⚙️ What is Quantization?

Quantization is the process of reducing the precision of numbers used to represent a model's parameters (weights, activations) from high precision (like 32-bit floating point) to lower precision (like 16-bit, 8-bit, or even 4-bit integers).

- Goal: Shrink model size, reduce memory bandwidth, and speed up inference.
- Trade-off: Slight loss in accuracy vs. huge gains in efficiency.

🔑 Why Quantization Matters

- Memory savings: A 175B parameter model in FP32 needs ~700 GB; INT8 reduces it to ~175 GB.
- Speed: Lower precision arithmetic runs faster on CPUs/GPUs/TPUs.
- Deployment: Enables running LLMs on edge devices or resource-constrained environments.
- Cost efficiency: Less compute → lower cloud costs.

🛠️ Types of Quantization

1. Post-Training Quantization (PTQ)

- Apply quantization *after* training.
- Simple, fast, but may cause accuracy drop.
- Example: Convert FP32 → INT8 weights.

2. Quantization-Aware Training (QAT)

- Simulate quantization during training.
- Model learns to adapt to lower precision.
- More accurate than PTQ, but requires retraining.

3. Dynamic Quantization

- Weights stored in low precision, activations quantized *on the fly*.
- Lightweight, good for RNNs and transformers.

4. Static Quantization

- Both weights and activations quantized ahead of time.
- Requires calibration dataset to determine scaling factors.
- More efficient than dynamic quantization.

5. Mixed Precision

- Use different precisions for different parts of the model.
- Example: FP16 for attention layers, INT8 for feed-forward layers.
- Balances accuracy and efficiency.

📊 Quantization Levels

Precision	Storage per weight	Typical Use Case
FP32	32 bits	Training, high accuracy
FP16/BF16	16 bits	Training & inference (GPUs/TPUs)
INT8	8 bits	Standard deployment, big speedup
INT4	4 bits	Extreme compression, edge devices
INT2/1	2–1 bits	Research stage, aggressive compression

🚀 Advanced Techniques

- Per-channel quantization: Different scales per weight channel → better accuracy.
- Quantization + Pruning: Combine with weight pruning for maximum compression.
- Quantization + Distillation: Train a smaller student model with quantized teacher outputs.
- Hardware-aware quantization: Tailor precision to target hardware (NVIDIA TensorRT, Intel MKL-DNN, ARM CPUs).

🏢 Enterprise AI Context (Your Use Case)

- LLMs in production: INT8 quantization is the sweet spot for balancing accuracy and efficiency.
- RAG pipelines: Quantized embeddings speed up vector search in Pinecone/FAISS.
- Agent frameworks: Mixed precision ensures compliance-critical tasks remain accurate while background tasks run faster.
- Salesforce Agentforce prep: Knowing quantization shows you understand *deployment efficiency*, a key enterprise concern.

✅ In short: Quantization is about trading a little accuracy for massive efficiency gains. It's the reason LLMs can move from research labs into real-world enterprise systems.