EXPLORING BIASED SAMPLING DISTRIBUTIONS FOR DATA PARALLEL NETWORKS

Brian Sharp
School of Computing
University of North Florida
Jacksonville, Fl 32256
briansharpdevelopment@gmail.com

March 19, 2020

ABSTRACT

The use of data parallel training has been an effective tool for reducing training time for deep neural networks. Conventional methods use a uniform sampling distribution when preparing training batches for workers with a weighted average reduction method for aggregating parameter gradients. In this paper, we explore the use of non-uniform distributions when distributing training data. Additionally, we explore the use of non-conventional reduction methods when aggregating gradients among workers. We analyze the resulting training dynamics for both the output predictions and the total network structure. Output predictions are monitored for the accuracy, loss, type I, and type II errors among class labels. Network structure is montitored for redudancies in convolutional layers and the maximum distance between gradient norms. Finally, we show that implementing biased sampling distributions allows for a reduction in the total number of training iterations required to train a network.

1 Introduction

With recent advancements in Deep Neural Networks (DNN), large networks are being utitlized to reach state of the art accuracy in fields like computer vision and natural language processing [1, 2, 4]. However, the addition of more parameters adds considerable computional cost during training time. Allocating additional hardware resources may allow for decreased training time only if the network training methodology exploits the resources efficiently with a parallel orchestration of tasks [3].

Methods of parallelizing neural networks usually fall in one of two major categories. Model parallelism distributes computation by designating each node in a cluster to performing forward and backwards compution on a successive section of the neural network. Typically, this is implemented by assigning each node a subset of layers. This form of parallelism is ideal for when networks have too many parameters for a single node to fit into memory. Data parallelism distributes computation by creating a replicated model in each node. With this design, each node is given a seperate set of data to perform forward and backwards computation upon. Before parameters are updated, update gradients are aggregated across all nodes so that all nodes share the same parameter updates in each iteration. Networks may be designed to exploit one or both forms of parallelism.

In this paper, we examine the distrubtion of training data among workers in a data parallel design. In a conventional methodology, the total training set is divided among workers randomly during each epoch. Each worker will have a similar distribution of classes in the subsample of training data they use. Instead, we look at providing seperate workers with a biased distribution of training data that favors one or more classes when compared to the distribution provided to other workers. We hypothesize

2 Related Work

3 Resources

In this section, we keep a record of the important resources used to investigate the current state of the art resource in distributed neural network training.

3.1 MapReduce Based Parallel Neural Networks in Enabling Large Scale Machine Learning

In this paper, the authors explore the use of MapReduce clusters to parallelize neural networks.

- MRBPNN 1 uses a classic scenario where each compute instance in the MapReduce cluster is given the same neural network. Its weird however that they don't describe how the instances share gradient information.
- MRBPNN 2 uses a ensemble technique to maintain a classifcation accuracy among several weak classifiers.
- MRBPNN 3 spread the compution of a single graph across nodes.
- The author uses very simplifie networks where training and testing accuracy reach a near 100% accuracy. I don't think this is fair since the true limitations of the model are not tested

3.2 Biased Importance Sampling for Deep Neural Network Training

In this paper, the authors use the loss value as a metric for importance sampling in parallel training of deep neural networks

- In the experiments, the authors use a comparison of the gradient norms as an important metric.
- Optimal training time is seen when approximating the loss of the training batch through an additional model.

https://www.hindawi.com/journals/cin/2015/297672/