Critique on Zero

Summary

Existing methods that exploit data and model parallelisms are inadequate when it comes to training large models into devices with limited memory. Zero aims to optimize this memory use through a novel method of splitting activations and input data across the devices involved.
The solution focus on three main factors that make up residual memory consumption: 1) Activation Memory is reduced by removing memory redundancies through partitioning the activation checkpoints across GPUs, and using primitives such as AllGather to reconstruct them on demand. 2) It uses constant size buffers to avoid temp buffers from blowing up 3) Memory Fragmentation is reduced through smart allocation of tensors according to their lifetimes: forward propagation activations and backward param grads are long-lived while the recomputed forward activations and the bacward activation gradients are shortlived. It is able to perform on-the-fly memory defragmentation through pre-allocation contiguous memory buffers

Strengths

- Able to run large models across devices with limited memory footprint
- No model refactoring is necessary, making it easy to use as a standard DP

Weaknesses

- A lot of communication between devices
- Computations are repeated, leading to energy consumptions

Possible Improvements

- Quantization of weights can be introduced to better improve bandwidth