Parallel and Distributed Deep Learning

Vishakh Hegde (vishakh) and Sheema Usmani (sheema)

ICME, Stanford University

1st June 2016
Background

- Build a neural network to classify images.
- Optimize parameters of the model to get a good classification rate.
- Use SGD to learn these parameters.
Problem

- Training on CPU takes a lot of time (order of days for big models)
- Solution: Use optimized GPU libraries for subroutine calls (training takes order of hours).
Empirical analysis on speed-up
Visualization

Ratio of CPU to GPU time v/s batchsize

Ratio

Batch size
Visualization

Computational time for matrix vector multiplication

Time (ms)

Matrix size
Visualization

Ratio of CPU to GPU computation time

Ratio

Matrix size
Can we do better?

- Multi-threading (embarrassingly parallel)

- Distributed learning
  - Model Parallelism
  - Data Parallelism
Data Parallelism

- Data stored across multiple machines.
- Parameters stored on the driver machine.

\[ w' = w - \eta \Delta w \]
Data Parallelism - Parameter update

- Synchronous update:
  - Parallel SGD
  - Alternating Direction Method of Multipliers

- Asynchronous update:
  - Downpour SGD
  - Dogwild (Distributed Hogwild!)

- Analysis in the report