

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Б1.В.20
(индекс дисциплины)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Практикум по машинному обучению и анализу данных

(наименование дисциплины)

по направлению подготовки
09.03.04 Программная инженерия

направленность (профиль)
Программная инженерия с применением ИИ-технологий

Форма обучения: заочная

Год набора: 2024

Общая трудоемкость: 4 ЗЕ

Распределение часов дисциплины по семестрам

| Семестр | 9 | Итого |
|--|------------|------------|
| Форма контроля | зачет | |
| Вид занятий | | |
| Лекции | 4 | 4 |
| Лабораторные | | |
| Практические | | |
| Руководство: курсовые работы (проекты) / РГР | | |
| Промежуточная аттестация | 0,25 | 0,25 |
| Контактная работа | 4,25 | 4,25 |
| Самостоятельная работа | 136 | 136 |
| Контроль | 3,75 | 3,75 |
| Итого | 144 | 144 |

Рабочую программу составил(и):

доцент института цифровых технологий, канд. техн. наук, Хрипунов Н.В.

(должность, ученое звание, степень, И.О. Фамилия)

Рецензирование рабочей программы дисциплины:



Отсутствует



Рецензент

(должность, ученое звание, степень, И.О. Фамилия)

Рабочая программа дисциплины составлена на основании ФГОС ВО и учебного плана направления подготовки

09.03.04 Программная инженерия

Срок действия рабочей программы дисциплины до «31» августа 2031 г.

УТВЕРЖДЕНО

На заседании института цифровых технологий

(протокол заседания № 1 от «05» сентября 2025 г.).

1. Цель освоения дисциплины

Цель – углубление знаний и совершенствование практических навыков по использованию систем искусственного интеллекта.

2. Место дисциплины в структуре ОПОП ВО

Дисциплины и практики, на освоении которых базируется данная дисциплина: «Алгоритмы и программирование на основе Python», «Системы искусственного интеллекта».

Дисциплины и практики, для которых освоение данной дисциплины необходимо как предшествующее: «Выполнение и защита выпускной квалификационной работы»

3. Планируемые результаты обучения

| Формируемые и контролируемые компетенции (код и наименование) | Индикаторы достижения компетенций (код и наименование) | Планируемые результаты обучения |
|--|--|--|
| ПК-2. Способен разрабатывать и отлаживать программный код на языках программирования | ПК-2.1. Знает основные правила конструирования и отладки программного кода | Знать: основные правила конструирования и отладки программного кода Уметь: применять основные правила конструирования и отладки программного кода, Владеть: навыками применения правил конструирования и отладки программного кода |
| | ПК-2.2. Умеет отлаживать программный код на языках программирования | Знать: языки программирования Уметь: отлаживать программный код на языках программирования Владеть: навыками отладки программного кода на языках программирования |
| | ПК-2.3. Владеет навыками разработки и отладки программного кода на языках программирования | Знать: языки программирования Уметь: разрабатывать программный код на языках программирования Владеть: навыками разработки программного кода на языках программирования |

4. Структура и содержание дисциплины

| Модуль (раздел) | Вид учебной работы | Наименование тем занятий (учебной работы) | Семестр | Объем, ч. | Баллы | Интерактив, ч. | Формы текущего контроля (наименование оценочного средства) |
|---|--------------------|---|---------|-----------|-------|----------------|--|
| Модуль 1 – Введение в искусственный интеллект, нейронные сети и машинное обучение | Лек 1 | Тема 1. Введение в дисциплину (часть 1). | 9 | 2 | - | - | |
| | СР | Самостоятельное изучение литературы по курсу при подготовке к практическим работам. | 9 | 4 | - | - | |
| Модуль 2 - Задачи классификации | Лек 2 | Тема 2. Проектирование нейросетевых архитектур | 9 | 2 | - | - | |
| | СР | Самостоятельное изучение литературы по курсу при подготовке к практическим работам. | 9 | 4 | - | - | |
| | СР | Пр 1. Бинарный классификатор на полносвязной НС. | 9 | 4 | 3 | - | Отчет по практической работе 1 |
| | СР | Пр 2. Многоклассовый классификатор на полносвязной НС. | 9 | 4 | 3 | - | Отчет по практической работе 2 |
| | СР | Пр 3. Многоклассовый классификатор на сверточной НС. | 9 | 4 | 3 | - | Отчет по практической работе 3 |
| | СР | Пр 4. Предварительный анализ данных. Регрессия, ассоциация, аномалии и визуализация. На примере датасета Титаник. | 9 | 4 | 3 | - | Отчет по практической работе 4 |
| | СР | Пр 5. Исследование влияния качества датасета на работу бинарного классификатора (часть 1). | 9 | 4 | 3 | - | Отчет по практической работе 5 |
| | СР | Пр 5. Исследование влияния качества | 9 | 4 | - | - | |

| Модуль (раздел) | Вид учебной работы | Наименование тем занятий (учебной работы) | Семестр | Объем, ч. | Баллы | Интерактив, ч. | Формы текущего контроля (наименование оценочного средства) |
|---|--------------------|--|---------|-----------|-------|----------------|--|
| | | датасета на работу бинарного классификатора (часть 2). | | | | | |
| | СР | Пр 6. Анализ произвольного табличного датасета. | 9 | 4 | 3 | - | Отчет по практической работе 6 |
| | СР | Пр 7. Исследование и визуализация произвольного табличного датасета. | 9 | 4 | 3 | - | Отчет по практической работе 7 |
| | СР | Пр 8. Применение бинарного классификатора к произвольному табличному датасету (часть 1). | 9 | 4 | 3 | - | Отчет по практической работе 8 |
| | СР | Пр 8. Применение бинарного классификатора к произвольному табличному датасету (часть 2). | 9 | 4 | - | - | |
| | СР | Пр 8. Применение бинарного классификатора к произвольному табличному датасету (часть 3). | 9 | 4 | - | - | |
| Модуль 3 – Обработка естественного языка | СР | Тема 3. Введение в обработку естественного языка | 9 | 4 | - | - | |
| | СР | Тема 4. Векторное представление слов. Word2vec | 9 | 4 | - | - | |
| | СР | Самостоятельное изучение литературы по курсу при подготовке к практическим работам. | 9 | 4 | - | - | |
| | СР | Пр 9. Анализ английского текста (часть 1). | 9 | 4 | 3 | - | Отчет по практической работе 9 |
| | СР | Пр 9. Анализ английского текста (часть 2). | 9 | 4 | - | - | |
| | СР | Пр 9. Анализ английского текста (часть 3). | 9 | 4 | - | - | |
| | СР | Пр 10. Анализ русского текста (часть 1). | 9 | 4 | 3 | - | Отчет по |

| Модуль (раздел) | Вид учебной работы | Наименование тем занятий (учебной работы) | Семестр | Объем, ч. | Баллы | Интерактив, ч. | Формы текущего контроля (наименование оценочного средства) |
|-----------------|--------------------|--|---------|-----------|-------|----------------|--|
| | СР | Пр 10. Анализ русского текста (часть 2). | 9 | 4 | - | - | практической работе 10 |
| | СР | Пр 10. Анализ русского текста (часть 3). | 9 | 4 | - | - | |
| | СР | Пр 11. Анализ тональности русского текста (часть 1). | 9 | 4 | 3 | - | Отчет по практической работе 11 |
| | СР | Пр 11. Анализ тональности русского текста (часть 2). | 9 | 2 | - | - | |
| | СР | Пр 11. Анализ тональности русского текста (часть 3). | 9 | 2 | - | - | |
| | СР | Пр 12. Классификация предложений с помощью сверточной нейросети (часть 1). | 9 | 2 | 3 | - | Отчет по практической работе 12 |
| | СР | Пр 12. Классификация предложений с помощью сверточной нейросети (часть 2). | 9 | 2 | - | - | |
| | СР | Пр 12. Классификация предложений с помощью сверточной нейросети (часть 3). | 9 | 2 | - | - | |
| | СР | Пр 13. Генерация текста с помощью рекуррентной нейросети (часть 1). | 9 | 2 | 3 | - | Отчет по практической работе 13 |
| | СР | Пр 13. Генерация текста с помощью рекуррентной нейросети (часть 2). | 9 | 2 | - | - | |
| | СР | Пр 13. Генерация текста с помощью рекуррентной нейросети (часть 3). | 9 | 2 | - | - | |
| | СР | Пр 14. Применение LSTM для генерации текста. | 9 | 2 | 3 | - | Отчет по практической работе 14 |
| | СР | Пр 15. Сравнение качества текстов на выходе разных модификаций LSTM . | 9 | 2 | 3 | - | Отчет по практической работе 15 |

| Модуль (раздел) | Вид учебной работы | Наименование тем занятий (учебной работы) | Семестр | Объем, ч. | Баллы | Интерактив, ч. | Формы текущего контроля (наименование оценочного средства) |
|---|--------------------|--|---------|-----------|-------|----------------|--|
| Модуль 4 – Распознавание образов. Машинное зрение | СР | Тема 5. Основы теории распознавания образов | 9 | 2 | - | - | |
| | СР | Самостоятельное изучение литературы по курсу при подготовке к практическим работам. | 9 | 2 | - | - | |
| | СР | Пр 16. Анализ произвольного датасета изображений (часть 1). | 9 | 2 | 3 | - | Отчет по практической работе 16 |
| | СР | Пр 16. Анализ произвольного датасета изображений (часть 2). | 9 | 2 | - | - | |
| | СР | Пр 16. Анализ произвольного датасета изображений (часть 3). | 9 | 2 | - | - | |
| | СР | Пр 17. Применение бинарного классификатора к произвольному датасету изображений (часть 1). | 9 | 2 | - | - | Отчет по практической работе 17 |
| | СР | Пр 17. Применение бинарного классификатора к произвольному датасету изображений (часть 2). | 9 | 2 | - | - | |
| | СР | Пр 17. Применение бинарного классификатора к произвольному датасету изображений (часть 3). | 9 | 2 | - | - | |
| | СР | Пр 18. Интеллектуальная обработка видео (часть 1). | 9 | 2 | 2 | - | Отчет по практической работе 18 |
| | СР | Пр 18. Интеллектуальная обработка видео (часть 2). | 9 | 2 | - | - | |
| | СР | Пр 18. Интеллектуальная обработка видео (часть 3). | 9 | 2 | - | - | |
| Модуль 5 - Предиктивная | СР | Тема 6. Основы теории прогнозирования временных рядов | 9 | 2 | - | - | |

| Модуль (раздел) | Вид учебной работы | Наименование тем занятий (учебной работы) | Семестр | Объем, ч. | Баллы | Интерактив, ч. | Формы текущего контроля (наименование оценочного средства) |
|---------------------------|--------------------|--|---------|-----------|-------|----------------|--|
| аналитика временных рядов | СР | Самостоятельное изучение литературы по курсу при подготовке к практическим работам. | 9 | 2 | - | - | |
| | СР | Пр 19. Предсказание цены актива на основании графика изменения цены с использованием нейросети на 1 шаг (часть 1). | 9 | 2 | 2 | - | Отчет по практической работе 19 |
| | СР | Пр 19. Предсказание цены актива на основании графика изменения цены с использованием нейросети на 1 шаг (часть 2). | 9 | 2 | - | - | |
| | СР | Пр 19. Предсказание цены актива на основании графика изменения цены с использованием нейросети на 1 шаг (часть 3). | 9 | 2 | - | - | |
| | СР | Пр 20. Предсказание цены актива на основании графика изменения цены с использованием нейросети на несколько шагов (часть 1). | 9 | 2 | 2 | - | Отчет по практической работе 20 |
| | СР | Пр 20. Предсказание цены актива на основании графика изменения цены с использованием нейросети на несколько шагов (часть 2). | 9 | 2 | - | - | |
| | СР | Пр 20. Предсказание цены актива на основании графика изменения цены с использованием нейросети на несколько шагов (часть 3). | 9 | 2 | - | - | |
| | СР | Пр 21. Предсказание цены актива на | 9 | 2 | 2 | - | Отчет по |

| Модуль (раздел) | Вид учебной работы | Наименование тем занятий (учебной работы) | Семестр | Объем, ч. | Баллы | Интерактив, ч. | Формы текущего контроля (наименование оценочного средства) |
|-----------------|--------------------|---|---------|-----------|-------|----------------|--|
| | | основании графика изменения цены, объёма торгов и цены связанного актива(индекса) с использованием нейросети на 1 шаг (часть 1). | | | | | практической работе 21 |
| | СР | Пр 21. Предсказание цены актива на основании графика изменения цены, объёма торгов и цены связанного актива(индекса) с использованием нейросети на 1 шаг (часть 2). | 9 | 2 | - | - | |
| | СР | Пр 21. Предсказание цены актива на основании графика изменения цены, объёма торгов и цены связанного актива(индекса) с использованием нейросети на 1 шаг (часть 3). | 9 | 2 | - | - | |
| | СР | Пр 22. Предсказание цены актива на основании графика изменения цены, объёма торгов и цены связанного актива(индекса) с использованием нейросети на несколько шагов. | 9 | 2 | 2 | - | Отчет по практической работе 22 |
| | ПА | Промежуточная аттестация | 9 | 0,25 | - | - | |
| | Контроль | Зачет | 9 | 8,75 | 40 | - | Итоговый тест |
| Итого: | | | | 144 | 100 | | |

5. Образовательные технологии

В рамках изучения дисциплины «Программирование для задач искусственного интеллекта и анализа данных» предусмотрено использование следующих образовательных технологий:

- технология традиционного обучения: лекции и практические работы, самостоятельная работа;
- технология проектного обучения: реализация и защита отчетов по практическим работам.

6. Методические указания по освоению дисциплины

6.1. Рекомендации по подготовке к лекционным занятиям

Изучение дисциплины требует систематического и последовательного накопления знаний, следовательно, пропуски отдельных тем не позволяют глубоко освоить предмет.

В ходе лекционных следует обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации. Задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций.

Обучающийся может дополнить список использованной литературы современными источниками, не представленными в списке рекомендованной литературы, и в дальнейшем использовать собственные подготовленные учебные материалы при написании курсовых и выпускных квалификационных работ.

6.2. Рекомендации по подготовке к практическим занятиям

Обучающимся следует доводить каждую практическую работу до окончательного решения, демонстрировать понимание проведенных расчетов (анализов, ситуаций), в случае затруднений обращаться к преподавателю.

Для того чтобы практические занятия приносили максимальную пользу, необходимо помнить, что упражнение и решение задач проводятся по рассмотренному на лекциях материалу и связаны, как правило, с детальным разбором отдельных вопросов лекционного курса. Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться обучающимся на практических занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью решения проблемных ситуаций, задач. При этих условиях Обучающийся не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (и это очень важно) для активной проработки лекции.

По результатам выполнения работы составляется отчет, который при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что выполнение каждой работы должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Полученный ответ следует проверить способами, вытекающими из существа данной задачи. Полезно также (если возможно) решать несколькими способами и сравнить полученные результаты. Решение задач данного типа нужно продолжать до приобретения твердых навыков в их решении.

6.3. Рекомендации по подготовке к зачету

Подготовка к зачету способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к зачету, обучающийся ликвидирует имеющиеся пробелы в знаниях, углубляет,

систематизирует и упорядочивает свои знания. На зачёте обучающийся демонстрирует то, что он приобрел в процессе обучения по конкретной учебной дисциплине.

Необходимо ориентировать обучающихся на систематическую подготовку к занятиям в течение семестра, что позволит использовать время экзаменационной сессии для систематизации знаний.

7. Оценочные средства

7.1. Паспорт оценочных средств

| Семестр | Код контролируемой компетенции (или ее части) | Наименование оценочного средства |
|----------------|--|--|
| 9 | ПК-2 | Отчеты по практическим работам 1-22 Вопросы к зачету 1-50 |

7.2. Типовые задания или иные материалы, необходимые для текущего контроля

7.2.1. Пример практических работ

Практическая работа № 1. Бинарный классификатор на полносвязной НС

Цель: построение бинарного классификатора с помощью полносвязной нейронной сети, определяющий тип цитрусовых

Задание.

1. Установите все необходимые библиотеки и импортируйте их в проект
2. Выполните считывание и просмотр набора данных
3. Подготовьте данные. Проверьте, что данный набор данных не содержит пропущенных значений и выбросов.

Убедитесь, что в данном наборе данных, классы сбалансированы

Выполните кодирование меток целевого класса цифрами: 1 — это апельсин, 0 — это грейпфрут

Разбейте всю выборку на обучающую и тестовую.

Выполните стандартизацию

4. Задайте модель полносвязной нейронной сети с использованием фреймворка Keras

В качестве модели предлагается использовать модель с тремя полносвязными слоями. Первому слою должна передаваться форма входных данных. Используйте функции активации для первого и второго слоя `relu`. Функция активации выходного слоя должна возвращать вероятность принадлежности экземпляра целевому классу.

Настройте процесс обучения. Задайте оптимизатор и функция(-и) потерь, которые должны использоваться моделью, а также все метрики для мониторинга во время обучения

5. Выполните обучение модели на 20 эпохах, 20 процентов обучающей выборки будет использовано для оценки модели, во время обучения, размер пакета 32.

Постройте предсказания и визуализируйте их. Постройте график для визуализации изменения оценки модели на разных эпохах.

Практическая работа № 2. Многоклассовый классификатор на полносвязной НС

Цель: Разработать и обучить многоклассовый классификатор с помощью полносвязной нейронной сети (MLP) для распознавания рукописных цифр из набора данных MNIST.

Задание.

1. Установка библиотек и импорт:

Установите все необходимые библиотеки: tensorflow (или keras), numpy, matplotlib, sklearn.

Импортируйте их в проект: tensorflow.keras.models, tensorflow.keras.layers, sklearn.model_selection, sklearn.preprocessing, numpy, matplotlib.pyplot.

2. Загрузка и обзор данных MNIST:

Загрузите набор данных MNIST, используя встроенные функции keras.datasets.mnist.load_data().

Просмотрите размерность загруженных данных (количество изображений, размеры изображений).

Визуализируйте несколько примеров изображений рукописных цифр вместе с их истинными метками.

3. Подготовка данных:

Проверка на пропуски и выбросы: Набор MNIST обычно не содержит пропущенных значений. Убедитесь в этом. Выбросы в контексте изображений маловероятны, но можно проверить, что значения пикселей находятся в допустимом диапазоне.

Сбалансированность классов: Набор MNIST хорошо сбалансирован по классам (каждая цифра представлена примерно одинаковым количеством примеров). Убедитесь в этом, посчитав количество примеров для каждой цифры.

Кодирование меток: Истинные метки (0-9) уже являются числовыми. Для многоклассовой классификации с помощью полносвязной сети, использующей функцию активации softmax в выходном слое, one-hot encoding (прямое кодирование) является предпочтительным. Преобразуйте метки в формат one-hot, где, например, цифра '3' будет представлена как [0, 0, 0, 1, 0, 0, 0, 0, 0, 0].

Нормализация пикселей: Преобразуйте значения пикселей изображений (которые обычно находятся в диапазоне 0-255) в диапазон [0, 1] или [-1, 1], разделив на 255.0.

Изменение формы данных: Плоские (flatten) изображения MNIST (28x28 пикселей) в одномерный вектор (784 элемента).

Разделение на выборки: Разделите весь набор данных на обучающую (train) и тестовую (test) выборки. В большинстве случаев MNIST уже поставляется разделенным, но убедитесь, что у вас есть отдельный набор для финальной оценки.

4. Построение модели полносвязной нейронной сети (MLP):

Используйте фреймворк Keras для создания модели.

Создайте модель типа Sequential.

Добавьте входной слой: InputLayer или Dense слой, указывающий input_shape=(784,).

Добавьте как минимум два скрытых полносвязных (Dense) слоя. Используйте функцию активации relu для скрытых слоев.

Добавьте выходной слой: Dense слой с количеством нейронов, равным количеству классов (10 для MNIST). Используйте функцию активации softmax, которая будет выдавать вероятности принадлежности к каждому из 10 классов.

Настройка процесса обучения:

- Оптимизатор: Используйте Adam или SGD.
- Функция потерь: Для многоклассовой классификации с one-hot encoded метками используйте categorical_crossentropy.
- Метрики: Задайте метрику accuracy для мониторинга во время обучения.

5. Обучение модели:

Обучите модель на 20-30 эпохах.

Установите размер пакета (batch size) равным 32 или 64.

Используйте 20% обучающей выборки для валидации (параметр validation_split).

Сохраните историю обучения (результаты эпох).

6. Оценка и визуализация:

Сделайте предсказания на тестовой выборке.

Постройте график изменения точности (accuracy) и функции потерь (categorical_crossentropy) на обучающей и валидационной выборках по эпохам.

Визуализируйте несколько предсказанных изображений из тестовой выборки, показывая исходное изображение, истинную метку и предсказанную метку.

Оцените общую точность модели на тестовой выборке.

Практическая работа № 3. Многоклассовый классификатор на сверточной НС

Цель: Разработать и обучить многоклассовый классификатор на основе сверточной нейронной сети (CNN) для распознавания рукописных цифр из набора данных MNIST.

Задание:

1. Установка библиотек и импорт:

Установите все необходимые библиотеки: tensorflow (или keras), numpy, matplotlib, sklearn.

Импортируйте их в проект: tensorflow.keras.models, tensorflow.keras.layers, sklearn.model_selection, sklearn.preprocessing, numpy, matplotlib.pyplot.

2. Загрузка и обзор данных MNIST:

Загрузите набор данных MNIST, используя встроенные функции keras.datasets.mnist.load_data().

Просмотрите размерность загруженных данных.

Визуализируйте несколько примеров изображений рукописных цифр вместе с их истинными метками.

3. Подготовка данных:

Проверка на пропуски и выбросы: Аналогично работе №2.

Сбалансированность классов: Аналогично работе №2.

Кодирование меток: Примените one-hot encoding к меткам целевого класса (10 классов, 0-9).

Нормализация пикселей: Преобразуйте значения пикселей в диапазон [0, 1].

Изменение формы данных: Для сверточных сетей данные должны иметь форму, соответствующую каналам изображения. Для MNIST это будет (количество_изображений, высота, ширина, количество_каналов). Для черно-белого изображения MNIST это (N, 28, 28, 1).

Разделение на выборки: Разделите весь набор данных на обучающую (train) и тестовую (test) выборки.

4. Построение модели сверточной нейронной сети (CNN):

Используйте фреймворк Keras для создания модели.

Создайте модель типа Sequential.

Сверточные слои: Добавьте как минимум два блока свертки (Conv2D) с последующим слоем пулинга (MaxPooling2D).

Conv2D: Выберите количество фильтров (например, 32, 64) и размер ядра (например, (3,3)). Используйте функцию активации relu.

MaxPooling2D: Используйте размер окна пулинга (например, (2,2)).

Слой сглаживания (Flatten): После сверточных блоков добавьте слой Flatten для преобразования двумерных карт признаков в одномерный вектор.

Полносвязные слои: Добавьте один или два полносвязных (Dense) слоя с функцией активации relu.

Выходной слой: Добавьте выходной Dense слой с 10 нейронами и функцией активации softmax.

Настройка процесса обучения:

- Оптимизатор: Используйте Adam.
- Функция потерь: categorical_crossentropy.
- Метрики: accuracy.

5. Обучение модели:

Обучите модель на 20-30 эпохах.

Установите размер пакета (batch size) равным 32 или 64.

Используйте 20% обучающей выборки для валидации.

Сохраните историю обучения.

6. Оценка и визуализация:

Сделайте предсказания на тестовой выборке.

Постройте график изменения точности (accuracy) и функции потерь (categorical_crossentropy) на обучающей и валидационной выборках по эпохам.

Визуализируйте несколько предсказанных изображений из тестовой выборки, показывая исходное изображение, истинную метку и предсказанную метку.

Оцените общую точность модели на тестовой выборке и сравните ее с точностью, достигнутой в работе №2 (MLP).

Практическая работа № 4. Предварительный анализ данных. Регрессия, ассоциация, аномалии и визуализация. На примере датасета Титаник

Цель: Выполнить предварительный анализ данных

Задание.

Датасет "Титаник" содержит информацию о пассажирах корабля "Титаник", включая их демографические данные, класс билета, возраст, пол, количество родственников на борту и факт выживания.

Доступность в библиотеке Seaborn (sns.load_dataset('titanic'))

Оценить объем и качество данных (номенклатура, тип данных, пропущенные значения, выбросы, избыточность).

Практическая работа № 5. Исследование влияния качества датасета на работу бинарного классификатора

Цель: Исследовать влияние качества данных на результат работы нейронной сети

Задание.

1. Внести искажения в датасет Титаник.

Пропущенные значения:

Выбросы (Аномальные значения):

Дубликаты (Повторяющиеся записи):

Неконсистентность

Перед внесением искажений необходимо сохранить копию исходного датасета, чтобы можно было сравнить результаты анализа до и после внесения искажений.

Цель состоит не в том, чтобы полностью испортить датасет, а в том, чтобы внести реалистичные искажения, которые могут встречаться в реальных данных.

2. Анализ качества данных: Напишите Python-код, который выявляет и подсчитывает следующие проблемы в датасете (до и после внесения искажений):

Количество пропущенных значений в каждом столбце.

Количество выбросов в столбце fare (определите порог выброса, например, с помощью метода IQR).

Количество выбросов в столбце age (значения больше 80).

Количество дубликатов.

Количество неконсистентных записей "возраст - выживание".

3. Визуализация данных:

Постройте гистограмму для столбца age (с учетом пропущенных значений и выбросов).

Постройте столбчатую диаграмму для столбца embarked (с учетом пропущенных значений).

Используйте boxplot для визуализации выбросов в столбце fare.

4. Анализ категориальных признаков:

Постройте столбчатую диаграмму для столбца `sex` с разделением по `survived` (показывает зависимость выживаемости от пола).

Постройте столбчатую диаграмму для столбца `pclass` с разделением по `survived` (показывает зависимость выживаемости от класса билета).

5. Формулировка задачи МО:

Определит задачу машинного обучения, которую можно решить с использованием этого датасета до внесения искажений (например, предсказание выживания на основе признаков пассажира).

Указать целевую переменную (`survived`).

Предложить метрику оценки, подходящую для выбранной задачи (например, `accuracy`, `precision`, `recall`, `F1-score`).

6. Спроектировать и обучить нейросеть 0 для предсказания выживания пассажира Титаника на базе исходного датасета.

Спроектировать и обучить нейросеть 1 для предсказания выживания пассажира Титаника на базе искаженного датасета.

Реализовать работу с нейросетями в диалоговом режиме – пользователь вводит данные пассажира и получает ответ, например, такой: Нейросеть 0 – выплыл, Нейросеть 1 – утонул

Проанализировать процесс и качество обучение нейросетей 0 и 1

Провести 5 вычислительных экспериментов в центре и с предельными значениями числовых (возраст) или квазичисловых (класс билета 1,2,3) параметров по каждому измерению. В каждом вычислительно эксперименте выбирать и поддерживать постоянным новое сочетание нечисловых параметров (пол, порт посадки и т.п.)

Практическая работа № 6. Анализ произвольного табличного датасета

Цель: Анализ качества данных и внесение искажений в произвольный датасет

Задание.

1. На ресурсе Kaggle (или другом) выполнить поиск датасета для задач бинарной классификации по табличным признакам. Согласовать найденный датасет с преподавателем.

Примеры датасетов

Heart Disease UCI (или другие вариации этого датасета): Данные о сердечных заболеваниях. Признаки: Возраст, пол, кровяное давление, уровень холестерина, результаты ЭКГ, наличие стенокардии. Задача: Предсказать наличие сердечного заболевания (0 или 1).

Breast Cancer Wisconsin (Diagnostic) Data Set: Характеристики опухолей молочной железы. Признаки: Различные параметры клеток (размер, форма, текстура, гладкость, компактность, симметрия, фрактальная размерность). Задача: Классифицировать опухоль как злокачественную (`malignant`) или доброкачественную (`benign`).

Credit Card Fraud Detection: Транзакции по кредитным картам. Признаки: Анонимизированные признаки (V1-V28), время транзакции, сумма транзакции. Задача: Определить, является ли транзакция мошеннической (0 или 1).

Diabetes Dataset (Pima Indians Diabetes Database): Данные о женщинах из племени Пима и наличие у них диабета. Признаки: Количество беременностей, концентрация глюкозы в плазме, кровяное давление, толщина кожной складки трицепса, уровень инсулина, индекс массы тела, генеалогическая функция диабета, возраст. Задача: Предсказать, есть ли у женщины диабет (0 или 1).

2. Дать общее описание датасета – какие данные содержит, как может быть использован для бинарной классификации.

3. Провести анализ данных

Выведите первые несколько строк датасета (`head()`).

Выведите общую информацию о датасете (`info()`), чтобы узнать типы данных и количество ненулевых значений в каждом столбце.

Описать качество данных – пропущенные значения, дублирующие столбцы и т.п.

Используя `describe()` вывести статистические показатели для числовых столбцов датафрейма.

Провести анализ качества числовых данных.

4. Сформулировать общие выводы

Практическая работа № 7. Исследование и визуализация произвольного табличного датасета

Цель: Исследование и визуализация зависимостей произвольного датасета

Задание.

Использовать датасет работы 6

1. Сформулировать и обосновать не менее 5 гипотез относительно данных датасета

2. Исследовать датасет на предмет подтверждения или опровержения гипотез.

Результаты представить в виде текстового описания, диаграмм и графиков.

3. Написать Заключение

Пример для датасета Титаника (фрагмент)

1. Гипотезы

Гипотеза 1: Более высокий класс каюты (Pclass) увеличивает вероятность выживания.

Гипотеза 2: Женщины выживали чаще, чем мужчины.

Гипотеза 3: Пассажиры с большим количеством родственников на борту имели меньше шансов на выживание.

Гипотеза 4: Молодые пассажиры (дети) выживали чаще, чем пожилые.

Гипотеза 5: Более высокая цена билета (Fare) коррелирует с более высокой вероятностью выживания.

Гипотеза 6: Порт посадки влияет на вероятность выживания

2. Исследование датасета и проверка гипотез

Используются библиотеки Pandas, Matplotlib и Seaborn.

...

Проверка гипотез:

Гипотеза 1: Более высокий класс каюты (Pclass) увеличивает вероятность выживания.

...

Мы видим, что выживаемость значительно выше для пассажиров 1-го класса по сравнению со 2-м и 3-м классами. Это подтверждает гипотезу.

Гипотеза 2: Женщины выживали чаще, чем мужчины.

...

Женщины выживали значительно чаще, чем мужчины. Это также подтверждает гипотезу.

Гипотеза 3: Пассажиры с большим количеством родственников на борту имели меньше шансов на выживание.

...

График показывает, что выживаемость сначала увеличивается с ростом размера семьи (до определенного момента), а затем снижается для очень больших семей. Таким образом, гипотеза о том, что только большие семьи имеют меньшую выживаемость, не совсем верна. Средние по размеру семьи (2-4 человека) имели больше шансов на выживание.

...

Заключение

На основе анализа датасета Titanic, несколько гипотез были подтверждены. В частности, установлено, что более высокий класс каюты (Pclass) значительно увеличивал шансы на выживание. Женщины выживали значительно чаще, чем мужчины, что, вероятно, связано с приоритетной эвакуацией. Молодые пассажиры также имели несколько более высокую выживаемость. Существует корреляция между ценой билета и выживаемостью, указывающая на то, что более богатые пассажиры имели больше шансов выжить. Размер

семьи также влиял на выживаемость, но не так просто, как предполагалось изначально. Пассажиры с небольшими и средними семьями имели больше шансов на выживание, чем одинокие или пассажиры из очень больших семей. Порт посадки также влиял на выживаемость.

Практическая работа № 8. Применение бинарного классификатора к произвольному табличному датасету

Цель: Использовать бинарный классификатор для проверки гипотез

Задание.

Проверить гипотезы работы 7 с использованием нейронных сетей.

Практическая работа № 9. Анализ английского текста

Цель: Выполнить начальную обработку английского текста с использованием NLP инструментов

Задание.

0. Найти и закрепить за собой какую-либо книгу в текстовом файле. Книга должна быть и на русском и на английском языках.

1. Загрузить английскую книгу

2. Выполнить подсчет слов

3. Выполнить подсчет частоты встречаемости слов, вывести первые 10

4. Использовать NLTK для токенизации и удаления стоп-слов. Выполнить подсчет частоты встречаемости слов, вывести первые 10. Сравнить результат с п.3

5. Выполнить лемматизацию. Выполнить подсчет частоты встречаемости слов, вывести первые 10. Сравнить результат с п.3 и 4

6. Выполнить анализ тональности и нарисовать облако слов

Практическая работа № 10. Анализ русского текста

Цель: Выполнить начальную обработку русского текста с использованием NLP инструментов

Задание.

0. Использовать книгу на русском языке с работы 9.

1. Загрузить русскую книгу

2. Выполнить подсчет слов

3. Выполнить подсчет частоты встречаемости слов, вывести первые 10

4. Использовать NLTK для токенизации и удаления стоп-слов. Выполнить подсчет частоты встречаемости слов, вывести первые 10. Сравнить результат с п.3

5. Выполнить лемматизацию. Выполнить подсчет частоты встречаемости слов, вывести первые 10. Сравнить результат с п.3 и 4

6. Нарисовать облако слов

Практическая работа № 11. Анализ тональности русского текста

Цель: Выполнить анализ тональности русского текста

Задание.

0. Использовать книгу на русском языке с работы 10.

1. Загрузить русскую книгу

2. Анализ тональности первых 12 предложений с выводом результата

3. Выполнить анализ тональности всего текста с выводом суммарного процента

Практическая работа № 12. Классификация предложений с помощью сверточной нейросети

Цель: Разработать и обучить сверточную нейронную сеть (CNN) для классификации текстовых предложений по их семантической принадлежности к различным категориям

Задание.

1. Установка библиотек и импорт:

Установите необходимые библиотеки: tensorflow (или keras), numpy, pandas, nltk (или gensim, spacy для предобработки текста), sklearn.

Импортируйте их: tensorflow.keras.models, tensorflow.keras.layers, tensorflow.keras.preprocessing.text, tensorflow.keras.preprocessing.sequence, numpy, pandas, nltk, sklearn.model_selection, sklearn.preprocessing.

2. Загрузка и обзор набора данных:

Выберите подходящий набор данных для классификации предложений (например, IMDB dataset для анализа тональности, 20 Newsgroups для тематической классификации, или другой датасет с соответствующими метками).

Загрузите данные (например, из CSV файла с помощью pandas).

Изучите структуру данных: тексты предложений и их соответствующие метки классов.

3. Предобработка текста:

Токенизация: Разбейте каждое предложение на отдельные слова (токены).

Очистка текста: Удалите знаки препинания, специальные символы, приведите текст к нижнему регистру.

Удаление стоп-слов: Удалите часто встречающиеся слова, не несущие значимого смысла (например, "the", "a", "is").

Лемматизация/Стемминг: Приведите слова к их базовой форме.

Создание словаря: Создайте словарь всех уникальных слов в корпусе, присвоив каждому слову уникальный целочисленный индекс.

Преобразование текста в последовательности: Преобразуйте каждое предложение в последовательность целочисленных индексов слов.

Паддинг (Padding): Убедитесь, что все последовательности имеют одинаковую длину, дополняя более короткие последовательности нулями (padding) или обрезая более длинные. Определите максимальную длину последовательности.

4. Построение сверточной нейронной сети для текста:

Используйте фреймворк Keras для создания модели.

Создайте модель типа Sequential.

Слой Embedding: Добавьте Embedding слой. Он будет отображать целочисленные индексы слов в плотные векторы (эмбединги). Укажите input_dim (размер словаря), output_dim (размерность эмбединга, например, 100-300) и input_length (максимальная длина последовательности).

Сверточные слои (Conv1D): Добавьте один или несколько слоев Conv1D. Они работают аналогично 2D сверткам, но для одномерных данных (текста). Используйте разные размеры ядер (filters) для захвата n-грамм разной длины (например, 2-граммы, 3-граммы, 4-граммы).

Пулинг (GlobalMaxPooling1D или MaxPooling1D): После сверточных слоев примените пулинг для получения наиболее важных признаков. GlobalMaxPooling1D часто эффективен для классификации текста.

Полносвязные слои: Добавьте один или два полносвязных (Dense) слоя с функцией активации relu.

Выходной слой: Добавьте выходной Dense слой с количеством нейронов, равным количеству классов, и функцией активации sigmoid (для бинарной классификации) или softmax (для многоклассовой классификации).

5. Настройка и обучение модели:

Оптимизатор: Adam.

Функция потерь: binary_crossentropy (для бинарной классификации) или categorical_crossentropy (для многоклассовой).

Метрики: accuracy.

Разделите данные на обучающую и тестовую выборки.

Обучите модель, настроив гиперпараметры (количество эпох, размер батча).

6. Оценка и визуализация:

Оцените модель на тестовой выборке, используя метрики (accuracy, precision, recall, F1-score).

Визуализируйте график изменения метрик по эпохам.

Продемонстрируйте предсказания модели на нескольких примерах предложений из тестовой выборки.

Практическая работа № 13. Генерация текста с помощью рекуррентной нейросети

Цель: Разработать и обучить простую рекуррентную нейронную сеть (RNN) для генерации нового текста, основанного на статистических закономерностях входного корпуса.

Задание:

1. Установка библиотек и импорт:

Установите необходимые библиотеки: tensorflow (или keras), numpy, nltk.

Импортируйте их: tensorflow.keras.models, tensorflow.keras.layers, tensorflow.keras.preprocessing.text, tensorflow.keras.preprocessing.sequence, numpy, nltk.

2. Загрузка и обзор текстовых данных:

Выберите простой текстовый корпус (например, несколько глав книги, коллекцию стихов, набор новостных статей). Небольшой корпус поможет быстрее обучить модель.

Загрузите и прочитайте текст.

3. Предобработка текста:

Токенизация: Разбейте весь текст на токены (слова или символы). Для начала лучше использовать символьную токенизацию (символ за символом), так как она проще в реализации и требует меньше данных.

Создание словаря: Создайте словарь уникальных символов (или слов) и присвойте каждому индекс.

Подготовка данных для модели: Создайте последовательности длиной `sequence_length` (например, 40-100 символов), где входные символы используются для предсказания следующего символа.

Например, если `sequence_length = 40`, то вход: `[char_1, char_2, ..., char_40]`, выход: `char_41`.

One-hot encoding: Преобразуйте входные последовательности и выходные символы в формат one-hot encoding.

4. Построение модели рекуррентной нейронной сети (RNN):

Используйте фреймворк Keras для создания модели.

Создайте модель типа Sequential.

Входной слой: Input слой с `shape=(sequence_length, num_unique_chars)`, где `num_unique_chars` – количество уникальных символов в словаре.

Простой RNN слой: Добавьте слой SimpleRNN с количеством нейронов (единиц) (например, 128-256). `return_sequences=True` может потребоваться, если вы планируете использовать несколько RNN слоев подряд.

Выходной слой: Dense слой с количеством нейронов, равным `num_unique_chars`, и функцией активации softmax для предсказания вероятности каждого символа быть следующим.

5. Настройка и обучение модели:

Оптимизатор: Adam.

Функция потерь: categorical_crossentropy.

Разделите подготовленные данные на обучающую и валидационную выборки.

Обучите модель, настроив гиперпараметры (количество эпох, размер батча).

Обучение может занять время.

6. Генерация текста:

Напишите функцию для генерации текста.

Начните с "зерна" (seed text) – небольшой начальной последовательности символов.

Подайте "зерно" в модель, чтобы получить вероятности следующего символа.

Выберите следующий символ (например, сэмплируя из распределения вероятностей или выбирая наиболее вероятный).

Добавьте сгенерированный символ к "зерну", удалите первый символ "зерна" и повторите процесс для генерации следующего символа.

Сгенерируйте текст заданной длины.

7. Анализ результатов:

Оцените качество сгенерированного текста: насколько он читаем, есть ли в нем смысловые паттерны, похож ли он на исходный текст.

Практическая работа № 14. Применение LSTM для генерации текста

Цель: Разработать и обучить модель на основе LSTM (Long Short-Term Memory) для генерации более качественного и связного текста по сравнению с простой RNN.

Задание.

1. Установка библиотек и импорт:

Аналогично работе №13: tensorflow (или keras), numpy, nltk.

2. Загрузка и обзор текстовых данных:

Используйте тот же (или больший) текстовый корпус, что и в работе №13.

3. Предобработка текста:

Токенизация: Используйте символьную или словарную токенизацию. Для LSTM словарная токенизация может дать лучшие результаты, если данных достаточно.

Создание словаря.

Подготовка данных: Создайте последовательности длиной `sequence_length`.

One-hot encoding входных последовательностей и выходных символов/слов.

4. Построение модели LSTM:

Используйте фреймворк Keras.

Создайте модель типа Sequential.

Слой LSTM: Добавьте один или несколько слоев LSTM. Укажите количество единиц (например, 128-256).

Если используете несколько LSTM слоев, у всех, кроме последнего, установите `return_sequences=True`.

Выходной слой: Dense слой с количеством нейронов, равным размеру словаря, и функцией активации softmax.

5. Настройка и обучение модели:

Оптимизатор: Adam.

Функция потерь: `categorical_crossentropy`.

Разделите данные на обучающую и валидационную выборки.

Обучите модель. Обучение LSTM может потребовать больше времени и данных, чем простая RNN. Настройте гиперпараметры (количество эпох, размер батча, скорость обучения).

6. Генерация текста:

Напишите функцию для генерации текста, аналогичную работе №13.

Начните с "зерна".

Используйте обученную LSTM модель для предсказания следующего символа/слова.

Сгенерируйте текст заданной длины.

7. Анализ результатов:

Сравните качество сгенерированного текста с текстом, сгенерированным простой RNN (из работы №13). Оцените связность, грамматику и семантику.

Практическая работа № 15. Сравнение качества текстов на выходе разных модификаций LSTM

Цель: Исследовать влияние различных конфигураций и модификаций LSTM (например, количество слоев, количество юнитов, использование Dropout, Bidirectional LSTM) на качество генерируемого текста.

Задание:

1. Подготовка корпуса и данных:

Используйте тот же корпус и подготовку данных, что и в работе №14. Убедитесь, что у вас есть хорошо подготовленный набор данных для обучения.

2. Создание и обучение нескольких моделей LSTM:

Разработайте 3-4 различные архитектуры LSTM:

Базовая LSTM: Одна LSTM слой с умеренным количеством юнитов (например, 128).

Глубокая LSTM: Два или более LSTM слоя.

LSTM с Dropout: Добавьте Dropout слои после LSTM или Dense слоев для борьбы с переобучением.

Bidirectional LSTM: Используйте Bidirectional(LSTM(...)) для учета контекста как в прямом, так и в обратном направлении.

(Опционально) GRU: Сравните с GRU (Gated Recurrent Unit), которая является упрощенной версией LSTM.

Для каждой модели:

Задайте архитектуру.

Настройте компиляцию (оптимизатор, функция потерь, метрики).

Обучите модель на одном и том же наборе данных, используя одинаковое количество эпох и размер батча (насколько это возможно). Используйте раннюю остановку (EarlyStopping) для более точного определения оптимального количества эпох.

Сохраните обученные модели.

3. Генерация текстов:

Для каждой обученной модели:

Используйте одинаковое "зерно" (seed text).

Сгенерируйте текст фиксированной длины.

4. Оценка и сравнение качества текстов:

Субъективная оценка:

Прочитайте и сравните сгенерированные тексты от каждой модели.

Оцените их читаемость, связность, грамматическую правильность, наличие смысловых паттернов, соответствие стилю исходного текста.

Объективная оценка (опционально, более сложная):

Рассчитайте метрики, такие как Perplexity (для оценки того, насколько хорошо модель предсказывает последовательность).

(Для более продвинутой оценки) Используйте BLEU, ROUGE (хотя они чаще применяются для задач перевода и суммаризации, их можно адаптировать).

Визуализация:

Постройте графики изменения метрик (accuracy, loss) по эпохам для всех моделей, чтобы показать, как разные архитектуры обучались.

5. Выводы:

Сформулируйте выводы о том, какая модификация LSTM (или другая архитектура) показала наилучшие результаты для данной задачи генерации текста и почему.

Изложите свой взгляд на компромиссы между сложностью модели, временем обучения и качеством генерации.

Практическая работа № 16. Анализ произвольного датасета изображений

Цель: Ознакомиться с процессом исследования и предобработки произвольного набора данных изображений.

Задание.

1. Установка библиотек и импорт:

Установите необходимые библиотеки: numpy, pandas, matplotlib, PIL (Pillow) или cv2 (OpenCV), sklearn (для статистики и разделения данных), os, glob.

Импортируйте их.

2. Выбор и загрузка датасета:

Выбор датасета: Найдите или создайте произвольный набор данных изображений. Это может быть:

Сборник фотографий разных видов цветов.

Фотографии различных животных.

Изображения различных объектов (например, фрукты, мебель).

Датасет с изображениями, где присутствуют или отсутствуют определенные объекты.

Структура датасета: Убедитесь, что изображения организованы таким образом, чтобы их можно было легко загрузить (например, в папках по классам, или в виде списка файлов с соответствующими метками).

Загрузка изображений: Напишите скрипт для загрузки изображений из указанных директорий.

3. Исследовательский анализ данных (EDA):

Просмотр изображений: Визуализируйте случайную выборку изображений из датасета, чтобы получить общее представление о содержании и качестве данных.

Размер изображений: Определите размеры (ширина, высота) всех изображений в датасете. Постройте гистограмму распределения размеров, чтобы выявить вариативность.

Количество изображений по классам (если применимо): Если датасет классифицирован, проверьте баланс классов. Постройте столбчатую диаграмму.

Цветовые пространства: Определите, в каком цветовом пространстве находятся изображения (RGB, Grayscale).

Анализ яркости/контраста (опционально): Можно рассчитать среднюю яркость или контраст изображений.

4. Предобработка изображений:

Приведение к единому размеру: Выберите стандартный размер для всех изображений (например, 224x224, 128x128). Все изображения, имеющие другие размеры, должны быть изменены (resize).

Нормализация: Преобразуйте значения пикселей из диапазона [0, 255] в диапазон [0, 1] или [-1, 1].

Преобразование цветового пространства (если необходимо): Если модель ожидает определенное цветовое пространство (например, RGB), убедитесь, что все изображения приведены к нему.

Аугментация данных (опционально, для подготовки к следующей работе): Примените базовые техники аугментации: случайные повороты, сдвиги, отражения, масштабирование. Это может быть сделано здесь или непосредственно перед обучением.

5. Разделение данных:

Разделите предобработанные данные на обучающую (train), валидационную (validation) и тестовую (test) выборки.

6. Сохранение предобработанных данных (опционально):

Если предобработка занимает много времени, можно сохранить обработанные изображения и метки, чтобы не выполнять эту операцию каждый раз.

Практическая работа № 17. Применение бинарного классификатора к произвольному датасету изображений

Цель: Разработать и обучить бинарный классификатор на основе сверточной нейронной сети (CNN) для решения задачи классификации изображений на два класса (например, "наличие объекта X" vs "отсутствие объекта X", "тип 1" vs "тип 2"). Целью является применение знаний о CNN к решению реальной задачи классификации изображений.

Задание:

1. Установка библиотек и импорт:

Установите необходимые библиотеки: tensorflow (или keras), numpy, matplotlib, PIL (Pillow) или cv2 (OpenCV), sklearn (для метрик и разделения данных), os, glob.

Импортируйте их.

2. Загрузка и подготовка данных:

Выбор датасета: Используйте датасет, подготовленный в работе №16, или найдите новый бинарный датасет изображений. Если датасет не является бинарным, модифицируйте его, выбрав два класса для классификации.

Предобработка: Выполните все шаги предобработки из работы №16:

Приведение к единому размеру.

Нормализация пикселей.

Формирование правильной размерности для CNN (например, (N, H, W, C)).

Кодирование меток: Кодировать метки классов цифрами: 0 для одного класса, 1 для другого.

Разделение на выборки: Разделите данные на обучающую (train), валидационную (validation) и тестовую (test) выборки.

3. Построение бинарной классификационной CNN:

Используйте фреймворк Keras.

Создайте модель типа Sequential.

Сверточные блоки: Добавьте несколько сверточных слоев (Conv2D) с функцией активации relu и слоев пулинга (MaxPooling2D).

Слой сглаживания (Flatten): Преобразуйте выход сверточных слоев в одномерный вектор.

Полносвязные слои: Добавьте один или два полносвязных (Dense) слоя с функцией активации relu.

Выходной слой: Dense слой с одним нейроном и функцией активации sigmoid. Sigmoid выдаст вероятность принадлежности изображения к классу "1".

4. Настройка и обучение модели:

Оптимизатор: Adam.

Функция потерь: binary_crossentropy (так как задача бинарной классификации).

Метрики: accuracy, Precision, Recall, AUC (Area Under the ROC Curve).

Обучите модель на 20-50 эпохах, используя валидационную выборку для мониторинга. Настройте размер батча.

Рассмотрите использование аугментации данных во время обучения (например, с помощью ImageDataGenerator в Keras) для увеличения размера обучающей выборки и улучшения обобщающей способности модели.

5. Оценка и визуализация:

Оценка на тестовой выборке: Сделайте предсказания на тестовой выборке.

Построение матрицы ошибок (Confusion Matrix): Визуализируйте матрицу ошибок, чтобы понять, какие классы модель путает.

ROC-кривая и AUC: Постройте ROC-кривую и рассчитайте значение AUC.

График метрик: Постройте графики изменения точности, потерь (loss) на обучающей и валидационной выборках по эпохам.

Визуализация предсказаний: Покажите несколько изображений из тестовой выборки с истинными метками и предсказанными вероятностями/классами.

6. Анализ результатов:

Проанализируйте полученные метрики. Насколько хорошо модель справляется с задачей?, Как аугментация данных повлияла на результаты? Предложите возможные улучшения.

Практическая работа № 18. Интеллектуальная обработка видео

Цель: Разработать и применить систему компьютерного зрения для автоматического обнаружения и классификации объектов типа "автомобиль", "автобус" и "пешеход" в видеопотоке. Целью является создание работоспособного детектора объектов, способного работать в динамических условиях реального мира, и оценка его эффективности.

Задание.

1. Выбор и подготовка данных:

- Выбор датасета: Найдите или сформируйте набор видеоданных, содержащих сцены с автомобилями, автобусами и пешеходами. Рекомендуется использовать общедоступные датасеты, специально предназначенные для детектирования объектов в городской среде (например, COCO, KITTI, Cityscapes).
- Аннотирование (при необходимости): Если выбранный датасет не имеет нужных аннотаций (ограничивающие рамки вокруг объектов с указанием класса), выполните ручное аннотирование части данных. Это включает отрисовку прямоугольников вокруг каждого целевого объекта (автомобиль, автобус, пешеход) и присвоение соответствующего класса.
- Разделение данных: Разделите аннотированные данные на обучающую, валидационную и тестовую выборки.

2. Выбор и настройка модели детекции объектов:

- Выбор архитектуры: Выберите современную архитектуру нейронной сети для детектирования объектов. Рекомендуемые варианты:
 - Одноэтапные детекторы (One-Stage Detectors): YOLO (v3, v4, v5, v7), SSD (Single Shot MultiBox Detector), RetinaNet. Они быстрее, что важно для видео.
 - Двухэтапные детекторы (Two-Stage Detectors): Faster R-CNN, Mask R-CNN. Часто более точные, но медленнее.
- Использование предобученных весов: Начните с использования модели, предобученной на большом общем датасете (например, ImageNet, COCO). Это значительно ускорит процесс обучения и повысит его эффективность (transfer learning).
- Адаптация к задаче: Переобучите (fine-tune) выбранную модель на своем подготовленном датасете, используя только классы "автомобиль", "автобус", "пешеход".

3. Обучение модели:

- Настройка гиперпараметров: Определите оптимальные гиперпараметры обучения, такие как скорость обучения (learning rate), размер батча (batch size), количество эпох (epochs), оптимизатор (Adam, SGD).
- Функция потерь: Используйте стандартные функции потерь для детектирования объектов, которые обычно включают потерю классификации (confidence loss) и потерю локализации (bounding box regression loss).
- Мониторинг: Отслеживайте метрики качества (см. ниже) на валидационной выборке во время обучения, чтобы предотвратить переобучение (overfitting) и выбрать наилучшую модель.

4. Применение и оценка модели на видео:

- Обработка видеопотока: Напишите скрипт, который читает видеофайл кадр за кадром.
- Детекция на каждом кадре: Для каждого прочитанного кадра подайте его на вход обученной модели детекции. Модель должна вернуть список обнаруженных объектов с их ограничивающими рамками, классами и уверенностью (confidence score).
- Фильтрация результатов: Отфильтруйте обнаружения с низкой уверенностью (например, ниже порога 0.5).

- Визуализация: Нарисуйте ограничивающие рамки и названия классов поверх исходных кадров видео.
- Оценка метрик:
 - mAP (mean Average Precision): Основная метрика для оценки точности детекции объектов. Рассчитывается отдельно для каждого класса, а затем усредняется.
 - Precision & Recall: Оценка доли правильных обнаружений и доли обнаруженных объектов от общего числа.
 - IoU (Intersection over Union): Используется для определения, считается ли обнаружение верным (сравнивается пересечение предсказанной и истинной рамки).
 - FPS (Frames Per Second): Оценка скорости работы модели, критически важная для обработки видео в реальном времени.

5. Анализ и улучшение:

- Проанализируйте полученные результаты: где модель ошибается (ложные срабатывания, пропуски), насколько высока скорость работы.
- Предложите возможные пути улучшения: использование более продвинутых архитектур, сбор большего количества разнообразных данных, аугментация данных, изменение параметров предобработки или пост-обработки.

Практическая работа № 19. Предсказание цены актива на основании графика изменения цены с использованием нейросети на 1 шаг

Цель: Разработать и обучить простую нейронную сеть для прогнозирования следующего значения цены актива, опираясь исключительно на исторические данные его ценовых колебаний

Задание.

1. Сбор и подготовка данных:

- Выберите один финансовый актив (например, акцию известной компании, криптовалюту, валютную пару) и найдите исторические данные о его цене (Open, High, Low, Close, Volume) за определенный период.
- Осуществите предобработку данных: очистка от пропусков, приведение к нужному формату, возможно, нормализация или стандартизация цен.
- Сформируйте пары "входные данные - выходные данные": входными данными будет последовательность предыдущих цен (например, 10-20 предыдущих значений), а выходными – следующее значение цены.

2. Построение модели:

- Используйте библиотеку для машинного обучения (например, TensorFlow/Keras или PyTorch) для построения простой рекуррентной нейронной сети (RNN), такой как LSTM (Long Short-Term Memory) или GRU (Gated Recurrent Unit), или даже простой полносвязный слой (MLP), если последовательность будет преобразована в фиксированный вектор.
- Определите архитектуру сети: количество слоев, количество нейронов в каждом слое, функцию активации.

3. Обучение модели:

- Разделите данные на обучающую и тестовую выборки.
- Обучите нейронную сеть на обучающей выборке, используя подходящую функцию потерь (например, Mean Squared Error - MSE) и оптимизатор (например, Adam).
- Настройте гиперпараметры модели (скорость обучения, количество эпох, размер батча) для достижения наилучших результатов на валидационной выборке.

4. Оценка результатов:

- Оцените качество предсказаний на тестовой выборке, используя метрики, такие как MSE, RMSE (Root Mean Squared Error), MAE (Mean Absolute Error) или R^2 .
- Визуализируйте реальные цены и предсказанные цены на графике для наглядной оценки работы модели.

Практическая работа № 20. Предсказание цены актива на основании графика изменения цены с использованием нейросети на несколько шагов

Цель: Разработать и обучить нейронную сеть, способную прогнозировать последовательность из нескольких будущих значений цены актива, основываясь на его исторической динамике.

Задание.

1. Сбор и подготовка данных:

- Используйте те же данные, что и в работе №19.
- Сформируйте пары "входные данные - выходные данные": входными данными будет последовательность предыдущих цен (например, 10-20 предыдущих значений), а выходными – последовательность из нескольких следующих значений цены (например, 5-10 шагов вперед).
- При необходимости адаптируйте предобработку данных под многошаговое прогнозирование.

2. Построение модели:

- Используйте рекуррентные нейронные сети (LSTM, GRU) или другие архитектуры, подходящие для генерации последовательностей (например, Encoder-Decoder).
- Архитектура должна быть способна выдавать предсказания на несколько временных шагов. Возможные подходы:
 - Recursive forecasting: Обучить модель предсказывать один шаг, а затем использовать предсказанное значение как вход для следующего предсказания.
 - Direct forecasting: Обучить модель предсказывать все N шагов вперед напрямую.
 - Seq2Seq модель: Использовать кодировщик для обработки входной последовательности и декодировщик для генерации выходной последовательности.

3. Обучение модели:

- Аналогично работе №19, разделите данные, обучите модель, подберите гиперпараметры.
- Функция потерь должна корректно работать с многомерными выходными данными (например, усредненная MSE по всем шагам).

4. Оценка результатов:

- Оцените качество предсказаний для каждого шага прогнозирования, используя метрики (MSE, RMSE, MAE).
- Сравните, как меняется точность прогноза с увеличением горизонта предсказания.
- Визуализируйте реальные цены и предсказанные последовательности цен на графике.

Практическая работа № 21. Предсказание цены актива на основании графика изменения цены, объема торгов и цены связанного актива(индекса) с использованием нейросети на 1 шаг

Цель: Исследовать влияние дополнительных факторов, таких как объем торгов самого актива и цена связанного актива (например, биржевого индекса), на точность прогнозирования цены.

Задание:

1. Сбор и подготовка данных:

- Выберите целевой актив.
- Найдите исторические данные о цене, объеме торгов целевого актива.
- Найдите исторические данные о цене связанного актива (например, S&P 500, NASDAQ, если целевой актив – акция американской компании, или соответствующий индекс для другого рынка).
- Произведите синхронизацию данных по времени.
- Выполните предобработку: очистка, нормализация/стандартизация всех признаков.

- Сформируйте пары "входные данные - выходные данные": входными данными будет последовательность прошлых цен, объемов и цен связанного актива; выходными – следующее значение цены целевого актива.

2. Построение модели:

- Используйте нейронные сети (LSTM, GRU), которые способны работать с многомерными входными данными.

- Архитектура модели должна предусматривать возможность обработки нескольких входных признаков на каждом временном шаге.

3. Обучение модели:

- Разделите данные.

- Обучите модель, используя мультивариантные входные данные.

- Сравните результаты обучения и предсказаний с моделью из работы №1 (обученной только на ценах), чтобы количественно оценить эффект от добавления новых признаков.

4. Оценка результатов:

- Оцените качество предсказаний на тестовой выборке, используя метрики.

- Проведите сравнительный анализ производительности модели с и без учета объема торгов и связанного актива.

- Визуализируйте реальные и предсказанные цены.

Практическая работа № 22. Предсказание цены актива на основании графика изменения цены, объема торгов и цены связанного актива(индеса) с использованием нейросети на несколько шагов

Цель: Объединить подходы из работ №20 и №21, разработав модель, которая прогнозирует последовательность будущих цен актива на несколько шагов вперед, учитывая при этом множество входных факторов (цена, объем, связанный актив).

Задание:

1. Сбор и подготовка данных:

- Используйте те же мультивариантные данные, что и в работе №21.

- Сформируйте пары "входные данные - выходные данные": входными данными будет последовательность прошлых цен, объемов и цен связанного актива; выходными – последовательность из нескольких следующих значений цены целевого актива.

2. Построение модели:

- Используйте архитектуры, подходящие для многошагового прогнозирования с многомерными входными данными (например, Seq2Seq с LSTM/GRU, или специализированные модели для мультивариантных временных рядов).

- Модель должна уметь обрабатывать несколько признаков на входе и генерировать последовательность предсказаний на выходе.

3. Обучение модели:

- Разделите данные.

- Обучите модель, оптимизируя ее для многошагового прогнозирования с учетом всех входных признаков.

- Сравните результаты с моделью из работы №2 (многошаговое прогнозирование только по цене), чтобы оценить добавочную ценность мультивариантного подхода.

4. Оценка результатов:

- Оцените качество предсказаний для каждого шага прогнозирования.

- Проанализируйте, как добавление новых признаков повлияло на точность прогноза на разных временных горизонтах.

- Визуализируйте реальные цены и предсказанные последовательности цен.

Требования к оформлению

Отчет должен содержать подробное описание (включая иллюстрации). Отчёт по практическому занятию выполняется на страницах формата А4 в электронном виде.

При оформлении отчёта используется сквозная нумерация страниц, считая титульный лист первой страницей. Номер страницы на титульном листе не ставится. Номера страницы ставятся по центру сверху.

При оформлении отчёта соблюдать следующие требования:

- Для заголовков: полужирный шрифт, 14 пт, центрированный.
- Для основного текста: нежирный шрифт, 14 пт, выравнивание по ширине.
- Во всех случаях тип шрифта – Times New Roman, отступ абзаца 1.25 см, полуторный междустрочный интервал.
- Поля: левое – 2 см, правое, верхнее и нижнее – 1 см.

Критерии оценки:

Процедура оценивания

Оценка за практические работы выставляется на основе письменного отчета обучающегося. Преподаватель может попросить обучающегося исправить замечания по оформлению или содержанию отчета по практической работе.

| Формы текущего контроля | Критерии и нормы оценки |
|--------------------------------------|--|
| Отчеты по практическим работам 1-16 | 1-3 балла ставится обучающемуся, который продемонстрировал результаты выполнения практической работы, соответствующие поставленным задачам, и предоставил отчет, оформленный должным образом и содержащий краткое описание полученных результатов 0 баллов – работа не была выполнена |
| Отчеты по практическим работам 17-22 | 1-2 балла ставится обучающемуся, который продемонстрировал результаты выполнения практической работы, соответствующие поставленным задачам, и предоставил отчет, оформленный должным образом и содержащий краткое описание полученных результатов 0 баллов – работа не была выполнена |

7.3. Оценочные средства для промежуточной аттестации по итогам освоения дисциплины

7.3.1. Вопросы к промежуточной аттестации

Семестр 8

| № п/п | Вопросы к зачету |
|-------|---|
| 1 | Основные понятия искусственного интеллекта (ИИ). |
| 2 | Взаимосвязь ИИ с различными областями науки и техники |
| 3 | Опишите архитектуру FCN, принципы прямого и обратного распространения ошибки. Приведите пример задачи. |
| 4 | Назовите ключевые параметры (слои, нейроны, активации) и принципы проектирования нейронных сетей для классификации. Как они влияют на модель? |
| 5 | Перечислите методы оптимизации. Объясните L1/L2 регуляризацию, Dropout и Batch Normalization. |
| 6 | Опишите назначение функций активации. Приведите примеры (ReLU, Sigmoid, |

| № п/п | Вопросы к зачету |
|----------|---|
| | Softmax) и их применение в различных слоях. |
| 7 | Дайте определения переобучения и недообучения. Назовите причины и методы предотвращения переобучения. |
| 8 | Самоорганизующиеся сети (SOM): Объясните принцип работы SOM. В чем их отличие от полносвязных сетей? Приведите примеры применения. |
| 9 | Классификация визуальных образов: Какие проблемы возникают при использовании FCN для классификации изображений? Предложите пути их решения. |
| 10 | Перечислите и объясните ключевые метрики классификации (Accuracy, Precision, Recall, F1-score, ROC-AUC). В каких случаях каждая из них наиболее информативна? |
| 11 | Объясните назначение обучающей, валидационной и тестовой выборок. Опишите принцип k-блочной кросс-валидации. |
| 12 | Дайте определение гиперпараметров. Приведите примеры и объясните их влияние на обучение. Назовите методы их настройки. |
| 13 | Дайте определение Обработки Естественного Языка (NLP). Назовите 3-4 основные задачи, решаемые с помощью NLP, и кратко опишите одну из них. |
| 14 | Объясните концепцию модели "мешка слов" для представления текста. Каковы её основные ограничения? |
| 15 | Опишите два основных подхода в Word2Vec: CBOW (Continuous Bag-of-Words) и Skip-gram. В чем их принципиальные различия и какую задачу они решают? |
| 16 | Какие семантические и синтаксические свойства слов могут улавливаться векторными представлениями (Word Embeddings)? Приведите примеры. |
| 17 | Что такое предварительно обученные (pre-trained) Word Embeddings (например, GloVe, FastText)? В чем их преимущество перед обучением эмбеддингов с нуля? |
| 18 | Опишите архитектуру и принцип работы базовой рекуррентной нейронной сети (RNN). Объясните, как обрабатывается последовательность данных. |
| 19 | Какова проблема затухающих (vanishing) и взрывающихся (exploding) градиентов в базовых RNN? Почему она возникает? |
| 20 | Объясните архитектуру сети LSTM. Опишите назначение и принцип работы "вентилей" (gates) в LSTM-ячейке (забывающий, входной, выходной). |
| 21 | Что такое двунаправленная LSTM (BiLSTM)? Для каких задач NLP она особенно эффективна и почему? |
| 22 | Опишите архитектуру управляемой рекуррентной ячейки GRU. В чем её отличие от LSTM и каковы её преимущества/недостатки? |
| 23 | Как можно использовать RNN или LSTM для задачи классификации предложений? Какие особенности архитектуры и выходов модели важны для этой задачи? |
| 24 | Кратко опишите концепцию RNN-CF. Какие "контекстные признаки" могут использоваться и как они интегрируются в модель? |
| 25 | Опишите общую архитектуру системы машинного перевода на основе Seq2Seq моделей (Encoder-Decoder). Какова роль энкодера и декодера? |
| 26 | Что такое механизм внимания (Attention) и почему он стал революционным в задачах NLP, особенно в машинном переводе? Как он помогает модели фокусироваться на релевантных частях входной последовательности? |
| 27 | Помимо RNN, какие другие подходы (например, на основе Bag-of-Words с ML-классификаторами, или на основе Transformer) могут использоваться для классификации предложений? |
| 28 | Кратко опишите архитектуру Трансформера. Какие ключевые компоненты (Self-Attention, Positional Encoding) делают его мощным для обработки последовательностей? |
| 29 | Что такое предварительно обученные языковые модели (PLMs) типа BERT, GPT? В |

| № п/п | Вопросы к зачету |
|----------|---|
| | чем их основная идея и почему они так успешны? |
| 30 | Опишите процесс "дообучения" (fine-tuning) предварительно обученных языковых моделей для решения конкретных задач NLP |
| 31 | Какие актуальные направления исследований и развития в NLP вы можете назвать (например, понимание диалогов, генерация текста, вопросно-ответные системы, мультимодальные NLP)? |
| 32 | Что такое задача распознавания образов? Опишите общую схему решения задачи распознавания образов, включая этапы предобработки, выделения признаков и классификации. |
| 33 | Объясните, что такое признаки в контексте распознавания образов. Приведите примеры простых признаков (например, края, углы, цветовые гистограммы) и более сложных (например, SIFT, HOG). |
| 34 | Как метод опорных векторов (SVM) может быть использован для классификации образов? Объясните принцип работы SVM, включая понятие гиперплоскости разделения и векторов поддержки. |
| 35 | Опишите архитектуру сверточной нейронной сети (CNN). Объясните роль сверточных слоев, слоев пулинга (pooling) и полносвязных слоев в CNN для обработки изображений. |
| 36 | Детально объясните операции свертки (convolution) и пулинга (pooling) в CNN. Каково их назначение и как они влияют на представление признаков изображения? |
| 37 | Приведите примеры современных архитектур CNN (например, AlexNet, VGG, ResNet, Inception) и кратко опишите их ключевые отличия или инновации, которые позволили достичь лучших результатов в распознавании образов. |
| 38 | Что такое Transfer Learning (перенос знаний) и как оно применяется в машинном зрении? Опишите процедуру fine-tuning (дообучения) предобученной модели для новой задачи классификации изображений. |
| 39 | Помимо классификации изображений, назовите и кратко опишите другие задачи машинного зрения, такие как детектирование объектов, сегментация изображений, отслеживание объектов |
| 40 | Какие основные этапы предобработки изображений используются перед подачей в модель распознавания? Приведите примеры (нормализация, аугментация данных, изменение размера). |
| 41 | Какие основные проблемы и ограничения существуют в задачах распознавания образов и машинного зрения (например, вариативность освещения, ракурса, масштаба, окклюзия)? Как современные методы пытаются их решить? |
| 42 | Что такое временной ряд? Опишите основные компоненты временного ряда (тренд, сезонность, цикличность, шум) и методы его декомпозиции (аддитивная и мультипликативная модели). |
| 43 | Объясните понятие стационарности временного ряда. Почему стационарность важна для классических моделей прогнозирования? Какие методы используются для приведения временного ряда к стационарному виду? |
| 44 | Опишите принцип работы и компоненты модели ARIMA(p,d,q). Как определяются порядки p, d, q? Приведите примеры задач, где ARIMA будет уместна. |
| 45 | Метрики прогнозирования: Какие метрики используются для оценки качества прогнозов временных рядов (MAE, MSE, RMSE, MAPE)? Объясните их различия и в каких ситуациях каждая из них предпочтительна |
| 46 | Какие новые признаки (features) можно извлечь из исходного временного ряда для использования в моделях машинного обучения (например, запаздывающие значения, скользящие статистики, календарные признаки)? Приведите примеры. |
| 47 | Какие общие модели машинного обучения (например, линейная регрессия, |

| № п/п | Вопросы к зачету |
|------------------|--|
| | случайный лес, градиентный бустинг) могут быть адаптированы для прогнозирования временных рядов? Какие особенности их применения в этом контексте? |
| 48 | Объясните, почему рекуррентные нейронные сети (RNN) и их разновидности (LSTM, GRU) особенно подходят для задач прогнозирования временных рядов. Каковы преимущества LSTM/GRU по сравнению с простыми RNN? |
| 49 | Помимо RNN-подобных сетей, какие еще нейросетевые архитектуры (например, 1D CNNs, Transformer-подобные модели) применяются для анализа и прогнозирования временных рядов? Кратко опишите их преимущества. |
| 50 | Что такое одношаговое (single-step) и многошаговое (multi-step) прогнозирование? Какие существуют основные стратегии для реализации многошагового прогнозирования (например, рекурсивная, прямая, многовыходовая)? |

7.3.2. Критерии и нормы оценки

| Семестр | Форма проведения промежуточной аттестации | Критерии и нормы оценки | |
|----------------|--|--------------------------------|--------------------|
| 9 | Зачет | зачтено | От 0 до 100 баллов |
| | | не зачтено | Менее 55 баллов |

8. Учебно-методическое и информационное обеспечение дисциплины

8.1. Обязательная литература

| № п/п | Авторы, составители | Заглавие (заголовок) | Тип (учебник, учебное пособие, учебно-методическое пособие, практикум, др.) | Год издания | Количество в научной библиотеке / Наименование ЭБС |
|----------|---------------------|---|---|-------------|---|
| 1. | Сысоев Д. В. | Введение в теорию искусственного интеллекта : учебное пособие / Д. В. Сысоев, О. В. Курипта, Д. К. Проскурин. — Москва : Ай Пи Ар Медиа, 2021. — 170 с. | Учебное пособие | 2021 | ЭБС «IPRbooks» |
| 2 | Сазонов С. Н. | Системы искусственного интеллекта : учебное пособие / С. Н. Сазонов. — Ульяновск : Ульяновский государственный технический университет, 2023. — 84 с. | Учебное пособие | 2023 | ЭБС «IPRbooks» |
| 3 | Орлов А.И. | Искусственный интеллект: экспертные оценки : учебник / Орлов А.И.. — Москва : Ай Пи Ар Медиа, 2022. — 436 с. | Учебник | 2022 | ЭБС «IPRbooks» |
| 4 | Ручкин В.Н. | Системы искусственного интеллекта. Нейросети и нейрокомпьютеры : учебник / Ручкин В.Н., Костров Б.В., Свирина А.Г.. — Москва : КУРС, 2024. — 288 с. | Учебник | 2024 | ЭБС «IPRbooks» |

8.2. Дополнительная литература

| № п/п | Авторы, составители | Заглавие (заголовок) | Тип (учебник, учебное пособие, учебно- методическое пособие, практикум, др.) | Год издания | Количество в научной библиотеке / Наименование ЭБС |
|----------|---------------------|---|---|-------------|---|
| 1. | Павлов С. Н. | Системы искусственного интеллекта. Часть 1 | Учебное пособие | 2011 | ЭБС «IPRbooks» http://www.iprbookshop.ru/13974.html |
| 2. | Павлов С. Н. | Системы искусственного интеллекта. Часть 2 | Учебное пособие | 2011 | ЭБС «IPRbooks» http://www.iprbookshop.ru/13975.html |

8.3. Перечень профессиональных баз данных и информационных справочных систем

| № пп | Наименование | Ссылка |
|---------|--|---|
| 1 | Springer Nature (Полнотекстовая коллекция журналов) | https://www.springernature.com/gp/products |
| 2 | Springer eBooks (Полнотекстовая коллекция электронных книг издательства Springer Nature) | https://link.springer.com/ |
| 3 | «Кодекс» | https://kodeks.ru/ |
| 4 | ELIBRARY.RU (электронная библиотека научных публикаций) | http://elibrary.ru |
| 5 | "Гарант" | https://www.garant.ru/ |
| 6 | "КонсультантПлюс" | https://www.consultant.ru/ |
| 7 | Техэксперт | https://cntd.ru/ |

8.4. Перечень программного обеспечения

| № п/п | Наименование ПО | Реквизиты договора (дата, номер, срок действия) |
|----------|---|---|
| 1 | WinPro 10 RUS Upgrd OLP NL Acdmc | Договор № 757 от 04.07.2018, срок действия – бессрочно; Контракт № 1653 от 14.12.2018, срок действия – бессрочно |
| 2 | Office Stdandard 2013 Russian OLP NL AcademicEdition) | Контракт № 690 от 19.05.2015, срок действия – бессрочно |
| 3 | Python 3.11 | Лицензия Python Software Foundation License (PSFL) |
| 4 | Jupyter Notebook | Лицензия: Modified BSD License (Open Source). |
| 5 | TensorFlow | Лицензия: Apache License 2.0. |
| 6 | Visual Studio Code 1.75 | Free Software |
| 6 | Django 1.11.29 | Free Software |

8.5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

| № п/п | Наименование оборудованных учебных кабинетов, лабораторий, мастерских и др. объектов для проведения практических и лабораторных занятий, помещений для самостоятельной работы обучающихся (номер аудитории) | Перечень основного оборудования |
|----------|--|---|
| 1 | Компьютерный класс. Учебная аудитория для проведения занятий лекционного типа. Учебная аудитория для проведения занятий семинарского типа. Учебная аудитория для проведения лабораторных работ. Учебная аудитория для курсового проектирования | Компьютер (монитор DEXP DF27N1, системный блок Intel Core i7-12700, 2100МГц, 16 Гб RAM, UHD Graphics 770), столы ученические, стол преподавательский, стулья, |

| № п/п | Наименование оборудованных учебных кабинетов, лабораторий, мастерских и др. объектов для проведения практических и лабораторных занятий, помещений для самостоятельной работы обучающихся (номер аудитории) | Перечень основного оборудования |
|----------|---|--|
| | (выполнения курсовых работ). Учебная аудитория для проведения групповых и индивидуальных консультаций. Учебная аудитория для проведения занятий текущего контроля и промежуточной аттестации. (ИТП-211) | акустическая панель. |
| 2 | Помещение для самостоятельной работы обучающихся (УЛК-105). | Столы, стулья, стеллажи (в т.ч. выставочные) с книгами, компьютеры, мобильные рабочие места. |
| 3 | Помещение для самостоятельной работы обучающихся (УЛК-406). | Столы компьютерные, стулья, микрокомпьютеры raspberry pi 32 bit. |