A Demo of Parameter-Efficient Fine Tuning with LoRA

George Zakka
New Jersey Institute of Technology
gz9@njit.edu

Abstract

Instead of completely retraining a model with millions or billions of parameters on the specific task data, one can freeze all the model weights and replace a subset of them with a much smaller number of trainable weights. Good performance, relative to a fully trained model, is achieved by clever use of rank decomposition of a matrix, whereby we have low-rank matrices that when multiplied together yield the same number of weights that were replaced (Yu 2023). We apply this technique to fine-tune DistilBert (Sanh 2020) for a sentiment analysis task on financial news tweets. We were able to achieve significant performance gains with very little training time. Source code has been released at https://github.com/gnz5/PEFT.

1 Introduction

A typical application of LoRA involves replacing Fully-Connected layers, usually at the beginning of Attention blocks, with so-called "LoRA layers." If the pretrained Fully-Connected (FC) layer we are replacing has X inputs and Y outputs, its LoRA layer will have two FC layers, the first with dimensions (X, r) and the second (r, Y), where r is a hyperparameter corresponding to the LoRA rank. This architecture resembles that of autoencoders and is fitting given that we are essentially trying to reduce the dimensionality of the text inputs to the space corresponding to the features relevant for our specific task. The important assumption is that, for downstream tasks, only a small subset of the pretrained model parameters are important for performance. This makes sense given that large LLMs learn a variety of general features of language from their training corpus, while a task like sentiment analysis only requires using a fraction of those features. To summarize, we are approximating weight matrices in the pretrained model with pairs of low-rank matrices while maintaining decent performance and significantly improving training speed.

2 Methodology

We implement fine-tuning with LoRA using various huggingface libraries including, most notably, PEFT, Dataset, and transformers. The relevant notebook can be found https://github.com/gnz5/PEFT. First, the sentiment analysis dataset published by huggingface user "zeroshot" was loaded using the Dataset library. This dataset contained over 12000 financial news tweets that were labeled as "bearish," (0) "bullish," (1) or "neutral" (2). Next, the transformers library was used to load DistilBERT. For preprocessing, the tweets were simply tokenized and padded. DistilBert has a built-in embedding layer so it was not necessary to compute token embeddings. Our PEFT parameters were a rank (r) of 4, lora alpha of 32, and a dropout probability of 0.01. The target module in the pretrained model was the "q_lin" layer in the Attention block.

3 Experiments

Model performance before and after fine-tuning was measured. Prior to fine-tuning, classification accuracy was significantly less than random chance (Figure 1). Moreover, the f1 score prior to training was 0.0799. After a single epoch of training on approximately 9000 examples, accuracy jumped to 84.67% on the test set and achieved an f1 score of 0.8443. Training a single epoch took 10.1 seconds on an NVIDIA GeForce RTX 4060 Laptop GPU. Training 5 epochs took 63.4 seconds on the same hardware. The performance difference between 1

and 5 epochs of training was not significant given our other hyperparameter choices.

4 Conclusion

This demo demonstrates the power of PEFT with LoRA. After optimizing only 0.93% of the pretrained weights while leaving all other parameters frozen, we achieved an over 10-fold increase in f1 score on the test set in a miniscule fraction of the time it took to train the original DistilBert model. For reference, DistilBert took 90 hours using 8 V100 GPUs to train (Sanh 2020). Moreover, no hyperparameter tuning was done. It is highly likely tuning matrix rank, adding LoRA layers, increasing the number of training epochs, and modifying the learning rate could further improve performance.

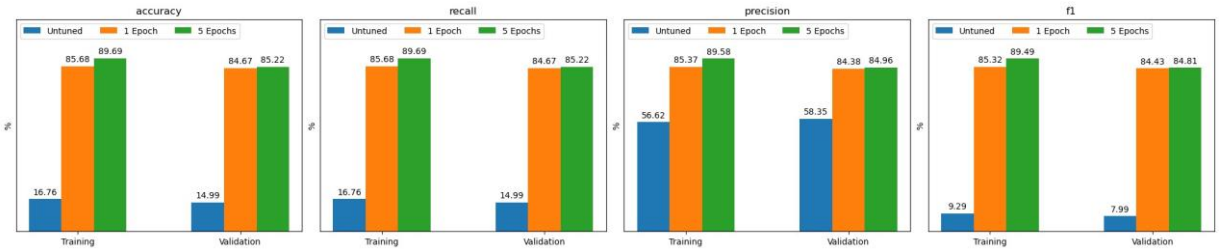

Figure 1

References

Yu, Y., Yang, C.-H. H., Kolehmainen, J., Shivakumar, P. G., Gu, Y., Ren, S. R. R., Luo, Q., Gourav, A., Chen, I.-F., Liu, Y.-C., Dinh, T., Filimonov, A. G. D., Ghosh, S., Stolcke, A., Rastow, A., & Bulyko, I. (2023, December). Low-Rank Adaptation of Large Language Model Rescoring for Parameter-Efficient Speech Recognition. 2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). https://doi.org/10.1109/asru57964.2023.10389632

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2020). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.

https://huggingface.co/datasets/zeroshot/twitter-financial-news-sentiment

https://huggingface.co/docs/peft/index

https://huggingface.co/docs/peft/v0.10.0/en/task_guides/lora_based_methods