WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

Analyzing Model Averaging for Data Parallelism in Distributed Learning

Ananth Sridhar, Ashwin Varadarajan

Overview

- Two approaches to distributed learning
  + Model Parallelism
    Split model across machines but use a single training iteration)
  + Data Parallelism
    Split training data across machines and use concurrent training iterations, followed by combining disparate models)

- What are the effects of different parameters of the training process on the model averaged performance?
  + Model Averaging Frequency
  + Maintaining Common Examples in Distributed Machines
  + Weight Initialization
  + Preferential Sampling Scheme
  + Weighted Averaging of Models

Setup

- Empirical study on the MNIST Dataset
  + (4-10) distributed machines simulated
  + Three cases evaluated: Convex, Strongly Convex and NonConvex
  + Convex: Perceptron with Softmax
  + Strongly Convex: Perceptron with Softmax and L2-regularization
  + NonConvex: Two layer ConvNet followed Dense layer (ReLUs) and Softmax layer.

Model Averaging Frequency

- For Convex and Strongly Convex losses, the averaging frequency was varied from every 40 iterations to every 5000 iterations for a total of 10000 iterations.
- The effect on the resultant iteration complexity appears to offer an nearly constant accuracy improvement, around 7%, during the first 2000 iterations (accuracy change from lowest [0.1] to almost maximum [0.8]
- For NonConvex losses, the trend is less transparent

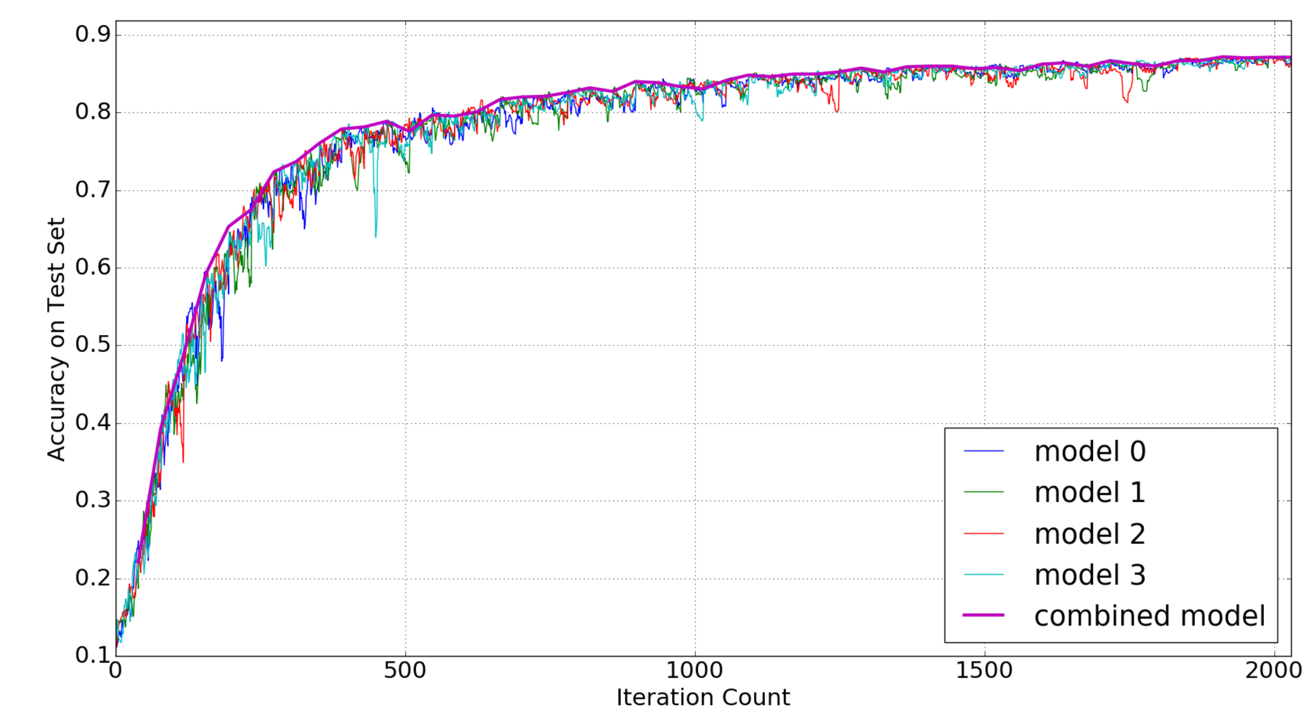

Figure 1. Effect of Model Averaging on Iteration Complexity with Convex loss function

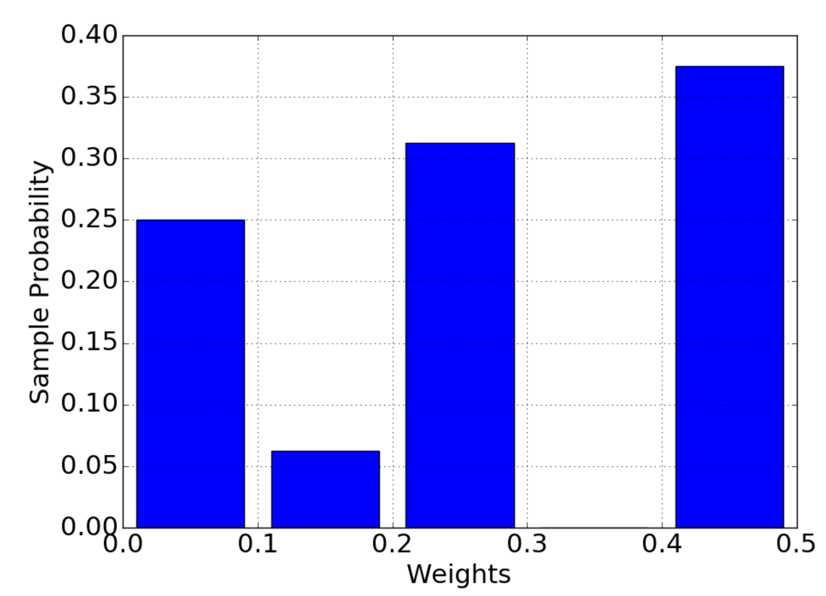

Figure 3. Distribution of best weights for model averaging with Strongly Convex loss function

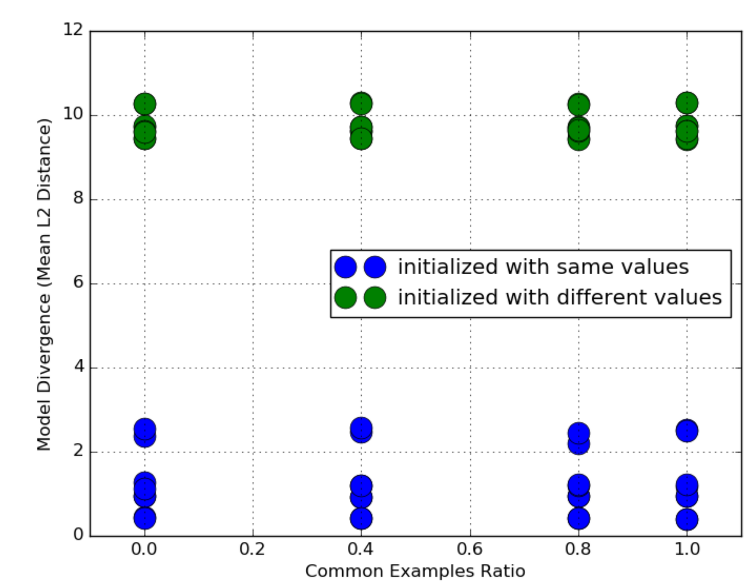

Figure 2. Effect of initialization and common examples on model divergence with NonConvex loss function

Weight Initialization

- For Convex and Strongly Convex losses, the divergence is self-contained by nature, as expected
- For NonConvex losses, maintaining common examples across machines appears to be ineffective in directing the examples towards the same minimum.
- Consequently, the models needed to be initialized with the same values to limit model divergence and to improve the effect of model averaging

Weighted Model Averaging

- For Convex and Strongly Convex losses, the best weights for averaging were numerically computed.
- The distribution is far from simple averaging. However, the improvement in accuracy from weighted averaging was only around 0.5%.
- For NonConvex losses, the improvements are drastic. In most cases, the best weights tend to favor the model with the highest accuracy. For example, a common observation was [0, 0, 0.9, 0.1] for a 4 machine case

Conclusions

- For Convex and Strongly Convex loss functions, model averaging frequency does not appear to have any impact on the iteration complexity
- For NonConvex losses, model initialization appears to be the most important factor that helps with model averaging.
- Maintaining common examples across machines does not appear to have any discernible impact