Efficient Variants of BERT: DistilBERT and ALBERT

Youssef Salah

Abstract

This report discusses two parameter-efficient fine-tuning techniques for large language models: LoRA (Low-Rank Adaptation) and QLoRA (Quantized LoRA). These methods allow large models to be adapted to downstream tasks with minimal computational and memory overhead. We analyze their underlying principles, training strategies, and performance trade-offs.

Contents

1 Introduction

Fine-tuning large language models like BERT or GPT can be computationally expensive due to the large number of parameters. Parameter-efficient fine-tuning (PEFT) techniques, such as LoRA and QLoRA, allow models to be adapted to specific tasks while only updating a small subset of parameters. This reduces memory usage, accelerates training, and makes large models accessible to environments with limited resources.

1.1 Motivation

Full fine-tuning requires storing and updating all model parameters, which is often infeasible for models with billions of parameters. LoRA and QLoRA address this limitation by introducing low-rank adapters and quantization strategies, enabling efficient fine-tuning without degrading model performance.

2 LoRA: Low-Rank Adaptation

2.1 Core Idea

LoRA, proposed by Hu et al. (2021), updates only a small, low-rank portion of the model's weight matrices instead of full fine-tuning. For a given weight matrix W, LoRA computes:

$$W' = W + \Delta W = W + BA \tag{1}$$

where $A \in R^{r \times d}$ and $B \in R^{d \times r}$, and $r \ll d$. This allows the model to adapt to new tasks by training only the low-rank matrices A and B, while keeping the original weights frozen.

2.2 Advantages

- Reduces the number of trainable parameters dramatically.
- Freezes the base model, reducing memory and computation requirements.
- No increase in inference latency since adapters are merged into the original weights after training.

2.3 Training Objective

LoRA typically applies the standard loss of the downstream task (e.g., cross-entropy for classification) while updating only the adapter matrices. Optionally, multiple adapters can be stacked or injected at different layers of the model to improve task adaptation.

2.4 Performance and Trade-offs

- Maintains performance comparable to full fine-tuning.
- Drastically reduces GPU memory consumption.
- Best suited for environments where updating all model weights is infeasible.

3 QLoRA: Quantized LoRA

3.1 Core Idea

QLoRA, introduced by Dettmers et al. (2023), extends LoRA by combining **4-bit quantization** with low-rank adapters. The base model weights are quantized, and only the LoRA adapters are trained, allowing extremely large models to be fine-tuned on limited GPU memory.

3.2 Quantization Strategy

QLoRA uses advanced quantization techniques:

- **4-bit representation:** Reduces memory usage for the base model by 75%.
- **NF4 encoding and double quantization:** Preserve accuracy while lowering memory footprint.

3.3 Training Pipeline

1. Quantize the pretrained base model to 4-bit precision.
2. Freeze the quantized weights.
3. Insert LoRA adapters into target layers.
4. Train only the adapter matrices for the downstream task.

3.4 Performance and Trade-offs

- Enables fine-tuning of very large models (e.g., 65B parameters) on a single GPU.
- Achieves performance close to full fine-tuning.
- Extremely low memory requirement compared to standard LoRA.

4 Comparison: LoRA vs QLoRA

Feature	**LoRA**	**QLoRA**
Base Model Precision	FP16/FP32	4-bit quantized
Trainable Parameters	Low-rank adapters only	Low-rank adapters only
Memory Usage	Medium	Very low
Performance	Comparable to full fine-tuning	Comparable to full fine-tuning
Scalability	Up to 10B models on multi-GPU	Up to 65B models on single GPU
Inference Latency	Unchanged	Unchanged

Table 1: Comparison of LoRA and QLoRA

5 Conclusion

LoRA and QLoRA enable parameter-efficient fine-tuning of large language models. LoRA reduces training parameters via low-rank adaptation, while QLoRA further incorporates quantization to allow extremely large models to be fine-tuned on limited hardware. Both methods maintain strong performance while drastically reducing memory usage, making them essential techniques for practical NLP applications.

References

- Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, L., Chen, W., & Chen, W. (2021). *LoRA: Low-Rank Adaptation of Large Language Models.* https://arxiv.org/abs/2106.09685
- Dettmers, T., et al. (2023). *QLoRA: Efficient Finetuning of Quantized LLMs.* https://arxiv.org/abs/2305.14314