Отчет по заданию: Реализация и анализ метода WARP [5]

Валеев Арслан Рустамович

август 2024.

Содержание

1 Введение

Метод Reinforcement Learning from Human Feedback (RLHF) популярный подход для алаймента языковых моделей, однако его реализация, особенно с использованием Proximal Policy Optimization (PPO), вызывает множество сложностей из-за нестабильности процесса обучения. В статье *Back to Basics: Revisiting*

REINFORCE Style Optimization for Learning from Human Feedback in LLMs [1] был предложен более простой подход на основе метода REINFORCE, который в ряде задач демонстрирует лучшие результаты. В работе *WARP: On the Benefits of Weight Averaged Rewarded Policies* [5] представлена модификация метода REINFORCE с использованием техник объединения моделей, что может привести к более стабильному и эффективному обучению. В данной работе приведена реализация и анализ этого метода.

2 Задачи

1. Обучить модель наград на основе датасета IMDb для классификации сантиментов отзывов
2. Разобраться с методом *WARP* [5] и реализовать его
3. Оценить средние значения награды и KL дивергенции для генераций обученной модели и сравнить их с генерациями SFT модели.
4. Проанализировать влияние изменения гиперпараметра μ на результаты.
5. Провести анализ полученных данных и предложить способы улучшения результатов.

3 Реализация

3.1 Создание датасета и обучение модели наград

Был использован датасет IMDb для создания пар (positive comment, negative comment). Модель наград была обучена на этом датасете с использованием библиотеки `accelerate` и модели `distilbert-base-cased`. Дополнительно была проведена валидация модели на `test` выборке при помощи метода RewardBench [4]

3.1.1 Детали обучения

Исходя из условия, сам по себе датасет достаточно объёмный (225 млн объектов), так что было решено на каждой эпохе брать 16 * 1000 (batch_size * batch_per_epoch) комментариев из датасета и обучать модель на них. Во время валидации для случайных 64 * 100 позитивных и негативных отзывов вычислялся скор, после чего считалось среднее значение награды для positive и negative классов и высчитывался RewardBench [4].

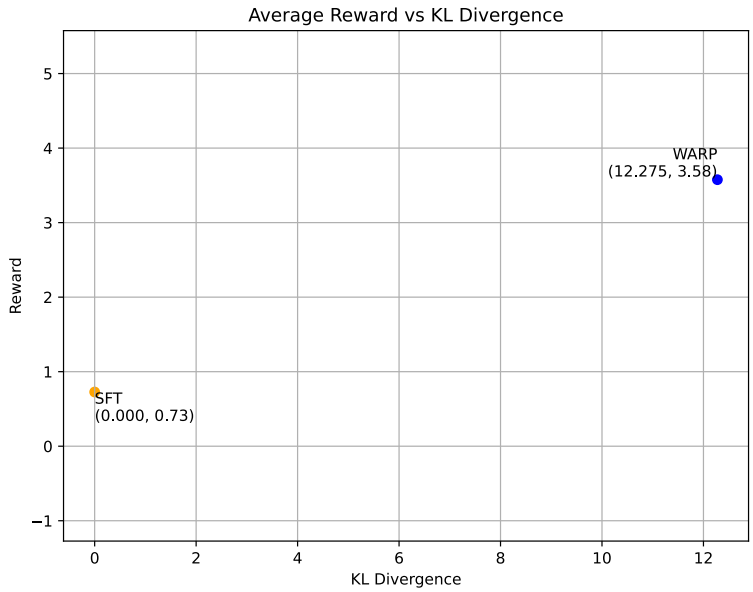

Рис. 1: средние награда и KL дивергенция SFT модели и WARP

Было решено обучать LoRA [2] для модели наград чтобы ускорить процесс и не обучать всю модель заново. Изначально предполагалось, что обучение LoRA частично решит проблему «забывчивости» модели, но, к сожалению, это не так [3]. Также был протестирован вариант обучения с заморозкой всех весов, кроме последнего слоя механизма attention и линейного классификатора в конце, но результаты были существенно хуже варианта loRA [2]: -0.07 на RewardBench [4] и разница между средней наградой для позитивных и негативных отзывов ниже на 6 значений - в два раза хуже варианта LoRA!

Из-за сложной логики training loop было решено не использовать библиотеку `trl` и реализовать обучение с использованием фреймворка accelerate из библиотеки Hugging Face.

Модель обучалась 15 эпох (8 часов на gpu kaggle notebooks), сейчас доступна по ссылке: https://huggingface.co/ChokeGM/reward_model_imdb

3.2 Реализация метода WARP

Метод WARP [5] был реализован на основе алгоритма из статьи с использованием предложенных гиперпараметров за исключением нескольких моментов:

1. был изменен learning rate оптимизатора Adam с 1e-6 на 5e-6 на основе эмпирических наблюдений (средняя награда модели, логическая согласованность генераций) и интуитивных доводов: в статье число шагов T = 9000, в нашем случае T = 100 - стало быть, с маленьким learning rate модель не успеет выучить существенных зависимостей.

2. batch size с 64 был изменен на 32 с шагом аккумуляции градиентов = 2, чтобы результат не отличался от начального варианта (модель с batch size = 64 не помещается на gpu бесплатных сервисов)

3. max и min длина генераций была установлена на 53 токена, чтобы не занимать много памяти gpu. Предполагается, что 53 токена достаточно для модели чтобы четко выразить свою позицию (positive, negative) в отзыве.

На бесплатных сервисах по предоставлению gpu (kaggle, google colab) обучение длится 20-30 минут. Готовая модель лежит здесь: https://huggingface.co/ChokeGM/WARP_model

3.3 Генерация продолжений и оценка метрик

Был создан датасет из тестовой части: 100 промптов длины 17 токенов, для которых были сгенерированы продолжения с использованием обученной модели WARP и модели SFT. Средние значения награды и KL дивергенции для обеих моделей отображены на графике 1.

3.3.1 Небольшой анализ результатов

Понятно, что разница в награде моделей происходит в основном засчёт генераций на негативных промптах: так как изначально модель обучена на генерацию продолжений, то на позитивных промптах она будет выдавать позитивный отзыв, что несильно отличается от обученной WARP [5] модели.

KL дивергенция немного высокая, но это не значит, что WARP модель несёт бред, лишь бы получить высокую награду - это лишь значит, что на одном и том же промпте модели ведут себя по-разному, т.е. к примеру на негативном промпте WARP [5] не генерирует негативный отзыв, в отличие от SFT модели.

Для оценки связности текста предлагается использовать большую opensource LM, по типу LLama 3.1 7B, однако пока неясно как это реализовывать: запускать локальную квантизованную модель или использовать api (возможен третий вариант).

3.4 Влияние изменения гиперпараметра

Гиперпараметр для варьирования был выбран μ - коэффицент сглаживания экспоненциального скользящего среднего EMA. Модель была обучена при значениях $\mu = 0.01$, $\mu = 0.025$ и $\mu = 0.05$, результаты отображены на графике 2

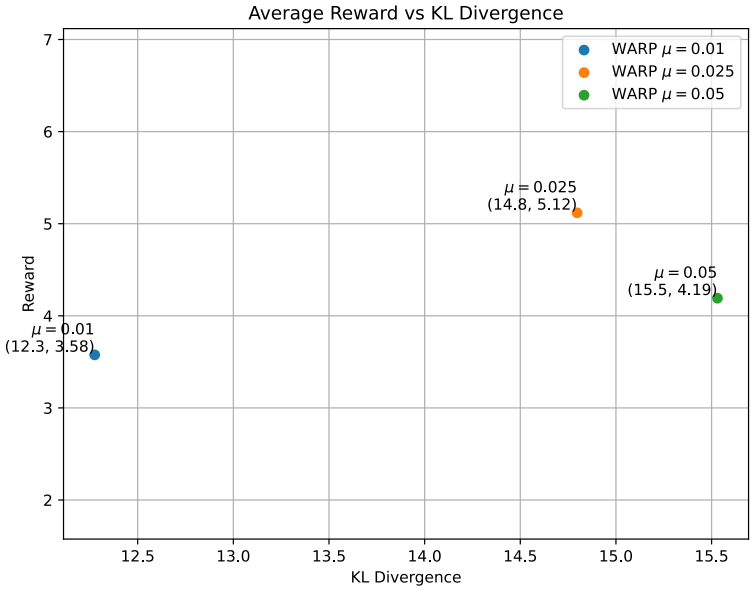

Рис. 2: средние награда и KL дивергенция модели WARP при различных параметрах μ

3.4.1 Интуиция, стоящая за выбором гиперпараметра

В статье [5] модель обучалась значительно дольше, чем в этой работе, вероятно поэтому малый коэффицент μ не даст модели сильно отойти от SFT. Возможно, при увеличении коэффицента WARP модель сможет заработать больше награды за те же сроки обучения. Да, она будет сильнее отличаться от SFT модели (KL дивергенция будет больше), но, как я уже писал выше, высокая KL дивергенция - не всегда плохо. Понятно, что совсем пренебрегать KL не стоит: иначе генерация WARP потеряет всякую смысловую нагрузку, просто стоит помнить, что KL дивергенция с SFT моделью $\neq$ мера логической согласованности генераций модели.

По этой же причине предлагалось рассмотреть для варьирования гиперпараметр T, но повышение числа шагов обучения сильно влияет на время обучения, и результаты анализа потенциально могли показать лишь, что число шагов $T = 100$ не раскрывает весь потенциал модели на данной задаче, тогда как изменение гиперпараметра μ потенциально может ускорить обучение модели, затрачивая те же ресурсы.

4 Результаты

На графике видно, что оптимальное значение $\mu = 0.025$ - повышение награды в 2 раза при незначительном увеличении KL. При $\mu = 0.05$ же модель слишком

быстро отходит от SFT, что ведёт к увеличению KL дивергенции и понижению смысловой нагрузки генераций.

5 Анализ

Полученные результаты показывают, что метод WARP действительно позволяет достичь более высоких значений награды, однако возникает вопрос стабильности модели при изменении гиперпараметров. Возможно, стоит рассмотреть дополнительные техники стабилизации обучения, такие как использование дополнительных методов регуляризации или изменения архитектуры модели наград.

6 To Do

Стоит рассмотреть llama 3.1 или (Tbank API)) для оценки смысловой связности генераций, так как модель наград с этим слабо справляется, а KL регуляризация не оптимизирует напрямую это значение. Также заметно, что при $T = 100$ угол между task векторами < 30 (в работе WARP [5] угол ≈ 90), причем при увеличении T угол также растёт. Стоит рассмотреть поведение модели при $T > 100$ (когда выдадут квоту в kaggle). Также следует обратить внимание на статью Back to Basics [1] и выделить оттуда схемы для дальнейшего улучшения алгоритма.

7 Заключение

Метод WARP демонстрирует хорошие результаты, но требует более тщательного подбора гиперпараметров и дополнительных техник стабилизации для обеспечения устойчивости модели. Возможно, стоит рассмотреть комбинацию методов из статьи Back to Basics [1] для дальнейшего улучшения результатов.

Список литературы

[1] Arash Ahmadian и др. *Back to Basics: Revisiting REINFORCE Style Optimization for Learning from Human Feedback in LLMs*. 2024. arXiv: 2402.14740 [cs.LG]. URL: https://arxiv.org/abs/2402.14740.

[2] Edward J. Hu и др. *LoRA: Low-Rank Adaptation of Large Language Models*. 2021. arXiv: 2106.09685 [cs.CL]. URL: https://arxiv.org/abs/2106.09685.

[3] Damjan Kalajdzievski. *Scaling Laws for Forgetting When Fine-Tuning Large Language Models*. 2024. arXiv: 2401.05605 [cs.CL]. URL: https://arxiv.org/abs/2401.05605.

[4] Nathan Lambert и др. *RewardBench: Evaluating Reward Models for Language Modeling*. 2024. arXiv: 2403.13787 [cs.LG]. URL: https://arxiv.org/abs/2403.13787.

[5] Alexandre Ramé и др. *WARP: On the Benefits of Weight Averaged Rewarded Policies*. 2024. arXiv: 2406.16768 [cs.LG]. URL: https://arxiv.org/abs/2406.16768.