Project Update 2

Timeline Given in Initial Proposal

November 4th: K-Nearest Neighbor algorithm completed
November 17th: Project Update with K-Nearest Neighbor complete, Least Squares nearing completion
November 20th: Least squares Algorithm Complete (This will line up with when we are learning neural networks so starting neural networks at this point makes sense).
December 4th: Neural Network Complete
December 8th: Any additional cross-validation, statistics of error rate and comparison between algorithms complete.
December 12th: Final Report Due

Done
In Progress
Not Started

Current Project Overall Status

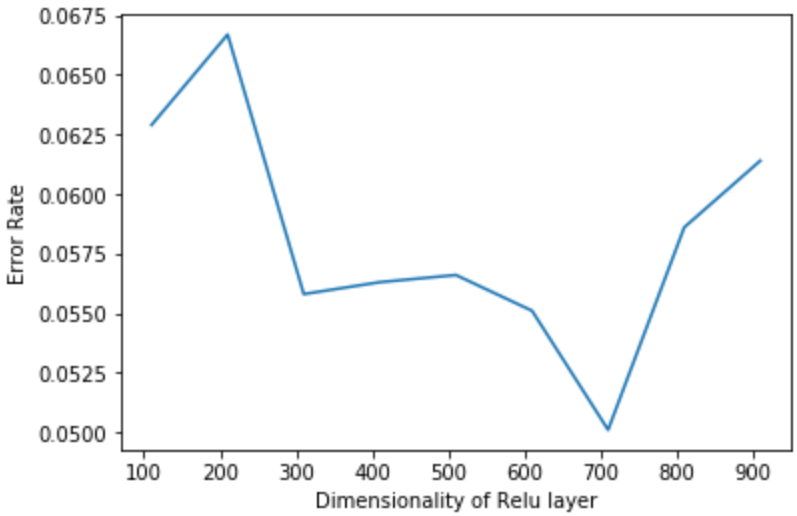
Link to github page where notebooks with results can be found:

https://github.com/mgiordano12/ece532_final_project

See Appendix for Progress Report 1 material

Neural Network

- Nearing completion, installed sklearn, tensorflow and am using them per piazza post.
- All parameters are explained within the python notebook but a quick overview.
- Same preprocessing as KNN, Least squares (see below)
- Two layer neural network used with
    - First Layer: Relu- For speed of learning, dimensionality of this layer determined through testing (see below).
    - Output Layer: Softmax- For classification of multi-class problems (see notebook for further explanation). Dimensionality of this layer is 10 since we have 10 output classes
    - Sparse categorical cross entropy- To avoid outliers having too much pull



-

Figure 1: Error Rate of Algorithm Versus Relu Layer Dimensionality

As you can see in the graph above, the error rate takes a dip ~700 dimensions. This makes sense, since we noticed there were 712 significant singular values. After a bit more digging (see notebook if you want specifics), the best dimensionality I found did turn out to be 712 with a test error of 4.16%.

Todo Before Final Submission

- Cross-Validation
- Comparison of Algorithms
- Clean-up Code
- Write Report

Appendix- Progress Report 1

Pre-Processing

- Same for KNN and Least Squares
- Took first 712 singular values to represent the MNIST image based on seeing a drop-off in significance in singular values from the chart below.
- Trained on SVD-reconstructed data based on 712 singular values

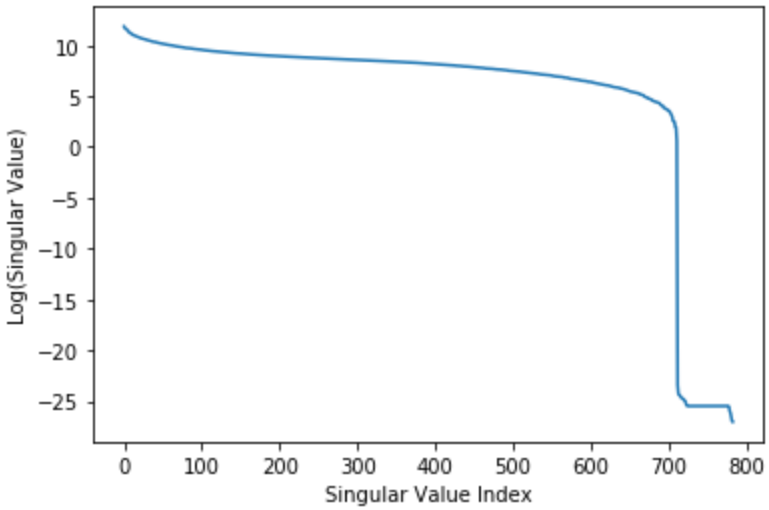


Figure 1: Graph of SVD Value vs. Index for 784 Pixels in MNIST Images Training Data

KNN

- Complete, error rates of [0.0309, 0.0309, 0.0338, 0.0347, 0.0396] for values of k = [1,5,9,13,27]

Least Square with Regularization

- Complete, minimum error rate of .1424 for lambda = 10e7
    - Lambda 10e-7 through 10e9 were evaluated
    - Most error rates for lambda were~.147