данные и другие факторы, влияющие на спрос.

Цель: Создать модель, которая сможет предсказывать спрос на жилые комплексы с высокой точностью.

Ожидаемый результат: Глубокая нейронная сеть, способная анализировать данные и выдавать точные прогнозы по спросу на жилые комплексы.

6. Анализ поведения клиентов на сайте

Описание: Девелоперская компания имеет сайт, на котором клиенты могут просматривать информацию о жилых комплексах и оставлять заявки. Задача — разработать модель, которая будет анализировать поведение клиентов на сайте и предсказывать их намерения.

Цель: Создать модель, которая сможет анализировать данные о поведении клиентов на сайте и предсказывать их намерения (например, покупка квартиры, запрос на консультацию).

Ожидаемый результат: Глубокая нейронная сеть, способная анализировать данные о поведении клиентов и предсказывать их намерения.

7. Оптимизация работы с клиентами в колл-центре

Описание: Девелоперская компания имеет колл-центр, который обрабатывает множество звонков от клиентов. Задача — разработать модель, которая будет оптимизировать работу колл-центра, улучшая распределение звонков и сокращая время ожидания клиентов.

Цель: Создать модель, которая сможет обучаться на основе данных о звонках и улучшать распределение звонков между операторами.

Ожидаемый результат: Система, способная оптимизировать работу колл-центра, что повысит удовлетворенность клиентов и снизит нагрузку на операторов.

8. Распознавание дефектов строительных конструкций (Construction Defect Detection)

Описание: Необходимо создать модель автоматического анализа фотографий строящихся зданий и выявлять дефекты на ранней стадии строительства. Это позволит минимизировать риски аварийных ситуаций и снизить затраты на ремонт.

Цель: Автоматизировать процесс контроля качества строительства, снизив потребность в ручном осмотре объектов и повысив безопасность эксплуатации сооружений.

Ожидаемые результаты:

Готовая модель распознавания дефектов на фотографиях.

Автоматизированный отчет о состоянии строительной площадки с указанием проблемных зон.

Сокращение затрат на инспекции и ремонты.

9. Проектирование энергоэффективных зданий (Energy-Efficient Building Design)

Описание: Используя большие объемы климатических данных и проектировочных характеристик, построить модель, которая подберёт оптимальное расположение окон, материалы стен и кровель здания таким образом, чтобы обеспечить минимальную энергозатратность конструкции.

Цель: Создание инструмента для проектирования энергоэффективных жилых комплексов, уменьшающих эксплуатационные расходы и улучшающих экологичность среды обитания.

Ожидаемые результаты:

Программа подбора материалов и конструктивных элементов здания.

Симуляция энергопотребления разных вариантов дизайна.

Экономически обоснованное предложение проекта энергоэффективного дома.

10. Управление проектом на основе интеллектуального анализа рисков (Risk Management with ML)

Описание: Спроектируйте модель для мониторинга рисков на крупных стройплощадках. Эта модель должна анализировать широкий спектр данных (например,

погодные условия, логистику поставок стройматериалов, трудовую занятость сотрудников) и давать прогнозы возможных сбоев в проекте.

Цель: Минимизировать риск отставания сроков сдачи проекта и сократить непредвиденные финансовые издержки благодаря раннему предупреждению потенциальных угроз.

Ожидаемые результаты:

Рабочий механизм анализа рисков и их количественной оценки.

Инструмент автоматической отчетности по рискам для менеджеров проектов.

Улучшенная управляемость крупными строительными объектами.

11. Умный контроль строительства с помощью дронов (Drone-Based Construction Monitoring)

Описание: Использование беспилотников для аэрофотосъемки строительства и последующего автоматизированного анализа снимков. Задача состоит в разработке нейросетевого классификатора, который сможет отслеживать прогресс строительства, определять отклонения от плана и оценивать общий объем выполненных работ.

Цель: Максимально автоматизировать мониторинг состояния строительных площадок, обеспечивая своевременную оценку прогресса и выявление отклонений от запланированного графика.

Ожидаемые результаты:

Нейросетевая модель, обеспечивающая точное распознавание типов строений и этапов строительства.

Интерактивная карта, отображающая текущее состояние строительства и отмечающую возможные нарушения.

Возможность автоматизации инспекций на площадках и оперативное реагирование на замечания.

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

Hugging Face Spaces – развертывание CV-моделей в виде демо

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

2. Системы управления версиями и коллаборации

Git/GitHub/GitLab – контроль версий кода и совместная разработка

3. Инструменты для работы с данными

Label Studio – разметка датасетов

DVC (Data Version Control) – управление версиями данных

Apache Spark – обработка больших текстовых корпусов

4. Система управления обучением

Moodle – сдача работ

7.2 Перечень лицензионного и свободно распространяемого программного обеспечения

1. Лицензионное ПО

VSCode – IDE для Python (свободнораспространяемое)

LibreOffice– оформление отчетов (свободнораспространяемое)

2. Свободное ПО (Open Source)

Hugging Face Transformers – предобученные модели (BERT, GPT, визуальные трансформеры)

Hugging Face Optimum – оптимизация и развёртывание моделей

Фреймворки и библиотеки для DL:

PyTorch/TensorFlow/Keras– разработка нейросетей

Инструменты для визуализации:

Streamlit/Gradio – создание веб-интерфейсов для моделей

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 индустриальным 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	Виртуальная машина с GPU	Виртуальная машина с GPU NVIDIA® Tesla® V100 2 GPU 8 vCPU 128 Гб RAM	1	1	Шт
		ОС Ubuntu_24.04	1		Шт
		Системный диск SSD	1		Шт
			2000		Гб
		Диск SSD	1		Шт
			4096		Гб
		Диск SSD	1		Шт
			4096		Гб
		Аренда публичного IP	1		Шт

3	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
4	ML Inference Instance Type GPU	Время работы в месяц	40	1	Ч
		Инстанс 8 x NVIDIA® H100 NVLink PCIe 160 vCPU 1520 GB RAM	1		Шт
		Количество запросов к ML-моделям	1		Млн. Шт
		Кэш ML-моделей	160		Гб
5	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.