School of Computer Engineering
Iran University of Science and Technology

Efficient Fine-Tuning of Llama-3 via Quantized Low-Rank Adaptation (QLoRA)

A Comparative Study of LoRA (Hu et al., 2021) and 4-bit Quantization Techniques

Project Lead:

Parmis Vazifeh

Team members:

P. Vazifeh · O. Farrokhi

December 22, 2025

Abstract

The introduction of **LoRA (Low-Rank Adaptation)** by Hu et al. (2021) revolutionized the adaptability of Large Language Models (LLMs) by reducing trainable parameters by up to 10,000x. However, loading the base model still requires significant GPU memory. Addressing the 2023/2024 trend of "Democratizing AI," this project implements and benchmarks **QLoRA**, which combines LoRA with **4-bit Normal Float (NF4)** quantization. The team will fine-tune the state-of-the-art **Llama-3-8B** model on a domain-specific dataset (e.g., Medical or Code), analyzing the trade-off between memory efficiency and generation quality compared to standard LoRA.

Contents

1 Introduction

Full fine-tuning of a 7B parameter model requires over 100GB of VRAM, making it inaccessible for most researchers. LoRA solved the *parameter* problem, but not the *memory* problem of loading the base model weights.

This project explores the intersection of the baseline paper (LoRA) with modern quantization techniques (Dettmers et al., 2023). We aim to prove that extreme compression (4-bit) combined with low-rank adapters can achieve performance comparable to 16-bit training while running on consumer hardware (e.g., Google Colab T4).

2 Theoretical Framework

2.1 1. The Baseline: LoRA (2021)

Hu et al. propose that weight updates ΔW in pre-trained models have a low "intrinsic rank." Instead of updating the full matrix $W \in \mathbb{R}^{d \times k}$, LoRA decomposes the update into two smaller matrices $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$, where $r \ll d$:

$$h = W_0 x + \Delta W x = W_0 x + BAx \tag{1}$$

Here, W_0 remains frozen, and only A and B are trained. The scaling factor is $\frac{\alpha}{r}$.

2.2 2. The Modern Twist: QLoRA (2023)

QLoRA enhances LoRA by quantizing the frozen weights W_0 to 4-bit precision using the **NF4 (Normal Float 4)** data type, which is information-theoretically optimal for normally distributed weights.

$$W_{QLoRA} = \text{dequant}(c_2, \text{dequant}(c_1, W_{4bit})) \tag{2}$$

This allows loading a 7B model in just 4GB of VRAM, leaving space for LoRA gradients.

3 System Architecture

3.1 Tech Stack

- **Base Model:** ‘meta-llama/Meta-Llama-3-8B‘ (The 2024 Standard).

- **Libraries:** ‘HuggingFace Transformers‘, ‘PEFT‘ (Parameter-Efficient Fine-Tuning), ‘bitsandbytes‘ (for quantization).
- **Dataset:** ‘PubMedQA‘ (Medical) or ‘StackOverflow‘ (Coding).

3.2 Training Pipeline

The project requires implementing two distinct training runs:

1. **Standard LoRA (fp16):** Loading the model in 16-bit precision.
2. **QLoRA (nf4):** Loading the model in 4-bit precision with Double Quantization.

4 Project Requirements

4.1 1. Adapter Configuration Analysis

Students must experiment with the Hyperparameters defined in the baseline paper:

- **Rank (r):** Test $r = 8$ vs. $r = 64$. Does increasing rank improve reasoning in Llama-3?
- **Target Modules:** Apply LoRA only to ‘q$_p roj, v_p roj$‘$(Attention) vs.$‘$all-linear$‘$layers$.

4.2 2. Memory Profiling

A key deliverable is measuring the hardware footprint. Use ‘nvidia-smi‘ or PyTorch profilers to log:

- Peak VRAM usage during training.
- Training time per epoch.

4.3 3. Implementation Logic

```python
from peft import LoraConfig, get_peft_model
from transformers import BitsAndBytesConfig

# 1. Define 4-bit Quantization (The Modern Part)
bnb_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_use_double_quant=True,
    bnb_4bit_compute_dtype=torch.bfloat16
)
```

```
11
12 # 2. Define LoRA Adapters (The Baseline Part)
13 peft_config = LoraConfig(
14     r=16,
15     lora_alpha=32,
16     target_modules=["q_proj", "v_proj"],
17     lora_dropout=0.05,
18     bias="none",
19     task_type="CAUSAL_LM"
20 )
```

Listing 1: QLoRA Configuration Code

5 Evaluation Methodology

5.1 Metrics

- **Perplexity (PPL):** Lower is better. Does 4-bit quantization degrade PPL significantly compared to 16-bit?
- **Inference Speed (Tokens/sec):** Impact of de-quantization overhead during generation.

5.2 Qualitative Analysis

Provide side-by-side generation examples. * *Prompt:* "Diagnose a patient with symptoms X, Y, Z." * *Base Model:* [Generic Output] * *QLoRA Fine-tuned:* [Specific Medical Output]

6 Conclusion

This project demonstrates the practical application of Hu et al.'s theory in the modern era of limited compute resources. By successfully fine-tuning Llama-3 on consumer hardware, the team validates that Low-Rank Adaptation, combined with aggressive quantization, is the key to accessible AI development.

References

[1] Hu, E. J., et al. (2021). LoRA: Low-Rank Adaptation of Large Language Models. *ICLR*.

[2] Dettmers, T., et al. (2023). QLoRA: Efficient Finetuning of Quantized LLMs. *NeurIPS*.

[3] Meta AI. (2024). Llama 3 Model Card.