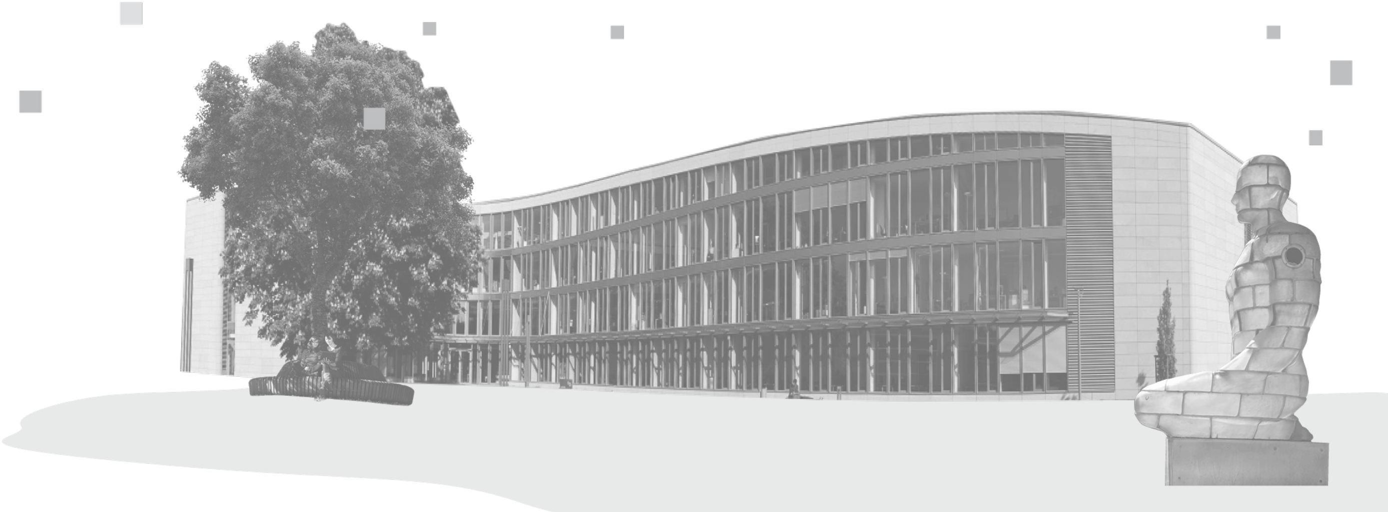
Building Local LLMs with Low-bit Quantization

Reducing Model Size Without Sacrificing Performance

PD Dr. Haojin Yang,

Multimedia and Machine Learning Group,

Hasso Plattner Institute

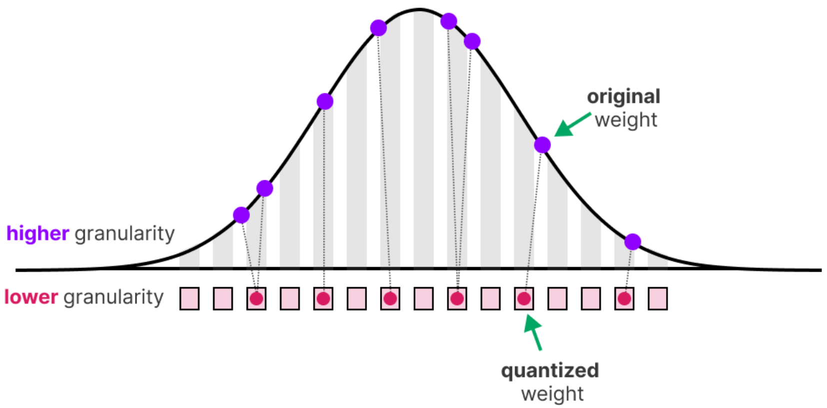


Introduction to Low-Bit Quantization

- Low-bit quantization reduces model parameter precision to make LLMs smaller
- Converts 32-bit floating-point numbers to 8-bit, 4-bit, or even lower formats
- Enables running powerful LLMs on consumer hardware with limited resources
- Significantly reduces memory requirements and computational demands
- Trades minimal accuracy loss for substantial efficiency gains
- Essential technique for local deployment of AI models

How Quantization Works

- Quantization reduces precision of model weights from 32-bit floating point to lower bit formats (8-bit, 4-bit, etc.)
- Process maps continuous values to a smaller, discrete set of values using scaling factors (α, β)
- Post-training quantization (PTQ) applies quantization after model is fully trained
- Quantization-aware training (QAT) incorporates quantization during the training process
- **Weight-only quantization** focuses on model parameters while preserving activation precision
- Different methods balance trade-offs between model size, inference speed, and accuracy



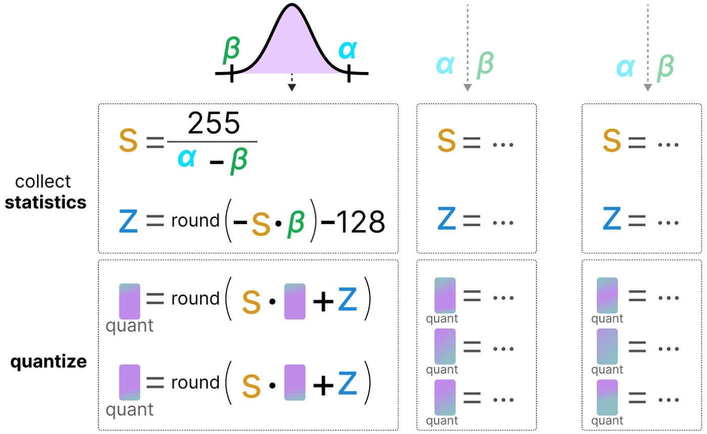


Image source: https://www.maartengrootendorst.com/blog/quantization/

Quantization Techniques Compared

- 8-bit quantization: ~2x smaller models with minimal performance loss
- 4-bit quantization: ~8x smaller models with acceptable quality tradeoffs
- 2-bit/3-bit quantization: Experimental with significant quality degradation
- GPTQ vs AWQ: Popular algorithms with different optimization approaches
- INT4 vs NF4: Different number formats with varying performance characteristics
- Mixed-precision: Using different bit-widths for different model components
  - Layer-level mixed bit-width or channel-level mixed bit-width

Model size estimation of a 70B Model with 70 billion parameters:

64-bit = 64/8 x 70B ≈ 560 GB
32-bit = 32/8 x 70B ≈ 280 GB
16-bit = 16/8 x 70B ≈ 140 GB
8-bit = 8/8 x 70B ≈ 70 GB
4-bit = 4/8 x 70B ≈ 35 GB
2-bit = 2/8 x 70B ≈ 17,7 GB

Symmetric Quantization

- Maps floating-point values to a symmetric range around zero
- Zero in floating-point space maps exactly to zero in quantized space
- Uses a linear mapping centered around zero
- Scale factor: $s = \frac{(2^{b-1} - 1)}{\alpha}$ (where α is the maximum absolute value)
- Quantization: $q(x) = round(s \times x)$
- Dequantization: $\hat{x} = \frac{q(x)}{s}$

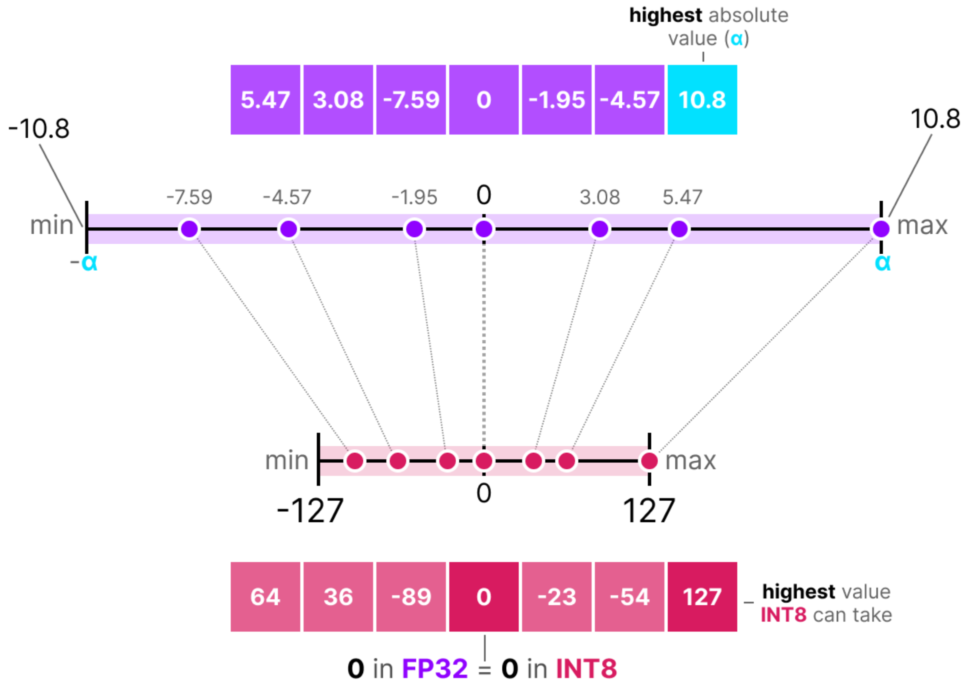


Image source: https://www.maartengrootendorst.com/blog/quantization/

Symmetric Quantization Example

- Example with original values: [8.5, 3.08, 3.02, -7.59, 10.8]
- Highest absolute value: 10.8
- Scale factor calculation: **s** = 127 / 10.8 = 11.76
- Quantized values: [100, 36, 36, -89, 127]
- **Advantages:**
  - Simple implementation
  - Preserves zero exactly
  - No need to store zero-point offset
- **Limitations:**
  - Less efficient for data not centered around zero
  - Similar values may map to same quantized value (loss of precision)

Asymmetric Quantization

- Not symmetric around zero
- Maps minimum (β) and maximum (α) values to full quantized range
- Uses zero-point quantization with shifted zero position
- Scale factor: $s = \frac{(2^b - 1)}{(\alpha - \beta)}$
- Zero point: $z = round(-\beta \times s)$
- Quantization: $q = round(x \times s + z)$
- Dequantization: $x = \frac{(q - z)}{s}$

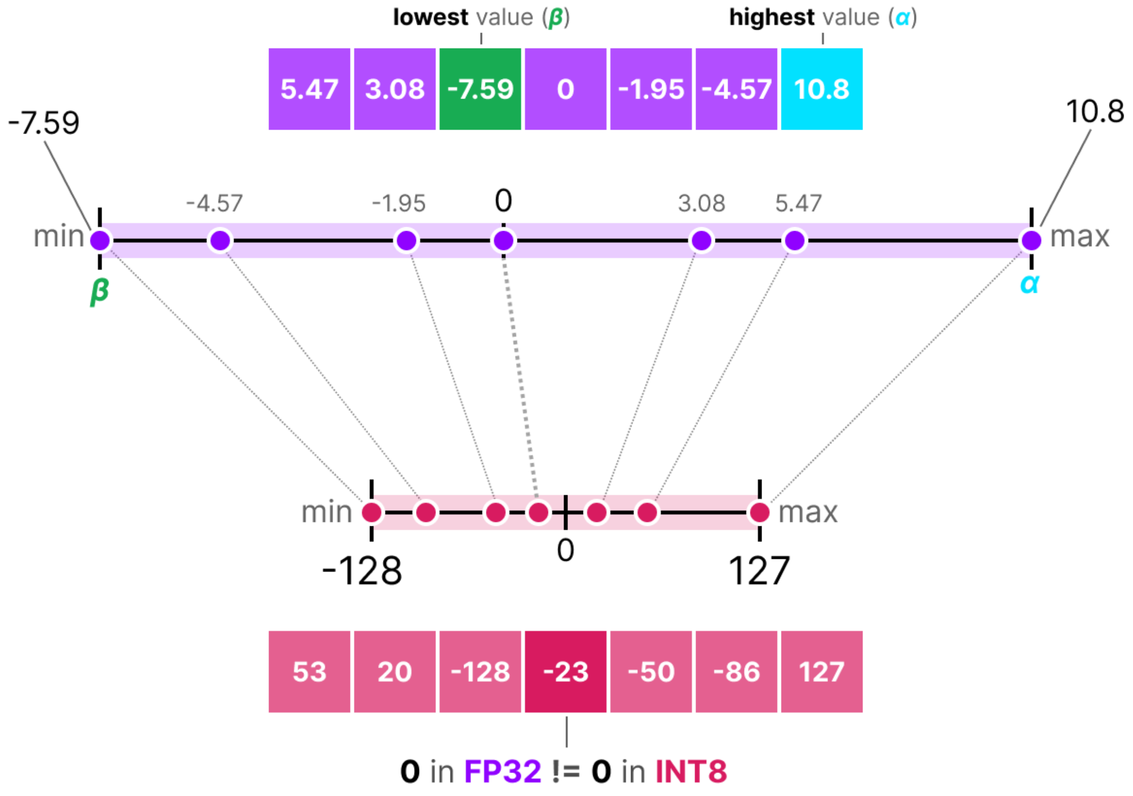


Image source: https://www.maartengrootendorst.com/blog/quantization/

Comparing Symmetric vs. Asymmetric Quantization

- Symmetric: Centered around zero, simpler calculations
- Asymmetric: Shifted zero position, better range utilization
- When to use Symmetric:
  - Data naturally centered around zero (like weights in neural networks)
  - When computational simplicity is important
  - When preserving zero exactly is critical
- When to use Asymmetric:
  - Data with skewed distribution (not centered around zero)
  - When maximizing representation accuracy is priority
  - For activations in neural networks (often positive-only)