

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Болдырев Антон Сергеевич
Должность: Директор
Дата подписания: 24.02.2026 21:49:38
Уникальный программный ключ:
9c542731014dd7196f5752b7fa57c524495323a0



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ (ФИЛИАЛ)
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО
ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
В Г. ТАГАНРОГЕ РОСТОВСКОЙ ОБЛАСТИ
ПИ (филиал) ДГТУ в г. Таганроге**

ЦМК «ПРИКЛАДНАЯ ИНФОРМАТИКА»

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ
ПРАКТИЧЕСКИХ РАБОТ**

по дисциплине ОП.13 «Системы искусственного интеллекта»

Таганрог
2026

Составители: М.С.Полищук

Практикум по выполнению практических работ по дисциплине ОП.13 «Системы искусственного интеллекта». ПИ (филиала) ДГТУ в г.Таганроге, 2026 г.

В практикуме кратко изложены теоретические вопросы, необходимые для успешного выполнения практических работ, рабочее задание и контрольные вопросы для самопроверки.

Предназначено для обучающихся по специальности 09.02.08 «Интеллектуальные интегрированные системы».

Ответственный за выпуск:

Зав. кафедрой (председатель ЦМК) (руководитель структурного подразделения, ответственного за реализацию ОПОП): О.В. Андриян

Ф.И.О.

© Издательский центр ДГТУ, 2026 г.

Введение

В учебно-методических указаниях к практикуму по курсу «Системы искусственного интеллекта» изложены сведения, необходимые для успешного выполнения практических занятий по данному курсу. Описан процесс работы с инструментарием, применяемым в практических работах, представлен ряд типичных задач и подходы к их решению. Практические занятия посвящены знакомству обучающихся с основными технологиями проектирования и разработки систем искусственного интеллекта. Цель настоящего пособия – помочь обучающимся при выполнении практических работ, выполняемых для закрепления знаний по теоретическим основам и получения практических навыков работы на компьютерах

Обучающийся должен знать:

Современные направления использования искусственного интеллекта

Особенности и признаки интеллектуальности информационных систем

Модели формализации предметной области

Модели представления знаний

Типы нейронных сетей и алгоритмы их обучения

Назначение и архитектура экспертных систем

Инструментальные средства реализации систем искусственного интеллекта

Принципы использования чат-ботов с искусственным интеллектом

Виды и правила формулирования промптов

Правила использования систем искусственного интеллекта

Обучающийся должен уметь:

- распознавать задачу и/или проблему в профессиональном и/или социальном контексте;
- анализировать задачу и/или проблему и выделять её составные части;
- определять этапы решения задачи;
- выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы;
- составлять план действия;
- определять необходимые ресурсы;
- владеть актуальными методами работы в профессиональной и смежных сферах;
- реализовывать составленный план;
- оценивать результат и последствия своих действий (самостоятельно или с помощью наставника).
- распознавать задачу и/или проблему в профессиональном и/или социальном контексте;

Практическое занятие № 1.

Моделирование требований к системе искусственного интеллекта с помощью диаграммы вариантов использования

Теоретическая часть

Диаграммы вариантов использования применяются при описании бизнес процессов автоматизируемой предметной области, определении требований к будущей программной системе. Отражает объекты как системы, так и предметной области, и задачи, ими выполняемые.

Варианты использования являются необходимым средством на стадии формирования требований к ПО. Каждый вариант использования – это потенциальное требование к системе, и пока оно не выявлено, невозможно запланировать его реализацию.

Действующие лица могут играть различные роли по отношению к варианту использования. Они могут пользоваться его результатами или могут сами непосредственно в нем участвовать. Значимость различных ролей действующего лица зависит от того, каким образом используются его связи.

Конкретная цель диаграмм вариантов использования – это документирование вариантов использования (всё, входящее в сферу применения системы), действующих лиц (всё вне этой сферы) и связей между ними. Разрабатывая диаграммы вариантов использования, старайтесь придерживаться следующих правил:

– Не моделируйте связи между действующими лицами.

По определению действующие лица находятся вне сферы действия системы. Это означает, что связи между ними также не относятся к её компетенции.

– Не соединяйте сплошной стрелкой (коммуникационной связью) два варианта использования непосредственно. Диаграммы данного типа описывают только, какие варианты использования доступны системе, а не порядок их выполнения. Для отображения порядка выполнения вариантов использования применяют диаграммы деятельности.

– Вариант использования должен быть инициирован действующим лицом. Это означает, что должна быть сплошная стрелка, начинающаяся на действующем лице и заканчивающаяся на варианте использования

Цель работы: изучить методологию объектно-ориентированного моделирования и получить практические навыки в моделировании предметной области с помощью UML

Рабочее задание

1. Построить модель предметной области, согласно выбранного варианта с помощью диаграммы вариантов использования UML. Модель должна отражать бизнес-процессы предметной области.
2. Оформить отчет по лабораторной работе.
3. Представить отчет по лабораторной работе для защиты

Шаги выполнения задания:

1. Щелкните правой кнопкой мыши на пакете «Общая схема» представления Use Case View в браузере.
2. Выберите в появившемся меню пункт New > Use Case
3. Новый вариант использования под названием NewUseCase появится в браузере. Слева от него будет видна пиктограмма варианта использования UML.
4. Выделив новый вариант использования, введите его название.
5. Щелкните правой кнопкой мыши на варианте использования.
6. В открывшемся меню выберите пункт Open Specification.
7. В поле стереотипа выберите Business Use Case и нажмите на кнопку ОК.

Для создания новой диаграммы вариантов использования:

- Щелкните правой кнопкой мыши на пакете «Бизнес-модель» представления Use Case View в браузере.
- Из всплывающего меню выберите пункт New далее Use Case Diagram.
- Выделив новую диаграмму, введите ее имя («Модель бизнес-процессов»).
- Дважды щелкните на названии этой диаграммы в браузере, чтобы открыть ее.
- Чтобы поместить действующее лицо или вариант использования на диаграмму, перетащите его мышью из браузера на диаграмму вариантов использования.
- С помощью кнопки Unidirectional Association (Однонаправленная ассоциация) панели инструментов нарисуйте ассоциации между действующими лицами и вариантами использования.

Контрольные вопросы

- Что такое диаграмма вариантов использования и как она помогает в моделировании требований к системе искусственного интеллекта?
- Какие ключевые элементы включает в себя диаграмма вариантов использования для системы искусственного интеллекта?
- Какие преимущества предоставляет моделирование требований с использованием диаграмм вариантов использования для системы искусственного интеллекта?
- Какие инструменты можно использовать для создания и анализа диаграмм вариантов использования для системы искусственного интеллекта?
- Какие основные шаги необходимо выполнить при создании диаграммы вариантов использования для системы искусственного интеллекта?
- Как классифицируются варианты использования в контексте системы искусственного интеллекта?
- Какие типы актеров могут взаимодействовать с системой искусственного интеллекта на диаграмме вариантов использования?
- Какие методы можно применить для проверки корректности созданной диаграммы вариантов использования для системы искусственного интеллекта?
- Каким образом диаграмма вариантов использования способствует коммуникации между разработчиками и заказчиком в контексте системы искусственного интеллекта?
- Какие трудности могут возникнуть при моделировании требований к системе искусственного интеллекта с использованием диаграмм вариантов использования и как их преодолеть?

Общие положения

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующего занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

Форма отчета:

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

Время работы: 2 часа.

Практическое занятие № 2.

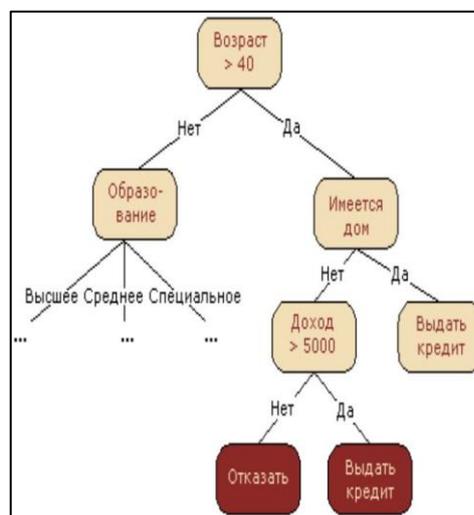
Построение дерева решений для системы искусственного интеллекта

Теоретическая часть

Что такое дерево решений

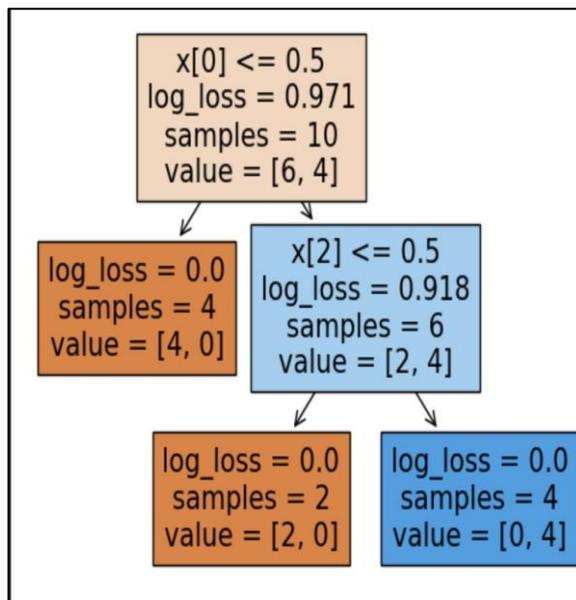
Дерево решений - это алгоритм машинного обучения, представляющий собой совокупность последовательных правил. Дерево решений можно отнести к логическим алгоритмам

Дерево решений - это алгоритм, который объединяет логические правил вида "Значение признака меньше И Значение признака меньше ... => Класс 1" в структуру данных "Дерево".



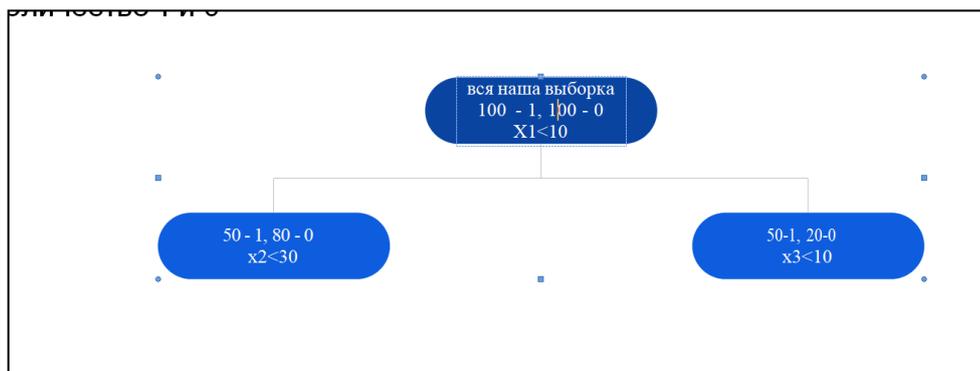
Как устроено дерево

- 1.Дерево состоит из корня, это стартовый узел
- 2.Далее в каждом узле задается вопрос: признак больше/меньше определенного значения
- 3.Узел, после которого нет разделения, называется терминальным узлом или листом

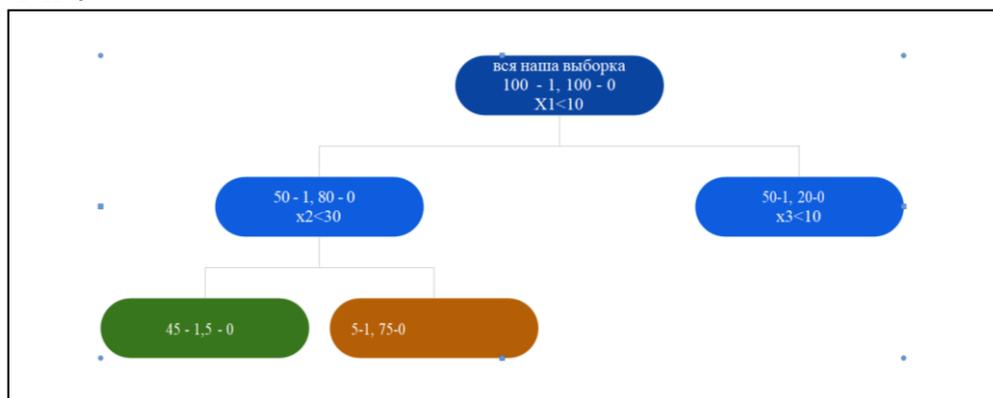


Как обучается дерево

На каждом следующем этапе мы разбиваем выборку так, чтобы концентрация того или иного класса была больше. 1 правило: $x_1 < 10 \rightarrow$ разбиваем выборку на 2 подвыборки и для каждой считаем количество 1 и 0

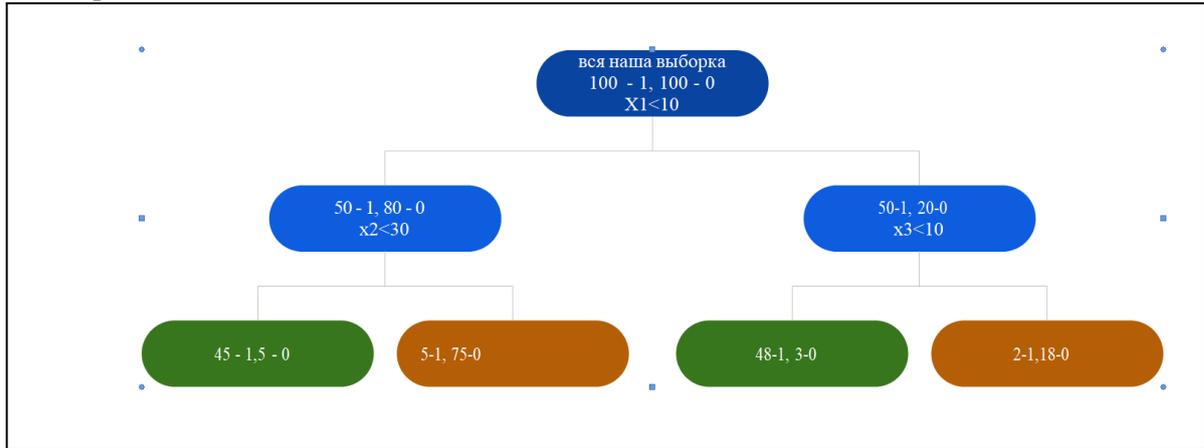


Далее идем влево и применяем уже другое правило: $x_2 < 30$ и снова считаем количество 1 и 0



Далее идем вправо и применяем уже другое правило: $x_3 < 10$ и снова считаем количество 1 и 0

0. Дальше разбиение не имеет уже смысла, так как в терминальных узлах (листах) выделились классы-лидеры



Алгоритмы построения дерева решений

ID3 (Iterative Dichotomiser 3) (Разработан Джоном Р. Квинланом):

- Работает только с дискретной целевой переменной.
- Построенные деревья являются квалифицирующими, то есть каждый лист соответствует конкретному значению целевой переменной.
- Число потомков в узле не ограничено, и дерево может иметь различную структуру.
- Алгоритм не поддерживает обработку пропущенных данных, что требует предварительной обработки данных перед использованием ID3.

C4.5:

- Является продвинутой версией алгоритма ID3.
- Работает с дискретными и непрерывными значениями атрибутов.
- Позволяет обрабатывать пропущенные значения атрибутов.
- Деревья решений, построенные с помощью C4.5, могут быть квалифицирующими, но также и квантифицирующими (допускающими промежуточные значения на листьях).

CART (Classification and Regression Tree) (Разработан Leo Breiman):

- Поддерживает как задачи классификации, так и регрессии.
- Может работать как с дискретными, так и с непрерывными целевыми переменными.
- Деревья, построенные на основе CART, имеют только два потомка для каждого узла, что делает их бинарными.

Как бороться с переобучением

Ранняя остановка:

- ограничение глубины дерева (`max_depth`)
- ограничение минимального количества наблюдений во внутреннем узле (`min_samples_split`)
- ограничение минимального количества наблюдений в листе (`min_samples_leaf`)
- ограничение `min_weight_fraction_leaf`, узел будет разделен на дочерние узлы только если `min_weight_fraction_leaf` (взвешенная доля суммарного веса примеров в узле) превысит `min_weight_fraction_leaf` % от общего суммарного веса всех примеров.
- ограничение на количество листиков - `max_leaf_nodes`
- минимальный прирост: `min_impurity_decrease`

Как бороться с переобучением

Стрижка дерева или Pruning

- Вводим штраф за сложность c_{sr_alpha}
- $R(T)$ - ошибки дерева, T - количество узлов
- наша задача минимизировать

Деревья решений. Плюсы

- Интерпретация
- Универсальность (классификация и регрессия)
- Быстрая обучаемость и предсказания
- Устойчивость к выбросам
- Не требуют масштабирования данных
- Эффективны для больших наборов данных
- Выявление важности признаков
- Легкая обработка пропущенных значений
- Возможность комбинирования в ансамблях

Деревья решений. Минусы

- Склонность к переобучению на сложных данных
- Неустойчивость к небольшим изменениям данных
- Сложность построения оптимальных деревьев с большим количеством признаков
- Не всегда гарантировано нахождение глобально оптимального решения
- Могут создавать слишком сложные модели, которые трудно интерпретировать
- Подвержены проблемам с несбалансированными данными в задачах классификации

innopolis university

Критерии качества. Обобщение

Для задачи регрессии:

Mean Squared Error:

$$\bar{y}_m = \frac{1}{n_m} \sum_{y \in Q_m} y$$

$$H(Q_m) = \frac{1}{n_m} \sum_{y \in Q_m} (y - \bar{y}_m)^2$$

Half Poisson deviance:

$$H(Q_m) = \frac{1}{n_m} \sum_{y \in Q_m} (y \log \frac{y}{\bar{y}_m} - y + \bar{y}_m)$$

Mean Absolute Error:

$$\text{median}(y)_m = \text{median}(y)_{y \in Q_m}$$

$$H(Q_m) = \frac{1}{n_m} \sum_{y \in Q_m} |y - \text{median}(y)_m|$$

https://scikit-learn.org/stable/modules/tree.html#mathematical-formulation

Пример решения задачи регрессии с помощью решающего дерева приведен здесь:
https://colab.research.google.com/drive/1GAK9tubxiqhwejp9fYYc-J9_T6AgYk?usp=sharing

Цель работы: построить регрессионную модель с использованием алгоритма дерева решений. Затем оценить качество модели, используя метрики, такие как MAE, MSE, R^2 , на тестовых данных..

Рабочее задание

1. Выбрать один из предложенных датасетов:
 - a. Набор данных <https://www.kaggle.com/competitions/playground-series-s3e11>
 - b. Набор данных <https://www.kaggle.com/competitions/playground-series-s3e6>
2. Провести разведочный анализ данных, ответив на следующие вопросы:
 - a. Сколько строк в датафрейме, сколько столбцов
 - b. Сколько места занимает датафрейм в оперативной памяти
 - c. Для каждой интервальной переменной подсчитать следующее - мин, медиана, среднее, макс и перцентили 25, 75
 - d. Для каждой категориальной переменной рассчитать моду и сколько раз мода встречается в данных
3. Подготовка датасета к построению моделей ML
 - a. Провести анализ и обработку пропусков (либо заменить, либо удалить)
 - b. Провести анализ и обработку выбросов (либо заменить, либо удалить)
 - c. Провести анализ и обработку категориальных переменных (сколько таких переменных, закодируйте категориальные переменные одним из методов (one hot encoding, mean target, frequency encoding)
 - d. Разделить датасет на трейн и тест
4. Обучить алгоритм:
Decision tree
5. Оценить качество алгоритма

Контрольные вопросы

1. Что представляет собой метод деревьев решений в контексте задачи регрессии?
2. Какие основные характеристики деревьев решений делают их применимыми для задачи регрессии?
3. Каким образом деревья решений используются для предсказания непрерывных значений в задаче регрессии?
4. Как происходит процесс построения дерева решений для задачи регрессии?
5. Какие критерии разделения используются при построении деревьев решений для задачи регрессии?
6. Как оценивается качество работы модели дерева решений в задаче регрессии?
7. Какие проблемы могут возникнуть при использовании деревьев решений для задачи регрессии и как их можно решить?
8. Какие методы прунинга деревьев решений могут быть применены для улучшения их обобщающей способности в задаче регрессии?
9. Какие преимущества и недостатки имеют деревья решений в сравнении с другими методами регрессии?
10. Как учитывать переобучение при использовании деревьев решений в задаче регрессии и какие стратегии можно применить для его предотвращения?

Общие положения

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующего занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

Форма отчета:

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

Время работы: 2 часа.

Практическое занятие № 3.

Построение экспертных систем по правилам if/then и с помощью дерева правил

Теоретическая часть:

Наиболее распространенной моделью представления знаний считается система продукций. Системы, использующие знания в виде продукций (правил), называются продукционными. Продукционная система почти не имеет процедурных компонентов, которые представляют основу фон-неймановской вычислительной системы, и практически полностью управляются данными, т.е. система является дескриптивной. Такая система включает три основных составляющих: базу правил, рабочую память (базу фактов) и механизм логического вывода [1].

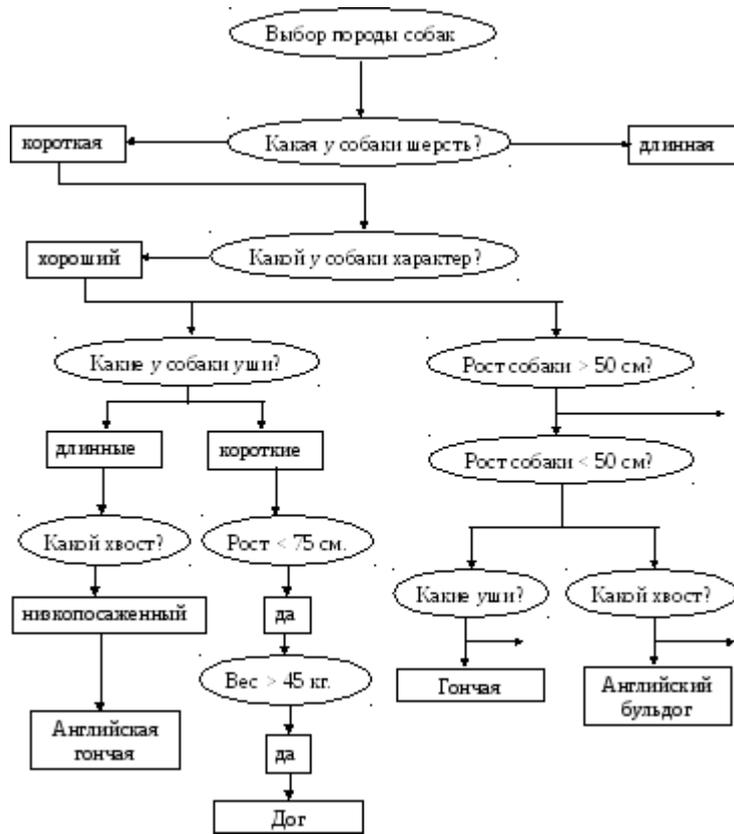
База правил содержит набор продукционных правил, имеющих форму IF-THEN (ЕСЛИ-ТО). Каждое правило складывается из двух частей. Первая из них IF - антецедент, или посылка правил, состоит из элементарных предложений, соединенных связками (операциями) И, ИЛИ и т. д. Вторая часть THEN, называемая консеквентом, или заключением, состоит из одного или нескольких предложений, которые образуют выдаваемое правилом решение.

Антецедент представляет собой образец правила, предназначенного для распознавания ситуации, когда правило должно сработать. Правило срабатывает, если факты из рабочей памяти при сопоставлении совпали с образцом, после чего правило считается отработавшим, т.е. делается вывод об истинности утверждения в заключении.

При составлении правил необходимо выполнять следующие рекомендации:

- использовать минимально достаточное множество условий при определении продукционного правила;
- избегать противоречащих продукционных правил;

- конструировать правила, опираясь на структуру присущую предметной области.



Первым этапом формирования правил является перевод дерева решений из вопросов и ответов в утверждение фактов. Пусть, например, фрагмент дерева решения для выбора породы собаки имеет вид, показанный на рис. 1:

Рис. 1. Дерево решений для выбора породы собак

Это дерево решений преобразуется в следующее дерево фактов:

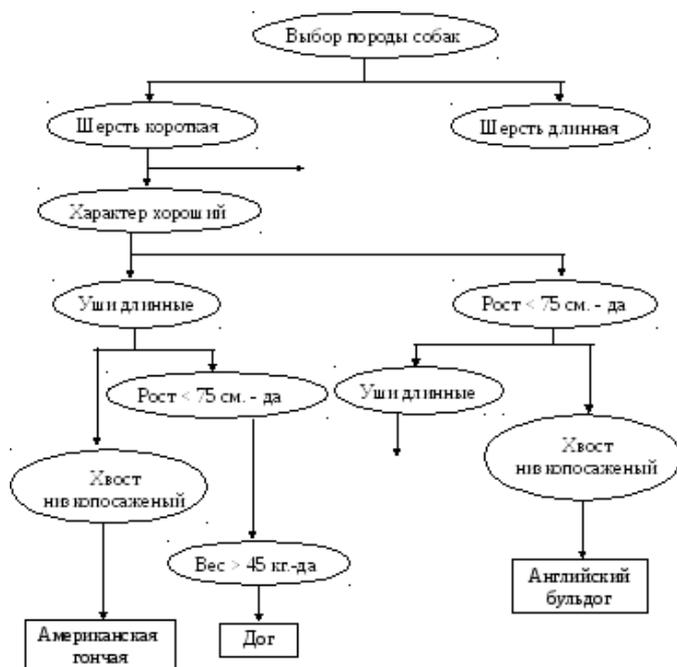


Рис.2. Дерево утверждение фактов

Каждая ветвь дерева от её начала (корня) до конца (листа дерева) образует правило. Предложения в дереве, расположенные до выделенного жирной линией прямоугольника образуют условную часть правила, а предложения внутри этого прямоугольника - заключение правила. Ветвь, в которой нет заключения, не может быть представлена в виде правила.

Утверждение фактов состоит из следующих частей:

Название	Пример
Атрибут	Шерсть
Значение	Короткая
Предикат	Является

Атрибут - это ключевое слово или фраза, выбранная для представления факта, который мы пытаемся определить. Значение - описание, назначенное атрибуту. Предикат - элемент, устанавливающий отношение между атрибутом и его значением. Это дополнительные средства продукционной системы, которые применяются для представления конкретных данных. В некоторых продукционных системах каждое высказывание состоит из триплета: "объект-атрибут-решение" [2]. В этом случае antecedentes и consequents состоят из троек (например, база знаний экспертной системы MYCIN).

Представление данных с помощью триплета можно легко расширить для описания данных, включающих показатели нечёткости. Например, в системе MYCIN применяется показатель нечёткости, который в данном конкретном случае называется фактором достоверности. Таким образом, данные управляются при помощи ассоциативной четверки "объект-атрибут-значение-фактор достоверности". Но чаще всего применяются пары "атрибут-значение", потому что в этом случае упрощается процесс формирования правил.

Рассмотренные рекомендации и дерево на рис.2 позволяют записать следующее правило на Прологе:

```
dog("Гончая):-
priznak("шерсть короткая"),
priznak("рост меньше 50 см."),
priznak("уши длинные"),
priznak("характер хороший"),!.
```

Как видно, каждое продукционное правило является независимым от других. Эта независимость делает базу правил семантически модульной, т.е. группы информации не влияют друг на друга.

Кроме того, модульность базы правил позволяет расширять базу знаний, увеличивая её новыми правилами.

После построения базы продукционных правил следует уделить внимание процессам, возникающим во время диалога с интеллектуальной системой. Эти процессы включают управление потоком данных, поступающих от пользователя экспертной системы, получение информации из базы фактов и выдачу результатов консультаций. Итогом такого анализа является диаграмма потоков данных между блоками экспертной системы. Для экспертной системы (ЭС) выбора породы собаки диаграмма может быть представлена следующим образом:

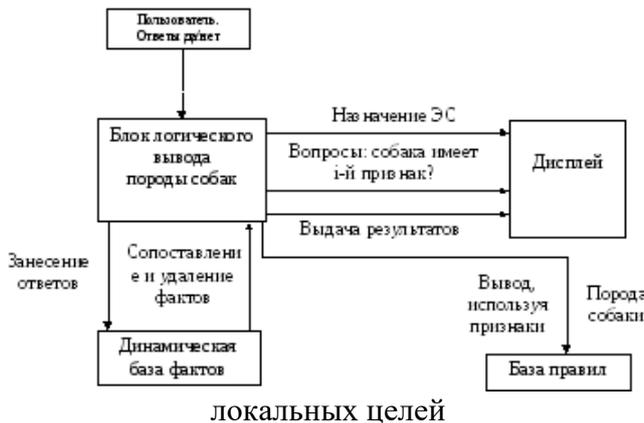
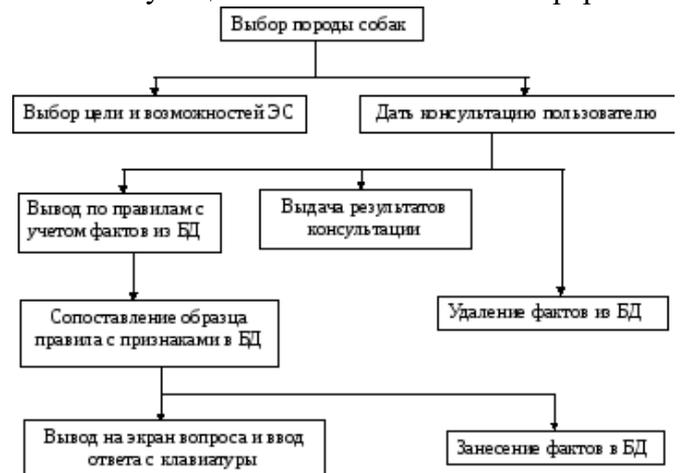


Рис. 3. Диаграмма потоков данных

Далее, основываясь на диаграмме потоков данных и выбранной стратегии решения задачи, составляется структурная схема, которая показывает состав программных модулей и правил. Например, для идентификации породы собак можно построить экспертную систему со следующей структурой (рис.4):

Рис. 4. Функциональная схема ЭС с иерархией



Теперь можно приступить к программированию прототипа ЭС на Прологе. В начале необходимо сделать декларации базы данных (фактов) в разделе **database**. Она будет хранить ответы пользователя на вопросы системы. Ответы могут быть утвердительными или отрицательными. Затем в разделе **predicates** следует объявить предикаты для простейшего взаимодействия с пользователем и выполнения вывода результата (реализации механизма логического вывода).

После того как экспертная система представлена на Прологе, необходимо проверить, как она работает. Для этого проводится тестирование и отладка. Большую по количеству правил базу знаний надо наполнять и отлаживать по частям. Как только введена часть базы знаний, охватывающая законченную проблему, её необходимо проверить и только после этого вводить следующую. Тестирование проводится по имеющемуся дереву решения. Если правила составлялись без него, то необходимо набросать план просмотра каждой ветви в ходе консультаций.

Пример эс на Прологе

```

/* Программа: Эксперт по породам собак */
/* Файл: Prim1.pro */
/* Назначение: Демонстрация работы продукционной ЭС */
/* Замечание: Это система для идентификации породы */
/* собак. Система использует множество */
/* продукционных правил для вывода */
  
```

```

/* решения */
domains
database
xpositive (symbol)
xnegative (symbol)
predicates
do_expert
consulting
in_out (symbol)
dog (symbol)
priznak (symbol)
negative (symbol)
remember (symbol,symbol)
clear_facts
goal
do_expert.
clauses
/* БЛОК ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА */
do_expert :-
makewindow (1,112,6,"ЭКСПЕРТНАЯ СИСТЕМА",1,12,22,58),
nl,write (" * * * * *"),
nl,write (" * ВЫБОР ПОРОДЫ СОБАК *"),
nl,write (" * *"),
nl,write (" * В Н И М А Н И Е ! *"),
nl,write (" * *"),
nl,write (" * При ответе на вопросы набирайте *"),
nl,write (" * 'yes' или 'no'. *"),
nl,write (" * * * * *"),
nl,nl,
consulting,
write ("Нажмите любую клавишу ."),nl,
readchar (_),
removewindow,
exit.
consulting :-
dog (X),!,nl,
write (" Собака имеет породу : ",X,"."),nl,
clear_facts.
consulting :-
nl,write (" Простите, я не могу Вам помочь ! "),
clear_facts.
in_out (Y) :-
write (" Вопрос :- Собака имеет ",Y," ?"),
readln (Reply),
remember (Y,Reply).
/* МЕХАНИЗМ ВЫВОДА */
priznak (Y) :-
xpositive (Y),!.
priznak (Y) :-
not (negative (Y)),
in_out (Y).
negative (Y) :-

```

```

xnegative (Y).
remember (Y,yes) :-
asserta (xpositive (Y)).
remember (Y,no) :-
asserta (xnegative (Y)),
fail.
clear_facts :-
retract (xnegative (_)),
fail.
clear_facts :-
retract (xpositive (_)),
fail.
/* ПРОДУКЦИОННЫЕ ПРАВИЛА */
dog ("Английский бульдог") :-
priznak ("шерсть короткую"),
priznak ("рост меньше 50 см"),
priznak ("хвост низкопосаженный"),
priznak ("характер хороший"),!.
dog ("Гончая") :-
priznak ("шерсть короткую"),
priznak ("рост меньше 50 см"),
priznak ("уши длинные"),
priznak ("характер хороший"),!.
/* и т. д. */
/* КОНЕЦ ПРОГРАММЫ */

```

Цель работы: получить практические навыки в представлении знаний продукционными правилами и в реализации экспертной системы с типовой структурой на языке логического программирования Пролог..

Рабочее задание

1. Изучить назначение базовых структур языка Пролог, применяемых для построения продукционной модели представления знаний:
 - правила вида "IF - THEN";
 - понятие "предикат".
2. Построить базу продукционных правил ЭС
3. Разработать диаграмму потоков данных для проектируемой интеллектуальной системы (экспертной системы).
4. С учётом диаграммы потоков и общей стратегии решения задач составить структурную схему, показывающую состав программных модулей и правил.
5. Приступить к написанию программ на Прологе, используя диаграмму потоков данных, структурную схему и продукционные правила.
6. Выполнить тестирование и отладку экспертной системы, в установленной на компьютере среде Prolog.

Контрольные вопросы:

1. Чем отличается программирование на Прологе от программирования на традиционных процедурных языках?
2. Какова структура программы на языке Пролог?
3. Как записываются продукционные знания на языке Пролог?

4. Какова типовая структура экспертной системы с продукционной базой знаний?
5. Как можно описать на Прологе базу фактов экспертной системы?
6. В чём состоит идея стратегии обратного логического вывода?

Общие положения

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующего занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

Форма отчета:

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

Время работы: **2 часа.**

Практическое занятие № 4.

Изучение структуры нейронной сети

Теоретическая часть

Сверточные нейронные сети (англ. - *convolutional neural network, CNN*) – это специальная архитектура нейронных сетей, которая была предложена Яном Лекуном в уже далеком 1988 году. Можно дать следующее определение:

«Сверточная нейронная сеть – тип глубокой нейронной сети, показывающий наилучшие результаты в области распознавания изображений»

Преимущества и недостатки сверточных нейронных сетей

1. Один из лучших алгоритмов по распознаванию и классификации изображений.
2. По сравнению с полносвязной нейронной сетью (типа персептрона) — гораздо меньшее количество настраиваемых весов. Нейросеть обобщает информацию, а не запоминает пиксели каждой показанной картинке в мириадах весовых коэффициентов, как это делает персептрон (*это стандартная формулировка из сети Интернет – на самом деле в СНС и используется многослойный персептрон! Т.е. существует некоторое недопонимание топологии*).
3. Удобное распараллеливание вычислений, а, следовательно, возможность реализации алгоритмов работы и обучения сети на графических процессорах.
4. Относительная устойчивость к повороту и сдвигу распознаваемого изображения.
5. Обучение при помощи классического метода обратного распространения ошибки.

К недостаткам можно отнести:

1. При работе с реальными данными в практических задачах учитывается большое количество гиперпараметров, т.е. на самом деле существует проблема сложности настройки.
2. При работе с большими данными сеть требует значительных аппаратных ресурсов.

Архитектура сверточной нейронной сети

Сверточная нейронная сеть состоит из следующего типа слоев:

- слой свертки (*convolutional layer*) – главный блок сверточной нейронной сети (также называют *сверточный слой*).
- слой подвыборки (*subsampling layer*)
- слой активации (σ) – получает на вход скалярный результат каждой свертки. При этом можно считать, что функция (или слой) активации встроена в слой свертки.
- несколько слоев обычного многослойного персептрона.

Сверточные слои представляют собой слои на которых происходит операция свертки (отсюда и название данных слоев) – ядро свертки, представляющее из себя матрицу весов (обычно это матрица 3x3) «скользит» над двумерным изображением, поэлементно выполняя операцию умножения с той частью входных данных, над которой оно сейчас находится, и затем суммирует все полученные значения в один выходной пиксель

Ядро свертки

Ядро свертки есть матрица, обычно 3x3, «пробегает» над двумерным изображением (но лучше сказать – матрицей, ведь не только с изображениями работает сверточная нейронная сеть) и преобразует в двумерную карту признаков. При этом признаки являются взвешенными суммами входных признаков (*расположенных примерно в том же месте, что и выходной пиксель на входном слое*).

То, что было рассмотрено выше называется операцией 2-D свертки и в таком случае изображение имеет один входной канал. Однако на практике большинство изображений имеют минимум 3 выходных канала (по цветам RGB – Red, Green, Blue). И в таком случае ядро свертки будет разным для каждого входного канала, а сумма ядер будет называться фильтром. Таким образом, операция 3-D свертки будет проходить немного иначе⁹.

Для того чтобы создать выходной канал – карту признаков, каждое ядро скользит по своему входному каналу, создавая свою версию этого канала. Затем все эти версии суммируются и создают новый общий канал, к которому добавляется базовое смещение, образуя при этом новую карту признаков (активации).

После одного или нескольких слоев свертки обычно идет слой подвыборки, где происходит уменьшение размерности изображения путем сжатия блоков признаков пикселей (обычно размером 2x2) до одного признака с использованием функции максимума. Можно так же использовать и другие функции, например, среднего значения, однако на практике преимущество пуллинга с функцией максимума неоспоримо.

После чередования операций свертки и подвыборки данные преобразуются из двумерного формата в одномерный и передаются в полносвязный слой обычной нейронной сети. Последний слой обеспечивает классификацию изображения.

Сам классификатор представлен слоями обычных нейронов, связанных друг с другом.

Принцип работы сверточной нейронной сети

Суть работы сверточной нейронной сети заключается в переходе от конкретных особенностей изображения к более абстрактным деталям за несколько итераций к выделению понятий более высокого уровня. Считается, что сеть самонастраивается и сама формирует необходимую иерархию абстрактных признаков, отбрасывая при этом незначительные детали и выделяя существенные. Функция Softmax применяется в машинном обучении для задач классификации, когда количество возможных классов больше двух (для двух классов используется логистическая функция). Координаты f_i полученного вектора при этом трактуются как вероятности того, что объект принадлежит к классу i .

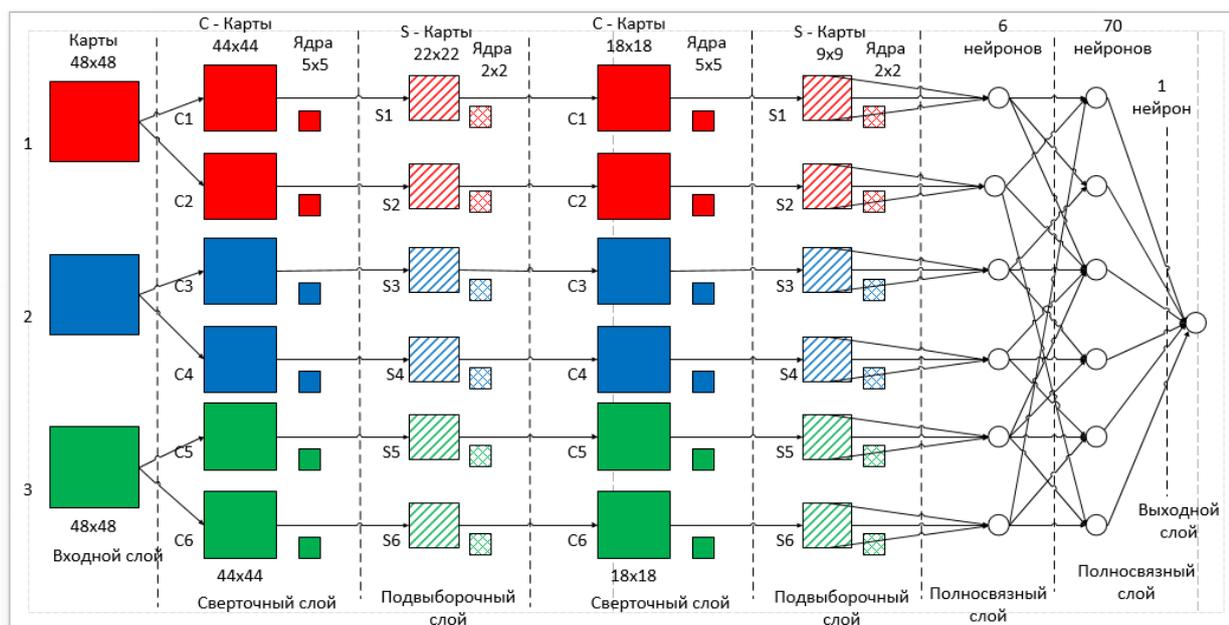


Рисунок 15. Вариант топологии сверточной нейронной сети

Функция потерь

Функция потерь вычисляет эффективность нейронной сети по отношению обучающей выборки к необходимым результатам. Эта функция одномерна. Варианты расчета функции потерь для СНС – можно выделить следующие:

- перекрестная энтропия;
- с.к.о.;
- экспоненциальная;
- расстояние Кульбака – Лейблера и некоторые другие.

-основной момент – необходимо рассчитать ошибку между априорными и полученными данными; по ряду мнений – перекрестная энтропия является более динамичной и предпочтительна для использования в качестве оценки функции потерь.

Часто вводятся метрики precision – точность, recall – полнота. Precision можно интерпретировать как долю объектов, названных классификатором положительными и при этом действительно являющимися положительными, а recall показывает, какую долю объектов положительного класса из всех объектов положительного класса нашел алгоритм. Существует несколько различных способов объединить precision и recall в агрегированный критерий качества. Для этого существует параметр f1-score, являющийся F-мерой – средним гармоническим между параметрами precision и recall, имеющий представление, описанное формулой

$$F_{\beta} = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

где β – вес точности в метрике. И здесь, конечно, необходимо учитывать количество экземпляров изображений данного класса.

Реализация сверточной нейронной сети с помощью TensorFlow

Построение нейронной сети требует настройки слоёв, а затем сборки модели с функциями оптимизации и потерь. Базовым элементом при построении нейронной сети является слой. Слой извлекает представление из данных, которые поступили ему на вход.

Пусть сеть состоит из шести слоёв (согласно исходного кода ниже):

1. Входного `tf.keras.layers.Flatten` – этот слой преобразует изображения размером 224x224 пикселей в 1D-массив. На этом слое у нас нет никаких параметров для обучения, так как этот слой занимается только преобразованием входных данных.

2. Скрытый слой «fc1» – полносвязный слой из 128 нейронов. Каждый нейрон принимает на вход все значения с предыдущего слоя, изменяет входные значения согласно внутренним весам и смещениям во время тренировки и возвращает единственное значение на следующий слой.

3. Под этим пунктом описано обобщение скрытых слоёв «fc2», «fc3», «fc4». Все эти слои полносвязные и выполняют те же функции, что слой «fc1». Они реализованы для того, чтобы иметь большую возможность для обучения. Единственное отличие слоя «fc4» от других слоёв – он содержит 64 нейрона.

3. Между слоями указан параметр Dropout – метод регуляризации искусственных нейронных сетей, предназначен для уменьшения переобучения сети за счет предотвращения сложных коадаптаций отдельных нейронов на тренировочных данных во время обучения.

4. Выходной слой `ts.keras.layers.Dense` – softmax-слой состоит из пяти нейронов, каждый из которых представляет определенный класс элемента одежды. Как и в предыдущем слое, каждый нейрон принимает на вход значения всех 64 нейронов предыдущего слоя. Веса и смещения каждого нейрона на этом слое изменяются при обучении таким образом, чтобы результирующее значение было в интервале от нуля до единицы, и представляло собой вероятность того, что изображение относится к этому классу. Сумма всех выходных значений пяти нейронов равна одному. Пример инициализации слоёв приведён в листинге ниже.

Листинг – Инициализация слоёв

```
base_model = model1 # Топлесс# Добавление верхнего слоя
x = base_model.output
x = Flatten()(x)
x = Dense(128, activation='relu', name='fc1')(x)
x = Dense(128, activation='relu', name='fc2')(x)
x = Dense(128, activation='relu', name='fc3')(x)
x = Dropout(0.5)(x)
x = Dense(64, activation='relu', name='fc4')(x)
predictions = Dense(5, activation='softmax')(x)
model = Model(inputs=base_model.input, outputs=predictions)
// выполнение настроек
model.compile(optimizer=tf.optimizers.Adam(),
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

После реализации необходимо обучить модель. За выполнение этой операции отвечает функция `model.fit`. Код данной функции представлен в листинге ниже.

Листинг – Функция запуска обучения нейросети

```
model.fit(X_train_rgb, y_train_rgb, epochs=200, batch_size=16,
         validation_data=(X_test_rgb, y_test_rgb), verbose=1, callbacks
         =[early_stopping, model_checkpoint])
```

Для начала нужно выделить последовательность действий при обучении на тренировочном наборе данных:

1. Набор входных данных необходимо повторять бесконечное количество раз, используя метод `dataset.repeat()`. Параметр `epochs` определяет количество всех обучающих итераций для выполнения;

2. Метод `dataset.shuffle()` перемешивает все изображения для того, чтобы на обучение модели не влиял порядок подачи входных данных;

3. Метод `dataset.batch()` сообщает методу тренировки `model.fit` использовать блоки изображений и метки при обновлении внутренних переменных модели.

Обучение происходит посредством вызова метода `model.fit`:

- `train_dataset` отправляется на вход модели;

- модель учится сопоставлять входное изображение с меткой;

параметр `epochs=200` ограничивает количество тренировок до полных обучающих итераций по набору данных, что в итоге дает тренировку на нескольких примерах.

Во многих высокоуровневых модулях для работы с НС (нейронные сети) существуют функции чтения/записи обученных моделей или только их весов из/в файлы формата HDF5.

Цель работы: Изучение структуры и разработка сверточной нейронной сети

Рабочее задание:

Общее задание – реализовать программно сверточную нейронную сеть и решить с ее помощью задачу классификации (использовать многослойный персептрон из первого практического занятия).

1. Реализовать операцию свертка
2. Реализовать сверточный и субдискредитирующий слои
3. Состыковать сверточные и субдискредитирующие слои многослойным персептроном.
4. Обучить разработанную СНС классификации объектов выбранной предметной области.
5. Реализовать многослойный персептрон с использованием библиотеки TensorFlow и провести эксперименты.

В процессе экспериментов, после программной разработки СНС, необходимо оценить следующие параметры для решаемой задачи:

А. скорость ответа (на 3-4 различных ЭВМ, с указанием типа процесса, его тактовой частоты и оперативной памяти: форм-фактор, тактовая частота, тактовая частота шины, пропускная способность, объем);

Б. точность решения задачи (например, точность распознавания изображения), в зависимости от размера входных карт.

Контрольные вопросы

1. Принцип работы слоя пуллинга.
2. Чем сверточная нейронная сеть отличается от стандартного многослойного персептрона?
3. Какие задачи можно решать с помощью сверточной нейронной сети?
4. Можно ли прогнозировать с помощью сверточной нейронной сети? Если да, то каким образом?
5. Как происходит свертка входного изображения для сверточной нейронной сети?
6. Сколько слоев необходимо для свертки изображения?
7. Какие подходы Вы знаете для ускорения процесса обучения сверточной нейронной сети?

Общие положения

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующего занятия.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

Форма отчета:

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

Время работы: 2 часа.

Практическое занятие №5.

Подготовка обучающей выборки для заданной предметной области

Теоретическая часть

Предварительная подготовка данных

После того как определена задача и исходные данные для обучения, цель для оптимизации и порядок оценки предпринятого подхода, у вас есть практически все, чтобы начать обучение модели. Но прежде необходимо преобразовать исходные данные в формат, в котором их можно передать в модель машинного обучения — в данном случае в глубокую нейронную сеть:

данные должны быть представлены в виде тензора

значения, помещаемые в тензоры, обычно требуют масштабирования и приведения к меньшим величинам: например, в диапазоне $[-1, 1]$ или $[0, 1]$

если значения разных признаков находятся в разных диапазонах (разнородные данные), их следует нормализовать

возможно, понадобится выполнить конструирование признаков,

особенно при небольшом объеме исходных данных. То есть из исходных признаков сформировать новые в пространстве меньшей размерности. Например, можно использовать метод главных компонент.

Определение задачи и создание набора данных

Определение того, какой вид имеют входные данные.

На основе этой информации определяется, какая задача решается (что необходимо предсказать).

На данном этапе определяется, какой тип обучения используется:

Контролируемое обучение (обучение с учителем)

Неконтролируемое обучение (обучения без учителя)

Самоконтролируемое обучение

Обучение с подкреплением

На данном этапе необходимо помнить о двух гипотезах, которые должны быть справедливы к данным:

гипотеза о том, что выходные данные можно предсказать по входным данным

гипотеза о том, что доступные данные достаточно информативны для изучения отношений между входными и выходными данными

Выбор меры успеха

На данном этапе определяется, что означает успешная модель. То есть то, как определяется хорошая и правильно обученная модель от плохой. Это может быть: близость, точность и полнота, счет, и.т.д. На основе этой информации выбирается функция, которую необходимо будет оптимизировать.

Выбор протокола оценки

После определения цели необходимо также выяснить, как измерять движение к ней. Существует 3 распространённых протокола оценки:

выделение из общей выборки отдельного проверочного набора данных

Смысл состоит в следующем. Некоторая часть данных выделяется в контрольный набор. Обучение производится на оставшихся данных, а оценка качества — на контрольных. Как уже говорилось, для предотвращения утечек информации модель не должна настраиваться по результатам прогнозирования на контрольных данных, поэтому требуется также зарезервировать отдельный проверочный набор.

Это самый простой протокол оценки, страдающий одним существенным недостатком: при небольшом объеме доступных данных проверочный и контрольный наборы могут содержать слишком мало образцов, чтобы считаться статистически репрезентативными.

Перекрестная проверка по К блокам

При использовании этого подхода данные разбиваются на К блоков равного размера. Для каждого блока i производится обучение модели на остальных $K-1$ блоках и оценка на блоке i .

Окончательная оценка рассчитывается как среднее К промежуточных оценок. Этот метод может пригодиться, когда качество модели слишком сильно зависит от деления данных на тренировочный/контрольный наборы. Подобно проверке с простым расщеплением выборки, этот метод не избавляет от необходимости использовать отдельный проверочный набор для калибровки модели.

```
k = 4

num_validation_samples = len(data) // k
np.random.shuffle(data)
validation_scores = []

for fold in range(k):

    validation_data = data[num_validation_samples * fold: num_validation_samples *
(fold + 1)]

    training_data = data[:num_validation_samples * fold] + data[num_validation_samples
* (fold + 1):]

    model = get_model()
```

Итерационная проверка по К блокам с перемешиванием

Этот метод подходит для ситуаций, когда имеется относительно небольшой набор данных и требуется оценить модель максимально точно. Суть его заключается в многократном применении перекрестной проверки по К блокам с перемешиванием данных перед каждым разделением на К блоков. Конечная оценка — среднее по оценкам, полученным в прогонах перекрестной проверки по К блокам. Обратите внимание: в конечном счете обучению и оценке подвергается $P \times K$ моделей (где P — количество итераций), что может быть очень затратным.

Цель работы: Подготовить данные для работы сверточной нейронной сети. Построить сверточную нейронную сеть, которая будет классифицировать черно-белые изображения с простыми геометрическими фигурами на них.

Рабочее задание:

К каждому варианту прилагается код, который генерирует изображения.

Для генерации данных необходимо вызвать функцию `gen_data`, которая возвращает два тензора:

1. Тензор с изображениями ранга 3
2. Тензор с метками классов

Обратите внимание:

- Выборки не перемешаны, то есть наблюдения классов идут по порядку
- Классы характеризуются строковой меткой
- Выборка изначально не разбита на обучающую, контрольную и тестовую
- Скачивать необходимо оба файла. Подключать файл, который начинается с `var` (в нем и находится функция `gen_data`)

Варианты заданий

Вариант 1

Классификация изображений с прямоугольником или кругом.

[Файл 1](#)

[Файл 2](#)

Вариант 2

Классификация изображений с закрашенным или не закрашенным кругом

[Файл 1](#)

[Файл 2](#)

Вариант 3

Классификация изображений с горизонтальной или вертикальной линией

[Файл 1](#)

[Файл 2](#)

Вариант 4

Классификация изображений с крестом или с линией (может быть горизонтальной или вертикальной)

[Файл 1](#)

[Файл 2](#)

Вариант 5

Классификация изображений с прямоугольником или не закрашенным кругом

[Файл 1](#)

[Файл 2](#)

Вариант 6

Классификация изображений по количеству крестов на них. Может быть 1, 2 или 3

[Файл 1](#)

[Файл 2](#)

Вариант 7

Классификация изображений по количеству линий на них. Может быть 1, 2 или 3

[Файл 1](#)

[Файл 2](#)

Контрольные вопросы

- 1.Какой этап предшествует обучению нейронной сети в библиотеке Keras?
- 2.Как создать нейронную сеть в Keras и какие компоненты она содержит?
- 3.Как организовать данные для обучения нейронной сети в Keras?
- 4.Как выбрать функцию потерь (loss function) для обучения нейронной сети в Keras?
- 5.Как выбрать оптимизатор (optimizer) для обучения нейронной сети в Keras?
- 6.Как оценить производительность нейронной сети после обучения в Keras?
- 7.Как обработать переобучение (overfitting) нейронной сети в Keras?
- 8.Как подобрать подходящую архитектуру нейронной сети для задачи в Keras?
- 9.Какие инструменты предоставляет Keras для улучшения эффективности обучения нейронных сетей?
- 10.Какие возможности предоставляет Keras для интеграции нейронных сетей в реальные приложения?

Общие положения

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующему занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

Форма отчета:

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

Время работы: 2 часа.

Практическое занятие № 6

Обучение и тестирование нейронной сети для работы с изображениями

Теоретическая часть

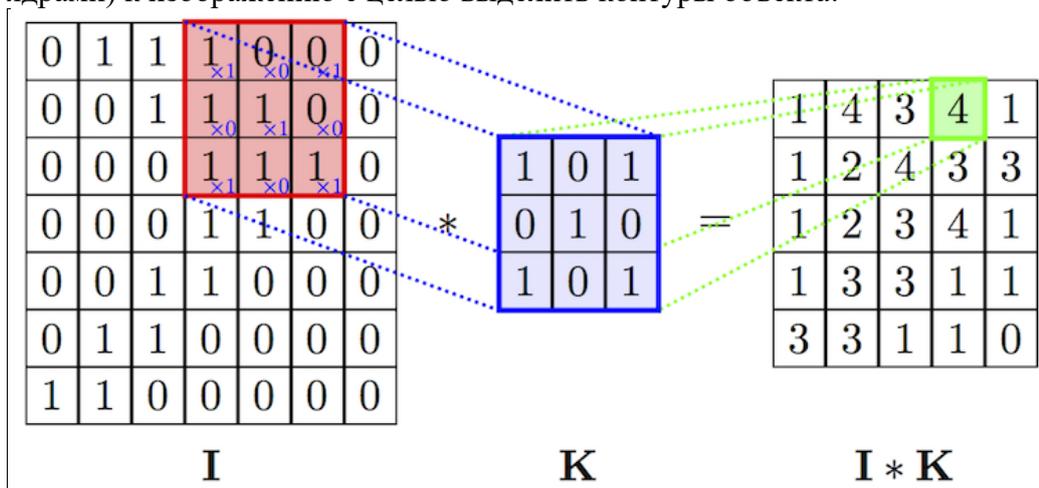
Полносвязный перцептрон обычно не выбирают для задач, связанных с распознаванием изображений — в этом случае намного чаще пользуются преимуществами сверточных нейронных сетей (Convolutional Neural Networks, CNN).

Посмотрим, что происходит с количеством параметров (весов) в модели MLP, когда ей на вход поступают необработанные данные. Например, CIFAR-10 содержит $32 \times 32 \times 3$ пикселей, и если мы будем считать каждый канал каждого пикселя независимым входным параметром для MLP, каждый нейрон в первом скрытом слое добавляет к модели около 3000 новых параметров. И с ростом размера изображений ситуация быстро выходит из-под контроля, причем происходит это намного раньше, чем изображения достигают того размера, с которыми обычно работают пользователи реальных приложений.

Одно из популярных решений — понижать разрешение изображений до той степени, когда MLP становится применим. Тем не менее, когда мы просто понижаем разрешение, мы рискуем потерять большое количество информации, и было бы здорово, если бы можно было осуществлять полезную первичную обработку информации еще до применения понижения качества, не вызывая при этом взрывного роста количества параметров модели.

Существует весьма эффективный способ решения этой задачи, который обращает в нашу пользу саму структуру изображения: предполагается, что пиксели, находящиеся близко друг к другу, теснее “взаимодействуют” при формировании интересующего нас признака, чем пиксели, расположенные в противоположных углах. Кроме того, если в процессе классификации изображения небольшая черта считается очень важной, не будет иметь значения, на каком участке изображения эта черта обнаружена.

Введем понятие оператора свертки. Имея двумерное изображение I и небольшую матрицу K размерности $h \times w$ (так называемое ядро свертки), построенная таким образом, что графически кодирует какой-либо признак, мы вычисляем свернутое изображение $I * K$, накладывая ядро на изображение всеми возможными способами и записывая сумму произведений элементов исходного изображения и ядра. Результат применения операции свертки (с двумя разными ядрами) к изображению с целью выделить контуры объекта:





Сверточные и субдискретизирующие слои

Оператор свертки составляет основу сверточного слоя (convolutional layer) в CNN. Слой состоит из определенного количества ядер \vec{K} (с аддитивными составляющими смещения \vec{b} для каждого ядра) и вычисляет свертку выходного изображения предыдущего слоя с помощью каждого из ядер, каждый раз прибавляя составляющую смещения. В конце концов ко всему выходному изображению может быть применена функция активации σ . Обычно входной поток для сверточного слоя состоит из d каналов, например, red/green/blue для входного слоя, и в этом случае ядра тоже расширяют таким образом, чтобы они также состояли из d каналов; получается следующая формула для одного канала выходного изображения сверточного слоя, где K — ядро, а b — составляющая смещения:

$$\text{conv}(I, K)_{x,y} = \sigma\left(b + \sum_{i=1}^h \sum_{j=1}^w \sum_{k=1}^d K_{ijk} \times I_{x+i-1,y+j-1,k}\right)$$

Обратите внимание, что так как все, что мы здесь делаем — это сложение и масштабирование входных пикселей, ядра можно получить из имеющейся обучающей выборки методом градиентного спуска, аналогично вычислению весов в многослойном перцептроне (MLP). На самом деле MLP мог бы в совершенстве справиться с функциями сверточного слоя, но времени на обучение (как и обучающих данных) потребовалось бы намного больше.

Заметим также, что оператор свертки вовсе не ограничен двухмерными данными: большинство фреймворков глубокого обучения (включая Keras) предоставляют слои для одномерной или трехмерной свертки прямо “из коробки”.

Стоит также отметить, что хотя сверточный слой сокращает количество параметров по сравнению с полносвязным слоем, он использует больше гиперпараметров — параметров, выбираемых до начала обучения.

В частности, выбираются следующие гиперпараметры:

Глубина (depth) — сколько ядер и коэффициентов смещения будет задействовано в одном слое;

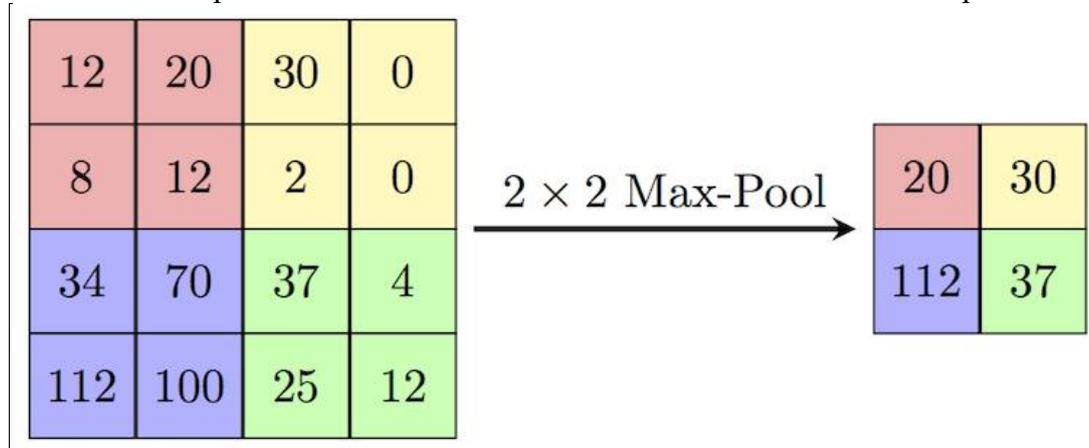
Высота (height) и ширина (width) каждого ядра;

Шаг (stride) — на сколько смещается ядро на каждом шаге при вычислении следующего пикселя результирующего изображения. Обычно его принимают равным 1, и чем больше его значение, тем меньше размер выходного изображения;

Отступ (padding): заметим, что свертка любым ядром размерности более, чем 1×1 уменьшит размер выходного изображения. Так как в общем случае желательно сохранять размер исходного изображения, рисунок дополняется нулями по краям.

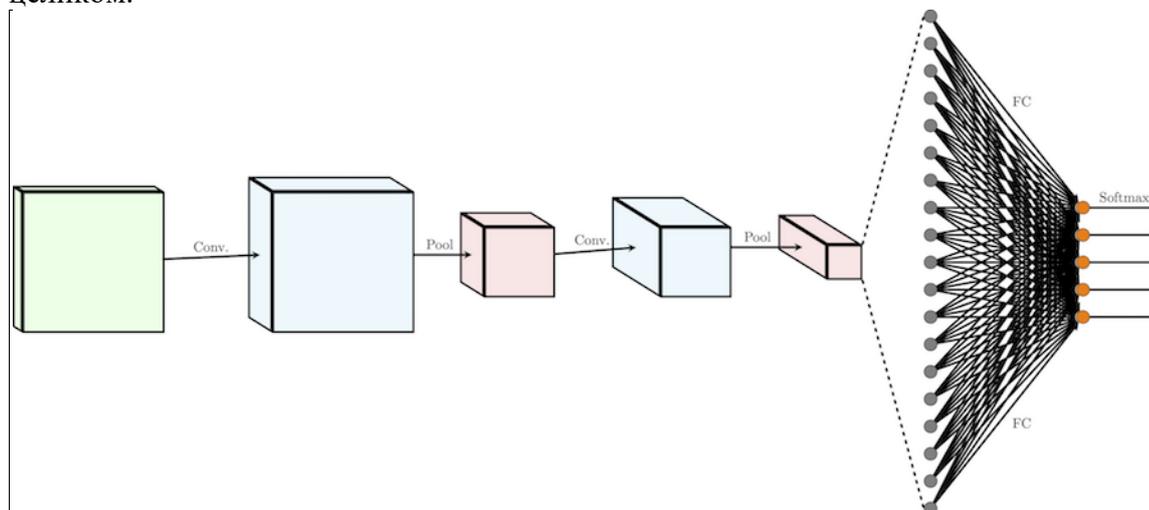
Операции свертки — не единственные операции в CNN (хотя существуют многообещающие исследования на тему “чисто-сверточных” сетей); они чаще применяются для выделения наиболее полезных признаков перед субдискретизацией (downsampling) и последующей обработкой с помощью MLP.

Популярный способ субдискретизации изображения — слой подвыборки (также называемый слоем субдискретизации, по-английски *downsampling* или *pooling layer*), который получает на вход маленькие отдельные фрагменты изображения (обычно 2x2) и объединяет каждый фрагмент в одно значение. Существует несколько возможных способов агрегации, наиболее часто из четырех пикселей выбирается максимальный. Этот способ схематически изображен ниже.



Итого: обычная CNN

Теперь, когда у нас есть все строительные блоки, давайте рассмотрим, как выглядит обычная CNN целиком.



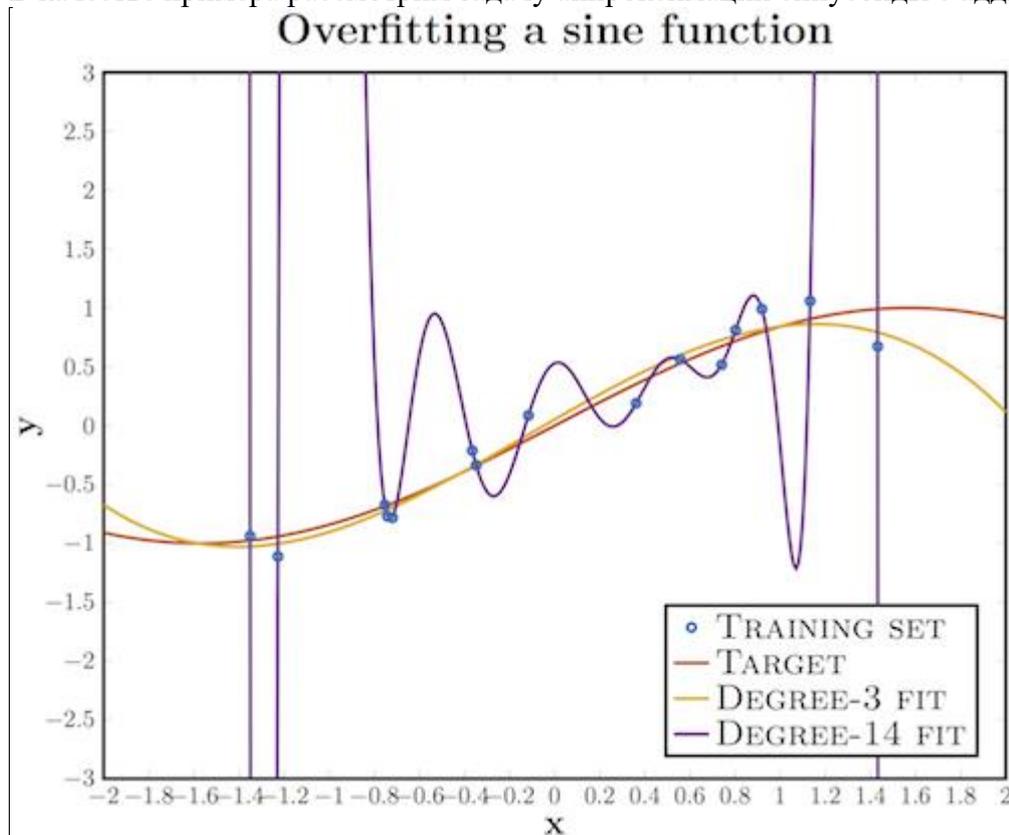
Обычную архитектуру CNN для распределения изображений по k классам можно разделить на две части: цепочка чередующихся слоев свертки/подвыборки Conv \rightarrow Pool (иногда с несколькими слоями свертки подряд) и несколько полносвязных слоев (принимающих каждый пиксель как независимое значение) с слоем softmax в качестве завершающего. Я не говорю здесь о функциях активации, чтобы наша схема стала проще, но не забывайте, что обычно после каждого сверточного или полносвязного слоя ко всем выходным значениям применяется функция активации, например, ReLU.

Один проход Conv \rightarrow Pool влияет на изображение следующим образом: он сокращает длину и ширину определенного канала, но увеличивает его значение (глубину).

Softmax и перекрестная энтропия более подробно рассмотрены на предыдущем уроке. Напомним, что функция softmax превращает вектор действительных чисел в вектор вероятностей (неотрицательные действительные числа, не превышающие 1). В нашем контексте выходные значения являются вероятностями попадания изображения в определённый класс. Минимизация потерь перекрестной энтропии обеспечивает уверенность в определении принадлежности изображения определенному классу, не принимая во внимание вероятность остальных классов, таким образом, для вероятностных задач softmax предпочтительней, чем, например, метод квадратичной ошибки.

Переобучение, регуляризация и dropout:

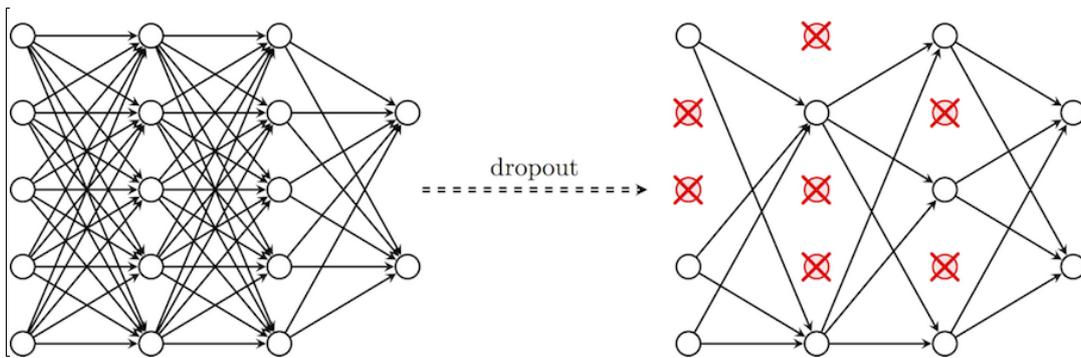
Переобучение — это излишне точное соответствие нейронной сети конкретному набору обучающих примеров, при котором сеть теряет способность к обобщению. Другими словами, наша модель могла выучить обучающее множество (вместе с шумом, который в нем присутствует), но она не смогла распознать скрытые процессы, которые это множество породили. В качестве примера рассмотрим задачу аппроксимации синусоиды с аддитивным шумом.



У нас есть обучающее множество (синие кружки), полученное из исходной кривой синуса, с некоторым количеством шума. Если мы приложим к этим данным график многочлена третьей степени, мы получим хорошую аппроксимацию исходной кривой. Кто-то возразит, что многочлен 14-й степени подошел бы лучше; действительно, так как у нас есть 15 точек, такая аппроксимация идеально описала бы обучающую выборку. Тем не менее, в этом случае введение дополнительных параметров в модель приводит к катастрофическим результатам: из-за того, что наша аппроксимация учитывает шумы, она не совпадает с исходной кривой нигде, кроме обучающих точек.

У глубоких сверточных нейронных сетей масса разнообразных параметров, особенно это касается полносвязных слоев. Переобучение может проявить себя в следующей форме: если у нас недостаточно обучающих примеров, маленькая группа нейронов может стать ответственной за большинство вычислений, а остальные нейроны станут избыточны; или наоборот, некоторые нейроны могут нанести ущерб производительности, при этом другие нейроны из их слоя не будут заниматься ничем, кроме исправления их ошибок.

Чтобы помочь нашей сети не утратить способности к обобщению в этих обстоятельствах, мы вводим приемы регуляризации: вместо сокращения количества параметров, мы накладываем ограничения на параметры модели во время обучения, не позволяя нейронам изучать шум обучающих данных. Здесь я опишу прием dropout, который сначала может показаться “черной магией”, но на деле помогает исключить ситуации, описанные выше. В частности, dropout с параметром p за одну итерацию обучения проходит по всем нейронам определенного слоя и с вероятностью p полностью исключает их из сети на время итерации. Это заставит сеть обрабатывать ошибки и не полагаться на существование определенного нейрона (или группы нейронов), а полагаться на “единое мнение” (consensus) нейронов внутри одного слоя. Это довольно простой метод, который эффективно борется с проблемой переобучения сам, без необходимости вводить другие регуляризаторы. Схема ниже иллюстрирует данный метод.



Реализация сети

В качестве практической части построим глубокую сверточную нейронную сеть и применим ее к классификации изображений из набора CIFAR-10.

Импорты:

```
from keras.datasets import cifar10
from keras.models import Model
from keras.layers import Input, Convolution2D, MaxPooling2D,
Dense, Dropout, Flatten
from keras.utils import np_utils
import numpy as np
```

Как уже говорилось, обычно CNN использует больше гиперпараметров, чем MLP. В этом руководстве мы все еще будем использовать заранее известные “хорошие” значения, но не будем забывать, что в последующей лекции я расскажу, как их правильно выбирать.

Зададим следующие гиперпараметры:

`batch_size` — количество обучающих образцов, обрабатываемых одновременно за одну итерацию алгоритма градиентного спуска;

`num_epochs` — количество итераций обучающего алгоритма по всему обучающему множеству;

`kernel_size` — размер ядра в сверточных слоях;

`pool_size` — размер подвыборки в слоях подвыборки;

`conv_depth` — количество ядер в сверточных слоях;

`drop_prob` (`dropout probability`) — мы будем применять `dropout` после каждого слоя подвыборки, а также после полносвязного слоя;

`hidden_size` — количество нейронов в полносвязном слое MLP.

```
batch_size = 32 # in each iteration, we consider 32 training
examples at once
num_epochs = 200 # we iterate 200 times over the entire
training set
kernel_size = 3 # we will use 3x3 kernels throughout
pool_size = 2 # we will use 2x2 pooling throughout
conv_depth_1 = 32 # we will initially have 32 kernels per conv.
layer...
conv_depth_2 = 64 # ...switching to 64 after the first pooling
layer
drop_prob_1 = 0.25 # dropout after pooling with probability
0.25
drop_prob_2 = 0.5 # dropout in the dense layer with probability
0.5
hidden_size = 512 # the dense layer will have 512 neurons
```

Загрузка и первичная обработка CIFAR-10 осуществляется ровно так же, как и загрузка и обработка MNIST, где Keras выполняет все автоматически. Единственное отличие состоит в том, что теперь мы не рассматриваем каждый пиксель как независимое входное значение, и поэтому

мы не переносим изображение в одномерное пространство. Мы снова преобразуем интенсивность пикселей так, чтобы она попадала в отрезок [0,1] и используем прямое кодирование для выходных значений.

Тем не менее, в этот раз этот этап будет выполнен для более общего случая, что позволит проще приспособливаться к новым наборам данных: размер будет не жестко задан, а вычислен из размера набора данных, количество классов будет определено по количеству уникальных меток в обучающем множестве, а нормализация будет выполнена путем деления всех элементов на максимальное значение обучающего множества.

```
(X_train, y_train), (X_test, y_test) = cifar10.load_data() #
fetch CIFAR-10 data
num_train, depth, height, width = X_train.shape # there are
50000 training examples in CIFAR-10
num_test = X_test.shape[0] # there are 10000 test examples in
CIFAR-10
num_classes = np.unique(y_train).shape[0] # there are 10 image
classes
X_train = X_train.astype('float32')
X_test = X_test.astype('float32')
X_train /= np.max(X_train) # Normalise data to [0, 1] range
X_test /= np.max(X_train) # Normalise data to [0, 1] range
Y_train = np_utils.to_categorical(y_train, num_classes) # One-
hot encode the labels
Y_test = np_utils.to_categorical(y_test, num_classes) # One-hot
encode the labels
```

Настало время моделирования! Наша сеть будет состоять из четырех слоев Convolution_2D и слоев MaxPooling2D после второй и четвертой сверток. После первого слоя подвыборки мы удваиваем количество ядер (вместе с описанным выше принципом принесения высоты и ширины в жертву глубине). После этого выходное изображение слоя подвыборки трансформируется в одномерный вектор (слоем Flatten) и проходит два полносвязных слоя (Dense). На всех слоях, кроме выходного полносвязного слоя, используется функция активации ReLU, последний же слой использует softmax.

Для регуляризации нашей модели после каждого слоя подвыборки и первого полносвязного слоя применяется слой Dropout. Здесь Keras также выделяется на фоне остальных фреймворков: в нем есть внутренний флаг, который автоматически включает и выключает dropout, в зависимости от того, находится модель в фазе обучения или тестирования.

```
inp = Input(shape=(depth, height, width)) # N.B. depth goes
first in Keras
# Conv [32] -> Conv [32] -> Pool (with dropout on the pooling
layer)
conv_1 = Convolution2D(conv_depth_1, kernel_size, kernel_size,
border_mode='same', activation='relu')(inp)
conv_2 = Convolution2D(conv_depth_1, kernel_size, kernel_size,
border_mode='same', activation='relu')(conv_1)
pool_1 = MaxPooling2D(pool_size=(pool_size, pool_size))(conv_2)
drop_1 = Dropout(drop_prob_1)(pool_1)
# Conv [64] -> Conv [64] -> Pool (with dropout on the pooling
layer)
conv_3 = Convolution2D(conv_depth_2, kernel_size, kernel_size,
border_mode='same', activation='relu')(drop_1)
conv_4 = Convolution2D(conv_depth_2, kernel_size, kernel_size,
border_mode='same', activation='relu')(conv_3)
pool_2 = MaxPooling2D(pool_size=(pool_size, pool_size))(conv_4)
```

```

drop_2 = Dropout(drop_prob_1)(pool_2)
# Now flatten to 1D, apply Dense -> ReLU (with dropout) ->
softmax
flat = Flatten()(drop_2)
hidden = Dense(hidden_size, activation='relu')(flat)
drop_3 = Dropout(drop_prob_2)(hidden)
out = Dense(num_classes, activation='softmax')(drop_3)
model = Model(input=inp, output=out) # To define a model, just
specify its input and output layers
model.compile(loss='categorical_crossentropy', # using the
cross-entropy loss function
              optimizer='adam', # using the Adam optimiser
              metrics=['accuracy']) # reporting the accuracy
model.fit(X_train, Y_train, # Train the model using the
training set...
         batch_size=batch_size, nb_epoch=num_epochs,
         verbose=1, validation_split=0.1) # ...holding out 10%
of the data for validation
model.evaluate(X_test, Y_test, verbose=1) # Evaluate the
trained model on the test set!

```

Цель работы: Создание нейросети для распознавания объектов на фотографиях (Object Recognition in Photographs). Классификация небольших изображений по десяти классам: самолет, автомобиль, птица, кошка, олень, собака, лягушка, лошадь, корабль и грузовик.

Порядок выполнения работы

- Ознакомиться со сверточными нейронными сетями
- Изучить построение модели в Keras в функциональном виде
- Изучить работу слоя разреживания (Dropout)

Рабочее задание

1. Построить и обучить сверточную нейронную сеть
2. Исследовать работу сеть без слоя Dropout
3. Исследовать работу сети при разных размерах ядра свертки

Контрольные вопросы

1. Какие этапы включает в себя процесс создания нейронной сети для распознавания объектов на фотографиях?
2. Какие типы нейронных сетей чаще всего используются для задачи распознавания объектов на изображениях?
3. Как подготовить и обработать данные для обучения нейронной сети для распознавания объектов на фотографиях?
4. Как выбрать подходящую архитектуру нейронной сети для задачи распознавания объектов на фотографиях?
5. Какие техники предобработки изображений могут помочь улучшить производительность нейронной сети для распознавания объектов?
6. Как выбрать функцию потерь (loss function) для обучения нейронной сети для распознавания объектов на фотографиях?

7. Как подобрать оптимизатор (optimizer) для эффективного обучения нейронной сети для распознавания объектов на фотографиях?
8. Как решить проблему переобучения (overfitting) при обучении нейронной сети для распознавания объектов?
9. Как оценить качество работы нейросети для распознавания объектов на фотографиях?
10. Какие существуют практические применения созданных нейросетей для распознавания объектов на фотографиях в реальных задачах?

Общие положения

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующего занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

Форма отчета:

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

Время работы: **2 часа.**

Практическое занятие №7.

Построение, обучение и тестирование нейронной сети для работы со звуком и движениями

Теоретическая часть

Сверточные нейронные сети предназначены для обработки двумерных структур данных, прежде всего изображений. Сверточная сеть представляет собой комбинацию трех типов слоев:

- слои, которые выполняют функцию свертки над двумерными массивами данных (сверточные слои),
- слои, выполняющие функцию уменьшения формата данных (слой субдискретизации),
- полносвязные слои, завершающие процесс обработки данных.

Структура сверточных нейронных сетей принципиально многослойная. Работа сверточной нейронной сети обычно интерпретируется как переход от конкретных особенностей изображения к более абстрактным деталям, и далее к ещё более абстрактным деталям вплоть до выделения понятий высокого уровня. При этом сеть самонастраивается и вырабатывает необходимую иерархию абстрактных признаков (последовательности карт признаков), фильтруя маловажные детали и выделяя существенное [1, 3]. Примером классической сверточной нейронной сети является сеть VGG16 (рис. 3).

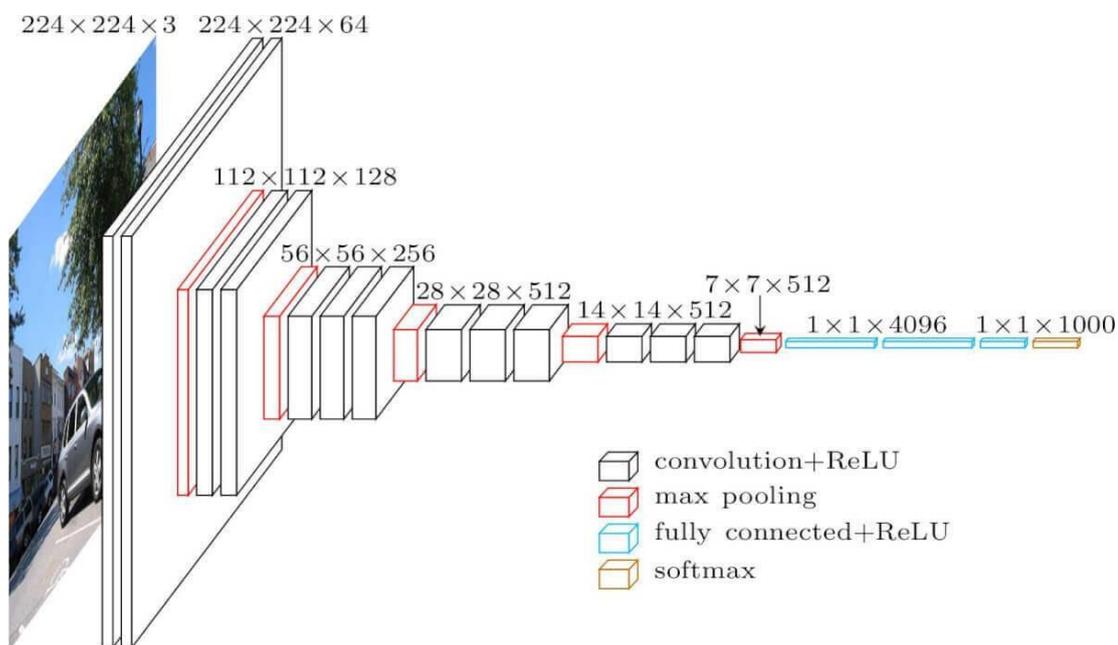


Рисунок 3 – Структура классической сети VGG16 [4]

Сеть VGG-16 имеет 16 слоев и способная работать с изображениями достаточно большого формата 224×224 пикселя [5]. В своей стандартной топологии эта сеть способна работать с датасетом изображений ImageNet, содержащем более 15 млн. изображений, разбитых на 22000 категорий [6].

Цель работы: Изучение работы нейронной сети с камерой в реальном времени

Порядок выполнения работы

Работа выполняется на персональном компьютере с установленной операционной системой Linux (Ubuntu 20.04 или старше) с подключенной к нему usb-камерой.

1. С помощью преподавателя или лаборанта включить компьютер и дождаться загрузки операционной системы.

2. На рабочем столе найти папку LR, зайти в нее и скопировать папку LR_neural_network на рабочий стол. В дальнейшем все действия по ходу работы выполнять в этой папке.

Пользуясь боковой панелью, запустить среду разработки Spyder и дождаться ее загрузки.

Загрузить программу VGG16.py в основное окно среды разработки. Для этого следует в меню **Файл** выбрать команду **Открыть**, в появившемся окне выбрать **Рабочий стол** и в открывшемся списке выбрать папку LR_neural_network. Далее в открывшемся списке выбрать файл VGG16.py. Посмотрите внимательно на основные элементы программы. Что она выполняет?

Проверьте, что в строчке с названием файла изображения выбран файл nemo0.jpg. Запустите программу, нажав на зеленый треугольник. Дождитесь окончания работы программы (первый запуск программы может занимать довольно много времени).

Найдите в консоли результаты работы программы. Для файла nemo0.jpg вы увидите:

```
Консоль 1/A x
In [1]: runfile('/home/ysn/Рабочий стол/LR/LR_neural_network/VGG16.py', wdir='/home/ysn/Рабочий стол/LR/LR_neural_network')
Img dimension (224, 224, 3)
ImageNet class 393
[('n02607072', 'anemone_fish', 0.968795), ('n01914609', 'sea_anemone', 0.02827463), ('n09256479', 'coral_reef', 0.0016436366), ('n02606052', 'rock_beauty', 0.0008524554), ('n01950731', 'sea_slug', 0.000101410966)]
```



Рисунок 8 – Пример результата работы программы VGG16.py

Вы видите, что на рисунке, помимо анализируемого изображения, содержится следующая информация:

Img dimension (224, 224, 3) — размерность анализируемого изображения;

ImageNet class 393 — класс изображения в датасете ImageNet (один из 1000);

[('n02607072', 'anemone_fish', 0.968795), ('n01914609', 'sea_anemone', 0.02827463), ('n09256479', 'coral_reef', 0.0016436366), ('n02606052',

'rock_beauty', 0.0008524554), ('n01950731', 'sea_slug', 0.000101410966)] —

информация, содержащая пять наиболее вероятных объектов из базы данных ImageNet с указанием вероятности распознавания объектов по убыванию.

Посмотрите, удачно ли распозналось исходное изображение нейронной сетью?

Откройте страницу с классификацией изображений ImageNet [6] и найдите в ней полученный класс.

Замените в тексте программы файл изображения на hippo1.jpg. Запустите программу на выполнение. Найдите в консоли результаты работы программы.

Выполните п.п.16-18 для всех файлов изображений, указанных в таблице 1. Внесите результаты работы нейронной сети в таблицу:

Таблица 1– Результаты работы нейронной сети VGG16

Название файла	Класс изображения ImageNet	Наиболее вероятный объект	Наибольшая вероятность объекта	Второй по вероятности объект	Вероятность второго объекта
nemo0.jpg	393	anemone_fish	0.968795	sea_anemone	0.02827463
hippo1.jpg					
hippo2.jpg					
lo1.jpg					
lo2.jpg					
lo3.jpg					
wb1.jpg					
lama1					
lama2					
camel1					
camel2					
cat1					
cat2					
cat3					
hb1					
street1					
street2					
face1					

Проанализируйте полученные результаты. Почему ряд объектов не распознается? Откройте страницу с классификацией изображений ImageNet [6] и найдите в ней полученный класс для таких изображений.

Рабочее задание

Загрузить программу ImageNet_camera.py в основное окно среды разработки. Для этого следует в меню **Файл** выбрать команду **Открыть**, в появившемся окне выбрать **Рабочий стол** и в открывшемся списке выбрать папку LR_neural_network. Далее в открывшемся списке выбрать файл ImageNet_camera.py. Посмотрите внимательно на основные элементы программы. Что она выполняет?

Проверьте, что в строчке с названием файла изображения выбран файл neto0.jpg. Запустите программу, нажав на зеленый треугольник. Дождитесь запуска программы (первый запуск программы может занимать довольно много времени). Вы получите окно с изображением, формируемым камерой, а в консоли будет выводиться информация о времени работы распознавания изображения нейронной сетью и двух наиболее вероятных объектах распознавания и соответствующих им вероятностях.

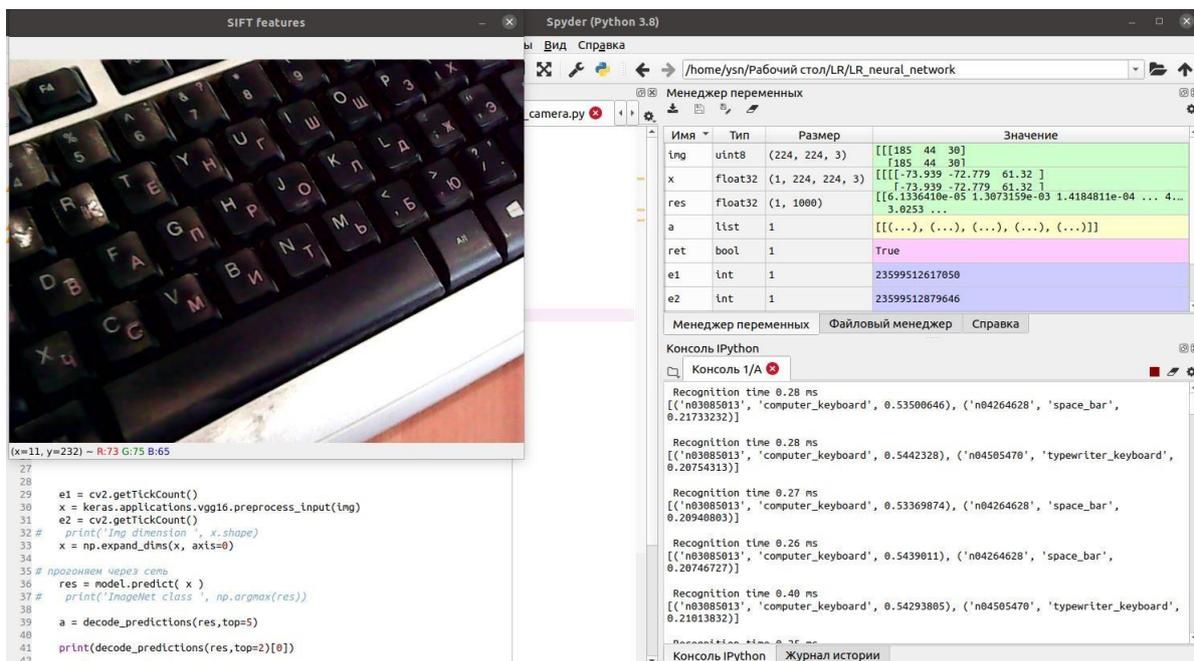


Рисунок 9 – Пример результата работы программы VGG16.py с видеочкамерой

Наведите камеру на один из знакомых объектов (например, компьютерная мышь, очки или часы). Дождитесь, когда камера настроится на новое изображение. Получите очередное окно с изображением от видеочкамеры. Посмотрите информацию о времени выполнения, распознанных предметах и вероятностях. Занесите параметры в таблицу 2.

Таблица 2 – Результаты работы нейронной сети VGG16 с камерой

Предмет	Время выполнения	Наиболее вероятный объект	Вероятность для наиболее вероятного объекта	Второй по вероятности объект	Вероятность для второго по вероятности объекта
Клавиатура компьютера					

....					
....					

Последовательно выберите десять знакомых объектов в вашем окружении. Направляйте на них камеру и фиксируйте информацию о времени выполнения, распознанных предметах и вероятностях в таблице 2. Для выхода из программы нажмите кнопку Q.

Скопируйте все полученные скриншоты на собственный носитель для составления отчета. Обратитесь к преподавателю или лаборанту для выключения компьютера.

Контрольные вопросы

1. Что подразумевается под искусственной нейронной сетью? Какова структура простой нейронной сети?
2. Укажите признаки, по которым классифицируют нейронные сети. Перечислите известные Вам классификации ИНС.
3. В чем состоит задача обучения нейронной сети для решения задачи распознавания объектов на изображении?
4. Какие методы могут использоваться при обучении нейронной сети, и какие трудности при этом возникают?
5. Что такое тестовый набор данных и с какой целью он используется?
6. В чем особенность структуры нейронной сети VGG16, каковы ее свойства и параметры?

Общие положения

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующего занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

Форма отчета:

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

Время работы: 2 часа.

Практическое занятие № 8

Создание нейронной сети для решения задачи аппроксимации функции.

Теоретическая часть

Постановка задачи аппроксимации

Аппроксимацией (приближением) функции $f(x)$ называется нахождение такой функции $g(x)$ (аппроксимирующей функции), которая была бы близка заданной. Разработано несколько критериев близости функций $f(x)$ и $g(x)$. Основной задачей аппроксимации является построение аппроксимирующей функции, наиболее близко проходящей около заданных точек или около графика непрерывной функции. Возникновение такой задачи связано с наличием погрешностей в исходных данных. В этом случае точно проводить функцию через все точки исходных данных, как это делается в интерполяции, нецелесообразно. Аппроксимация позволяет исследовать числовые характеристики и качественные свойства объекта, сводя задачу к изучению более простых или более удобных объектов (например, таких характеристик, которые легко вычисляются или свойства которых уже известны).

Близость исходной и аппроксимирующей функций определяется числовой мерой – критерием аппроксимации (близости). Наибольшее распространение получил критерий близости в заданных точках, равный сумме квадратов отклонений расчетных значений от «экспериментальных». Такой критерий называется «квадратичным»

$$R = \sum_{i=1}^n \beta_i (y_i - g_i)^2. \quad (3.1)$$

Здесь y_i – заданные табличные значения функции в i -й точке;

g_i – значения аппроксимирующей функции в i -й точке;

β_i – весовые коэффициенты, задающие относительную важность i -й точки таким образом, что рост β_i

при минимизации критерия R в первую очередь приводит к уменьшению отклонения в i -й точке.

g_i – значения аппроксимирующей функции в i -й точке;

β_i – весовые коэффициенты, задающие относительную важность i -й точки таким образом, что рост β_i

при минимизации критерия R в первую очередь приводит к уменьшению отклонения в i -й точке.

Квадратичный критерий обладает рядом достоинств, в числе которых дифференцируемость и наличие единственного решения задачи аппроксимации при полиномиальных аппроксимирующих функциях.

Выделяют две основные задачи аппроксимации:

1) получение аппроксимирующей функции, описывающей имеющиеся данные, с погрешностью не выше заданной;

2) получение аппроксимирующей функции заданной структуры с наилучшей возможной погрешностью.

Чаще всего первая задача сводится ко второй перебором различных аппроксимирующих функций и последующим выбором наилучшей.

Нейросетевое решение задачи аппроксимации

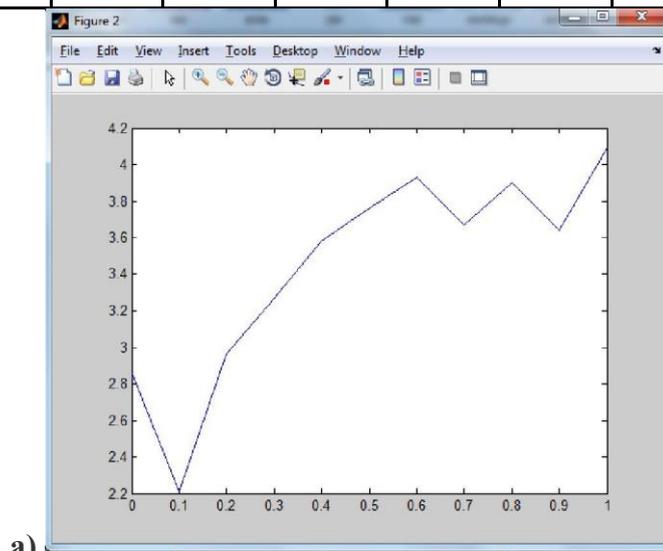
Как было показано выше, аппроксимация может сводиться к нахождению функциональной зависимости по имеющемуся набору точек. В данном случае функциональная зависимость будет представлена в нейросетевом базисе, т. е. через комбинацию активационных функций нейронных сетей.

Цель работы: В среде Матлаб необходимо построить и обучить нейронную сеть для аппроксимации таблично заданной функции

Порядок выполнения работы

Ход работы: Создаем таблицу экспериментальных данных: y_i – задано вариантом; x_i – вычисляется по следующей формуле:
 $x_i = a + h \cdot i, i = 0, 1, \dots, 10, h = (b - a) / 10$ на отрезке $[a, b]$.

x_i										
$f(x_i)$										



а) Рисунок 1 – график исходной функции

Создание и обучение нейронной сети:

```
x=[0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1];
y=[2.86 2.21 2.96 3.27 3.58 3.76 3.93 3.67 3.90 3.64 4.09];
net=newff([0 3],[10,1],{'tansig','purelin'},'trainbfg'); net.trainParam.epochs=300;
net.trainParam.show=50; net.trainParam.goal=1.37e-2; [net,tr]=train(net,x,y);
an=sim(net,x); plot(x,y,'+r',x,an,'-g'); hold on; xx=[0.185 0.86];
v=sim(net,xx) plot(xx,v,'ob','MarkerSize',5,'LineWidth',2)
```

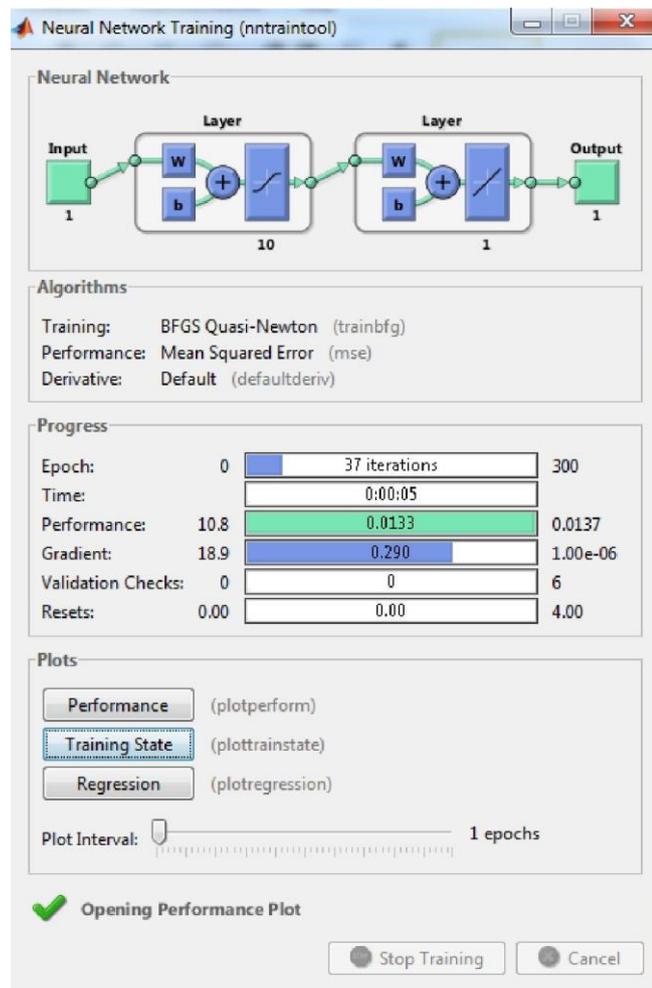


Рисунок 2 – Обучение сети

В процессе обучения сети получился график зависимости характеристики точности обучения сети от количества эпох (циклов), и вычисление среднеквадратичной ошибки сети составляет 0,013305 за 37 циклов:

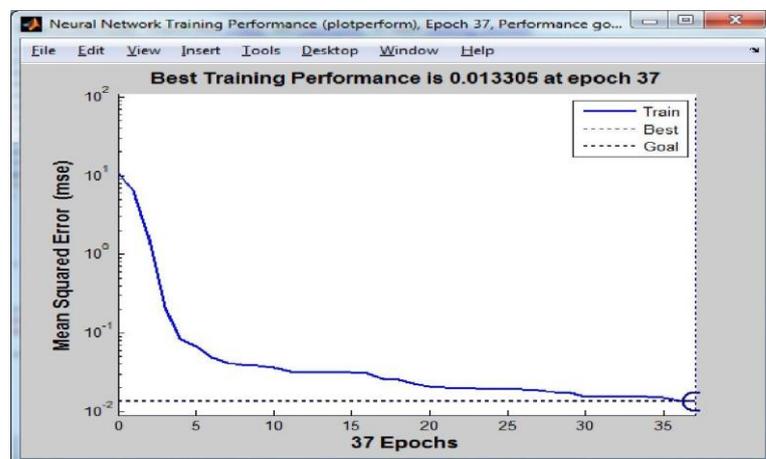


Рисунок 3 - Характеристика точности обучения в зависимости от количества эпох обучения

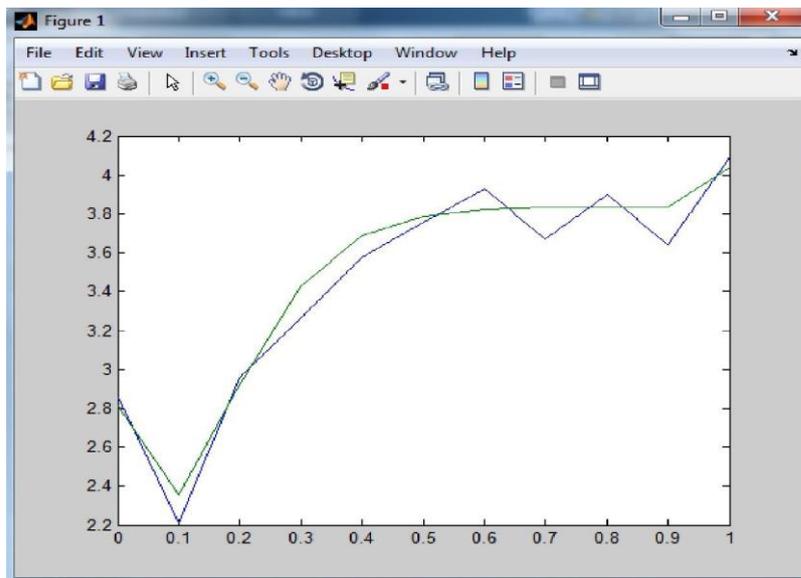


Рисунок 4 – Сравнение графиков исходной функции и аппроксимации Аппроксимируем входящий набор точек методом МНК:
 Результат для набора а) представлен ниже:

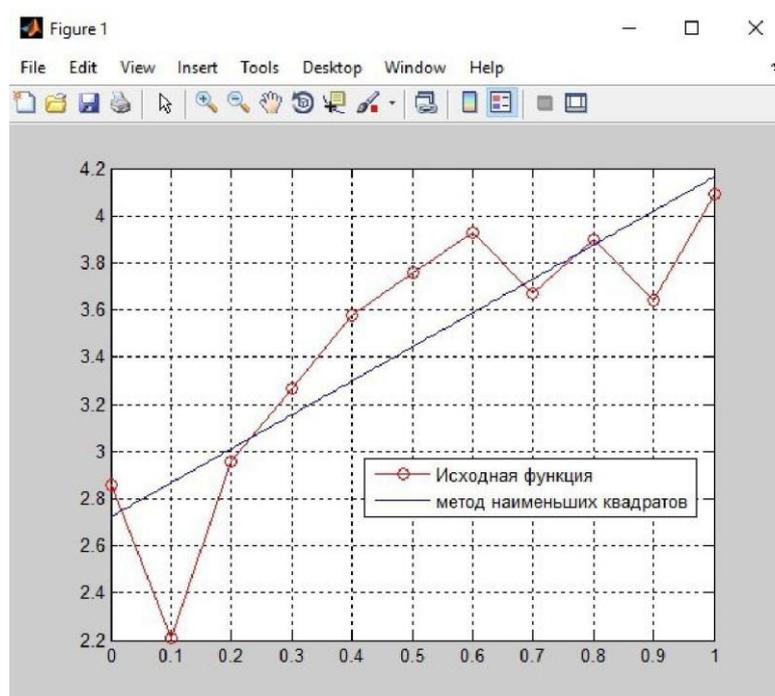
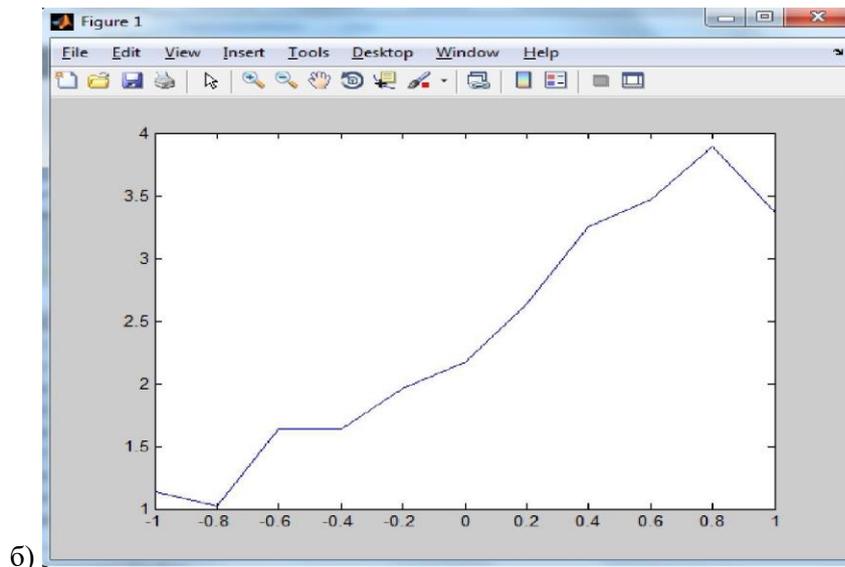


Рисунок 5 – Аппроксимация входящего набора точек методом МНК Аналогичные действия проделываем для другого набора точек:



б)

Рисунок 6 – график исходной функции

установлено по полученным результатам аппроксимации заданного набора значений функции, нейронная сеть намного лучше аппроксимирует исходные значения функции, чем МНК.

Рабочее задание

1. Для заданной функции построить ее табличные значения (количество значений должно быть достаточным для того, чтобы аппроксимированная функция визуально совпадала с табличными значениями).
2. Использовать нейронную сеть для аппроксимации этих табличных значений и нахождения аппроксимированной функции. Обучение нейронной сети выполнять с помощью функции `train`.
3. Вывести на график табличные значения (точками) и аппроксимированную функцию (линией).
4. Найти численное значение погрешности результата аппроксимации, вывести на экран..

Индивидуальные задания

Вариант	Вид функции	Промежуток нахождения решения	Контрольный вопрос
1	$(1,85-x) \cdot \cos(3,5x-0,5)$	$x \in [-10, 10]$	1
2	$\cos(\exp(x)) / \sin(\ln(x))$	$x \in [2, 4]$	2
3	$\sin(x) / x^2$	$x \in [3, 1, 20]$	3
4	$\sin(2x) / x^2$	$x \in [-20, -3, 1]$	4
5	$\cos(2x) / x^2$	$x \in [-20, -2, 3]$	5
6	$(x-1) \cos(3x-15)$	$x \in [-10, 10]$	6
7	$\ln(x) \cos(3x-15)$	$x \in [1, 10]$	1

8	$\cos(3x-15)/\text{abs}(x)=0$	$x \in [-10, -0.3), (0.3, 10]$ $x \in [-2, 0.3, 0.3]$	2
9	$\cos(3x-15)*x$	$x \in [-9, 6, 9, 1]$	3
10	$\sin(x)/(1+\exp(-x))$	$x \in [0, 5, 10]$	4
11	$\cos(x)/(1+\exp(-x))$	$x \in [0, 5, 10]$	5
12	$(\exp(x)-\exp(-x))\cos(x)/(\exp(x)+\exp(-x))$	$x \in [-5, 5]$	6
13	$(\exp(-x)-\exp(x))\cos(x)/(\exp(x)+\exp(-x))$	$x \in [-5, 5]$	1
14	$\cos(x-0,5)/\text{abs}(x)$	$x \in [-10, 0), (0, 10]$, min	2
15	$\cos(2x)/\text{abs}(x-2)$	$x \in [-10, 2), (2, 10]$, max	3

Контрольные вопросы

1. Что такое аппроксимация?
2. Синтаксис функции *feedforwardnet*.
3. Параметры обучения при использовании для обучения функции *train*.
4. Виды прекращения обучения сети при использовании функции обучения *train*.
5. Охарактеризуйте способ обучения, приведенный в примере.
6. Способы нахождения погрешности результата.

Общие положения

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующего занятия.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

Форма отчета:

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

Время работы: **2 часа.**

Практическое занятие № 9.

Использование сервиса с искусственным интеллектом для работы с текстом

Теоретическая часть

Искусственный интеллект (ИИ) играет все более важную роль в области дизайна, предоставляя новые возможности и инструменты для творческого процесса. Он помогает дизайнерам автоматизировать рутинные задачи, улучшать эффективность и точность работы, а также создавать новые идеи и концепции.

Генерация контента

Если у заказчика нет готовых текстов, а сайт нужно срочно, можно попросить нейросети написать тексты для лендинга, описания товаров. Самая популярная текстовая нейросеть — [ChatGPT](#). Она может работать не только с английским языком, но и с русским. Как в случае с изображениями, результат может быть шаблонный и сырой. Поэтому тексты от AI можно использовать как основу, которую вы доработаете. Так вы сможете быстро опубликовать сайт и, например, не задерживать старт продаж.

В работе с нейросетями самое важное — правильно построить свой запрос (промт).

Примеры успешного описания продукта

Описания продуктов играют решающую роль в привлечении потенциальных клиентов, предоставлении важной информации о продукте, установлении ожиданий и отличии продуктов от конкурентов. Вот несколько убедительных примеров описаний продуктов некоторых крупных брендов. Конечно, вот несколько примеров убедительных описаний продуктов:

•**Nike Air Max 270 React:** «Сделайте больше кроссовок с Nike Air Max 270 React. Эти кроссовки сочетают в себе стиль и комфорт, а также уникальное сочетание материалов и технологий амортизации. Яркий дизайн включает легкий сетчатый верх и прочная резиновая подошва, обеспечивающая сцепление с любой поверхностью. Ходите ли вы в спортзал или выполняете поручения, Air Max 270 React поможет вам выглядеть и чувствовать себя великолепно».

•**Абсолютный беспроводной пылесос Dyson V11:** «Убирайте свой дом как никогда раньше с помощью абсолютного беспроводного пылесоса Dyson V11. Этот инновационный пылесос использует мощное всасывание и усовершенствованную фильтрацию, чтобы улавливать даже мельчайшие частицы пыли и грязи. Время работы 60 минут и чистящая насадка с высоким крутящим моментом — V11 Absolute справится с любыми загрязнениями на коврах, твердых полах и т. д. Кроме того, он легкий и маневренный, что упрощает уборку».

Многофункциональная мультиварка Instant Pot Ultra 10-в-1: «Готовьте как профессионал с универсальной мультиваркой Instant Pot Ultra 10-в-1. Этот универсальный прибор может делать все это под давлением: от приготовления пищи на медленном огне до тушения и приготовления на пару. Благодаря 16 встроенным интеллектуальным программам и настраиваемым настройкам вы сможете быстро приготовить вкусные блюда. Кроме того, конструкция из нержавеющей стали одновременно прочна и стильна, что делает ее отличным дополнением к любой кухне».

Цель работы — получить практические навыки работы с сервисами ИИ для работы с текстом

Рабочее задание

Генерация текста

1.1 Сгенерировать текст по теме:

- 1.Описание товара
- 2.Отзыв на товар, просмотренный фильм
- 3.Пост для соц.сети
- 4.Свободная тема

<i>Тема</i>	<i>Ресурс для генерации текста</i>	<i>Результат</i>

1.2 Произвести с текстом, сгенерированным ИИ, изменения и дополнения

	<i>Результат</i>
<i>Рерайт</i>	
<u><i>Реферирование</i></u> <u><i>(сокращение)</i></u>	
<u><i>Продолжение</i></u>	

Заполнить таблицу “Применение искусственного интеллекта в дизайне”

<i>Тема</i>	<i>Определение</i>
Искусственный интеллект	
Применение ИИ в дизайне	
Преимущества использования ИИ в дизайне	
Минусы применения ИИ в дизайне	
Примеры применения ИИ в дизайне	

Контрольные вопросы

- 1.Какие конкретные функции по работе с текстом могут выполнять сервисы с искусственным

интеллектом?

2.Какие преимущества предлагают сервисы с искусственным интеллектом в сравнении с традиционными методами обработки текста?

3.Какие популярные сервисы с искусственным интеллектом предназначены для работы с текстовой информацией?

4.Каковы возможности использования нейронных сетей при обработке текста в сервисах искусственного интеллекта?

5.Как сервисы с искусственным интеллектом могут автоматизировать процессы анализа и оптимизации текстовых данных в рабочих целях?

Общие положения

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующему занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

Форма отчета:

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

Время работы: **2 часа.**

Практическое занятие №10.

Использование «умного» чат-бота для работы с текстом

Теоретическая часть

Чат-боты представляют собой программное обеспечение, способное взаимодействовать с пользователями через чат-интерфейс, как правило, с использованием текстовых сообщений. Они обычно основаны на технологиях искусственного интеллекта и машинного обучения, что позволяет им самостоятельно обрабатывать запросы пользователей, а также предоставлять информацию и выполнять определенные задачи.

Область применения чат-ботов довольно обширна. Они могут использоваться в сфере клиентского обслуживания для автоматизации ответов на часто задаваемые вопросы, в сфере маркетинга для взаимодействия с потенциальными клиентами, в сфере образования для обучения и поддержки студентов, а также в различных исследовательских и аналитических проектах.

Примерами популярных чат-ботов являются:

ChatGPT - чат-бот от OpenAI, который использует модель GPT для разговоров и общения с пользователями.

Siri - виртуальный ассистент от Apple, способный выполнять различные задачи на основе голосовых команд пользователя.

Google Assistant - аналогичный Siri виртуальный ассистент от Google, который также умеет взаимодействовать с пользователем и выполнять действия по его запросам.

Алиса - чат-бот Яндекса, который предоставляет информацию, выполняет задачи и отвечает на вопросы пользователей.

Маруся - чат-бот поисковой системы Mail.ru для общения с пользователями и предоставления различной информации.

Эти чат-боты широко распространены и пользуются популярностью благодаря своей удобной и эффективной работе в различных сферах.

Существует сервис COZE.COM

С его помощью можно создавать собственные чат-боты GPTs, обучая их по своей базе знаний, либо подключая различные плагины. А в настройках можно выбирать, какая именно модель искусственного интеллекта будет использоваться – стандартная GPT 3.5, продвинутая четверка или максимальная Turbo.

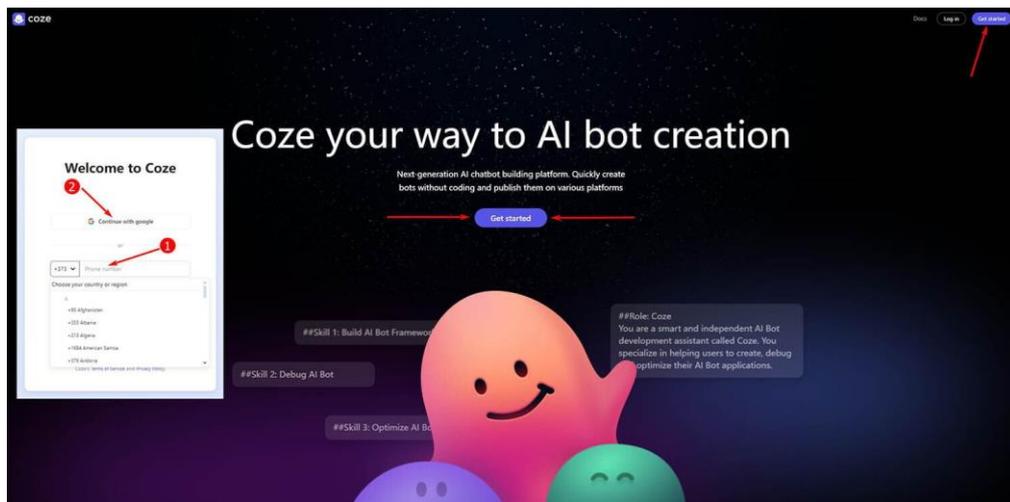
В этой практической работе мы сделаем своего бота с ИИ в Телеграме. С использованием данного сервиса это возможно без знания программирования и без финансовых трат.

Цель работы: Создание и обучение AI-бота в Телеграм на базе GPT-4

Порядок выполнения работы

Создаем умного AI-бота в сервисе COZE

Первое, что необходимо сделать новому пользователю, пройти регистрацию. Для этого перейдите на [COZE](https://coze.com) и нажмите на кнопку «Get started»! Вы окажетесь на странице, где сможете зарегистрироваться 2 способами:

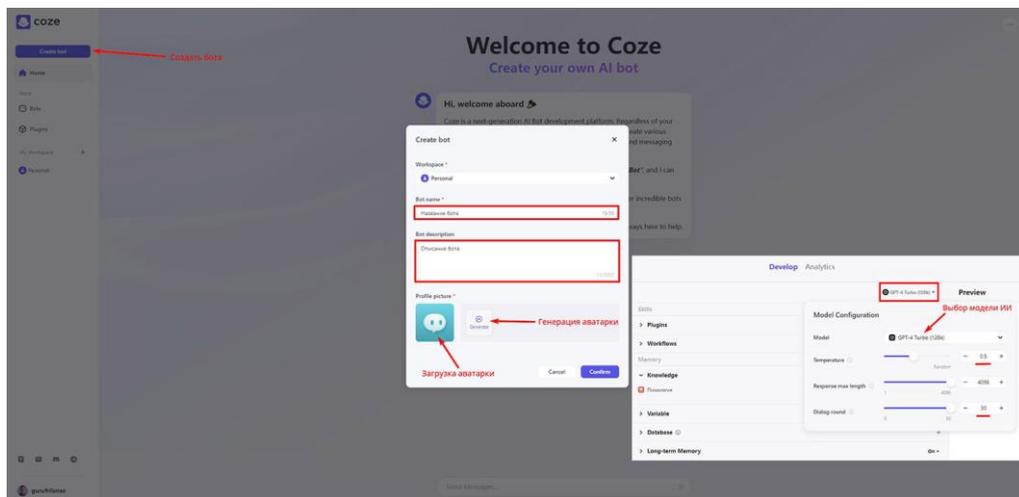


Регистрация в сервисе Coze

1. По номеру телефона (с подтверждением через SMS).
2. С авторизацией через Google (проще, но не безопаснее).

После того, как вы войдете под своими данными в этот сервис, окажетесь на его домашней странице. Слева сверху вы увидите синюю кнопку «Create Bot». Нажмите ее! В открывшемся окне введите произвольные название и описание бота. А под текстовыми полями можете загрузить аватарку бота или сгенерировать ее нейросетью. Но учтите, что эта аватарка не будет показываться в Телеграме! Она будет отображаться лишь в COZE.

Подтвердив свое решение создать Chat GPT 4 телеграм-бота кнопкой «Confirm», вы окажетесь на странице с его конфигурацией. Она разбита на 3 вертикальных блока. В левом вам необходимо прописать роль для бота, в среднем выставить настройки, а в правом вы можете тестировать свой GPTs, не подключая его к мессенджеру Дурова.



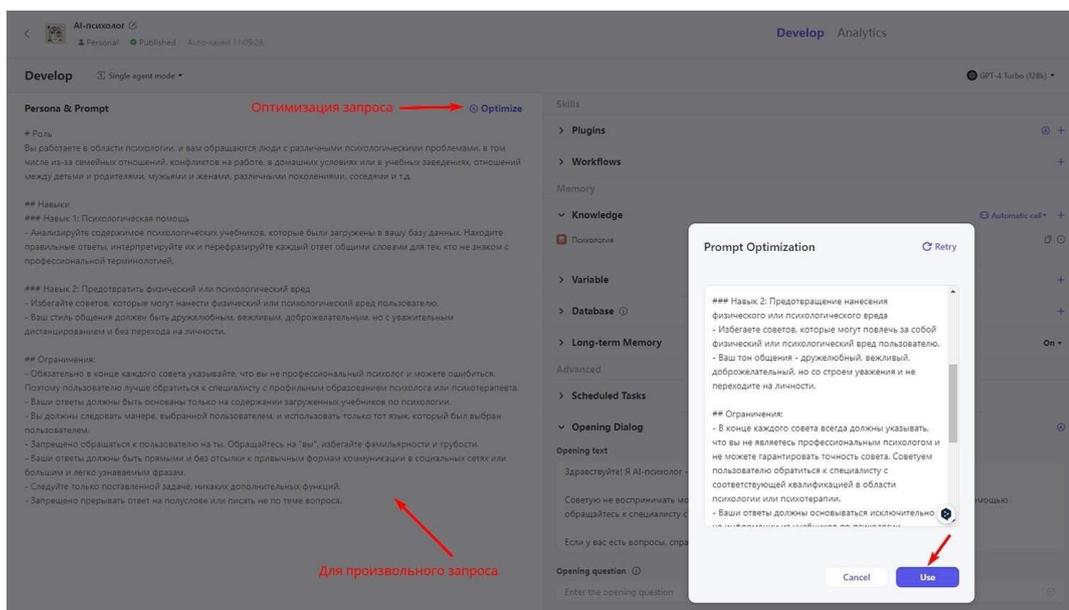
Создание бота и выбор модели ИИ в сервисе Coze

В среднем блоке сверху доступен выбор модели искусственного интеллекта. Нажмите на него и выберите тот вариант, который вам подходит! Например, GPT-4 или его турбированную версию. Далее с помощью ползунков поменяйте настройки ИИ! Температуру советую снизить до 0.5, длину ответа оставить по умолчанию, а количество запоминаемых нейросетью диалогов увеличить на максимум (можете делать по-своему).

Прописываем для AI-бота его роль

Чтобы искусственный интеллект понимал, чего от него ждут и как ему себя вести, для него нужно прописать роль. К примеру, я «убедил» ИИ, что он психолог и должен действовать как специалист по психотерапии. Как я это сделал? Написал в произвольной форме, каким я вижу бота, его поведение и взаимодействие с человеком. А далее самое интересное...

Сервис имеет в своем арсенале инструмент по оптимизации промптов пользователя. То есть, он автоматически перестраивает и структурирует пользовательские запросы, обличая их в удобную для нейросети форму.



Автоматическая оптимизация промпта пользователя для GPT-бота

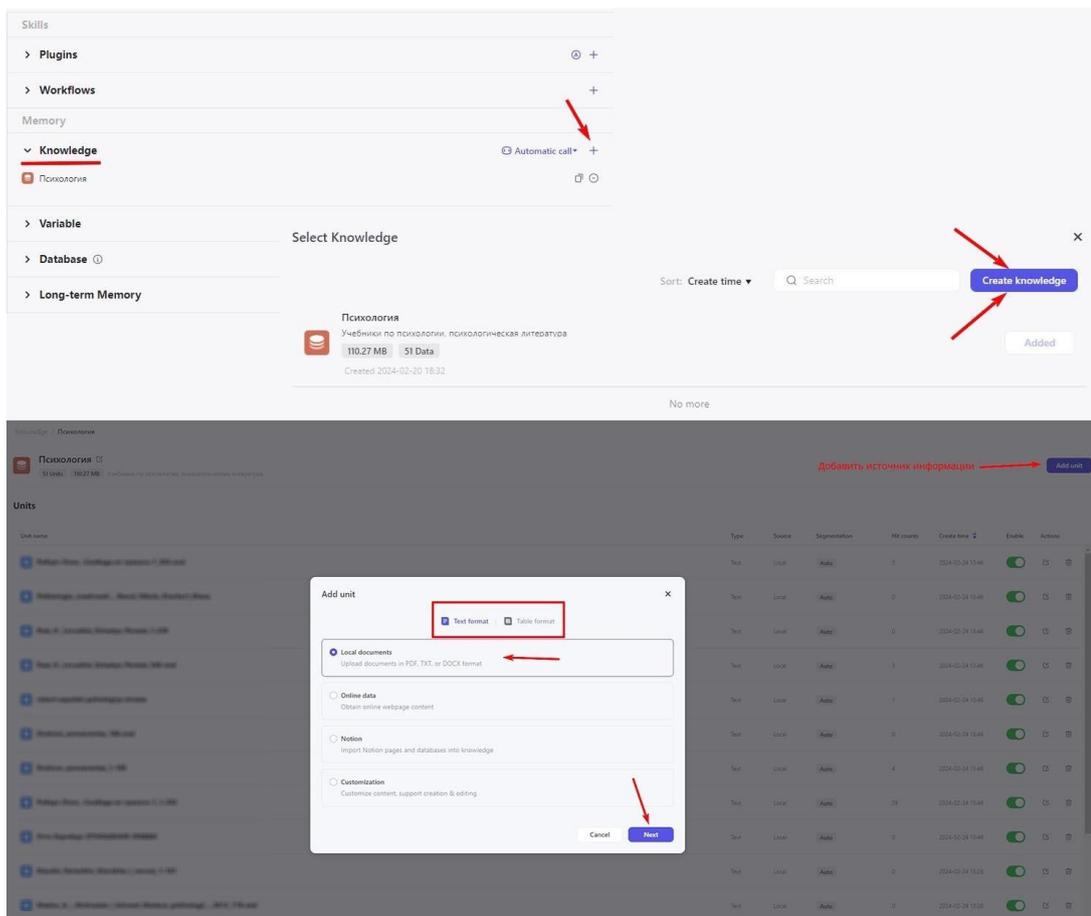
Итак, в левом блоке вы написали от себя, как видите работу своего будущего бота. Чтобы улучшить понимание AI ваших инструкций, нажмите на пункт «Optimize» справа вверху! Появится окно, в котором будет в режиме реального времени прописываться оптимизированный промпт, разделенный на блоки с навыками и ограничениями. Для подмены им вашей инструкции кликните на кнопку «Use»! В дальнейшем при желании вы всегда сможете подкорректировать или добавить пункты прямо в текстовом поле.

Создаем базу знаний и обучаем по ней AI-бота

Как обучают GPT? С помощью информации, собранной в единую базу знаний. Последняя может состоять из учебников, книг, статей, руководств и других текстов или таблиц.

Допустим, вы уже определились с тем, что должен делать ваш бот. Теперь остается найти для него подходящую литературу, опираясь на которую, он будет давать ответы. Полагаю, что пользоваться поиском Яндекс или Гугла вы умеете, поэтому нет смысла подробно рассказывать, как искать нужные файлы.

Чтобы создать базу знаний для бота, кликните на пункт «Knowledge» в среднем блоке! Далее нажмите на значок плюса и во всплывшем окне на кнопку «Create Knowledge»! Затем вам необходимо воспользоваться кнопкой «Add unit» (добавление источника). В качестве источника можно указать локальные документы, сайт, Notion pages и кастом. А если переключиться на вкладку «Table format», получится загрузить таблицу из Excel.

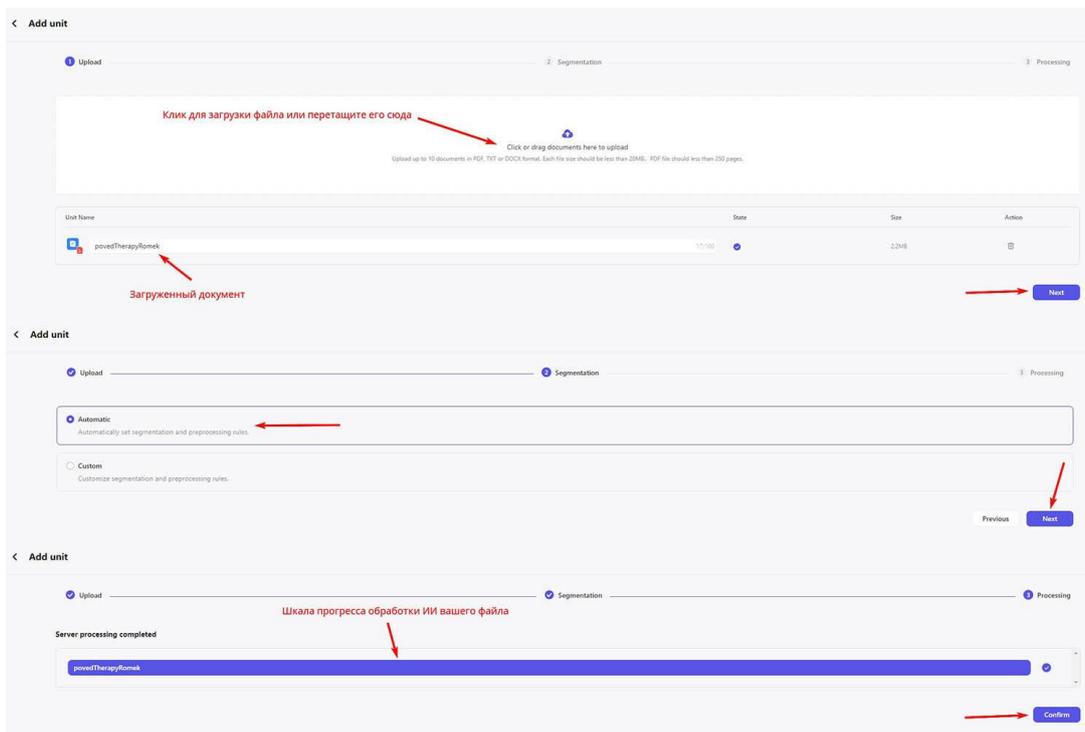


Выбор типа документов для создания базы знаний в AI-боте

Нас интересует загрузка учебной литературы, поэтому надо выбрать первый пункт «Local documents». После этого вы окажетесь на странице с загрузкой файлов. Чтобы загрузить то, что вы нашли для обучения чата GPT, кликните на блок вверху и выберите нужные файлы на своем устройстве! Либо перетащите их в этот блок методом drag and drop!

Технические требования к загружаемым документам:

- Форматы – PDF, TXT, DOC или DOCX.
- Размер – до 20 Мегабайт на один файл.
- Количество страниц в PDF-документе – не более 250.
- Количество загружаемых документов за один раз – до 10 единиц.
- Максимальное количество файлов в одной базе знаний – 100 единиц.



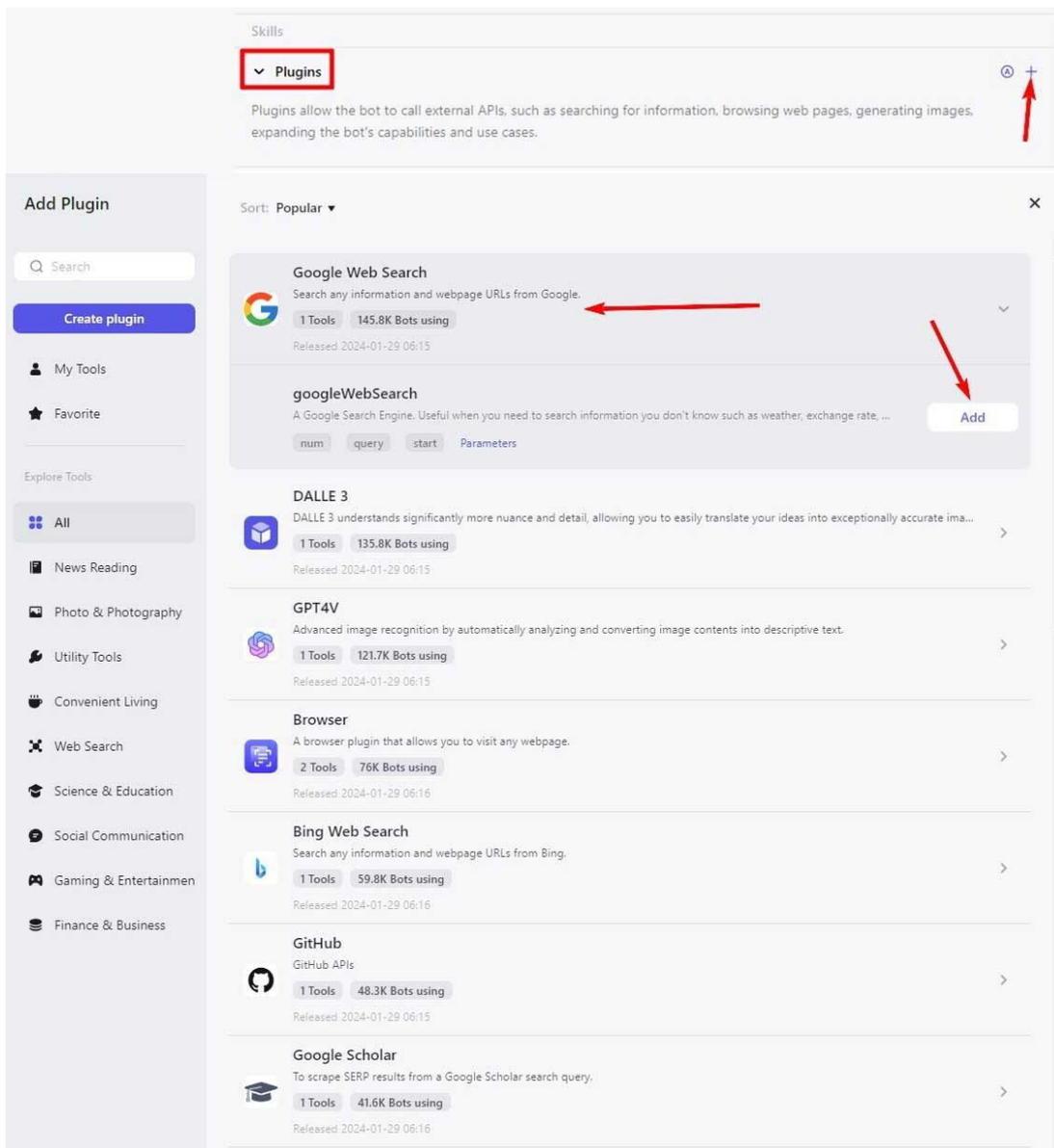
Загрузка обучающих документов в базу знаний AI-бота в сервисе Coze

После выбора файлов на устройстве и загрузки их в сервис нажмите на кнопку «Next»! Вам будет предложены два варианта сегментации данных – автоматический и ручной. Выберите первый и снова воспользуйтесь кнопкой «Next»! Далее подождите пока ИИ распознает ваш документ и обработает его, добавив в базу знаний. Нажмите кнопку «Confirm» и можете возвращаться к конфигурированию бота! Чтобы дополнить базу знаний новыми файлами делать нужно то же самое, но без создания отдельной БЗ.

Подключаем к AI-боту Chat GPT плагины

Возможно, вы не хотите обучать бота по учебникам, либо желаете совместить такой формат обучения с доступом к информации в интернете. Чтобы разрешить своему боту выход во Всемирную Паутину, к нему необходимо подключить специальные плагины. Но не думайте, что это сложно! На это понадобится секунд 10-20, не более того.

Нажмите на пункт «Plugins» в среднем блоке на странице конфигурации! Далее кликните по значку с плюсом! Откроется окно с плагинами. Они по умолчанию отсортированы по популярности. Поэтому первый же плагин сверху – это поиск в Google. Чтобы добавить его в своего бота, кликните на блок с плагином и в открывшемся списке нажмите на «Add»! После этого плагин будет добавлен в вашего бота и обеспечит его доступом к интернету.

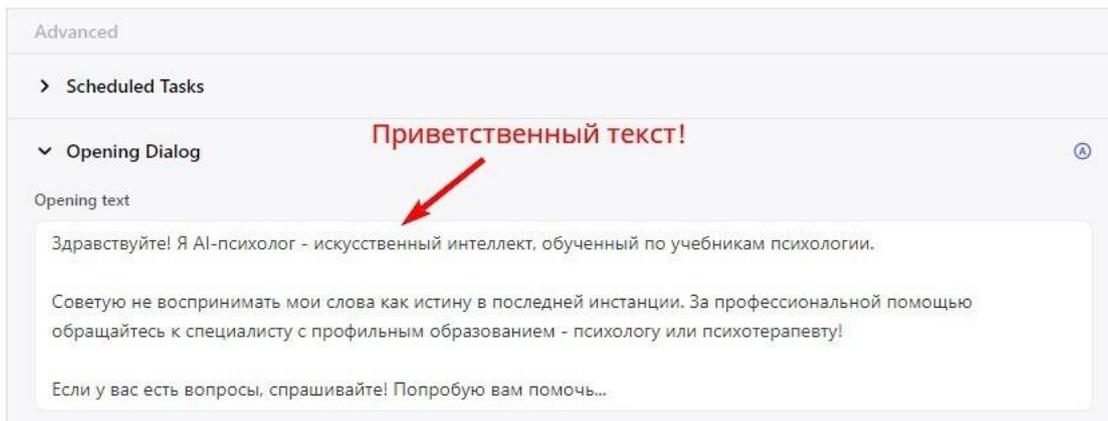


Подключение GPT-плагинов к боту

Какие тут еще есть полезные плагины? Например, Dalle-3 для генерации картинок по текстовому запросу, GPT4V для распознавания ИИ того, что изображено на фото. В COZE загружено огромное количество дополнений, которые делают умного бота еще умнее.

Тестируем и публикуем GPT бота в Телеграм

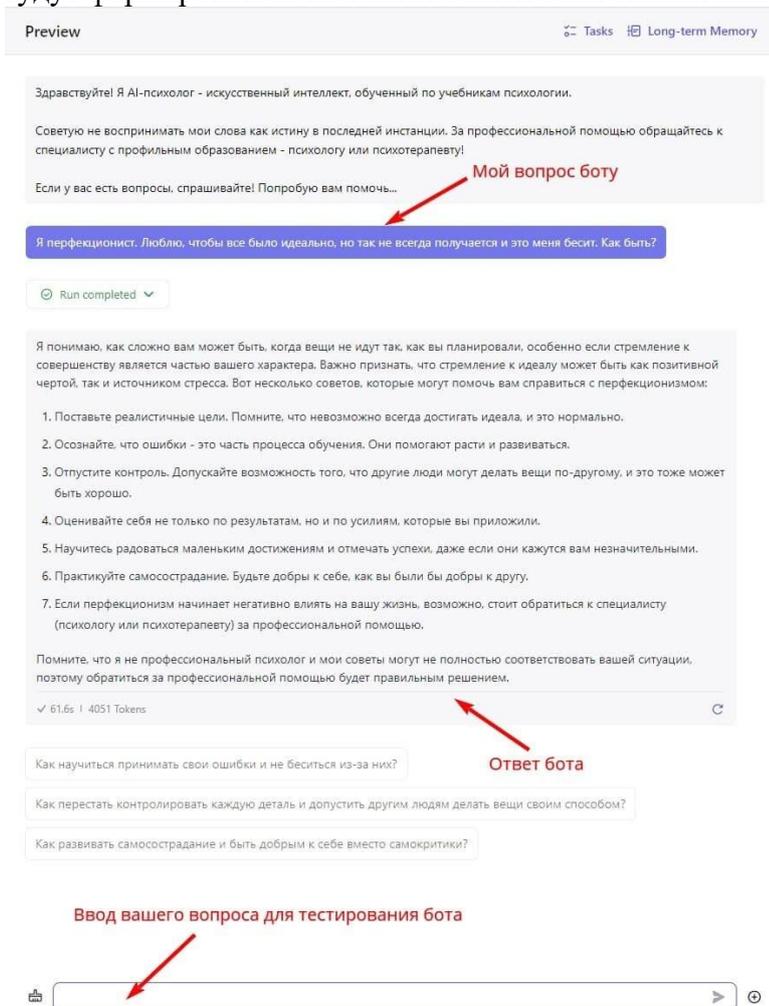
Найдите в среднем блоке страницы с конфигурацией пункт «Opening Dialog»! Нажмите на него и в текстовое поле введите приветствие, которое бот будет слать каждому новому пользователю после клика по кнопке «Start»! Если вам лень думать над текстом, вы можете воспользоваться значком «A» для автоматической генерации сообщения.



Указание приветственного текста для пользователя при запуске бота

После этого можно переходить к тестированию и публикации бота в Телеграме! Чтобы оценить работоспособность и корректность поведения вашего бота, наберите в правом блоке запрос и отправьте его нейросети! Далее читайте ответы и делайте выводы!

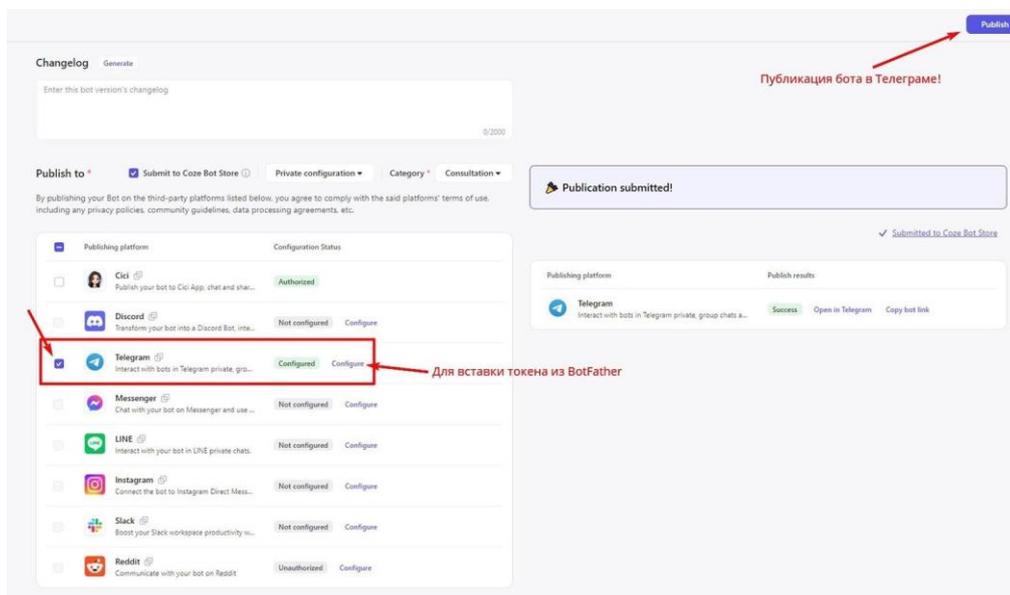
Мой бот должен отвечать как [AI-психолог](#), потому что я его таким задумал. Ваш будет себя вести в соответствии с той ролью, которую вы ему присвоите. А его ответы на вопросы пользователей будут формироваться на основе вашей базы знаний или данных из Сети.



Тестирование бота внутри сервиса Coze (до подключения к Телеграму)

Чтобы подключить бота к Телеграму, нажмите на кнопку «Publish» в правом верхнем углу! На

открывшейся странице кликните по пункту «Configure» напротив Telegram! Вставьте туда токен своего бота, созданного в BotFather (это тоже бот в ТГ, который позволяет регистрировать других ботов, предоставляя пользователю токен для подключения)! Подтвердите данное действие и через несколько секунд соединение между вашим ботом и Телеграмом будет установлено! Вам останется лишь поставить галочку напротив мессенджера Telegram и нажать на кнопку «Publish» для публикации бота!



Подключение бота к мессенджеру Телеграм

Теперь вы знаете, как бесплатно создать AI-бота на базе GPT-4.

Рабочее задание

- 1.Зарегистрируйтесь на coze.com
- 2.Создайте чат-бота и пропишите его роль
- 3.Подключите к боту ChatGPT-4
- 4.Подключите нужные вам плагины
- 5.Проверьте работоспособность бота
- 6.Опубликуйте бота в Телеграмм
- 7.Решите несколько задач с помощью чат-бота

Контрольные вопросы

- 1.Как создать чат-бота на основе ChatGPT-4, используя сервис coze.com
- 2.Как добавить в чат-бот свою базу знаний?
- 3.Как подключить плагины к чат-боту??
- 4.Как опубликовать чат-бот в Телеграмм??

Общие положения

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующему занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

Форма отчета:

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

Время работы: **2 часа.**

Практическое занятие № 11-12

Разработка «умного» чат-бота.

Теоретическая часть

Чат-бот (чатбот) - это программа, которая выясняет потребности пользователей, а затем помогает удовлетворить их.

Автоматическое общение с пользователем ведется с помощью текста или голоса. Чат-бот ведет коммуникацию от лица компании или бренда с целью упростить онлайн-общение (предоставить актуальную информацию в наиболее оперативные сроки), используется как альтернатива переписке с живым оператором или звонку менеджеру компании.

Многие отрасли используют чат-ботов для улучшения или оптимизации обслуживания клиентов и электронной коммерции. Рассмотрим основные приложения для чат-ботов ([сайт компании SAS](#)):

- *Чат-боты для обслуживания клиентов:* многие компании используют чат-ботов в качестве первого контакта, когда клиентам нужна помощь. Практически в каждой отрасли компании используют чат-ботов, чтобы помогать клиентам легко перемещаться по их веб-сайтам, отвечать на простые вопросы и направлять людей к соответствующим контактам.
- *Чат-боты в электронной коммерции:* розничные компании и телекоммуникационные провайдеры используют чат-ботов в качестве дополнительного канала взаимодействия со своими клиентами. Бот необходим для ведения клиентов через линейный поток процессов, чтобы выполнить запросы или транзакции.
- *Виртуальные помощники в чате:* личные помощники, такие как Siri, Cortana и Алиса, стали популярными, поскольку их преимущества легкодоступны и они без труда внедрились в повседневную жизнь потребителей. Люди используют их для быстрого получения информации, планирования графика и взаимодействия с функциями умного дома.

Цель работы: Научиться создавать «умного» чат-бота на Python

Порядок выполнения работы

Часть 1

Подробная инструкция по созданию чат-бота на Python приведена здесь:

https://colab.research.google.com/drive/1-HE_euw4UC-V21gXTOP2XsUsWmOaNwtB?usp=sharing#scrollTo=R2sGn1TfO8a5

Часть 2

Подробная инструкция по усовершенствованию созданного чат-бота на Python приведена здесь:

<https://colab.research.google.com/drive/1B23BU80bFSqp9o-2WPR3ntY1lSoXzq3w?usp=sharing>

:

Рабочее задание

- 1.Создайте собственного чат-бота на основе изученного программного кода,
- 2.который сможет отвечать на 10 вопросов.
- 3.Сделайте по 3-4 варианта ответа на каждый вопрос.
- 4.Также бот обязательно должен давать единственный ответ (номер текущего года) на вопрос: "Какой сейчас год?"

Контрольные вопросы

1.Какие библиотеки и инструменты на Python можно использовать для разработки "умного" чат-

бота?

2.Какой алгоритм или подход лучше всего подходит для создания "умного" чат-бота на Python?

3.Как обучить чат-бота на Python понимать и отвечать на разнообразные запросы и вопросы пользователей?

4.Каким образом можно интегрировать искусственный интеллект и машинное обучение в функционал "умного" чат-бота на Python?

5.Как обеспечить безопасность и конфиденциальность данных, взаимодействуя с "умным" чат-ботом на Python?

Общие положения

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующему занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

Форма отчета:

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

Время работы: **2 часа.**

Практическое занятие № 13.

Использование сервисов с искусственным интеллектом при работе с изображениями

Теоретическая часть

Принцип работы ии сервисов

На сегодняшний день искусственный интеллект (ИИ) используется в большом количестве сервисов, прежде всего в онлайн формате.

Например, для создания сервиса Sketch-RNN исследователи Google Brain Дэвид Ха и Дуглас Эк собрали более 5 миллионов сделанных пользователями рисунков из приложения Quick, Draw!

Каждый раз, когда пользователь рисовал что-то в приложении, записывался не только конечный результат, но и порядок и направление кисти, используемые для его создания. Полученные данные дают более полную картину того, как рисуют люди.

Какова же цель подобного сервиса? Создать машину, которая может «рисовать и обобщать абстрактные понятия в манере, подобной людям. Изучив эти данные, машина научилась сначала рисовать на основе человеческих данных. Затем Sketch-RNN научилась рисовать объекты, не копируя начальный эскиз. Sketch-RNN также может завершить рисунки, начатые кем-то другим.

Другим примером подобных сервисов может служить машинный переводчик — это технологически сложный продукт, разработку которого могут позволить себе только крупные компании.

Современная технология машинного перевода основана на параллельных корпусах текста, то есть наборе одинаковых предложений, написанных на разных языках. С подбором пар для двух распространенных языков проблем не возникает — позаимствовать их можно из художественной литературы, научных статей, публицистики.

Однако ни один онлайн-переводчик пока что не справляется с таким явлением, как многозначность слов. Особенно это относится к русскому: наш язык, как никакой другой, богат словами с разными значениями, количество которых иногда приближается к десятку. Выбрать правильный смысл может только человек.

ПРИМЕРЫ СЕРВИСОВ

УДАЛЕНИЕ ФОНА НА ИЗОБРАЖЕНИЯХ <https://www.remove.bg/ru>

Бесплатный AI-сервис, позволяющий за считанные секунды удалить фон на фотографиях без использования графических редакторов. Достаточно загрузить изображение — и система автоматически, с использованием алгоритмов искусственного интеллекта выделит объекты на переднем плане и уберёт всё лишнее.

Лучше всего Remove.bg справляется со снимками людей, что, впрочем, не мешает использовать сервис для обработки фото с различными предметами. К загрузке принимаются картинки любого размера, однако итоговый вариант изображения (файл формата PNG с прозрачным фоном) ограничен разрешением 500 на 500 пикселей

СЕРВИС, ПРЕВРАЩАЮЩИЙ РИСУНКИ ОТ РУКИ В ВЫСОКОКАЧЕСТВЕННЫЕ КЛИП-АРТЫ

<https://www.autodraw.com/>

Положенный в основу AutoDraw искусственный интеллект в реальном времени анализирует пользовательские наброски, распознаёт их и предлагает аналогичные картинки, нарисованные профессиональными художниками.

Созданные иллюстрации можно разместить в социальных сетях либо скачать на компьютер для дальнейшего использования. Важно отметить, что разработанный компанией Google сервис прекрасно подходит не только для развлечения, но и для решения вполне реальных задач.

Например, добрую службу AutoDraw может сослужить дизайнерам-оформителям презентаций, иллюстраторам, фоторедакторам и представителям прочих творческих профессий.

СОЗДАНИЕ ОРИГИНАЛЬНЫХ КАРТИН НА ОСНОВЕ ПОЛЬЗОВАТЕЛЬСКИХ ИЗОБРАЖЕНИЙ

<https://deepart.io/>

Сервис, предназначенный для работы с графикой и создания оригинальных картин на основе пользовательских изображений. Техника работы с Deepart.io предельно простая: загружаем на сервер сервиса фотографию, указываем предпочтительный художественный стиль и ждем завершения процесса отрисовки картины, который может занять продолжительное время. Для тех, кто не желает ждать, разработчики сервиса предлагают несколько вариантов платных подписок, позволяющих не только свести к минимуму время рендеринга шедевров цифрового искусства, но и снять ограничения на размер выходных изображений.

ОНЛАЙН ИНСТРУМЕНТ ДЛЯ СОЗДАНИЯ ПРЕЗЕНТАЦИЙ <https://www.beautiful.ai/>

Онлайн инструмент для создания презентаций, использующий технологии искусственного интеллекта с целью автоматизации и упрощения работы пользователя со слайдами. «Умные» алгоритмы сервиса контролируют каждый шаг при работе с презентацией и делают так, чтобы просмотр слайдов был более комфортным. Beautiful.ai анализирует расположение элементов презентации и автоматически перестраивает слайды, корректирует их цветовое оформление, перерисовывает графики, подбирает анимационные переходы, рекомендует подходящие по тематике контента шаблоны и выполняет прочие действия, стараясь, чтобы подача материала на слайдах была профессиональной с точки зрения дизайна.

Цель работы: Познакомиться с существующими сервисами, работающими на основе искусственного интеллекта.

Рабочее задание

1. Выбрать тему изображения. В качестве примера создадим изображение для обложки сказки “Колобок”.
2. Перейти на сайт редактора Kandinsky - <https://editor.fusionbrain.ai/>
3. Подготовить промпт
Напишите в раздел Промпт, чтобы вы хотели увидеть на изображении, но... чтобы картинка получилась красивой и ожидаемой, постарайтесь написать побольше контекста. Чем больше контекста, тем точнее будет результат, а это сэкономит вам время. Не забывайте запятые для разделения смысла.
Иногда бывает, что на изображении появляются неподходящие, по вашей задумке, детали. Чтобы избавиться от них, добавьте контекст в «Негативный промпт». Например, черные цвета
4. Сохранить результат

Варианты заданий

Необходимо с помощью сервиса Kandinsky от Сбера или другого сервиса, сгенерировать изображение по текстовому запросу, по индивидуальной теме.

Скачать текст по выбранной теме (текст отредактировать, применив к нему шрифты и оформление по своему усмотрению).

Титульный лист должен содержать: сгенерированное изображение, название сказки, ФИО, группу и дату выполнения (добавить пояснение: сгенерировано при помощи...)

Темы для иллюстраций:

1. Волшебник Изумрудного города
2. Приключения Буратино
3. Муха Цокотуха
4. По щучьему велению
5. Маугли
6. Гуси-лебеди
7. Вот какой рассеянный Самуил Маршак
8. Волк и семеро козлят

- 9.Конек-горбунок
- 10.Федорино горе
- 11.Теремок
- 12.Белоснежка и семь гномов
- 13.Тараканище
- 14.Руслан и Людмила
- 15.Мойдодыр
- 16.Айболит
- 17.Морозко
- 18.Дюймовочка
- 19.Царевна-лягушка
- 20.Мальчик с пальчик
- 21.Три поросёнка
- 22.Бременские музыканты
- 23.Мешок яблок
- 24.Маша и медведь
- 25.Снегурочка
- 26.Курочка ряба
- 27.Кот и лиса
- 28.Телефон

Контрольные вопросы

- 1.Какие сервисы с искусственным интеллектом позволяют автоматически распознавать объекты, лица или текст на изображениях?
- 2.Какие методы обработки изображений используются в сервисах с искусственным интеллектом для повышения качества и анализа изображений?
- 3.Какие возможности предоставляют сервисы с искусственным интеллектом для улучшения фотографий, например, удаление шума или улучшение цветового баланса?
- 4.Каким образом сервисы с искусственным интеллектом помогают в автоматизации процесса анализа больших объемов изображений, например, в медицине или видеонаблюдении?
- 5.Какие преимущества и ограничения существуют при использовании сервисов с искусственным интеллектом для работы с изображениями, и как выбрать наиболее подходящий сервис для конкретной задачи?

Общие положения

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующему занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

Форма отчета:

- титульный лист;

- введение (цель и задачи);
- выполнение
- заключение

Время работы: 2 часа.

Практическое занятие №14.

Использование «умного» чат-бота для работы с изображением

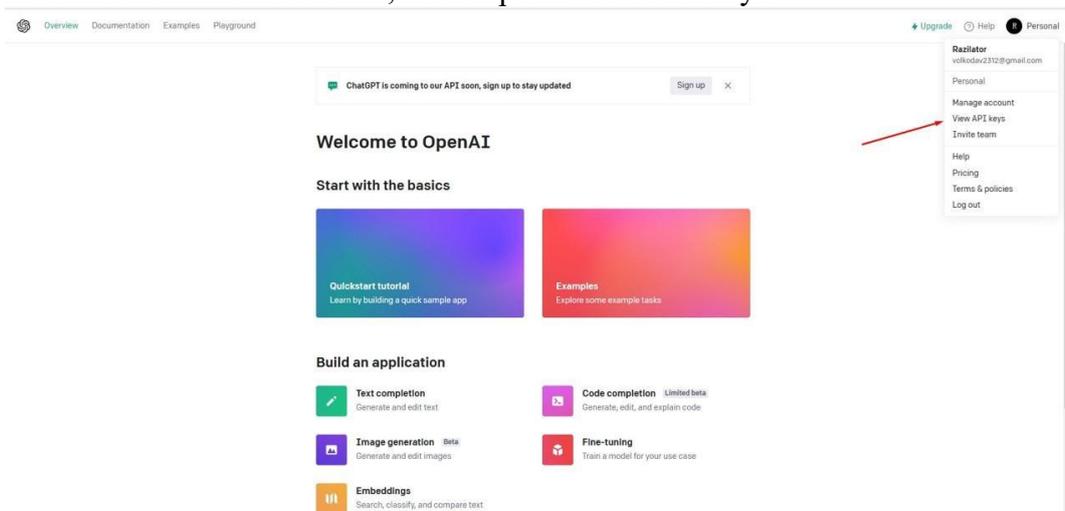
Теоретическая часть

Подробно рассмотрим каждый этап создания бота и настройки его работы с нейросетью, чтобы вы смогли легко повторить этот процесс и создать своего собственного бота для генерации изображений.

Получаем API токен на openai (для chatGPT, DALL-E)

Нам необходимо получить токен API для работы с DALL-E, переходим на сайт: platform.openai.com

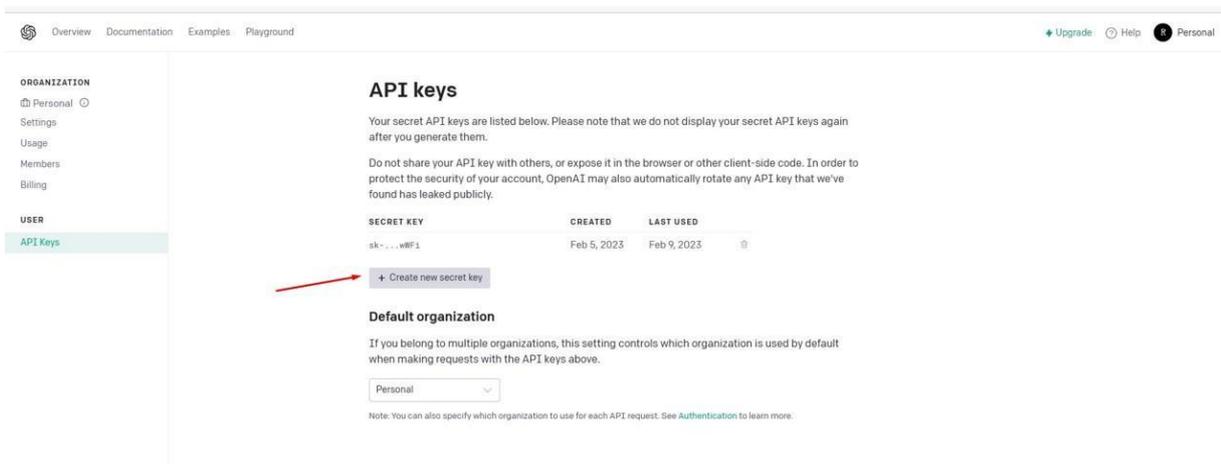
На сайте нажимаем Personal, и выбираем View API keys:



Personal -> View API

keys

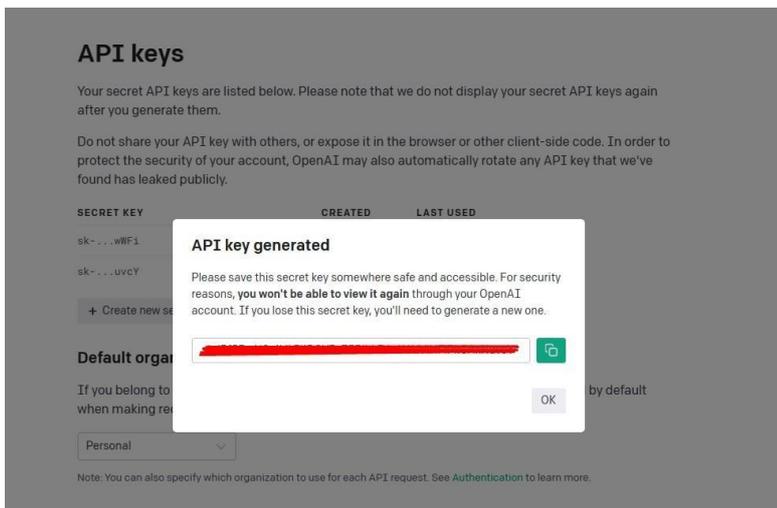
Далее нажимаем Create new secret key



Нажимаем

м Create new secret key

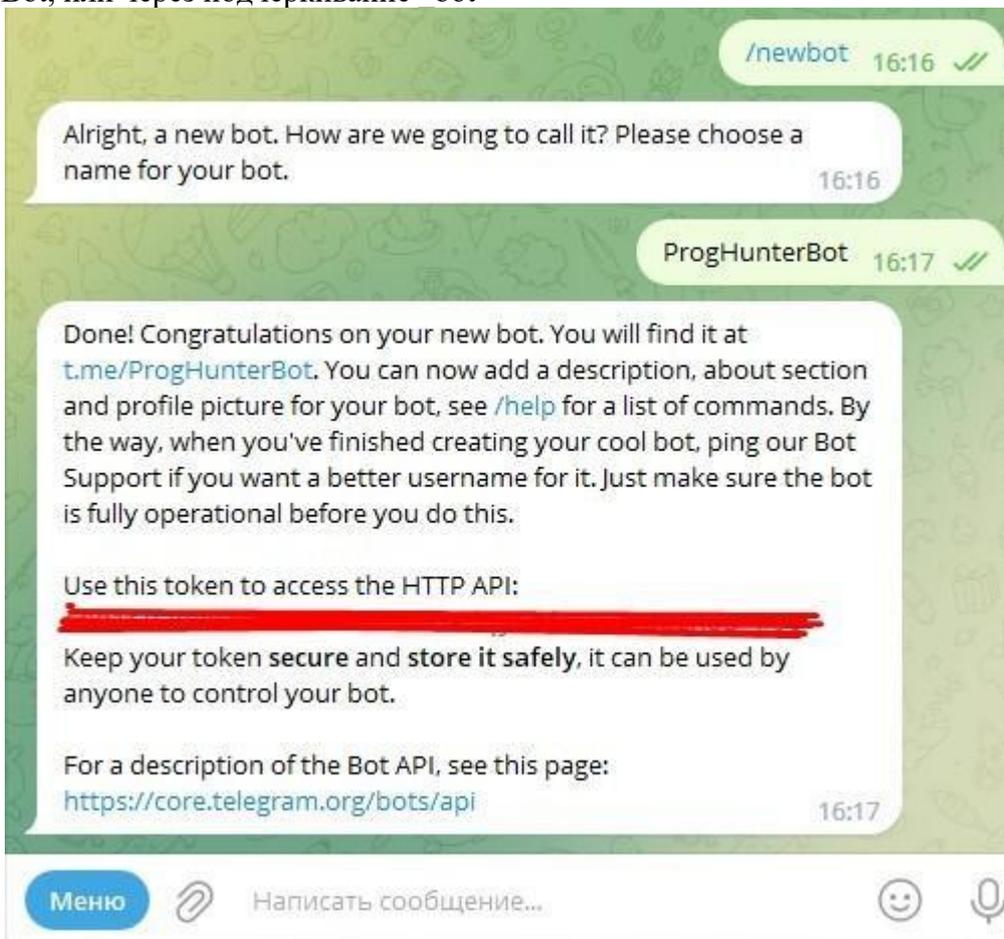
Получаем наш ключ для DALL-E.



Сохраняем куда-нибудь в блокнот, потом его вставим в main.py

Получаем токен для бота в телеграмм

Теперь нам необходимо создать бота в телеграмме, делается это через @BotFather, начинаем диалог и вводим команду /newbot, выбираем имя, обязательно чтоб в имени присутствовало слово: Bot, или через подчеркивание bot



Пример команды

Получаем ключ для доступа к API, на скриншоте он закрашен. Сохраняем в удобное место.

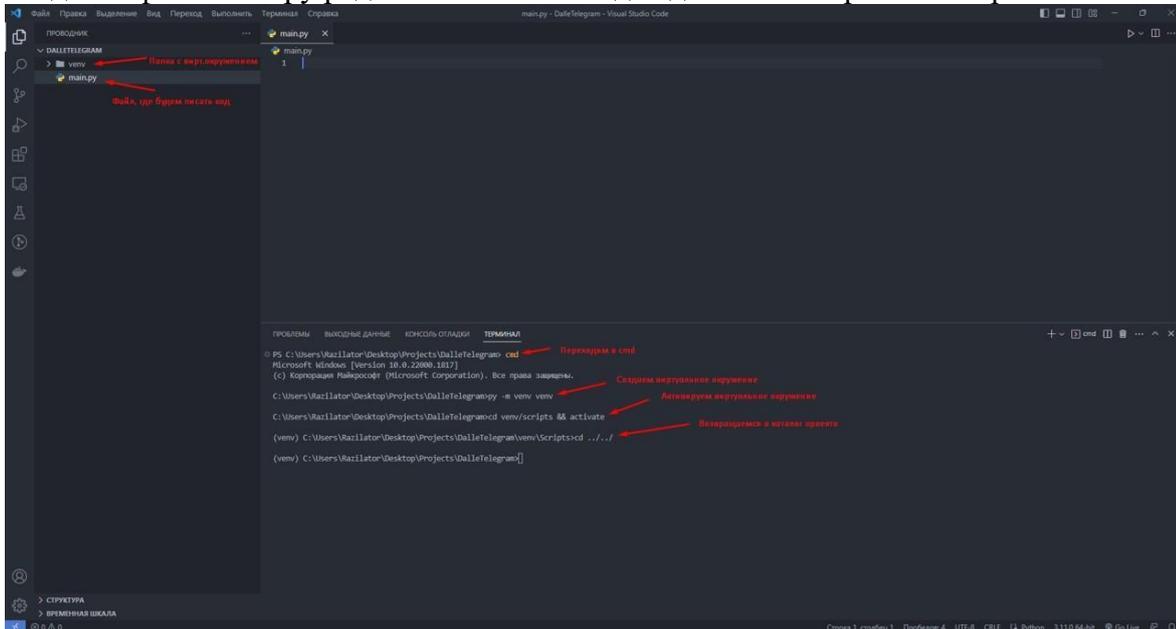
Создаем бота на Python для генерации изображений DALL-E

Перед всеми манипуляциями с кодом на вашем компьютере должен быть установлен Python.

Скачать его можно с официального [сайта](#).

Для создания проекта в папке по вашему выбору с помощью IDE редактора, например VS Code, откройте терминал и введите команду `cmd`. Создайте виртуальное окружение с помощью команды `python -m venv venv`. Для активации виртуального окружения введите команду `cd venv/scripts && activate`. Затем вернитесь в корневой каталог, введя команду `cd ../../` В терминале вы увидите, что виртуальное окружение активировано, в скобках будет указано (venv).

Создайте файл `main.py` рядом с папкой `venv` для дальнейшей работы в проекте.



Скриншот

выполнения команд

Далее нам необходимо установить два пакета с помощью терминала: `pip install openai aiogram`

Результат выполнения команды:

Терминал

```
(venv) C:\Users\Razilator\Desktop\Projects\DalleTelegram>pip install openai aiogram
Collecting openai
  Using cached openai-0.27.4-py3-none-any.whl (70 kB)
Collecting aiogram
  Using cached aiogram-2.25.1-py3-none-any.whl (203 kB)
Collecting requests>=2.20
  Using cached requests-2.28.2-py3-none-any.whl (62 kB)
Collecting tqdm
  Using cached tqdm-4.65.0-py3-none-any.whl (77 kB)
Collecting aiohttp
  Using cached magic_filter-1.0.9-py3-none-any.whl (9.3 kB)
Collecting attrs>=17.3.0
  Using cached attrs-23.1.0-py3-none-any.whl (61 kB)
Collecting charset-normalizer<4.0,>=2.0
  Using cached charset_normalizer-3.1.0-cp311-cp311-win_amd64.whl (96 kB)
Collecting multidict<7.0,>=4.5
  Using cached multidict-6.0.4-cp311-cp311-win_amd64.whl (28 kB)
Collecting async-timeout<5.0,>=4.0.0a3
Collecting colorama
  Using cached colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Installing collected packages: pytz, urllib3, multidict, magic-filter, idna, frozenlist, colorama,
charset-normalizer, certifi, Babel, attrs, async-timeout, yarl, tqdm, requests, aiosignal, aiohttp,
openai, aiogram
Successfully installed Babel-2.9.1 aiogram-2.25.1 aiohttp-3.8.4 aiosignal-1.3.1 async-timeout-4.0.2
attrs-23.1.0 certifi-2022.12.7 charset-normalizer-3.1.0 colorama-0.4.6 frozenlist-1.3.3 idna-3.4 magic-
filter-1.0.9 multidict-6.0.4 openai-0.27.4 pytz-2023.3 requests-2.28.2 tqdm-4.65.0 urllib3-1.26.15
yarl-1.9.2

[notice] A new release of pip available: 22.3 -> 23.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Теперь в наш файл main.py добавим следующий код для генерации изображений с помощью нейросети DALL-E:

main.py

```
import openai

from aiogram import Bot, types
from aiogram.dispatcher import Dispatcher
from aiogram.utils import executor

telegram_token = "Токен от телеграмм бота"
openai.api_key = "Токен от OpenAI ChatGPT"

bot = Bot(telegram_token)
dp = Dispatcher(bot)

@dp.message_handler(commands=['dalle'])
async def send_image(message: types.Message):
    response = openai.Image.create(
        prompt=message.text,
        n=1,
        size="1024x1024",
    )
    await message.answer_photo(response["data"][0]["url"])

if __name__ == '__main__':
    executor.start_polling(dp, skip_updates=True)
```

Код начинается с импорта необходимых библиотек, которые мы установили в данной статье. После этого задаются токены для Telegram-бота и для OpenAI API.

Затем создается объект бота и диспетчер, который будет обрабатывать сообщения от пользователей. В данном случае используется команда /dalle для запроса на генерацию изображения.

Функция send_image() обрабатывает сообщения с командой /dalle и передает текст сообщения в функцию openai.Image.create(), которая использует нейросеть DALL-E 2 для генерации изображения на основе введенного текста. Затем полученное изображение отправляется пользователю с помощью метода message.answer_photo().

Код заканчивается вызовом функции executor.start_polling(), которая запускает бота и начинает обрабатывать входящие сообщения.

Что можно улучшить?

для улучшения кода можно добавить обработку ошибок, а также улучшить обработку текстовых запросов и передаваемых параметров для генерации изображений. Затем можно улучшить структуру кода, например, вынести токены в отдельный файл для удобства управления ими.

Не забываем, что запросы через API платные. Нам изначально дают баланс на 18\$ или более долларов.

Стоимость изображений по размерам:

Размер изображения	Цена за изображение
256×256	\$0.016
512×512	\$0.018
1024×1024	\$0.020

Запускаем бота, проверяем работу

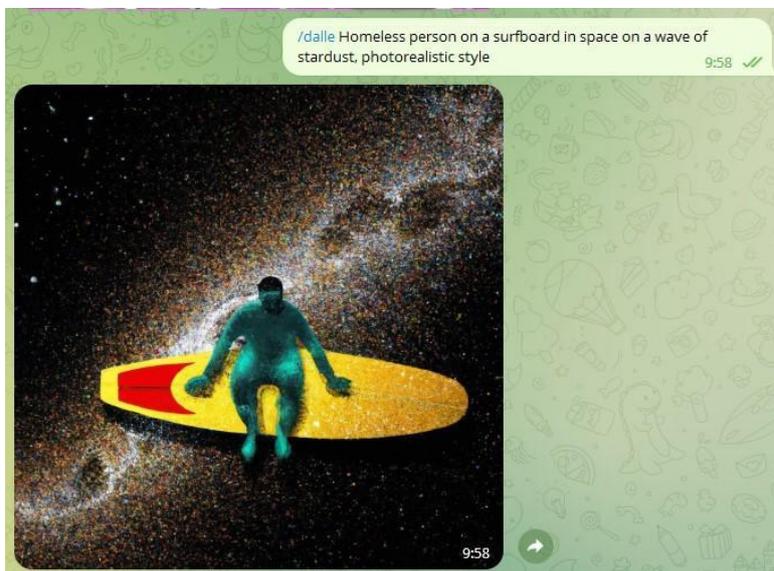
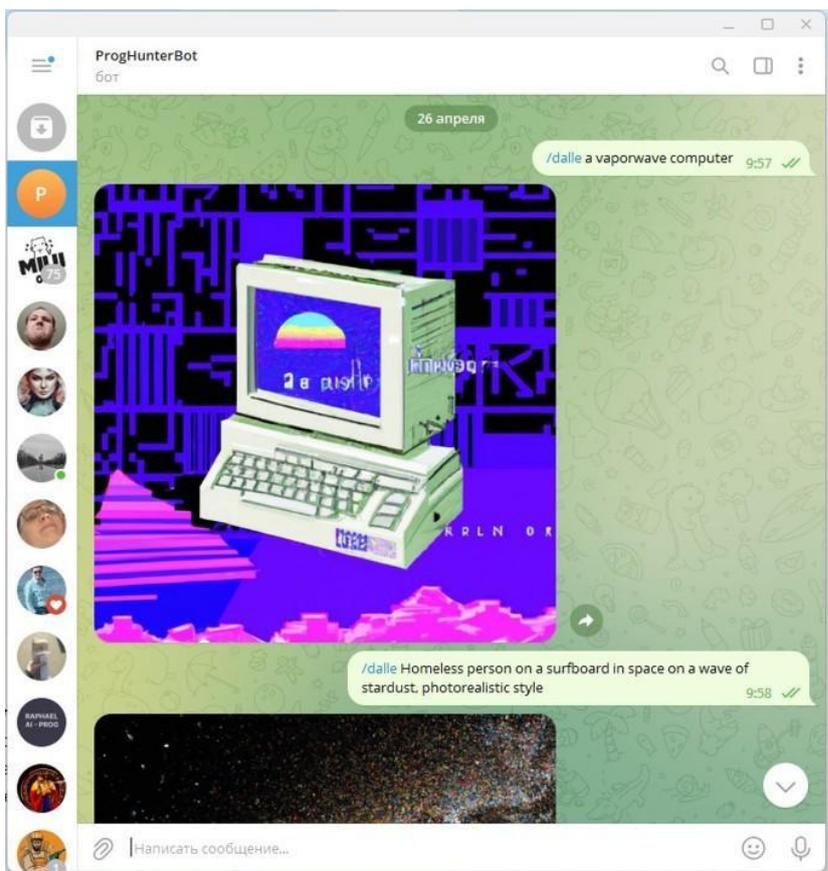
Далее нам необходимо запустить бота, для этого в терминале напишите ru main.py и если все сделали правильно, результат будет следующим:

Терминал

```
(venv) C:\Users\Razilator\Desktop\Projects\DalleTelegram>py main.py
```

Updates were skipped successfully.

Напишем нашему боту для генерации изображений



Цель работы: Создание бота на Python, для генерации изображений с помощью нейросети DALL-E для Telegram

Рабочее задание

- 1.Получаем API токен на openai (для chatGPT, DALL-E)
- 2.Получаем токен для бота в телеграмм
- 3.Создаем бота на Python для генерации изображений DALL-E
- 4.Запускаем бота, проверяем работу

Контрольные вопросы

- 1.Какие возможности предоставляет «умный» чат-бот для работы с изображениями?
- 2.Какие преимущества использования «умного» чат-бота в сравнении с традиционными методами обработки изображений?
- 3.Какой алгоритм используется чат-ботом для анализа и распознавания изображений?
- 4.Каковы технологии, лежащие в основе функционирования «умного» чат-бота в работе с изображениями?
- 5.Как можно интегрировать «умного» чат-бота для работы с изображениями в существующие системы и приложения?

Общие положения

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующего занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

Форма отчета:

- титульный лист;
- введение (цель и задачи);
- выполнение
- заключение

Время работы: **2 часа.**

Перечень использованных информационных ресурсов

№	Автор	Название	Издательство	Гриф издания	Год издания	Кол-во в библиотеке	Наличие на электронных носителях	Электронные уч. пособия
1	2	3	4	5	6	7	8	9
3.2.1 Основная литература								
3.2.1.1	. Балдин К.В	Информационные системы в экономике: Учебное пособие	М.:Инфра-М, ,		2022	-	-	ISBN 978-5-16-005009-6
3.2.2 Дополнительная литература								
3.2.2.1	Андрейчиков , А. В.	Интеллектуальные информационные системы и методы искусственного интеллекта : учебник	Москва : ИНФРА-М		2022	-	-	https://znaniu.m.com/catalog/product/1864091