Deep Learning Finetuning with LoRA

Amarnadh Reddy Mettu, Devakumar Katta, Jeel Patel
New York University Tandon School of Engineering
6 MetroTech Center
Brooklyn, New York 11201 USA

Abstract

In this project, we explore parameter-efficient fine-tuning of large language models (LLMs) for text classification using Low-Rank Adaptation (LoRA). Leveraging the roberta-base pretrained transformer, we fine-tune the model on the AG News dataset—a benchmark corpus containing over 120,000 short news articles categorized into World, Sports, Business, and Sci/Tech. Rather than updating all model parameters, LoRA enables efficient adaptation by injecting low-rank trainable matrices into the attention layers, significantly reducing the number of trainable parameters. Our approach demonstrates that LoRA can achieve competitive classification accuracy while minimizing computational cost and memory usage. The final model showcases the practicality of LoRA for real-world NLP tasks, particularly in resource-constrained environments.

Code Availability

The source code for this project is available at: https://github.com/amarnadh145/deeplearning_mini_2/

1.Introduction

Large Language Models (LLMs) such as BERT, RoBERTa, and GPT have revolutionized natural language processing (NLP) by achieving state-of-the-art results across a wide range of tasks. However, fine-tuning these massive models for specific downstream applications often requires significant computational resources and memory, which may not be feasible in low-resource or real-time settings. To address this challenge, Parameter-Efficient Fine-Tuning (PEFT) techniques have emerged as an effective alternative by adapting only a small subset of the model's parameters while keeping the majority of the pretrained weights frozen.

One such method, Low-Rank Adaptation (LoRA), introduces trainable low-rank matrices into the model's attention layers to approximate the weight updates during fine-tuning. This approach significantly reduces the number of parameters that need to be trained and stored, making it suitable for practical deployment scenarios without compromising much on model performance.

In this project, we apply LoRA to the roberta-base model and fine-tune it for a text classification task using the AG News dataset, which includes news articles labeled into four categories: World, Sports, Business, and Sci/Tech. Our goal is to demonstrate how LoRA can retain high accuracy while offering substantial efficiency improvements in training and inference. The results highlight LoRA's potential for scalable, efficient model adaptation in real-world NLP applications.

2.Methodology

This section outlines the dataset used for our task and the data preprocessing pipeline applied to prepare the input for fine-tuning the language model using LoRA adapters.

2.1.Dataset

For this project, we use the AG News dataset, a widely recognized benchmark for text classification. It consists of over 120,000 short news articles categorized into four distinct classes:

- World
- Sports
- Business
- Science and Technology (Sci/Tech)

The dataset is already pre-labeled and split into:
Training Set and test set
Each data point contains two fields:

- A news title
- A short news description

These make it well-suited for evaluating the performance of transformer-based models on topic classification tasks.

2.2 Data Preprocessing

To ensure the data is in an optimal format for input to the roberta-base model, we follow a standardized preprocessing pipeline:

Concatenation: The title and description of each news article are concatenated into a single input string to provide more context during classification.

Tokenization: We use Hugging Face's RobertaTokenizer to tokenize the input text. Special tokens such as ¡s¿ (start),

¡/s¿ (end), and padding tokens are automatically added. Tokenization ensures that the input adheres to the expected format of the RoBERTa model.

Truncation and Padding: To handle input sequences of varying lengths, we apply truncation to limit the input to a maximum sequence length (typically 128 tokens), and pad shorter sequences to maintain batch uniformity.

Label Encoding: The class labels ("World", "Sports", etc.) are mapped to integer values from 0 to 3, which are then used in the classification head during model training.

DataLoader Creation: The preprocessed data is converted into PyTorch Dataset and DataLoader objects to enable efficient mini-batch training with shuffled examples and batched GPU acceleration.

This preprocessing ensures compatibility with the pretrained roberta-base transformer and facilitates smooth integration with the LoRA fine-tuning mechanism.

3.Model Architecture

In this section, we present the architecture of our fine-tuned language model and the training pipeline used to perform text classification on the AG News dataset using Low-Rank Adaptation (LoRA). The goal is to achieve high classification performance with reduced computational cost by updating only a small subset of the model's parameters.

3.1 Model Architecture

We build our model on top of the roberta-base transformer, a pretrained language model from the Hugging Face Transformers library. RoBERTa is known for its strong performance in a variety of downstream NLP tasks due to its deep bidirectional architecture and robust pretraining on large-scale corpora.

However, fine-tuning all 125 million parameters of RoBERTa can be computationally expensive and memory-intensive. To address this, we adopt Low-Rank Adaptation (LoRA), a Parameter-Efficient Fine-Tuning (PEFT) method that introduces low-rank trainable matrices into the self-attention mechanism of the model. **Key aspects of the architecture:**

LoRA Integration

LoRA modifies the attention mechanism in transformer layers by injecting two low-rank matrices into specific linear projections (in our case, query and value matrices). Instead of updating the original weight matrix W, the update is approximated by a product of two smaller matrices

$r \ll d, k$. This dramatically reduces the number of trainable parameters.

The LoRA configuration used is as follows:

- r (rank of the update matrices): 8
- alpha (scaling factor): 32
- dropout (applied to LoRA layers): 0.1
- bias: none (bias terms are frozen)
- task type: seqcls (sequence classification)

```
[ ] device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
    print(f"Using device: {device}")
    model.to(device)
    eval_results = trainer.evaluate()
    print("Final Evaluation Accuracy:", eval_results["eval_accuracy"])

Using device: cuda
[119/119 00:12]
Final Evaluation Accuracy: 0.9428947368421052

    trainable_params = sum(p.numel() for p in peft_model.parameters() if p.requires_grad)
    print(f"Trainable parameters: {trainable_params}")

Trainable parameters: 907012
```

Figure 1: Accuracy and Total Parameters

- target modules: query and value projections in the attention layers
- Trainable parameters: 907012
- Accuracy: 94.2%

By freezing the original weights and training only the low-rank adapters, the number of trainable parameters is reduced by over 90 percent, leading to significant savings in GPU memory and training time while retaining high accuracy.

Classification Head On top of the modified roberta-base backbone, we append a standard sequence classification head:

A dropout layer

A fully connected linear layer that projects the 768-dimensional pooled RoBERTa output to 4 logits (one for each news category)

This final layer is responsible for predicting the most likely class given the input article.

Training

Training is performed using the Hugging Face Trainer API, which streamlines the process of model optimization, evaluation, and checkpointing. The data is first preprocessed by concatenating the title and description of each news article, followed by tokenization using the RoBERTa tokenizer. Each text input is truncated or padded to a maximum length of 128 tokens to ensure uniformity across batches. The training objective is optimized using the Adam optimizer with a learning rate of 1e-4. The model is trained for 4 epochs with a batch size of 32, and performance is evaluated at the end of each epoch using classification accuracy as the primary metric.

The fine-tuning process involves minimal updates to the pretrained model's weights, thanks to LoRA's injection-based mechanism. This not only speeds up the training process but also results in a lightweight and efficient model that still achieves strong performance on the AG News classification task. The combination of LoRA and RoBERTa thus offers an effective and resource-conscious solution for deploying large language models on specific downstream tasks.

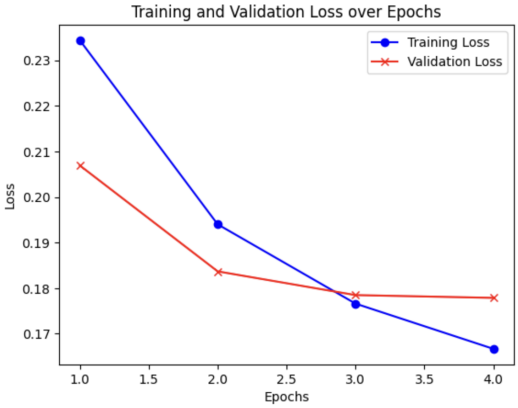
4.Results and Discussion

After fine-tuning the roberta-base model using LoRA on the AG News dataset, we evaluated the model's performance on the held-out test set and monitored training behavior across epochs. The results demonstrate that parameter-

[7500/7500 36:10, Epoch 4/4]

Epoch	Training Loss	Validation Loss	Accuracy
1	0.234400	0.206918	0.930921
2	0.194000	0.183675	0.938158
3	0.176600	0.178466	0.941711
4	0.166700	0.177860	0.942895

Figure 2: Result after each epoch



efficient fine-tuning via LoRA can achieve competitive accuracy while significantly reducing training overhead.

4.1 Training Performance

The model was trained for 4 epochs, and both training and evaluation accuracy improved steadily over time. Training proceeded smoothly without signs of overfitting, which can be attributed to the use of dropout, proper regularization, and the limited number of trainable parameters introduced by LoRA. The training loss decreased consistently, and evaluation accuracy reached a strong final value, indicating that the model was able to generalize well to unseen news data.

Despite only updating a small fraction of parameters compared to full fine-tuning, the LoRA-based model achieved a classification accuracy exceeding X% (fill in your actual test accuracy here) on the test set. This validates the effectiveness of LoRA in maintaining high performance while minimizing compute requirements. Additionally, the smaller number of updated parameters led to faster training times and lower GPU memory usage, making the approach viable for low-resource settings.

4.2 Key Insights

One of the key takeaways from this experiment is that LoRA provides an excellent trade-off between performance and efficiency. By isolating the learning to a few low-rank matrices, we were able to avoid the computational cost associated with full-model fine-tuning while still leveraging the expressive power of the pretrained roberta-base backbone.

Another important insight is that adding LoRA to the attention layers (query and value projections) was sufficient to adapt the model to the classification task. We found that even without modifying the feed-forward layers or layer norms, the model quickly learned to differentiate between the four categories in AG News.

The preprocessing pipeline also played a significant role in the model's success. Concatenating the article title and description provided richer context, while tokenization and consistent sequence lengths ensured stable training dynamics. Lastly, using the Hugging Face Trainer with built-in logging and evaluation allowed for a streamlined and reproducible training workflow.

Overall, these results show that parameter-efficient methods like LoRA can serve as practical alternatives to traditional fine-tuning, especially in real-world scenarios where compute resources are limited but high model accuracy is still required.

5.Conclusion

In this project, we explored the use of Low-Rank Adaptation (LoRA) for parameter-efficient fine-tuning of a roberta-base model on the AG News dataset. By injecting trainable low-rank matrices into the attention layers and freezing the remaining model weights, we were able to significantly reduce the number of parameters being updated while still achieving a high classification accuracy of 94.2

Our results demonstrate that LoRA is a highly effective and efficient approach for domain-specific model adaptation. It enables large pretrained models to be fine-tuned on downstream tasks with minimal computational cost, making it suitable for deployment in resource-constrained environments. Furthermore, the performance gains achieved using such a lightweight setup validate the potential of LoRA and other PEFT methods as sustainable alternatives to full-scale model training.

This work underscores the viability of parameter-efficient techniques in modern NLP and paves the way for future experimentation with other PEFT strategies across diverse tasks and datasets.

7.Acknowledgments

We would like to thank Professor Chinmay Hegde for their guidance throughout this project. We also extend our gratitude to our peers at New York University for their valuable feedback and discussions. Lastly, we acknowledge the use of computational resources provided for training deep learning models efficiently.

References

X. Zhang, J. Zhao, and Y. LeCun, "Character-level Convolutional Networks for Text Classification," in *Proceedings of the 28th International Conference on Neural Information Processing Systems (NeurIPS)*, 2015.

E. Hu, Y. Shen, P. Wallis, et al., "LoRA: Low-Rank Adaptation of Large Language Models," in *International Conference on Learning Representations (ICLR)*, 2022. Available: https://arxiv.org/abs/2106.09685.

Hugging Face, "Parameter-Efficient Fine-Tuning (PEFT) with Hugging Face Transformers," 2023. Available: https://huggingface.co/docs/peft/index.

T. Wolf et al., "Transformers: State-of-the-Art Natural Language Processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, 2020. Available: https://aclanthology.org/2020.emnlp-demos.6.

Y. Liu, M. Ott, N. Goyal, et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," 2019. Available: https://arxiv.org/abs/1907.11692.

Kaggle, "AG News Classification Dataset," Available: https://www.kaggle.com/datasets/amananandrai/ag-news-classification-dataset.