

Государственное бюджетное негетиповое образовательное учреждение  
«Академия цифровых технологий»  
Санкт-Петербурга

ПРИНЯТО

На педагогическом совете  
ГБНОУ «Академия цифровых технологий»

Протокол  
от «30» августа 2024 г. № 3

УТВЕРЖДЕНО

Приказ от «2» сентября 2024 г. № 105-У

И. о. директора ГБНОУ  
«Академия цифровых технологий»

  
Зотова В.А.



Дополнительная общеразвивающая программа

**«Основы программирования нейронных сетей и применение ИИ»**

Срок освоения – 1 год

Возраст обучающихся: 15-17 лет

Разработчик:  
Рытов Алексей Максимович,  
педагог дополнительного образования

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

### Основные характеристики ДОП

*Направленность.* Дополнительная общеразвивающая программа «Основы программирования нейронных сетей и применение ИИ» имеет техническую направленность.

*Актуальность* реализации программы заключается в том, она обеспечивает работу по профориентации школьников в области инженерно-технических профессий, позволяет сделать предпрофессиональные пробы и страховку профессионального становления.

*Адресат:* рассчитана на обучающихся 15-17-летнего возраста, как мальчиков, так и девочек, которые проявляют интерес к программированию и современным информационным технологиям, с имеющимся базовым опытом программирования, в частности, на языке Python.

*Уровень освоения* – базовый.

*Объем и срок освоения:* всего 72 часа в год, 1 год.

*Цель и задачи:*

Цель: развить интерес и предоставить обучающимся начальные знания и практические навыки в области машинного обучения, нейронных сетей и искусственного интеллекта, достаточные для дальнейшего развития в этой области.

Задачи:

#### **Обучающие**

- Понимать и применять минимально необходимые математические основы в рамках алгоритмов машинного обучения и искусственного интеллекта (основы операций с матрицами, вычисление производных, функция потерь и ее минимизация как основа обучения нейронных сетей, и т.п.).
- Работать с языком программирования Python и соответствующими библиотеками (NumPy, PyTorch, Pandas, Matplotlib, Scikit-Learn), эффективно использовать их для создания, обработки и отображения данных.
- Владеть навыками локальной установки библиотек и моделей нейронных сетей с открытым кодом, различного назначения. Понимать основы использования GitHub, как основного репозитория проектов с открытым кодом, HuggingFace как основного репозитория для открытых моделей нейронных сетей.
- Уметь использовать существующие модели нейронных сетей с открытым кодом и применять их для решения разнообразных задач, включая классификацию изображений, генеративное искусство, чат боты и т.п.
- Уметь проводить обучение или до-обучение моделей нейронных сетей, работать с наборами данных для обучения нейронных сетей, оценивать эффективность обучения.
- Применять инструменты генеративной графики, такие как StableDiffusion+ComfyUI, и создавать workflows (схемы автоматической обработки данных) на их основе.
- Применять LLM ("большие языковые модели") с открытым кодом, уметь устанавливать и настраивать, строить "интеллектуальные" системы на их основе (langchain, векторные БД)

### **Развивающие:**

Способствовать развитию:

- аналитического мышления и способности решать задачи на основе логики.
- навыков программирования и работы с кодом.
- навыков выражения своих мыслей, постановки задач, формулировки запросов информации, понятных как для окружающих людей, так и для систем ИИ.
- коммуникативных навыков, необходимых для эффективной работы в группе.

### **Воспитательные:**

- Воспитание ответственного отношения к создаваемым решениям и проектам.
- Понимание этических аспектов при разработке и использовании систем ИИ.
- Поддержание интереса к творческому и исследовательскому подходу при решении задач.
- Формирование понимания важности технической и информационной культуры в современном обществе.
- Стимулирование осознанного выбора будущей профессиональной сферы деятельности

### **Планируемые результаты.**

Реализация данной образовательной программы предоставит старшеклассникам возможность формировать осмысленное отношение к области искусственного интеллекта, машинного обучения и программирования, и практические базовые умения в этой области, необходимые для дальнейшего профессионального развития.

### **Предметные результаты:**

Обучающиеся **будут знать:**

- Классификацию задач и области применения машинного обучения и искусственного интеллекта (включая задачи классификации, регрессии, кластеризации, обработки естественного языка и другие).
- Разновидности и особенности применения нейронных сетей (сверточные, рекуррентные, генеративные и др.) в различных областях, таких как обработка изображений, текстов и аудио.
- Основные понятия и термины, используемые в машинном обучении (нейрон, вес, активация, функция потерь и др.), а также в области искусственного интеллекта (распознавание образов, генерация контента, автоматизированный анализ данных и др.).
- Математические основы "обучения" нейронных сетей (градиентный спуск, обратное распространение ошибки, активационные функции и др.).
- Основы работы с библиотеками Python для анализа данных и машинного обучения (pyTorch, NumPy, Pandas, Matplotlib, Scikit-Learn и др.), включая основные методы работы с данными, визуализации и обучения моделей.
- Практические методы создания и обучения простых нейронных сетей и моделей искусственного интеллекта.

Обучающиеся **будут уметь** (получат практический опыт):

- Находить в сети тренировочные наборы данных (datasets) для различных задач машинного обучения, такие как классификация изображений, обработка естественного языка и другие (например, на платформах Kaggle, HuggingFace и др.).
- Самостоятельно формировать и обучать простейшие нейронные сети с использованием фреймворка PyTorch, включая решение задач классификации (распознавания изображений) на примере датасета MNIST, генерации текстов и изображений.
- Находить, устанавливать, до-обучать и использовать модели нейронных сетей с открытым исходным кодом, доступные на платформах HuggingFace, GitHub и других ресурсах (например, модели GPT-3, LLAMA, StableDiffusion и др.) для решения разнообразных задач в области машинного обучения и искусственного интеллекта.

### **Метапредметные результаты:**

Обучающиеся разовьют:

- внимание, память, логическое и алгоритмическое мышление;
- умение эффективно взаимодействовать в групповой работе;
- способность осознанно и эффективно применять инженерные технологии в своей повседневной и учебной деятельности.

### **Личностные результаты:**

У обучающихся будут сформированы:

- ответственность и понимание важности выбора будущей профессиональной направленности;
- навыки креативного подхода к решению задач;
- культура конструктивного общения в рамках коллективной работы.

Таким образом, данная образовательная программа направлена на формирование у старшеклассников навыков и знаний в области искусственного интеллекта и машинного обучения, способствующих их активному участию в современной технической среде.

## ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ

*Язык реализации:* русский.

*Форма обучения:* очная.

*Особенности реализации:*

Программа и/или ее части могут быть реализованы с применением электронного обучения (ЭО) и/или дистанционных образовательных технологий (ДОТ). В процессе очного обучения педагог использует авторские цифровые продукты, которые размещены в сети Интернет. Это описание проектов, заданий, образцы выполнения, методические рекомендации и требования к выполнению заданий и проектов. Поскольку база заданий и проектов достаточно обширная, обучающиеся могут также воспользоваться в любое время, чтобы приобрести дополнительный опыт и закрепить свои знания. Дистанционные образовательные технологии актуальны для обучающихся, которые по каким-либо причинам не могут посетить конкретное занятие.

Образовательный процесс имеет развивающий характер, направлен на развитие природных задатков учащихся, на реализацию их интересов и способностей. Широко применяются личностно-ориентированные технологии обучения. Выбор методов обучения осуществляется педагогом и зависит от возрастных особенностей обучающихся, темы и формы занятий. Широко используется проектная технология.

Методика проведения занятий предполагает постоянное создание ситуаций успешности, радости от преодоления трудностей в освоении изучаемого материала и при выполнении самостоятельной работы. Этому способствуют совместные обсуждения технологии выполнения заданий и проектов, а также поощрение, создание положительной мотивации, актуализация интереса, олимпиады и конкурсы.

*Условия набора и формирования групп*

- Программа адаптирована для школьных групп, учащихся 9-11 классов.
- Формирование группы производится школой, обучающиеся должны иметь практический опыт программирования (предпочтительно на языке Python).
- Добор в группу в течение учебного года, как правило, не производится.
- Численный состав обучающихся от 10 до 15 человек в группе. Рекомендуется, чтобы численность группы не превышала количество имеющихся компьютерных рабочих мест.

*Форма организации и проведения занятий:*

В ходе образовательного процесса применяются различные формы организации деятельности обучающихся: индивидуальные, групповые. Виды занятий по дополнительной общеобразовательной общеразвивающей программе определяются содержанием программы. Очными формами проведения занятий являются: инструктаж, опрос, рассказ, беседа, диспут, демонстрация, самостоятельная работа, коллективный анализ интересных индивидуальных решений, презентация работ, защита проектов, конкурс.

## **Материально-техническое обеспечение.**

### **Для проведения учебного процесса необходимы:**

- Рабочее место педагога (компьютер, интерактивная доска или проектор, МФУ)
- компьютерный класс с персональными компьютерами, по числу обучающихся, с NVidia GPU не хуже GTX 3060 (поддержка CUDA, не менее 12GB VRAM), и SSD накопителями с не менее 512 GB свободного пространства, что необходимо для локальной установки наборов данных и моделей нейронных сетей (как правило, имеющих большой объем).
- сетевое оборудование для выхода в Интернет, крайне желательно кэширование «больших» файлов на уровне прокси-сервера в локальной сети учреждения.

### **Программное обеспечение:**

**Отметим, что все ПО для данного курса работает и в Linux и в Windows и является бесплатным.**

- Операционная система: Windows 10/11 или Linux (Debian based), желательно с dual boot (двойной загрузкой).
- Язык Python 3.xx, с пакетным менеджером pip и библиотеками для анализа данных и машинного обучения (NumPy, Pandas, Matplotlib, Seaborn, Scikit-Learn, PyTorch и др., могут устанавливаться по мере необходимости).
- Visual Studio Code (VS Code): Интегрированная среда разработки для написания, отладки и выполнения кода Python.
- PyTorch: Фреймворк для глубокого обучения.
- Библиотека Transformers (от HuggingFace): для работы с пред-обученными моделями в области NLP и генеративных моделей.
- Git: Для управления версиями кода и совместной работы.
- Ollama - локальный сервер для работы с языковыми моделями
- LM Studio - приложение для работы с локальными языковыми моделями

### **Расходные материалы (из расчета на группу 15 человек):**

- Бумага (5 пачек) и картриджи для МФУ (по 1) на год

### **Кадровое обеспечение**

Педагог дополнительного образования.

## УЧЕБНЫЙ ПЛАН

№ п/п	Название раздела, темы	Количество часов			Формы контроля
		Всего	Теория	Практика	
<b>1</b>	<b>Введение</b>	<b>2</b>	<b>1</b>	<b>1</b>	Опрос, беседа
1.1	Введение, инструктаж по ТБ и взаимное знакомство, ознакомление с оборудованием и ПО компьютерного класса	2	1	1	
<b>2.</b>	<b>Основы построения и обучения нейронных сетей</b>	<b>20</b>	<b>6</b>	<b>14</b>	Практическая работа
2.1	Особенности языка Python и библиотеки анализа данных, среды выполнения и управление библиотеками	4	1	3	Практическая работа
2.2	Библиотеки анализа данных и практические примеры работы с ними. "Классические" задачи оптимизации.	4	1	3	Практическая работа
2.3	Базовые принципы и математические основы нейронных сетей и "глубокого обучения"	6	2	4	Практическая работа
2.4	Библиотека pyTorch, тензоры и описание нейронных сетей в pyTorch	6	2	4	Практическая работа
<b>3.</b>	<b>Обработка изображений нейронными сетями</b>	<b>20</b>	<b>6</b>	<b>14</b>	Практическая работа
3.1	Задачи классификации изображений, простой персептрон, сверточные сети.	6	2	4	Практическая работа
3.2	Автоэнкодеры, их использование, латентное пространство. Вариационный автоэнкодер.	6	2	4	Практическая работа
3.3	Глубокие сверточные сети для работы с ImageNet. Перенос стиля.	4	1	3	Практическая работа
3.4	Генеративные модели. Принцип работы StableDiffusion. Практическое освоение SD с интерфейсом ComfyUI	4	1	3	Практическая работа

<b>4.</b>	<b>Обработка текста и языковые модели</b>	<b>28</b>	<b>10</b>	<b>18</b>	Практическая работа
4.1	Задачи обработки и предсказания текста, "трансформеры" и понятие "внимания"	6	2	4	Практическая работа
4.2	Большие языковые модели. ChatGPT и другие. Программирование приложений на базе LLM, Web-программирование в Streamlit.	12	4	8	Практическая работа
4.3	RAG: Вызов функций из LLM. Векторные базы данных, как способ хранения знаний.	8	3	5	Практическая работа
4.4	Знакомство с когнитивными и агентными архитектурами.	2	1	1	Практическая работа
<b>5</b>	<b>Подведение итогов</b>	<b>2</b>	<b>0</b>	<b>2</b>	Семинар
5.1	Финальный семинар, обсуждение итогов курса	2	0	2	Семинар
	<b>Итого</b>	<b>72</b>	<b>23</b>	<b>49</b>	



**Государственное бюджетное нетиповое образовательное учреждение  
«Академия цифровых технологий»  
Санкт-Петербурга**

УТВЕРЖДЕН

Приказ от «\_\_» \_\_\_\_\_ 2024 г. № \_\_

И. о. директора ГБНОУ  
«Академия цифровых технологий»

\_\_\_\_\_/Зотова В. А./

«\_\_\_\_\_» \_\_\_\_\_ 2024г.

**КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК**

реализации дополнительной общеразвивающей программы

**«Основы программирования нейронных сетей и применение ИИ»**

**на 2024/2025 учебный год**

Год обучения, группа	Дата начала занятий	Дата окончания занятий	Количество учебных недель	Количество учебных дней	Количество учебных часов	Режим занятий <sup>1</sup>
1 год, группа 1.1	9.09.24	По окончании реализации программы <sup>2</sup>	36	36	72	1 раз в неделю по 2 часа

<sup>1</sup> С указанием продолжительности академического часа в соответствии с локальным нормативным актом образовательной организации.

<sup>2</sup> Дата окончания обучения по программе может изменяться в связи с корректировкой в календарно-тематических планах дополнительных общеобразовательных (общеразвивающих) программах в части плановых дат проведения занятий, с переносом на фактические даты, в соответствии с приказами ГБНОУ «АЦТ» в связи с отсутствием педагогов по уважительной причине и невозможностью организовать замещение в указанные в календарно-тематических планах программ даты до полного их исполнения.

## РАБОЧАЯ ПРОГРАММА

<i>Особенности обучения:</i>	В соответствии с ДОП.
<i>Задачи:</i>	В соответствии с ДОП
<i>Планируемые результаты:</i>	В соответствии с ДОП

### *Содержание программы*

#### **Раздел 1. Введение**

**Тема 1.1.** Введение, инструктаж по ТБ и взаимное знакомство, ознакомление с оборудованием и ПО компьютерного класса.

**Практика:** Сесть за компьютеры, найти где что лежит, запустить Python и сделать "Hello, world" в разных средах:

- python, запущенный в командной строке
- python, запущенный из IDLE
- Jupyter Notebook в VS Code
- В браузере через pyodide
- В repl.it (просто питон, без Jupyter)

#### **Раздел 2. Основы построения и обучения нейронных сетей**

**Тема 2.1.** Особенности языка Python и библиотеки анализа данных, среды выполнения и управление библиотеками

**Теория:** установка библиотек в Python с использованием pip, виртуальные окружения Python, работа в Jupyter/ Google Colab, библиотеки numpy, matplotlib. Работа с массивами, способа индексации, арифметические операции, np.linspace() и другие функции numpy. Построение графиков с matplotlib.

**Практика:** Создаём «блокноты» с несложной задачей, например - генерируем график функции.

**Тема 2.2.** Библиотеки анализа данных и практические примеры работы с ними. «Классические» задачи оптимизации.

**Теория:** подготовка данных, нормализация, унитарный код (one-hot encoding), библиотека pandas, линейная регрессия, логистическая регрессия.

**Практика:** повторяем одно из упражнений курса «ИИ Старт».

**Источник:** Курс по машинному обучению. Проект «ИИ Старт» (<https://stepik.org/course/125587/syllabus>)

**Тема 2.3.** Базовые принципы и математические основы нейронных сетей и "глубокого обучения"

**Теория:** как устроены биологические нейронные сети, каким образом это отображается в перемножение векторов и матриц. Зачем нужно вводить нелинейность (ReLU(), atanh() и др.). Как аппроксимируется произвольная функция. Параметры нейронной сети: веса и смещения. Понятие о производных, частные производные. Ч.п. по параметрам. Метод градиентного спуска и "обратное распространение ошибки" (back propagation). Функция потерь: что такое, как задать.

**Практика:** Используем Jupyter-«блокнот» с библиотекой интерактивности ipywidgets, чтобы показать, как складываются несколько линейных функций (от одной переменной) с наложенной нелинейностью и без нее (без нее сумма любых линейных функций даст линейную функцию).

Решаем "задачу распознавания" по двум параметрам в "песочнице" <https://playground.tensorflow.org/>.

Написание собственной библиотеки (класса) градиентного спуска, по мотивам видеоурока от Андрея Карпати. Библиотека `micrograd` сделана так, чтобы, в упрощенной форме, максимально походила на *тензоры* в `pyTorch`.

**Источники:**

(<https://www.youtube.com/watch?v=VMj-3S1tku0>) и на основе его библиотеки `micrograd` (<https://github.com/karpathy/micrograd>).

**Тема 2.4.** Библиотека `pyTorch`, тензоры и описание нейронных сетей в `pyTorch`

**Теория:** тензоры в `torch`, их сходства и отличия от массивов в `numpy`. Что еще умеет тензор (`autograd`). Определение класса для нейронной модели в `pyTorch` (`torch.nn.Module`), загрузка данных и обучение модели. Выбор функции потерь для разных задач. Разбиение на тренировочную и тестовую выборки, определение степени обученности (график снижения функции потерь). Как строится последний слой нейронки для задач классификации.

**Практика:** (а) экспериментируем с *тензорами* `pyTorch` как заменой многомерных массивов в `numpy`: разные способы их инициализировать, разные операции и пр. Например, можно построить графики каких-нибудь функций. (б) обучаем какие-нибудь простые нейронки в `TorchStudio`.

**Источники:**

учебные видео с сайта `pyTorch` (english) - <https://pytorch.org/tutorials/beginner/introyt.html>, курс на Степике "ИИ Старт" (<https://stepik.org/lesson/813151/step/4?unit=816457>), тренировочные "ноутбуки" для `Jupyter/Colab` по ссылкам из этого курса.

**Раздел 3. Обработка изображений нейронными сетями.**

**Тема 3.1:** Задачи классификации изображений, простой перцептрон, сверточные сети.

**Теория:** "линейные" массивы нейронов и 2-мерные изображения. Задача распознавания (регрессия). Структура последнего слоя, функция потерь `nn.CrossEntropyLoss()`. Сверточные фильтры как способ передать геометрические особенности изображения. Подготовка изображений. Загрузчики и фильтры в `torchvision`.

**Практика:** Строим и обучаем сначала MLP (перцептрон), а затем сверточную сеть для распознавания рукописных цифр датасета MNIST.

**Источники:**

курс "ИИ Старт", видеоуроки от Татьяны Гайнцевой (<https://stepik.org/lesson/813153/step/1?unit=816459> и <https://www.youtube.com/@AladdinPersson> (english) дальше),

**Тема 3.2. Автоэнкодеры, их использование, латентное пространство. Вариационный автоэнкодер.**

**Теория:** Автоэнкодер, как нейронная сеть с одинаковым входом и выходом. Латентное пространство, изображение как точка в латентном пространстве. Понижение размерности данных, удаление помех и искажений. Вариационный автоэнкодер, как способ собрать похожие объекты вместе, а непохожие разнести друг от друга. Плавное изменение объекта при движении точки в латентном пространстве.

**Практика:** Строим вариационный автоэнкодер на примере датасета MNIST (превращения цифр), в виде интерактивного "блокнота" (предпочтительно - локально в `VS Code/Jupyter` либо в `Google Colab`),

**Тема 3.3.** Глубокие сверточные сети для работы с ImageNet. Перенос стиля.

**Теория:** краткий исторический экскурс в распознавание изображений, датасет ImageNet, виды готовых сверточных сетей и их сравнение, замена "голов" (последних полносвязных слоев) сверточных сетей и их до-обучение, визуализация работы слоев сверточной сети, математическое выражение "стиля" изображения, алгоритм переноса стиля.

**Практика:** загрузка и использование готовых сверточных сетей для классификации изображений, получение характеристик модели, до-обучение сети (например, ResNet19) для нестандартной задачи распознавания (например, возраст / пол / раса по лицу, используя UTKFace или другой аналогичный датасет), визуализация работы сверточных слоев. Способы "внедрения" в готовую модель. Пример реализации алгоритма переноса стиля.

**Тема 3.4. Модели для генерации изображений.** Принцип работы StableDiffusion. Практическое освоение SD с интерфейсом ComfyUI.

**Теория:** структура и использование U-net, нейронная сеть stable diffusion, латентные изображения, кондиционирование, нейронка CLIP, энкодеры, декодеры, сэмплеры и пр. Inpainting, outpainting, обучение text embeddings, LoRa, ControlNet и пр. Программный интерфейс для использования ComfyUI из своих приложений.

**Практика:** (1) упражнение по восстановлению изображения из шума, (2) разнообразная работа с ComfyUI, как минимум - по всем примерам из его репозитория.

**Источники:**

<https://tree.rocks/make-diffusion-model-from-scratch-easy-way-to-implement-quick-diffusion-model-e60d18fd0f2e>, множество видео по ComfyUI, примеры workflows, включенные с различными "нодами" (узлами).

<https://9elements.com/blog/hosting-a-comfyui-workflow-via-api/> - подробное объяснение и код для программного доступа к ComfyUI через его API (endpoints).

#### **Раздел 4. Обработка текста и языковые модели**

**Тема 4.1.** Задачи обработки и предсказания текста, понятие "внимания"

**Теория:** токенизация, получение наборов данных из массива текста. Задача предсказания следующего токена. Эмбединги. Отслеживание контекста, возможно - "внимание" (attention), если до этого дойдет.

**Практика:** Воспроизводим генератор имен, воспроизводим "шекспирогенератор", можно заменить входные данные на Толстого, Пушкина и пр. Для этого находим и подготавливаем наборы текстов (например, "Войну и мир"), обучаем модель.

**Источники:**

1. видеоуроки Андрея Карпати про "makemore" (генератор имен - <https://www.youtube.com/watch?v=PaCmpygFfXo>), Исходный код: <https://github.com/karpathy/makemore>
2. его же урок "Let's build GPT: from scratch, in code, spelled out". (<https://www.youtube.com/watch?v=kCc8FmEb1nY>) - супер-упрощенный GPT для генерации Шекспиро-образного текста. Исходный код: <https://github.com/karpathy/nanoGPT>.

**Тема 4.2.** Большие языковые модели. ChatGPT и другие. Программирование приложений на базе LLM, Web-программирование в Streamlit.

**Теория:** способы и особенности установки моделей с github/huggingface. Выбор модели LLM, уровни "обученности" моделей (базовая модель, диалоговая модель). Квантованный формат GGUF. Способы подключения к API модели - напрямую через llama\_cpp, через Ollama, через LM\_Studio (как сервер). Стандартный OpenAI API и его структура чат-

сообщений. Режим "чат" и режим "дополнение". Простейшее веб-приложение в Streamlit, цикл работы скрипта streamlit, кэширование ресурсов, основные типы widgets (элементов экранного интерфейса)

#### **Практика:**

- Выбираем и устанавливаем локально языковые модели
- Проверяем разные способы доступа (локально llama\_cpp и ollama, groq из repl.it)
- Реализуем простой ИИ-чат как веб-приложение в Streamlit, учимся формировать и отображать контекст диалога.

**Тема 4.3. RAG: Вызов функций из LLM.** Векторные базы данных, как способ хранения знаний.

**Теория:** Формат представления структурированных данных JSON. JSON как способ описания вызова функций. "Схема" структуры в JSON. Включение структурированной генерации в LLM. Модели, обученные генерировать вызовы функций. Назначение и использование векторных баз данных. "Эмбединги" слов и фрагментов текста, поиск по близости векторов, векторные БД. Чем генерировать эмбединги (Word2Vec, FastText, BERT и другие). Подготовка текстов для включения в векторную БД. Структура типичного приложения с использованием векторной БД. Распространенные локальные и "облачные" векторные БД.

#### **Практика:**

- 1) добавление в чат бот вызовов функций (например, для получения времени, географического положения по ИП, погоды и пр.)
- 2) Создание векторной БД из произвольного длинного текста (книга, учебник и пр). Учим LLM задавать вопросы по каждому абзацу текста, по вопросам генерируем эмбединги, заливаем в локально установленную векторную БД (Chroma или Milvus). Заставляем наш чат-бот отвечать на вопросы по текстам.

#### **Источники:**

1. БД Milvus (<https://milvus.io/>)
2. БД Chroma (<https://docs.trychroma.com/>)
3. Word2Vec, BERT (найти аналог с поддержкой русского языка - ruBERT от Сбер?)

**Тема 4.4.** Когнитивные и агентные архитектуры

**Теория:** Понятия "цепочка мыслей", "дерево мыслей" (chain of thought, tree of thought). Библиотеки LangChain, ReAct. Чем может быть "приложение с LLM": чем оно может отличаться от просто чат-бота. Понятие "обучения с подкреплением" (reinforcement learning)

**Практика:** продолжаем делать свои мини-проекты, у кого какие были, добавляя в них всякое причудливое.

#### **Источники:**

1. ReAct: <https://www.promptingguide.ai/techniques/react>
2. LangChain: <https://github.com/langchain-ai/langchain>
3. Обзоры LangChain: [https://www.youtube.com/watch?v=aywZrzNaKjs&ab\\_channel=Rabbitmetrics](https://www.youtube.com/watch?v=aywZrzNaKjs&ab_channel=Rabbitmetrics)  
(и многие другие)

**Тема 5.** Финальный семинар, обсуждение итогов курса.

**Практика:** обсуждение итогов курса.

**Государственное бюджетное нетиповое образовательное учреждение  
«Академия цифровых технологий» Санкт-Петербурга**

СОГЛАСОВАНО:

\_\_\_\_\_/Панкратова Л.П./  
методист

\_\_\_\_\_/Тихонова А.И./  
заведующий сектором

\_\_\_\_\_/Григорьев Т.М./  
заведующий УМЦ

«\_\_» \_\_\_\_\_ 2024 г.

**Календарно-тематический план  
к дополнительной общеразвивающей программе  
«Основы программирования нейронных сетей и применение ИИ»  
на 2024/2025 учебный год**

Педагог: Рытов А.М.

Год обучения: 1

Группа: № 1.1

Подпись педагога \_\_\_\_\_

№ п/п	Дата		№ темы	Тема занятия	Количество часов		
	план	факт			Всего	Теория	Практика
1	12.09.2024		1.1	Вводное занятие	2	1	1
2	19.09.2024		2.1	Особенности Python и библиотеки анализа данных	2	0.5	1.5
3	26.09.2024		2.1	Особенности Python и библиотеки анализа данных	2	0.5	1.5
4	03.10.2024		2.2	Библиотеки анализа данных и задачи оптимизации.	2	0.5	1.5
5	10.10.2024		2.2	Библиотеки анализа данных и задачи оптимизации.	2	0.5	1.5
6	17.10.2024		2.3	основы нейронных сетей и принципы их обучения	2	1	1
7	24.10.2024		2.3	основы нейронных сетей и принципы их обучения	2	0.5	1.5
8	31.10.2024		2.3	основы нейронных сетей и принципы их обучения	2	0.5	1.5

9	07.11.2024		2.4	Нейронные сети в pyTorch	2	1	1
10	14.11.2024		2.4	Нейронные сети в pyTorch	2	0.5	1.5
11	21.11.2024		2.4	Нейронные сети в pyTorch	2	0.5	1.5
12	28.11.2024		3.1	Задачи классификации изображений, простой персептрон, сверточные сети.	2	1	1
13	05.12.2024		3.1	Задачи классификации изображений, простой персептрон, сверточные сети.	2	0.5	1.5
14	12.12.2024		3.1	Задачи классификации изображений, простой персептрон, сверточные сети.	2	0.5	1.5
15	19.12.2024		3.2	Автоэнкодеры и латентное пространство.	2	1	1
16	26.12.2024		3.2	Автоэнкодеры и латентное пространство. Вариационный автоэнкодер.	2	0.5	1.5
17	09.01.2025		3.2	Вариационный автоэнкодер.	2	0.5	1.5
18	16.01.2025		3.3	Глубокие сверточные сети для работы с ImageNet. Перенос стиля.	2	0.5	1.5
19	23.01.2025		3.3	Глубокие сверточные сети для работы с ImageNet. Перенос стиля.	2	0.5	1.5
20	30.01.2025		3.4	Генерация изображений, StableDiffusion, ComfyUI	2	0.5	1.5
21	06.02.2025		3.4	Генерация изображений, StableDiffusion, ComfyUI	2	0.5	1.5
22	13.02.2025		4.1	Задачи обработки и предсказания текста	2	1	1
23	20.02.2025		4.1	Задачи обработки и предсказания текста	2	0.5	1.5
24	27.02.2025		4.1	Задачи обработки и предсказания текста	2	0.5	1.5
25	06.03.2025		4.2	Большие языковые модели. Выбор и установка	2	1	1
26	13.03.2025		4.2	Программный доступ к LLM	2	1	1
27	20.03.2025		4.2	Программный доступ к LLM	2	0.5	1.5
28	27.03.2025		4.2	Web-приложения со Streamlit	2	0.5	1.5

29	03.04.2025		4.2	Пишем ИИ-чат	2	0.5	1.5
30	10.04.2025		4.2	Пишем ИИ-чат	2	0.5	1.5
31	17.04.2025		4.3	Вызов функций языковыми моделями.	2	1	1
32	24.04.2025		4.3	Вызов функций языковыми моделями.	2	0.5	1.5
33	08.05.2025		4.3	Текстовые эмбединги, векторные базы данных	2	1	1
34	15.05.2025		4.3	Текстовые эмбединги, векторные базы данных	2	0.5	1.5
35	22.05.2025		4.4	Знакомство с когнитивными и агентными архитектурами.	2	1	1
36	29.05.2025		5.1	Итоговое занятие	2	0	2
				<b>ИТОГО:</b>	<b>72</b>	<b>23</b>	<b>49</b>



## МЕТОДИЧЕСКИЕ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

### Методические материалы

Для проведения занятий по программе используются практические задания, презентации, иллюстративные материалы, теоретический анализ соответствия выполняемых индивидуальных проектов, сравнительный анализ результатов по практическим заданиям.

Распределение методического обеспечения по темам (разделам) дополнительной общеразвивающей программы «Основы программирования нейронных сетей и применение ИИ» в соответствии с учебным планом.

Раздел или тема программы	Формы проведения занятий	Форма организации деятельности обучающихся на занятии	Приёмы и методы, используемые педагогом	Дидактический материал	Формы подведения итогов	
					Очно	с применением ЭО
<p><b>1. Введение</b> 1.1. Введение, инструктаж по ТБ и взаимное знакомство, ознакомление с оборудованием и ПО компьютерного класса</p> <p><b>2. Основы построения и обучения нейронных сетей</b> 2.1. Особенности языка Python и библиотеки анализа данных, среды выполнения и управление библиотеками 2.2. Библиотеки анализа данных и практические примеры работы с ними. "Классические" задачи оптимизации. 2.3. Базовые принципы и математические основы нейронных сетей и "глубокого</p>	Лекция, практическое занятие, индивидуально-групповая работа	Групповая, фронтальная	Объяснительно-иллюстративный, деятельностный, самообучение Словесный, наглядный, проблемно-поисковый, метод демонстрационного примера	Презентации, карточки с заданиями, инструкции, методические рекомендации	Очно: Беседа Тестирование Практическая работа Презентация проектов	С применением ЭО

<p>обучения"</p> <p>2.4. Библиотека pyTorch, тензоры и описание нейронных сетей в pyTorch</p> <p><b>3.1. Обработка изображений нейронными сетями</b></p> <p>3.1. Задачи классификации изображений, простой персептрон, сверточные сети.</p> <p>3.2. Автоэнкодеры, их использование, латентное пространство. Вариационный автоэнкодер.</p> <p>3.3. Глубокие сверточные сети для работы с ImageNet. Перенос стиля.</p> <p>3.4. Генеративные модели. Принцип работы StableDiffusion. Практическое освоение SD с интерфейсом ComfyUI</p> <p><b>4. Обработка текста и языковые модели</b></p> <p>4.1. Задачи обработки и предсказания текста, "трансформеры" и понятие "внимания"</p> <p>4.2. Большие языковые модели. ChatGPT и другие. Программирование приложений на базе LLM, Web-программирование в Streamlit.</p> <p>4.3. RAG: Вызов функций из LLM. Векторные базы данных, как способ хранения знаний.</p>						
--	--	--	--	--	--	--

4.4. Знакомство с когнитивными и агентными архитектурами. <b>5. Подведение итогов</b> 5.1. Финальный семинар, обсуждение итогов курса						
---	--	--	--	--	--	--

## **Оценочные материалы**

*Форма оценивания:* балльно-рейтинговая система оценивания (см. *Приложение № 1* «Диагностическая карта оценки уровня образовательных возможностей обучающихся.»).

*Критерии (см. Приложение № 2 «Таблица параметров и критериев оценивания по программе»):*

1. Мотивация (выраженность интереса к занятиям).
2. Самооценка деятельности на занятиях.
3. Ответственность и организованность.
4. Умение вести поиск, анализ, отбор информации.
5. Умение работать в группе.
6. Коммуникативная компетенция.
7. Навыки анализа поставленной задачи по проекту с целью определения необходимой конструкции и алгоритма программирования робота.
8. Навыков проектирования, сборки и программирования конструкций под задачи проекта.
9. Навыки составления технической документации под задачи проекта.

***В течение учебного года*** обучающиеся ДОП «Основы программирования нейронных сетей и применение ИИ» должны выполнить практические задания и представить их на итоговом занятии.

## Информационные источники

### Нормативные документы

1. Федеральный закон Российской Федерации от 29.12.2012 №273-ФЗ «Об образовании в Российской Федерации».
2. Распоряжение Правительства Российской Федерации от 29.05.2015 № 996-р «Об утверждении Стратегии развития воспитания в РФ на период до 2025 года».
3. Концепция развития дополнительного образования детей, утвержденной распоряжением Правительства Российской Федерации от 31 марта 2022г. №678-р.
4. Приказ Министерства просвещения Российской Федерации от 27.07.2022 №629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам» (вступает в силу с 01.03.2023 года, действует до 28.02.2029 года).
5. Приказ Министерства образования и науки Российской Федерации и Министерства просвещения Российской Федерации от 05.08.2020 №882,391 «Об организации и осуществлении образовательной деятельности при сетевой форме реализации образовательных программ» (действует до 01.09.2027 года).
6. Профессиональный стандарт «Педагог дополнительного образования детей и взрослых», утвержденный Приказом Министерства труда и социальной защиты Российской Федерации от 22.09.2021г. №652н (действует до 01.09.2028 года).
7. Постановление Главного государственного санитарного врача РФ от 28.09.2020 №28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи».
8. Закон Санкт-Петербурга от 26 июня 2013 года №461-83 «Об образовании в Санкт-Петербурге».
9. Закон Санкт-Петербурга от 29 июня 2016 года №453-87 «О патриотическом воспитании в Санкт-Петербурге».
10. Распоряжение Комитета по образованию от 25.08.2022 №1676-р «Об утверждении критериев оценки качества дополнительных общеразвивающих программ, реализуемых организациями, осуществляющими образовательную деятельность, и индивидуальными предпринимателями Санкт-Петербурга».
11. Устав ГБНОУ «Академия цифровых технологий».
12. Локальными документами ГБНОУ «Академия цифровых технологий», регламентирующими организацию образовательного процесса.
13. Действующие законодательные акты, регламентирующие образование в Российской Федерации, в том числе законодательные акты и распорядительные документы в области применения дистанционных образовательных технологий, дистанционного и электронного обучения.

### Для обучающихся

1. Видеоуроки Андрея Карпати про "makemore" (генератор имен - <https://www.youtube.com/watch?v=PaCmpygFfXo>),
2. Видеоуроки Андрея Карпати "Let's build GPT: from scratch, in code, spelled out". (<https://www.youtube.com/watch?v=kCc8FmEb1nY>) - супер-упрощенный GPT для генерации Шекспиро-образного текста. Исходный код: <https://github.com/karpathy/nanoGPT>.
3. БД Milvus (<https://milvus.io/>)
4. БД Chroma (<https://docs.trychroma.com/>)

5. Word2Vec, BERT (найти аналог с поддержкой русского языка - ruBERT от Сбер?)
6. ReAct: <https://www.promptingguide.ai/techniques/react>
7. LangChain: <https://github.com/langchain-ai/langchain>







Таблица параметров и критериев оценивания по программе:  
«Цифровое мейкерство и проектная деятельность», ФИО педагога \_\_\_\_\_

Параметры		Уровни	Степень выраженности качества	Оценка параметров
Личностные	Мотивация (выраженность интереса к занятиям)	Высокий	Проявляет интерес и творческое отношение к изучаемым темам, стремится получить дополнительную информацию	3
		Средний	Интерес возникает к новому материалу, но не к способам его применения на практике	2
		Низкий	Интерес практически не обнаруживается	1
	Самооценка деятельности на занятиях	Высокий	Может самостоятельно оценить свои возможности в выполнении задания, учитывая изменения известных способов действия	3
		Средний	Может с помощью педагога оценить свои возможности в решении задания, учитывая изменения известных ему способов действий	2
		Низкий	Учащийся не умеет, не пытается и не испытывает потребности в оценке своих действий – ни самостоятельной, ни по просьбе педагога	1
	Ответственность и организованность	Высокий	Проявляет самостоятельность, пунктуальность и ответственность в подготовке к занятиям.	3
		Средний	Проявляет самостоятельность, но при подготовке к занятиям требуется внешняя стимуляция.	2
		Низкий	Уровень самостоятельности учащихся низкий, при подготовке к занятиям требуется постоянная внешняя стимуляция.	1
Метапредметные	Умение работать в группе	Высокий	Способен к сотрудничеству, умеет слушать педагога и партнера, легко приходит к согласию.	3
		Средний	Способен к сотрудничеству, но не всегда умеет аргументировать свою позицию и слушать партнера	2
		Низкий	В совместной деятельности не пытается договориться, не может прийти к согласию, настаивает на своем, конфликтует или игнорирует других	1
	Коммуникативная компетенция	Высокий	Проявляет умение передавать правильно свои мысли, чувства, эмоции.	3

		Средний	Обладает способностью передавать свои мысли и чувства, но иногда требуется внешняя стимуляция.	2
		Низкий	Обладает слабой способностью передавать свои мысли и чувства, постоянно требуется внешняя стимуляция.	1
Предметные	Навыки анализа поставленной задачи по проекту с целью определения необходимой конструкции и алгоритма программирования робота	Высокий	Навыки анализа на высоком уровне. Может проводить анализ самостоятельно, возможно с небольшими замечаниями.	3
		Средний	Навыки анализа на среднем уровне. Может проводить анализ по представленному шаблону, самостоятельно устранять замечания.	2
		Низкий	Навыки анализа на низком уровне. Может проводить анализ только вместе с преподавателем.	1
	Навыков проектирования, сборки и программирования конструкций под задачи проекта	Высокий	Навыки создания готового проекта на высоком уровне. Может выполнять работы самостоятельно, устранять замечания с помощью преподавателя.	3
		Средний	Навыки создания готового проекта на среднем уровне. Может выполнять работы под руководством преподавателя, устранять замечания самостоятельно.	2
		Низкий	Навыки создания готового проекта на низком уровне. Может выполнять работы поэтапно под постоянным контролем преподавателя. Не умеет самостоятельно устранять замечания.	1
	Навыки составления технической документации под задачи проекта	Высокий	Может самостоятельно составлять техническую документацию, устранять замечания.	3
		Средний	Техническую документацию может составлять под руководством педагога, по шаблону, вносить изменения, не обладает навыками самостоятельного устранения замечаний или внесения дополнений.	2
		Низкий	Техническую документацию может составлять под руководством педагога поэтапно по шаблону, не умеет самостоятельно устранять замечания.	1

