

**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
к.ф.-м.н. Жданов Э.Р.
« 01 » _____ 2026 года
М.П.



**Комплект оценочных материалов
по дисциплине «Технологии искусственного интеллекта»**

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.....	4
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 «Технологии искусственного интеллекта».

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 «Технологии искусственного интеллекта».

2. Результаты освоения дисциплины, подлежащие проверке

В результате освоения дисциплины обучающийся должен обладать следующими компетенциями

ПК-1	Создание информационных технологий нового поколения
ПК-1.1	Управление этапами жизненного цикла методологической и технологической инфраструктуры анализа больших данных в организации
ПК-1.2	Разработка и внедрение новых методов и технологий исследования больших данных

Код индикатора компетенции	Планируемые результаты обучения по дисциплине
ПК-1.1	<i>Знать:</i> - этапы жизненного цикла модели ИИ (от сбора данных до мониторинга); - требования к инфраструктуре для поддержки полного цикла ML-проектов. <i>Уметь:</i> - планировать и распределять ресурсы для создания и поддержки инфраструктуры ИИ; - выбирать подходящие инструменты для хранения, обработки и версионирования данных и моделей экспериментов. <i>Иметь практический опыт:</i> - проектирования простого пайплайна развертывания модели (CI/CD для ML); - работы с системами контроля версий для кода и данных (Git, DVC).
ПК-1.2	<i>Знать:</i> - современные state-of-the-art методы в областях NLP, Computer Vision и анализа табличных данных; - математические принципы, лежащие в основе этих методов; - критерии оценки эффективности новых подходов. <i>Уметь:</i> - адаптировать существующие передовые модели под специфику новой прикладной задачи; - реализовывать и обучать сложные нейросетевые архитектуры с нуля или на базе готовых решений; - проводить сравнительный анализ разработанного метода с базовыми линиями. <i>Иметь практический опыт:</i> - интеграции разработанного модуля в существующий аналитический комплекс.

3. Контроль и оценка освоения учебной дисциплины

Элемент учебной дисциплины	Текущий контроль		Промежуточная аттестация	
	Форма контроля	Проверяемые идк	Форма контроля	Проверяемые идк
Тема 1. Основы машинного обучения и инфраструктура проектов ИИ	Отчет по практикуму по решению задач	ПК-1.1		
Тема 2. Глубокое обучение: анализ изображений и временных рядов	Отчет по практикуму по решению задач	ПК-1.2		
Тема 3. Современные архитектуры и генеративный ИИ	Отчет по практикуму по решению задач	ПК-1.2		
Тема 4. Внедрение и управление моделями ИИ в организации	Отчет по практикуму по решению задач	ПК-1.1		
			Зачет	ПК-1.1 ПК-1.2

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
25-20 – 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; 19-14– 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; 13-8 – работа и отчет выполнены в срок, в основном самостоятельно, выводы сделаны частично, слабо аргументированы, даны ответы не на все вопросы; 7-1 – обучающийся подготовил работу и отчет несамостоятельно и/или не завершил в срок, описание содержит 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

Тема 1. Основы машинного обучения

Цель: научиться решать классическую задачу ML в среде Jupyter Notebook и настроить базовое логирование эксперимента.

Задача 1. Прогнозирование цен на недвижимость.

Используя предоставленный датасет (например, данные о недвижимости), необходимо решить задачу регрессии — предсказать стоимость объекта:

- провести исследовательский анализ данных (EDA). Выявить пропущенные значения и выбросы. Преобразовать категориальные признаки в числовой формат

(например, с помощью one-hot encoding).

- разделить данные на обучающую и тестовую выборки. Обучить несколько базовых моделей регрессии (линейная регрессия, решающее дерево, случайный лес) из библиотеки Scikit-learn.

- оценить качество предсказаний на тестовой выборке с помощью метрик MAE, MSE или R^2 . Сравнить результаты разных моделей и выбрать лучшую.

- настроить систему логирования эксперимента (например, с помощью стандартного модуля logging или MLflow) для записи параметров модели, используемых метрик и времени обучения.

Тема 2. Глубокое обучение: анализ изображений и временных рядов

Цель: получить опыт разработки нейронных сетей для задач компьютерного зрения и анализа последовательностей.

Задача 1. Классификатор рукописных цифр

Решить задачу многоклассовой классификации на датасете MNIST:

- спроектировать сверточную нейронную сеть (CNN) с использованием фреймворка PyTorch или TensorFlow/Keras. Модель должна включать слои свертки (Conv2D), пулинга (Pooling) и полносвязные слои (Dense).

- реализовать цикл обучения модели, используя функцию потерь (например, кросс-энтропию) и оптимизатор (например, Adam).

- построить графики изменения функции потерь и точности на обучающей и валидационной выборках от эпохи к эпохе.

Задача 2. Прогноз временного ряда

На основе синтетического или реального временного ряда (например, температуры или курса акций) построить прогноз на будущие шаги

- преобразовать одномерный временной ряд в набор образцов для обучения рекуррентной сети (создать окна из N предыдущих значений для предсказания $N+1$ -го).

- построить модель на базе LSTM-слоев для улавливания долгосрочных зависимостей в данных.

- обучить модель и сделать прогноз на заданное количество шагов вперед, визуализировав результат.

Тема 3. Современные архитектуры и генеративный ИИ

Цель: изучить работу трансформеров и основ генеративного моделирования.

Задача 1. Анализ тональности отзывов

Выполнить дообучение (fine-tuning) предобученной языковой модели на небольшом корпусе текстов (например, отзывы о фильмах или товарах):

- загрузить предобученную модель (например, DistilBERT) и токенизатор из библиотеки Hugging Face Transformers.

- добавить классификационный слой поверх модели и дообучить её на ваших данных для бинарной классификации (позитив/негатив).

- оценить точность полученной модели и сравнить её с базовой моделью, которая не проходила дообучение.

Задача 2. Генератор простых изображений

Реализовать простую генеративно-сопоставительную сеть (GAN) для генерации изображений:

- создать две нейросети: генератор (создает изображения из шума) и дискриминатор (отличает реальные изображения от сгенерированных).

- реализовать процесс их совместного обучения в режиме состязания, где генератор учится обманывать дискриминатор, а дискриминатор — распознавать подделку.

- после обучения продемонстрировать примеры изображений, сгенерированных сетью из случайного шума.

Тема 4. Внедрение и управление моделями ИИ в организации
Овладеть навыками подготовки модели к промышленному использованию и её развёртывания.

Задача 1. Развёртывание API для модели

Взять одну из ранее обученных моделей (например, классификатор изображений из Темы 2) и развернуть её как веб-сервис.

Сохранить веса обученной модели в файл.

Написать скрипт с использованием фреймворка FastAPI (или Flask), который загружает сохранённую модель при запуске и предоставляет эндпоинт (например, /predict), принимающий на вход данные для предсказания.

Создать Dockerfile для упаковки приложения и модели в Docker-контейнер.

Собрать образ контейнера и запустить его локально. Проверить работоспособность API, отправив ему запрос с данными и получив ответ с предсказанием.

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. Дайте определение машинного обучения (ML) и опишите основные типы задач, решаемых в рамках этого подхода: классификация, регрессия, кластеризация и снижение размерности.

2. Какие ключевые метрики используются для оценки качества моделей в задачах бинарной классификации и регрессии? Приведите примеры и объясните их смысл.

3. Опишите концепцию MLOps. Какие этапы жизненного цикла модели машинного обучения она охватывает и какова её основная цель?
4. Перечислите основные компоненты технологического стека для разработки проектов в области ИИ: языки программирования, фреймворки и системы управления базами данных.
5. Каковы основные этапы подготовки и решения задачи регрессии или классификации в среде Jupyter Notebook, включая настройку окружения и логирование эксперимента?
6. Опишите архитектуру сверточных нейронных сетей (CNN). Каковы их ключевые компоненты (свертка, пулинг, полносвязные слои) и для каких задач они применяются?
7. В чем заключается принцип работы рекуррентных нейронных сетей (RNN) и сетей LSTM? Как LSTM решает проблему исчезающего градиента при анализе длинных последовательностей?
8. Назовите основные проблемы обучения глубоких нейронных сетей (vanishing/exploding gradients, переобучение, нехватка данных) и методы их решения.
9. Опишите механизм внимания (Attention Mechanism) и его роль в архитектуре трансформеров. Как этот механизм изменил подход к обработке последовательностей по сравнению с RNN?
10. Сравните принципы работы и области применения языковых моделей BERT и GPT. В чем заключается разница между их архитектурами и типами решаемых задач?
11. Объясните архитектуру генеративно-сопоставительных сетей (GAN). Как устроено взаимодействие между генератором и дискриминатором в процессе обучения?
12. Какие этические аспекты необходимо учитывать при разработке и внедрении систем искусственного интеллекта (предвзятость, прозрачность, приватность данных)?
13. Опишите процесс файнтюнинга (fine-tuning) предобученной языковой модели для решения задачи анализа тональности текста.
14. Какие шаги включает в себя базовая реализация генератора изображений на основе нейронных сетей?
15. Что такое сериализация модели машинного обучения? Какие форматы (например, ONNX, pickle, joblib) используются для сохранения и загрузки моделей?
16. Каковы основные этапы подготовки обученной модели к продакшену с точки зрения оптимизации производительности и снижения ресурсоемкости?
17. Сравните фреймворки FastAPI и Flask с точки зрения создания REST API для развертывания моделей машинного обучения. В чем их преимущества и недостатки?
18. Опишите процесс контейнеризации модели ИИ с помощью Docker: от создания Dockerfile до сборки образа и запуска контейнера.
19. Какие метрики и методы используются для мониторинга производительности модели в продакшене, а также для обнаружения дрейфа данных (data drift)?
20. Опишите последовательность действий по упаковке обученной модели в Docker-контейнер и развертыванию REST API для предоставления доступа к её предсказаниям внешним сервисам.

Задания 2 типа

1. В компании внедрена модель кредитного скоринга, но за последний месяц точность предсказаний (ассигасу) резко упала, хотя метрика AUC-ROC остается стабильной. Как бы вы проанализировали ситуацию и какие метрики предложили бы для детального мониторинга?
2. Заказчик требует внедрить модель распознавания лиц на входе в офис. У вас есть небольшая выборка размеченных фотографий сотрудников. Опишите пошаговый план действий: от выбора архитектуры (CNN) до борьбы с переобучением и повышения точности.
3. Вы развернули модель прогнозирования спроса в виде Flask-приложения, но при 50 одновременных запросах API начинает отвечать с большой задержкой. Предложите

конкретные архитектурные и программные решения для масштабирования сервиса.

4. В задаче прогнозирования временных рядов (например, цен на акции) данные имеют сильную сезонность и тренд. Как бы вы подготовили данные, какую архитектуру (RNN/LSTM/Prophet) выбрали бы и как оценили качество модели?

5. При обучении GAN для генерации медицинских снимков (например, рентгеновских) наблюдается mode collapse — генератор выдает только несколько похожих изображений. Какие изменения в архитектуре или процессе обучения вы бы применили?

6. У вас есть предобученная языковая модель (BERT). Необходимо адаптировать её для задачи классификации отзывов о товарах на русском языке. Опишите процесс подготовки датасета, выбора гиперпараметров и оценки качества файнтюнинга.

7. Модель классификации изображений работает с ошибками из-за изменения освещения на новых тестовых данных, которых не было в обучающей выборке. Какие методы аугментации и регуляризации вы бы применили для повышения робастности модели?

8. В продакшене модель начала выдавать некорректные результаты. Анализ показал, что статистические свойства входных данных изменились (дрейф данных). Как бы вы организовали автоматический мониторинг дрейфа и процесс переобучения модели?

9. Вам нужно развернуть ансамбль из трех моделей (LightGBM, CatBoost, нейросеть) в виде единого API. Как организовать агрегацию их предсказаний (например, усреднение или стекинг) внутри FastAPI-приложения?

10. При разработке системы распознавания автомобильных номеров на видео с камер наблюдения видео имеет низкое разрешение и дрожание кадра. Какие подходы к предобработке видео и выбору архитектуры CNN вы бы использовали?

11. Модель для анализа тональности текста показывает высокую точность на обучающей выборке, но крайне низкую на реальных отзывах клиентов. Как бы вы диагностировали проблему переобучения и какие методы регуляризации применили?

12. Необходимо реализовать систему предсказания оттока клиентов телеком-оператора. Данные хранятся в распределенной СУБД, а признаков более 500, многие из них категориальные. Какой технологический стек (Python, Spark MLlib, XGBoost) вы выберете и почему?

13. Вы обучили сложную нейросеть для сегментации медицинских изображений, но её инференс занимает 2 секунды на одном изображении, что неприемлемо для реального времени. Опишите методы оптимизации модели (например, квантование, дистилляция знаний).

14. При разработке чат-бота на базе GPT ответы модели иногда содержат фактические ошибки или токсичный контент. Какие методы контроля генерации (prompt engineering, фильтрация) и этические ограничения вы бы внедрили?

15. В задаче классификации текстов на 1000 классов обучающая выборка сильно несбалансирована: 90% данных относятся к 5 основным классам. Как бы вы решали проблему дисбаланса классов при обучении модели?

16. Заказчик хочет отслеживать производительность модели в реальном времени через Grafana. Какую информацию о логах экспериментов и метриках инференса необходимо логировать в коде модели для передачи в систему мониторинга?

17. Вам необходимо перенести обученную модель из локального Jupyter Notebook в облачную инфраструктуру с использованием Kubernetes. Опишите процесс: от создания Docker-образа до написания манифестов развертывания (Deployment, Service).

18. В системе компьютерного зрения количество ложноположительных срабатываний (False Positives) критично (например, в медицинской диагностике), а ложноотрицательных — нет. Как бы вы изменили порог принятия решения и какие метрики приоритезировали бы?

19. При обучении LSTM-сети для прогнозирования временных рядов возникает проблема исчезающего градиента на длинных последовательностях, а увеличение числа слоев не помогает. Какие альтернативные архитектуры или техники обучения вы бы

рассмотрели?

20. Необходимо обеспечить воспроизводимость эксперимента по обучению модели через полгода после его проведения. Какие практики управления окружением (virtualenv/conda), версиями кода (Git) и зависимостями (requirements.txt, lock-файлы) вы внедрите?

Задания 3 типа

1. Настройте виртуальное окружение, установите необходимые библиотеки (например, scikit-learn, pandas, matplotlib), загрузите датасет (например, Iris или Titanic) и решите задачу бинарной/тривиальной классификации. Визуализируйте результаты и настройте базовое логирование эксперимента с помощью модуля logging.

2. Используя фреймворк PyTorch или TensorFlow/Keras, разработайте сверточную нейронную сеть (CNN) для классификации изображений на стандартном датасете (например, CIFAR-10). Реализуйте пайплайн: предобработка данных, аугментация, обучение модели и построение графиков точности/потерь.

3. Возьмите реальный временной ряд (например, цены акций, данные о погоде). Проведите разведочный анализ данных, создайте обучающую и тестовую выборки в формате временных окон. Реализуйте на PyTorch или TensorFlow рекуррентную сеть (LSTM) для прогнозирования будущих значений и постройте график сравнения прогноза с реальными данными.

4. Используя библиотеку Hugging Face Transformers, проведите фэнтюнинг предобученной модели BERT на собственном небольшом датасете для задачи анализа тональности текста (например, классификация отзывов о фильмах). Оцените качество модели с помощью метрик accuracy и F1-score.

5. Обучите любую модель машинного обучения (например, регрессор для предсказания цен на недвижимость). Упакуйте её в REST API с помощью фреймворка FastAPI. Реализуйте эндпоинты для получения предсказаний по HTTP-запросу (например, через POST с JSON-данными).

6. Для сервиса, созданного в задании №5, напишите Dockerfile, соберите Docker-образ и запустите контейнер. Убедитесь, что сервис доступен по сети и корректно возвращает предсказания модели.

7. С помощью PyTorch или TensorFlow реализуйте генеративно-сопоставительную сеть (GAN) для генерации простых изображений (например, цифр из датасета MNIST). Обучите модель и визуализируйте процесс генерации изображений по мере обучения.

8. Возьмите обученную нейросеть из задания №2 или №3. Сериализуйте её в формат ONNX. Проведите сравнительный анализ скорости инференса оригинальной модели и экспортированной в ONNX-формат.

9. Напишите скрипт на Python, который сравнивает статистические характеристики (среднее, стандартное отклонение) признаков на обучающей выборке и на данных, поступающих в продакшен в реальном времени. При обнаружении значимого отклонения скрипт должен записывать предупреждение в лог-файл.

10. Объедините навыки из предыдущих заданий: разработайте полный цикл обработки данных (загрузка, очистка), обучите модель (например, классификатор), сериализуйте её, создайте Flask/FastAPI-приложение для инференса, упакуйте его в Docker-контейнер и напишите простой скрипт мониторинга производительности.