Note on Direct Preference Optimization

Tianbing Xu

October 28, 2023

1 Main idea

Direct Preference Optimization (DPO) focuses on the direct optimization of a language model (the policy network in reinforcement learning) to align with human preferences, all without the need for explicit reward modeling through a preference model or traditional reinforcement learning techniques for policy optimization. Notably, DPO achieves this by employing a change of variables to establish the preference loss as a direct function of the policy and then proceeds to optimize the policy through a straightforward binary cross-entropy objective. Figure 1 provides a visual comparison between DPO and Reinforcement Learning from Human Feedback (RLHF).

2 Derivation

The reward of completion y given the input x, w.r.t policy (language model),

$$r(x, y) = \beta \left(\log \frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)} \right) + \beta \log Z(x) \tag{1}$$

where, this reward is the optimal policy with KL-constrained RL,

$$\pi_\theta(y|x) = \frac{1}{Z(x)} \pi_{ref}(y|x) \exp(\frac{1}{\beta} r(x, y))$$

The Bradley–Terry model is the probability of pairwise comparison between completions y_1 and y_2, as y_1 is preferred or better than y_2 turns out to be true,

$$p(y_1 > y_2|x) = \sigma\left(r(x, y_1) - r(x, y_2)\right)$$

where $\sigma(x) = 1/(1 + \exp(-x))$ is the logistic function.

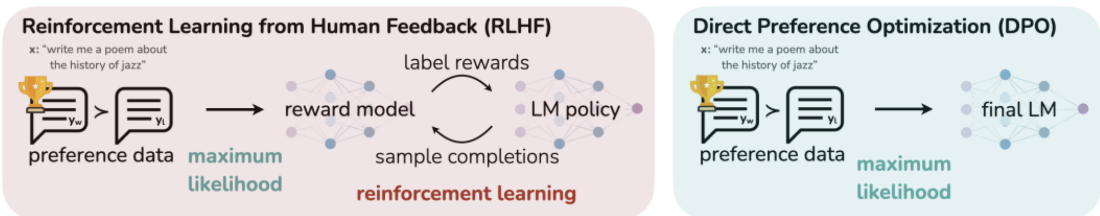

Figure 1: **DPO optimizes for human preferences while avoiding reinforcement learning.** Existing methods for fine-tuning language models with human feedback first fit a reward model to a dataset of prompts and human preferences over pairs of responses, and then use RL to find a policy that maximizes the learned reward. In contrast, DPO directly optimizes for the policy best satisfying the preferences with a simple classification objective, without an explicit reward function or RL.

Figure 1: DPO vs RLFH

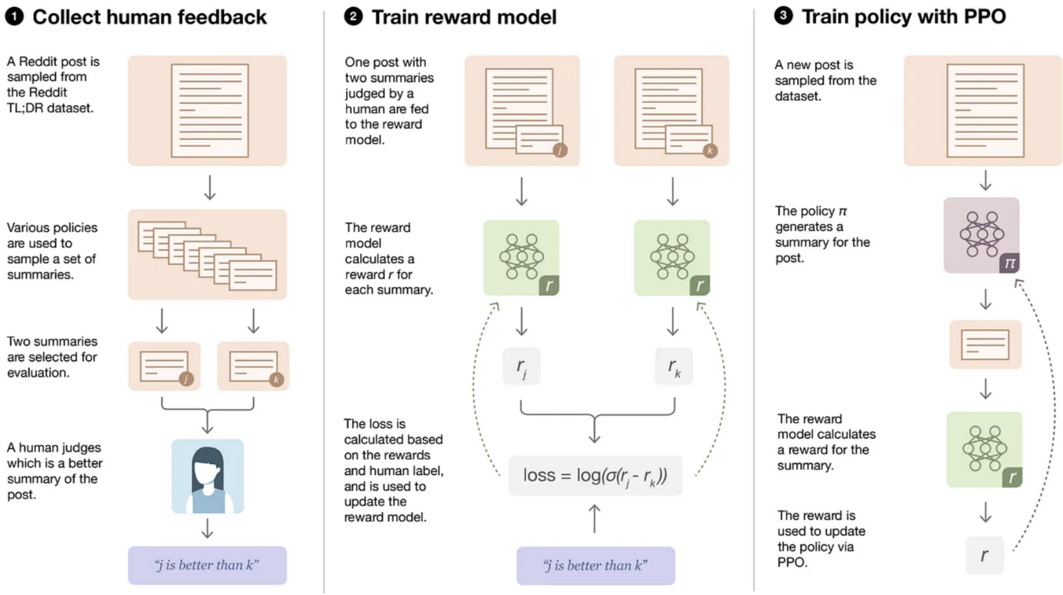

Figure 2: RLFH

That is,

$$P(y_1 > y_2|x) = \frac{1}{1 + \exp\left(\beta log \frac{\pi(y_2|x)}{\pi_{ref}(y_2|x)} - \beta \log \frac{\pi(y_1|x)}{\pi_{ref}(y_1|x)}\right)} \tag{2}$$

Then, the cross-entropy loss of Direct Preference Optimization w.r.t the policy,

$$L_{DPO}(\pi_\theta, \pi_{ref}) = -E_D \left[\log P(y_1 > y_2|x)\right] = -E_D \left[\beta \log \frac{\pi_\theta(y_2|x)}{\pi_{ref}(y_2|x)} - \beta \log \frac{\pi_\theta(y_1|x)}{\pi_{ref}(y_1|x)}\right] \tag{3}$$

Last, the policy gradient w.r.t θ is,

$$\nabla_\theta L_{DPO}(\pi_\theta, \pi_{ref}) = -\beta E_D \left[\sigma(\hat{r}(x, y_2) - \hat{r}(x, y_1)) \left[\nabla_\theta \log \pi_\theta(y_1|x) - \nabla \log \pi_\theta(y_2|x)\right]\right] \tag{4}$$

where $\hat{r}(x, y) = \beta \left(\log \pi_\theta(y|x) - \log \pi_{ref}(y|x)\right)$ is the implicit reward (see 1).

3 RL from Human Feedback (Figure 2)

3.1 SFT phase

We fine-tuned a pre-trained language model with high-quality data for downstream tasks and obtained π^{SFT}.

3.2 Reward modeling with Preference Model

Similarly, The Bradley–Terry model for the human preference w.r.t the reward r_ϕ,

$$p(y_1 > y_2|x) = \sigma\left(r_\phi(x, y_1) - r_\phi(x, y_2)\right) = \frac{\exp(r_\phi(x, y_1))}{\exp(r_\phi(x, y_1)) + \exp(r_\phi(x, y_2))} \tag{5}$$

where the reward model r_ϕ is initialized from π^{SFT} (superivsed fine-tuning) with addition of a linear layer on top of it and produce a single prediction for the reward value.

The negative log-likelihood by framing the problem as a binary classification,

$$L_R(r_\phi, D) = -E_D \left[\log \sigma\left(r(x, y_1) - r(x, y_2)\right)\right] \tag{6}$$

3.3 RL fine-tuning Phase

Here, we optimize the policy (language model) to maximize the expected rewards,

$$max_{\pi_\theta} E_D[r_\phi(x,y)] - \beta D_{KL}\left[\pi_\theta(y|x)||\pi_{ref}(y|x)\right] \tag{7}$$

where the corresponding reward function is,

$$r(x,y) = r_\phi(x,y) - \beta(\log \pi_\theta(y|x) - \log \pi_{ref}(y|x)) \tag{8}$$

References

[1] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, Chelsea Finn, *Direct Preference Optimization: Your Language Model is Secretly a Reward Model, 2023*

[2] L. Ouyang, et. al OpenAI, *Training language models to follow instructions with human feedback, NIPS 2022*