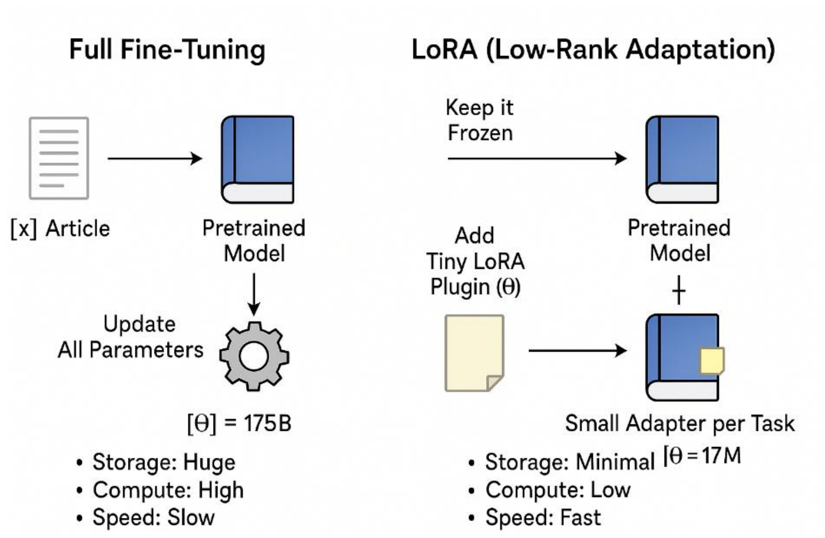
LORA & QLORA

LORA & QLORA: EFFICIENT FINE-TUNING TECHNIQUES

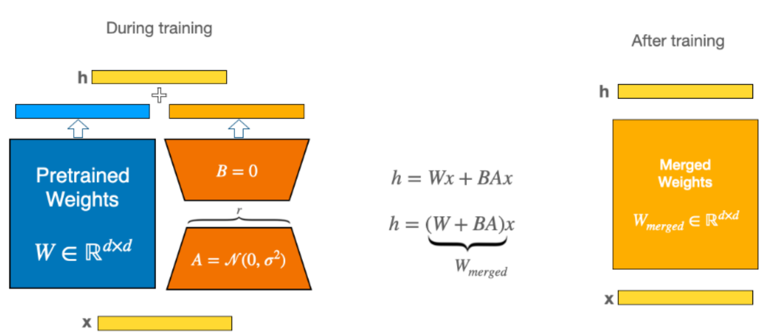
NOURAN SHABAN EL-SHORA

1. Introduction

Fine-tuning large language models is one of the most powerful ways to adapt them for specific tasks, like chatbots, summarization, or sentiment analysis. But there's a big challenge: **full fine-tuning requires huge computational resources and memory**. Some of these models have billions of parameters, which makes training on a regular GPU impossible.



This is where **LoRA (Low-Rank Adaptation)** and **QLoRA (Quantized LoRA)** come in. These methods make it possible to **fine-tune large models efficiently**, saving both **memory and time**, without losing much accuracy.



2. LoRA (Low-Rank Adaptation)

What is LoRA?
LoRA is a method that allows fine-tuning of large models by **training only a small set of additional matrices**, called **low-rank matrices**, while keeping the original model weights frozen.

How it works:

1. The original model weights are **frozen**, so they don't change.
2. A pair of **low-rank matrices** is added to each layer that we want to adapt.
3. Only these matrices are trained, which drastically reduces the number of trainable parameters.

Benefits of LoRA:

- Saves memory, because only the low-rank matrices are trained
- Faster training
- Can fine-tune **very large models** on smaller hardware

Drawbacks:

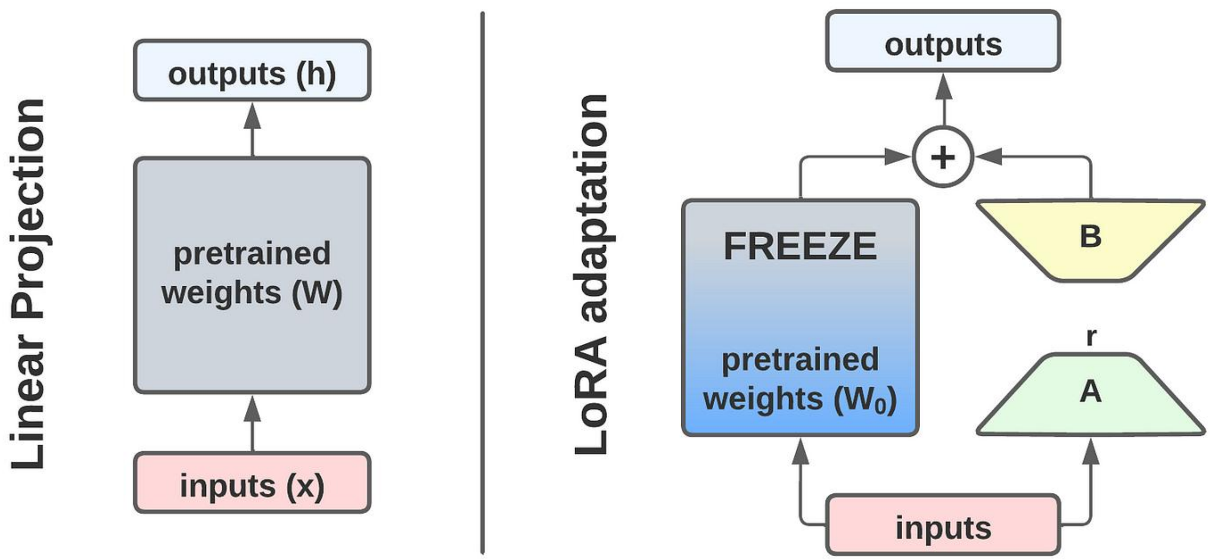
- Slightly lower accuracy compared to full fine-tuning (usually minimal)
- Limited flexibility if you want to modify the model extensively

Use Cases:

- Task-specific LLM adaptation (chatbots, summarization, classification)
- Academic research or experimentation with large models

LoRA vs Full Fine-Tuning:

Method	Trainable Parameters	Memory Usage	Training Time	Accuracy
Full Fine-Tuning	100%	High	Long	High
LoRA	1–5%	Low	Short	High



3. QLoRA (Quantized LoRA)

What is QLoRA?
QLoRA is an extension of LoRA that **adds quantization** to further reduce memory usage. Essentially, it combines **LoRA's low-rank adaptation** with **model quantization** (reducing the bit-width of model weights, e.g., 4-bit or 8-bit).

How it works:

1. The large model is **quantized**, meaning its weights are stored in lower precision.
2. Low-rank matrices from LoRA are added and trained as usual.
3. Fine-tuning can now be done on **consumer GPUs**, even for huge models.

Benefits:

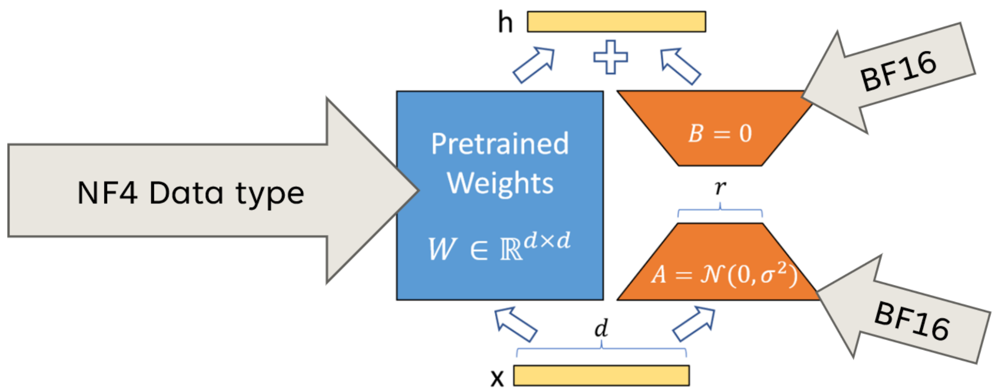
- Huge memory savings (sometimes 4–8x smaller)
- Allows fine-tuning **multi-billion parameter models** on ordinary hardware
- Minimal impact on model performance

Use Cases:

- Fine-tuning LLMs for chatbots
- Task-specific models for text summarization, translation, or question answering

- Academic or small-company setups where large GPUs are unavailable

QLoRA quantization diagram



4. Conclusion

- **LoRA** allows training only a small part of the model, making fine-tuning **efficient and lightweight**.
- **QLoRA** adds quantization on top of LoRA, enabling training of **very large models on limited hardware**.
- Both techniques are **revolutionizing how we adapt massive models** to specific tasks, making state-of-the-art NLP models more accessible.

Comparison of LoRA vs QLoRA

Feature	LoRA	QLoRA
Memory Usage	Low	Very Low
Trainable Params	1–5%	1–5% + quantized weights
Hardware Needed	Moderate	Low (consumer GPU)
Accuracy	High	High