***ФГБОУ ВО «Кубанский государственный аграрный университет имени***

***И.Т. Трубилина»,***

***Российская Федерация***

***Колесников Роман Юрьевич, ПИ2102***

***roman563412@gmail.com***

**Видеозанятие Никиты Головина о том, как устанавливать языковые модели на виртуальной машине Google Colab**

**Заголовок**

Запуск языковых моделей (LLM) в Google Colab через веб-интерфейс

**Резюме**

Данное руководство объясняет, как запускать большие языковые модели (LLM) бесплатно в Google Colab, используя веб-интерфейс text-generation-webui, даже при отсутствии мощного локального компьютера.

Основные шаги:

Настройка Google Colab:

Перейдите на colab.research.google.com.

В настройках блокнота ("Изменить" -> "Настройки блокнота") выберите аппаратный ускоритель "TPU v2" для максимальной производительности с нейросетями.

Подключитесь к среде выполнения.

Запуск веб-интерфейса:

Создайте ячейку кода и вставьте предоставленный скрипт.

Скрипт клонирует репозиторий text-generation-webui с GitHub, устанавливает необходимые библиотеки (включая PyTorch) и запускает веб-интерфейс.

Он также использует ngrok для создания публичной ссылки (URL), по которой будет доступен интерфейс.

В процессе установки скрипт запросит тип вашего GPU; для TPU v2 в Colab выберите "N) None".

Дождитесь завершения установки (может занять несколько минут).

Перейдите по сгенерированной ngrok ссылке (вида \*.ngrok-free.app) и нажмите "Visit Site".

Работа с интерфейсом:

Вкладка Model:

Для загрузки модели: перейдите на сайт Hugging Face (huggingface.co), найдите нужную модель (например, Qwen/Qwen1.5-1.8B для быстрого теста или codellama/CodeLlama-7b-Python-hf для Python-кодинга), скопируйте её имя.

Вставьте имя модели в поле "Download model or LoRA" и нажмите "Download". Скачивание происходит на серверах Google и обычно быстрое.

После скачивания обновите список моделей (кнопка с круговой стрелкой), выберите скачанную модель из выпадающего списка.

Выберите загрузчик (обычно "Transformers" для TPU). Настройте параметры (например, cpu, auto-devices) и нажмите "Load" для загрузки модели в память.

Вкладка Parameters: Содержит множество настроек генерации текста (температура, max\_new\_tokens, top\_p, top\_k и т.д.), влияющих на креативность и предсказуемость ответов. Для начала можно оставить значения по умолчанию.

Вкладка Chat: Основная вкладка для взаимодействия.

Режимы: Chat (общение с персонажем), Instruct (строгое следование инструкциям без личности), Chat-Instruct (гибрид).

Character: Настройка личности ИИ (имя, контекст/описание, приветствие). Можно использовать готовых персонажей или создать своего.

User: Настройка вашего имени и описания (личности) для ИИ.

Поле ввода: Введите ваш запрос и нажмите "Generate".

Дополнительные кнопки (ПКМ на кнопке Generate): Regenerate (перегенерировать последний ответ), Continue (продолжить генерацию, если ИИ остановился), Impersonate (ИИ генерирует ответ за вас), Copy/Replace/Remove last reply (редактирование последнего ответа).

Вкладка Training: Предназначена для обучения (fine-tuning) моделей или создания/применения LoRA-адаптеров на своих данных (датасетах). Требует более глубоких знаний.

Руководство демонстрирует базовое взаимодействие с моделью, включая генерацию кода и текста, а также объясняет основные функции интерфейса.

Детальная расшифровка текста

1. Введение и цель

(0:01-0:11) Здравствуйте. Сегодня я обучу вас, как запускать нейросети, большие языковые модели в Google Colab, если у вас недостаточно мощный компьютер для этого, а хочется изучать модели.

2. Настройка Google Colab

2.1. Доступ к Google Colab

(0:12-0:19) Для начала мы заходим в Google Colab. Он доступен по такой вот ссылке: colab.research.google.com.

(0:19-0:25) Либо можно просто написать Google Colab в Гугле, он его найдет сразу же. Это же их сервис, конечно, найдет.

2.2. Подготовка среды

(0:26-0:28) Для начала удаляем вот этот мусор здесь.

(0:28-0:37) Потом мы должны попросить Google подключить нам нормальное железо, потому что сейчас у нас какой-то калькулятор.

2.3. Выбор аппаратного ускорителя (TPU)

(0:37-0:43) Чтобы подключить нормальное железо, мы нажимаем "Изменить". Тут "Настройки блокнота".

(0:43-0:46) И нам нужно выбрать, какая именно у нас будет конфигурация.

(0:47-0:53) Мы выбираем лучше всего ТПУ v2. То есть тензорный вычислительный юнит.

(0:54-1:02) Это конкретно самый мощный юнит для нейросетей. Он специально для них и был разработан. Он бесплатный, ничего не нужно платить, безграничный.

2.4. Подключение к среде выполнения

(1:02-1:06) После подключения, то есть когда тут галочка появляется, всё подключено, мы можем начинать работу.

3. Запуск веб-интерфейса (text-generation-webui)

3.1. Создание ячейки и вставка кода

(1:07-1:10) Мы создаем новую ячейку с кодом, в которую мы вставим определённый код...

(1:11-1:16) ...который у меня есть в этом файле, я его скину.

(1:17-1:24) Этот код я написал, скрипт такой, который полностью автоматически ставит интерфейс и запускает для нейронок.

3.2. Выполнение скрипта установки

(1:25-1:28) Что ж, запускаем этот код на исполнение.

(1:29-1:33) Он забирает репозиторий с GitHub, ставит библиотеки.

(1:34-1:36) Потом он создает вот такую ссылку. Нам нужно по ней перейти.

(1:37-1:44) Но ничего на ней не делать пока что, продолжать смотреть, что происходит в Google Colab. Итак, он у нас работает.

(1:45-1:47) Подождем немного, я поставлю на паузу.

3.3. Выбор типа GPU (для TPU выбрать "None")

(1:48-1:52) После некоторого ожидания он спрашивает, какая у тебя видеокарта.

(1:53-2:00) Если мы используем ТПУ v2, тензорный вычислительный юнит, то мы указываем "none", то есть отсутствует, потому что это не видеокарта, это процессор.

(2:01-2:06) Если же вы там будете дома запускать этот интерфейс, то ставим, например, Nvidia, AMD, там, ну что у вас там стоит.

3.4. Завершение установки

(2:07-2:17) После этого начинается установка всех... короче, всей этой фигни для искусственного интеллекта и машинного обучения, всех этих библиотек в Пайтона.

(2:18-2:22) Это займет большое время, минуты три-пять.

(2:23-2:31) После некоторого времени появляется вот такое предупреждение, что модели вы еще не скачали, и потом появляется данные о том, что всё готово.

3.5. Доступ к веб-интерфейсу через ngrok

(2:31-2:35) После этого мы переключаемся опять на ту вкладку, где мы ссылку открывали, нажимаем "Visit site".

(2:36-2:38) И попадаем на определённый сайт.

(2:39-2:47) Это и есть интерфейс, который сейчас мы запустили на Google Colab. В этом интерфейсе мы будем учиться, как пользоваться языковыми моделями.

4. Работа с веб-интерфейсом

4.1. Вкладка Model: Загрузка моделей

(2:48-2:55) Итак, перейдем к сразу к практике. Чтобы загрузить модель, чтобы с ней чатиться, нужно загрузить её в оперативку.

(2:55-2:57) Для этого мы переходим во вкладку "Model".

(2:58-3:04) Смотрим, какие у нас есть модели. У нас никаких нету, потому что мы только что вообще скачали интерфейс.

(3:05-3:07) Сейчас мы перейдем к скачиванию моделей.

(3:07-3:12) Чтобы скачать модель, мы должны перейти на сайт Hugging Face.

(3:12-3:16) Он доступен по ссылке huggingface.co.

(3:16-3:18) Также легко находится в Гугле.

(3:18-3:36) Там нужно выбрать какую-нибудь модель. Вообще любую, там их сотни, тысячи языковых моделей совершенно разных для разных задач, как программирование, ролевые игры, какое-нибудь написание историй, может быть, докладов, всё там есть. Есть даже большие языковые модели, такие как ChatGPT почти по силе.

(3:37-3:45) Но я для нашего теста выбрал модель Qwen1.5-1.8B. Эта модель очень маленького размера и очень быстро работающая, нам подходит.

(3:46-3:50) Чтобы её скачать, мы вставляем её имя сюда и нажимаем "Download".

(3:51-3:57) Можем смотреть за её скачиванием на... собственно, здесь вот, во вкладке Google Colab.

(3:58-4:04) Она весит всего 3.4 гигабайта, то есть она скачается за 40 секунд. Но я поставлю на паузу.

(4:05-4:14) Также стоит заметить, что вам вообще не важно, какая скорость интернета у вас, важно, чтобы она хоть что-то тянула, потому что скачивание модели всё равно будет очень быстрое.

(4:14-4:21) Ведь на серверах Гугла, с которыми мы сейчас сидим, очень высокая скорость интернета, и модель уже скачалась.

(4:21-4:23) Всё. Теперь мы можем её загрузить.

(4:23-4:27) Для этого мы нажимаем сюда, на этот значок, он означает "обновить".

(4:27-4:29) И мы видим, что у нас модель появилась новая. Вот она.

(4:30-4:32) Теперь мы должны модель загрузить.

(4:33-4:54) Все модели хранятся в разном формате, в разных кодировках, квантованиях, но наша конкретно сейчас модель в формате SafeTensors, безопасные тензоры. Она обычно загружается с помощью либо библиотеки... либо загрузчика ExLlama HF v2, либо Transformers. Но поскольку у нас нет видеокарты, лучше будет через Transformers.

(4:55-5:05) Тут мы указываем такие параметры. Можно поиграться с разными параметрами. Я не буду сейчас объяснять, что означает каждый. Можно загуглить, либо спросить ChatGPT, имеется разная информация.

(5:06-5:12) Потом указываем, сколько нужно ей дать оперативки. Всю оперативку, 300 ГБ, которая у меня есть.

(5:13-5:16) Нажимаем "Load" и модель загружается в нашу оперативку.

(5:17-5:18) Всё, она загрузилась.

4.2. Вкладка Chat: Взаимодействие с моделью

(5:18-5:21) Теперь мы можем начать с ней общаться.

(7:09-7:11) Дальше вкладка "Chat".

(7:11-7:19) Тут это самая важная вкладка для чатинга, потому что мы тут можем указывать персонажа, которым... с которым будет себя ассоциировать искусственный интеллект.

(7:19-7:33) По умолчанию стоит вот этот персонаж. Его описание таково: Диалог с большой обученной языковой моделью. Искусственный интеллект был обучен отвечать на вопросы, давать рекомендации, помогать в принятии решений. Искусственный интеллект следует запросам пользователя и он мыслит нестандартно.

(7:33-7:37) Это нейросеть то, что она будет думать, собственно, о себе. Она будет понимать, кто она, за счет этого.

(7:38-7:43) Тут еще вместе с этим интерфейсом поставляется один персонаж для ролевых игр. Вот такой.

(7:44-7:47) Это Чихару Ямада. Без понятия, кто это.

(7:48-8:07) Но важно то, что это персонаж для ролевых игр, и тут, в принципе, имеется некоторое описание, которое позволит нам генерировать своих персонажей, если мы захотим. Ну, вот тут написано её личность и контекст ролевой игры самой. Ну, в общем, нейросети нужно вот это весь текст, чтобы она могла нормально играть роль.

(8:07-8:15) Дальше вкладка "Greeting". Тут мы указываем, как нейросеть здоровается с нами, то есть сообщение, которое она впервые нам отправляет. Вот таким вот образом.

(8:16-8:22) Дальше "User". Тут мы можем описать наше имя, поставить нашу аватарку.

(8:23-8:38) Либо описать нашу личность. Например, я у нас любитель нейросетей и машинного обучения. И нейросеть будет тогда понимать, кто я такой.

(8:39-8:45) Вот эти вкладки нам не понадобятся. Тут просто мы загружаем чаты, историю или персонажей.

(8:45-8:49) Это можно их можно прямо делать или сразу файлы вот сюда скидывать.

(8:50-8:57) Кстати, где хранятся файлы, сейчас тоже покажу. Вот тут хранятся модели. Тут хранятся персонажи.

(8:58-9:02) А вот тут у нас хранятся датасеты. Ну, это я потом покажу.

(9:08-9:17) Проверим, как же работает сама нейросеть. Мы напишем ей для этого "Hello". Нейросеть спрашивает, как она может нам помочь.

(9:18-9:21) То есть она работает, всё. Значит, она работает.

(9:22-9:24) У нас заготовлен определённый промпт, который мы ей дадим.

(9:25-9:28) Для начала я объясню, что такое режим работы нейросети.

(9:28-9:43) В режиме "Chat" нейросеть обычно общается с тобой как человек, в таком вот простом уровне. Она играет определённого персонажа, который у неё загружен, и всё. Она иногда, конечно, может путать, кто есть кто, она может думать, что она - это ты, и наоборот.

(9:44-9:45) И всё.

(9:45-10:09) Режим "Chat-instruct" позволяет более кастомизировать режим чата. Можно немножечко кастомизировать то, как нейросеть будет себя вести. То есть, например, сейчас она написана: "Продолжите диалог в чате ниже, напишите один ответ для персонажа с определённым именем". То есть нейросеть будет просто отвечать за определённого персонажа. Это нужно, например, такой конкретный промпт нужен, когда нейросеть может путать себя с тобой, например, и ты указываешь ей, типа, за кого писать. Но также ты тут можешь указать, каким должен быть ответ, там, например, более эмоциональным или более простым, если я там ты тупой, тебе нужно, чтобы она проще всё объяснила. Всё это здесь указываешь, и нейросеть будет себя вести немного кастомизируемо.

(10:09-10:10) И самый гениальный режим - это "Instruct".

(10:10-10:16) В этом режиме у нейросети нет никакой личности. Она просто бездушная машина, которая выполняет твои команды и не спрашивает, зачем и что.

(10:36-10:40) Она просто строго следует инструкциям, поэтому он так и называется "Instruct".

(10:41-10:48) И сейчас мы будем испытывать нейросеть именно в этом режиме, потому что в нём она работает очень эффективно, и я хочу показать, какие нейросети бывают эффективные.

(10:49-10:54) Я хочу, чтобы наша нейросеть написала нам код на Пайтоне для рисования спирали Архимеда.

(10:55-11:08) Напиши код на python для рисования спирали архимеда.

(11:09-11:13) Скидываем ей это и ждем, пока нейросеть завершит.

(11:15-11:22) Вот такой код вышел. Почему-то она, блин, неправильно его пишет, потому что она типа не поставила его в кавычки.

(11:22-11:24) Всегда можно нажать "перегенерировать".

(11:31-11:39) Вот я наконец-таки добился, чтобы она написала нормально. Просто эта модель очень маленькая и быстро работающая, она нужна просто для презентации, то есть она тупая.

(11:39-11:58) Есть модели, у которых гораздо более умные. Конкретно сейчас я вижу, что нейросеть просто пишет грубый скрипт, она не делает каких-либо формул в него, она просто пишет, куда двигаться черепашке, чтобы нарисовать его, похоже. Ну, в принципе, там, если она действительно будет рисоваться спираль, несмотря на это, то мне и так нормально. Всё-таки важно, чтобы она типа выполняла именно приказ.

(11:59-12:01) Сейчас мы проверим, как это работает.

(12:03-12:14) Что ж, как мы видим, этот код... этот код не работает. Он, возможно, стилизует спираль, но, по-моему, где-то рисует её за экраном, потому что я ни фига не вижу. Вот действительно не вижу.

(12:14-12:19) Так что мы это не засчитываем нейросети. Значит, она не выполнила эту задачу.

(12:20-12:29) Но это просто из-за того, что эта нейросеть маленькая и нужна для быстрой работы. Она также может выполнять другие инструкции, например...

(12:30-12:37) Напиши историю о котах на Марсе.

(12:39-12:49) И она начнет её генерировать. То есть она не только для кода, конечно же, она для всего. Это универсальная модель. Она спокойно пишет историю, как мы видим.

(12:50-13:08) Более большие модели с большим количеством параметров лучше будут программировать и вообще выполнять любые задачи, но они будут работать медленнее. А вот эта модель имеет 1.8 миллиарда параметров, что очень мало. Да и, в принципе, от неё можно добиться нормальных кодов на Пайтоне, просто нужно много попыток делать и смотреть, что она допускает какие-то ошибки и говорить ей, что надо исправить.

(13:09-13:14) Например, могу посоветовать хорошую модель для кодинга. Сейчас найду.

(13:15-13:18) А вот эта вот модель: codellama/CodeLlama-7b-Python-hf.

(13:19-13:23) Эта модель специально обучена, чтобы понимать и анализировать и писать на Пайтоне.

(13:24-13:34) Собственно, нейросеть тут нам написала уже историю про котов на Марсе, причем довольно не маленькую. Значит, она работает.

(13:34-13:36) Это что насчет чата.

4.3. Дополнительные функции чата (меню ПКМ на кнопке Generate)

(13:36-13:45) Дальше я объясню, как мы можем выполнять различные функции вот в этом менюшке вот этой...

(13:46-13:51) В этой менюшке мы можем давать нейросети какие-то особенные инструкции. Например, "Impersonate".

(13:52-14:02) Эта кнопка позволяет нейросети отвечать за нас. Сейчас она... то есть она написала, что мы могли бы сказать в ответ ей на это сообщение. Что же она сказала? Она сказала: "Я хочу узнать больше о глубоком обучении и нейронных сетях. Что самое важное, что я должен знать?"

(14:07-14:11) Она вот это спрашивает у самой же себя буквально, но только за другого персонажа.

(14:11-14:21) И самое прикольное, что она, скорее всего, это сказала, потому что я вот тут написал, что я любитель нейросетей и машинного обучения. Она сразу же спросила про машинное обучение и нейросети.

(14:21-14:24) То есть она понимает, кто она такая и какая у неё личность.

(14:25-14:34) Что же дальше? "Copy last reply" - эта кнопка копирует ответ нейросети в наше окошко. Тут мы можем, например, что-нибудь поменять.

(14:35-14:40) Просто поменяем одно слово, чтобы для примера. Вот таким вот образом.

(14:40-14:49) Смысл остается такой же, просто по факту оно так работает. И "Replace last reply" заменяет наш ответ, который хранится вот тут, на то, что сказала нейросеть.

(14:49-15:06) Например, если она допустила какую-то ошибку, мы можем таким образом ей заменить всё. Мы копируем её ответ, изменяем в нём ошибку и нажимаем "Replace". И тогда... собственно, изменяется нейросеть, и она видит в контексте, что типа она сказала правильно, но изначально она написала неправильно.

(15:06-15:11) "Remove last reply" - это просто удаляет последнее сообщение. Мы не можем ничего удалить, потому что тут всего одно сообщение.

(15:12-15:16) "Continue" - эта кнопка продолжает... заставляет нейросеть продолжать писать.

(15:16-15:29) То есть, например, если она где-то остановилась... Тут она остановилась, потому что я ей прекратил её генерацию.

(15:26-15:31) Но тут мы можем нажать "Continue", и она продолжит писать.

(15:31-15:39) А всё это потому, что эта кнопка нужна для того, чтобы, если нейросеть вдруг резко остановилась, типа у неё кончились токены или типа того, ты можешь нажать "Продолжить", и всё.

(15:41-15:43) И самая важная кнопка - это "Regenerate".

(15:44-15:52) Если нам что-то не нравится, мы всегда можем нажать "перегенерировать", и мы получим другой ответ.

4.4. Вкладка Parameters: Настройки генерации

(5:22-5:27) Для начала я не буду начинать общаться пока, я сначала покажу, как работает генерация токенов.

(5:28-5:30) Можете перескочить, если вам это не интересно.

(5:30-5:38) Так вот, это все параметры генерации. Я не понимаю ни хрена тут. Тут очень всё сложно.

(5:39-5:53) Это всё разные параметры генерации, которые будут влиять на то, как модель общается. Это, наверное, неважно особо сейчас. Но, в общем, во всяком случае... это нам не нужно обычным пользователям, только прямо каким-то учёным.

(5:54-5:56) Нам важна только вот эта верхняя часть.

(5:56-6:00) Максимальное количество новых токенов - это вот этот вот параметр.

(6:01-6:13) То есть максимальное количество новых токенов, которые нейросеть выдает в одном сообщении. Если это значение увеличить, то нейросеть будет писать более креативные тексты, потому что она может использовать больше разных слов, разных токенов в одном сообщении.

(6:14-6:17) Но также она может с ума сойти из-за этого, если поставить прямо очень много.

(6:17-6:20) Температура - это вероятность появления того или иного токена.

(6:20-6:27) Если температура меньше, то нейросеть будет говорить более предсказуемо, железно, безэмоционально, но эффективно.

(6:27-6:35) А если увеличить, то нейросеть будет говорить более по-человечески, в общем, такие токены генерировать, более рандомные и креативные.

(6:36-6:42) Вот этот параметр отвечает за активацию автоматического количества максимальных токенов, то есть он дает ей столько новых токенов, сколько ей нужно.

(6:43-6:46) Это может сделать ответы более креативными.

(6:46-6:57) Вот этот вот... вот это вот включать не надо ни в коем случае. Это запрещает нейросети генерировать токен, означающий конец текста, то есть она будет генерировать, пока всё, она не усрётся и не кончится оперативка. Это не нужно.

(6:58-7:08) Вот этот вот параметр, если его отключить, нейросеть может выдавать более креативные тексты. Но я его включать не буду, его редко отключают.

(9:03-9:07) В принципе, это и всё, что нам нужно для параметров генерации.

4.5. Вкладка Training: Обучение и оценка моделей

(16:12-16:14) Всё. Теперь следующая вкладка "Training" и последняя.

(16:15-16:34) В этой... в этой вкладке мы обучаем свою нейросеть, либо даем ей Лору. Я, если честно, не знаю, что такое Лора. Но, сколько слышал, это какая-то модель поведения нейросети, которая работает на более низком уровне, чем персонаж. Прямо изменяет её личность и её поведение. Но я не умею это делать, так что я ничего не могу про это сказать.

(16:35-16:44) Могу сказать, что здесь делаются Лоры и изменяются, но я не делал такого. Тут есть даже кнопка "Tutorial". Вот нажмите, можете почитать, как это делается.

(16:45-16:54) Но что я знаю, так это то, что тут ей можно скармливать различные форматированные датасеты. То есть мы выбираем формат данных, который тут есть разные.

(16:54-16:59) Выбираем сам датасет. Его для этого нужно закинуть вот сюда, "Training", "Datasets".

(17:00-17:09) И промпт... э... эволюции датасета и шаги для эволюции. И нажимаем "Start LoRA Training", и нейросеть начинает обучаться на этом датасете.

(17:10-17:12) Так это работает.

(17:12-17:25) Дальше какая-то вот штука означает "perplexity", не знаю. Сейчас переведу. Недоумение. Ладно, не знаю.

(17:26-17:31) Короче, это, наверное, эволюция модели, так скажем, evaluation, эволюция перехода в следующий шаг.

(17:31-17:38) Тут мы выбираем модель какую-нибудь. Нажимаем датасет, которым мы хотим её тренировать. Тут уже есть встроенных три датасета.

(17:39-17:41) Например, WikiText, Википедия.

(17:41-17:44) Тут указываем два параметра, а именно "stride" и "max\_length".

(17:44-18:08) "Stride" - это то, насколько она внимательно учится. То есть мы можем учиться как какая-нибудь отличница, запоминая каждое слово по три раза. И она идеально будет тогда следовать датасету. Ну, он просто отлично выучит, но только нужно будет очень-очень много времени. Или мы можем учиться как я и как вы, с максимальной скоростью, ничего не запоминая. Может быть, максимум только самое важное. Это сильно увеличивает скорость.

(18:09-18:11) Ну, базовое значение - это 512.

(18:12-18:22) А вот эта вот длина обозначает длину контекста. Если у нас нейросеть будет с маленькой длиной контекста жрать, то она будет запоминать этот датасет маленькими кусками.

(18:23-18:31) Например, то есть как мы можем... мы можем типа учить сразу по несколько страниц, а мы можем на каждом предложении прерываться, чуть-чуть отдыхать и дальше учить.

(18:32-18:39) Ну, собственно, вы сами решаете, как вам удобнее. Если мы поставим ноль, то нейросеть будет просто максимально долго читать, как она может.

(18:41-18:45) Дальше мы можем нажать "Evaluate loaded model", и она начнет обучать модель.

(18:46-18:54) Потом тут укажется еще, сколько процентов. Вот, загрузка модели, вот процессинг, evaluating, нейросеть эволюционирует, 0% закончено.

(18:55-18:57) Чтобы прекратить, нажимаем "Interrupt".

(18:58-19:00) Это вот как работает обучение.

5. Заключение

(19:01-19:03) В принципе, все основы этого интерфейса.

(19:04-19:12) Остальное можно либо найти в интернете, либо спросить ChatGPT, либо как-то еще нарыть. Это всё, все основы, которые нужно было знать.

(19:13-19:29) Также хочу сказать, что этот интерфейс очень неплохо работает на мощных компьютерах локально. Он даже лучше работает, чем на Google серверах. Но просто у меня компьютер-то достаточно мощный, но просто не уверен, что у вас, поэтому и говорю, как нужно его использовать на Google серверах именно.

(19:30-19:42) Это конец нашего урока. Теперь вы знаете, как примерно пользоваться этими нейронками. Ну и всё. До свидания. Удачной генерации токенов.