Intermediate direct preference optimization

Atsushi Kojima
Money Forward, Inc.
kojima.atsushi@moneyforward.co.jp

Abstract

We propose the intermediate direct preference optimization (DPO) method to calculate the DPO loss at selected intermediate layers as an auxiliary loss for fine-tuning large language models (LLMs). The conventional DPO method fine-tunes a supervised fine-tuning (SFT) model by calculating the DPO loss using logits from the final layer. In our intermediate DPO approach, DPO losses are calculated using the logits from K-selected intermediate layers and averaged to obtain the intermediate DPO loss. For training the intermediate DPO model, the final loss is obtained by calculating the weighted sum of the DPO and intermediate DPO losses. During inference, the intermediate DPO model decodes using the final layer logits similarly to the conventional DPO model. In experiments using the ultrafeedback dataset, the performance of the intermediate DPO model was evaluated using GPT-4. As a result, the intermediate DPO model trained using the intermediate DPO loss calculated at the 22nd layer of a 32-layer SFT model achieved win rates of 52.5% and 67.5% against the conventional DPO and SFT models, respectively, demonstrating the effectiveness of the proposed method. Furthermore, we report the relationships among the position of the selected intermediate layers, the number of layers, and performance.

1 Introduction

Preference optimization methods [1–5], which align large language models (LLMs) with human preference, have been recently explored. Among such methods, direct preference optimization (DPO) is the most promising and efficient method, because it can fine-tune a model without training a reward model or complex reinforcement algorithms such as proximal policy optimization [5].

A DPO model can be obtained by fine-tuning a supervised fine-tuning (SFT) model based on DPO loss. The DPO method encourages the SFT model to learn that the desirable completion is preferred over the rejected completion for a given prompt. DPO loss is calculated using logits obtained from the final layer of the SFT model.

In this paper, we propose the intermediate DPO method to calculate the DPO loss at selected intermediate layers as an auxiliary loss. In our intermediate DPO approach, DPO losses are calculated using the logits from K-selected intermediate layers and averaged to obtain the intermediate DPO loss. For training the intermediate DPO model, the final loss is obtained by calculating the weighted sum of the DPO and intermediate DPO losses. The parameters of the linear layers used to convert the hidden vectors to the logits are shared at each intermediate layer. In experiments using the ultrafeedback dataset[1], we evaluated the performance of the intermediate DPO model using GPT-4. The results showed that the intermediate DPO model trained using the intermediate DPO loss calculated at the 22nd layer of a 32-layer SFT model achieved win rates of 52.5% and 67.5% against the conventional DPO and SFT models, respectively, demonstrating the effectiveness of the proposed method.

---

[1]HuggingFaceH4/ultrafeedback_binarized

2 DPO

The DPO model is obtained by fine-tuning an SFT model. The DPO loss formula is as follows Equation (1):

$$\lambda_{\mathrm{DPO}} = -\log \sigma(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\mathrm{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\mathrm{ref}}(y_l|x)}). \tag{1}$$

Here, x is the prompt, and y_l and y_w represent the undesirable and desirable completions, respectively. π_θ and π_{ref} represent the optimal and reference policies (SFT), respectively. σ is the sigmoid function, and β represents a hyperparameter.

3 Intermediate DPO

In this paper, we propose calculating the DPO loss at intermediate layers as an auxiliary loss to enhance the performance of the DPO model. The conceptual overview of the proposed method is shown in Figure 1. In the proposed method, logits are calculated from the hidden vectors of intermediate layers and then the DPO loss is computed. The parameters of the linear layers used to convert the hidden vectors to the logits at each intermediate layer are shared.
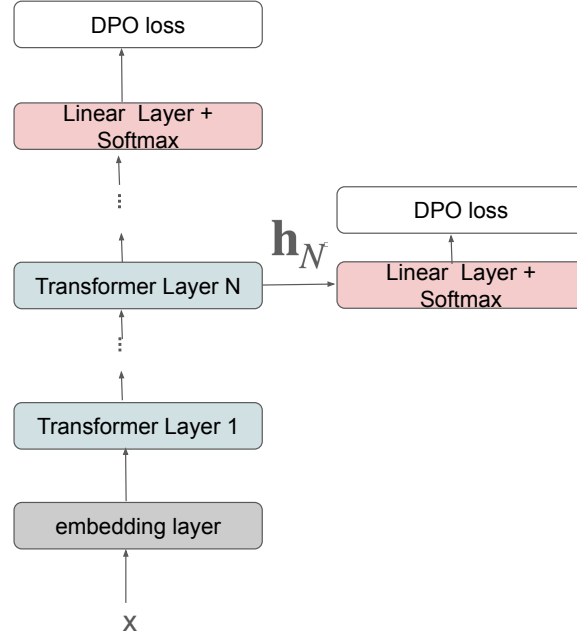

Figure 1: Overview of intermediate DPO

The formula for the intermediate DPO loss is shown in Equation (2), where K denotes the indices of the selected intermediate layers. The intermediate DPO loss is obtained by averaging the DPO losses computed at the intermediate layers. For example, if $K = 5, 8$, the DPO loss is calculated by using logits from the 5th and 8th layers and then averaging these values.

$$\lambda_{\mathrm{DPO}_{\mathrm{intermediate}}} = \frac{1}{|K|} \sum_{k \in K} \lambda_{\mathrm{DPO}_k}. \tag{2}$$

To train the model using the intermediate DPO loss as an auxiliary loss function, we compute the weighted sum of DPO loss. The final loss for training the model is shown in Equation (3), where γ is a hyperparameter.

$$\lambda = \gamma \lambda_{\mathrm{DPO}} + (1 - \gamma)\lambda_{\mathrm{DPO}_{\mathrm{intermediate}}}. \tag{3}$$

During inference, the intermediate DPO model calculates posterior probabilities from the logits of the output layer and decodes similarly to the conventional DPO model.

4 Experiments

4.1 Experimental Conditions

All experiments were conducted using the ultrafeedback dataset. For the training of the intermediate DPO model, the entire training dataset was used, and for evaluation, 40 randomly selected prompts from the test dataset were utilized. The evaluation was conducted using GPT-4.

The intermediate DPO model was trained by fine-tuning a 7B SFT model (32-layer model)[2] by low-rank adaptation (LoRA) [6]. The training conditions for the intermediate DPO model are shown in Table 1. The hyperparameter in the proposed method was set at $\gamma = 0.9$.

Table 1: Training conditions for intermediate DPO model

Parameter	Value
Adapter position	k_proj, q_proj, v_proj, o_proj, gate_proj, down_proj, up_proj
Lora attention dimension	16
α for LoRA scaling	16
Dropout rate for LoRA	0.05
Learning rate	5e-4
β	0.1
γ	0.9
Batch size	32

Inference was performed by temperature sampling. We set to a temperature of 0.3 and a repetition penalty of 1.1.

4.2 Results

Figure 2 shows the win rates of the intermediate DPO model against the SFT and DPO models. The layers at which the intermediate DPO loss was calculated were set to $K = 11, 22$. The win rates of the intermediate DPO model against the SFT and DPO models were 67.5% and 52.5%, respectively, indicating that the intermediate DPO model outperforms the conventional DPO model.
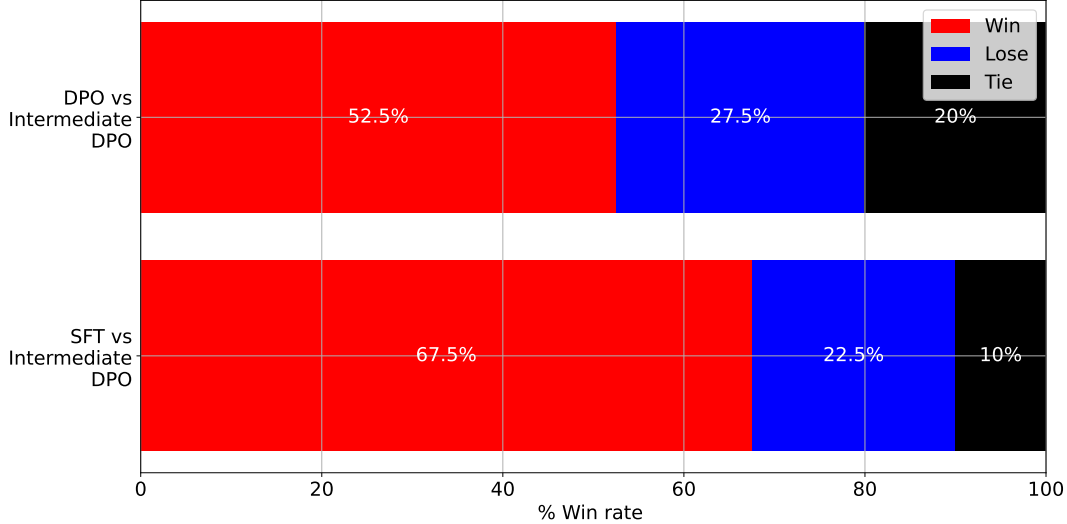

Figure 2: Win rates of the intermediate DPO model against the SFT model and DPO model ($K = 11, 22$)

---

[2]mistralai/Mistral-7B-Instruct-v0.1

Table 2 shows the impact of the layer depth at which the intermediate DPO loss is calculated on accuracy. The numbers in the table indicate the win rates of the intermediate DPO model against the SFT model. The performance of the model in calculating the DPO loss at layers closer to the output (id5) was observed to be higher than those in calculating the DPO loss in the middle (id9) or closer to the input layers (id4), suggesting that calculating the intermediate DPO loss closer to the output layer improves performance.

Table 2: Impact of layer depth at which the intermediate DPO loss is calculated on accuracy (32-layer model)

Exp ID	K	Win rate	Lose rate	Tie
id4	11	52.5	27.5	20
id9	16	50	27.5	22.5
id5	22	60	15	25

Table 3 shows the impact of the selection method on the accuracy when calculating DPO loss from multiple intermediate layers. The numbers indicate the win rates of the intermediate DPO model against the SFT model. Comparing the results of selecting from one layer (id4, id5, and id9) with those from two layers (id3 and id10), shows that increasing the number of layers from one to two improves the win rate, indicating that selecting multiple layers to calculate intermediate DPO loss is beneficial.

Table 3: Impact of selection methods on accuracy

Exp ID	K	Win rate	Lose rate	Tie
id4	11	52.5	27.5	20
id5	22	60	15	25
id9	16	50	27.5	22.5
id3	11, 22	67.5	22.5	10
id10	16, 22	67.5	22.5	10

Furthermore, since calculating intermediate DPO loss on all layers is computationally expensive, the effect of a layer selection method on the accuracy was investigated. Table 4 shows a comparison of two methods of selecting intermediate layers: selecting dispersed layers across the network (id3) and choosing consecutive layers closer to the output (id7). The numbers in the table indicate the win rates of the intermediate DPO model against the SFT model. The numbers indicate the win rates of the intermediate DPO model against the SFT model. Selecting from spread out layers (id3) performed better than selecting continuously from layers closer to the output (id7), indicating that a spread out selection method is effective in the proposed intermediate DPO approach.

Table 4: Impact of layer selection methods on accuracy

Exp ID	K	Win rate	Lose rate	Tie
id3	11, 22	67.5	22.5	10
id7	30, 31	65	22.5	12.5

4.3 Discussion

To verify the effect of the proposed method, we analyzed the likelihood of DPO across all layers of the intermediate DPO model given by $\log \sigma(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)})$. The analysis was conducted by randomly selecting 30 prompts from the ultrafeedback test dataset and calculating their average.

Figure 3 shows a comparison of likelihoods between the DPO model and the intermediate DPO model (id5) trained with settings K to 22.

In this figure, the horizontal axis represents the intermediate layer indices and the vertical axis represents the DPO likelihood. Since in the proposed method the intermediate DPO loss is calculated
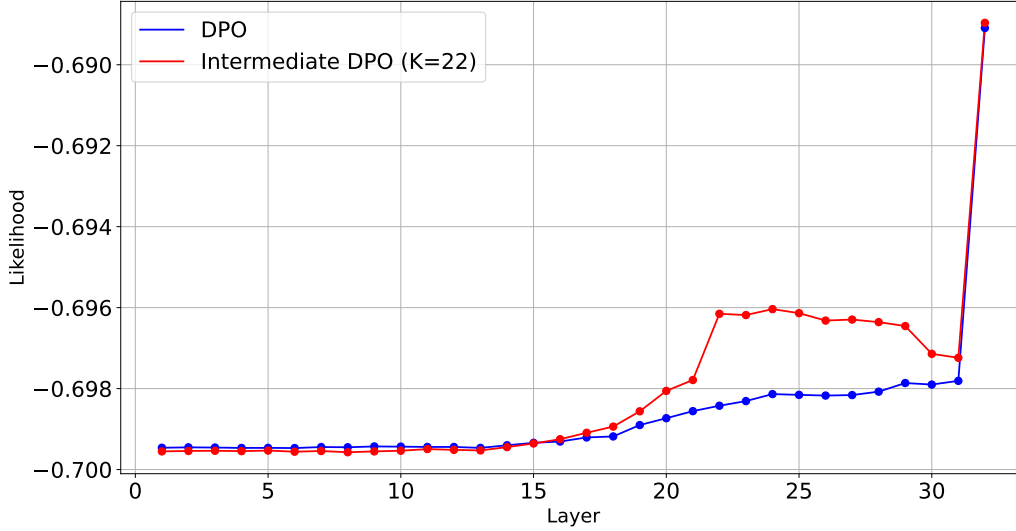
Figure 3: Comparison of likelihood from intermediate layers

at the 22nd layer, a peak can be observed at this layer. Furthermore, owing to the effect of the proposed method, the likelihood in the output layer of the intermediate DPO model is higher than that of the DPO model, which is considered to contribute to the improved performance.

5 Conclusion

In this paper, we proposed calculating the DPO loss at intermediate layers as an auxiliary loss when fine-tuning LLMs based on DPO, referred to as intermediate DPO. The intermediate DPO loss is obtained by calculating and averaging the DPO losses using logits from the selected K-intermediate layers. When training the model, the final loss is obtained by computing the weighted sum of the DPO and intermediate DPO losses. In experiments using the ultrafeedback dataset, the performance of the intermediate DPO model was evaluated using GPT-4. The results showed that the proposed method achieved win rates of 52.5% against the DPO model and 67.5% against the SFT model, demonstrating the effectiveness of the proposed method. In our future work, we will investigate the effectiveness of the proposed method across different model sizes.

References

[1] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In Thirty-seventh Conference on Neural Information Processing Systems, 2023.

[2] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. arXiv preprint arXiv:1909.08593, 2019.

[3] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. Advances in Neural Information Processing Systems, 33:30083021, 2020.

[4] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, C. Chen, C. Olsson, C. Olah, D. Hernandez, D. Drain, D. Ganguli, D. Li, E. Tran-Johnson, E. Perez, J. Kerr, J. Mueller, J. Ladish, J. Landau, K. Ndousse, K. Lukosuite, L. Lovitt, M. Sellitto, N. Elhage, N. Schiefer, N. Mercado, N. DasSarma, R. Lasenby, R. Larson, S. Ringer, S. Johnston, S. Kravec, S. E. Showk, S. Fort, T. Lanham, T. Telleen-Lawton, T. Conerly, T. Henighan, T. Hume, S. R. Bowman, Z. Hatfield-Dodds, B. Mann, D.

Amodei, N. Joseph, S. McCandlish, T. Brown, and J. Kaplan. Constitutional AI: Harmlessness from AI feedback, 2022.

[5] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

[6] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685, 2021.