LoRA Finetuning for Creating a Tutor Chatbot

Son Tung Do
Fordham University
New York, NY
sdo3@fordham.edu

December 2023

Abstract

Low-rank adaptation (LoRA) is one of the most popular technique for parameter-efficient finetuning of a Large Language Model (LLM). For this project, I propose using LoRA to finetune an LLM on course materials to act as an AI tutor. The finetuning process should give the model knowledge of the course materials that the base model does not sufficiently possess.

Keywords LoRA · LLM.

1 Introduction

LLMs have demonstrated the ability to be an unsupervised multitask learner (1) and a few-shot learner (2). Thanks to the efficient Transformer architecture (3), the sizes of language models are successfully scaled up to build more and more powerful models. This larger sizes also come with prohibitively expensive training cost in all phases, from pretraining to finetuning, which can reach up to hundreds of millions of dollars for the current state-of-the-art models. Due to this limitation, parameter-efficient finetuning methods can help individuals and organizations finetune large foundation models with much smaller computing budgets. LoRA (4) is one of the leading technique for finetuning LLMs, and its small weight size makes it an ideal candidate for my project with limited computing resources.

2 Background

2.1 Autoregressive Large Language Models

LLMs are language models based on the Transformer architecture. The Transformer learns sequence modelling tasks using multi-headed self-attention modules, which compute the dot product between all pairs of tokens within a given context windows. This self-attention mechanism comes in two flavors: the encoder and the decoder (Figure 1. The Transformer decoder is different from the encoder in the way that half of the decoder's attention matrix is masked. This masking prevents the decoder from seeing the future tokens. Because of this property, autoregressive LLMs, which are trained to predict the next tokens, utilize the decoder-only architecture, while masked LLMs, such as BERT (5), utilize the encoder-only architecture. There are also encoder-decoder models, but decoder-only architecture is the most popular now due to the success of autoregressive LLMs. For this project, I will choose an open-source decoder-only LLM as the foundational model, with a maximum size of 7 billion parameters.

2.2 Low-rank Adaptation

LoRA is a popular parameter-efficient finetuning technique, widely used in both image generation diffusion models and text generation language models. This method attempts to finetune by training only the residue of finetuned weights compared to base weights while keeping the base weights frozen. Crucially, the residue weights are decomposed into a product of two matrices of much lower ranks (Figure 2, thus substantially lowering the total amount of trainable weights during the finetuning process. Another advantage of LoRA is its adaptability. The user can insert specific LoRA weights into the base models to give it additional functionalities specifically trained in that set of LoRA weights.

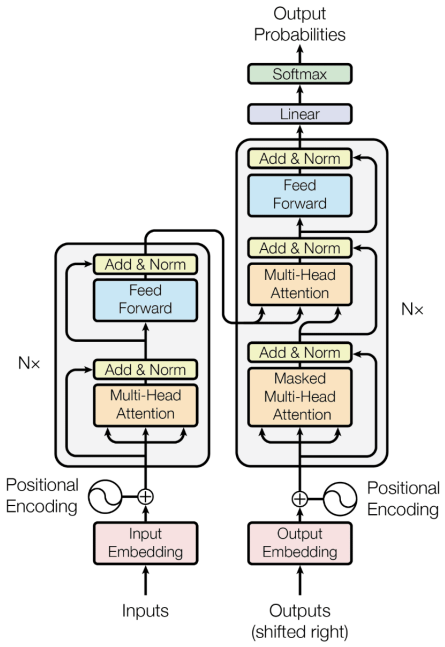

Figure 1: Transformer architecture from the original paper. The encoder modules are on the left, and the decoder modules are on the right of this figure

Different sets of weights can be substituted in depending on the purpose of the user. For a tutoring chatbot, each set of LoRA weights can contain be trained on one specific course/subject. The same base model can then act as a tutor for any subject when it's hooked to the that subject's set of LoRA weights. For the purpose of this project, I will only train one set of LoRA weights for one subject.

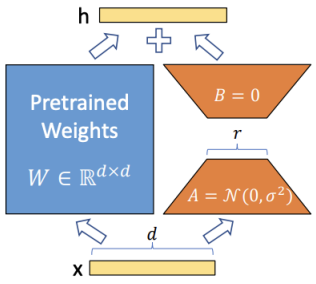

Figure 2: Low-rank adaptation method

3 Implementation

3.1 Foundational Model

I plan to use a foundational model with a maximum size of 7B parameters due to limited computing resources. This is a popular parameter size for many open-source LLMs, such as Llama (6) or Alpaca (7). If 7B parameter models are still too intensive to train, smaller models in the 2B-3B parameter range will be used. Further experimentation is needed to determine the feasibility of finetuning a 7B parameter LLM.

3.2 Dataset

The dataset will be text corpus extracted from pdf slides of this course, CISC 6210 Natural Language Processing. These slides contain a variety of topics in NLP, some of which might already be in the foundational model's training dataset. However, there will be contents where these base models don't have access tom such as new contents or private contents. Furthermore, the smaller sizes of these models, usually ranging from 2B to 7B parameters, will limit them from performing well at answering questions about the course materials. Thus, LoRA finetuning will help the model acquire additional domain knowledge required to be a tutor chatbot.

3.3 Computing Resources

I plan to use Google Colab to utilize its GPU capability and ease of use with setting up the environment and necessary dependencies. Further experimentation will determine the computing resource needed to train the LoRA weights, which will dictate whether I can use a 7B parameter. If computing resources needed become excessive, I will choose a smaller 2B or 3B parameter model. Colab Pro is also an affordable option for increasing the amount of computing resource and decreasing the model training time.

4 Conclusion

The project will be carried out in the next two weeks. If successful, it can be expanded by training multiple sets of LoRA weights for multiple different courses.

References

[1] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[4] 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," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018.

[6] 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.

[7] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, "Stanford alpaca: An instruction-following llama model," https://github.com/tatsu-lab/stanford_alpaca, 2023.