Brandon Ferencik
Jacob Gomez
Professor Cao
Math 10, Spring 2019

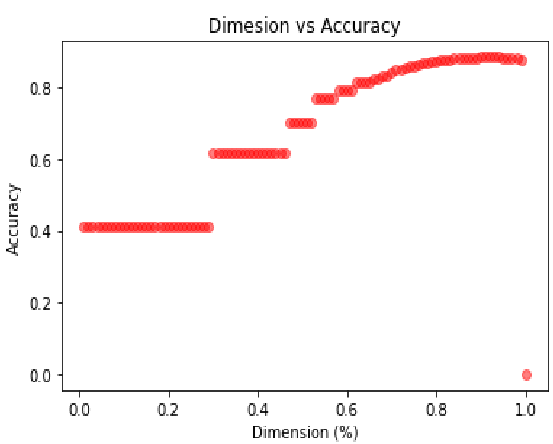
An Analysis of F-MNIST Classification Models

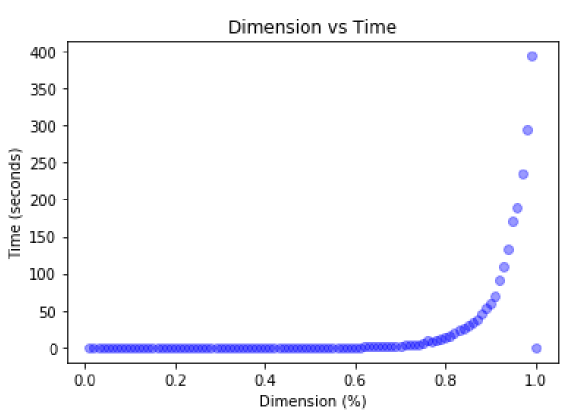
Objective:

The F-MNIST set is a database of 60,000 samples of clothing items, amounting to a total of 10 different items to be classified. As the title suggests, the problem requires the use of classification algorithms in order to predict the value of a particular class. For example, a picture of a sandal might have a class of seven, and a purse might have a class of eight. The two models of interest to classify the F-MNIST clothing items are K-Nearest Neighbors and Neural Network. For convenience, K-Nearest-Neighbors and Neural Network will be abbreviated as “kNN” and “NN,” respectively. The accuracy, advantages, and disadvantages of both models are to be discussed.

kNN:

The kNN model works by assessing the distances that a testing sample has with the rest of the training data. Then, there are votes amongst “k” closest neighbors. The highest proportion of votes established for a particular class amongst k training neighbors is decided as the class for the testing sample. For example, using five nearest neighbors to classify a sandal image would involve identifying at least three closest training samples amongst five that are sandal images.The advantage of the kNN model is that it is parameter-free. That is, there are no weights to train and no cost function to minimize. However, the disadvantage is computation time. While the model lacks weights to train, every iteration requires thousands of distance calculations in order to make a prediction, thus costing time. In the long run, with many images to classify, kNN might prove inefficient. Principal Component Analysis, or “PCA” for short, allows for the dimension of the images to be reduced while retaining the vital information contained in them. Dimension reduction serves to mitigate the burden of long computation time. The figures on





the left show a plot of dimension, expressed as a percentage, vs accuracy achieved and time of computation. The plots convey that there are diminishing returns on accuracy as dimension increases, capping at 90%. Additionally, the diminishing returns in accuracy is met with an exponential increase in time of computation.Thus, determining the appropriate dimension for kNN depends on how much the user is willing to trade time for accuracy. A reasonable dimension reduction would be 75%, where the observed average accuracy across all folds is 86.2%, and the time of computation is a mere 5.6 seconds. While the computation time is a disadvantage in the kNN model, a trained NN model may be chosen to rectify this issue.

Neural Network:

A neural network is essentially a collection of functions that act on a set of data. Each layer of a neural network consists of several neurons whose individual activations are dependent on the previous layer's output, where the first layer of the network is primarily the input of the dataset. Each subsequent input essentially becomes a single transformation of the data with the use of adjusted weight parameters.

The neural network's input, in context of the F-MNIST dataset, is an input of 60,000 images, each sample being 28x28 resolution and flattened to a 784-dimension vector. The output of the system is a list of different probabilities for each class of clothing given to a sample. For example, given an input sample image, the output of a feedforward linear regression model would be 10 cumulative probabilities given for that sample image. The actual prediction outputted by a linear regression-based NN is the highest probability of that list . Data was pushed through a Multi-Layer Perceptron Classifier (MLPClassifier) and we were able to achieve 85% accuracy on test sets using K-Fold CV methods. Hidden layer sizes were kept at the default size of 100 neurons and we used an 'adam' solver to minimize the log-loss function. The NN's learning rate *eta* was also fixed to 1e-4 and yielded an average loss of about 0.2 for each fold.

Conclusion:

We found that although we can achieve a maximum accuracy of ~89% using a K-Nearest-Neighbors algorithm and Principal Component Analysis, it is time-consuming and CPU intensive. With a feed-forward MLP Classifier, we are able to achieve similar results with an accuracy of 85% without any need for dimension reduction. Training time for a NN took approximately 7 minutes to yield these results whereas a KNN algorithm took upward of an hour for similar results. Due to such a large amount of data and the nature of the kNN algorithm, the algorithm falls short in its ratio of time to accuracy. Thus, a neural network seems to be the better choice for classifying images such as these.