Slimming Down LLMs Without Losing Their Minds

Qingda(Michael) Mai

University of Waterloo

March 6, 2025

Introduction

- Poorly understood of the necessity to train a large networks with hundreds of billions of parameters
- Training costs grow exponentially
- Required parameter-efficient training techniques as an approach to training large neural networks

Main Research Questions

We focus on two research questions:

Is there a significant improvement for LLM after re-fine? How?

We usually have a dataset used to do supervised learning on the model, so that this model can behavour better in this specific task/domain. But is that true?

What's the Math behind the `parameter-efficient fine-tuning methods ?(LoRA, QLoRA, full fine-tune)`

By focusing on the mathematics behind these methods, we can develop a more fundamental understanding of:
(1) why these cost-efficient methods can help us to improve the performance of the LLM.

Data Sources, Benchmark, Statistical Methods

- Foundation Model: TinyLlama-1.1B and more (details shown in the last few slides)
- Training Data:

  Alpaca dataset (52K self-instruct examples)

- Evaluation Benchmarks

  HellaSwag (commonsense reasoning)
  GSM8K (mathematical reasoning)
  MMLU subset (knowledge-based tasks)

Expected Outcomes & Significance, Q & A

- Expected Findings:

  Quantified performance-efficiency tradeoffs across methods

- Significance:

Empirical foundation for further optimization of parameter-efficient methods

Any Question or Suggestion?

Our experiment compares different parameter-efficient fine-tuning methods across various model sizes using a unified instruction-following dataset.

Table 1: Model and Training Method Comparison Matrix

Size Category	Model	Base	Full	LoRA	QLoRA
Tiny (1-1.5B)	TinyLlama-1.1B	✓	✓	✓	✓
Small (2-3B)	Phi-2 (2.7B)	✓	✓	✓	✓
Medium (7B)	Mistral-7B	✓	55*	✓	✓
Medium (7B)	LLaMA-2-7B	✓	55*	✓	✓

* Full fine-tuning omitted for medium-sized models for constraints

1. **For each model size category:**
   - Evaluate the **base model** performance (without fine-tuning)
   - Apply each fine-tuning method (Full, LoRA, QLoRA) to the model
   - Evaluate the fine-tuned model's performance on benchmark tasks
2. **Resulting model variants (14 total):**
   - **TinyLlama:** Base + Full + LoRA + QLoRA = 4 variants

- **Phi-2:** Base + Full + LoRA + QLoRA = 4 variants
- **Mistral-7B:** Base + LoRA + QLoRA = 3 variants
- **LLaMA-2-7B:** Base + LoRA + QLoRA = 3 variants

The experiment differs take into research questions' related design in several ways:

- **Unified Fine-tuning Dataset:** We now use a single instruction dataset (Alpaca) rather than multiple task-specific datasets
- **Focus on Fine-tuning Methods:** We prioritize comparing different parameter-efficient methods over task-specific adaptations
- **Model Size Comparison:** We explicitly compare models across different parameter scales (1B to 7B)
- **Hardware Optimization:** The experiment is optimized for macOS with Apple Silicon, using MPS acceleration

All models will be evaluated on multiple benchmarks:

- **HellaSwag:** Common sense reasoning capabilities
- **GSM8K:** Mathematical reasoning abilities
- **MMLU:** Multi-task language understanding