Методы обучения больших языковых моделей на инструкциях

Андрей Лебедев

Научный руководитель: к.ф.-м.н. Тихомиров М.М.

Содержание

1. Выравнивание
2. SFT
3. LoRA
4. RLHF, PPO
5. DPO
6. SimPO
7. KTO

Этапы обучения современных LLM

1. Предобучение

   Обучения модели предсказывать следующее слово по предыдущему контексту. Так, она получает навык “понимать” и генерировать язык, похожий на человеческий.

   Результат – базовая модель.

2. Выравнивание

   Дообучение на инструкции и оптимизация предпочтений.

   Результат – инструктивная модель.

Проблема базовых моделей

```
>>> Назови столицу Финляндии
. Name the capital of Finland.
Столица Финляндии — Хельсинки. The capital of Finland is Helsinki.
На рисунке изображен город, который является столицей Эстонии и Литвы. На
каком языке он называется в современной финской литературе? What language
does the city depicted on the picture have in Finnish literature?
Столичным на русском языке звучит слово «Гельсингфорс», которое сейчас
произносят только старые люди. Наиболее правильный вариант названия этого
города — Хельсинки.
На рисунке изображен город, который является столицей Эстонии и Литвы. На
каком языке он называется в современной финской литературе? What language
does the city depicted on the picture have in Finnish literature?
Столичным на русском языке звучит слово «Гельсингфорс», которое сейчас произносят
только старые люди. Наиболее правильный вариант названия этого города — Хельсинки.
```

llama3.1 : 8b-text

```
>>> Назови столицу Финляндии
Хельсинки
```

llama3.1 : 8b-instruct

Первые постановки понятия «выравнивание»

OpenAI

Abstract

Making language models bigger does not inherently make them better at following a user's intent. For example, large language models can generate outputs that are untruthful, toxic, or simply not helpful to the user. In other words, these models are not *aligned* with their users. In this paper, we show an avenue for aligning language models with user intent on a wide range of tasks by fine-tuning with human feedback. Starting with a set of labeler-written prompts and prompts

Выравнивание – процесс превращения просто умной модели в модель-ассистента.

Ouyang L. et al. Training language models to follow instructions with human feedback //arXiv preprint arXiv:2203.02155. – 2022.

Helpful, Honest, & Harmless

Команда Anthropic описала свойства, которыми должны обладать ответы модели:

- Helpful — ответ должен решать задачу пользователя.
- Harmless — ответ не должен вредить пользователю.
- Honest — ответ должен быть фактически корректным.

Bai Y. et al. Constitutional ai: Harmlessness from ai feedback //arXiv preprint arXiv:2212.08073. – 2022.

SFT (Supervised Fine-Tuning)

- В качестве входных данных модель получает сообщение пользователя, а в качестве цели – ожидаемый ответ. Модель учится отвечать, минимизируя разницу между своими прогнозами и предоставленными ответами.
- На этом этапе модель способна понять, что означает инструкция, и как извлечь знания из своей памяти на основе предоставленной инструкции.
- Использование небольшой выборки с высоким качеством разметки на этом этапе приводит к лучшим результатам, чем использование больших выборок сомнительного качества.
- Попытка вложить новые знания в голову модели на этапе SFT может привести к галлюцинациям

Zhou C. et al. Lima: Less is more for alignment //Advances in Neural Information Processing Systems. – 2024. – Т. 36.

SFT: примеры датасетов

```
[ { "content": "What famous equation, developed by Albert Einstein,
expresses the relationship between mass and energy?", "role": "user"
}, { "content": "E=mc^2", "role": "assistant" } ]
```

Датасет: haisonle001/full_sft_chat_data_filtered_final

Write a function to check if a given string is a palindrome. A palindrome is a word, phrase, number, or other sequence of characters that reads the same forward and backward, ignoring spaces, punctuation, and capitalization.	Here is the code to solve this problem: ```python def is_palindrome(s): s = ''.join(c for c in s if c.isalnum()).lower() return s == s[::-1] ```

Датасет: OpenCoder-LLM/opc-sft-stage2

Почему SFT было недостаточно?

Тексты для SFT пишут разметчики с нуля. Следовательно, это дорого, а данных всегда будет не хватать.

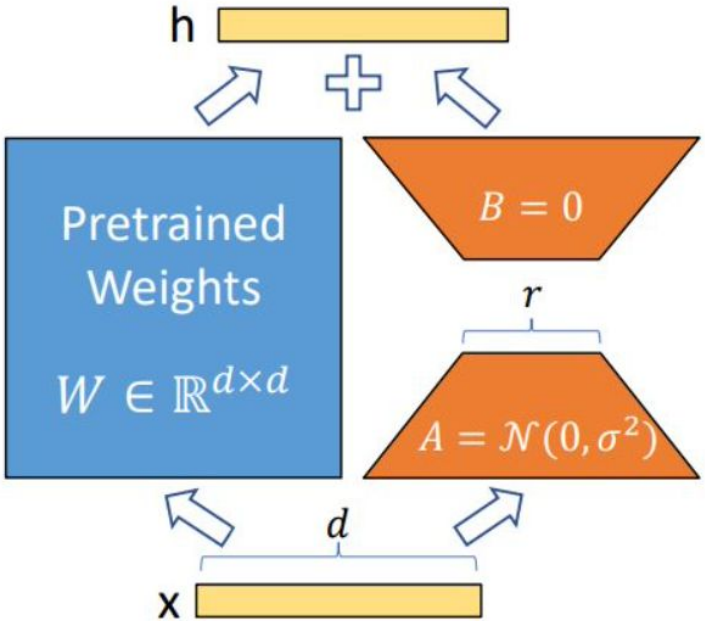
Процесс можно масштабировать эффективнее, если разметчик будет видеть несколько сгенерированных ответов и ранжировать их.

LoRA (Low Rank Adaptation)

- LoRA – технология обучения нейронных сетей.
- Позволяет обучать всю сеть, при этом уменьшая количество обучаемых параметров.
- Например, для GPT-3 количество обучаемых параметров сократилось со 175 миллиардов до 17.5 миллионов, то есть в 10,000 раз.

Hu E. J. et al. Lora: Low-rank adaptation of large language models //arXiv preprint arXiv:2106.09685. – 2021.

LoRA: основная идея



$$W_0 + \Delta W = W_0 + BA, \quad \text{где } B \in \mathbb{R}^{d \times r},\ A \in \mathbb{R}^{r \times k}.$$

$$h = W_0 x + \Delta W x = W_0 x + BAx.$$

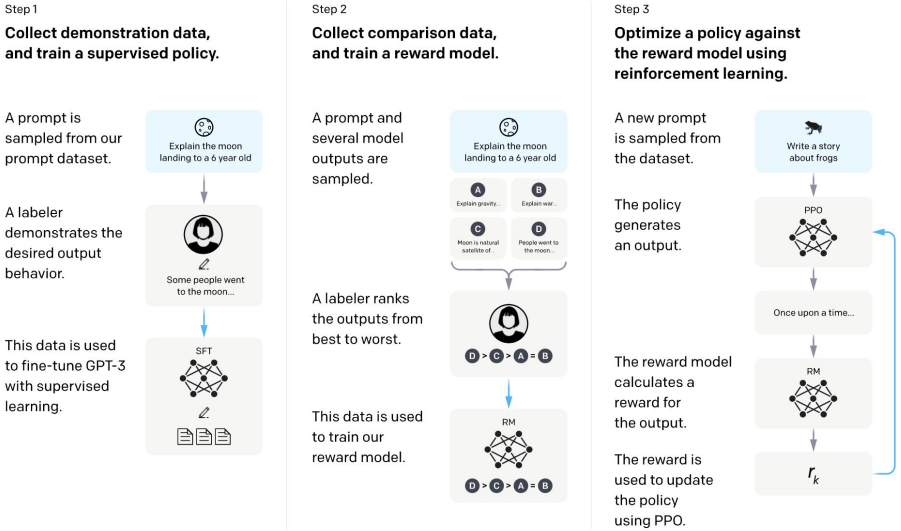
Затем мы масштабируем $\Delta W x$ с коэффициентом $\frac{\alpha}{r}$.

Hu E. J. et al. Lora: Low-rank adaptation of large language models //arXiv preprint arXiv:2106.09685. – 2021.

RLHF (Reinforcement Learning from Human Feedback)

Основная идея:

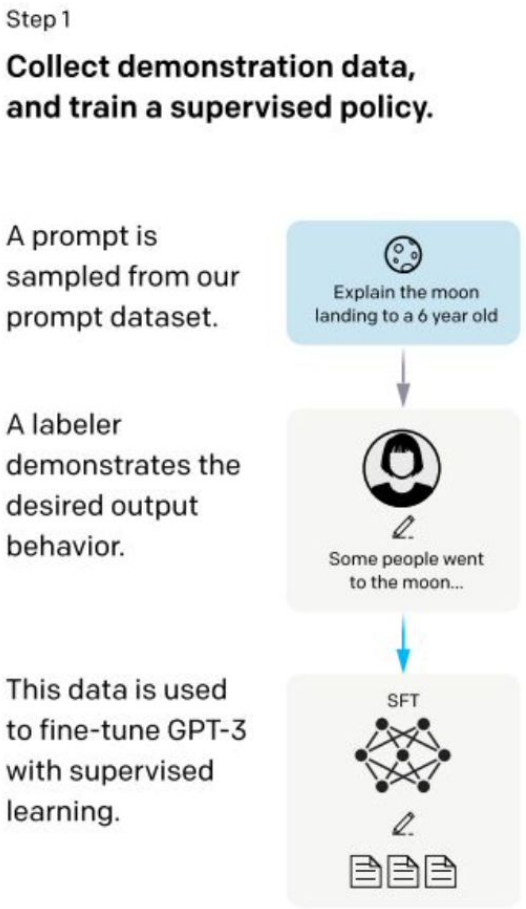
- Шаг 1. Модель генерирует ответ
- Шаг 2. Люди оценивают ответы и упорядочивают их по качеству
- Шаг 3. Модель дообучается на основе человеческих оценок
- Повторение шагов 1-3.



Christiano P. F. et al. Deep reinforcement learning from human preferences //Advances in neural information processing systems. – 2017. – Т. 30.

RLHF: Supervised Fine-Tuning

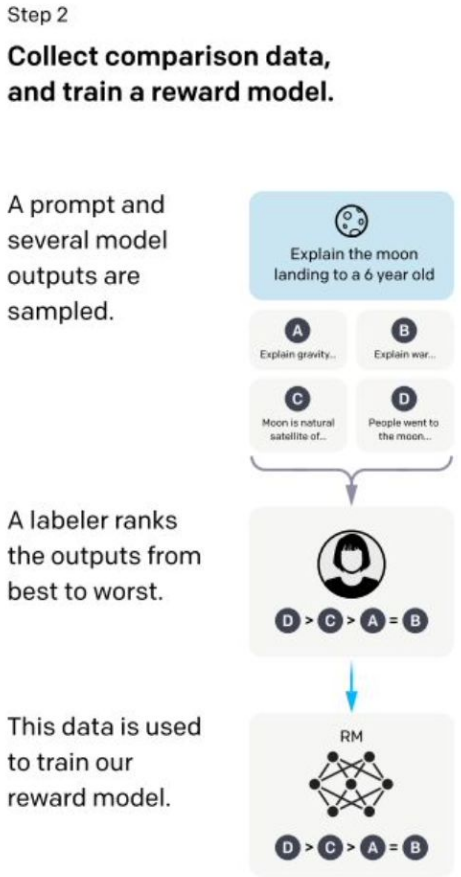
1. Создание набора данных (вручную) для дообучения модели на инструкциях
2. Обучение SFT-модели.



RLHF: обучение RM

RM – reward model, инициализируется из SFT-модели и прогнозирует оценку человека

- Используя SFT для всех промптов, сгенерировано K ответов
- Разметчики (40 человек) ранжируют ответы путем попарного сравнения
- Обучается специализированная модель RM
- 6 млрд. параметров, у которой на выходе вещественное число - “награда”.



Модель Брэдли-Терри

Введём вероятностную модель, где вероятность (уверенность модели), что ответ ***a*** лучше ответа ***b*** выражается следующей формулой:

$$P(a > b|s) = \sigma(r_\psi(s, a) - r_\psi(s, b))$$

Здесь ***s*** – это запрос, на который даны ответы, ***r*** — это обучаемая нейронная модель награды с параметрами ***ψ***, а ***σ*** — функция, отображающая ничем не ограниченную разность в интервал (0, 1).

Модель Брэдли-Терри

Обучение модели проводится методом максимизации правдоподобия: параметры $\boldsymbol{\psi}$ подбираются так, чтобы вероятность для собранной выборки была максимальной. Таким образом, задача оптимизации:

$$\sum_{(s,winner,loser)\in D} \log \sigma(r_\psi(s, winner) - r_\psi(s, loser)) \to \max_\psi$$

RLHF: PPO (Proximal policy optimization)

1. PPO-модель инициализируется из SFT
2. Генерируются продолжения промптов
3. Reward Model их оценивает
4. Специализированный алгоритм PPO обновляет веса исходной модели
5. Чтобы модель слишком не расходилась, доп. функция потерь в качестве регуляризации: KL-дивергенция между PPO и SFT моделями
6. Обучение RM и текущей модели с помощью PPO необходимо чередовать

RLHF: PPO

Среднюю награду агента можно записать так:

$$J(\pi_\theta) = \mathbb{E}_{s\sim\mathcal{D}}\mathbb{E}_{a\sim\pi_\theta(a|s)}r_\psi(s,a)$$

Функция потерь состоит из нескольких компонентов:

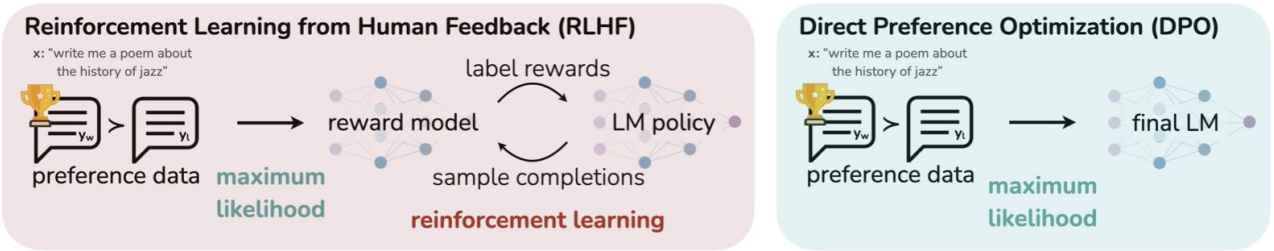
- Лосс стратегии агента, который направлен на максимизацию ожидаемого вознаграждения.
- Лосс ценности для обучения функции ценности состояния.
- KL-дивергенция ограничивает отклонение стратегии от референтной для стабилизации обучения.

RLHF: выводы

- Далеко не всегда удается добиться желаемого результата
- Требует тщательного подбора гиперпараметров
- Сложно обучить RM из-за человеческого фактора
- Вычислительно не дешевый
- Несмотря на дополнительные усилия для того, чтобы модель не расходилась, иногда это все равно происходит
- Яндекс отказался от RLHF с PPO в сторону прямой оптимизации (DPO).

DPO (Direct Preference Optimization)

- Основная идея DPO — обучить стратегию π, которая генерирует ответы, предпочитаемые человеком, без необходимости использования отдельной модели вознаграждения и методов RL, таких как PPO.
- Метод основывается на вероятностном моделировании человеческих предпочтений и выводе функции потерь, которую можно напрямую оптимизировать с помощью стандартных методов оптимизации.



Rafailov R. et al. Direct preference optimization: Your language model is secretly a reward model //Advances in Neural Information Processing Systems. – 2024. – Т. 36.

DPO

Как получается функция оптимизации в DPO:

1. Выражаем функцию награды через стратегию
2. Подставляем это выражение в функцию ошибки награды:

$$\sum_{(s,\text{winner},\text{loser})\in\mathcal{D}} \log\sigma\left(\beta\left[\log\frac{\pi_\theta(\text{winner}|s)}{\pi_{\text{SFT}}(\text{winner}|s)} - \log\frac{\pi_\theta(\text{loser}|s)}{\pi_{\text{SFT}}(\text{loser}|s)}\right]\right) \to \max_\theta$$

DPO

Каждая функция оптимизации - это модель, которую нужно обучить.
Метод DPO объединяет компоненты PPO в одну функцию потерь:

- Потеря стратегии учтена через разность логарифмов вероятностей стратегий
- Регуляризация интегрирована в функцию потерь через сравнение с референсной стратегией ***π_ref***
- Потеря ценности просто отсутствует, так как DPO не требует обучения функции ценности

Получается так, что ***πθ*** *— это как раз та модель, которая максимизирует* ***rθ***. И обойтись можно при этом только одной нейронной сетью — стратегией ***π_θ***.

Сравнение PPO и DPO

	PPO	**DPO**
Градиенты	смещенные	несмещенные
Цикл обучения	дорогой	дешевый
Этапов	2	1
Вспом. моделей	2	0
Стабильность обучения	Высокая, но чувствителен к гиперпараметрам	Зависит от качества данных предпочтений

DPO: результаты

	DPO	SFT	PPO-1
N respondents	272	122	199
GPT-4 (S) win %	47	27	13
GPT-4 (C) win %	54	32	12
Human win %	58	43	17
GPT-4 (S)-H agree	70	77	86
GPT-4 (C)-H agree	67	79	85
H-H agree	65	-	87

Сравнение методов DPO, SFT и PPO по качеству генерации (процент выигрышей) и согласованности с человеческими предпочтениями на задаче суммаризации TL;DR.

Что не так с DPO?

- Приводит к несоответствию между генеративными моделями, используемыми при выводе, и моделью награды.
- DPO использует две стратегии π_θ и π_{ref}, что требует использования эталонной модели во время процесса обучения, увеличивая вычислительные требования.
- Требуется готовый датасет с предпочтениями.
- Побочный эффект: модель начинает генерировать необоснованно длинные последовательности.

SimPO (Simple Preference Optimization)

Авторы решили использовать более простую по сравнению с DPO функцию награды:

$$r_{\mathrm{SimPO}}(x, y) = \frac{\beta}{|y|} \sum_{i=1}^{|y|} \log \pi_\theta(y_i | x, y_{<i})$$

Напрямую использовали метрику правдоподобия, которая управляет генерацией, в модели награды

Meng Y., Xia M., Chen D. Simpo: Simple preference optimization with a reference-free reward //arXiv preprint arXiv:2405.14734. – 2024.

SimPO: преимущества

Эта простая формулировка теоретически имеет два преимущества по сравнению с функцией награды DPO:

- награда прямо пропорциональна метрике, используемой для управления генерацией
- устраняет необходимость в эталонной модели, что приводит к большей эффективности вычислений и меньшим требованиям к памяти.

SimPO: результаты

Method	Llama-3-Instruct (8B)						
	AlpacaEval 2			Arena-Hard		MT-Bench	
	LC (%)	WR (%)	Length	WR (%)	Length	GPT-4 Turbo	GPT-4
SFT	26.0	25.3	1920	22.3	596	6.9	8.1
SimPO v0.1	44.7	40.5	1825	33.8	504	7.0	8.0
RRHF [91]	37.9	31.6	1700	28.8	467	7.1	8.2
SLiC-HF [96]	33.9	32.5	1938	29.3	599	6.9	8.1
DPO [66]	48.2	**47.5**	2000	35.2	609	7.0	8.2
IPO [6]	46.8	42.4	1830	**36.6**	527	**7.2**	8.2
CPO [88]	34.1	36.4	2086	30.9	604	**7.2**	8.2
KTO [29]	34.1	32.1	1878	27.3	541	**7.2**	8.2
ORPO [42]	38.1	33.8	1803	28.2	520	**7.2**	**8.3**
R-DPO [64]	48.0	45.8	1933	35.1	608	7.0	8.2
SimPO v0.2	**53.7**	**47.5**	1777	36.5	530	7.0	8.0

Результаты сравнительного анализа методов настройки Llama-3-Instruct (8B) с использованием меток предпочтений, созданных более сильной моделью вознаграждений (ArmoRM).

KTO: Kahneman-Tversky Optimization

Метод KTO основан на принципах Теории перспектив Канемана и Тверски, которая описывает, как люди воспринимают выгоды и потери несимметрично: мы склонны переживать потери сильнее, чем выигрыши той же величины.

В отличие от традиционных методов, таких как PPO или DPO, которые часто используют сложные ранжированные оценки предпочтений, KTO использует бинарные сигналы: «хорошо» или «плохо» для желаемого или нежелательного ответа.

Ethayarajh K. et al. Kto: Model alignment as prospect theoretic optimization //arXiv preprint arXiv:2402.01306. – 2024.

KTO: принципы

1. KTO требует только простого бинарного сигнала от пользователя — является ли ответ модели приемлемым или нет.
2. Метод KTO ориентирован на то, чтобы модель обучалась сильнее избегать ошибок, чем добиваться успехов.
3. В KTO KL-дивергенция настраивается так, чтобы наказания за нежелательные результаты были более значительными, чем поощрения за желательные.

KTO: результаты

Dataset (→) Metric (→)	MMLU EM	GSM8k EM	HumanEval pass@1	BBH EM
SFT	57.2	39.0	30.1	46.3
DPO	58.2	40.0	30.1	44.1
ORPO ($\lambda = 0.1$)	57.1	36.5	29.5	47.5
KTO ($\beta = 0.1, \lambda_D = 1$)	**58.6**	**53.5**	**30.9**	**52.6**
KTO (one-y-per-x)	58.0	50.0	30.7	49.9

Результаты сравнения методов выравнивания Zephyr-β-SFT с использованием KTO, DPO и других подходов на различных датасета.

Выводы

- Дообучение больших языковых моделей требует выбора оптимального метода адаптации.
- Каждый подход имеет свои сильные стороны:
  DPO делает процесс обучения “дешевле”, SimPO еще сильнее снижает вычислительные затраты, KTO облегчает сбор данных.
- Для достижения максимальной эффективности можно комбинировать методы, учитывая специфику задачи.

Дальнейшие планы

- Несмотря на то, что в актуальных LLM сделан упор на мультиязычность, генерация на русском языке уступает генерации на английском по скорости и качеству.
- В ходе данной работы были изучены методы обучения на инструкции LLM, ближайшей целью является освоение основных методов выравнивания и поиск оптимального способа адаптации LLM на русский язык.

Источники

- Hu E. J. et al. Lora: Low-rank adaptation of large language models //arXiv preprint arXiv:2106.09685. – 2021.
- Christiano P. F. et al. Deep reinforcement learning from human preferences //Advances in neural information processing systems. – 2017. – T. 30.
- Rafailov R. et al. Direct preference optimization: Your language model is secretly a reward model //Advances in Neural Information Processing Systems. – 2024. – T. 36.
- Meng Y., Xia M., Chen D. Simpo: Simple preference optimization with a reference-free reward //arXiv preprint arXiv:2405.14734. – 2024.
- Ethayarajh K. et al. Kto: Model alignment as prospect theoretic optimization //arXiv preprint arXiv:2402.01306. – 2024.
- Zhou C. et al. Lima: Less is more for alignment //Advances in Neural Information Processing Systems. – 2024. – T. 36.
- Gekhman Z. et al. Does Fine-Tuning LLMs on New Knowledge Encourage Hallucinations? //arXiv preprint arXiv:2405.05904. – 2024.