CMSC 254: Homework 5

Ryan D'Mello

1 Feed-Forward NN Framework

Comments in Python code denote the purpose of necessary functions and variables

1.1 Design

We have a Neural Network with 2 hidden layers. For activation, we use the Sigmoid function. We will have 4 - 1 = 3 weight matrices, one for each layer's connection to the next one.

1.2 Functions

We include a Softmax loss function, a mean-squared error loss function, and cross-entropy loss, which will be used during back-propagation. Moreover, we recalculate the weights using Stochastic Gradient Descent

1.3 Fixed Parameters

From the nature of the dataset, we have 784 Neurons in Input layer and 10 Neurons in Output layer.

2 Results

I experimented extensively with various numbers of hidden layers (and the number of neurons in each of those layers), with varying learning rates ('eta' in the .py file), and varying sizes of epochs. For each of these combinations of layers and learning rates, I noted the prediction rate. After over 50 such "experiments", it turned out that the highest correct prediction rate I could achieve on the training data was .92346 or .7654 error rate). This highest correct prediction rate (.92346) on the training data was realized with the following parameters:

learning rate (eta): 0.8 Number of hidden layers: 2 Size of hidden layers (of neurons): 128 Number of epochs: 1000

The total run time (for training) with these parameters was about 20 minutes

I then used the weights and biases from these parameters to conduct the two test runs, and the correct prediction rate held up nicely: .9141 on the first set (TestDigitX) of test digits. The predicted output for this first test set is in the attached file "Test1-Predictions.csv". The predicted output for the second set of test digits (TestDigitX2, which has no corresponding labels) is in the attached file Test2-Predictions.csv.

The .py file which I have attached a smaller number of epochs, so you can run it quickly and verify that it executes correctly. At the end of each epoch, the error rate is printed. You can see that the error rates decrease fairly steadily. If you wish to replicate the 92.346 correct prediction rate

I achieved, you will need to change epochs to 1000 (but the total run time will then be about 20 minutes).

3 Submission and Running Python file

The following documents are submitted: Two CSV's with test results. In addition, the Python file can be run by simply running the main method and adjusting the parameters accordingly. This can be done in Python console as well.