The 7th International Conference on Interdisciplinary Research
on Computer Science, Psychology, and Education (2023)

Which is Better Training for Reward Model, by Ranking or Classification?

Jeongwook Kim, Imatitikua Danielle Aiyanyo
Department of Computer Science and Engineering, Korea University
{k0s1k0s1k0, titi}@korea.ac.kr

Abstract

The Reinforcement Learning from Human Feedback (RLHF) methodology has recently been applied to many high-performance language models. It utilizes a reward model and human feedback to help language models produce responses that are more likely to be preferred by humans. However, when it comes to RLHF applied to commercial large language models, it is not clear exactly how it is implemented. In particular, how to set up the reward model, which is responsible for the environment in reinforcement learning, is the most important aspect, but open-sourced models have different implementations. In this study, we test which method is more efficient for the two main branches of reward model training: ranking-based training method and classification-based training method. We also estimate the reasons for the difference in efficiency based on the analysis of the experimental results.

Keywords: Reinforcement Learning from Human Feedback (RLHF), Reward model

1. Introduction

Reinforcement learning is a method for training an agent to choose actions that maximize its reward, usually in an environment with rewards. In order to train a language model with reinforcement learning, the environment must be well defined. A recent application of reinforcement learning that has led to high performance in chat-enabled language models is Reinforcement Learning from Human Feedback (RLHF). The RLHF methodology introduces a reward model that rewards model responses that are more likely to be preferred by humans, and applies additional reinforcement learning training after cross-entropy-based training. Models using this RLHF methodology are not only capable of understanding and producing language, but also of generating responses that are more likely to be preferred by humans. However, compared to the number of modern commercial language models that apply reinforcement learning, the number of studies is very small. ChatGPT [1], a representative commercial language model, was announced to have applied a methodology similar to InstructGPT [2], but the specific implementation method was not specified, and the technical report of GPT-4 only presented the performance of the model and did not disclose any information about the creation of the model. Therefore, the objective function, hyperparameters, and data used to train the reward model in the reinforcement learning phase were not disclosed. The reason why it is difficult to apply reinforcement learning to language models is that reinforcement learning is more sensitive to the environment and hyperparameters than general training methods based on cross-entropy, so it is more difficult for general researchers who lack information.

In this study, we experiment with effective training methods for reward models and provide a theoretical basis for our results. There are two main training methods for reward models. The first is the ranking-based method, in which the responses generated by the model under training are viewed by a human and ranked from the best to the worst, which was used in InstructGPT. The second is a binary classification-based training method, as proposed by LLAMA-2 [3], CarperAI [4], and others, which trains like a model that classifies preferred and non-preferred responses. Reward models trained with either method are used as the environment for reinforcement learning, where a scalar reward is given for the output produced by the language model, just like a hypothetical person. Based on the reward values given by the reward model, the language model is trained to

increase the sum of the total reward values using a reinforcement learning algorithm. The Proximal Policy Optimization (PPO) [5] algorithm is the most popular modern reinforcement learning algorithm, which can train a language model as an actor to increase a given reward.

2. Methodology

Commercially available reward model training methods can be divided into two main categories. They are ranking-based methods and classification-based methods. In this study, we implement and experiment with both methods under the same conditions to present the features of each method and how to train a better reward model. In the RLHF of commercial language models, human judgment of model generation results is used, but it is expensive, so ChatGPT and prompts are used to replace human judgment. Recent studies such as QLoRA [6], LLAMA-2, and G-EVAL [7] have concluded that ChatGPT is a good substitute for humans in model evaluation because it produces quality evaluations comparable to humans, and even more consistent than humans.

2.1 Comparison group 1: Training a rank-based reward model

In rank-based training, we collect k different responses from the SFT model for the same input x. We then rank the k responses in order of human preference. Next, we choose two of the k responses. Consider one as good response y and the other as bad response $y\prime$ according to the ranking. This results in a total of ${}_kC_2$ training data D. Within D, we train to generate higher rewards for good responses and lower rewards for bad responses. The loss function for the reward function model r_θ parameterized by θ is defined as follows.

$$\text{loss}(r_\theta) = -E_{(x,y,y\prime)\sim D}[\log(\sigma(r_\theta(x,y) - r_\theta(x,y\prime)))]$$

Specifically, we set $k = 4$, which means that we sampled four different responses to the same model's input and prompted ChatGPT to rank them in order of the best response. Thus, six training examples were generated for one x. In our experiments, we found that the six generated training examples all shared the same x, leading to rapid overfitting even though we only trained on one width. To solve this problem, we set up the environment so that all 6 training examples are in the same minibatch as suggested by [2]. We also fixed the ChatGPT API version to GPT-4-0613 to prevent the environment from changing in the middle of the experiment.

2.2 Comparison group 2: Training a classification-based reward model

In classification-based training, you train the model to binary classify good and bad responses. In a pre-prepared dataset D, good responses are labeled 1 and bad responses are labeled -1. The model is trained to classify good and bad responses using typical supervised learning methods. Fitting a sigmoid function to the logit values given by the trained model outputs values in the interval $(-1, 1)$. This value is later utilized as the reward value in the PPO algorithm.

In our experiments, we considered the set of correct answers on a given dataset as good responses and the responses generated by the SFT model as bad responses. Therefore, the reward model was trained to reward closer to the correct answers. The reward models trained by both methods were normalized after training to have a mean of zero and a variance of one.

2.3 Dataset for experiment

We used FairytaleQA as our experimental dataset. The FairytaleQA dataset consists of fairy tale prints for children and their question and answer pairs. The dataset consists of 10580 fingerprint-question-answer pairs from a total of 278 books. The model is tasked with reading a given fairy tale from the FairytaleQA dataset and generating questions. The FairytaleQA dataset is characterized by the fact that experts in the field of education have created high-quality questions that require reasoning to develop children's reading comprehension skills. Therefore, the model is required to generate questions of appropriate difficulty, not just simple questions. During the reinforcement learning phase, the reward model is adjusted to reward more educational, reasoning, and grammatically correct questions.

3. Experiment

The reward model trained based on classification did not work effectively. In addition, the model received lower rewards from the beginning of training compared to the ranking-based method. We would like to extrapolate from our experiments why the classification-based method was less effective than the ranking-based method. When training the

reward model with the classification-based method, the reward model is trained to consider the responses generated by the SFT model as bad responses and give them a reward close to -1. The RL model is trained to reward responses close to +1 when they are produced like the correct response sentences in the dataset. Therefore, the RL model will continue to receive responses close to -1 from the reward model. Therefore, when reinforcement learning with a classification-based reward model, it will continue to receive low rewards. In the PPO algorithm, the behavioral model is tuned to receive relatively high rewards, and if it continues to receive low rewards, it will encourage unintended token generation, which in turn increases the randomness of the model.

4. Conclusion

In this study, we investigated how to train an effective reward model for RLHF tuning, which has recently been used to train high-performance language models. The experimental results show that the ranking-based reward model, which has been applied to ChatGPT and InstructGPT, is a more effective method. However, the ranking-based reward model training method requires a lot of work for humans to rank the model's responses, so it is an expensive method and difficult to conduct at the general research level. Therefore, in this study, we fixed the prompts of ChatGPT and conducted experiments under the assumption of a consistent human. However, training a classification-based reward model is not without its drawbacks. It is a training method that is worthy of further research because it can be used to create a reward model from a given dataset without any human intervention. The number of studies on RLHF in academia is small compared to the continuous application of RLHF methodology in commercial high-performance language models. In addition, while RLHF methodology requires complex training steps and many hyperparameters, commercial models do not disclose the detailed training process, making it difficult for general researchers to access. We hope that this study will stimulate further research on RLHF methodology.

Acknowledgements

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2022-2018-0-01405) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation). This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2021R1A6A1A03045425).

Reference

[1] OpenAI-Blog, "Chatgpt: Optimizing language models for dialogue," 2022. [Online]. Available: https://openai.com/blog/chatgpt/

[2] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," 2022.

[3] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom, "Llama 2: Open foundation and fine-tuned chat models," 2023.

[4] CarperAI, "trlx," 2023. [Online]. Available: https://github.com/CarperAI/trlx

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.

[6] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," 2023.

[7] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, "G-eval: Nlg evaluation using gpt-4 with better human alignment," 2023.