Parameter-Efficient Fine-Tuning of RoBERTa on AGNEWS using LoRA

Utkarsh Mittal[1], **Rushabh Bhatt**[1], **Aryan Ajmera**[1]

[1]New York University
{um2113, rb5726, aa1290}@nyu.edu
https://github.com/AryanAjmera18/Deep-Learning

Abstract

This paper presents our work on fine-tuning RoBERTa using Low-Rank Adaptation (LoRA) for the AGNEWS text classification task under a constraint of fewer than 1 million trainable parameters. Our final model achieved a private leaderboard score of 0.8500 and a public score of 0.85475 with approximately 980K trainable parameters. We outline our methodology, architectural decisions, experimentation with QLoRA, and lessons learned.

Introduction

Fine-tuning large language models can be resource-intensive. LoRA offers a parameter-efficient alternative by introducing trainable low-rank matrices into frozen pretrained models. We fine-tuned a RoBERTa-base model on AGNEWS using LoRA while keeping trainable parameters under 1M.

Methodology

Dataset and Preprocessing

We used the AGNEWS dataset from Hugging Face. Tokenization was performed using the RoBERTa-base tokenizer, truncating sequences to 256 tokens. Outliers beyond the 1st and 99th percentile in token length were filtered to reduce noise.

LoRA Configuration

We extended the **RobertaForSequenceClassification** model using the PEFT library.
LoRA hyperparameters:

- Rank (r): 7, Alpha: 77
- Target modules: **query**, **key**, **value**
- Dropout: 0.01

Training Setup

We trained with the following configuration:

- Optimizer: AdamW, Learning Rate: 3e-5
- Epochs: 6, Batch Size: 64
- Scheduler: Cosine Annealing
- Weight Decay: 0.1

Validation was done using `load_best_model_at_end=True` with Hugging Face's `evaluate` library.

QLoRA Attempt

We experimented with QLoRA for 4-bit quantization to reduce memory usage. However, compatibility issues on NYU HPC (bitsandbytes-related) prevented successful deployment, so we proceeded with standard LoRA.

Results

- Validation Accuracy: $\sim$85%
- Private Test Score: 0.8500
- Public Test Score: 0.85475
- Trainable Parameters: 980,740

Reproducibility

Our GitHub repository contains training scripts, evaluation notebooks, logs, and instructions for reproducing our experiments.

Lessons Learned

- LoRA achieves strong results with minimal trainable parameters.
- Token length filtering improves generalization.
- QLoRA offers potential but requires compatible environments.
- LoRA hyperparameter tuning is non-trivial but critical.

Pros and Cons

Pros

- Strong performance with $\sim$860K trainable parameters.
- Seamless integration using Hugging Face's PEFT.
- Robust accuracy despite frozen pretrained weights.
- Extensive community and library support.

Cons

- Adaptation limited to a few modules.
- Significant tuning required to meet parameter constraints.
- QLoRA was unusable due to platform issues.
- Memory usage remains high without quantization.

References

1. Hugging Face Transformers: https://huggingface.co/transformers/
2. AGNEWS Dataset: https://huggingface.co/datasets/ag_news
3. LoRA Paper: https://arxiv.org/abs/2106.09685
4. PEFT Library: https://github.com/huggingface/peft