Direct Preference Optimization

1. 基础知识

对比学习优质与劣质Response，让模型回答更偏优质响应，DPO旨在最小化对比损失。

2. 损失函数

DPO损失实际上是对重新参数化奖励模型的奖励差异的交叉熵损失

$$L_{\mathrm{DPO}} = -\log\sigma\left(\beta\left(\log\frac{\pi_\theta(y_{\mathrm{pos}} \mid x)}{\pi_{\mathrm{ref}}(y_{\mathrm{pos}} \mid x)} - \log\frac{\pi_\theta(y_{\mathrm{neg}} \mid x)}{\pi_{\mathrm{ref}}(y_{\mathrm{neg}} \mid x)}\right)\right)$$

σ： sigmoid函数；

β： 超参数，β 值越高，这个对数差值就越重要

大括号内，是两个对数的差值，分别关注正样本和负样本

πθ： 一个微调后的模型，θ 是在这里想要调整的参数；πref：原始模型，参数不可调整

ypos丨x：在给定提示的情况下，产生正面回复的概率是多少

这个对数比值项可以被看作是奖励模型的重新参数化。如果你将其视为奖励模型，那么这个DPO损失实际上就是正样本和负样本之间奖励差异的sigmoid函数。DPO试图最大化正样本的奖励，并最小化负样本的奖励。

具体看论文https://blog.csdn.net/qq_54708219/article/details/149353780

Direct Preference Optimization: Your Language Model is Secretly a Reward Model

Rafael Rafailov[*†] **Archit Sharma**[*†] **Eric Mitchell**[*†]

Stefano Ermon[†‡] **Christopher D. Manning**[†] **Chelsea Finn**[†]

[†]Stanford University [‡]CZ Biohub
{rafailov,architsh,eric.mitchell}@cs.stanford.edu

Abstract

While large-scale unsupervised language models (LMs) learn broad world knowledge and some reasoning skills, achieving precise control of their behavior is difficult due to the completely unsupervised nature of their training. Existing methods for gaining such steerability collect human labels of the relative quality of model generations and fine-tune the unsupervised LM to align with these preferences, often with reinforcement learning from human feedback (RLHF). However, RLHF is a complex and often unstable procedure, first fitting a reward model that reflects the human preferences, and then fine-tuning the large unsupervised LM using reinforcement learning to maximize this estimated reward without drifting too far from the original model. In this paper we introduce a new parameterization of the reward model in RLHF that enables extraction of the corresponding optimal policy in closed form, allowing us to solve the standard RLHF problem with only a simple classification loss. The resulting algorithm, which we call *Direct Preference Optimization* (DPO), is stable, performant, and computationally lightweight, eliminating the need for sampling from the LM during fine-tuning or performing significant hyperparameter tuning. Our experiments show that DPO can fine-tune LMs to align with human preferences as well as or better than existing methods. Notably, fine-tuning with DPO exceeds PPO-based RLHF in ability to control sentiment of generations, and matches or improves response quality in summarization and single-turn dialogue while being substantially simpler to implement and train.

1 Introduction

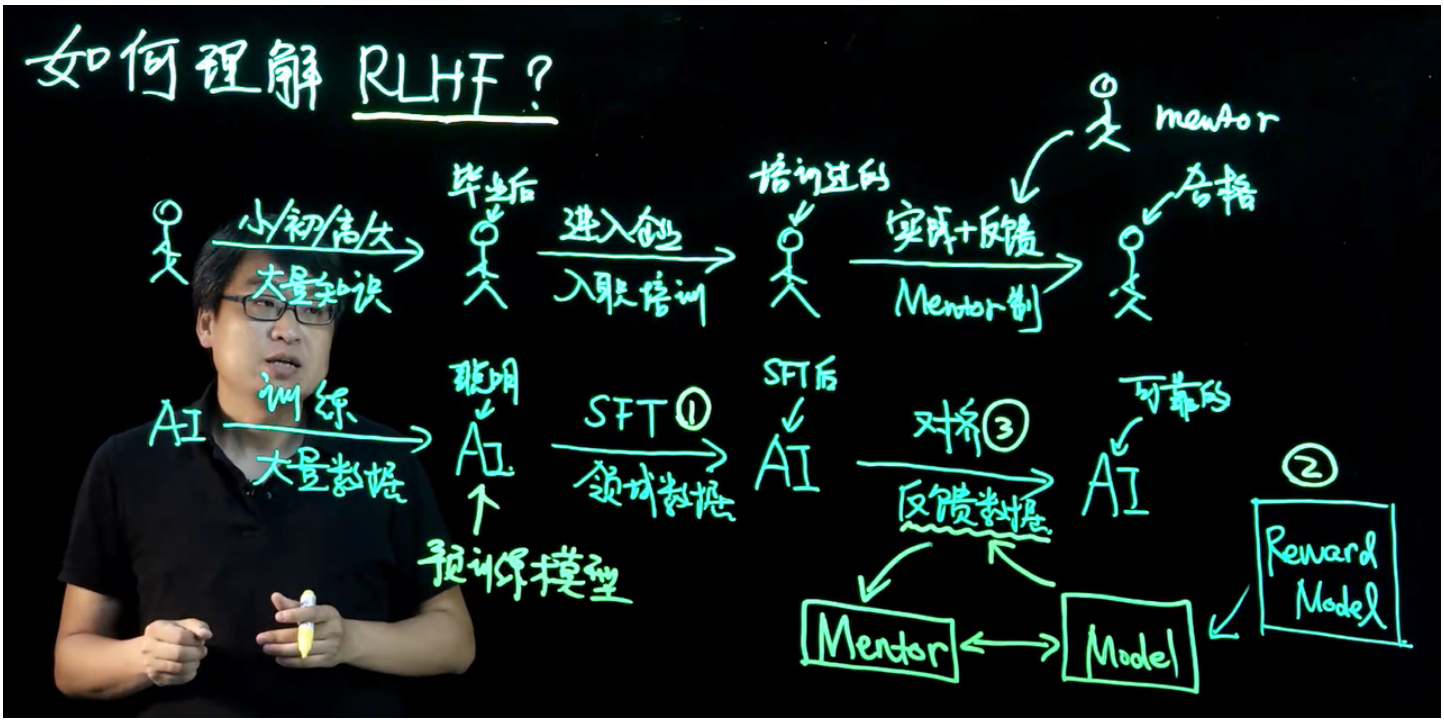
Large unsupervised language models (LMs) trained on very large datasets acquire surprising capabilities [11, 7, 42, 8]. However, these models are trained on data generated by humans with a wide variety of goals, priorities, and skillsets. Some of these goals and skillsets may not be desirable to imitate; for example, while we may want our AI coding assistant to *understand* common programming mistakes in order to correct them, nevertheless, when generating code, we would like to bias our model toward the (potentially rare) high-quality coding ability present in its training data. Similarly, we might want our language model to be *aware* of a common misconception believed by 50% of people, but we certainly do not want the model to claim this misconception to be true in 50% of queries about it! In other words, selecting the model's *desired responses and behavior* from its very wide *knowledge and abilities* is crucial to building AI systems that are safe, performant, and controllable [28]. While existing methods typically steer LMs to match human preferences using reinforcement learning (RL),

[*]Equal contribution; more junior authors listed earlier.

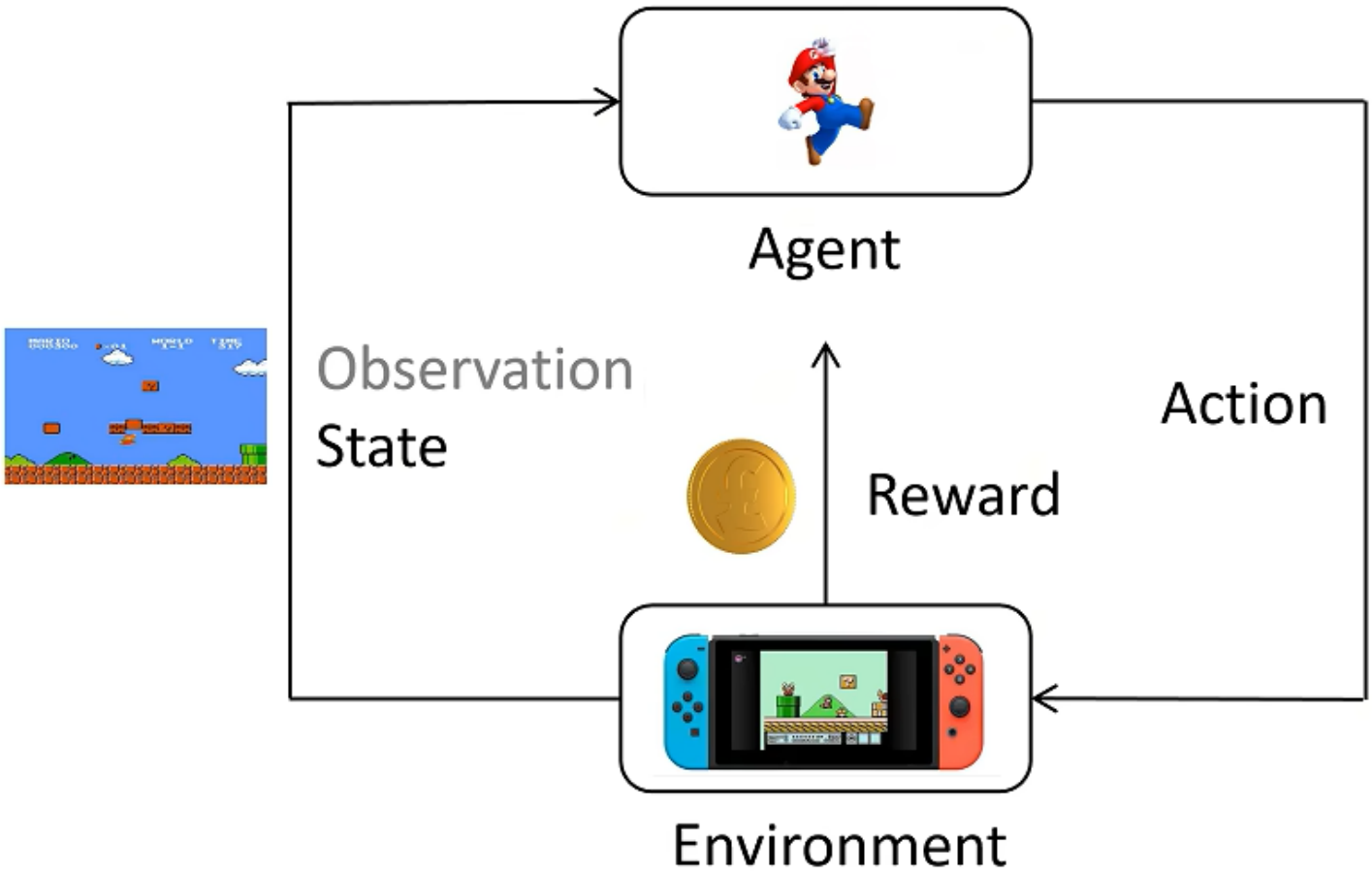
37th Conference on Neural Information Processing Systems (NeurIPS 2023).

DPO.pdf

RLHF（reinforcement learning from human feedback）：SFT——>奖励模型——>对齐


如何理解 RLHF？
Mentor
Reward Model
Mentor
Model

PPO（）：


Agent
Observation
State
Action
Reward
Environment

Action Space: 可选择的动作，比如 {left，up，right}

Policy: 策略函数，输入State，输出Action的概率分布。一般用π表示。

$\pi(left|s_t) = 0.1$
$\pi(up|s_t) = 0.2$
$\pi(right|s_t) = 0.7$

探索多样性

输出多样性

→ 根据概率分布采样

Trajectory: 轨迹,用τ表示， 一连串状态和动作的序列。Episode,Rollout。 $\{s_0, a_0, s_1, a_1, \ldots\}$

$s_{t+1} = f(s_t, a_t)$ 确定

$s_{t+1} = P(\cdot|s_t, a_t)$ 随机

Return: 回报，从当前时间点到游戏结束的Reward的累积和。

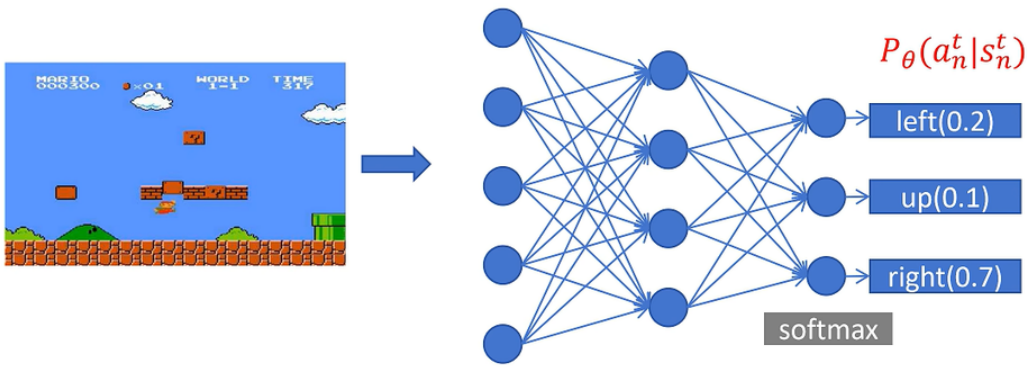
期望： 每个可能结果的概率与其结果值的乘积之和

$$\mathrm{E(x)}_{x\sim p(x)} = \sum_x x * p(x) \approx \frac{1}{n}\sum_{i=1}^{n} x \quad x\sim p(x)$$

目标：训练一个Policy神经网络π，在所有状态S下，给出相应的Action，得到Return的期望最大。

目标：训练一个Policy神经网络π，在所有的Trajectory中，得到Return的期望最大。

$$\mathrm{Loss} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n} R(\tau^n)\log P_\theta(a_n^t|s_n^t)$$



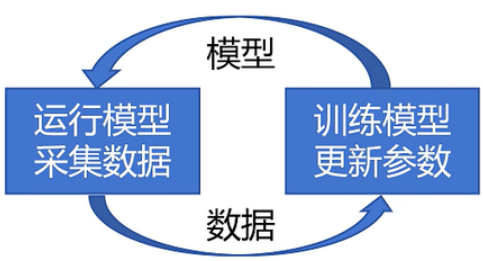
$R(\tau^n)$

$\tau^1 -> R(\tau^1)$

$\tau^2 -> R(\tau^2)$

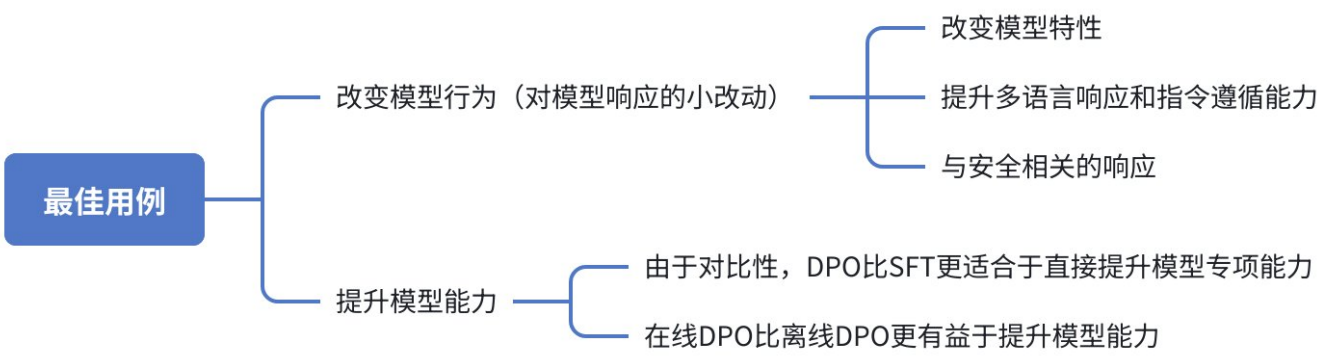
$\vdots$

$\tau^n -> R(\tau^n)$



On Policy

3. 最佳用例



4. DPO的数据整理策略

校正法	针对需要修改的响应提问，原始模型生成的响应为劣质响应，在其基础上修改为我们想要的响应作为优质响应。
在线/奖励政策	同一个提示，让原始模型生成多个响应，利用奖励函数或人为选定其中的最佳响应作为正样本和最差响应作为负样本

- 避免过拟合：当正样本总是包含一些特殊词汇，而负样本不包含时，模型可能会学习某种捷径，那么在这个数据集上进行训练可能非常不稳定，可能需要更多的超参数调整才能让DPO在这里发挥作用。

5.代码解析

DPO in practice.md