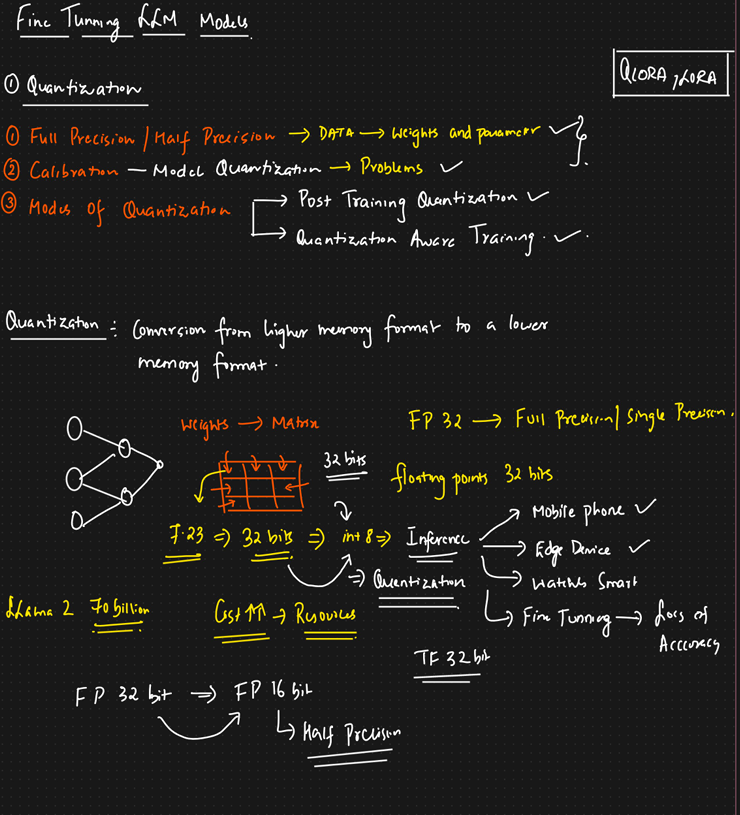
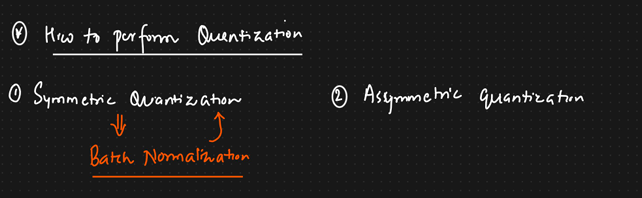
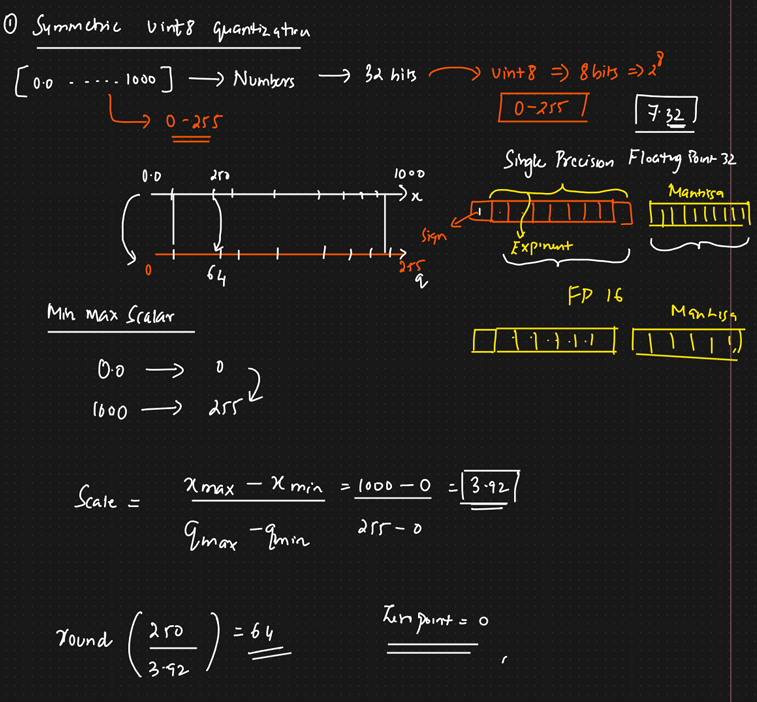
|  |  |
| --- | --- |
| GEN AI LEARNING  **Complete Generative AI Learning – New Year Challenge** | WRITTEN BY: ALOY |

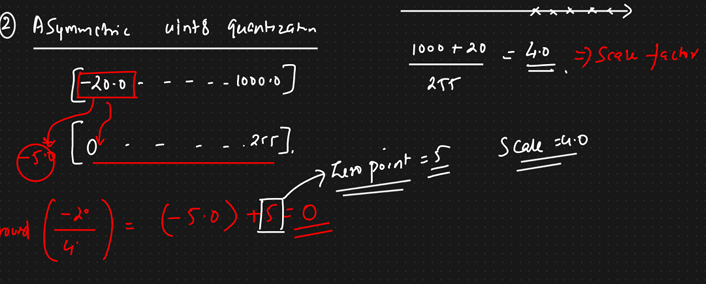
***Day – 11***

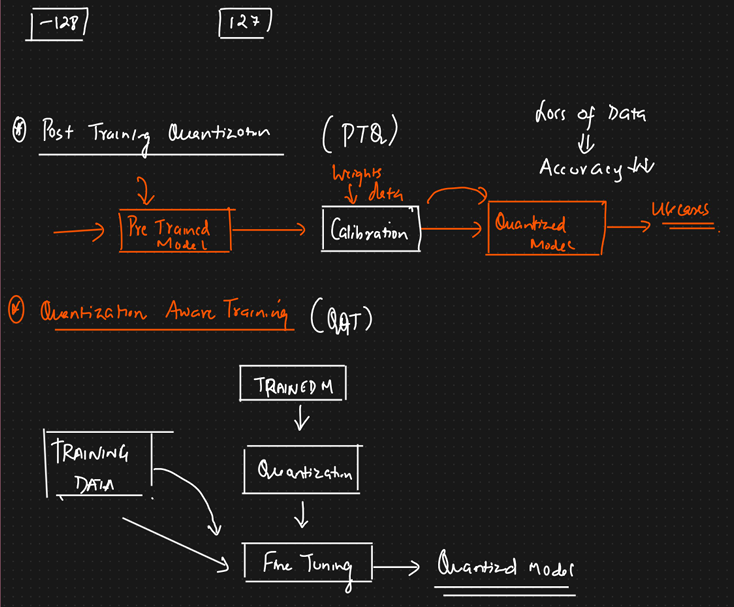
LLM Finetuning:











**Notes on LLM Quantization and Related Topics**

**1. Half & Full Precision in LLM Quantization**

Quantization is a process of reducing the precision of numerical representations to improve the efficiency of machine learning models like Large Language Models (LLMs). Two key precision formats are:

* **Half Precision (16-bit Floating Point - FP16)**:
  + FP16 uses 16 bits to represent numbers.
  + It reduces memory usage and speeds up computation while maintaining reasonable model performance.
  + Commonly used in modern GPUs.
* **Full Precision (32-bit Floating Point - FP32)**:
  + FP32 uses 32 bits to represent numbers, offering higher numerical precision.
  + It is the default format for training deep learning models, ensuring minimal loss of accuracy.

**LLM Quantization:** Transitioning an LLM from FP32 to FP16 (or lower precision) significantly reduces memory and computational requirements. Advanced quantization methods even push to INT8 or lower, with trade-offs in accuracy.

**2. Calibration Process in Quantization (Symmetric and Asymmetric)**

Calibration is essential for quantization, ensuring the model's numerical representations align with its original range and performance.

**Symmetric Quantization:**

* Both positive and negative values are scaled symmetrically around zero.
* **Key Advantage:** Simpler and computationally efficient.
* **Key Limitation:** May not be ideal for datasets with skewed distributions.

**Asymmetric Quantization:**

* Allows for a non-zero offset, improving the representation of skewed distributions.
* **Key Advantage:** Accommodates asymmetric data distributions.
* **Key Limitation:** Slightly more complex.

**3. Modes of Quantization**

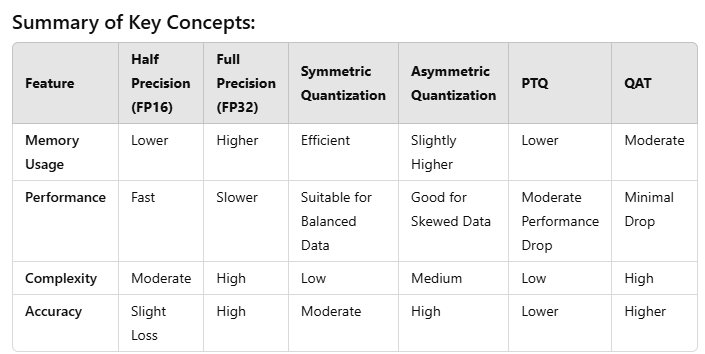
Quantization can be applied at different stages of the model lifecycle:

**1. Post-Training Quantization (PTQ):**

* Applied after the model has been fully trained.
* Calibration data is used to calculate scale and zero-point values.
* Types of PTQ:
  + **Static PTQ:** Calibration dataset is explicitly used.
  + **Dynamic PTQ:** Calibration occurs during inference.
* **Key Advantage:** Simpler and doesn’t require retraining.
* **Key Limitation:** Potential loss in model accuracy.

**2. Quantization-Aware Training (QAT):**

* Quantization is simulated during training.
* Includes quantization noise in forward passes, allowing the model to adapt.
* **Key Advantage:** Maintains higher accuracy compared to PTQ.
* **Key Limitation:** Requires additional training effort and resources.

****

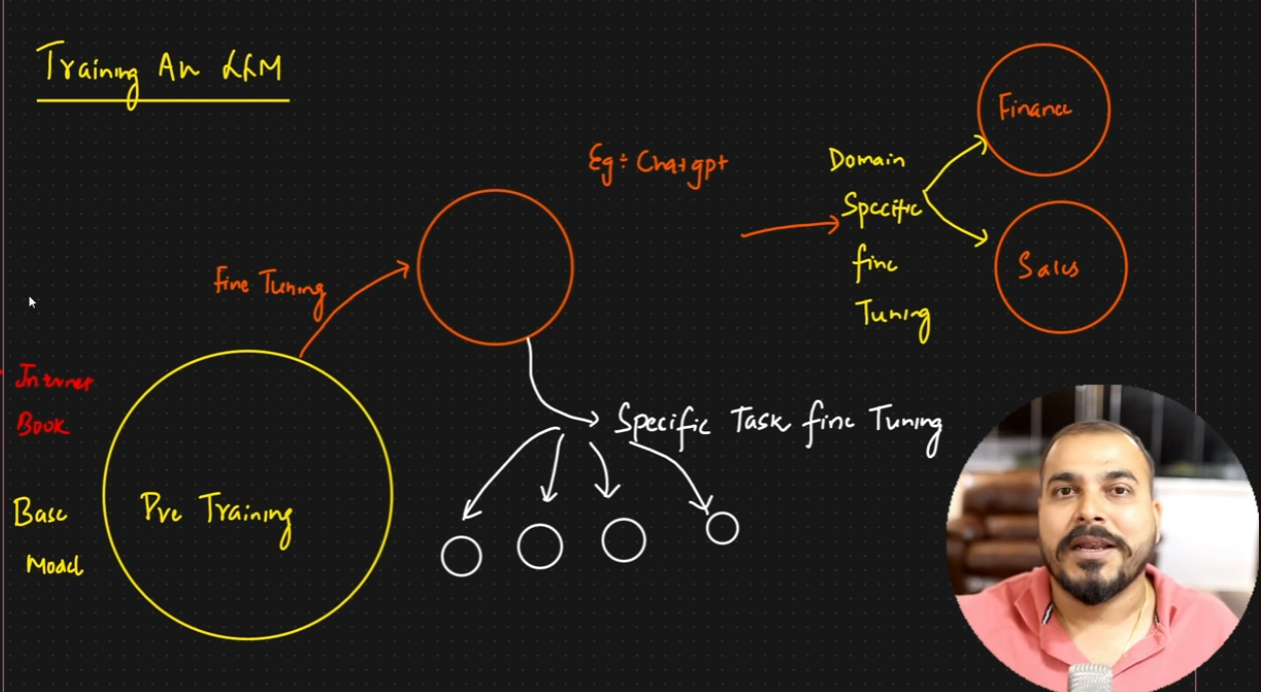
**LoRA (Low-Rank Adaptation) and QLoRA:**

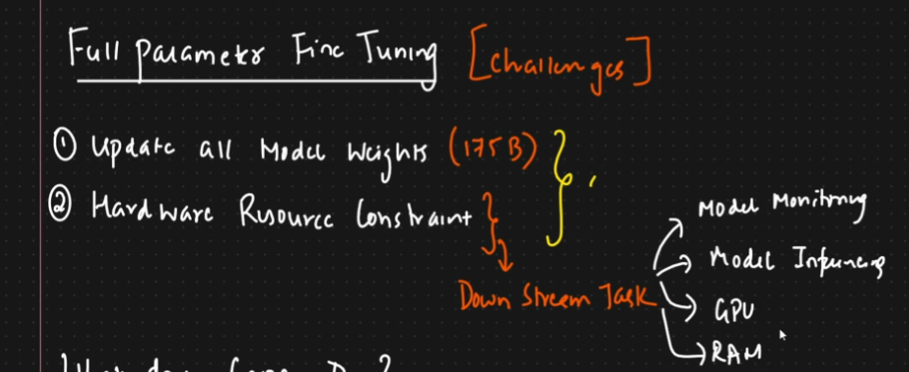
**LoRA (Low-Rank Adaptation):**

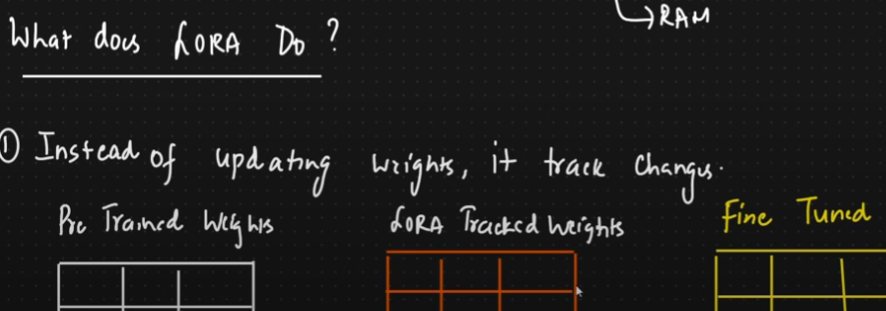
* A method to fine-tune pre-trained language models efficiently by adapting only a small number of parameters.
* Instead of updating all model weights, LoRA learns low-rank matrices that approximate the updates.
* Key benefits:
  + Reduces the computational and memory costs of fine-tuning.
  + Can be applied to very large models, even with limited hardware.
  + Ensures that the base model remains unchanged, making it reusable across tasks.

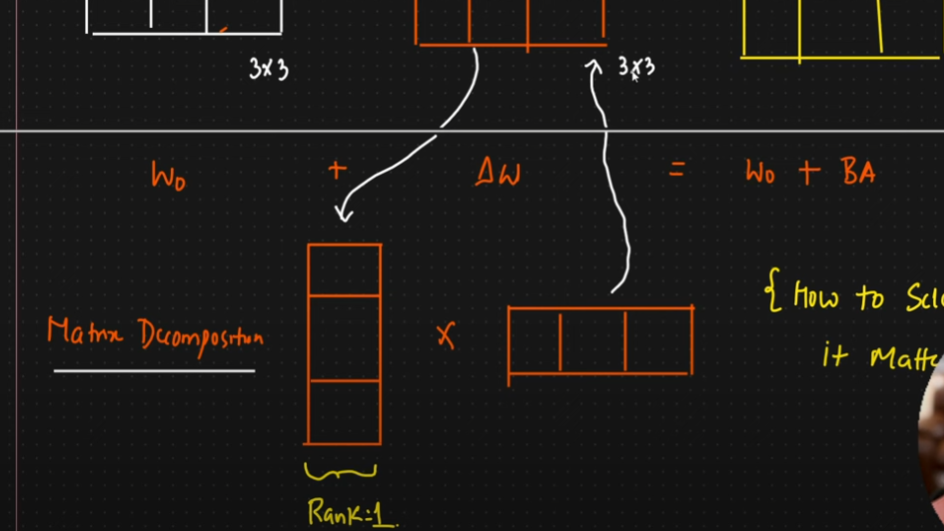
**QLoRA (Quantized LoRA):**

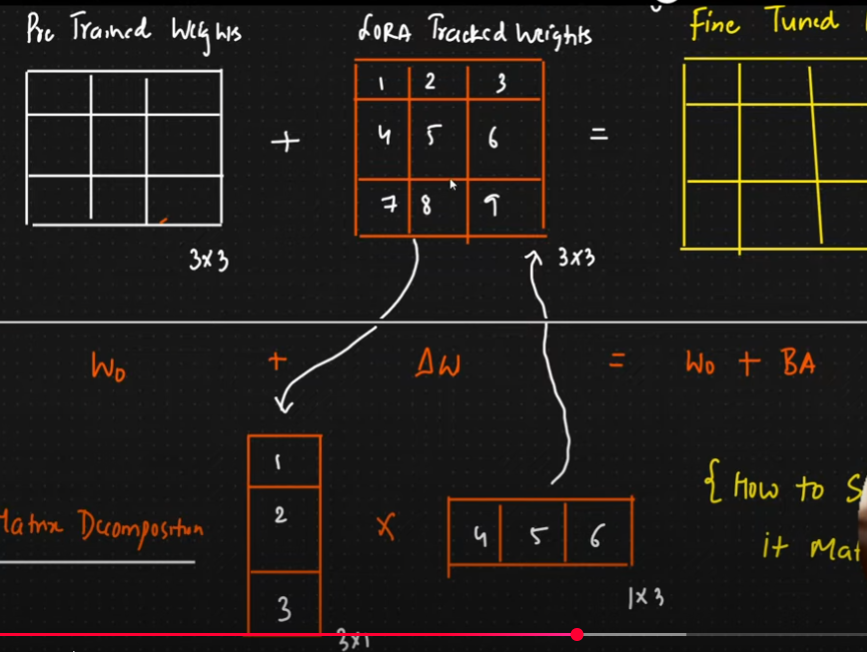
* Combines quantization and LoRA to further optimize large-scale model adaptation.
* Key features:
  + Quantizes the model weights (often to 4-bit precision) to reduce memory footprint.
  + Fine-tunes only the low-rank matrices, leveraging the benefits of both quantization and LoRA.
  + Allows for efficient and scalable fine-tuning of extremely large language models on consumer-grade hardware.
* Benefits:
  + Drastically reduces memory usage without sacrificing significant accuracy.
  + Enables adaptation of models exceeding billions of parameters on modest hardware setups.

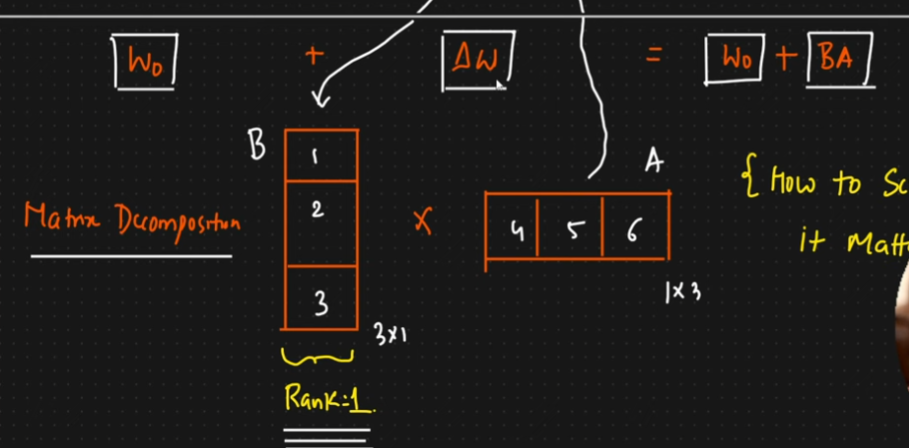




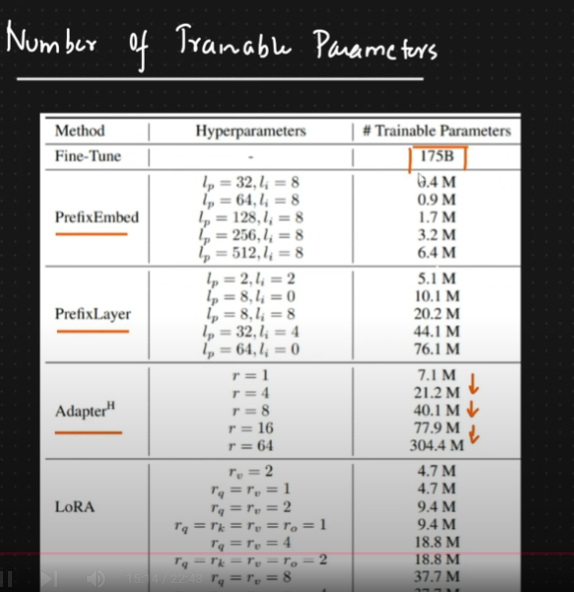


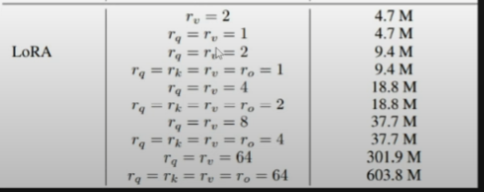


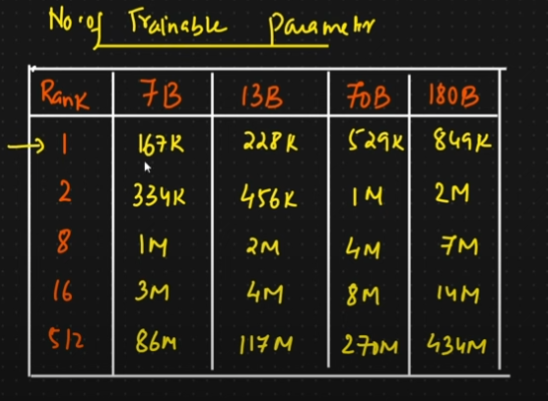




Smaller set of two matrix is decomposed based on the Rank of the matrix.







A higher rank must be used if the model wants to learn complex things.

