Michael Giordano

ECE 532 Project Proposal

Project Dataset: My project will use the MNIST database of handwritten digits. It can be found here. There are 60,000 training examples and 10,000 test examples in this dataset. The classification goal is to determine what digit (between 0-9) is written in the 28x28 pixel framework. These digits are the labels for the problem. Each training example has 28 dimensions (since it's 28x28).

Algorithms that will be investigated:

- *K-nearest Neighbors*- The first algorithm I'll investigate is the k-nearest neighbors algorithm. This is an algorithm that memorizes training examples and classifies based on those most similar. I'll use Euclidean distance between images as a measure of proximity. The formula for Euclidean distance is included below. Where x_{ij} is the pixel in the ith row and jth column of the 28x28 example X and y_{ij} is the pixel in the ith row and jth column of the 28x28 example Y.

$$\text{Eq1. } \left(\sum_{i=1}^{28}\sum_{j=1}^{28}(x_{ij}-y_{ij})^2\right)^{1/2}$$

    - Parameters: I'll try 3,5, and 7 (odd so I don't have to break ties) for k and compare results.
- *Least-Squares with SVD*- The second algorithm I'll use is least-squares with SVD. I'll first try the normal least squares solution

$$\text{Eq2. } w = (A^T A)^{-1} A^T d$$

with all of the training images included in the matrix A. There is a good chance this will not end up being invertible and/or overfits the data significantly. As a result, I'll analyze the data to come up with the SVD where

$$\text{Eq3. } A = u \Sigma V^T$$

I'll remove all values where $\sigma = 0$ to come up with a full rank matrix and then solve the least squares problem using

$$\underline{\text{Eq4.}}\ w = V\Sigma^{-1}u_{tilde}{}^{T}d$$

- Parameters: I'll adjust the minimum value of σ needed for the column to be included in the SVD. As my minimum value for σ rises, the accuracy of the approximation of the training data will decrease. However, it may generalize better as the effect of noise will also decrease. I'll adjust these to try to find a good medium. If time allows, I'll do this using a bias variance tradeoff graph for the low-rank approximation.

- *Neural Networks*- The third algorithm I'll use is neural networks. This will be a 2-layer neural network which uses cross-entropy loss as it's loss algorithm. The neural network will include a bias node.
  - Parameter: Since we will be learning more about neural networks in the coming weeks, the loss function/number of layers in the network are subject to change.
- *Validation*- All of the algorithms will be trained on the 60,000 length training set and validated on the 10,000 length testing set. Misclassification rate (% of examples misclassified using 1/0 misclassification loss) will be the measure of success to evaluate and compare the algorithms. If time allows, I'll also combine the training and testing set and run 7-fold cross validation (7 different combinations of 60,000 training, 10,000 testing examples) for each algorithm and compare results to the original results.

Project Github: https://github.com/mgiordano12/ece532_final_project

Project timeline:

November 4th: K-Nearest Neighbor algorithm completed
November 17th: Project Update with K-Nearest Neighbor complete, K-Means nearing completion
November 20th: K-means Algorithm Complete (This will line up with when we are learning neural networks so I 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