Paper: https://arxiv.org/pdf/2210.17323

This week, one of the readings our mentors assigned to us dealt with GPTQ—a quantization technique to compress massive LLMs, like the GPT-175B model.

One interesting thing from the reading was the ability of the technique to reduce precision from 16-bit to 3- and 4-bit, while preserving model accuracy. Personally, I have worked with Llama models that are much smaller than this GPT model and they required multiple GPUs for inference. The paper claims to be able to run the compressed version of this GPT model on a single GPU, which is very impressive and has a lot of practical implications in terms of computation cost.

One section of the paper I had difficulty understanding fully was the description of the OBQ (Optimal Brain Quantization) technique, which is the algorithm upon which this paper builds its novel technique. The specific part I struggled to understand is how exactly the weights are iteratively quantized using the two equations mentioned in the paper.

Based on this reading and some external knowledge-gathering I did on the topic of quantization, there seem to be two methods for quantization: PTQ and QAT. QAT is essentially a technique used to quantize weights during training. PTQ is a method used to quantize weights after training. In this specific application area of LLMs, the authors chose to develop a PTQ algorithm because it is very resource-intensive to re-train an LLM with 175B parameters. The authors themselves state that to re-train a model of this size, it would take “tens-to-hundreds of GPU years.” Hence, the choice of using a PTQ algorithm for this task seems to be a resource-efficient decision.