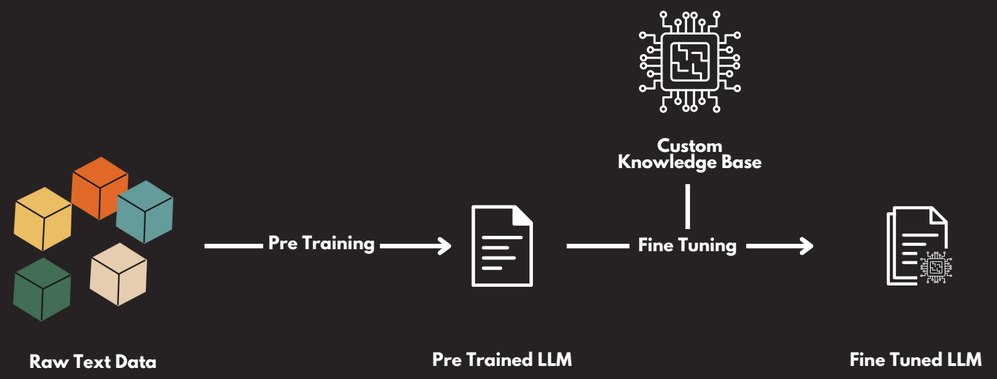
@bhavishya-pandit

# FINE TUNING TECHNIQUES



# 1. FEATURE EXTRACTION

Feature Extraction (Repurposing) is a primary approach to fine-tuning.

If you have an already pre-trained model and then with feature extraction you can use the same model for different input set.

This approach is suitable when the pre-trained model has already learned high-level features that are relevant to the new task. By keeping the early layers fixed, the model retains the knowledge of these generic features, and the later layers are adapted to the specific task.

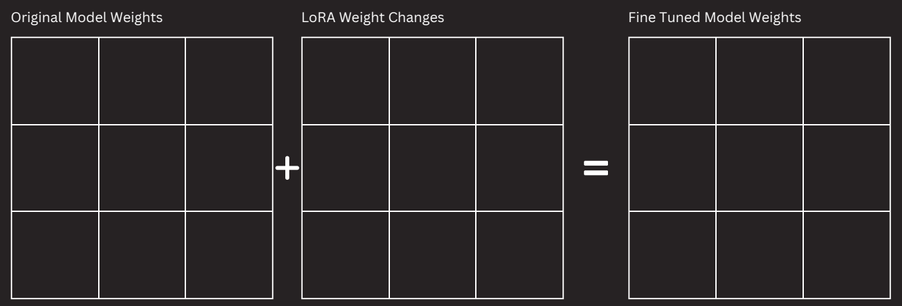
@bhavishya-pandit

# 2. LoRA(Low Rank Adaption)

LoRA helps in fine tuning a model in less cost without comprising the entire efficiency.

In tradidtional fine tuning we alter all the model weights and iterate through the params multiple times. For example, for a 13B model we have 13B weights and we iterate through it multiple times to train.

LoRA Instead of updating weights directly we track changes. These changes are tracked in two separate smaller matrices that get multiplied to form a matrix the same size as the models weight matrix



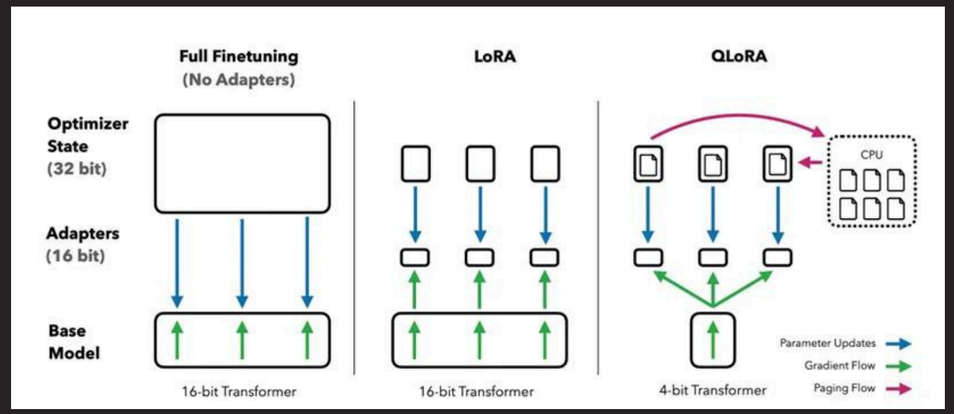

**Implementation** https://www.kaggle.com/code/krist0phersmith/lora-implementation

**@bhavishya-pandit**

# 3. QLoRA(Quantized Low Rank Adaption)

QLoRA is an updated version of LoRA. It works by quantizing the precision of the weight parameters in the pre-trained LLM to 4-bit precision.

Typically, parameters of trained models are stored in a 32-bit format, but QLoRA compresses them to a **4-bit format**. This reduces the memory footprint of the LLM, making it possible to fine-tune it on a single GPU.




**Implementation** https://www.kaggle.com/code/philculliton/fine-tuning-with-llama-2-qlora

**Adapters** **"Adapters are model weights generated during fine tuning that allow the LLM to adapt to new scenarios without changing its original parameters"**

**@bhavishya-pandit**

# 4. PROMPT TUNING

Prompt Tuning is a technique to get better results from a model without altering its core architecture.

In prompt-tuning, the best cues, or front-end prompts, are fed to your AI model to give it task-specific context. The prompts can be extra words introduced by a human, or AI-generated numbers introduced into the model's embedding layer.

Model recognizes the pattern in the input prompt and acts accordingly. This technique is very cheap as the model parameters are not changed in any way.

Unlike other techniques like **Feature Extraction, LoRA , QLoRA** Prompt Tuning works on higher levels of model rather than changing the core architecture.

**@bhavishya-pandit**

# FOLLOW FOR MORE AI/ML POSTS

@bhavishya-pandit