Report

Quantization:

· Quantization is a technique used in machine learning to reduce the
memory footprint of a model by reducing the precision of its weights.. Un-quantized weights in a model come as 32-bit floating point numbers.
This means for each parameter/weight in the model it takes up 4 bytes of
Memory.

To load in a 7B parameter model, we will need 28GB of space. For a 175B parameter model you would need 700 GB of GPU RAM!

Imagine a model's weights are like super-detailed measurements (like 32.487213), but you don't always need that level of detail. So, quantization rounds or maps those numbers to simpler forms (like 32 or 32.5) that are smaller and faster to work with.

It reduces the number of bits needed to represent a model parameter.
We have access to the following precisions
to quantize to:

Format	size	storage	Speed	Accuracy
FP32 (float)	32-bit	Large	Slow	High
FP16 (half)	16-bit	Medium	Fast	Good
INT8 (int)	8-bit	Small	Fastest	Minor loss

Types of Quantization :

1. **Post-Training Quantization (PTQ):**
    - Done after training the model.
    - Fast and easy, but sometimes loses accuracy.
2. **Quantization-Aware Training (QAT):**
    - Simulates quantization during training.
    - Slower to train but keeps better accuracy.

Uses:

- Fast inference
- Resource restricted environments

Knowledge Distillation:

Knowledge distillation or model distillation is the process of transferring knowledge from a large model to a smaller one. It's just like a teacher teaching a student.

Knowledge distillation transfers knowledge from a large model to a smaller one without loss of validity. As smaller models are less expensive to evaluate, they can be deployed on less powerful hardware (such as a mobile device).

How it Works

Instead of training the student model only on correct answers (hard labels), we let it learn from the teacher’s predictions (soft labels) — which give much more information.
As the distilled model has less learning capacity than the larger model , if we train both on the same dataset the distilled one might not be able to grasp the knowledge comparatively to the larger one.

What are Soft Labels

In machine learning classification, the model's final layer is usually a softmax function. It turns raw scores (called logits) into probabilities for each class.

These probabilities reflect how confident the model is about each class.Soft labels give more information than hard labels.

When we apply **temperature (T)** to the softmax, we control **how soft or sharp** the probabilities are.

Formula:

$$y_i(\mathbf{x}|t) = \frac{e^{\frac{z_i(\mathbf{x})}{t}}}{\sum_j e^{\frac{z_j(\mathbf{x})}{t}}}$$

- **T = 1**: normal softmax
- **T > 1**: softer outputs (more uncertainty, more knowledge transfer)
- **T < 1**: sharper outputs (more confident, but less informative)

Steps:

1. Feed the same input to both models (teacher + student)
2. Get the larger model and distilled soft predictions
3. Compute two losses: one comparing distilled to original (soft), and one comparing distilled to label (hard)

4. Use the following loss Functions for training :

   If ground labels not available :

   Loss = CrossEntropy(output of distilled , output of larger model)

   If ground labels are available:

   Loss = CrossEntropy(output of distilled , output of larger model) +
   CrossEntropy(y true, y predicted)

5. Now train the student using the loss