Mathematical Foundations of Fine-Tuning and Retrieval in Domain-Specific LLMs

Amir Dhillon

June 16, 2025

1 LoRA: Low-Rank Adaptation

Instead of fine-tuning the full weight matrix $W \in \mathbb{R}^{d \times k}$, LoRA learns a low-rank update:

$$\Delta W = AB, \quad \text{where } A \in \mathbb{R}^{d \times r}, B \in \mathbb{R}^{r \times k}, r \ll \min(d, k)$$

The updated output becomes:

$$\hat{y} = Wx + \Delta W x = Wx + ABx$$

Only A and B are updated, while W remains frozen.

2 Cosine Similarity for Semantic Retrieval

Used to rank document chunks against a user query:

$$\cos(\theta) = \frac{u \cdot v}{\|u\|\|v\|} = \frac{\sum u_i v_i}{\sqrt{\sum u_i^2}\sqrt{\sum v_i^2}}$$

A similarity of 1 means identical direction; 0 means orthogonal; -1 means opposite.

3 Self-Attention (Simplified)

A fundamental component of transformers:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Where $Q, K, V \in \mathbb{R}^{L \times d}$. Each token computes a weighted average of all tokens.

4 Effective Batch Size with Gradient Accumulation

Given:

- Per device batch size = 2
- Gradient accumulation steps = 4
- Number of GPUs = 1

Then:

$$\text{Effective Batch Size} = 2 \times 4 \times 1 = 8$$

This simulates a larger batch without increasing GPU memory usage.