MegaBlocks: Efficient Sparse Training with Mixture-of-Experts

Main Idea(s): MegaBlocks aims to improve the challenges of load imbalance and token dropping in MoE architecture using block sparse matrices. The idea is to present a router that dynamically handles the token allocation to experts. While in a regular MoE architecture each expert is assigned to a single GPU in a fixed allocation system (each expert gets the same amount of compute), having to drop tokens in the case of overflow to a specific expert/GPU, while at the same time padding tokens to compensate for idle computational resources in experts which were not assigned enough tokens in a batch, MegaBlocks makes this allocation dynamically from the start, so the computational resources assigned to an expert is variable, being adjusted on a per-batch basis based on the tokens assigned to the expert on that specific batch.

OBS: Tutel, a previous work, used a similar strategy, by implementing a dynamic CF (capacity factor) for each expert, but this leads to computational inefficiencies.

MegaBlocks is possible by making use of block-sparse matrix multiplication as opposed to batched matrix multiplication. This approach maps efficiently to hardware accelerators and allows for variable expert size and allocation.

MegaBlocks leads to training speedups, which is logical since it makes optimum use of computational resources at each update.

My takeaways:

- MegaBlocks is an approach for maximizing computing efficiency when training MoE models. It dynamically adjusts how much compute to be given to each expert at every batch, preventing token dropping and idle resources. Although this is interesting, per the experiments of ST-MoE, this seems to only be useful at pre-training, as load balancing does not seem to affect fine-tuning much.