Quantization explained: GGUF, AWQ and running big models on small hardware

26 October 2026 · 6 min read · blogs.iictp.org/llm-quantization-explained-gguf-awq

Quantization stores a model's weights using fewer bits, so the model takes less memory and often runs faster. The trade-off is some loss of accuracy, which grows as the precision falls.

TL;DR The short version

- Quantization stores model weights in fewer bits, so the model uses less memory and often runs faster, at some cost in accuracy.
- GGUF is the single-file format used by llama.cpp; AWQ protects the most important weights for GPU inference.
- Pick the highest precision that fits with room for the context cache, and test maths, code and rare languages first, as they suffer earliest.

Read online, listen to it, or change the text size: https://blogs.iictp.org/llm-quantization-explained-gguf-awq

The idea

A full-precision weight might use 16 bits. Quantizing to 8 or 4 bits maps each weight to a coarser set of values. Memory falls roughly in proportion: a model at 4 bits needs about a quarter of the weight memory of the same model at 16 bits.

Formats you will meet

- **GGUF:** a single-file format used by llama.cpp and tools built on it, with many quantization levels.
- **AWQ:** a method that protects the weights that matter most to the output, designed for GPU inference.
- **GPTQ:** another post-training method, also aimed at GPU inference.

Choosing a level

1. Start from the memory you have.
2. Pick the highest precision that fits with room for context.
3. Test on your own examples; compare a lower level against it.
4. Stop lowering when quality drops in ways that matter.

Pitfalls

- Quantization loss shows up unevenly: maths, code and rare languages are often hit first.
- Leave memory for the context cache, which grows with the conversation.
- Re-test after changing quantization, like any other change.

Related credential: Local LLM Practitioner. Explore the exam facts and syllabus.

Sources and further reading

1. Lin et al. “AWQ: Activation-aware Weight Quantization.” arXiv:2306.00978.
2. llama.cpp project. github.com/ggml-org/llama.cpp.

Published by IICTP, an independent technical credential body, on 26 October 2026. The current version of this post is at https://blogs.iictp.org/llm-quantization-explained-gguf-awq. More guides: blogs.iictp.org.