- TOC {:toc}

{:#Introduction}

Introduction

The LLMs are generally trained on large corpora, and most of them are consisted with web-crawled data. The LLMs are trained by learning to imitate the data. This is called *Supervised Fine-Tuning* (SFT). THe objective function of the SFT can be expressed as below.

$$ \mathcal{L}|*text{SFT} (|theta) = |*frac{1}{T} |sum |limits {t = 1}^T |log p|theta (y_t, x, y{<t}) $$

The objective **doesn't reflect the human preference**. Instead, the LLMs naively learn how to shadow the plain text. At the result, LLMs' behaviors don't align with human values. The RLHF became the one of the most effective values.

However, RLHF is generally instable and requires much computing resources due to the nature of the reinforcement learning. The *Direct Preference Optimization* address this problem by changing the KL-regularized bandit into the log-likelihood margin maximization problem.

{:#CoreIdea}

Core Idea

Problem Formulation

The DPO's contribution is to represent the reward model with the policy LLM. Here's the detailed step.

1. **Train a Reward Model (RM):** A dataset of human preferences is collected. For a given prompt x, two responses (y_w, y_l) are sampled from the SFT model, and a human tutors label which one they "win" (y_w) and which they "lose" (y_l). In the most of the previous works, a separate model, the RM $r_\phi(x, y)$, is trained to predict this preference. It's optimized on a ranking loss, often modeled by the Bradley-Terry model (BT model):

$$ p_*(x, y_w \succ y_l) = \frac{1}{1 + \exp (r(x, y_l) - r(x, y_w))} $$

2. Optimize the Policy via RL: Next, this frozen RM r_ϕ is used as the reward function. The reference policy, π_ref, is optimized using an RL algorithm like PPO to maximize the reward, while a KL-divergence penalty keeps the policy from moving too far from the original SFT model. The KL penalty is essential for preventing LLM's reward hacking problem. This is the KL-regularized bandit setting. The objective is:

$$\max_{\pi} \mathbb{E}*{y |*sim |*pi(y|x)} |*left[r\phi(x, y) \right] - \beta D_\text{KL}(\pi(y|x) || \pi_\text{ref}(y|x))$$

This stage is the source of all the complexity and instability: it requires sampling from the policy, managing a separate critic and reward model, and dealing with the high variance of RL optimization.

The DPO Derivation

The authors of DPO had a key insight: this entire two-stage process can be collapsed into one.

They show that the optimal solution π_r to the PPO objective above has an analytical form:

$$ \begin{aligned} \pi_r(y|x) &= \frac{1}{Z(x)} \pi_\text{ref}(y|x) \exp\left(\frac{1}{\beta} r_\phi(x, y)\right)\ Z(x) &= \sum \limits *{y |in |mathcal{Y}} |pi*\text{ref}(y|x) \exp\left(\frac{1}{\beta} r_\phi(x, y)\right)\ \end{aligned} $$

where $Z(x)$ is a partition function. This equation shows that the optimal RLHF policy is just the original SFT policy, re-weighted by the exponentiated reward from the RM. More specifically, this result shows that for any preference, there **exists an optimal LLM that is dominant to other LLMs**.

Authors use this relationship to re-parameterize the reward model's loss function. They can express the reward r_ϕ in terms of the policies π_r and π_ref. By plugging this back into the original RM ranking loss, they derive a new objective that optimizes a single policy π_θ *directly* on the preference data $\mathcal{D}$. By substituting the reward functions in the BT model with the result given above, **the partition function is simply canceled out**, removing the worry of intractability.

This new objective is the **DPO Loss Function**:

$$ \mathcal{L}|*text{DPO} = - |*mathbb{E}*{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_\text{ref}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_\text{ref}(y_l|x)} \right) \right] $$

Here, π_ref is just the initial SFT policy (the reference model). This elegant loss function has a clear interpretation: it's a log-likelihood loss that directly maximizes the probability of the preferred response y_w and minimizes the probability of the rejected response y_l, weighted by how far each is from the reference policy.

There is no more explicit RM, no PPO, and no sampling. The SFT model is simply fine-tuned on this new loss function, and the "reward model" is implicitly optimized at the same time.

{:#Results}

The Results

The authors tested DPO on several tasks, including summarization on the TL;DR dataset and dialogue generation on the Anthropic Helpful, Harmless, and Honest (HHH) dataset.

They compared DPO to traditional PPO-based RLHF, as well as other simplified methods. They show that their method is equivalent or dominant to the previous methods.:

1. **Performance:** DPO matched or exceeded the performance of PPO-based RLHF in terms of final response quality, as judged by human evaluators and GPT-4.
2. **Simplicity & Stability:** DPO was far more stable, easier to implement, and computationally cheaper. It completely removes the need to sample from the policy during training, fit a separate reward model, or manage the complex moving parts of PPO.
3. **Controllability:** DPO also offered better control over the KL-divergence, preventing the policy from "over-optimizing" the implicit reward and collapsing.

DPO achieved all the benefits of RLHF without the associated costs and instabilities.

{:#MyDisscussions}

Discussions & Questions

- **Strengths:**
    - **Elegant Simplicity:** The primary strength is its conversion of a complex, multi-stage, online RL problem into a single, stable, offline training problem. It feels like finding a "closed-form" solution to a problem everyone else was solving with brute force.
    - **Theoretical Grounding:** The derivation is clean and well-explained. The paper doesn't just present a new loss function; it shows *why* it's the correct loss function by deriving it directly from the established RLHF objective.
    - **Accessibility:** By removing the RL barrier, DPO made preference-based alignment accessible to the entire research community, not just large, well-resourced labs. This has sparked a massive wave of innovation (which we'll get to).
- **Limitations:**
    - **Offline Constraint:** As an offline method, DPO can only learn from the preferences in its static dataset. A potential advantage of online RL (like PPO) is its ability to explore and find *new*, high-reward responses that weren't in the original dataset. DPO gives up this (largely theoretical) exploration capability. Recent studies (Ren et al.,2025) showed that offline RLHF methods are more likely to degenerate or hallucinate due to the absence of the interactions.
- **Future Works:**
    - Because the DPO loss is so simple, it's easy to modify. This has led to follow-up work like **IPO (Identity-Preference Optimization)**, which improves stability, and **KTO (Kahneman-Tversky Optimization)**, which allows for training on "win/loss" labels for single responses, not just pairs.
    - DPO fundamentally proved that the path to alignment doesn't *have* to go through the complexity of reinforcement learning.

{:#Conclusion}

Conclusion

"Direct Preference Optimization" is an impactful paper that delivers on the promise in its title. It revealed that the explicit reward model, which the community saw as a cornerstone of RLHF, was "secretly" just an intermediate step that could be optimized away.

{:Reference}

Reference

- Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., & Finn, C. (2024). Direct Preference Optimization: Your Language Model is Secretly a Reward Model (No. arXiv:2305.18290). arXiv. https://doi.org/10.48550/arXiv.2305.18290
- Ren, Y., & Sutherland, D. J. (2024, October 4). Learning Dynamics of LLM Finetuning. The Thirteenth International Conference on Learning Representations. https://openreview.net/forum?id=tPNHOoZFl9