Week 6 Questions

LoRA: Low-Rank Adaptation of Large Language Models

Brief Summary: The LoRA method trains low-rank matrices on top of the frozen parameters of a pretrained model (frozen meaning they stay the same throughout training). This method takes less memory than full fine-tuning and in certain cases may have better task performance.

1. How do they initialize the matrices for LoRA?
2. Why does LoRA not add additional compute at inference time?
3. Why does LoRA reduce the memory needed for training?
4. LoRA involves learning low-rank matrices on top of pretrained model parameters. Do you think it would work to just only learn low-rank matrices for pretraining? In other words, could train a network with only low-rank matrices and still have good performance? (Bonus: modify the minGPT implementation to actually try this)

Methods and tools for efficient training

Brief Summary: Don’t focus too much on the details of each technique discussed. Try to have a general idea of things you may want to try when training a model. The two most important techniques discussed are gradient accumulation and gradient checkpointing. Instead of computing gradients for every example in a batch simultaneously, in gradient accumulation the gradients for smaller batches are computed and these are summed to get the final gradients for a full batch. In gradient checkpointing, memory is reduced by not storing as many intermediate activations that are used for the backward pass. However, these need to be recomputed during the backward pass, which takes some more compute time.