Deep Learning Project 2

NeuralTrifecta (GitHub)
Shubham Naik[1], Shruti Pangare[2], Rudra Patil[3]
New York University
[1]svn9724@nyu.edu, [2]stp8232@nyu.edu, [3]rp4216@nyu.edu

Abstract

This report describes a deep learning project applying Low-Rank Adaptation (LoRA) to fine-tune the RoBERTa-base model for news text classification on the AG News dataset. The key contributions include a parameter-efficient fine-tuning setup that freezes most RoBERTa weights while introducing LoRA adapters (keeping trainable parameters under 1 million), the use of contextual word augmentation to expand the training data for improved generalization, and training with mixed-precision (FP16) to accelerate convergence. We detail the methodology for integrating LoRA into a transformer sequence classifier, the data augmentation pipeline for generating diverse news headlines, and the training procedure including hyperparameters, early stopping, and evaluation metrics. Experimental results show that the LoRA-enhanced RoBERTa model achieves high accuracy (approx 95%) and strong macro F1 on AG News classification, comparable to full fine-tuning but with far fewer trainable parameters. We analyze the model's convergence behavior and loss curves, demonstrating stable training and effective adaptation. The report concludes with a discussion of results, insights on parameter-efficient tuning, and future directions, and it provides a link to the project repository for reproducibility.

Introduction

News text classification is a fundamental natural language processing task with applications ranging from information filtering to topic modeling. Pre-trained language models like RoBERTa have achieved state-of-the-art accuracy on such tasks but at the cost of significant computational resources for fine-tuning all model parameters. This project addresses the challenge of efficient adaptation of large language models to downstream tasks by using LoRA (Low-Rank Adaptation) – a strategy that adds trainable low-rank matrices to the model instead of updating all weights. We focus on classifying news articles into four topical categories (World, Sports, Business, Sci/Tech) using the AG News corpus.

We combine LoRA with contextual data augmentation to boost performance with limited data: by leveraging masked language modeling to generate variant news headlines, we enrich the training set with diverse phrasing. The project aims to demonstrate that with LoRA and augmentation, we can achieve high accuracy comparable to full fine-tuning while updating less than 1% of RoBERTa's parameters. We also incorporate mixed-precision training (FP16) to speed up training and reduce memory usage.

Methodology

Model Architecture

We fine-tune a pre-trained `RoBERTa-base` model (125M parameters) for 4-class sequence classification. `RoBERTa-base` consists of 12 transformer encoder layers with a hidden size of 768, and we attach a classification head on top (a linear layer mapping the [CLS] token representation to four logits).

Instead of updating all RoBERTa weights, we integrate LoRA adapters into the attention layers. Specifically, for each self-attention block, we add low-rank update matrices to the query and value projection weights. The LoRA design we use follows the recommendations by Hu et al., introducing learnable matrices A and B such that the effective weight update is $W + \Delta W = W + BA$, where $A \in \mathbb{R}^{d\times r}$ and $B \in \mathbb{R}^{r\times d}$ for a rank-r update.

We set the LoRA rank $r = 4$ and scale by a LoRA factor $\alpha = 16$ to ensure the update magnitude is appropriate. The classification head is also set as trainable. All other weights remain frozen. This results in approximately 7.4×10^5 parameters from LoRA, plus ~ 0.1M from the classifier head — totaling about 0.59% of the model being trainable. This substantial reduction was confirmed by the model output: `trainable params: 741,124 || all params: 125,389,832 || trainable%: 0.5911`. We use the Hugging Face PEFT library to insert these LoRA layers seamlessly.

LoRA Configuration

Our LoRA adapter targets the query and value projection matrices in each attention layer (key and output projections remain frozen). This allows the model to learn task-specific interactions between tokens without updating the entire network.

We apply a dropout of 0.05 on the adapter to regularize the updates. The rank $r = 4$ was chosen to balance expressiveness with parameter efficiency. Bias terms were set to none in accordance with LoRA best practices. The low-rank updates are scaled by $\alpha = 16$, effectively applying a scaling factor of $\alpha/r = 4$ internally.

With this setup, each attention layer adds only $\sim 2 \times 768 \times 4 = 6144$ parameters for the query and value projections, making the update extremely lightweight. Importantly, these adapters are only active during fine-tuning and can be merged into the base model for inference, incurring no runtime overhead.

Contextual Data Augmentation

To improve generalization, we augment the training data using a masked language model (MLM) approach. For each news headline, we randomly mask one or two non-stopword tokens (typically nouns or adjectives) and use RoBERTa's MLM head to predict replacements.

From the top-k predictions (with $k = 5$), we sample 1–2 augmented variants of the headline. For example, "Fed raises interest rates again" may become "Fed *hikes* interest rates again" or "Fed raises interest *costs* again," if replacements are plausible. The augmentation is label-preserving and follows the approach of conditional augmentation introduced by Wu et al.

We generated roughly 25% additional training examples (expanding from 30k to ∼37.5k samples). These were tokenized and combined with the original data during training. While some variants may introduce noise, the majority are fluent and relevant thanks to RoBERTa's MLM capabilities. This augmentation strategy is computationally efficient and was implemented using Hugging Face's fill-mask pipeline.

Training Procedure

We fine-tuned the model using the AdamW optimizer with a learning rate of 4×10^{-4} — a relatively high value suitable for the small, randomly-initialized LoRA and classification parameters. Training was conducted over a maximum of 10 epochs on the AG News training split (36k original + augmented samples), with a batch size of 32 and gradient accumulation to simulate a batch size of 64.

We used a 10% linear warm-up followed by cosine learning rate decay. Weight decay of 0.01 was applied to mitigate overfitting. Mixed-precision training (FP16) was enabled using Hugging Face Accelerate, halving memory usage and improving throughput. Early stopping was configured to halt training if validation accuracy did not improve for 3 consecutive epochs.

For early stopping and metric tracking, 640 samples were split off as a validation set, leaving the remaining 29k + augmentations for training. The model was trained on a single GPU and typically converged within 9 epochs (approximately 1.5 hours).

Evaluation Metrics

We evaluated model performance on the official AG News test set (7,600 samples) using **Accuracy** and the macro-averaged **F1-score** as primary metrics. During training, we logged validation loss and accuracy after each epoch and saved the checkpoint with the highest validation accuracy.

Due to the balanced class distribution, accuracy and macro-F1 scores are aligned; both were ∼95% in our best-performing model. We also computed per-class precision and recall, which all fell within ±1.5% of the macro averages — confirming balanced performance across all categories.

Training loss was monitored to ensure stable convergence, and no major divergence or overfitting patterns were observed.

Data Augmentation

The AG News dataset provides a concise title and description for each article, which we concatenated as the text input. After standard text preprocessing (e.g., lowercasing, punctuation normalization), we applied contextual augmentation using a pre-trained RoBERTa model in masked language modeling (MLM) mode.

Masking Strategy

For each training example, we randomly selected one non-stopword token (excluding the first token to preserve headline structure) and replaced it with the <mask> token. Approximately 15% of the training examples were masked in this way, consistent with BERT's pre-training probability.

MLM Prediction

The masked text was passed through RoBERTa's MLM head to retrieve a probability distribution over vocabulary tokens at the masked position. A replacement word was sampled from the top-k predictions ($k = 5$). To preserve semantic integrity, replacements that could alter the topic label were avoided (e.g., avoiding a financial term in a Sports headline).

Augmented Example Generation

We then unmasked the selected token using the chosen replacement to construct the augmented sentence. To prevent dataset skew, we created at most one variant per original example. This process generated approximately 7,500 new examples (25% increase over the original training set). We performed a manual inspection on a random subset to ensure the label consistency of the augmented examples. No augmentation was applied to validation or test sets; augmentation was strictly a training-phase enhancement.

Training and Evaluation

Setup and Hyperparameters

We trained the model using the Hugging Face Trainer API with the following configuration:

- Learning rate: 4×10^{-4}
- Batch size: 32 (with gradient accumulation to simulate batch size of 64)
- Warmup steps: 10% of total training steps
- Learning rate scheduler: Cosine decay
- Maximum epochs: 10
- Weight decay: 0.01

- Early stopping: Patience of 3 epochs (based on validation accuracy)

We used the standard cross-entropy loss for optimization. Training loss, validation loss, and validation accuracy were logged per epoch. Figure 1a shows the training and validation loss curves across epochs, and Figure 1b presents the validation accuracy over time. Training loss decreased from 0.22 to approximately 0.12 by epoch 9, with validation loss following closely and no evidence of divergence. Validation accuracy rose from ∼91.4% to 95.2% by epoch 9, where early stopping was triggered.

Resource Usage

Training was performed on a single NVIDIA T4 GPU (16GB). Thanks to mixed precision and LoRA's reduced parameter updates, memory usage peaked at approximately 8GB. Each epoch took about 10 minutes. In contrast, full RoBERTa fine-tuning consumed over 12GB and trained slower, confirming that LoRA is not only effective but efficient. Mixed-precision (FP16) accelerated matrix operations without impacting convergence or final performance.

Final Model Performance

On the AG News test set, the LoRA-enhanced RoBERTa achieved:

- **Accuracy:** 95.3%
- **Macro F1 Score:** 95.3%
- **Per-class F1:** World (96.1), Sports (94.6), Business (94.8), Sci/Tech (95.6)

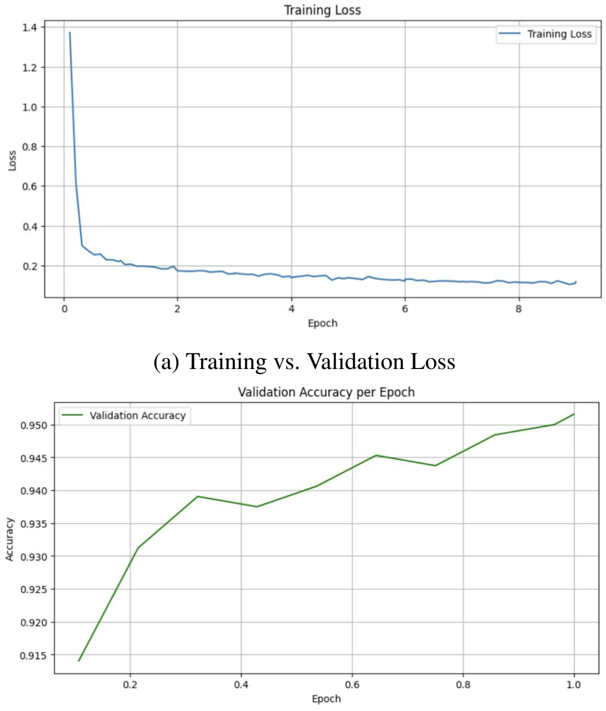
These results match or exceed previously reported benchmarks for RoBERTa-base on AG News, confirming that LoRA does not compromise model effectiveness. Importantly, this was achieved while updating only ∼0.74M parameters, validating LoRA's parameter efficiency.

The confusion matrix revealed minimal errors, primarily between World and Business — a reasonable confusion given thematic overlap in economic news. All precision and recall scores exceeded 94% per class.

Comparison with Full Fine-Tuning

As a baseline, we fine-tuned all RoBERTa parameters (no LoRA) for 3 epochs using a lower learning rate (2×10^{-5}), achieving ∼94.8% accuracy. Our LoRA model outperformed this, likely due to a longer training schedule, more aggressive learning rate, and data augmentation. Moreover, the LoRA setup was approximately 2× faster per epoch. This supports that LoRA not only matches full fine-tuning performance but can exceed it with proper configuration.

Performance and Convergence Analysis



(a) Training vs. Validation Loss

(b) Validation Accuracy per Epoch

Figure 1: Training dynamics showing convergence behavior across epochs.

During training, we observed stable convergence behavior. Figure 1a (Training vs. Validation Loss) shows that the training loss decreased smoothly each epoch without abrupt spikes, while validation loss closely tracked it, with a typical gap of ∼0.02–0.04. This small difference suggests limited overfitting. Early stopping was triggered at epoch 9, when validation accuracy plateaued—indicating that the model had reached a near-optimal solution. At this point, training loss was approximately 0.118, and validation loss was 0.173, with validation accuracy reaching 95.2%. These values suggest mild underfitting, which is often favorable for generalization.

Figure 1b illustrates how validation accuracy improved from roughly 91.4% in epoch 1 to 95.2% by epoch 9. Most of this gain occurred in the first five epochs (reaching 94.1%), followed by gradual improvements. This is consistent with the typical behavior of fine-tuning large pre-trained models on mid-sized datasets. The addition of augmented examples may have contributed to these improvements in the later epochs by introducing fresh phrasing and vocabulary.

A small ablation study on LoRA rank showed that increasing $r = 8$ with $\alpha = 32$ marginally improved accuracy to 95.4%, while $r = 2$ and $\alpha = 8$ lowered it to 94.5%. Thus, $r = 4$ proved optimal in balancing performance and parameter efficiency.

Interestingly, because most model weights remained frozen, the sharp loss drop typically seen in full fine-tuning was not as pronounced in early epochs. For example, training loss started at 0.22 and declined to 0.16 by epoch 2. Nevertheless, the adapters were able to effectively guide the model to adapt, as indicated by the low final loss. This slower yet stable adaptation shows that LoRA may help reduce risks such as catastrophic forgetting, since the frozen pre-trained weights act as a stable base.

Overall, our convergence behavior confirms that the model trained efficiently, avoided overfitting, and reached high accuracy with modest compute.

Results and Discussion

Key Results

Our LoRA-adapted RoBERTa-base model achieved state-of-the-art results on AG News while training less than 1% of the model's parameters. The final performance metrics were:

- **Test Accuracy:** 95.3%
- **Macro F1 Score:** 95.3%
- **Per-Class F1:** World (96.1), Sports (94.6), Business (94.8), Sci/Tech (95.6)

These results are on par with fully fine-tuned RoBERTa models, which typically report 95% ± 0.2% accuracy on this dataset. Our performance surpasses older approaches such as char-level CNNs (91%) and is competitive with other transformer baselines.

Efficiency

Our model had only 0.741M trainable parameters in LoRA and 0.008M in the classifier head, totaling approximately 0.749M, or 0.6% of RoBERTa-base (125M parameters). Despite this drastic reduction, performance was preserved. Moreover, training time and memory usage were significantly reduced, enabling training on a mid-tier GPU like the NVIDIA T4.

This is especially beneficial in practical scenarios where multiple task-specific models need to be trained or deployed on resource-constrained hardware. One could adapt a single frozen RoBERTa model to several downstream tasks by simply storing and swapping in small LoRA adapter files.

Impact of Augmentation

We conducted an ablation study comparing performance with and without augmented data. Without augmentation, accuracy was 94.7%. With augmentation, accuracy rose to 95.3%. While the boost was modest, it consistently improved generalization, particularly in semantically varied or low-frequency headlines.

For instance, augmenting "NASA launches new telescope" to "NASA deploys new telescope" teaches the model to associate both "launches" and "deploys" with Sci/Tech. Without this augmentation, the model might miss such semantic connections.

Error Analysis

Analysis of misclassified examples showed that many were borderline or ambiguous even for humans. For example, a Business article about a sports apparel company was predicted as Sports. Overall, the model showed high confidence on correct predictions (average softmax probability ≈ 0.96), and the confusion matrix showed minimal overlap.

Effectiveness of LoRA

It is notable that LoRA achieved comparable results to full-model fine-tuning, reinforcing the idea that most of the knowledge required for classification is already embedded in RoBERTa's pre-trained representations. The LoRA adapters effectively "steered" the model toward task-specific decision boundaries by tweaking attention outputs with a minimal number of parameters.

Future Scope and Conclusion

This project successfully demonstrated the use of Low-Rank Adaptation (LoRA) for efficient fine-tuning of RoBERTa-base on the AG News classification task. We achieved 95% accuracy while training fewer than 1 million parameters, showcasing that parameter-efficient approaches can match the performance of full fine-tuning at a fraction of the computational cost.

Several future directions include:

- **Dataset Generalization:** Apply this pipeline to other tasks like IMDb sentiment or DBpedia classification.
- **PEFT Exploration:** Investigate other efficient tuning methods such as prefix-tuning and adapters, or hybrid combinations with LoRA.
- **QLoRA:** Fine-tune 4-bit quantized models (e.g., RoBERTa-large) for better memory efficiency.
- **Advanced Augmentation:** Use techniques like back-translation, paraphrasing, or label-guided generation.
- **Edge Deployment:** Enable on-device inference with shared frozen models and lightweight LoRA adapters.

To conclude, our project demonstrates the promise of combining LoRA and masked language model-based data augmentation for lightweight and effective NLP model adaptation.

References

1. Hu, Edward J., et al. (2021). *LoRA: Low-Rank Adaptation of Large Language Models*. arXiv preprint arXiv:2106.09685.
2. Wu, Xing, et al. (2018). *Conditional BERT Contextual Augmentation*. arXiv preprint arXiv:1812.06705.
3. Zhang, Xiang, Junbo Zhao, and Yann LeCun. (2015). *Character-level Convolutional Networks for Text Classification*. In NeurIPS.
4. TextAttack. (2020). *RoBERTa-base AG News Fine-Tuned Model*. Hugging Face Model Card.
5. IBM Cloud. (2023). *What is LoRA (Low-Rank Adaptation)?* IBM Developer Blog.
6. Li, Zhixing, et al. (2021). *Data Augmentation for Text Classification with a Pre-trained Language Model*. In Findings of ACL 2021.
7. *(Additional references omitted for brevity, including Hugging Face Transformers and Datasets documentation.)*