Final Project First Update

Jiaqi Chen

Since the initial proposal, I have finished reading the MNIST dataset and completed the classification of the data using ridge regression.

First of all, in the MNIST data set, there were 60,000 samples in the training set and 5,000 in the testing set, we need to read the data. All images are normalized to 28*28 pixels, with pixel values ranging from 0 to 255, with 0 representing the background and 255 representing the foreground. Here I use the unpack function in the struct module and the Numpy in Python. The struct module can process binary data stored in files, and the transformation of binary files can be achieved through the unpack function of this module. np.fromfile is a function that reads binary data using a known data type. It returns an array. After I read the file by np. fromfile, it returns a one dimensional array. We need to take out every image, so I use the reshape function. Then I normalize the pixel value of the image to 0.0~1.0 by deviding 255. Then I define a function to convert the label to one hot encoding. That is for each data, if it belongs to class i, the ith element is 1 and other elements are 0. For example, for the number 1, its one hot encoding is [0 1 0 0 0 0 0 0 0 0].

Then I use the linear regression method ridge regression to achieve the classification. To point out that, in the initial proposal, I planned to use the least squares method to do the classification. But when I applied the least squares method to train the model, it showed the feature matrix is singular. So I turned to use the ridge regression to train the model. Here I achieved the multiclass classification by train the weight k for class k by one- vs- rest. The label is 1 if the data is in class k, otherwise is 0. Then I used the ridge regression with 10000 training data to calculate the weight matrix w (since there are 10 classes). The lambda is chosen by cross validation. Next I calculated the predicted label by test_data*w with 1000 test data. The class with the largest value is the predicted class. Then I calculated the error number and the error rate by comparing the predicted class and the correct class (it has been provided in the test data set). The error rate is 0.195.

Next I plan to use the SVM to do the classification to find a more accurate classifier.