Отчет по домашнему заданию 3 по курсу NLP 2024

по теме «Обучение LLaMa»

Выполнил:

Студент 4 курса

группы БПМИ211

НИУ ВШЭ ФКН

Торбахов Тимофей

Москва, 2024

Оглавление

[Введение 2](#_Toc182229843)

[Написание архитектуры 2](#_Toc182229844)

[Выбор функции обучения 2](#_Toc182229845)

[Первый запуск обучения 2](#_Toc182229846)

[Обучение на датасфере 3](#_Toc182229847)

[Детали обучения 3](#_Toc182229848)

[Результаты обучения 3](#_Toc182229849)

[Приложения 4](#_Toc182229850)

# Введение

В процессе выполнение домашнего задания мною была прочитана статья “[LLaMA: Open and Efficient Foundation Language Models”](https://arxiv.org/pdf/2302.13971). Далее используя архитектуру описанную в статье, я воспроизвёл реализацию и обучил эту модель на предложенном корпусе данных – урезанном OpenWebText. Опишу детальнее процесс реализации.

# Написание архитектуры

Для начала я создал питоновский проект, разбивая код насмысловые части, реализовывая разные слои трансфомера в разных файлах, самым сложным оказалось реализовать Rotary Position Encoding, а имеено понять, как он работает и применить. После долгих часов написания кода и отладки у меня получилось завести модель на данных одного батча.

# Выбор функции обучения

Далее встал вопрос на какой лосс и как обучать модель, потому что у нас нет таргетных данных. Тогда я решил предсказывать следующий токен для каждого токена и обучаться на Кросс Энтропию.

# Первый запуск обучения

Даже он довольно долго обрабатывался локально на ЦПУ, поэтому я принял решение перейти в сервис Kaggle и попробовать обучить мою модель на ГПУ. К сожалению, предоставленный там ускоритель не позволял обучить модель с запрашиваемыми условия, поэтому я снизил гиперпараметры таким образом, что она ей хватило место. А именно embedding\_dim=256(размерность скрытых представлений для каждого токена), batch\_size=4, max\_seq\_len=256(длина последовательности в батче (на вход приходит batch\_sizeтекстов, они переводятся в один длинный текст, далее нарезаются на max\_seq\_len тексты, таких текстов берётся batch size, при необходимости добавляются паддинги в последней текст)), n\_layers=3(количество слоёв трансформера). Обучаемых параметров там было 40 миллионов, используя оптимизатор из статьи, а именно ADAMW с learning rate 3e-4, betas (0.9, 0.95), weight decay=0.1. Таким образом я понял, что архитектура написана правильно, потому что loss падал.

# Обучение на датасфере

Далее я дождался, когда выдали датасферу для обучения полной модели, которая удовлетворяла бы выданному заданию. Я перешёл из проекта в ноутбук, потому что так оказалось удобнее работать в датасфере. С помощью ускорителя V100, я смог обучить архитектуру со следующими гиперпараметрами embedding\_dim=768, max\_seq\_len=1024, n\_layers=8, batch\_size=8.

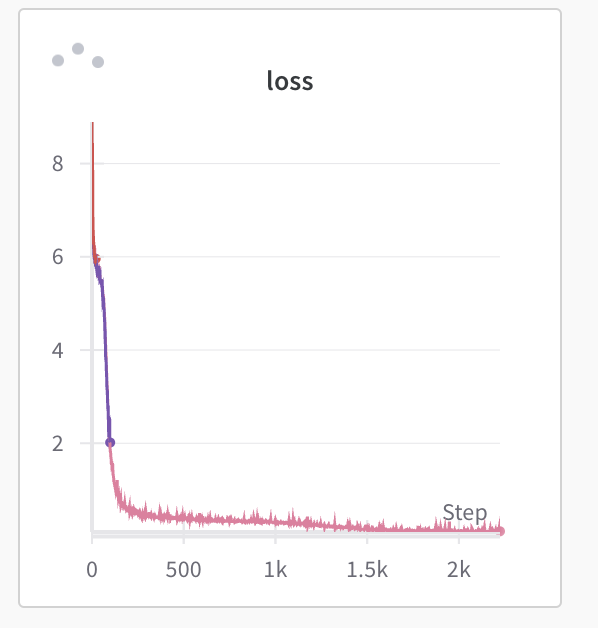
# Детали обучения

Поскольку с помощью предобученного токенайзера предложенного в тексте домашнего задания тексты токенизируются долго, я принял решения обучать текст на частях по 100000 текстов (10% всей выборки). Соответственно я поставил обучение 10 эпох (1 раз прошёлся по каждому элементу датасета). В процессе обучения я логировал данные о лоссе и сохранял модель в сервис W&B, чтобы не потерять данные в случае проблем на датасфере. К сожалению, мне не удалось поставить обучение в фоновом режиме, поэтому пришлось обучать модель неудобно, не выключая ноутбук. Ниже прилагаю график лосса.

# Результаты обучения

С помощью написанной мной реализации LLAMA мне удалось добиться схождения модели по лоссу и добиться качества в 0.1 CrossEntropy. К сожалению, мне не удалось разобраться как зарегистрировать модель на hugging face, потому что не хватило времени, поэтому я не смог получить других метрик для оценки качества модели.

# Приложения



[Ссылка](https://drive.google.com/file/d/1b907F-r1xHMQWkNif23GB_1HuSJi6dOz/view?usp=sharing) на веса обученной модели