Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных систем

Щукин Илья Вячеславович

# Реализация адаптера на основе mixture of experts для дообучения с учителем

Отчёт по учебной практике

Научный руководитель:
ассистент кафедры ИАС Чернышев Г. А.

Санкт-Петербург
2024

# Оглавление

# 1 Введение

Дообучение с учителем является обязательной частью обучения больших языковых моделей. После обучения модели на большом корпусе текстов без учителя следует дообучение на размеченных даных. Современные большие языковые модели могут иметь сотни миллиардов параметров, что делает этот этап, где может потребоваться множество дополнительных экспериментов, довольно дорогим. Для того, чтобы сократить необходимое время для дообучения используются различные подходы, одним из которых является применение адаптеров. Адаптеры представляют собой дополнительные наборы параметров для моделей, которые обучаются в процессе дообучения, при этом оригинальные параметры моделей не изменяются.

Метод mixture of experts [7] позволил обучение больших моделей за счёт использования условного вычисления активаций. Этот метод позволяет распределять токены между отдельными частями сети — экспертам. Это позволяет существенно увеличить число параметров модели без значительного увеличения времени необходимого для обучения и предсказаний. Данный метод позволяет использовать в качестве эксперта любую нейронную сеть, поэтому в качестве экспертов можно использовать LoRA [6]. Использование LoRA в качестве экспертов позволит увеличить число параметров адаптера и при этом не значительно увеличить стоимость дообучения.

В данной работе описывается разработка адаптера для дообучения больших языковых моделей с применением mixture of experts и LoRA. Также описаны эксперименты показывающие потенциал для данного метода.

# 2 Постановка задачи

Целью данной работы является реализация нового адаптера на основе mixture of experts и LoRA. Для её достижения были сформулированы следующие задачи:

- Выполнить обзор предметной области.
- Реализовать адаптер с новой архитектурой на основе mixture of experts и LoRA.
- Провести эксперименты, измерить качество дообученой модели на интересующих бенчмарках. Сравнить с обычным дообучением.

# 3 Обзор

## 3.1 LoRA

LoRA или Low-Rank Adaptation [3] — это метод дообучения при котором веса исходной модели не обучаются, и в каждый блок трансформеров добавляются адаптеры в виде факторизаций матриц меньшего ранга. Для матрицы $W_0 \in \mathbb{R}^{d \times k}$ обозначим как $\Delta W$ её изменение после дообучения и представим его в виде произведения матриц $B \in \mathbb{R}^{d \times r}$ и $A \in \mathbb{R}^{r \times k}$, где $r$ значительно меньше $d$ и $k$. Тогда вычисление результирующего вектора можно представить вот так $h = W_0 x + \Delta W x = W_0 x + BAx$. На рисунке 1 представлена параметризация, используемая в LoRA.

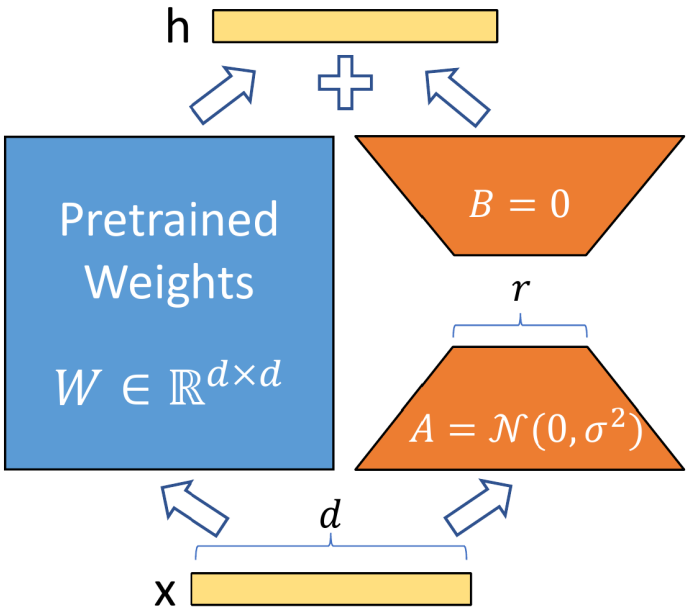


**Рис. 1:** LoRA адаптер. Изображение взято из [3].

Данный метод позволяет существенно уменьшить число обучаемых параметров и как следствие потребление памяти на этапе дообучения и значительно ускорить процесс обучения. В сценариях с ограниченными ресурсами LoRA может быть единственным доступным способом дообучения. К сожалению, часто на практике LoRA даёт результаты хуже полного дообучения [4].

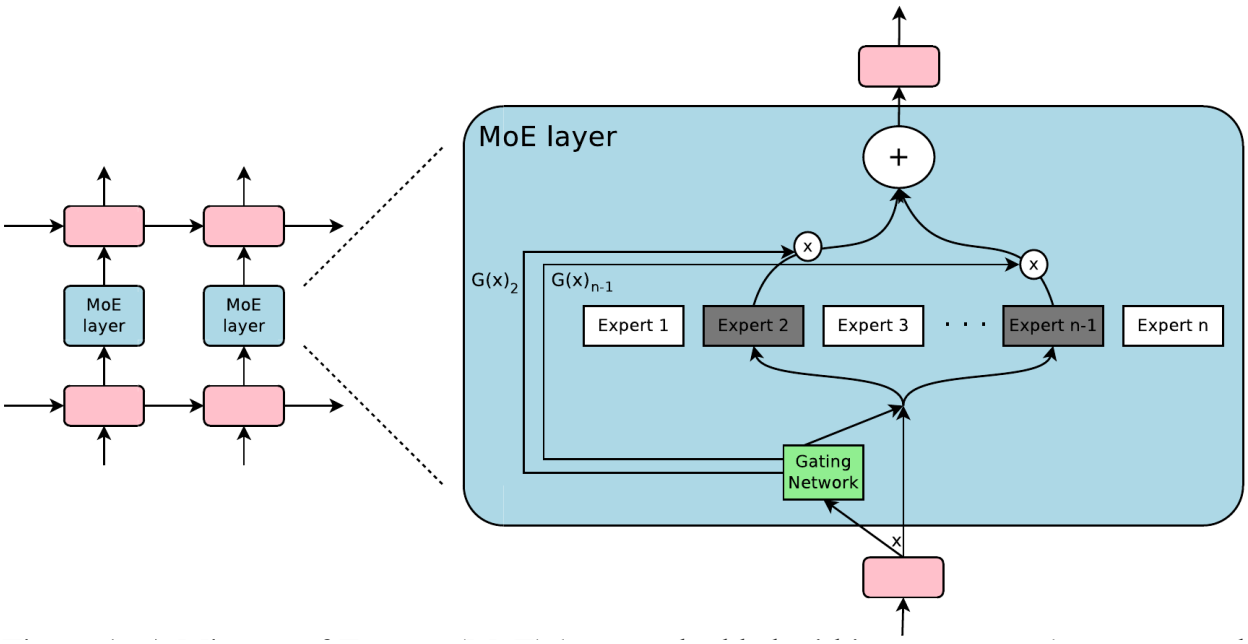


**Рис. 2:** Mixture of experts слой между LSTM слоями. Изображение взято из [7].

## 3.2 Mixture of experts

Mixture of experts — это особое архитектурное решение, которое предполагает наличие слоёв-экспертов и слоя-роутера. Роутер обрабатывает входные данные и распределяет их между обработчиками-экспертами с весами-оценками того, насколько данные подходят конкретному эксперту. Эксперты могут представлять собой произвольные нейронные сети, но чаще всего для языковых моделей они представляют MLP (Multilayer perceptron) слои трансформеров. На практике часто используется разреженный вариант MoE [7], где после получения оценок применяется операция top-k и используются только наиболее подходящие токену эксперты. В оригинальной работе предлагалось использовать sparse MoE в RNN сетях между LSTM слоями, как показано на изображении 2. Формально MoE можно представить следующим образом:

$$y = \sum_{i=1}^{n} G(x)_i E_i(x) \tag{1}$$

где $y$ выход MoE слоя, $x$ вход, $G$ роутер, а $E_i$ эксперт. В случае sparse MoE в качестве роутера можно использовать следующую функцию:

$$G(x) = Softmax(topK(x \cdot W_g))$$

При этом, если $G(x)_i = 0$, то $E_i(x)$ в выражении (1) не вычисляется, за счёт чего достигается разреженность.

Одним из недостатков sparse MoE является проблема "вырождения" роутеров, роутер может в значительной степени предпочитать отдельных экспертов и игнорировать других, что в свою очередь приводит к уменьшению фактически используемого числа параметров и упрощению модели. Крайней степенью "вырождения" может являться постоянный выбор одних и тех же экспертов, где при этом остальные эксперты не используются. Такая модель практически является обычной dense моделью, лишенной преимуществ MoE. Для борьбы с этой проблемой необходимо использовать добавки к функции потерь, штрафующие роутеры за неравномерное распределение токенов между экспертами. Так например в работе [2] предлагается использовать следующую добавку:

$$\alpha \cdot N \cdot \sum_{i=1}^{N} f_i \cdot P_i$$

где $\alpha$ — гиперпараметр, $N$ — число экспертов, $f_i$ — это доля токенов, которые попали в $i$-ого эксперта.

$$f_i = \frac{1}{T} \sum_{x \in \mathcal{B}} 1 \{\text{argmax } p(x) = i\}$$

и $P_i$ — это доля вероятностной массы, которая досталась $i$-ому эксперту.

$$P_i = \frac{1}{T} \sum_{x \in \mathcal{B}} p_i(x)$$

где $T$ — число токенов в батче $\mathcal{B}$.

## 3.3 MoELoRA

В работе [6] авторы предлагают использовать LoRA адаптеры в качестве MoE экспертов, но при этом они добавляют их параллельно каждому линейному слою, а также используют один внешний роутер. Роутер в MoELoRA отличается тем, что он один на всю модель и на вход принимает не входные токены, а id downstream задачи, при этом он сразу для каждого слоя выбирает экспертов и их веса. Такой подход позволил авторам обучить одну гибкую модель под нескольких downstream задач. Подход MoELoRA похож на описанный в данной работе использованием LoRA в качестве экспертов, но существенно отличается в устройстве роутера и не может быть применён на этапе дообучения с учителем, поскольку нет соответствующей разметки данных и на этапе предсказания потребуется отдельно предсказывать тип задачи, что, вероятно, потребует отдельную модель и может быть не тривиально.

# 4 Реализация

Для реализация было принято решение использовать библиотеку для работы с адаптерами Peft [8] и имплементацию MoE FastMoE [1], в которой реализованы эффективные cuda ядра для некоторых операций. Peft позволяет после инициализации модели модифицировать исходные слои с помощью адаптеров.

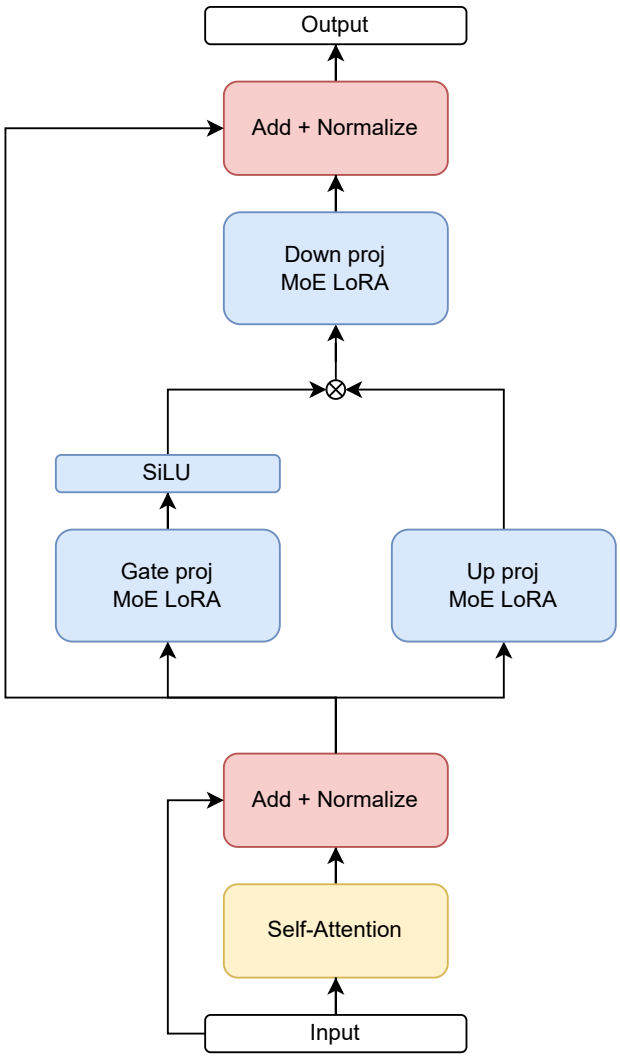


**Рис. 3:** Блок трансформера с обозначением, к каким слоям применяется адаптер.

Внутри каждого слоя MLP в трансформерах есть три матрицы, которые соответствуют трём проекциям, на рисунке 3 это Down proj, Up

proj и Gate proj. К каждой из этих матриц добавляется адаптер с роутером и экспертами-LoRA. В качестве роутера используется линейный слой с softmax. До softmax к получаемым логитам роутера применяется операция top-k. Каждому токену таким образом ставится в соответствие один или два эксперта. После применения экспертов их выходы домножаются на веса, получаемые из роутера.

Адаптер применяется к MLP из каждого блока трансформера в модели. В процессе обучения веса исходной модели остаются замороженными и обучаются только эксперты и роутеры.

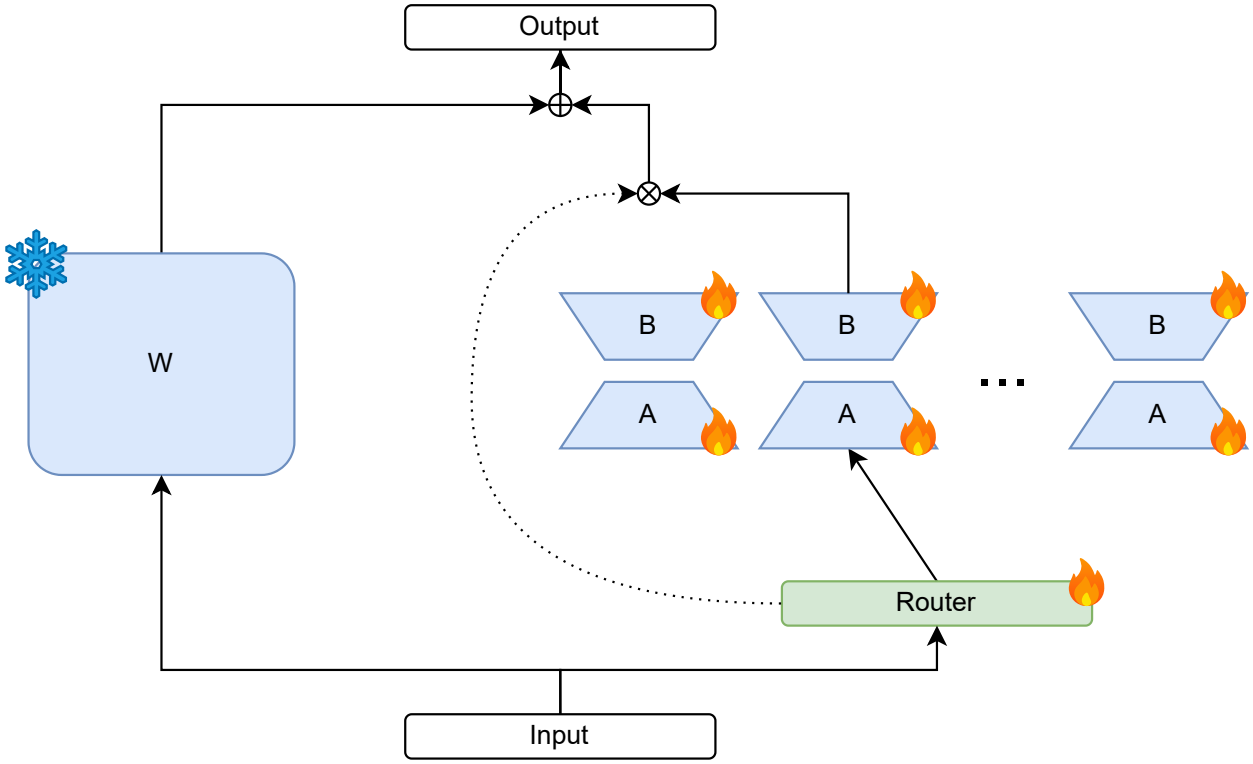


**Рис. 4:** Линейный слой W с MoE + LoRA адаптером

# 5 Эксперименты

Эксперименты проводились на предобученной модели размера 7b без учета новых параметров от адаптеров. В экспериментах модель дообучалась на закрытом датасете для дообучения с учителем. Он состоит из инструкций и подготовленных человеком ответов. При проведении экспертиментов были рассмотрены 4 конфигурации числа экспертов и параметра top-k. Число экспертов увеличивает итоговое число параметров, а параметр top-k определяет число экспертов, которое будет применено к каждому токену.

Figure 3: Examples from the Microeconomics task.

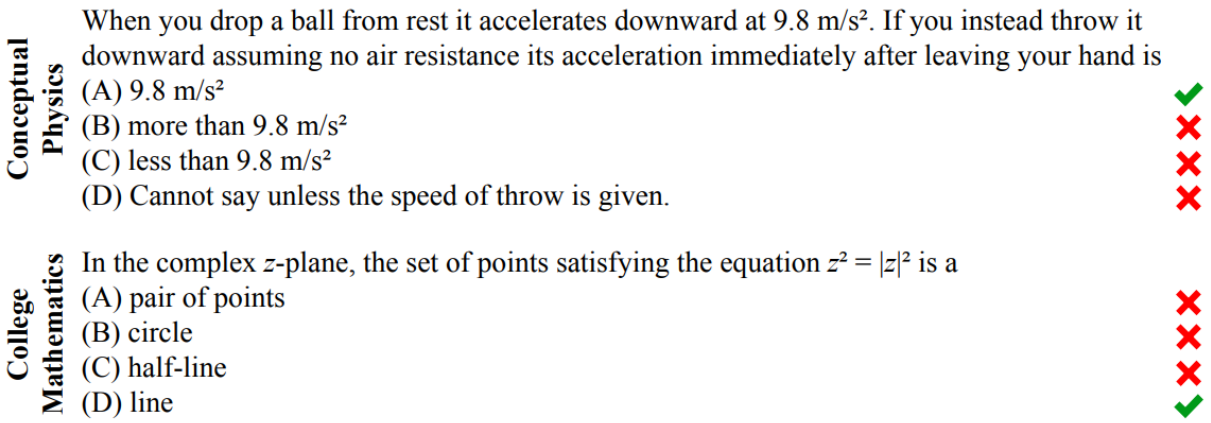


**Рис. 5:** Примеры вопросов из MMLU [5].

Для оценивания модели используются бенчмарки: MMLU или Massive Multitask Language Understanding [5] который проверяет модель на знание различных фактов в множестве категорий, переведенная версия MMLU на русский язык, RUBQ или Russian Knowledge Base Questions[1], тоже фактологический бенчмарк, но вопросы в нем составлены были изначально на русском языке, и бенчмарк на основе продуктовых задач, которые интересны клиентам. На изображении 5 приведены примеры вопросов из MMLU. При оценивании модели подается на вход вопрос из бенчмарка и 4 варианта ответа, модели необходимо выбрать правильный ответ. Соответственно случайным угадыванием можно добиться

[1]https://github.com/vladislavneon/RuBQ

точности в 0.25.

Полученные результаты сопоставимы с обычным методом дообучения по качеству при этом обучение требует меньше времени на **∼15%**. В таблице 1 приведены метрики для различных бенчмарков.

**Таблица 1:** Результаты экспериментов с разными конфигурациями MoE.

| Model | MMLU en (5-shot) | MMLU ru (5-shot) | RUBQ | Продуктовые задачи |
|---|---|---|---|---|
| Baseline | 0.505 | 0.454 | 0.515 | 0.706 |
| LoRA | 0.466 | 0.408 | 0.443 | 0.613 |
| 16 experts top-2 | **0.520** | **0.461** | 0.575 | **0.711** |
| 16 experts top-1 | 0.483 | 0.442 | 0.573 | 0.685 |
| 8 experts top-2 | 0.493 | 0.439 | 0.597 | 0.669 |
| 8 experts top-1 | 0.502 | 0.448 | **0.601** | 0.648 |

# 6 Заключение

В ходе работы были достигнуты следующие результаты:

1. Выполнен обзор предметной области.
2. Реализован адаптер на основе mixture of experts и LoRA.
3. Проведены эксперименты, показавшие пригодность метода в качестве альтернативы обычному дообучению. При сопоставимых метриках наблюдается уменьшение времени необходимого на дообучение на $\sim$**15%**.

# Список литературы


[1] He Jiaao, Qiu Jiezhong, Zeng Aohan, Yang Zhilin, Zhai Jidong, and Tang Jie. Fastmoe: A fast mixture-of-expert training system // arXiv preprint arXiv:2103.13262. — 2021.

[2] Fedus William, Zoph Barret, and Shazeer Noam. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity // Journal of Machine Learning Research. — 2022. — Vol. 23, no. 120. — P. 1–39.

[3] Hu Edward J, Shen Yelong, Wallis Phillip, Allen-Zhu Zeyuan, Li Yuanzhi, Wang Shean, Wang Lu, and Chen Weizhu. Lora: Low-rank adaptation of large language models // arXiv preprint arXiv:2106.09685. — 2021.

[4] Biderman Dan, Ortiz Jose Gonzalez, Portes Jacob, Paul Mansheej, Greengard Philip, Jennings Connor, King Daniel, Havens Sam, Chiley Vitaliy, Frankle Jonathan, et al. Lora learns less and forgets less // arXiv preprint arXiv:2405.09673. — 2024.

[5] Hendrycks Dan, Burns Collin, Basart Steven, Zou Andy, Mazeika Mantas, Song Dawn, and Steinhardt Jacob. Measuring massive multitask language understanding // arXiv preprint arXiv:2009.03300. — 2020.

[6] Liu Qidong, Wu Xian, Zhao Xiangyu, Zhu Yuanshao, Xu Derong, Tian Feng, and Zheng Yefeng. Moelora: An moe-based parameter efficient fine-tuning method for multi-task medical applications // arXiv preprint arXiv:2310.18339. — 2023.

[7] Shazeer Noam, Mirhoseini Azalia, Maziarz Krzysztof, Davis Andy, Le Quoc, Hinton Geoffrey, and Dean Jeff. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer // arXiv preprint arXiv:1701.06538. — 2017.

[8] Mangrulkar Sourab, Gugger Sylvain, Debut Lysandre, Belkada Younes, Paul Sayak, and Bossan Benjamin. PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods. — https://github.com/huggingface/peft. — 2022.