Low-rank self-play fine-tuning for small LLMs

***П. Ю. Мун*[1], *Н. В. Охотников*[2], *А. В. Грабовой*[3]**

В работе исследуется проблема дообучения больших языковых моделей (LLM) в условиях ограниченных ресурсов. Под ограниченными ресурсами понимается видеопамять, человеческое участие и время обучения. В работе рассматриваются модели до 1.5B. Предлагается метод дообучения, основанный на внедрении адаптеров LoRA, малоранговых раложений матриц, в слои архитектуры трансформера, и использовании стратегии self-play - текущая итерация генерирует предсказания, а обучающаяся повышает качество с помощью разграничения настоящих предсказаний от сгенерированных. Метод может снизить количество обучаемых параметров в 10000, и память в три раза, также он не требует размеченных данных помимо используемых на этапе SFT. Для анализа качества метода будет использована группа датасетов, таких как MMLU, Winogrande.

Ключевые слова: *LLM, LoRA, selp-play, SFT*

DOI:

1 Введение

Большие языковые модели (LLM) демонстрируют исключительные возможности в широком спектре областей, которые могут требовать специализированных знаний. Примерами таких областей могут служить: программирование [1], генерация текстов [6], обращения к базам данных [8]. Обычно процесс обучения LLM состоит из нескольких этапов: предварительного обучения, контролируемой тонкой натройки(SFT) и обучения с подкреплением на основе отзывов людей (RLHF). Предварительное обучение требует огромных вычислительных ресурсов, поэтому часто используют публичные предобученные модели (Llama3 [4], Qwen2.5 [7]) и дообучают под целевую задачу. Часто проблема жесткой ограниченности ресурсов встречается при дообучении маленьких LLM. Ограниченность ресурсов проявляется в недостатке видеопамяти для хранения и обновления параметров модели, необходимости в размеченных данных для повышения качества и времени обучения.

По вышеперечисленным причинам хотелось бы исследовать методы, которые бы использовали меньше ресурсов и не понижали качество модели. В данной работе предлагается метод дообучения LLM, который значительно снижает потребление видеопамяти, а также убирает необходимость в прямом человеческом участии без снижения качества модели. Метод основан на двух идеях.

- Во-первых, внедрение адаптеров LoRA в слои трансформера. Предполагается сравнительно маленькая внутренняя размерность обучаемых матриц модели, и вместо обучения всей матрицы добавляют приращение в виде малорангового разложения в две обучаемые матрицы, замораживая параметры изначальной.

- Во-вторых, механизме самостоятельной игры (self-play) для обучения адаптеров. Общий метод исследуется в статье [2], а в данной работе он применяется исключительно к адаптерам LoRA. Механизм состоит из последовательных игр модели со своими прошлыми итерациями. Прошлая итерация генерирует ответы по промптам части датасета на этапе SFT, а модель пытается различить настоящий ответ от сгенерированного. Предполагается наличие распределения у настоящих ответов датасета и полученная модель лучше настроена под предполагаемое распределение.

Предложенный метод развивается в двух направлениях: снижение требований к видео памяти и снижение человеческого участия. В обоих направлениях есть близкие альтернативы:

– Многие существующие решения по снижению требуемой памяти также применяют адаптеры LoRA с использованием других интсрументов. Одним из лучших методов является QLoRA [3], который применяет 4х битные NormalFloat и квантизацию, значительно снижая необходимую память. В отличии от LoRA, которая снижает количество обучаемых параметров, QLoRA также уменьшает размер параметров модели с помощью 4х битных NormalFloat, что приводит в меньшим требованиям к видеопамяти. С другой стороны, QLoRA требует передового оборудования для применения, что редко доступно в условиях ограниченных ресурсов.

– В направлении снижения человеческого участия можно выделить методы оптимизирующие этап RLHF. Одним из таких являетя метод DPO [5], который снижает зависимость от человеского оценивания результата. В то же время предложенный метод не требует размеченных данных, помимо тех, что используются на SFT.

Конечной целью работы является исследование оправданности применения предложенного метода к маленьким LLM в условиях ограниченных ресурсов.

[Тут будет сравнительная таблица, и по мере исследования она будет наполняться.]

2 Связанные работы

TODO:

3 Постановка задачи

Будем обозначать за $\mathbf{\Theta}$ - пространство всевозможных параметров трансформера, $p_{\boldsymbol{\theta}}$ - модель , а $\boldsymbol{\theta} \in \mathbf{\Theta}$ - ее параметры. На вход модель принимает промпт $\mathbf{x} = [x_1, ..., x_n]$ и возвращает ответ $\mathbf{y}' = [y'_1, ..., y'_m]$, который должен согласовываться с истинным ответом $\mathbf{y} = [y_1, ..., y_m]$.

Так как рассматривается дообучение модели, то у модели есть начальные параметры $\boldsymbol{\theta}_0$ и параметры итераций дообуения $\boldsymbol{\theta}_t$

Также для дальнейшей работы с методом введем вероятностные предположения о распределениях данных: $\mathbf{x} \sim q(\cdot)$, $\mathbf{y} \sim p_{data}(\cdot|\mathbf{x})$.

Теперь перейдем к введению понятий для обозначений методов.

3.1 Адаптеры LoRA

Адаптеры LoRA встраиваются на всех слоях в матрицы: W_q, W_k, W_v - query/key/value матрицы блоков self-attention в архитектуре трансформера. Тогда итоговая матрица параметров имеет вид: $W = W_0 + A * B$, где W_0 - изначальная матрица замороженных весов, а $A * B$ - адаптер.

Обобщая с матриц на все параметры модели, получим формулу

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_0 + \boldsymbol{\Delta\theta}_t$$

, где $\boldsymbol{\Delta\theta}_t$ - веса адаптеров LoRA на всех слоях, а $\boldsymbol{\theta}_0$ - замороженные веса исходной модели

3.2 Подробное описание SPIN

В центре механизма self-play находится игра между: игроком и противником. Цель противника сгенерировать ответы, которые были бы неразличимы с настоящими ответами, а цель игрока уметь различать настоящие ответы от сгенерированных. В рамках дообучения опонентом является прошлая итерация модели, а игроком текущая итерация, которая обучается.

Рассмотрим итерацию t+1, противником на данной итерации является $p_{\boldsymbol{\theta}_t}$, которая по промптам $\mathbf{x}$ генерирует ответы $\mathbf{y}'$, а игроком $p_{\boldsymbol{\theta}_{t+1}}$. Метод состоит из двух шагов: обучение игрока, обновление опонента

– В качестве целевой функции для первого шага используется:

$$f_{t+1} = \arg\max_{f \in \mathcal{F}_t} \mathbb{E}_{\mathbf{x}\sim q(\cdot),\mathbf{y}\sim p_{data}(\cdot|\mathbf{x})}\big[l(f(\mathbf{x},\mathbf{y}) - f(\mathbf{x},\mathbf{y}'))\big], \tag{1}$$

, где $l = log(1 + exp(-t))$, семейство функций $\mathcal{F}_t$ будет явно задано позже

– Для обновления опонента используется следующая функция:

$$p_{\boldsymbol{\theta}_{t+1}} = \arg\max_{p} \mathbb{E}_{\mathbf{x}\sim q(\cdot),\mathbf{y}\sim p(\cdot|\mathbf{x})}[f_{t+1}(\mathbf{x},\mathbf{y})] - \lambda\mathbb{E}_{\mathbf{x}\sim q(\cdot)}\mathrm{KL}\big(p(\cdot|\mathbf{x})||p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})\big),$$

, где λ - коэффициент регуляризации похожести итераций, а KL - дивергенция Кульбака-Лейблера. Регуляризация добавлена с целью контроля обучаемой итерации, чтобы $p_{\boldsymbol{\theta}_{t+1}}$ не сильно отличалась от $p_{\boldsymbol{\theta}_t}$

3.3 Одношаговое описание SPIN

Из последней формулы можно вывести оптимальное значение

$$\hat{p}(\mathbf{y}|\mathbf{x}) \propto p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x}) \exp\big(\lambda^{-1} f_{t+1}(\mathbf{x},\mathbf{y})\big). \tag{2}$$

Данная модель может не быть в рассматриваемом семействе моделей $\{p_{\boldsymbol{\theta}}|\boldsymbol{\theta} \in \boldsymbol{\Theta}\}$, но отсуда можно вывести семейство моделей для f_{t+1}:

$$\mathcal{F}_t = \left\{ \lambda \cdot \log \frac{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_\mathrm{t}}(\mathbf{y}|\mathbf{x})} \middle| \boldsymbol{\theta} \in \boldsymbol{\Theta} \right\}, \tag{3}$$

Подставляя элементы полученного семейства в формулу 1, получим искомый критерий качества, используемый при обучении.

$$L_{SPIN} = \mathbb{E}_{\mathbf{x}\sim q(\cdot),\mathbf{y}\sim p_{data}(\cdot|\mathbf{x})}\left[\ell\left(\lambda \log \frac{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})} - \lambda \log \frac{p_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})}\right)\right], \tag{4}$$

Тогда правило обновление параметров выглядит так:

$$\boldsymbol{\theta}_{t+1} = \arg\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} L_{SPIN}(\boldsymbol{\theta}, \boldsymbol{\theta}_t) \tag{5}$$

3.4 Применение SPIN к адаптерам

Обозначим за $\mathbf{\Omega} \subset \mathbf{\Theta}$ - подпространство весов адаптеров LoRA в архитектуре трансформера.

$$\boldsymbol{\Delta\theta}_{t+1} = \arg\min_{\boldsymbol{\Delta\theta}\in\mathbf{\Omega}} L_{SPIN}(\boldsymbol{\theta}_0 + \boldsymbol{\Delta\theta}, \boldsymbol{\theta}_0 + \boldsymbol{\Delta\theta}_t) \tag{6}$$

4 *

Список литературы

[1] Mark Chen и др. «Evaluating Large Language Models Trained on Code». В: *CoRR* abs/2107.03374 (2021). DOI: 10.48550/arxiv.2107.03374. arXiv: 2107.03374. URL: https://arxiv.org/abs/2107.03374.

[2] Zixiang Chen и др. «Self-Play Fine-Tuning Converts Weak Language Models to Strong Language Models». В: *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL: https://openreview.net/forum?id=O4cHTxW9BS.

[3] Tim Dettmers и др. «QLoRA: Efficient Finetuning of Quantized LLMs». В: *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. Под ред. Alice Oh и др. 2023. URL: http://papers.nips.cc/paper_files/paper/2023/hash/1feb87871436031bdc0f2beaa62a049b-Abstract-Conference.html.

[4] Abhimanyu Dubey и др. «The Llama 3 Herd of Models». В: *CoRR* abs/2407.21783 (2024). DOI: 10.48550/ARXIV.2407.21783. arXiv: 2407.21783.

[5] Rafael Rafailov и др. «Direct Preference Optimization: Your Language Model is Secretly a Reward Model». В: *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. Под ред. Alice Oh и др. 2023. URL: http://papers.nips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.

[6] Hugo Touvron и др. «Llama 2: Open Foundation and Fine-Tuned Chat Models». В: *CoRR* abs/2307.09288 (2023). DOI: 10.48550/ARXIV.2307.09288. arXiv: 2307.09288.

[7] An Yang и др. «Qwen2.5-1M Technical Report». В: *CoRR* abs/2501.15383 (2025). DOI: 10.48550/ARXIV.2501.15383. arXiv: 2501.15383.

[8] Victor Zhong, Caiming Xiong и Richard Socher. «Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning». В: *CoRR* abs/1709.00103 (2017). DOI: 10.48550/arxiv.1709.00103. arXiv: 1709.00103. URL: http://arxiv.org/abs/1709.00103.