Отчёт по заданию: Реализация и анализ метода WARP [6]

Валеев Арслан Рустамович

август 2024.

Содержание

1 Введение

Метод обучения с подкреплением на основе обратной связи от человека (RLHF) является популярным подходом для корректировки языковых моделей, однако

его реализация, особенно при использовании Proximal Policy Optimization (PPO), сталкивается со значительными сложностями, связанными с нестабильностью процесса обучения. В статье *Back to Basics: Revisiting REINFORCE Style Optimization for Learning from Human Feedback in LLMs* [1] предложен более упрощённый подход на основе метода REINFORCE, который в ряде задач показывает лучшие результаты. В работе *WARP: On the Benefits of Weight Averaged Rewarded Policies* [6] предложена модификация метода REINFORCE с использованием техник усреднения весов, что может способствовать более стабильному и эффективному обучению. В данной работе представлена реализация и анализ этого метода.

2 Задачи

1. Обучить модель наград на основе датасета IMDb для классификации сентиментов отзывов.
2. Изучить метод *WARP* [6] и реализовать его.
3. Оценить средние значения награды и KL-дивергенции для генераций обученной модели и сравнить их с генерациями модели SFT.
4. Проанализировать влияние изменения гиперпараметра μ на результаты.
5. Провести анализ полученных данных и предложить пути улучшения результатов.

3 Реализация

3.1 Создание датасета и обучение модели наград

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

3.1.1 Детали обучения

Исходя из объёма датасета (225 млн объектов), на каждой эпохе было выбрано 16 * 1000 (batch_size * batch_per_epoch) комментариев для обучения модели. В процессе валидации для случайных 64 * 100 позитивных и негативных отзывов вычислялся скор, после чего усреднялись значения награды для positive и negative классов и рассчитывался RewardBench [4].

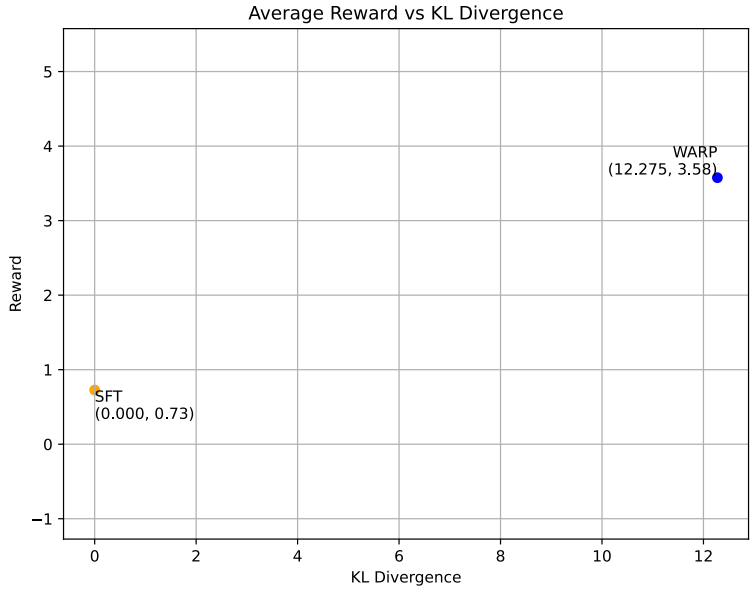

Рис. 1: Средние значения награды и KL-дивергенции моделей SFT и WARP

Для ускорения процесса обучения и избежания повторного обучения всей модели было решено использовать LoRA [2]. Однако, вопреки ожиданиям, обучение LoRA не решило проблему «забывчивости» модели [3]. Альтернативный подход с заморозкой всех весов, кроме последнего слоя механизма attention и линейного классификатора, показал худшие результаты по сравнению с LoRA [2]: -0.07 на RewardBench и разница между средней наградой для позитивных и негативных отзывов оказалась ниже на 6 пунктов, что в два раза хуже результата LoRA.

Из-за сложности логики training loop было принято решение отказаться от использования библиотеки `trl` и реализовать обучение с использованием фреймворка accelerate из библиотеки Hugging Face.

Модель обучалась в течение 15 эпох (8 часов на GPU Kaggle Notebooks) и доступна по ссылке: https://huggingface.co/ChokeGM/reward_model_imdb.

3.2 Реализация метода WARP

Метод WARP [6] был реализован на основе алгоритма из статьи с использованием предложенных гиперпараметров, за исключением некоторых изменений:

1. Размер batch был изменён с 64 на 32, с шагом аккумуляции градиентов = 2, чтобы результат не отличался от начального варианта (модель с batch size = 64 не помещалась на GPU бесплатных сервисов).

2. Максимальная и минимальная длина генераций была установлена на 53 токена, чтобы не перегружать память GPU. Предполагается, что 53 токена

достаточно для того, чтобы модель ясно выразила свою позицию (positive, negative) в отзыве.

На бесплатных сервисах для предоставления GPU (Kaggle, Google Colab) обучение длится 20-30 минут. Готовая модель доступна по ссылке: https://huggingface.co/ChokeGM/WARP_model.

3.3 Генерация продолжений и оценка метрик

Для тестовой части был создан датасет из 100 промптов длиной 17 токенов, для которых были сгенерированы продолжения с использованием обученной модели WARP и модели SFT. Средние значения награды и KL-дивергенции для обеих моделей представлены на графике 1.

3.3.1 Краткий анализ результатов

Разница в наградах моделей обусловлена в основном генерациями на негативных промптах: поскольку изначально модель была обучена на генерацию продолжений, на позитивных промптах она выдаёт позитивные отзывы, что не сильно отличается от обученной WARP модели [6].

Хотя KL-дивергенция несколько выше, это не свидетельствует о том, что WARP модель генерирует несвязный текст ради получения высокой награды — это лишь указывает на различие поведения моделей на одних и тех же промптах, например, на негативных промптах WARP модель [6] может генерировать текст, отличающийся от SFT модели.

Для оценки связности текста предлагается использовать более крупную открытоисходную языковую модель, такую как LLama 3.1 7B, однако пока неясно, как это лучше реализовать: запускать локальную квантизированную модель или использовать API.

3.4 Влияние изменения гиперпараметра

В качестве варьируемого гиперпараметра был выбран $\mathbf{I}$ — число операций Linear Interpolation Towards Initialization (LITI). Модель обучалась при значениях $\mathbf{I} = 2$, $\mathbf{I} = 4$ и $\mathbf{I} = 6$, результаты представлены на графике 2.

3.4.1 Интуиция выбора гиперпараметра

В оригинальной статье [6] модель обучалась значительно д

ольше, чем в данной работе, что предполагает, что малое число итераций $\mathbf{I}$ не позволяет модели существенно отклоняться от SFT. Возможно, $\mathbf{I} = 2$ не раскрывает весь потенциал модели на данной задаче. Хотя модель при этом будет

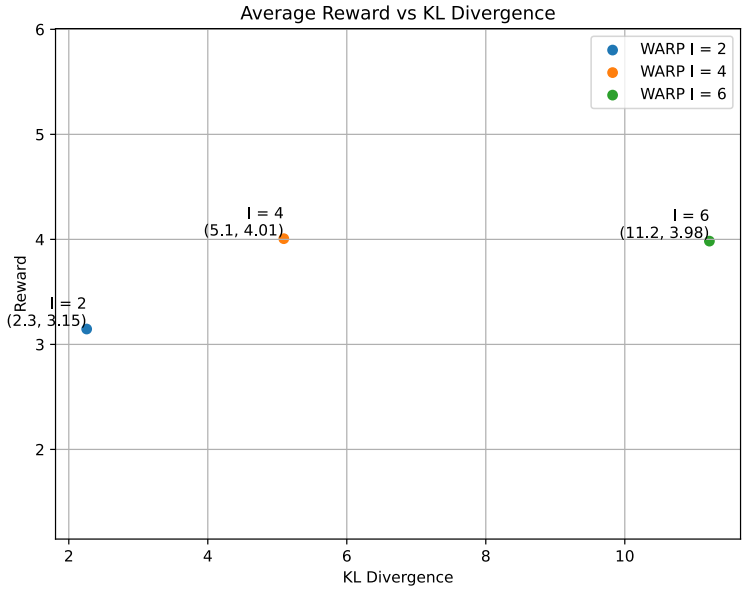

Рис. 2: Средние значения награды и KL-дивергенции модели WARP при различных значениях $\mathbf{I}$

больше отличаться от SFT модели (KL-дивергенция будет выше), как уже было отмечено, высокая KL-дивергенция не всегда является показателем низкого качества. Однако стоит помнить, что полное игнорирование KL-дивергенции может привести к утрате смысловой нагрузки генераций WARP.

Также рассматривался параметр T, однако изменения были незначительными. За одну итерацию угол между векторами ≈ 90 и модель ведёт себя стабильно при обучении, что позволяет заключить, что $T = 100$ является оптимальным значением.

4 Результаты

График показывает, что при значении $\mathbf{I} = 4$ модель достигает максимального значения награды (на 100 отложенных промптах), тогда как при $\mathbf{I} = 6$ увеличивается только KL-дивергенция. Возможно, за 4 итерации модель достигает оптимального состояния, а дальнейшие операции LITI лишь увеличивают расхождение модели с SFT, повышая только KL-дивергенцию.

5 Анализ

Полученные результаты свидетельствуют о том, что метод WARP действительно позволяет достигать более высоких значений награды, однако возникает

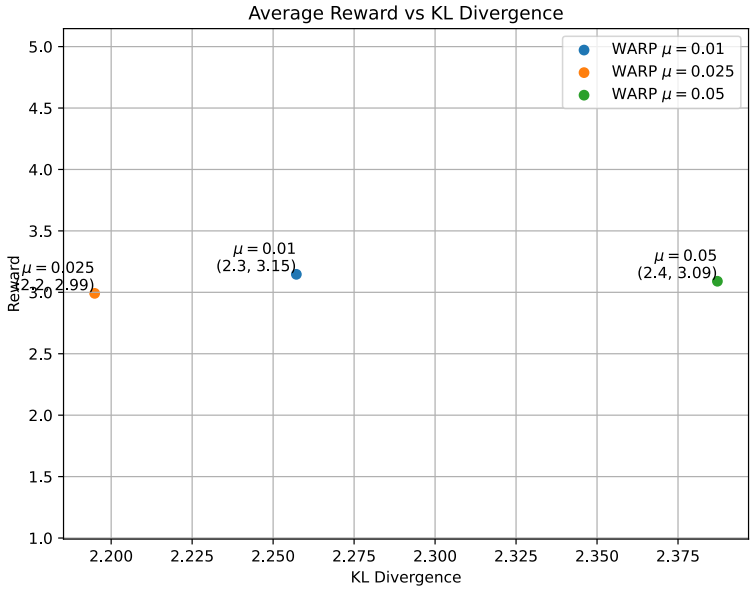

Рис. 3: Средние значения награды и KL-дивергенции модели WARP при различных значениях μ

вопрос о стабильности модели при изменении гиперпараметров. Возможно, следует рассмотреть дополнительные техники стабилизации обучения, такие как использование методов регуляризации или модификация архитектуры модели наград.

6 Недостатки и трудности

1. Изменение learning rate. В связи с небольшим числом итераций T (по сравнению с работой [6]) была предпринята попытка увеличения learning rate до 5e-6. Однако при этом угол между task vectors оказался менее 40 градусов, и генерация продолжений сводилась к повторению ключевых фраз, например: *10 out of 10! 10 out of 10! 9 out ....*

2. Измерение влияния EMA коэффициента μ. Предполагалось, что при увеличении значения μ метод WARP сможет достичь более высоких значений награды, поскольку задача корректировки модели в данном случае значительно проще, чем в задаче, представленной в оригинальной статье. Однако проведённые эксперименты показали, что увеличение μ не привело к ожидаемому росту награды.

7 Перспективы

Необходимо рассмотреть использование модели LLama 3.1 или API Tbank для оценки смысловой связности генераций, так как модель наград справляется с этим слабо, а KL-регуляризация не оптимизирует данное значение напрямую.

Результаты алгоритма показывают, что KL-дивергенция не полностью справляется с регуляризацией модели, возможно, стоит добавить оценку human-like показателей (согласованность, точность, логическая связность) для генераций через промптинг, как это было реализовано в работе RLAIF [5].

Также целесообразно обратить внимание на статью Back to Basics [1] и выделить из неё подходы для дальнейшего улучшения алгоритма.

8 Заключение

Метод WARP показывает хорошие результаты, но требует более тщательной настройки 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] Harrison Lee и др. «Rlaif: Scaling reinforcement learning from human feedback with ai feedback». В: *arXiv preprint arXiv:2309.00267* (2023).

[6] 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.