

Развертывание и настройка собственной языковой модели нейронной сети на локальной машине

Звайгзне Алексей Юрьевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье описывается процесс развёртывания локальной версии языковой модели нейронной сети. Для реализации используются инструментальные программные средства LM studio, подобраны несколько языковых моделей нейронных сетей, распространяемых бесплатно. Проведена проверка на ликвидность выдаваемых ответов языковыми моделями.

Ключевые слова: LM studio, нейронные сети, языковая модель, mistral

Deploying and configuring your own neural network language model on a local machine

Zvaigzne Alexey Yurievich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article describes the process of deploying a local version of the neural network language model. For implementation, LM studio software tools are used, several language models of neural networks distributed for free have been selected. The liquidity of the responses provided by the language models was checked.

Keywords: LM studio, neural networks, language model, mistral

1 Введение

1.1 Актуальность

Развёртывание и конфигурация индивидуальных языковых моделей нейронной сети на локальных вычислительных ресурсах продолжают представлять актуальное и важное направление в области научных исследований и разработок в области искусственного интеллекта. Ряд основополагающих факторов подчёркивают важность возможности создания и тонкой настройки собственных языковых моделей. В контексте языковых моделей, учитывающих индивидуальные потребности и специфику конкретных организаций и исследовательских задач, локальное развертывание предоставляет уникальные преимущества. Важным аспектом становится обеспечение удобства управления и хранения собственных данных, особенно в случаях, когда безопасность данных является высшим приоритетом. Кроме того, локальное развертывание предоставляет полный

контроль над инфраструктурой, что важно при стремлении к высокой степени индивидуализации или использовании специфических аппаратных ресурсов. Этот подход также способствует более эффективным экспериментам и обучению моделей на ограниченных объемах данных, ускоряя цикл разработки и позволяя оперативно внедрять улучшения. С учётом возможности работы оффлайн и обеспечения экономической эффективности для небольших исследовательских групп или стартапов, локальное развертывание выступает важным фактором. Контроль над производительностью при оперировании крупными моделями или обработке больших объёмов данных предоставляется более эффективно при локальном развертывании.

1.2 Обзор исследований

В своей статье А.К. Крутиков описали процесс разработки в среде MATLAB прототипа программной системы прогнозирования спортивных результатов, основанных на нейросетевом подходе [1]. И.С. Кипяткова в статье «Разновидности глубоких искусственных нейронных сетей для систем распознавания речи» рассмотрели гибридный и тандемный подходы объединения скрытых марковских моделей и искусственных нейронных сетей для акустического моделирования, описали построение языковых моделей с применением сетей прямого распространения и рекуррентных нейросетей [2]. В.В. Котенко в своей рассмотрели ключевые программные продукты с использованием технологий машинного перевода и искусственного интеллекта, рассмотрена их эффективность и потенциал применения в контексте развития открытого образования [3]. В своей работе Г.А. Однооконный рассмотрел чат GPT как цифровую технологию современности, его влияние на жизнь и деятельность людей, привел примеры использования нейросети в бизнесе, определил направления дальнейшего развития событий относительно данной темы [4].

1.3 Цель исследования

Целью исследования является – развертывание языковой модели на локальной машине с использованием LM studio [5], которое позволит без подключения к сети Интернет взаимодействовать с языковой моделью для решения рабочих вопросов.

2 Материалы и методы

Процесс подключения языковой модели нейронной сети произведен в среде работы с языковыми моделями нейронных сетей LM studio, выбраны, установлены и настроены языковые модели Mistral 7B Instruct v0.1.Q4 К М и LLAMA 2 7B chat.Q4 К М для сравнения. Также для быстрой и корректной работы моделей потребуется мощная вычислительная техника, в рамках статьи для работы моделей будет использоваться графический ускоритель Nvidia 3070 и 16 ГБ оперативной памяти.

3 Результаты и обсуждения

Для начала работы скачивается и устанавливается средство работы с языковыми моделями нейронных сетей LM studio с официального сайта (рис. 1) [6]. Данное программное обеспечение позволяет скачивать для работы языковые модели нейронных сетей с Hugging Face [7] в виде контейнеров формата gguf.

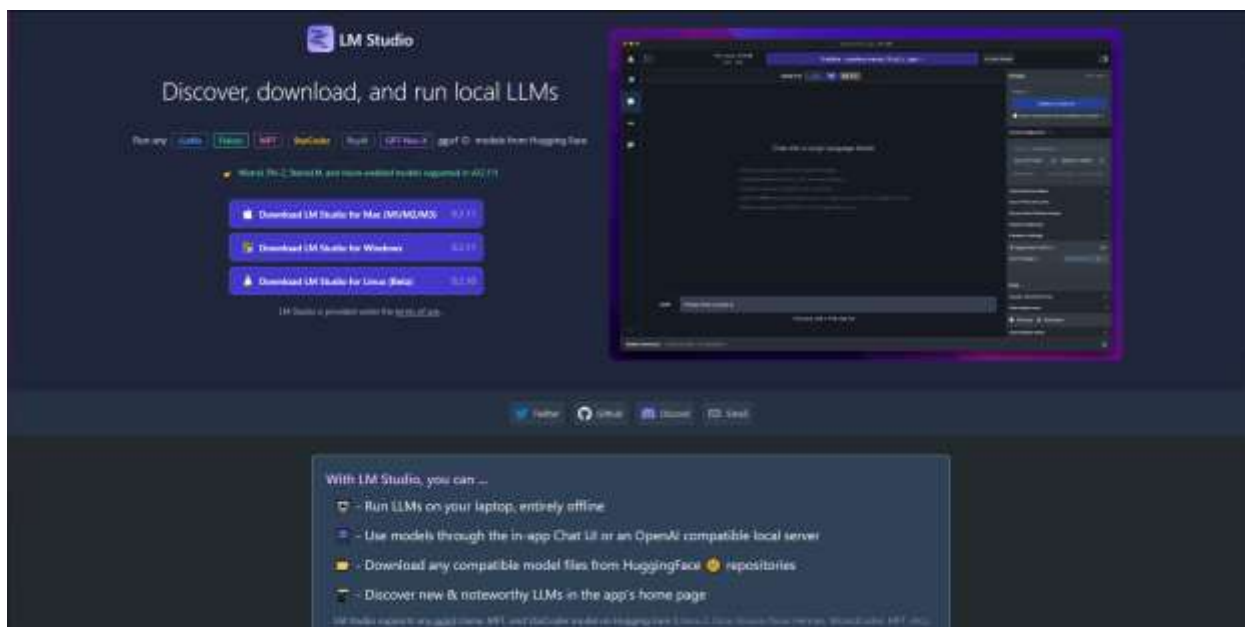


Рисунок 1. Официальный сайт LM studio

Выбрав версию для определенной операционной системы начнется загрузка установочного файла. Загрузив файл производится стандартная процедура установки. После становки программа запускается с помощью ярлыка на рабочем столе. При запуске программы будет отображена вкладка с поиском и установкой предварительно обученных различных типов языковых моделей нейронных сетей от разных авторов размещенных на Hugging face (рис. 2).

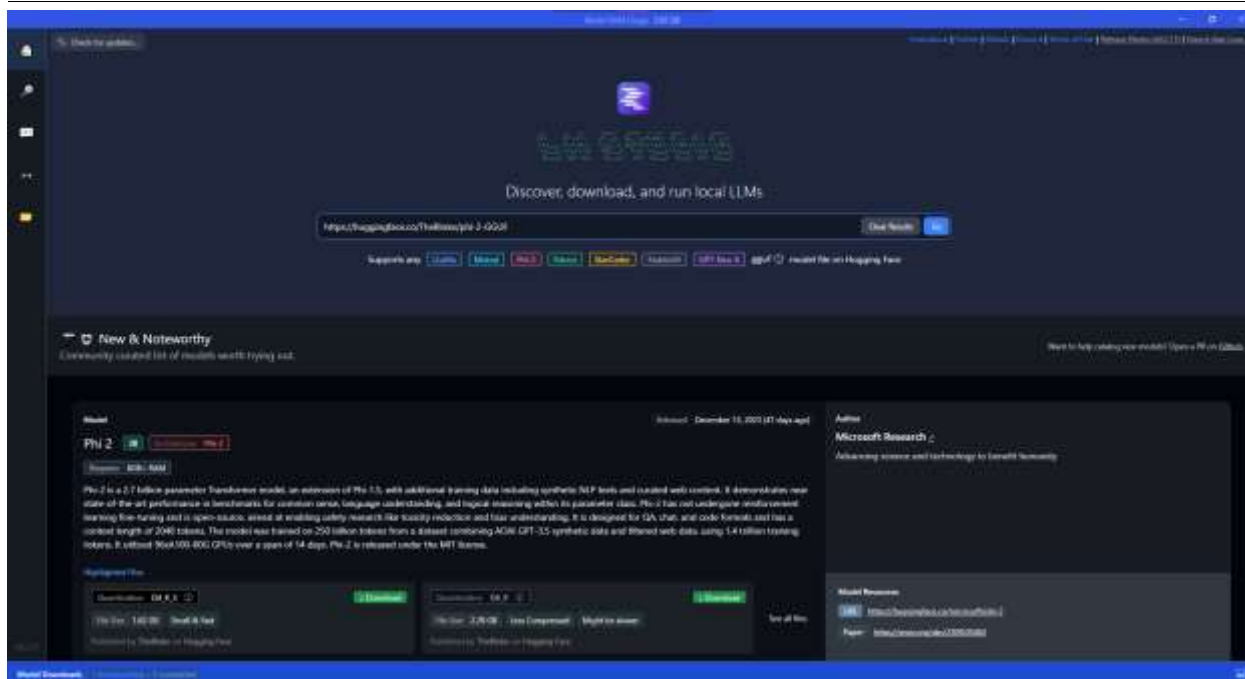


Рисунок 2. Главный экран LM studio

В рамках статьи были выбраны две языковых модели нейронных сетей это `Llama-2-7b-chat.Q4_K_M.gguf` (рис. 3) [8] и `mistral-7b-instruct-v0.1.Q4_K_M.gguf` (рис. 4) [9] от одного автора под ником Tom Jobbins [10]. В основу обеих языковых моделей взята архитектура LLa. Данные модели можно найти через стандартный фильтр поиска в программе на главном экране, есть версии на 7, 13 и 70 миллиардов входных параметров, размерность весов модели определяет точность ответов нейронных сетей. Модели на 70 миллиардов входных параметров обладают Grouped-Query Attention (GQA) методом позволяющим ускорять получение ответа конечному пользователю, но используют больше вычислительных ресурсов устройства. Если для большинства моделей на 7 миллиардов входных параметров достаточно от 3 до 5 Гб встроенной памяти и от 5 до 7 Гб оперативной памяти (в зависимости от настроек и моделей), то для более мощных моделей потребуется уже в разы больше вычислительных ресурсов (рис. 5). Также стоит обратить внимание на то, что если выбран графический ускоритель для работы, то модель будет загружаться в оперативную память видеокарты. При недостаточном объеме оперативной памяти при попытке загрузки модели будет выдаваться ошибка о нехватке памяти (out of memory).

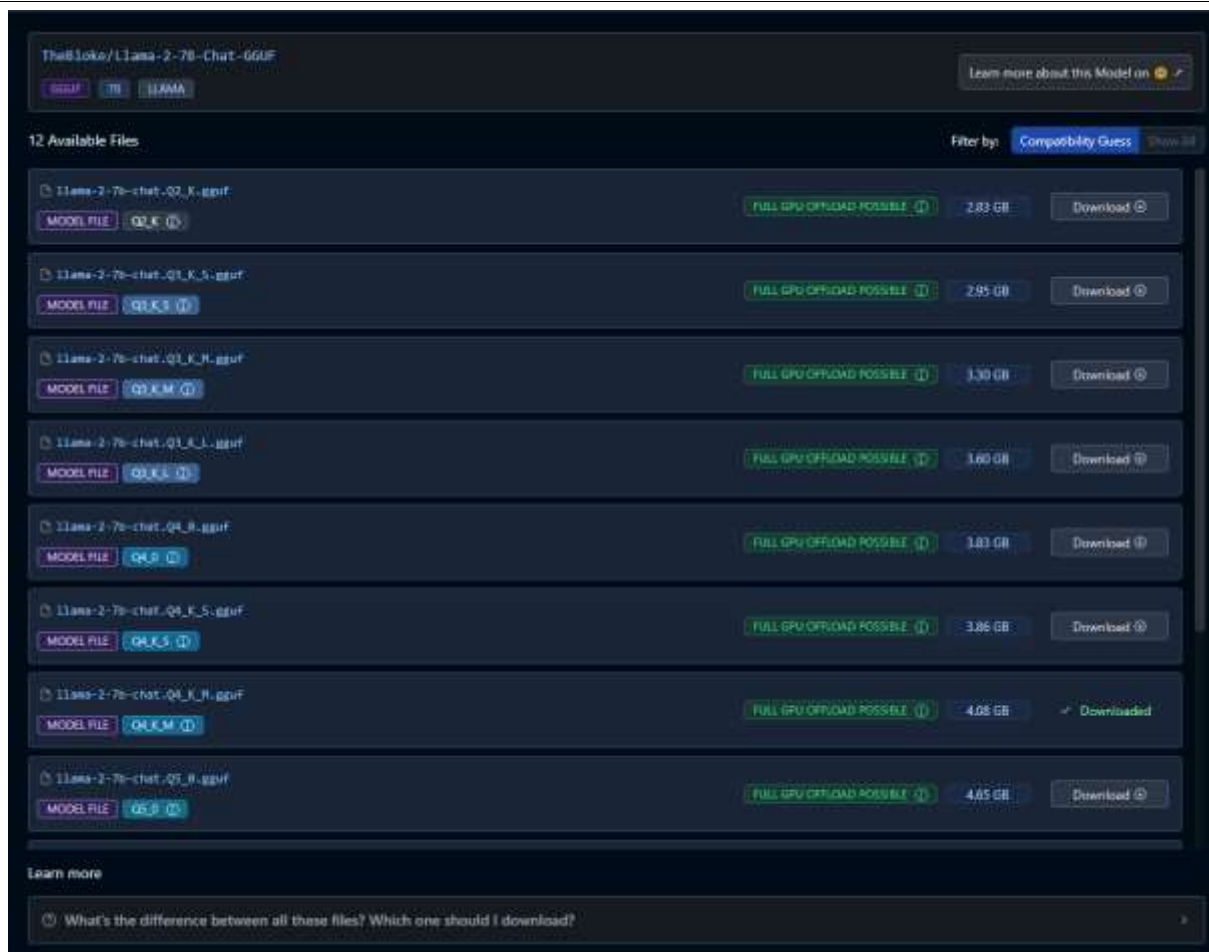


Рисунок 3. LLaMA 2 7b доступные языковые модели для загрузки

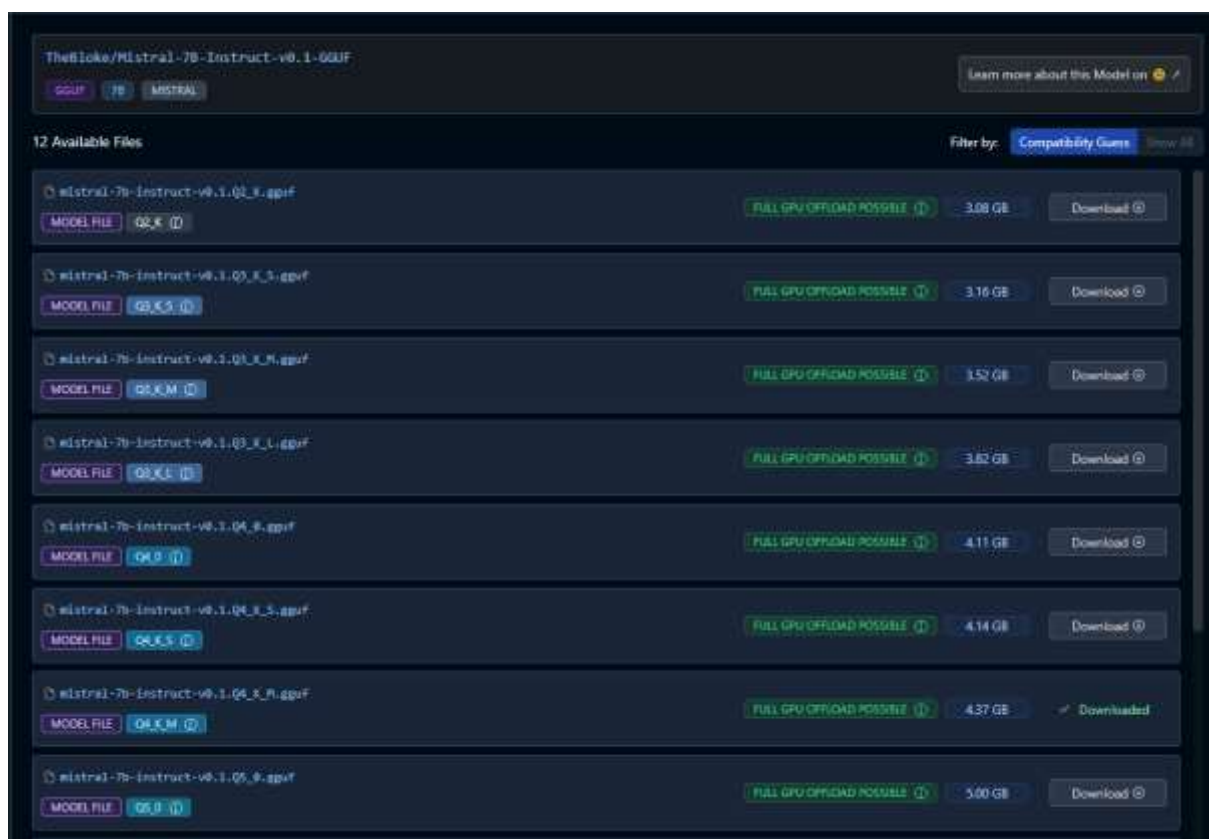


Рисунок 4. Mistral 7b доступные языковые модели для загрузки

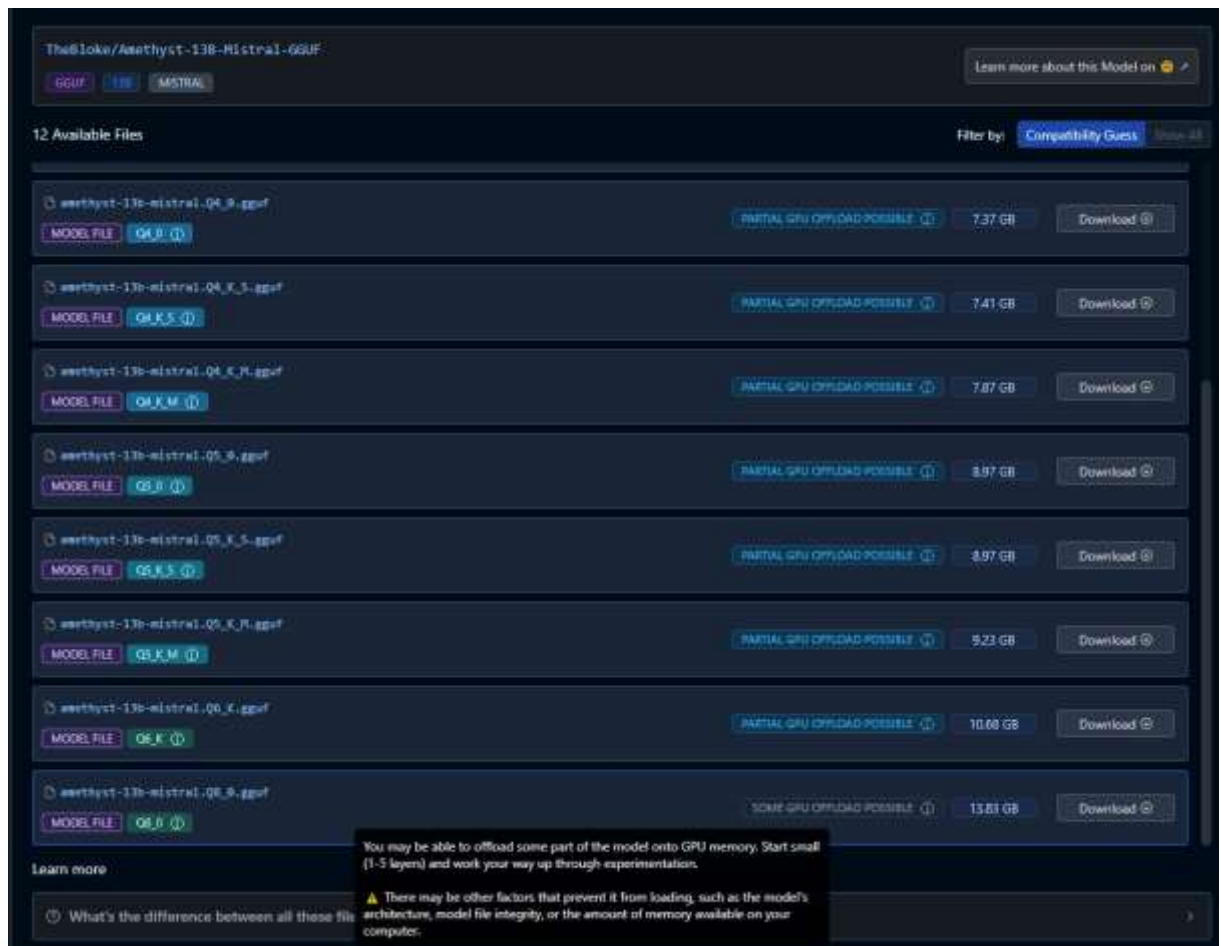


Рисунок 5. Пример языковой модели на 13 миллиардов входных данных

После выбора модели можно выбрать битность квантования. Данный параметр моделей отвечает за точность при обработке данных, уменьшение битности модели увеличивает пропускную способность, снижает расходы на электроэнергию, а также сокращает выделяемое тепло при обработке запроса, но уменьшает точность полученных данных. В рамках статьи обе модели были выбраны с одинаковой битностью – 4 бита квантования.

После установки моделей в левой части экрана выбрать раздел AI chat (рис. 6).

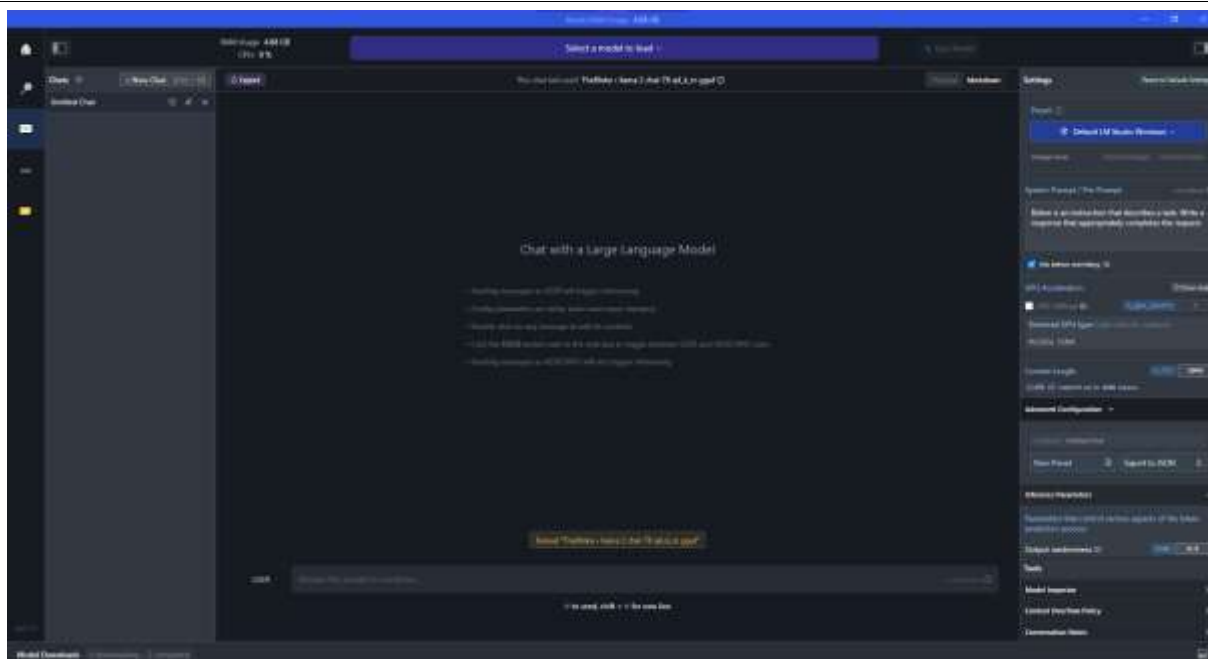


Рисунок 6. Окно AI chat

Здесь в элементе *Select a model to load*, выбрать нужную языковую модель. Сразу после выбора модели произойдет загрузка модели в оперативную память, но перед началом использования модели необходимо изменить параметры в правой части экрана на панели *Settings* для упрощения взаимодействия с моделью, путем указания первоначального запроса (*Prompt*), настройкой подключения графического ускорителя для ускорения работы нейронной сети с указанием количества слоев модели, указать длину контекста токена нейронной сети, данный параметр отвечает за объем выдаваемой текстовой информации (рис. 7). Для разных языков полученное значение токенов отличается, к примеру ответ на русском языке будет содержать в разы больше токенов в отличие от англоязычного ответа (рис. 8).



Рисунок 7. Пример настройки LLaMA 2



Рисунок 8. Пример вывода нейронных сетей на разных языках с подсчетом токенов

После настройки модели, произвести перезагрузку модели для применения параметров (рис. 9).

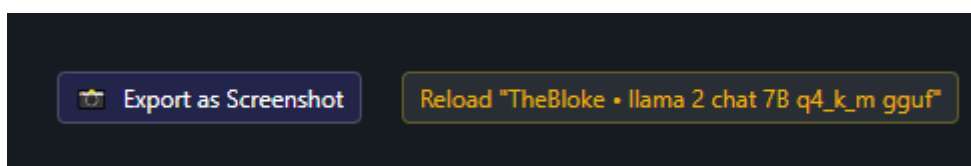


Рисунок 9. Кнопка перезагрузки модели

После перезагрузки модели можно начинать работать с нейронной сетью через чат (рис. 10).

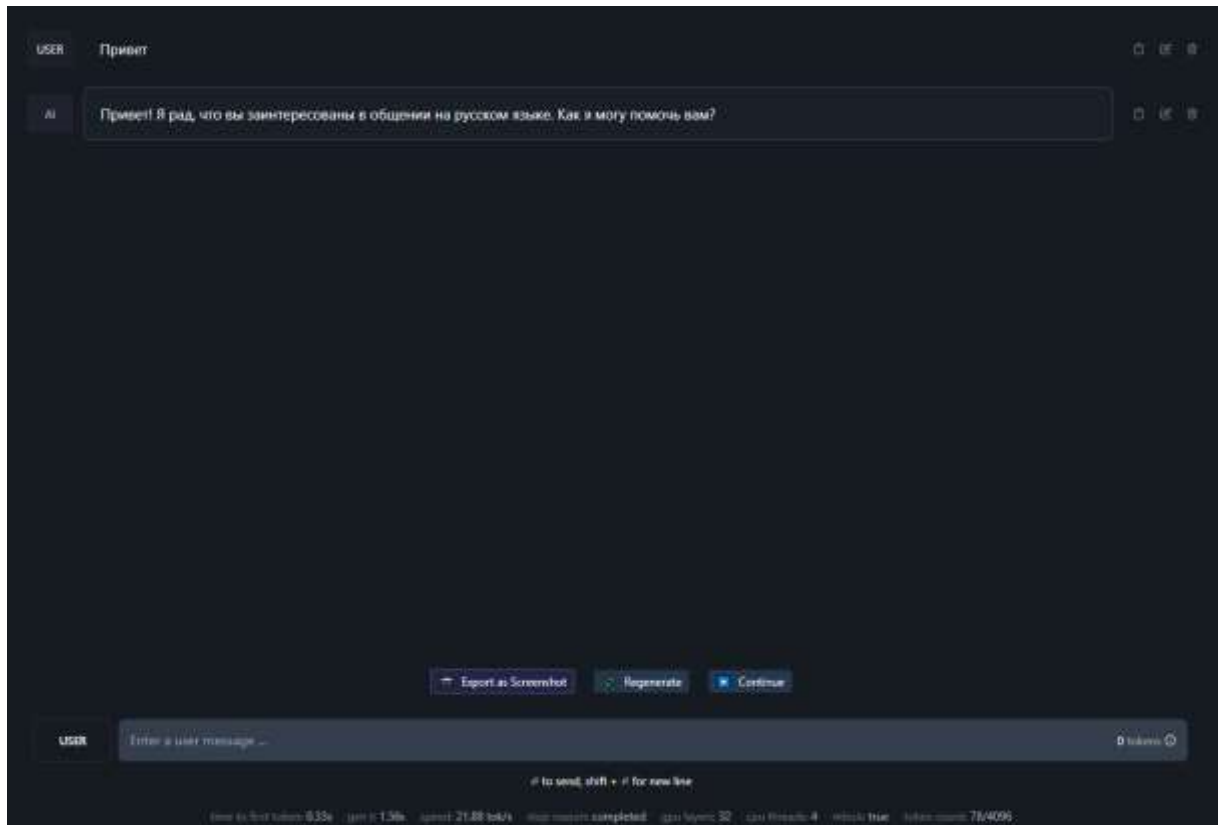


Рисунок 10. Пример работы LLaMA 2

Для сравнения двух разных моделей нейронных сетей будут установлены одинаковые настройки и заданы одинаковые запросы.

Так для обеих моделей был задан вопрос: Приведи пример SQL запроса на создание таблиц для ИС автомойки.

Вывод языковой модели нейронной сети LLaMA 2:

Для создания таблиц для ИС автомойки можно использовать SQL-запросы, которые позволят создавать таблицы и выполнять различные операции с ними. Ниже приведен пример SQL-запроса на создание таблиц для ИС автомойки:

```
...  
-- Создаем таблицу "автоматические_заправки"  
CREATE TABLE автоматические_заправки (  
  id INT PRIMARY KEY,  
  номер_заправки INT,  
  кол-во_литров DECIMAL(10, 2),  
  стоимость DECIMAL(10, 2)  
);  
  
-- Создаем таблицу "заказы"  
CREATE TABLE заказы (  
  id INT PRIMARY KEY,  
  идентификатор_заправки INT,
```

```
    дата_заказа DATE,  
    время_заказа TIME,  
    номер_заправки INT,  
    кол-во_литров DECIMAL(10, 2),  
    стоимость DECIMAL(10, 2)  
);  
  
-- Создаем таблицу "завершения"  
CREATE TABLE завершения (  
    id INT PRIMARY KEY,  
    идентификатор_заказа INT,  
    дата_завершения DATE,  
    время_завершения TIME,  
    количество_литров DECIMAL(10, 2),  
    стоимость DECIMAL(10, 2)  
);  
  
-- Создаем таблицу "статус"  
CREATE TABLE статус (  
    id INT PRIMARY KEY,  
    идентификатор_заказа INT,  
    дата_статуса DATE,  
    время_статуса TIME,  
    status_text VARCHAR(50)  
);  
  
-- Создаем индекс на таблице "автоматические заправки" для быстрого  
поиска номеров заправки  
CREATE INDEX autmoatic_fill ON автоматические_заправки  
(номер_заправки);  
  
-- Создаем индекс на таблице "заказы" для быстрого поиска  
идентификатора заказа и номера заправки  
CREATE INDEX заказ_fill ON заказ (idenitificator_заказа, номер_заправки);  
  
-- Создаем индекс на таблице "завершения" для быстрого поиска даты и  
времени заключения и номера заправки  
CREATE INDEX finish_fill ON завершения (дата_завершения,  
время_завершения, номер_заправки);  
````
```

В этом примере создаются следующие таблицы:

- \* `автоматические\_заправки` - таблица для хранения информации о заправочных операциях, включая номер заправки, количество литров и стоимость.

\* `заказ` - таблица для хранения информации о заказах, включая идентификатор заказа, дата и время заказа, номер заправки и количество литров.

\* `завершения` - таблица для хранения информации о заключениях, включая идентификатор заказа, дата и время заключения, количество литров и стоимость.

\* `статус` - таблица для хранения информации о состоянии заказа, включая идентификатор заказа, дата и время состояния и текст статуса. Кроме того, создаются индексы на каждой из этих таблиц для быстрого поиска соответствующих столбцов.

Это лишь пример SQL-запросов для создания таблиц и индексов для ИС автомойки, и вам может потребоваться дополнительная обработка данных и настройка параметров в зависимости от ваших нужд.

### **Вывод языковой модели нейронной сети Mistral:**

Создать таблицы для ИС автомойки можно с помощью следующих SQL-запросов:

```

-- Создаем таблицу автоматические_заправки

```
CREATE TABLE автоматические_заправки (  
  id INT PRIMARY KEY,  
  номера_заправки INT,  
  кол-во_литров DECIMAL(10, 2),  
  стоимость DECIMAL(10, 2)  
);
```

-- Создаем таблицу заказы

```
CREATE TABLE заказы (  
  id INT PRIMARY KEY,  
  идентификатор_заправки INT,  
  дата_заказа DATE,  
  время_заказа TIME,  
  номера_заправки INT,  
  кол-во_литров DECIMAL(10, 2),  
  стоимость DECIMAL(10, 2)  
);
```

-- Создаем таблицу завершения

```
CREATE TABLE завершения (  
  id INT PRIMARY KEY,  
  идентификатор_заказа INT,  
  дата_завершения DATE,  
  время_завершения TIME,  
  кол-во_литров DECIMAL(10, 2),
```

```
стоимость DECIMAL(10, 2)
);

-- Создаем таблицу статус
CREATE TABLE статус (
  id INT PRIMARY KEY,
  идентификатор_заказа INT,
  дата_статуса DATE,
  время_статуса TIME,
  статус_текст VARCHAR(50)
);
...
```

Эта коллекция SQL-запросов создаёт таблицы для хранения информации о заправочных операциях, включая номер заправки, количество литров и стоимость, а также информацию об заказах, заключениях и их статусе.

В обоих случаях было создано четыре таблицы с одинаковыми названиями, параметрами и ошибками оформления самих таблиц, а именно название столбцов выполнены на русском, но в варианте ответа от LLaMA 2 есть ошибки описания в последнем абзаце хоть и более детально описаны сгенерированные таблицы, когда как Mistral дала лишь общую характеристику проделанной работы.

Выводы

Данные языковые модели нейронных сетей вряд ли смогут конкурировать с более масштабными проектами от различных компаний на данном этапе. Большую эффективность данные модели смогут показать в частных организациях, где нет доступа к сети Интернет для сохранения конфиденциальности разрабатываемого продукта и избежание возможных утечек, связанных с этим. Дополнительно дообучив языковую модель литературой можно получить индивидуального помощника в написании кода в домашних условиях, не прибегая к взаимодействию с другими популярными языковыми моделями нейронных сетей сохраняя анонимность.

Библиографический список

1. Крутиков А. К. Особенности обучения нейронной сети векторного квантования при разработке специализированного программного модуля //Южно-Сибирский научный вестник. 2019. №. 2. С. 150.
2. Кипяткова И. С., Карпов А. А. Разновидности глубоких искусственных нейронных сетей для систем распознавания речи //Информатика и автоматизация. 2016. Т. 6. №. 49. С. 80-103.
3. Котенко В. В. Перспективы развития нейронного машинного перевода в контексте концепции открытого образования //Ученые записки университета им. ПФ Лесгафта. 2020. №. 4 (182). С. 225-230.

4. Одноконный Г. А., Рогач С. И. Влияние чата GPT на развитие современного общества и бизнеса //Проблемы современного социума глазами молодых исследователей-XV. 2023. С. 363-364.
5. LM Studio URL: <https://lmstudio.ai>
6. Hugging Face URL: <https://huggingface.co/>
7. TheBloke/Llama-2-7B-Chat-GGUF URL:
<https://huggingface.co/TheBloke/Llama-2-7B-Chat-GGUF>
8. TheBloke/Mistral-7B-Instruct-v0.1-GGUF URL:
<https://huggingface.co/TheBloke/Mistral-7B-Instruct-v0.1-GGUF>