Large Language Model Fine Tuning For GPT-2

121090386 Lu Yi

May 2024

Abstract

Large language model has been a cutting-edge technology in artificial intelligence history with rapidly increasing numbers of paper published in this area. It has a promising future and more applications for human beings. In this independent study with the topic of large language model, I conducted a research on the fine tuning process for a given GPT-2 model with LoRA algorithm. GPT-2, as an early version of GPT, was pre-trained with relatively smaller numbers of parameters(1.5 billion parameters) so that it required less expensive computation resources compared with more advanced and complicated LLM. I adopted it as a toy example to carry on the task. The main task in this study is to conduct fine-tuning on the given pre-trained GPT-2 model, test its performance, and derive the conclusion of the role of fine tuning for the final LLM performance.

1 Introduction

In general, the large language model, abbreviated as LLM, aims to generate human-like texts according to the training from a huge amount of training texts. As stated by Ewald(2023), "Large language models are a subset of deep learning, as a kind of generative AI." As indicated by the name, the training dataset and number of parameters are often quite large so that it can include the major information of human language grammar and commonality in various occasions.

During the training procedure, the model gradually learns the basic knowledge with vast amount of training text data. Similar to common machine learning process, given a loss function for predicted tokens and true tokens, the model learned the parameters from the training dataset so as to minimize the loss and fit the training data. More specifically, given the input text, the model derives the probability distribution of the next word and selects word the most likely to occur iteratively. Finally, it can generate the most appropriate language response corresponding to the given prompts.

1.1 GPT-2

GPT, abbreviated for "Generative Pre-trained Transformer", is a type of deep learning model that predicts the next word of the output sentence based on the previous words, so that it can generate desirable texts corresponding to the prompt iteratively. One pre-released GPT-2 model was considered in this study, which was produced by Open AI in 2019 as the successor to GPT. It is a model with 1.5 billion parameters, trained on a dataset of 8 million web pages. The training data include texts from a variety of fields that is close to human life so that the generated dialogues can be quite coherent for as long as a page of text or even more. This model is relatively elementary currently because of its smaller number of parameters and shorter training time. However, the drawbacks of this model are also quite obvious. Several kinds of failures occur from time to time in the text generation process such as repetitive text, world modeling failures and unnatural topic switching, especially when it is asked to generate long passages(Barry, 2019).

1.2 Tokenization

Before stepping into the main process of the training, we firstly need to do some pre-processing for the given training and testing data, called tokenization. This step parses the text into non-decomposing units called tokens, which can be characters, subwords, simbols or words(Naveed et al., 2023). Then the model will treat the transformed tokens as the input training dataset for future model fitting. The tokenization can be understood as a transformation from given texts to a sequence of numbers so that the model receive them as input information, build a reasonable model to fit these data and generate the most appropriate token for the texts to be generated. Finally, it will detokenize the output tokens into texts.

1.3 Training

Obviously, the core part of generation lies in the model training based on input tokens. There are two main stages for the construction of one large language model after pre-processing, pre-training and fine-tuning. Both of them are quite critical for the final performance.

1.3.1 Pre-training

Pre-training process often requires super expensive computation resources since the model is trained in a self-supervised manner on a large corpus to predict the next tokens given the input(Naveed et al., 2023). In this step, we often require training data from various occasions, and it can be considered for the model to learn about the human language rules for the first time. It equips the model with a comprehensive grasp of language, encompassing grammar, world

knowledge, and rudimentary reasoning. The model learns about the most fundamental rules of human languages so that the generated text can be at least understandable, if not perfect, by human beings after the whole process of pretraining.

1.3.2 Fine tuning

As stated by Naveed et al.(2019), "Pre-trained LLMs have excellent generalization abilities to unseen tasks. However, because they are generally trained with the objective of next token prediction, LLMs have limited capacity to follow user intent and are prone to generate unethical, toxic or inaccurate responses." That is why fine tuning stage is also indispensable for one mature LLM. This process requires a dataset on some specific fields with much smaller sample size, and it is designed to help the model generate more accurate texts in terms of the prompts in those specific fields while preserving its general language knowledge. There are several types of different fine-tuning methods, including transfer learning which focuses on task-specific data, instruction tuning which contains instruction formatted data and alignment-tuning which aims to make the generated texts helpful, honest, and harmless(Ewald et al., 2023). In this study, I mainly focused on transfer fine-tuning with labeled data.

1.4 LoRA Algorithm

LoRA method(Low-Rank Adaptation) was adopted for the fine-tuning process. During the fine tuning procedure, the neural network contains many dense layers, which makes it super expensive to update all the parameters simultaneously. As a result, we consider decomposing the weight matrix and only updating a part of them.

LoRA method relies on the hypothesis that the learned over-parameterized models reside on a low intrinsic dimension, so that the updates for weight matrices possess an intrinsic rank as well. The main idea of LoRA is that only some layers are updated while others are frozen, which can significantly improve the training efficiency(Hu et al., 2019). The number of trainable parameters for downstream tasks are greatly reduced.

For a pre-trained weight matrix $W_0 \in R^{d\times k}$, constrain its update by representing the latter with a low-rank decomposition $W_0 + \Delta W = W_0 + BA$, where ΔW is the accumulated gradient update during adaptation. $B \in R^{d\times r}, A \in R^{r\times k}$, and $r \ll k$, as a result, we will replace ΔW by BA to save memory and improve training efficiency.

1.5 Prompt

After the whole training process of LLM, we simply need to input a prompt to which the model is asked to respond. It can be any word, sentence, paragraph that provides a context for the conversation. It can be understood as requirement from human for the model, and the text quality generated afterwards can be considered as the generation performance of the model for evaluation intention.

The ultimate produced LLM can be applied in various fields, such as medicine, education, science, robotics(Naveed et al., 2023). The model with better generative ability can be much more helpful in practice. As a result, we always want to improve the dialogue ability of the model whenever pre-training a LLM or conducting fine-tuning in some specific areas.

2 Motivation

I want to investigate the role of fine tuning with LoRA method for GPT-2 as the pre-trained model in this study because of the following reasons.

Firstly, it is a critical part for the construction of the mature large language model. Successful fine tuning can not only significantly improve the quality of the output text of the large language model, but also only require reasonable memory storage and limited training time.

In addition, compared with huge amount of training data requirement in pre-training process, fine-tuning doesn't require a large amount of training resources so that it is easier to realize.

Lastly, as a earlier released LLM model, GPT-2 has weaker text generation capacity so that the improvement brought by fine-tuning can be more obvious.

Through this study, I mainly want to find out the specific improvement of fine tuning process for the ultimate LLM. I intend to evaluate and compare the text generation quality for the GPT-2 model with and without the fine tuning process for a clearer understanding of the role of fine tuning in LLM training.

3 Experiment

With a given GPT-2 model on the github website, I set up the conda configuration for the model and executed the code to conduct supervised fine tuning with Low-Rank Adaptation(LoRA) on kaggle. All the codes were executed on kaggle with the GPU P100 resources there, and the training loss for each step was recorded.

3.1 Datasets

I adopted Anthropic's "HH-RLHF" dataset for the dataset of fine-tuning, which consisted of 43835 samples for training dataset and 2354 samples for test dataset. This dataset is highly related to medicine therapy so that we expect the model after the supervised fine tuning would have better response performance with respect to medical suggestions.

In this dataset, the form of the data is basically the conversation between human and assistant. As a consequence, we expect that the model can have a better dialogue generating performance for some prompts in the related fields.

3.2 Loss function

For each step of the training, the loss is computed with the given loss function to evaluate the fitted model. We adopted cross entropy loss in this study, with the formula as $-log\frac{exp(logits_l)}{\sum_{i=1}^{C} exp(logits_i)}$.
We will focus on the change of loss function during the whole training process.

3.3 Evaluation Criteria

I adopted one well-trained model, "OpenAssistant/rewardmodel-deberta-v3-large-v2", to perform the auto-evaluation and judge the dialogue ability. It would output a score for each input generated dialogue. Higher score means more desirable response for the given response with respect to certain prompt.

4 Result

4.1 Loss value

Firstly, I searched for the loss of the pre-trained GPT-2 loss. Thanks to the experiment results of Lee & Hisang(2020), the training loss value is shown in figure 1. Then I conducted LoRA fine-tuning on the model and recorded the training loss again. I halted the training procedure around 60000 training iterations when the training loss roughly converges. Since the number steps of training is too large, I computed the average loss of every 1000 steps to represent the loss at that small period of the total training process.

As we can see from the output(figure 1&2), the loss function decreases with the increasing of training steps, and finally it decreases to a desirable value and converges. With the help of fine tuning, the dialogue ability of the model is significantly improved since the loss after convergence is lower.

Besides, I computed the score for generated texts from both the GPT-2 with and without fine tuning from open assistant auto-evaluation. I selected 50 questions

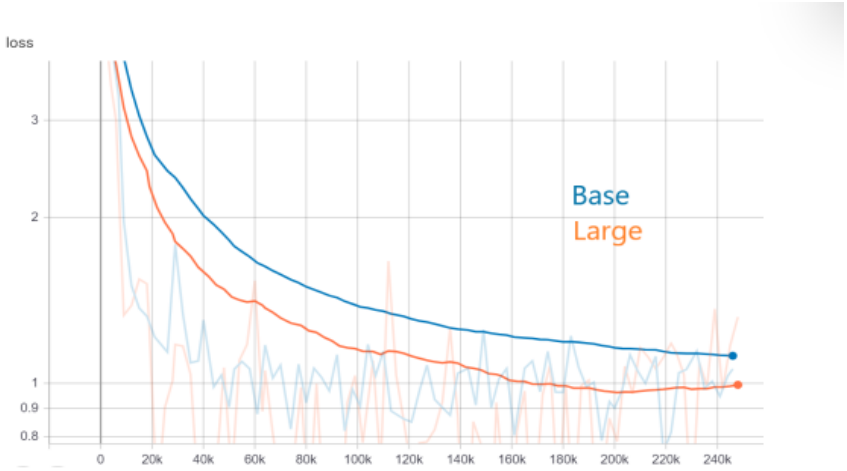

Figure 1: Loss vs iteration number for pre-trained GPT-2

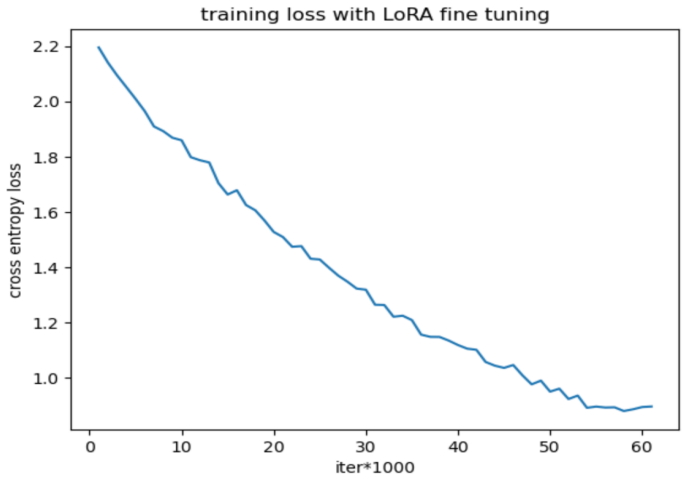

Figure 2: Loss vs iteration number after fine-tuning

from the testing dataset and got the answers for both the GPT-2 model with and without fine-tuning, finally I computed the average score for each model. The average score for the GPT-2 model without fine tuning is -0.6233, while the average score for the GPT-2 model without fine tuning is -0.5658. It turns out that the evaluation score generated is also higher for the model with fine tuning. The results of score are not quite objective and convincing because of the small testing number and dubiety of evaluation criteria. In fact, there is no absolute metric to evaluate one response corresponding to a specific question. So the scores here can only be referred to as a reference.

In general, the model performance improves with fine-tuning method in terms of those specific fields.

5 Discussion

Above all, this study provided some basic introduction for large language model and tested the role of fine tuning for the LLM training effect. The experiment showed that appropriate fine tuning can improve the generated text quality of LLM.

Meanwhile, there are also several aspects for further improvement potentially. Above all, only the text generation quality for the selected fields was investigated in this study. In future, more comprehensive approaches for tuning the parameters can be considered for better model performance achieved in all potential aspects of involved prompts. Certain methods of pre-processing the initial training datasets can be considered so that we can derive higher quality training data to improve model performance in all kinds of fields.

Another noticeable aspect for improvement is the training efficiency. With the current training method, it takes more than half of a day to get the training results, which is quite time consuming and undesirable. More efficient ways of optimization algorithms for loss function minimization for fine tuning algorithms can be considered so that the loss value can converge faster.

6 Conclusion

In conclusion, we gain an insight into the relationship of fine tuning with the performance of the final model with an early released GPT version in this study. Fine tuning is a critical step in the training procedure and plays an important role in the final dialogue ability of LLM. It can efficiently update weight matrices and save a huge amount of memories and running time.

We've already found out that appropriate tuning of parameters can improve the dialogue ability of the final model. However, there are also quite a lot of

working remaining to be completed for the large language model to have a more satisfactory performance. The study of LLM is a tough and lengthy process, continuous efforts in this field are urgently needed for better quality of the generated text.

LLM has a promising future since it has critical applications in a variety of occasions, which can be quite helpful to human's daily life. More advanced version of LLM will appear sooner or later with the researchers' effort.

7 Reference

Barry, B.,(2019, Feb 14), Better language models and their implications. Open AI. https://openai.com/index/better-language-models/

Ewald, J., (2023, May 9). Introduction to large language models [Video]. YouTube. https://m.youtube.com/watch?v=zizonToFXDs

Fine-tuning llms: Overview, methods & best practices. Fine-Tuning LLMs: Overview, Methods & Best Practices. (n.d.). https://www.turing.com/resources/finetuning-large-language-models

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2021). Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Ledzy. (n.d.). GitHub - Ledzy/MDS5210-23fall: Project code for MDS5210-23fall. GitHub. https://github.com/Ledzy/MDS5210-23fall

Lee, Jieh-Sheng & Hsiang, Jieh. (2020). Prior Art Search and Reranking for Generated Patent Text.

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., ... & Mian, A. (2023). A comprehensive overview of large language models. arXiv preprint arXiv:2307.06435.

Stephen M. W., Pretraining LLMs(n.d.), https://klu.ai/glossary/llm-pretraining

Pretraining Llms. Klu. (n.d.). https://klu.ai/glossary/llm-pretraining Tokenization. Mistral AI Large Language Models. (n.d.). https://docs.mistral.ai/guides/tokenization/

Wikipedia contributors. (2024, April 30). GPT-2. Wikipedia. https://en.wikipedia.org/wiki/GPT-2