Deep Learning Project 2: LoRA-BERT_News_Classifier

Team SPAR: Aryan Mamidwar, Spandan Rout
Github Codebase

arm9337@nyu.edu, sr7729@nyu.edu

Abstract

We present a Parameter-Efficient Fine-Tuning (PEFT) approach for the AG News classification task using a RoBERTa-base model enhanced with LoRA (Low-Rank Adaptation). The model was trained with only a small fraction of parameters while achieving competitive performance, illustrating the value of PEFT in reducing computational costs without major loss in accuracy.

Introduction

Fine-tuning large language models (LLMs) like RoBERTa for specific NLP tasks is powerful but often resource intensive. In this project, we explore a PEFT strategy using LoRA, a lightweight and modular approach that trains only a subset of the model's parameters. We apply this to the AG News dataset, a four-class text classification benchmark, and analyze the model's efficiency and performance trade-offs.

Methodology

Dataset

We use the AG News data set via the datasets library. It consists of 120K training and 7.6K test samples, split into four categories: World, Sports, Business, and Sci / Technology.

Pre-Processing

Text data are tokenized using RobertaTokenizer with truncation and dynamic padding enabled. Labels are mapped using id2label for interpretability during evaluation.

Model Architecture

- Base model: roberta-base
- Classification head: Automatically appended with RobertaForSequenceClassification
- Classes: 4
- Input size: Variable-length sequences (tokenized)

PEFT Configration

We used the peft library to configure LoRA with:

- Rank (r) = 4
- Alpha = 16
- Dropout = 0.1
- Target modules = query, value
- Task type = Sequence Classification (SEQ_CLS)

Training and Optimization

We configured the model training using the TrainingArguments setup with learning_rate = 2e-4, weight_decay = 0.01 and a cosine learning rate scheduler. The training was carried for 5 epochs with a warmup period of 500 steps, using a per-device batch size of 16 for both training and evaluation. The model was evaluated and saved at the end of each epoch, with the best model selected based on the highest accuracy metric. Mixed precision training (fp16) was enabled and 4 data loader workers were utilized.
Training was performed on the NYU High Performance Computing(HPC) GPU cluster, with logging every 100 steps, and gradient checkpointing was disabled.

Lessons Learned

LoRA Initial experiments showed that r=4 provided a good balance between parameter efficiency and model performance, while higher ranks increased trainable parameters without proportional accuracy gains.
Dropout in LoRA: A 0.1 LoRA dropout was optimal to prevent overfitting, while higher values led to underfitting and reduced validation accuracy.
Warmup steps: Using 500 warm-up steps stabilized early training with AdamW optimizer, but extending it slowed convergence without noticeable benefits.
Batch size Impact: A per-device batch size of 16 maximized GPU utilization while larger batch sizes did not provide any significant improvements in training stability or accuracy.

Results & Discussion

Our model achieved 94.53125% accuracy after 5 epochs with 741,124 parameters. Figure 2 shows the validation loss, training loss and accuracy for each epoch aligning with performance gains.

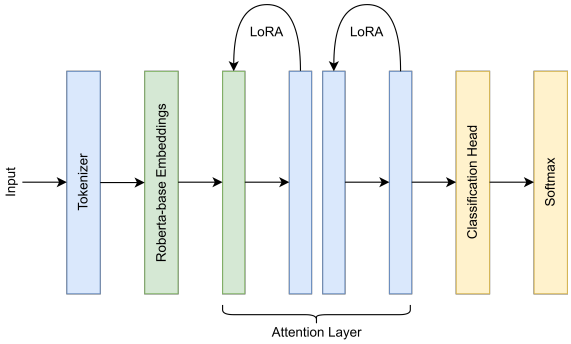

Figure 1: Model Architecture

[37300/37300 32:50, Epoch 5/5]

Epoch	Training Loss	Validation Loss	Accuracy
1	0.232100	0.216412	0.931250
2	0.184900	0.203782	0.939063
3	0.187000	0.183040	0.945312
4	0.202600	0.190651	0.937500
5	0.161500	0.191191	0.940625

Figure 2: Validation Loss, Training Loss and Accuracy

References

[1] Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2): 3.

[2] Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

[3] Micikevicius, P.; Narang, S.; Alben, J.; Diamos, G.; Elsen, E.; Garcia, D.; Ginsburg, B.; Houston, M.; Kuchaiev, O.; Venkatesh, G.; and Wu, H. 2018. Mixed Precision Training. arXiv:1710.03740.

[4] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2023. Attention Is All You Need. arXiv:1706.03762.

[5] Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

[6] Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; and He, Q. 2021. A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, 109(1): 43–76.

Disclousre

We have taken inspiration from the assignments/labs of another course (Intro to ML @ NYU). We have also taken inspiration of the above cited papers, articles, code bases. We have also used some LLM models to understand more about the project (ChatGPT, Grok, Perplexity). We have also used some more online resources like StackOverflow, official documentation of the packages used.