

**Федеральное государственное бюджетное учреждение науки
Институт машиноведения им. А.А. Благонравова
Российской академии наук
(ИМАШ РАН)**

УТВЕРЖДАЮ
Заместитель директора по вопросам
подготовки кадров высшей
квалификации ИМАШ РАН
« 01 » _____ 2026 года
к.ф.-м.н. Жданов Э.Р.
М.П. _____



**Комплект оценочных материалов
по дисциплине «Современные архитектуры нейронных сетей»**

Направление подготовки:	02.04.03 Математическое обеспечение и администрирование информационных систем
Профиль подготовки:	Искусственный интеллект и интеллектуальная обработка данных
Квалификация:	Магистр
Форма обучения	Очная

Содержание

1. Общие положения.....	3
2. Результаты освоения дисциплины, подлежащие проверке.....	3
3. Контроль и оценка освоения учебной дисциплины.....	4
4. Оценочные материалы для проведения текущего контроля успеваемости.....	5
5. Оценочные материалы для проведения промежуточной аттестации.....	6

1. Общие положения

Комплект оценочных материалов (далее - ОМ) предназначен для контроля и оценки образовательных достижений обучающихся, освоивших программу дисциплины «Современные архитектуры нейронных сетей».

ОМ включают контрольные материалы для проведения текущего контроля успеваемости и промежуточной аттестации в форме зачета.

ОМ разработаны на основании:

- основной профессиональной образовательной программы по направлению подготовки 02.04.03 Математическое обеспечение и администрирование информационных систем;

- рабочей программы дисциплины «Современные архитектуры нейронных сетей».

2. Результаты освоения дисциплины, подлежащие проверке

В результате освоения дисциплины обучающийся должен обладать следующими компетенциями

ПК-1	Создание информационных технологий нового поколения
ПК-1.2	Управление разработкой продуктов, услуг и решений на основе больших данных
ПК-1.3	Разработка и внедрение новых методов и технологий исследования больших данных

Код индикатора компетенции	Планируемые результаты обучения по дисциплине
ПК-1.2	<i>Знать:</i> - современные нейросетевые архитектуры (CNN, Transformer, GNN, GAN) и критерии их выбора в зависимости от типа больших данных (изображения, текст, графы, временные ряды); - методы оценки качества и вычислительной эффективности моделей при работе с большими объемами данных; - принципы построения пайплайнов обработки больших данных с использованием глубокого обучения (ETL, батчирование, шардирование). <i>Уметь:</i> - обосновывать выбор конкретной архитектуры для бизнес-задачи на основе анализа данных и требований к продукту; - оценивать временные и вычислительные затраты обучения и инференса на реальных кластерах данных; - организовывать эксперименты по сравнению нескольких архитектур для выбора лучшего решения. <i>Иметь практический опыт:</i> - управления процессом разработки: от постановки задачи до внедрения обученной модели в пилотный продукт; - настройки гиперпараметров и распределенного обучения на нескольких GPU для обработки датасетов терабайтного размера; - применения MLOps-практик (версионирование данных, модели, экспериментов) при разработке продуктов на больших данных.
ПК-1.3	<i>Знать:</i> - передовые архитектуры нейронных сетей (Transformer с разреженным вниманием, Graph Transformer, диффузионные модели, GAN с усилением); - методы сжатия и ускорения моделей (Quantization-Aware Training, Pruning, Knowledge Distillation) для внедрения на граничных устройствах; - принципы построения гибридных систем (например, нейро-символьный подход) для анализа больших слабоструктурированных данных. <i>Уметь:</i> - адаптировать и дорабатывать существующие нейросетевые архитектуры под специфику новых типов больших данных (например, пространственно-

	временные, мультимодальные); - проводить сравнительный анализ существующих методов и предлагать улучшения (модификация функции потерь, архитектуры, метода оптимизации); - развертывать исследовательские прототипы в экспериментальной среде и оценивать их масштабируемость на реальных больших данных. <i>Иметь практический опыт:</i> - разработки и реализации новой нейросетевой архитектуры (или значительной модификации существующей) для нестандартной задачи анализа больших данных; - создания пайплайна обучения и оценки с поддержкой mixed precision и распределенных вычислений на кластере; - оптимизации инференса для промышленного внедрения (TensorRT, ONNX, аппаратные ускорители) с обеспечением SLAs по latency и throughput.
--	--

3. Контроль и оценка освоения учебной дисциплины

Элемент учебной дисциплины	Текущий контроль		Промежуточная аттестация	
	Форма контроля	Проверяемые идк	Форма контроля	Проверяемые идк
Тема 1. Эволюция архитектур и основы глубокого обучения	Отчет по практикуму по решению задач	ПК-1.2 ПК-1.3		
Тема 1. Эволюция архитектур и основы глубокого обучения	Отчет по лабораторному практикуму	ПК-1.2 ПК-1.3		
Тема 2. Современные архитектуры для обработки последовательностей и внимания	Отчет по практикуму по решению задач	ПК-1.2 ПК-1.3		
Тема 2. Современные архитектуры для обработки последовательностей и внимания	Отчет по лабораторному практикуму	ПК-1.2 ПК-1.3		
Тема 3. Генеративные модели и работа с неструктурированными данными	Отчет по практикуму по решению задач	ПК-1.2 ПК-1.3		
Тема 3. Генеративные модели и работа с неструктурированными данными	Отчет по лабораторному практикуму	ПК-1.2 ПК-1.3		
Тема 4. Масштабирование, оптимизация и внедрение моделей	Отчет по практикуму по решению задач	ПК-1.2 ПК-1.3		
Тема 4. Масштабирование, оптимизация и внедрение моделей	Отчет по лабораторному практикуму	ПК-1.2 ПК-1.3		
			Зачет	ПК-1.2 ПК-1.3

4. Оценочные материалы для проведения текущего контроля успеваемости

*Показатели и критерии оценивания компетенций на различных этапах их формирования,
описание шкал оценивания*

Практикум по решению задач
10-8 – работа и отчет выполнены в срок, самостоятельно, правильно определены соответствующие спецификации, использована требуемая информация, правильно выполнены требуемые расчеты, сделаны необходимые выводы, хорошо аргументированы, даны исчерпывающие ответы на все поставленные вопросы; 7-5– работа и отчет выполнены в срок, самостоятельно, правильно определены соответствующие спецификации, использована требуемая информация, правильно выполнены требуемые расчеты, необходимые выводы сделаны частично, хорошо аргументированы, даны ответы на все поставленные вопросы; 4-1 – работа и отчет выполнены в срок, в основном самостоятельно, выводы сделаны частично, слабо аргументированы, даны ответы не на все вопросы; 0 – обучающийся подготовил работу и отчет несамостоятельно и/или не завершил в срок, описание содержит ошибки, выводы и ответы на вопросы отсутствуют.
Лабораторный практикум
15-13 – работа выполнена в срок, самостоятельно, правильно поняты и использованы соответствующие формулы, правильно определены соответствующие спецификации, использована требуемая информация, правильно выполнены требуемые расчеты, правильно выбраны совместимые комплектующие, сделаны необходимые выводы, хорошо аргументированы, даны исчерпывающие ответы на все поставленные вопросы; 12-9 – работа выполнена в срок, самостоятельно, правильно поняты и использованы соответствующие формулы, правильно определены соответствующие спецификации, использована требуемая информация, правильно выполнены требуемые расчеты, правильно выбраны совместимые комплектующие, необходимые выводы сделаны частично, хорошо аргументированы, даны ответы на все поставленные вопросы; 8-6 – работа выполнена в срок, в основном самостоятельно, использованы соответствующие формулы; определены соответствующие спецификации, имеются ошибки в расчетах; выбраны совместимые комплектующие необходимые, выводы сделаны частично, слабо аргументированы, даны ответы не на все вопросы; 5-1 – обучающийся подготовил работу несамостоятельно или не завершил в срок, описание спецификации содержит незначительные ошибки, выводы и ответы на вопросы отсутствуют.

Порядок начисления баллов

Если студент набрал достаточное количество баллов ТКУ по дисциплине (более 70 баллов) в течение семестра, он может отказаться от сдачи зачета, и результат будет выставлен автоматически по его согласию.

*Типовые контрольные задания и иные материалы, необходимые для процедуры
оценивания знаний, умений, навыков и(или) опыта деятельности*

Типовые задания для практикумов по решению задач

Тема 1. Эволюция архитектур и основы глубокого обучения

Задача 1. Анализ норм градиентов в глубокой сети (ручной расчет и построение графиков).

Тема 3. Генеративные модели и работа с неструктурированными данными

Задача 1. Диагностика коллапса GAN по графикам потерь – предложение методов лечения.

Тема 4. Масштабирование, оптимизация и внедрение моделей

Задача 1. Расчет ускорения от mixed precision и распределенного обучения (теоретическая оценка).

Типовые задания для лабораторных практикумов

Тема 1. Эволюция архитектур и основы глубокого обучения

Задание 1. Реализация CNN для CIFAR-10 – эксперименты с гиперпараметрами (размер ядра, число фильтров, Dropout).

Тема 2. Современные архитектуры для обработки последовательностей и внимания

Задание 1. Fine-tuning BERT для анализа тональности текстов. Визуализация карт внимания. Сравнение с LSTM.

Тема 3. Генеративные модели и работа с неструктурированными данными

Задание 1. Реализация DCGAN для генерации цифр MNIST.

Задание 2. Классификация узлов в социальном графе (Zachary's Karate Club) с помощью GCN в PyTorch Geometric – сравнение с MLP без учета структуры

Тема 4. Масштабирование, оптимизация и внедрение моделей

Задание 1. Оптимизация инференса ResNet через ONNX Runtime и int8-квантование – замеры latency и точности.

Задание 2. Развертывание оптимизированной модели как REST API (FastAPI + Docker) с логированием времени ответа.

5. Оценочные материалы для проведения промежуточной аттестации

Промежуточная аттестация по дисциплине проводится в форме зачета.

Зачет представляет собой выполнение обучающимся заданий билета, включающего в себя:

Задание №1 – теоретический вопрос на знание базовых понятий предметной области дисциплины, а также позволяющий оценить степень владения обучающегося принципами предметной области дисциплины, понимание их особенностей и взаимосвязи между ними.

Задание №2 – задание на анализ ситуации из предметной области дисциплины и выявление способности обучающегося выбирать и применять соответствующие принципы и методы решения практических проблем, близких к профессиональной деятельности.

Задания №3 – задания на проверку умений и навыков, полученных в результате освоения дисциплины.

Выполнение обучающимся заданий билета оценивается по следующей балльной шкале:

Задание 1: 0-30 баллов

Задание 2: 0-30 баллов

Задание 3: 0-40 баллов

«Зачтено»

– 90-100 – ответ правильный, логически выстроен, использована профессиональная терминология. Задания решены правильно. Обучающийся правильно интерпретирует полученный результат.

– 70 -89 – ответ в целом правильный, логически выстроен, использована профессиональная терминология. Ход решения заданий правильный, ответ неверный. Обучающийся в целом правильно интерпретирует полученный результат.

– 50 - 69 – ответ в основном правильный, логически выстроен, использована профессиональная терминология. Задание решено частично.

«Не зачтено»

– менее 50 – ответы на теоретическую часть неправильные или неполные. Задания не решены.

Типовые задания для проведения промежуточной аттестации обучающихся

Задания 1 типа

1. Проблема градиентного взрыва и исчезновения: причины возникновения, математическая сущность (произведение якобианов) и методы борьбы (BatchNorm, ResNet-

связи, инициализация Xavier/He, градиентное клиппирование).

2. Сравнение функций активации: Sigmoid, Tanh, ReLU, Leaky ReLU, ELU, Swish. Когда и почему ReLU предпочтительнее, а когда он проигрывает? Как мёртвые нейроны ReLU влияют на обучение?

3. Сверточные нейронные сети (CNN): что такое свертка (cross-correlation), параметры (stride, padding, dilation). Как меняется размер карты признаков после свертки и пулинга? Вывести формулу для выходной размерности.

4. Рецептивное поле (receptive field): определение, как оно увеличивается с глубиной сети. Зачем нужно большое рецептивное поле в задачах сегментации и обнаружения объектов?

5. Рекуррентные нейронные сети (RNN): проблема долгосрочных зависимостей. Устройство LSTM (входной, забывающий, выходной вентили) и GRU. Почему LSTM лучше базовой RNN?

6. Глубина и ширина сети: компромиссы между глубиной и шириной. Что такое деградация точности при увеличении глубины и как её решают остаточные связи (ResNet)?

7. Механизм внимания (Attention): общая формула (Query, Key, Value). Объясните скалярное внимание (dot-product attention) и аддитивное внимание (Bahdanau). Почему внимание решает проблему узкого места RNN?

8. Self-attention vs Cross-attention: в чём разница, где применяется каждый вид (Transformer, BERT, GPT). Что такое masked self-attention и для чего он нужен в декодере?

9. Архитектура Transformer: устройство энкодера и декодера (multi-head attention, feed-forward, add&norm, позиционное кодирование). Почему позиционное кодирование необходимо, если в трансформере нет рекуррентности?

10. Multi-head attention: зачем использовать несколько голов внимания? Что происходит, когда головы учатся разным паттернам зависимостей (пример: локальные vs глобальные связи)?

11. BERT (Bidirectional Encoder Representations from Transformers): принцип предобучения (Masked Language Model + Next Sentence Prediction). В чём отличие BERT от GPT по направленности внимания? Для каких задач BERT не подходит и почему?

12. GPT (Generative Pre-trained Transformer): архитектура (только декодер), задача предобучения (авторегрессионное предсказание следующего токена). Сравнение GPT и BERT для генерации vs понимания текста.

13. **Generative Adversarial Networks (GAN)**: архитектура генератора и дискриминатора, функция потерь (min-max игра). Как обучать GAN стабильно (чередование шагов, метки реальное/фейковое)?

14. **Проблемы обучения GAN**: мод коллапс (mode collapse), нестабильность градиентов, несходимость. Объясните каждую проблему и методы лечения (WGAN, WGAN-GP, label smoothing, spectral norm).

15. **Вариационные автокодировщики (VAE)**: отличие от обычного автоэнкодера (вероятностная интерпретация, репараметризация). Что такое ELBO (evidence lower bound) и почему KL-дивергенция служит регуляризатором?

16. **VAE vs GAN**: сравнительный анализ – качество генерации, разнообразие выборки, устойчивость обучения, скорость инференса. Когда какой подход предпочтительнее?

17. **Графовые нейронные сети (GNN)**: принцип обмена сообщениями (message passing). Чем GCN (Graph Convolutional Network) отличается от обычной свертки на изображениях? Задача классификации узлов, предсказания связей.

18. **Over-smoothing в GNN**: что это такое, почему происходит при увеличении числа слоёв? Какие архитектуры решают эту проблему (GAT, GraphSAGE, JK-Net)?

19. **Пакетная нормализация (BatchNorm)**: как работает (нормировка по батчу, обучаемые сдвиг и масштаб). Почему BatchNorm помогает бороться с внутренним ковариатным сдвигом и позволяет использовать более высокую скорость обучения?

20. **Оптимизаторы в глубоком обучении:** сравнение SGD с моментом, Adam, AdamW, RMSProp. Что такое weight decay и в чём отличие Adam от AdamW (исправление регуляризации)?
21. **Смешанная точность (Mixed Precision, FP16):** зачем нужна, как работает (сохранение мастер-весов в FP32, масштабирование градиентов). Во сколько раз может ускорить обучение на современных GPU (A100, H100)?
22. **Распределённое обучение:** DataParallel vs DistributedDataParallel (PyTorch). В чём проблема синхронизации градиентов (All-Reduce). Как оценить эффективность масштабирования по закону Амдала?
23. **Методы сжатия моделей:** Quantization (пост-тренировочное и QAT), Pruning (неструктурированное, структурное), Knowledge Distillation. Сравнение по степени сжатия, потере точности и ускорению инференса.
24. **Инференс-оптимизации:** ONNX Runtime (графовые оптимизации, слияние операций) и TensorRT (оптимизации под NVIDIA GPU, автоматическое квантование). Какие операции наиболее критичны для ускорения?
25. **MLOps для нейронных сетей:** компоненты системы (версионирование данных DVC, модели, экспериментов MLflow, мониторинг дрейфа данных/предсказаний). Как организовать CI/CD для переобучения модели при появлении новых данных?

Задания 2 типа

1. Выбор архитектуры для мультимодальных данных
Компания разрабатывает систему мониторинга состояния промышленного оборудования. Данные: вибродатчики (временные ряды, 1000 точек/сек), тепловизионные изображения (64×64), акустические сигналы (спектрограммы). Какую архитектуру нейронной сети вы предложите? Нужно ли обрабатывать каждый тип данных отдельным модулем или существует единая архитектура? Обоснуйте выбор.
2. Проблема градиентного взрыва в глубокой RNN
Вы обучаете глубокую RNN (6 слоёв LSTM) для прогнозирования курса акций. На 500-й итерации потери становятся NaN, а нормы градиентов достигают 10^5 . Ваши действия? Какие методы диагностики примените и как измените процесс обучения?
3. Классификация спутниковых снимков с высоким разрешением ($10\,000 \times 10\,000$)
У вас есть 10 000 спутниковых снимков огромного разрешения. Вычислительных ресурсов недостаточно, чтобы подавать полные изображения в CNN. Какие подходы существуют для обработки таких данных (патчинг, пирамида признаков, вниманием поверх патчей)? Опишите компромиссы каждого подхода.
4. Медицинская диагностика – несбалансированный датасет
Задача: обнаружение редкого заболевания на рентгеновских снимках (1% положительных примеров). Вы обучили стандартную CNN и получили accuracy 99%, но ни одного больного модель не нашла. В чём проблема? Предложите конкретные изменения в архитектуре, функции потерь и процессе обучения.
5. Выбор функции активации для очень глубокой сети (100+ слоёв)
Вы проектируете сеть для распознавания объектов. Глубина > 100 слоёв. Sigmoid и Tanh в средних слоях дают градиенты близкие к нулю. ReLU приводит к мёртвым нейронам. Какие альтернативы вы рассмотрите (Swish, GELU, ELU)? Почему выбор активации критичен для такой глубины?
6. Ограниченная память GPU при обучении CNN
Ваша модель ResNet-50 не помещается в память одной GPU (batch size = 1, но даже этого недостаточно из-за промежуточных активаций). Какие техники вы примените: градиентная контрольная точка (checkpointing), обучение с накоплением градиентов, уменьшение разрешения? Какая техника сохранит точность лучше всего?
7. Обработка очень длинных последовательностей (100 000+ токенов)
Задача: анализ всей книги (100 000 слов) для извлечения связей между персонажами. Стандартный Transformer не работает из-за квадратичной сложности внимания $O(n^2)$. Какие

методы вы предложите (Longformer, Big Bird, разреженное внимание, скользящее окно, использование RNN поверх энкодера)? Сравните их по сложности и потенциальной потере информации.

8. Система вопросно-ответная (QA) для закрытой корпоративной документации. Вам нужно построить QA-систему, которая отвечает на вопросы по внутренним документам компании (10 000 PDF-файлов, каждый до 50 страниц). Запрещено отправлять данные во внешние API (OpenAI и т. д.). Какую архитектуру вы выберете: RAG (retrieval-augmented generation) с BERT для ранжирования и локальной LLM для генерации? Или fine-tune кодировщика? Обоснуйте.

9. Низкая латентность инференса для чат-бота. Вы развернули GPT-подобную модель для чат-бота. Время ответа – 3 секунды. Требование бизнеса – менее 0.5 секунды. Какие методы вы примените: KV-caching, спектральное квантование, уменьшение числа слоёв (pruning), speculative decoding? Какой метод даст наибольший выигрыш при минимальной потере качества?

10. Мультиязычный анализ тональности без разметки на всех языках. У вас есть разметка тональности только для английского (100k примеров) и русского (50k). Требуется классифицировать отзывы на 10 других языках (корейский, арабский и т.д.) без разметки. Как использовать предобученные мультиязычные модели (mBERT, XLM-R)? Как оценить качество на целевых языках без тестовых меток?

11. Transformer не сходится при обучении с нуля. Вы обучаете Transformer-энкодер для классификации медицинских текстов с нуля (не fine-tune). Потери валидации не уменьшаются после 10 эпох. Что вы проверите: масштабирование инициализации, learning rate с warmup, позиционное кодирование, использование LayerNorm до или после attention? Какие настройки критичны для устойчивости трансформера?

12. Борьба с галлюцинациями (hallucinations) в генеративных моделях. Генеративная модель на основе GPT при составлении отчётов по базе знаний выдумывает факты (галлюцинирует). Как модифицировать архитектуру или процесс генерации, чтобы снизить этот эффект (контрастивный поиск, декодирование с ограничениями, обучение с подкреплением, дообучение на проверенных фактах)?

33. Генерация синтетических медицинских изображений при дефиците данных. У вас только 100 снимков MPT с опухолью (класс 1) и 1000 здоровых (класс 0). Нужно сгенерировать реалистичные изображения с опухолью для увеличения обучающей выборки. Какую модель вы выберете: GAN, VAE или диффузионную? Какие риски (режим коллапса, нереалистичные детали) и как их контролировать?

14. Обнаружение аномалий на графе банковских транзакций. У вас есть граф транзакций между счетами (узлы – счета, рёбра – переводы). Нужно найти мошеннические счета (аномалии). Предложите: а) какой тип GNN использовать (GCN, GAT, GraphSAGE); б) как адаптировать самоконтролируемое обучение (например, предсказание маскированных рёбер) при отсутствии разметки.

15. Мод коллапс в GAN для генерации лиц. Ваш DCGAN на датасете лиц генерирует только один тип лица (мужчина средних лет с бородой и очками). Диагностируйте проблему. Какие методы вы примените: мини-батчевая дискриминация, unrolled GAN, WGAN-GP, дифференцированная аугментация данных? Объясните, почему каждый метод может помочь.

16. VAE даёт размытые изображения. Вы используете VAE для генерации рукописных цифр, но изображения получаются размытыми, в отличие от GAN. Почему так происходит? Как модифицировать VAE (β -VAE, векторное квантование VQ-VAE, перцептуальная потеря) для улучшения резкости? Когда размытость допустима, а когда – нет?

17. Классификация узлов в очень большом графе (100 млн узлов). Социальный граф гигантского размера не помещается в память GPU целиком. Вы хотите применить GNN для предсказания интересов пользователей. Какие методы вы используете:

кластеризацию графа, Neighbor Sampling (GraphSAGE), кластерный GCN (Cluster-GCN)? Опишите пайплайн обучения для такого масштаба.

18. Генерация текста с контролируемым стилем
Задача: сгенерировать техническую документацию в двух стилях: «для экспертов» (с жаргоном) и «для новичков» (упрощённо). Как модифицировать генеративную модель (контролируемая генерация, адаптеры стиля, conditioning на эмбединг стиля)? Нужно ли дообучать всю модель или достаточно добавить небольшую подсеть?

19. Инференс на граничном устройстве (Raspberry Pi, 2 ГБ ОЗУ)
Вы обучили модель распознавания дорожных знаков (точность 97%, размер 500 МБ, время инференса 1.2 сек на GPU). Нужно развернуть её на Raspberry Pi для работы в реальном времени (требование <50 мс/кадр, память <300 МБ). Какие методы сжатия и оптимизации вы примените и в какой последовательности (квантование до int8, pruning, дистилляция, использование TensorRT Lite)?

20. Дрейф данных (data drift) в системе кредитного скоринга
Модель, обученная 2 года назад, стала хуже предсказывать дефолты. Вы обнаружили, что распределение доходов клиентов сместилось из-за инфляции. Как настроить мониторинг дрейфа (PSI, KLD) и что делать при его обнаружении: автоматический перезапуск обучения, взвешивание свежих примеров, онлайн обучение (online learning)? Каковы риски каждого подхода?

21. Распределённое обучение на 8 GPU даёт суб-линейное ускорение
Вы запустили DistributedDataParallel на 8 GPU (NVIDIA A100) для обучения Vision Transformer на ImageNet. Ускорение составило всего 4.5x вместо ожидаемых 8x. Где «узкое место»? Как профилировать (NVIDIA Nsight, PyTorch Profiler)? Какие шаги предпринять для улучшения масштабирования (увеличить batch size, использовать градиентную синхронизацию с компрессией, переключиться на FSDP)?

22. Оптимизация латентности в REST API с высоким RPS (1000 запросов/с)
Ваша модель развёрнута в Kubernetes, но при росте нагрузки латентность p95 растёт с 20 мс до 500 мс. Какие архитектурные решения вы предложите: динамический батчинг (dynamic batching), использование async-инференса, автоскейлинг подов, кэширование частых запросов, модель-конвейер с несколькими репликами? Обоснуйте выбор.

23. Сравнение ONNX Runtime и TensorRT для промышленного внедрения
В вашей компании используются как NVIDIA GPU, так и Intel CPU (для резервных серверов). Какую инфраструктуру для оптимизированного инференса вы построите: универсальный ONNX Runtime (с разными бэкендами) или TensorRT только для GPU + OpenVINO для CPU? Какие метрики (точность, скорость, совместимость) определяют решение?

24. Версионирование эксперимента в команде из 5 ML-инженеров
Два инженера независимо дообучают BERT на одном датасете, но используют разные гиперпараметры и случайные сиды. Через месяц никто не может воспроизвести лучший результат. Как внедрить MLOps-практики: DVC для данных, MLflow для экспериментов, DVC для моделей? Опишите конкретный workflow от запуска эксперимента до регистрации модели в реестре.

25. A/B тестирование новой модели на части пользователей
Вы обучили новую архитектуру (Transformer вместо LSTM) для рекомендательной системы. Запуск сразу для всех пользователей рискован. Как организовать canary-развертывание или A/B тест с трафиком 5% → 20% → 100%? Какие метрики нужно мониторить во время раскатки (latency, CTR, процент ошибок, потребление памяти) и на каких порогах останавливать развертывание?

Задания 3 типа

Задание 1. Напишите код функции attention(Q, K, V, mask=None) на PyTorch или NumPy. Покажите, как она работает на примере с Q, K, V размерности (2, 4, 8). Вычислите веса внимания для первого элемента батча (первые 4×4).

Что сдаётся: код + первые 4 строки матрицы весов.

Задание 2. Дан фрагмент кода обучения 10-слойной MLP на синтетических данных. Постройте графики нормы градиента для каждого слоя после 1 эпохи (код уже есть, нужно только вставить вызов `backward()` и `grad.norm()`). Ответьте: в каком слое градиент минимален? Какая функция активации (`sigmoid` или `ReLU`) даст более стабильные градиенты и почему?

Что сдаётся: график + вывод (2–3 предложения).

Задание 3. Загрузите `bert-base-uncased` и датасет из 200 твитов с разметкой тональности (прилагается). Обучите только классификационную голову (заморозив BERT) на 1 эпоху. Выведите ассигасу на валидации (50 примеров). Затем разморозьте последние 2 слоя BERT и обучите ещё 1 эпоху. Сравните ассигасу.

Что сдаётся: две цифры ассигасу + краткое объяснение, почему результат может отличаться.

Задание 4. Вам даны графики потерь дискриминатора и генератора для трёх запусков GAN на MNIST:

- Запуск А: дискриминатор падает до 0.1, генератор растёт до 10.
- Запуск Б: потери колеблются (0.5–1.5) без тренда.
- Запуск В: дискриминатор ≈ 0.7 , генератор ≈ 0.7 .

Какой из запусков демонстрирует мод коллапс? Почему? Предложите одно конкретное изменение в обучении для лечения проблемы (например, WGAN-GP, label smoothing). Обоснуйте выбор.

Что сдаётся: номер графика + название метода + обоснование (3–4 предложения).

Задание 5. Вам выданы результаты двух моделей на тестовой выборке Coqa:

- MLP (только признаки узлов): Ассигасу = 0.62 ± 0.02 .
- GCN (с учётом структуры графа): Ассигасу = 0.81 ± 0.01 .

Объясните разницу. Какой вывод о гомофильности графа можно сделать? Что такое гомофильность и как её измерить (формула)? Если бы граф был гетерофильным, мог бы MLP оказаться лучше? Почему?

Что сдаётся: развёрнутый ответ (5–7 предложений) + формула для измерения гомофильности.

Задание 6. Дан скрипт, который загружает ResNet-18 и прогоняет 100 изображений. Ваша задача:

1. Экспортировать модель в ONNX (написать код экспорта).
2. Замерить среднее время инференса в PyTorch (eager mode) и в ONNX Runtime.
3. Записать результаты в таблицу.

Что сдаётся: таблица с двумя значениями (мс) + указание, какой вариант быстрее и почему (убрали накладные расходы Python, оптимизация графа).

Задание 7. Напишите сервер FastAPI, который загружает любую предобученную модель (например, `sklearn.linear_model.LogisticRegression` для игрушечного примера). Реализуйте:

- `POST /predict`, принимающий JSON `{"features": [1.2, 3.4, ...]}` и возвращающий `{"prediction": 0}`.
- `GET /health`, возвращающий `{"status": "ok"}`.

Запустите сервер (локально) и отправьте один тестовый запрос через `requests`. Приложите скриншот терминала с ответом.

Что сдаётся: код сервера + скриншот успешного ответа.

Задание 8. Задача: классификация коротких текстов (до 100 токенов) на мобильном устройстве. Ограничения: модель ≤ 50 МБ, время инференса ≤ 10 мс на CPU. Даны 3 кандидата:

- BERT-base (440 МБ, 50 мс)
- DistilBERT (65 МБ, 15 мс)
- LSTM с 2 слоями (20 МБ, 8 мс)

Какую модель вы выберете? Если ни одна не подходит, предложите модификацию

(например, квантование DistilBERT до int8). Оцените новый размер и время (квантование уменьшает размер в 4 раза, ускоряет в 2–3 раза). Уложитесь ли в ограничения?

Что сдаётся: выбор + расчёт (размер, время) + краткое обоснование.

Задание 9. Вы обучаете ViT (Vision Transformer) на 4 GPU с batch size = 128. Без mixed precision одна эпоха занимает 120 секунд. С mixed precision (FP16) вычисления ускоряются в 2 раза. Однако накладные расходы на синхронизацию градиентов составляют 10% от времени вычислений (без mixed precision). Какое время одной эпохи будет при mixed precision? Во сколько раз ускорится обучение по сравнению с FP32?

Что сдаётся: формула, расчёт, конечный ответ (цифра ускорения).

Задание 10. Дано распределение признака возраст в обучающей выборке (1000 записей) и в новых данных за текущий месяц (1000 записей). Гистограммы (бины: 0–20, 21–35, 36–50, 51–100):

- Обучающая выборка: [100, 400, 300, 200]
- Новые данные: [250, 450, 200, 100]

Рассчитайте Population Stability Index (PSI) по формуле:

$$PSI = \sum (P_i - Q_i) \cdot \ln(P_i / Q_i)$$

где P – доля в обучающей, Q – доля в новых. PSI > 0.1 сигнализирует о значимом дрейфе. Есть ли дрейф? Какой бин внёс наибольший вклад?

Что сдаётся: таблица с расчётами по каждому бину + итоговый PSI + вывод (есть дрейф / нет дрейфа).