Julian Singkham
CS 2300/11
Dr. Berisha

Lab 10: Multi-GPU

Abstract

The purpose of this lab is to familiarize ourselves with using Horovod to utilize multiple GPUs, and quantify the benefits of distributed computing. When compared to single GPU performance, multiple GPUs have a much more stable accuracy curve (see Fig 1 and Fig 2), and computational time is vastly superior with large datasets.

Part 1:

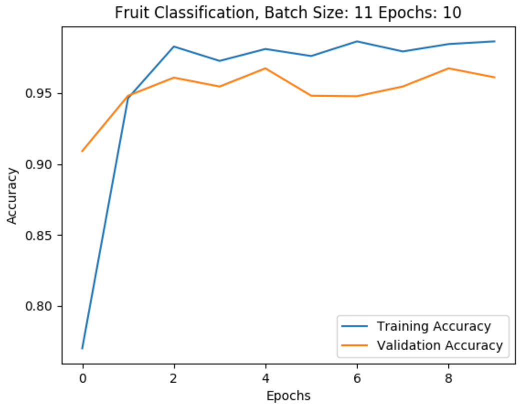


Figure 1: Fruit Classification Using Multiple GPUs Set to Lab 9's Optimized Hyperparameter

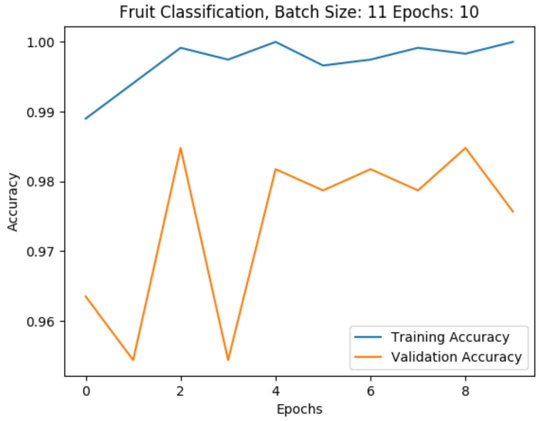


Figure 2: Lab 9 Fruit Classification with Optimal Hyperparameters

Using the hyperparameters from Lab 9, you can see how the multi-GPU model is slightly less accurate than the single GPU model. One cause for the discrepancy is that the learning rates between the two are not interchangeable. Lab 9's version had a learning rate of 0.0005 while this version has a learning rate of 0.1. In terms of timing, the multi-GPU model took 1 minute and 3 seconds to compute with an average epoch time of 5 seconds. The single GPU model took 9 minutes and 41 seconds to compute with an average epoch time of 56 seconds.

Part 2:

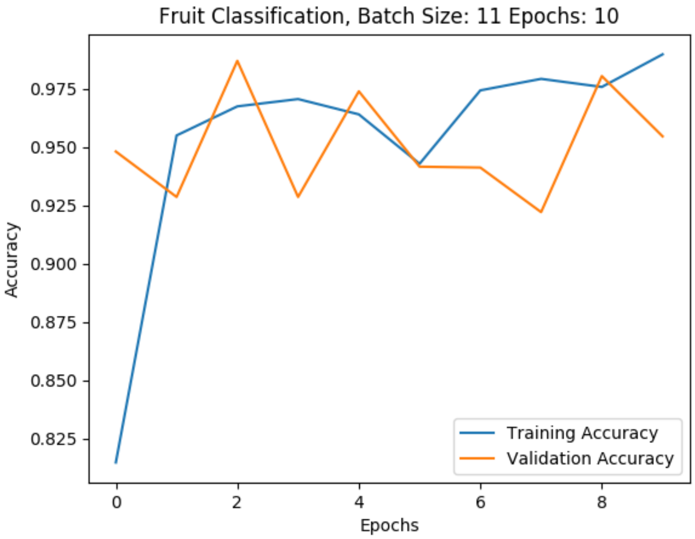


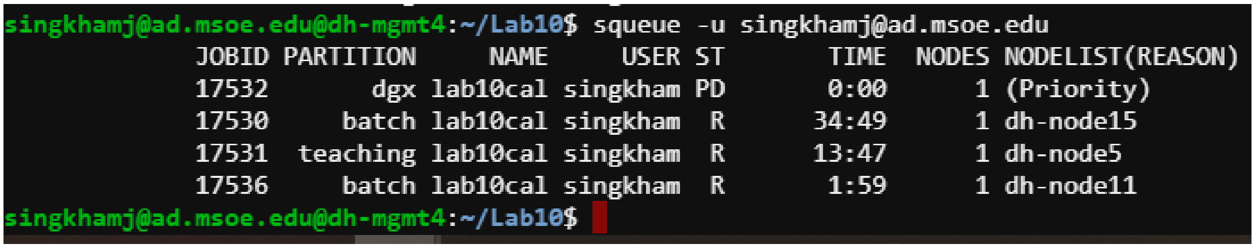
Figure 3: Fruit Classification model with Four GPUs

With 4 GPUs, the model took 1 minute and 14 seconds to compute and had a much higher level of accuracy at 0.9869.

Part 3:

The only difference between the fruit classifier and caltech256 is that caltech256 includes validation_split as an augmentation. This means that the caltech256 splits the data into three groups: training, testing, validation. Validation is very helpful in that it can be used to help fine tune the hyperparameters of the model. The test set can be reserved for the final test for model, which helps with measuring generalization and notifies of overfitting as the model has never seen the test set, whereas the validation set is used recursively.

Problems:



```
singkhamj@ad.msoe.edu@dh-mgmt4:~/Lab10$ squeue -u singkhamj@ad.msoe.edu
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
             17532       dgx lab10cal singkham PD       0:00      1 (Priority)
             17530     batch lab10cal singkham  R      34:49      1 dh-node15
             17531  teaching lab10cal singkham  R      13:47      1 dh-node5
             17536     batch lab10cal singkham  R       1:59      1 dh-node11
singkhamj@ad.msoe.edu@dh-mgmt4:~/Lab10$
```

I've had issues with part 3 and 4 in that the batch jobs would get stuck in queue and never run. Additionally the jobs have been taking a very long time to compute. I'm not sure if its an issue on my part or the sheer size of the data set is to blame