Advanced NLP Assignment 3: Comparison of Fine-tuning Methods for GPT-2 on Summarization Task

Himanshu Pal
2023701003

October 28, 2024

1 Experimental Setup

1.1 Model Architecture and Training Configuration

The experiment utilized the GPT-2 small model as the base architecture with the following configurations:

- **Base Model**: GPT-2 Small
- **Task**: Text Summarization
- **Loss Function**: GPT-2 standard language modeling loss (cross-entropy)

1.2 Hyperparameters

The training process used the following hyperparameters consistently across all three methods:

Parameter	Value
Number of Epochs	10
Learning Rate	1e-4
Optimizer	AdamW

2 Results and Analysis

2.1 Performance Metrics

The following table presents the comparative results of the three fine-tuning methods:

Method	Test Loss	ROUGE-1	ROUGE-2	ROUGE-L
Prompt Tuning	8.2943	0.1864	0.1008	0.1308
LoRA	3.1044	0.1502	0.0667	0.1042
Last Layer	6.5138	0.1613	0.0708	0.1124

Table 1: Comparative Performance Metrics

2.2 Method-wise Analysis

2.2.1 Prompt Tuning

The prompt tuning method demonstrated the following characteristics:

- Highest test loss ($\mathcal{L} = 8.2943$)
- Best ROUGE scores among all methods:
  - ROUGE-1: 0.1864
  - ROUGE-2: 0.1008
  - ROUGE-L: 0.1308
- Shows superior performance in capturing both unigram and bigram overlaps

2.2.2 LoRA (Low-Rank Adaptation)

LoRA exhibited the following performance characteristics:

- Lowest test loss ($\mathcal{L} = 3.1044$)
- Moderate ROUGE scores:
  - ROUGE-1: 0.1502
  - ROUGE-2: 0.0667
  - ROUGE-L: 0.1042
- Best optimization performance despite lower ROUGE scores

2.2.3 Last Layer Tuning

The traditional fine-tuning of last layers showed:

- Moderate test loss ($\mathcal{L} = 6.5138$)
- ROUGE scores comparable to LoRA:
  - ROUGE-1: 0.1613
  - ROUGE-2: 0.0708
  - ROUGE-L: 0.1124
- Balance between Optimization and Generating Quality

3 Comparative Analysis

3.1 Loss Analysis

The test loss values demonstrate significant variations across methods:

$$\mathcal{L}_{\text{LoRA}}(3.1044) < \mathcal{L}_{\text{LastLayer}}(6.5138) < \mathcal{L}_{\text{Prompt}}(8.2943)$$

This ordering suggests that LoRA achieves the most efficient optimization, while prompt tuning struggles with convergence despite producing better summaries.

3.2 ROUGE Score Analysis

The ROUGE metrics reveal an interesting pattern:

- Prompt tuning consistently outperforms other methods across all ROUGE metrics
- The relative performance ordering remains consistent:

$$\text{Prompt Tuning} > \text{Last Layer} > \text{LoRA}$$

- The gap between methods is most pronounced in ROUGE-2 scores, indicating varying capabilities in capturing phrase-level patterns

4 Key Findings

1. **Trade-off Pattern**: An inverse relationship exists between test loss and ROUGE scores, suggesting that lower loss doesn't necessarily translate to better summarization quality.
2. **Method Effectiveness**:
   - LoRA excels in optimization but produces lower quality summaries, could be issue with truncation of train dataset
   - Prompt tuning struggles with optimization but generates better summaries
   - Last layer tuning provides a middle-ground solution
3. **Performance Stability**: All methods maintain consistent relative performance across different ROUGE metrics, indicating stable behavior across different evaluation criteria.

Metric	**Task1 (PT)**	**Task2 (LoRA)**	**Task3 (FT)**
Total Parameters	124.4M	125.0M	124.4M
Trainable Parameters	124.4M	589.8K	124.4M
Execution Time	03:24:55	06:56:01	10:29:48

Table 2: Comparison of Model Parameters and Execution Times (PT: Prompt Tuning, LoRA: Low-Rank Adaptation, FT: Fine Tuning)