Parallelization;
Large Scale Deep Learning

Machine Learning for Big Data
CSE547/STAT548, University of Washington
Sham Kakade

©Sham Kakade 2017

The limitation of GPU

- Speed-up is small when the model doesn’t fit in the GPU memory

- Model parallelization: for dealing with large models.
Model Parallelism

Training Data

Machine

Model Parallelism: AlexNet

AlexNet architecture (May look weird because there are two different "streams". This is because the training process was so computationally expensive that they had to split the training onto 2 GPUs)
Can just toss all the ingredients in…
• Asynchronous updating…
• Model parallellization/Data Parallelization
• Mini-batching…

• Hope for the best??

• Two methods: DistBelief
  – (1) Downpour SGD
  – (2) Sandbalster L-BFGS

Model Parallelism

DistBelief enables model parallelism
(1) Across machines via message passing (blue box)
(2) Within a machine via multithreading (orange box)
What’s next?

The setup is designed for larger minibatches (over 100s of samples) of data at a time.

How can we add another dimension of parallelism, and have multiple model instances train on data in parallel?

Data Parallelism

Parameter server

Downpour SGD  Sandblaster L-BFGS
Downpour SGD

• A variant of asynchronous stochastic gradient descent

• Divide the training data into a number of subsets and run a copy of the model on each of these subset

• Update the derivatives through a centralized parameter server

• Two asynchronous aspects: model replicas run independently and parameter shards run independently

Acknowledgments

• Some slides modified from:
  https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwiyr5eVrl7AhUT5GMKHfOOCBYQFggknMAA&url=http%3A%2F%2Fweb.cs.ucla.edu%2Fclasses%2Fspring16%2Fcs239%2FCS239%2520Lecture%252015.pptx&usg=AFQjCNH8OCCAB5rig-Btq1fUvslEVz5uDg&sig2=LUAxGWzQ7ljqi3IlbamMA