Journal of Machine Learning Research 23 (2025) 1-5 Submitted 1/21; Revised 5/22; Published 9/22

Supervised Learning Preference Optimization: Rethinking RLHF and DPO as Supervised Learning

Yaoshiang Ho YAOSHIANG@GMAIL.COM

Editor:

Abstract

Direct Policy Optimization (DPO) is a popular approach to aligning large language models with human preferences. In this paper, we analyze the underlying math and propose a new algorithm which we call Supervised Learning Preference Optimization (SLPO).

Keywords: Reinforcement Learning from Human Feedback (RLHF), Direct Policy Optimization (DPO)

1 Introduction

Alignment is the task of ensuring that the behavior of a Large Language Model (LLM) is consistent with human preferences.

A key difference between the alignment phase and other phases of training an LLM is that the alignment phase considers full sequences of text, rather than simply predicting the next token, as in the pretraining and supervised fine-tuning (SFT) phases.

The alignment approach popularized by the commercial success of ChatGPT was Reinforcement Learning from Human Feedback (RLHF, Ouyang et al. (2022)). Despite its effectiveness, RLHF requires training a second model, called a reward model, as well as Proximal Policy Optimization (PPO), resulting in a technique that is more complex than the basic supervised learning. It also requires a Kullback-Leibler (KL) divergence term to regularize the changes to the LLM during alignment training.

Direct Policy Optimization (DPO) is a simpler approach to alignment which does not require a secondary reward model. In the paper introducing DPO, the authors examine the underlying approach of RLHF and propose the DPO objective to align the target LLM directly using maximum likelihood estimation (MLE). The key insight from the DPO paper is that an LLM's outputs can be reparameterized into a reward model using ratios, logs, and the Bradley-Terry model Bradley and Terry (1952)

The specific contribution of this paper is to reframe the alignment phase away from reward modeling entirely and treating it simply as a pure supervised learning problem by training a model to align to a directly modified probability distribution. We call this approach Supervised Learning Preference Optimization (SLPO).

2 Related Work

Related work...

3 Preliminaries

We review and continue the analysis of the DPO objective by its author.

The DPO objective is defined as follows:

$$L_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = \underbrace{-\mathbb{E}_{(x,y_w,y_l)\sim D} \log}_{1} \left[\underbrace{\sigma}_{2} \left(\underbrace{\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)}}_{3} - \underbrace{\beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}}_{4} \right) \right]. \quad (1)$$

In the underbraced section 1, we see the standard negative log likelihood (NLL) objective. In the underbraced section 2, we see the Bradley-Terry model [1]. In the underbraced sections 3 and 4, we see how the reference and language model's predictions are reparameterized into a winning and losing score: they are the log of the ratio of the language model to the reference model, for the winning and losing completion, respectively. This score is later described as the reward function.

With simple algebraic manipulation, we can rewrite this objective as:

$$L_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = \underbrace{-\mathbb{E}_{(x,y_w,y_l)\sim D} \log}_{1} \left[\underbrace{\sigma}_{2} \left(\underbrace{\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}}_{3} - \underbrace{\beta \log \frac{\pi_{\text{ref}}(y_w \mid x)}{\pi_{\text{ref}}(y_l \mid x)}}_{4} \right) \right]. \quad (2)$$

We can interpret the undercomponents as follows: The first underbrace is the NLL, as before. The second underbrace is the Bradley-Terry model, as before. The third underbrace is the log of the ratio language models probability of the winner divided by the loser. This ratio is optimized to be bigger, given that the log, sigmoid, and log from underbraces 1, 2, and 3 are all monotonic functions. Since a ratio of probabilities is an odds ratio, we will refer to this as the language model's log-odds ratio.

The fourth underbrace is reference model's log-odds ratio. This is a constant per y_w and y_l and is not differentiated. However, these values and their ratio vary across different y_w and y_l - that is, each row of training data will have a different value for this constant. More specifically, since π_θ is initialized as π_{ref}, the difference between underbraces 3 and 4 starts at zero during training, and the shape of the sigmoid function (underbrace 2) and its gradient are known. During training, the language model's log-odds ratio will increase, however, the sigmoid function will naturally regularize and decelerate the increase. This is the exact outcome described by the DPO authors in their analysis of the gradient of DPO. *But we have developed a different intuition the DPO loss: rather than reparameterizing the language model's output into a reward model, we are simply regularizing the optimization of the language model's log-odds ratio.*

1. A ranking method that is mathematically equivalent and perhaps more widely understood is the ELO score, used to rank Chess players, and, LLMs in the Chatbot Arena Elo (1978); Chiang et al. (2024). Both ELO and Bradley-Terry assign scores to players, and pass the difference through a sigmoid function to assess the probability of the LHS player of winning.

Let's analyze the regularization effect. The DPO authors investigated the gradient of the sequence of tokens... let's go two steps further by considering each token individually, and the logit behind it.

Our variable y is a sequence of tokens. More formally, it is

$$\pi_{\text{ref}}(y \mid x) = \prod_{t=1}^{T} \pi_{\text{ref}}(y_t \mid x, y_{<t}), \tag{3}$$

where $y = (y_1, y_2, \ldots, y_T)$ is the output sequence, x is the input context, and $y_{<t} = (y_1, y_2, \ldots, y_{t-1})$ represents the tokens generated prior to time step t. The term $\pi_{\text{ref}}(y_t \mid x, y_{<t})$ denotes the conditional probability of generating token y_t given the input x and the previously generated tokens $y_{<t}$.

Since both π language model are LLMs, they are activated using the softmax function. Logits can be normalized using the log-softmax function, allowing them to be exponentiated to yield the probability of the token, avoiding the need for a softmax function and its reliance on the entire probability distribution.

Let us establish the term g for the layers of the model up to the softmax activation, namely, the feature extractor and log-softmax normalization:

$$\pi(y \mid x) = \exp(\mathrm{g}(y \mid x)) \tag{4}$$

where

$$\mathrm{g}(y \mid x) = \text{logsoftmax}(\mathrm{f}(y \mid x)) \tag{5}$$

Plugging Equations 3 and 5 into the DPO objective, with a focus just on the score inside the sigmoid of the Bradley-Terry model, we have:

$$\beta \log \frac{\prod_{t=1}^{T_w} \exp(\mathrm{g}_\theta(y_{w,t} \mid x, y_w{<}\, t))}{\prod_{t=1}^{T_w} \exp(\mathrm{g}_{\text{ref}}(y_{w,t} \mid x, y_w{<}\, t))} - \beta \log \frac{\prod_{t=1}^{T_l} \exp(\mathrm{g}_\theta(y_{l,t} \mid x, y_w{<}\, t))}{\prod_{t=1}^{T_l} \exp(\mathrm{g}_{\text{ref}}(y_{l,t} \mid x, y_w{<}\, t))}$$

$$\beta \left(\sum_{t=1}^{T_w} [\mathrm{g}_\theta(y_{w,t} \mid x, y_{w,<t}) - \mathrm{g}_{\text{ref}}(y_{w,t} \mid x, y_{w,<t})] - \sum_{t=1}^{T_l} [\mathrm{g}_\theta(y_{l,t} \mid x, y_{l,<t}) - \mathrm{g}_{\text{ref}}(y_{l,t} \mid x, y_{l,<t})] \right)$$

This formulation of the DPO objective provides additional insight. Each logit of the language model, which was originally conceived to predict the probability of a token, has been normalized by the logit of the reference model via a shift (this normalization should not to be confused with the normalization by the log-softmax function, which was purely for mathematical convenience). The sum of these normalized logits is then passed through a standard sigmoid activation and negative log likelihood function.

The starting value for the difference between the language model and the reference model's logits is zero, since the language model is initialized by the reference model, so the gradient passed to each logit on the first step of optimization is -0.5 for the winning token and 0.5 for the losing token. As the winning sequence increases its odds, and the losing sequence decreases its odds, the the gradient will decrease, as is well known for the negative log loss applied to a sigmoid activated function (e.g. classic binary crossentropy loss for

logistic regression). Although sigmoid is not an additive function, e.g. $\sigma(a+b) \neq \sigma(a)+\sigma(b)$, essentially as the logit of each winning token increases and each losing token decreases, the magnitude of the gradient applied to all logits decreases.

And yet... the DPO equation has a lot of machinery for what ends up being a standard negative log likelihood loss, applied to a normalized logit. Is there a way to further simplify the normalization, eliminating the need for the Bradley-Terry model, the log of ratios, and the concept of reward functions?

4 Supervised Learning Preference Optimization

Section body

Here is a citation Chow and Liu (1968).

5 Results

Section body

Here is a citation Chow and Liu (1968).

6 Conclusion

Section body

Here is a citation Chow and Liu (1968).

Acknowledgments and Disclosure of Funding

The author thanks Chiara Cerini, Peter Tran, and Sam Wookey for their invaluable reviews of early drafts of this work.

Appendix A.

[appendix]

Appendix B.

[appendix] **Proof**. We use the notation:
Remainder omitted in this sample. See http://www.jmlr.org/papers/ for full paper.

References

R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. doi: 10.2307/2334029. URL `https://doi.org/10.2307/2334029`.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*, 2024.

C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, IT-14(3):462–467, 1968.

Arpad E. Elo. *The Rating of Chessplayers, Past and Present*. Arco Publishing, New York, 1978. ISBN 9780668047210.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.