Deep Learning CS 6953 / ECE 7123 Spring 2025
Project 2: Finetuning with LoRA

Krittin Nagar (kn2670), Samridh Srivastava (ss18906), Shikhar Malik (sm12762)
Kaggle Team: "BackProp to the Multiverse"
Github Repo

Abstract

This project explores Parameter-Efficient Fine-Tuning (PEFT) for natural language classification using Low-Rank Adaptation (LoRA), a technique that significantly reduces computational costs while maintaining model performance. We fine-tuned a pre-trained RoBERTa base model on the AG News dataset to classify news articles into four categories: World, Sports, Business, and Science/Technology. Our approach involved freezing the pre-trained weights while adding trainable low-rank matrices to specific attention layers. The implementation incorporated enhanced dropout techniques and carefully tuned hyperparameters, cosine learning rate scheduler, and gradient accumulation steps. The resulting model achieved superior accuracy on the evaluation set and surpassed the benchmark, demonstrating that LoRA can effectively adapt large language models for specialized tasks while training only on a fraction of the model parameters. This work highlights how PEFT methods can make advanced NLP capabilities more accessible with limited computational resources.

Introduction

Motivation

This project was conducted as part of a Kaggle competition organized by our professor, with the objective of applying LoRA to fine-tune a RoBERTa-based model for the task of news article classification, demonstrating the feasibility and strength of PEFT in real-world NLP applications.

Problem Statement

The primary research question guiding this project was:

> "Can Low-Rank Adaptation (LoRA) enable efficient fine-tuning of a pre-trained RoBERTa model for news classification while maintaining competitive performance?"

This research question aims to explore the efficiency-accuracy tradeoff of LoRA in adapting a large-scale model (roberta-base) for the multi-class classification task on the AG News dataset, which consists of short news articles labeled as World, Sports, Business, or Sci/Tech.

Key Contributions

- **Enhanced Dropout Integration:** Implemented an optimized dropout strategy by increasing the hidden dropout probability and attention probability dropout in the RoBERTa configuration, which helped prevent overfitting despite having fewer trainable parameters.
- **Strategic LoRA Application:** Applied LoRA to select attention layers within the RoBERTa model, focusing on layers 0, 1, 5, 10, and 11, with specific attention to query, key, value, and dense components. This targeted approach achieved a remarkably small trainable parameter footprint of just 0.69% of the total.
- **Optimized Training Regime:** Incorporated several performance-enhancing techniques such as cosine learning rate scheduler with a warmup ratio, gradient accumulation steps to improve training stability with larger effective batch sizes, weight decay to regularize the model, selecting batch sizes for training and for evaluation carefully to balance memory constraints and performance.
- **Efficient Inference Pipeline:** Developed a comprehensive inference pipeline capable of both batch processing for evaluation datasets and individual text classification, enabling seamless deployment of the model in various application contexts.

Methodology

Dataset Description

We utilized the AG News dataset, a well-established benchmark for text classification tasks. The AG News dataset comprises over 120,000 news articles collected from more than 2,000 news sources across the web. Each article is pre-classified into one of four categories: *World, Sports, Business and Science/Technology*. The dataset provides a balanced distribution across these four categories, making it ideal for multi-class classification tasks.

Preprocessing Steps

Tokenization We used the RoBERTa tokenizer to convert raw text into token IDs that can be processed by the model. The tokenization process included truncation and padding to ensure uniform sequence lengths across the dataset.

Dataset Splitting We split the original training dataset into training and evaluation sets using a 95%-5% ratio, reserving 640 examples for evaluation while using the rest for training. This split was created with a fixed random seed (42) to ensure reproducibility.

Model Architecture

Base Model Selection We selected RoBERTa-base as our foundation model. RoBERTa is a robustly optimized BERT variant that has demonstrated superior performance across various NLP tasks. The base version contains 12 transformer layers with approximately 125 million parameters.

Enhanced Dropout Configuration To improve regularization and prevent overfitting, we implemented an enhanced dropout strategy, where we increased the dropout probability for both hidden states and attention probabilities from the default 0.1 to 0.2, providing stronger regularization during training.

Low-Rank Adaptation (LoRA) The core of our approach was the application of LoRA for parameter-efficient fine-tuning. LoRA works by representing weight updates as a product of two low-rank matrices (A and B), significantly reducing the number of trainable parameters. Key aspects of our LoRA configuration:

- Rank (r) of 16 for the low-rank matrices
- LoRA scaling factor (alpha) of 32
- Additional dropout of 0.05 specific to LoRA layers
- Strategic targeting of specific layers rather than applying LoRA to all layers:
  - Query, key, and value components in the first layer for capturing initial representations
  - Query components in layers 1 and 5 for mid-level feature extraction
  - Query components and dense output in layers 10 and 11 for high-level reasoning and classification

This selective application resulted in only 864,004 trainable parameters (0.69% of the total model parameters), dramatically reducing computational requirements while maintaining performance.

Training Process

Key elements of our training config:

- Learning rate of 3e-5, slightly higher than typical defaults to facilitate faster convergence
- Weight decay of 0.1 for regularization
- Evaluation and model saving after each epoch
- Automatic loading of the best model at the end of training
- Cosine learning rate scheduler with a 0.15 warmup ratio
- Gradient accumulation with 2 steps to simulate larger batch sizes
- Optimization for accuracy as the primary metric

Evaluation Metrics

We implemented a comprehensive evaluation framework using multiple metrics to assess model performance:

- Accuracy: The proportion of correctly classified examples
- F1 Score: The harmonic mean of precision and recall
- Precision: The ratio of true positives to all predicted positives
- Recall: The ratio of true positives to all actual positives

All metrics were calculated using weighted averaging to account for potential class imbalances.

Architectural Choices: Pros and Cons

Our model integrates several design choices that contributed to performance improvements while presenting certain trade-offs. Below, we break down what worked well and what didn't.

Successful Modifications

- **Strategic Layer Selection for LoRA:** Targeting specific layers rather than applying LoRA uniformly across all layers yielded significant efficiency gains while maintaining performance. Specifically, focusing on early layers (0, 1) for basic feature extraction, middle layers (5) for intermediate representations, and final layers (10, 11) for classification logic proved highly effective.
- **Enhanced Dropout Strategy:** The increased dropout probabilities (0.2 instead of 0.1) in both attention and hidden layers helped prevent overfitting despite the reduced parameter count, which was crucial for maintaining generalization capabilities.
- **Cosine Learning Rate Schedule with Extended Warmup:** The combination of a cosine learning rate scheduler with a relatively long warmup period (15% of total steps) provided stable optimization dynamics, preventing divergence in early training stages and allowing for better convergence.
- **Gradient Accumulation:** Implementing gradient accumulation with 2 steps effectively doubled our batch size without increasing memory requirements, improving training stability and gradient estimate quality.
- **Weight Decay Optimization:** The relatively high weight decay value of 0.1 provided strong regularization, which complemented the dropout strategy and helped prevent the model from overfitting to the training data.

Unsuccessful Modifications

- **Extended Training Duration:** Increasing the number of epochs beyond 5-6 did not improve performance; in fact, it often led to overfitting as evidenced by decreasing validation accuracy despite improving training metrics.
- **Batch Size Variation:** Changing the batch size (e.g., 8, 16, 32) showed negligible difference in accuracy or convergence speed, suggesting that our model was relatively robust to this hyperparameter within the tested range.

Component	Details
Base Model	`roberta-base` (pretrained from HuggingFace)
Tokenizer	`RobertaTokenizer` with truncation and padding
Dataset	AG News dataset (4 classes)
Number of Classes	4
Labels	World, Sports, Business, Sci/Tech
Custom Config	`RobertaConfig` with modified dropout values
Hidden Dropout	0.2 (increased from default 0.1)
Attention Dropout	0.2 (increased from default 0.1)
Model Loaded	`RobertaForSequenceClassification` with the above config
Custom Wrapper Class	`EnhancedRobertaClassifier`
Extra Dropout Layer	`nn.Dropout(0.3)` added before classification head
Forward Pass Logic	Returns logits if available; else modifies and returns `pooler_output`

Table 1: Model Architecture Details

Parameter	Value
`r` (rank)	16
`lora_alpha`	32
`lora_dropout`	0.05
`bias`	none
`task_type`	`SEQ_CLS` (Sequence Classification)
`target_modules`	- `roberta.encoder.layer.0.attention.self.query` - `roberta.encoder.layer.0.attention.self.key` - `roberta.encoder.layer.0.attention.self.value` - `roberta.encoder.layer.1.attention.self.query` - `roberta.encoder.layer.5.attention.self.query` - `roberta.encoder.layer.10.attention.self.query` - `roberta.encoder.layer.10.output.dense` - `roberta.encoder.layer.11.output.dense`

Table 2: PEFT LoRA Configuration

Hyperparameter	Value
`learning_rate`	3E-05
`batch_size`	16, 32, 8
`num_epochs`	5
`dropout`	0.05
`weight_decay`	0.1
`eval_strategy`	Epoch
`gradient_accumulation`	2
`warmup_ratio`	0.15

Table 3: Training Hyperparameters

- **LoRA Dropout Adjustment:** Increasing lora_dropout from its default value reduced overall model accuracy. While dropout is generally helpful for regularization, excessive dropout in the LoRA layers appeared to interfere with the model's ability to learn effective adaptations.
- **Aggressive Learning Rate Warmup:** Using aggressive learning rate warmup (greatear than 0.2) slowed convergence and slightly harmed final performance, likely because the model spent too many steps at suboptimal learning rates.
- **Excessive Dropout:** Applying dropout layers beyond 0.3 introduced too much regularization and degraded classification confidence, creating a model that was underfit and unable to capture the necessary patterns for accurate classification.

Result and Analysis

Our parameter-efficient fine-tuning approach using LoRA achieved impressive results on the AG News classification task. The results were obtained on the evaluation dataset consisting of 640 examples. We achieved a high F1 score, nearly identical to the accuracy, indicating balanced performance across all four news categories with minimal class bias. The training process demonstrated consistent improvement across all 5 epochs. Several key observations during training:

- **Rapid Early Convergence:** The model showed substantial improvement in the first three epochs, with accuracy increasing from 90.16% to 92.81%.
- **Stability in Later Epochs:** Performance stabilized after the third epoch, with only minor fluctuations in subsequent epochs, suggesting that the model reached a convergence point.
- **No Overfitting Signs:** The validation loss closely tracked the training loss throughout the training process, indicat-

ing that our regularization strategies (enhanced dropout, weight decay) were effective at preventing overfitting despite the relatively small number of trainable parameters.

- **Consistent Metric Alignment:** The close alignment between accuracy, F1 score, precision, and recall across all epochs suggests balanced performance across all four news categories.

One of the most significant achievements of our approach was the dramatic reduction in trainable parameters:

- Total Model Parameters: 125,512,712
- Trainable Parameters: 864,004
- Percentage of Parameters Trained: 0.69%

This represents a greater than 99% reduction in trainable parameters compared to full fine-tuning. The computational advantages of this approach include:

- **Reduced Memory Requirements:** The memory footprint during training was significantly smaller than what would be required for full fine-tuning.
- F**aster Training:** Despite running for 5 epochs, the total training time was substantially shorter than would be expected for full fine-tuning.
- **Minimal Storage Overhead:** The adapter-only checkpoint is orders of magnitude smaller than a fully fine-tuned model checkpoint.

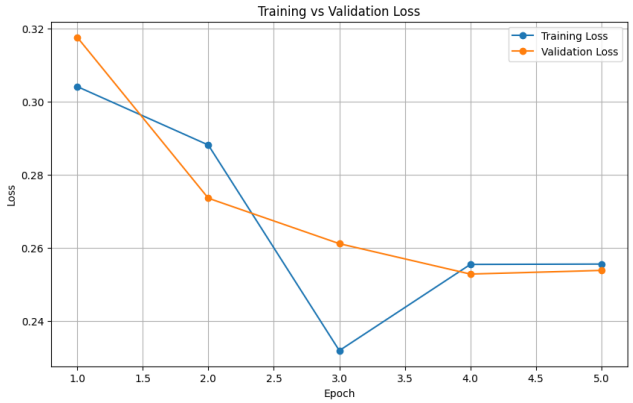

Figure 1: Training vs Validation Loss

Conclusion

Our project successfully demonstrated the effectiveness of Parameter-Efficient Fine-Tuning using Low-Rank Adaptation (LoRA) for text classification. By strategically applying LoRA to specific layers of a pre-trained RoBERTa model, we achieved 92.8% accuracy on the AG News classification task while training only 0.69% of the model parameters. Our approach incorporated enhanced dropout (0.2), a cosine learning rate scheduler with 0.15 warmup ratio, and gradient accumulation to improve training stability. The strong performance metrics : 0.928 F1 score, 0.928 precision, and 0.928 recall, validate that parameter-efficient approaches can maintain competitive performance while drastically reducing computational requirements.

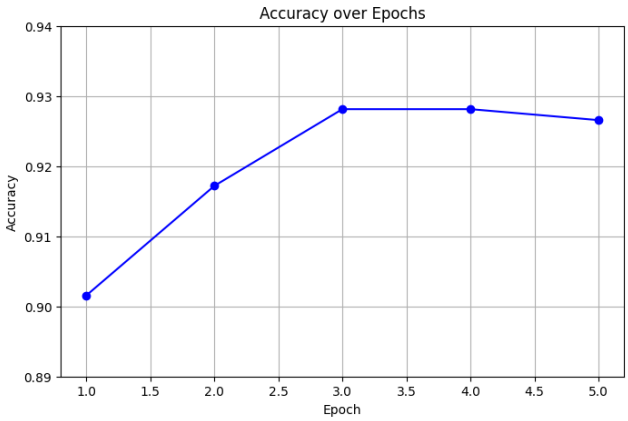

Figure 2: Accuracy over Epochs

References

1. Qing, P., Gao, C., Zhou, Y., Diao, X., Yang, Y., Vosoughi, S. (2024). AlphaLoRA: Assigning LoRA Experts Based on Layer Training Quality. Retrieved from https://arxiv.org/abs/2410.10054
2. Shun, J., Zheng, C. Revolutionizing Large Model Fine-Tuning: The Role of LoRA in Parameter-Efficient Adaptation. Authorea Preprints. Retrieved from https://www.techrxiv.org/doi/pdf/10.36227/techrxiv.174015835.57150536
3. Fang, Z., Wang, Y., Yi, R., Ma, L. (2024, September). Dropout Mixture Low-Rank Adaptation for Visual Parameters-Efficient Fine-Tuning. In European Conference on Computer Vision (pp. 369-386). Cham: Springer Nature Switzerland. Retrieved from https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/01163.pdf
4. For idea generation and grammar, https://chatgpt.com