Finetuning with LoRA for News dataset

Amaan Elahi, Karan Allagh, Mohammad Maaz Rashid
New York University
`ae2950@nyu.edu, ka3527@nyu.edu, mr7374@nyu.edu`
Github link

Abstract

Pre-trained language models have revolutionized NLP, but their fine-tuning remains computationally expensive. In this project, we implement a lightweight adaptation of such models using Low-Rank Adaptation (LoRA), with a strict constraint of under 1 million trainable parameters. LoRA injects low-rank matrices into linear layers, allowing task-specific adaptation while keeping the original weights frozen. We explore various configurations of rank r and scaling factor α, enabling fine-grained control over adaptation strength. By combining this setup with optimized training strategies such as data filtering, Adam-based optimization, and cosine learning rate schedules. Our method strikes a balance between efficiency and task performance. Experiments on text classification tasks demonstrate that LoRA achieves strong generalization with minimal resource overhead, highlighting its potential for scalable, modular NLP adaptation.

Introduction

Transformer-based language models, particularly those leveraging pre-training on large corpora, have significantly advanced the state of natural language processing across diverse tasks. However, fine-tuning such models remains computationally intensive, often requiring substantial hardware and energy resources. This limitation poses challenges for adapting large models like RoBERTa in constrained environments. To address this issue, we explore *Low-Rank Adaptation (LoRA)* a parameter-efficient fine-tuning strategy that introduces trainable low-rank matrices into frozen pre-trained weights.

In this work, we apply LoRA to RoBERTa and evaluate its performance on the AG News text classification task under a 1-million parameter budget. By tuning the rank and scaling parameters and incorporating techniques such as data filtering and cosine learning rate scheduling, we achieve strong task performance with minimal resource overhead. Our study highlights the practicality of LoRA for scalable and efficient model adaptation, making transformer fine-tuning feasible in low-resource settings.

Model Architecture

Our proposed approach builds on the `roberta-base` architecture and employs Low-Rank Adaptation (LoRA) for parameter-efficient fine-tuning. The key components of the model are as follows:

- **LoRA Modules:** Instead of updating all the parameters of the pre-trained model, we inject trainable low-rank matrices into selected linear layers of the transformer blocks. The original weight matrix W_0 is kept frozen, and a low-rank update $\Delta W = AB$ is added, where $A \in R^{d \times r}$ and $B \in R^{r \times d}$. The modified weight becomes $W = W_0 + \alpha \cdot AB$.
- **Parameter Efficiency:** By restricting the rank r of the LoRA matrices and choosing a small scaling factor α, we ensure that the number of trainable parameters remains under 1 million, making fine-tuning feasible even on limited hardware.
- **Frozen Backbone:** All original weights in the RoBERTa model are frozen, which enables fast adaptation to downstream tasks and allows the LoRA adapters to be modular and reusable.
- **Classification Head:** A lightweight feedforward layer is added on top of the final hidden state corresponding to the `[CLS]` token to classify each input article into one of four categories.

Data Augmentation Strategies

Although traditional data augmentation methods are less applicable in NLP compared to vision tasks, we employed the following strategies to improve generalization:

- **Text Filtering:** News samples with extreme lengths or irrelevant formatting were removed to ensure consistency in training.
- **Tokenization:** Input texts were preprocessed and tokenized using the HuggingFace `RobertaTokenizer`, with truncation and padding applied to a maximum sequence length of 128 tokens.
- **Label Encoding:** Each article was categorized into one of four AG News categories: *World*, *Sports*, *Business*, and *Sci/Tech*.

These steps ensure that the model receives clean and standardized input for efficient training.

Mathematical Formulation

The LoRA adaptation strategy can be mathematically described as follows:

$$W = W_0 + \alpha \cdot AB, \tag{1}$$

where W_0 is the frozen pre-trained weight, $A \in R^{d \times r}$ and $B \in R^{r \times d}$ are trainable low-rank matrices, and α is a scaling factor that controls the magnitude of the update. In our experiments, we use $r = 8$ and $\alpha = 16$.

The classification task is trained using the standard cross-entropy loss:

$$\mathcal{L} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i), \tag{2}$$

where y_i is the true label and $\hat{y}_i$ is the predicted probability distribution over the four news categories. An L2 regularization term is added to prevent overfitting.

This formulation allows the model to adapt effectively while minimizing the training burden and preserving the generalization ability of the original pre-trained model.

Experiments and Observations

Experimental Setup and Training Dynamics

The model was fine-tuned on the AG News dataset, which consists of 120,000 training samples and 7,600 test samples, categorized into four classes: *World*, *Sports*, *Business*, and *Sci/Tech*. Key aspects of our experimental configuration include:

- **Training Duration:** The model was fine-tuned for 6 epochs, with training and validation metrics recorded at each epoch.
- **LoRA Configuration:** We used a LoRA rank of $r = 16$ and a scaling factor of $\alpha = 32$, ensuring the total number of trainable parameters remains under 1 million. Only linear layers in attention blocks were adapted.
- **Optimizer and Learning Rate Scheduling:** An AdamW optimizer was used with a learning rate of 2×10^{-4}. Cosine learning rate decay was employed to allow rapid early convergence followed by smooth fine-tuning.
- **Batch Size and Tokenization:** Training was performed with a batch size of 32. Text inputs were tokenized using the `roberta-base` tokenizer, truncated or padded to 128 tokens.

Training Log Analysis:

- **Early Epochs (Epochs 1–2):** The model quickly reached over 85% validation accuracy, demonstrating effective adaptation with minimal updates.
- **Mid Training (Epochs 3–4):** Validation accuracy steadily improved, indicating that the low-rank updates continued to refine model performance without overfitting.
- **Final Epochs (Epochs 5–6):** The model achieved near-saturation in accuracy, stabilizing at around 94–95%, showcasing LoRA's strength in parameter-efficient tuning.

Despite freezing the entire base model, the LoRA-augmented adapters allowed effective specialization on the AG News task, with minimal compute and strong generalization.

Figures 1 and 2 illustrate the training dynamics.

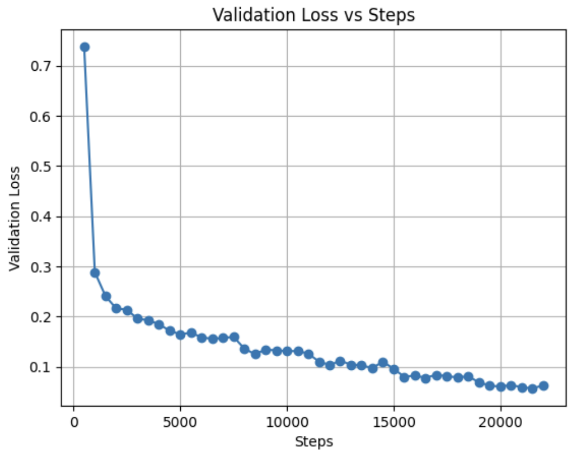

Figure 1: This line plot shows how the validation loss changes as training progresses.

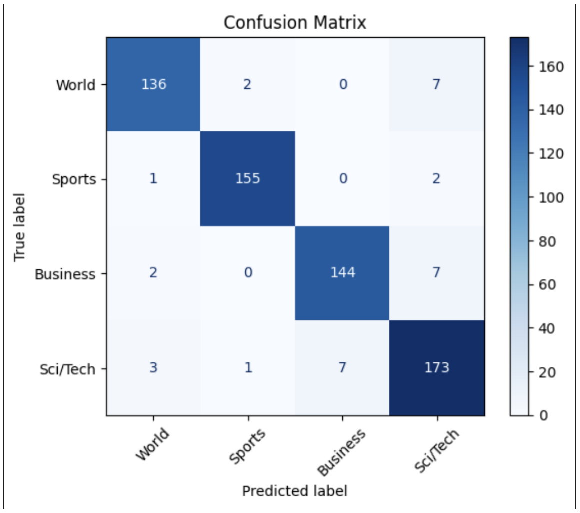

Figure 2: This represents a snapshot of final model performance using a confusion matrix.

Conclusion

This project demonstrates that parameter-efficient fine-tuning using Low-Rank Adaptation (LoRA) can effectively adapt large language models like RoBERTa for downstream tasks with minimal computational overhead. By injecting low-rank trainable adapters and carefully tuning hyperparameters such as rank and scaling factor, the model achieves strong performance on the AG News dataset while keeping the total number of trainable parameters under 1 million. These results highlight the feasibility of deploying transformer-based models in resource-constrained environments without compromising classification accuracy. Future work may investigate more granular LoRA configurations, adapter fusion across tasks, or complementary techniques like quantization and distillation to further enhance efficiency and transferability.

References

[1] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, L., & Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models. https://arxiv.org/pdf/2106.09685

[2] Hugging Face PEFT Library. Parameter-Efficient Fine-Tuning. https://github.com/huggingface/peft

[3] Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2022). 8-bit Optimizers via Block-wise Quantization. https://arxiv.org/pdf/2110.02861

[4] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020). Transformers: State-of-the-Art Natural Language Processing. In EMNLP. https://arxiv.org/pdf/1910.03771

[5] Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level Convolutional Networks for Text Classification. In NeurIPS. https://arxiv.org/pdf/1509.01626

[6] Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. In ICLR. https://arxiv.org/pdf/1412.6980

[7] Loshchilov, I., & Hutter, F. (2017). SGDR: Stochastic Gradient Descent with Warm Restarts. In ICLR. https://arxiv.org/pdf/1608.03983

[8] Dettmers, T., Lewis, M., Belkada, Y., & Zettlemoyer, L. (2023). QLoRA: Efficient Finetuning of Quantized LLMs. https://arxiv.org/pdf/2305.14314