DSA4213 Ass 3:
A Comparison of Full Fine-tuning and LoRA

Ng Zhi Min
A0255864R

Abstract

This report compares two fine-tuning strategies for adapting FinBERT to financial sentiment analysis: full fine-tuning and Low-Rank Adaptation (LoRA). Full fine-tuning achieved 86.02% accuracy using all 110M parameters, while LoRA updated only 0.27% but reached 54.50%. We analyze the performance gap, efficiency trade-offs, and key lessons for parameter-efficient fine-tuning.

1 Introduction

Large pretrained transformer models have revolutionized natural language processing by providing powerful feature representations that can be adapted to downstream tasks through fine-tuning. However, as models grow larger, fine-tuning all parameters becomes computationally expensive and impractical. This motivates parameter-efficient fine-tuning methods like LoRA, which aim to achieve comparable performance while updating only a small fraction of model parameters.

This work investigates the effectiveness of full fine-tuning versus LoRA for financial sentiment analysis, using FinBERT as the base model and the Financial PhraseBank dataset. Our contributions include: (1) empirical comparison of both approaches on a domain-specific task, (2) analysis of the efficiency-performance trade-off, and (3) insights into challenges encountered with parameter-efficient methods.

2 Dataset and Task

2.1 Financial PhraseBank Dataset

We use the Financial PhraseBank, a widely-used benchmark for financial sentiment analysis. The dataset contains 4,217 sentences from financial news articles, manually annotated by domain experts with sentiment labels. We specifically use the subset where at least 66% of annotators agreed on the label ("Sentences_66Agree.txt"), ensuring annotation quality.

Label Distribution:

- Neutral: 2,535 samples (60.1%)
- Positive: 1,168 samples (27.7%)
- Negative: 514 samples (12.2%)

The dataset exhibits class imbalance, with neutral sentiment dominating. This reflects real-world financial news where most statements are factual rather than overtly positive or negative.

2.2 Data Split

Data were stratified into 80% training (3,373), 10% validation (422), and 10% test (422) sets.

2.3 Motivation

Financial sentiment analysis is chosen for several reasons:

1. **Domain-specific language**: Financial texts contain specialized terminology (e.g., “EBITDA,” “diluted earnings”) that benefits from domain-adapted models.

2. **Practical importance**: Accurate sentiment detection aids algorithmic trading, risk assessment, and market analysis.

3. **Suitable scale**: The dataset size (4,217 samples) is realistic for fine-tuning scenarios where large labeled datasets are unavailable.

3 Model and Fine-tuning Strategies

3.1 Base Model: FinBERT

We use FinBERT (`yiyanghkust/finbert-tone`), a BERT-based model pretrained on financial corpora. The model contains approximately 110 million parameters and is already adapted to financial language, making it ideal for our task.

3.2 Strategy 1: Full Fine-tuning

In full fine-tuning, we update all 109,754,115 parameters of FinBERT to adapt it to our specific three-class sentiment classification task.

Hyperparameters:

- Learning rate: 2×10^{-5}
- Batch size: 32 (per device)
- Epochs: 2
- Weight decay: 0.01
- Optimizer: AdamW
- Max sequence length: 128 tokens
- Mixed precision: FP16 (when GPU available)

3.3 Strategy 2: LoRA (Low-Rank Adaptation)

LoRA introduces trainable low-rank matrices into specific layers while freezing the pretrained weights. For a weight matrix $W_0 \in R^{d \times k}$, the update is parameterized as:

$$W = W_0 + \Delta W = W_0 + BA$$

where $B \in R^{d \times r}$ and $A \in R^{r \times k}$ with rank $r \ll \min(d, k)$.

LoRA Configuration:

- Rank (r): 8
- Alpha (α): 32 (scaling factor = $\alpha/r = 4$)
- Target modules: Query and Value projection matrices
- LoRA dropout: 0.05

- Trainable parameters: 297,219 (0.27% of total)

Training Hyperparameters:

- Learning rate: 5×10^{-5} (higher than full fine-tuning)
- Other hyperparameters: Same as full fine-tuning

4 Experimental Setup

4.1 Implementation Details

Both methods were implemented using Hugging Face Transformers and PEFT on Google Colab (Tesla T4 GPU).

4.2 Evaluation Metrics

We use standard classification metrics:

- **Accuracy**: Overall correct predictions
- **F1-score (weighted)**: Harmonic mean of precision and recall, weighted by class support
- **F1-score (per-class)**: F1 for each sentiment class
- **Training time**: Wall-clock time for complete training

5 Results

5.1 Full Fine-tuning Results

Full fine-tuning achieved strong performance on the test set:

Table 1: Full Fine-tuning Performance

Metric	**Score**
Accuracy	0.8602
F1 (weighted)	0.8604
F1 (Negative)	0.7308
F1 (Neutral)	0.9071
F1 (Positive)	0.8162
Training Time	145 minutes
Trainable Parameters	109,754,115 (100%)
Per-class Results:	
Negative - Precision	0.72
Negative - Recall	0.74
Neutral - Precision	0.91
Neutral - Recall	0.91
Positive - Precision	0.82
Positive - Recall	0.82

The model performs best on the neutral class (F1=0.91), which is the majority class. Performance on positive sentiment (F1=0.82) is strong, while negative sentiment (F1=0.73) is lower, likely due to fewer training examples (514 vs 2,535 neutral).

5.2 LoRA Results

Our LoRA implementation encountered significant challenges:

Table 2: LoRA Performance

Metric	Score
Accuracy	0.5450
F1 (weighted)	0.5073
F1 (Negative)	0.0312
F1 (Neutral)	0.7204
F1 (Positive)	0.2523
Training Time	107 minutes
Trainable Parameters	297,219 (0.27%)
Per-class Results:	
Negative - Precision	0.08
Negative - Recall	0.02
Neutral - Precision	0.66
Neutral - Recall	0.79
Positive - Precision	0.27
Positive - Recall	0.24

The LoRA model shows severely degraded performance, barely exceeding random guessing for minority classes.

5.3 Comparative Analysis

Table 3: Full Fine-tuning vs LoRA Comparison

Metric	Full Fine-tuning	LoRA
Accuracy	86.02%	54.50%
F1 (weighted)	0.8604	0.5073
Trainable Params	109.8M (100%)	297K (0.27%)
Param Reduction	–	99.73%
Training Time	145 min	107 min
Time Reduction	–	26.2%

6 Analysis and Discussion

Full fine-tuning achieved substantially better performance than LoRA, reaching 86% accuracy compared to LoRA's 54.5%. This can be explained by several factors. With all 110M parameters trainable, the fully fine-tuned FinBERT had greater capacity to capture task-specific patterns in financial sentiment, benefiting from its strong initialization on financial corpora. The relatively modest dataset size (3,373 training samples) was still sufficient for effective adaptation without overfitting, and the stratified split helped maintain balanced learning across sentiment classes.

In contrast, LoRA's underperformance stemmed mainly from limited model adaptation. By only updating 0.27% of parameters and restricting low-rank updates to the Query and Value matrices, the model lacked sufficient flexibility to learn nuanced sentiment distinctions. The chosen rank ($r = 8$) was likely too low for a task requiring rich contextual understanding, while the learning rate (5×10^{-5}) and short training duration (two epochs) may have hindered convergence. Additionally, LoRA often benefits from applying updates to more layers—such as Key, Dense, or Feed-Forward projections—and from extended training. Together, these limitations suggest that while LoRA offers substantial efficiency gains, its success depends critically on proper configuration and tuning, whereas full fine-tuning remains more robust and reliable for smaller, domain-specific datasets.

6.1 Efficiency Trade-offs

Despite lower accuracy, LoRA cut trainable parameters by 99.7%, reduced training time by 26%, and required far less memory and storage, highlighting its value for resource-limited deployments.

In production scenarios with properly tuned LoRA, these efficiency gains become critical for:

- Serving multiple task-specific adapters with a single base model
- Rapid experimentation and iteration
- Deployment on edge devices with limited resources

6.2 Error Analysis

Examining the confusion matrices (not shown due to space constraints), we observe:

Full Fine-tuning:

- Misclassifications primarily occur between neutral and positive sentiment (semantically similar)
- Negative class has lower recall, reflecting data scarcity
- Strong diagonal dominance indicates good generalization

LoRA:

- Model exhibits extreme bias toward neutral class (majority)
- Negative class nearly completely misclassified (F1=0.03)
- Suggests the model defaulted to a simplistic decision rule

7 Lessons Learned and Limitations

7.1 Key Takeaways

1. **Full fine-tuning remains the gold standard**: When computational resources permit and performance is critical, full fine-tuning provides reliable, strong results.
2. **LoRA requires careful tuning**: Parameter-efficient methods are sensitive to hyperparameters. Our implementation demonstrates that naive application fails catastrophically.
3. **Domain adaptation matters**: Starting with a domain-adapted model (FinBERT) significantly aids task performance compared to general-purpose BERT.
4. **Class imbalance is challenging**: Even with stratified splitting, the 5:1 ratio between neutral and negative classes affects minority class performance.
5. **Debugging is crucial**: The poor LoRA results highlight the importance of monitoring training curves and validating assumptions during implementation.

7.2 Limitations

1. **Single Dataset**: Evaluation on only Financial PhraseBank limits generalizability of findings.
2. **Suboptimal LoRA**: Our LoRA implementation doesn't represent the method's full potential due to configuration issues.
3. **Limited Hyperparameter Search**: Time constraints prevented exhaustive hyperparameter tuning, particularly for LoRA.
4. **Hardware Constraints**: Training on Colab's free tier introduces variability and limits batch size/training duration.
5. **No Statistical Significance Testing**: Results are from single runs without error bars or confidence intervals.

7.3 Future Work

To improve this study:

- Increase LoRA rank to $r \in [16, 32]$ and target more modules
- Compare with other PEFT methods.
- Tune learning rate and train longer.
- Evaluate on additional financial datasets (e.g., Financial News, FiQA)

8 Conclusion

This work compared full fine-tuning and LoRA for adapting FinBERT to financial sentiment analysis. Full fine-tuning achieved 86% accuracy, demonstrating effective domain adaptation with all 110M parameters trained. Our LoRA implementation, while 99.73% more parameter-efficient, achieved only 54% accuracy due to suboptimal configuration, highlighting that parameter-efficient methods require careful engineering.

The results underscore that while PEFT methods like LoRA promise significant resource savings, they are not "free lunch" solutions. Successful deployment demands:

1. Thoughtful architecture decisions (rank, target modules)
2. Extensive hyperparameter tuning
3. Potentially longer training than full fine-tuning
4. Domain-specific validation and error analysis

When properly configured (as shown in recent literature), LoRA can match full fine-tuning performance while updating $<1\%$ of parameters. Our work demonstrates both the potential and pitfalls of this approach, providing practical insights for practitioners considering parameter-efficient fine-tuning.