EECE 5640 Project Proposal

For my final project, I will evaluate the performance differences on several different GPU platforms of the K-Nearest Neighbors (KNN) algorithm, a non-parametric supervised learning method used for classification that uses proximity to make predictions about the grouping of an individual data point. This project aligns with option (b) of the possible project proposals, which I will be completing alone.

To evaluate the performance of KNN, I will use three datasets of increasing size and complexity:

1. UC Irvine Machine Learning Repository: Iris Dataset
    a. A small dataset of 150 images of plants
    b. Features physical characteristics of the plants (sepal/petal length and width)
2. MNIST Dataset: Kaggle
    a. A collection of 70,000 28x28 anti-aliased images depicting handwritten digits with a training set of 60,000 examples and a test set of 10,000 examples
3. CIFAR-10 or CIFAR-100 Datasets: CIFAR
    a. A collection of 60,000 32x32 color images evenly split among 10 classes (50,000 training images and 10,000 test images)

These datasets will be used and represented as a high-dimensional feature vector so that it can be used by the KNN algorithm. In the case of Iris, it's the physical characteristics of the plants. For the MNIST and CIFAR, the features represent the intensity values of the pixels. Since KNN computes the Euclidean distance between a query point and all training samples, it results in more computationally expensive efforts for larger datasets (MNIST/CIFAR) as opposed to a smaller baseline dataset (Iris). However, since calculating the Euclidean distance is a relatively inexpensive operation for the feature vectors of these datasets (as opposed to other algorithms which rely on much more computationally expensive machine learning techniques), I anticipate that the duration of training on the test images will be feasible.

I will evaluate the performance of the algorithm on two different GPU systems. The Explorer Cluster offers NVIDIA Tesla P100 GPUs and NVIDIA V100 GPUs on the courses-gpu partition (4 GPUs per node) which will be the two GPU platforms I will experiment with. If time permits, I will also experiment with the A100-SXM4 8-GPU nodes on the ai-jumpstart partition since I have allocation access for that partition, and compare that directly to the 4-GPU nodes on the courses-gpu partition. Since I am trying to utilize a GPU instead of a CPU for this algorithm, I will either use a custom CUDA kernel or an optimized CUDA-based library (such as cuML), or both to compare their performance and scalability.

I will attempt to collect the following three general metrics during the experiments: latency/duration, memory utilization, and FLOPS. Latency is the easiest to implement, and I

believe I can use nvidia-smi or a tool like nvprof to collect data for the other two metrics. These results will vary with the different dataset sizes. I will also collect metrics on the KNN algorithm's accuracy to determine how many predictions were correct for each dataset, as these will be important in determining how robust the dataset or algorithm was (with it most likely being caused by the implementation of the algorithm as these datasets are popular and widely used).

The following lists the expected grades I would receive depending on their requirements:

- A: 1 workload evaluated, 3 different inputs used on 2 different platforms, all results reported and analyzed thoroughly in the project writeup.
- A-: 1 workload evaluated, 3 different inputs used on 2 different platform, all results reported, but little analysis of the data included in the project writeup.
- B+: 1 workload evaluated, 2 different inputs used on 2 different platform, all results reported and analyzed thoroughly in the project writeup
- B: 1 workload evaluated, 1 input used on 2 different platform, all results reported and analyzed thoroughly in the project writeup
- C: 1 workload evaluated, 1 input used on 1 platform, all results reported and analyzed thoroughly in the project writeup
- F: Any submission lower than the above grading criteria

Extra Credit Opportunities:

- Evaluate the performance of KNN on the eight A100 GPU Nodes on top of the other two platforms
- Implement the algorithm in both CUDA and a CUDA library to evaluate the performance difference between my own raw CUDA implementation and via the library
- Evaluate a second algorithm with the same criteria as the first (either K-Means Clustering or a Convolutional Neural Network (CNN))