Announcements

• Extra credit assignment 7 is posted: due on December 13 11:59 pm

• Final exam will be posted on gradescope at December 15 12:01 am and will be due on December 16 11:59 pm local time
  • No late submissions allowed

• Course evaluation: https://www.courseevalum.umd.edu
Contact me:

- CMSC416: If you are interested in HPC research
- CMSC818X: If you are interested in collaborating
- If you are an undergrad interested in participating in International Student Cluster Competitions
Deep neural networks

- Neural networks can be used to model complex functions
- Several layers that process “batches” of the input data
Other definitions

- Learning/training: task of selecting weights that lead to an accurate function
- Loss: a scalar proxy that when minimized leads to higher accuracy
- Gradient descent: process of updating the weights using gradients (derivates) of the loss weighted by a learning rate
- Mini-batch: Small subsets of the dataset processed iteratively
- Epoch: One pass over all the mini-batches
Parallel/distributed training

- Many opportunities for exploiting parallelism
- Iterative process of training (epochs)
- Many iterations per epoch (mini-batches)
- Many layers in DNNs
Parallel/distributed training

- Many opportunities for exploiting parallelism
- Iterative process of training (epochs)
- Many iterations per epoch (mini-batches)
- Many layers in DNNs
Parallel/distributed training

- Many opportunities for exploiting parallelism
- Iterative process of training (epochs)
- Many iterations per epoch (mini-batches)
- Many layers in DNNs
Data parallelism

- Divide training data among workers (GPUs)
- Each worker has a full copy of the entire NN and processes different mini-batches
- All reduce operation to synchronize gradients
Data parallelism

- Divide training data among workers (GPUs)
- Each worker has a full copy of the entire NN and processes different mini-batches
- All reduce operation to synchronize gradients

![Data Parallelism Diagram](image-url)
Intra-layer parallelism

- Enables training neural networks that would not fit on a single GPU
- Distribute the work within a layer between multiple processes/GPUs
Inter-layer parallelism

- Distribute entire layers to different processes/GPUs
- Map contiguous subsets of layers
- Point-to-point communication (activations and gradients) between processes/GPUs managing different layers
- Use a pipeline of mini-batches to enable concurrent execution
Inter-layer parallelism

- Distribute entire layers to different processes/GPUs
- Map contiguous subsets of layers
- Point-to-point communication (activations and gradients) between processes/GPUs managing different layers
- Use a pipeline of mini-batches to enable concurrent execution
Inter-layer parallelism

- Distribute entire layers to different processes/GPUs
- Map contiguous subsets of layers
- Point-to-point communication (activations and gradients) between processes/GPUs managing different layers
- Use a pipeline of mini-batches to enable concurrent execution
Hybrid parallelism

- Using two or more approaches together in the same parallel framework
- 3D parallelism: use all three
- Popular serial frameworks: pytorch, tensorflow
- Popular parallel frameworks: DDP, MeshTensorFlow, Megatron-LM, ZeRO