Chris Culver

Jiafang Liu

Torch and TensorFlow on GPU’s - Midterm Report

**Introduction**

Machine learning has been making significant advances in the recent years. AI programs, such as DeepLearning have beaten human champions at their own games such as Go. On the hardware side, GPU acceleration is proving to be a very powerful tool. Although machine learning has been around for decades, it has primarily used CPU power. With GPU advances it is natural for machine learning programs to take advantage of GPU speed. This project will compare popular machine learning programs which use different platforms.

This project’s goal is to compare the most popular machine learning programs that use GPU acceleration such as TensorFlow and Torch. The programs will be performance tested to see which is better optimized for GPUs. Reliability, accuracy and usability will also be compared between each software package. The conclusion of this project will result in a comparison to assist people deciding between TensorFlow and Torch.

This report is an update on the progress of the project at this point in the semester. We will start by introducing machine learning and how it works. We then discuss the results we already have, which are applying machine learning concepts to some fairly image data sets. The results from these runs will be included. After this we explain what we are currently working on, which is running the programs on GPU’s. We lastly discuss what is left to be done in the project, and what results can be expected in our final report.

**Machine Learning Basics**

The purpose of machine learning is to literally teach the computer how to detect things in different categories. The first step of this process is called the training step. This requires a set of data whose training data and training label are known. We first want to show the computer how to correctly classify the data in the dataset. Then it will attempt to classify other pieces of data without a priori knowing the answer. To classify the data the computer can use and user defined model. Depending on the data, different models will have different success rates.

The two main model categories are linear and non-linear models. Two linear models one could use are Logistic Regression(LR) and Linear Discriminant Analysis(LDA). Three non-linear models that could be used are K-Nearest Neighbor(KNN), Classification and Regresstion Tree(CART) and Gaussian Naive Bayes(GNB). Each of these models can be trained on the data, then can be tested for accuracy. This would then inform the user how well the models work to classify unknown data.

**Results**

Torch and TensorFlow have been compiled and installed on local machines and as well as Colonial one. After installation the first test application was to perform image classification on the MNIST data set. This data set contains various images of the 10 numerical images. Doing this exercise helps learn to construct the machine learning model . It can also be used to test the effectiveness of different models in Torch and Tensorflow. After completing the MNIST classification we made it slightly harder and classified the NotMNIST data set. This contains the letters A through J with a wide variety of unorthodox fonts. The dataset was designed to look like classic MNIST dataset, while looking a little more like real data. This creates an additional challenge for the image recognition model since there are more classification categories and we also need to clean the data before using it . The wide variety of fonts is also meant to distract computer recognition programs which is used in “captcha codes” for many websites.

**Current Work**

With the basics set up we then went to getting the programs running on Colonial One so we can re-run the programs with GPU computing. This has been met with issues since we do not have sudo access to Colonial One. Getting the required dependencies installed locally and correctly linking to them was quite difficult. We do have both Torch and TensorFlow working on Colonial One. This means that we can begin performance testing our programs. This will be the very next thing we do.

We have also begun researching the advanced portion of our project. We have found a video data set, UCF-101 which has 101 different categories. We have also found a collection of papers which attempt to classify the data set. We are still searching through the papers to see if anybody has used Torch or TensorFlow to complete the task.

**Remaining Work**

Having both programs running on colonial one we can now start serious performance tests using their GPU’s. The colonial one GPU nodes are equipped with Nvidia K20’s. We will first do some basic image recognition performance testing using the MNIST and NotMNIST data sets. This will help us determine which of the two programs is better if the main concern is speed. Although these datasets are not particularly large, there is an abundance of data available online. This makes the performance of an application very important when deciding on which to use to do a classification type problem.

The above will be the completion of our projects baseline. We will have learned about machine learning and applied it to some fairly simple problems. Also these problems will have been evaluated on GPU machines and tested for performance. The remainder of our project will be devoted to a more complicated application such as video recognition to test the GPU’s performance on Colonial one.