EECE6036 - Homework 3

Long Dang Vo

November 10th, 2022

1 Problem 1

1.1 System Description

This neural network consists of 784 input layers, each corresponding to a single pixel in a 28 x 28 MNIST dataset image, 180 hidden neurons, 10 output neurons, momentum (alpha) of 0.5, a high output threshold of 0.75, a low output threshold of 0.25, a learning rate of 0.01, and 800 epochs. Each of the 10 output neurons is assigned a unique number (from 0 - 9).
I began by initializing the hidden neurons to 150, which produced a respectable accuracy rate. When I set it to 180, the accuracy was then noticeably improved and the training time was unaffected. The number of epochs is set to 800 because 1,000 epochs would be too large for my model and 500 epochs would be too small for training. 800 epochs can produce the best overall results. I initially trained with the learning rate set to 0.02, but later changed it to 0.01 because I needed to input more epochs to get a good result.

1.2 Results

TRAIN CONFUSION MATRIX

	0	1	2	3	4	5	6	7	8	9
0	390.000000	0.000000	0.000000	1.000000	0.000000	2.000000	2.000000	2.000000	1.000000	2.000000
1	0.000000	393.000000	2.000000	0.000000	0.000000	2.000000	0.000000	1.000000	2.000000	0.000000
2	1.000000	1.000000	373.000000	1.000000	1.000000	4.000000	3.000000	3.000000	8.000000	5.000000
3	0.000000	0.000000	13.000000	359.000000	0.000000	11.000000	2.000000	4.000000	7.000000	4.000000
4	0.000000	0.000000	1.000000	0.000000	381.000000	1.000000	4.000000	1.000000	2.000000	10.000000
5	3.000000	1.000000	0.000000	10.000000	6.000000	347.000000	8.000000	6.000000	13.000000	6.000000
6	4.000000	0.000000	1.000000	1.000000	1.000000	3.000000	384.000000	3.000000	3.000000	0.000000
7	3.000000	3.000000	1.000000	2.000000	2.000000	0.000000	1.000000	375.000000	3.000000	10.000000
8	1.000000	2.000000	3.000000	9.000000	3.000000	7.000000	4.000000	4.000000	363.000000	4.000000
9	1.000000	2.000000	1.000000	8.000000	11.000000	2.000000	1.000000	15.000000	6.000000	353.000000

TEST CONFUSION MATRIX

	0	1	2	3	4	5	6	7	8	9
0	95.000000	0.000000	1.000000	0.000000	0.000000	1.000000	0.000000	2.000000	1.000000	0.000000
1	0.000000	95.000000	0.000000	2.000000	0.000000	0.000000	1.000000	2.000000	0.000000	0.000000
2	0.000000	2.000000	79.000000	5.000000	2.000000	0.000000	2.000000	5.000000	5.000000	0.000000
3	0.000000	1.000000	1.000000	84.000000	0.000000	5.000000	1.000000	2.000000	6.000000	0.000000
4	1.000000	0.000000	0.000000	0.000000	82.000000	2.000000	5.000000	2.000000	0.000000	8.000000
5	3.000000	1.000000	2.000000	2.000000	1.000000	81.000000	4.000000	1.000000	4.000000	1.000000
6	4.000000	0.000000	3.000000	0.000000	1.000000	3.000000	89.000000	0.000000	0.000000	0.000000
7	1.000000	1.000000	2.000000	1.000000	1.000000	0.000000	1.000000	85.000000	1.000000	7.000000
8	0.000000	2.000000	2.000000	4.000000	2.000000	4.000000	2.000000	2.000000	79.000000	3.000000
9	1.000000	0.000000	0.000000	0.000000	8.000000	0.000000	0.000000	5.000000	2.000000	84.000000

Figure 1.1. Confusion Matrix for Training Set and Test Set

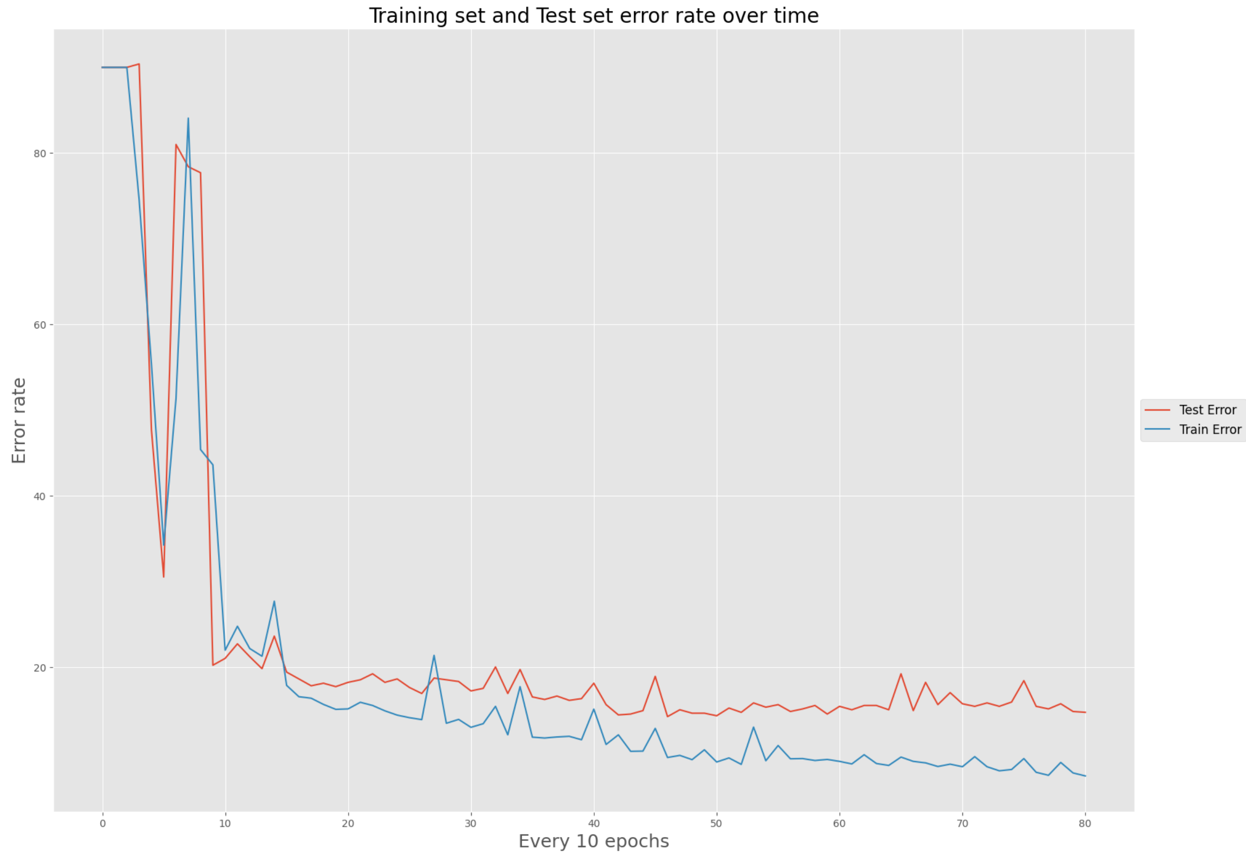

Figure 1.2. Training Set and Test Set error rate over time

1.3 Analysis of Results

Throughout the entire training, a very noticeable decrease in the error rate can be seen. Figure 1.1 shows that the accuracy is approximately 92% after training, and the model can classify an estimated 370/400 data points for each image digit, with the images of digit 2 having the highest accuracy (393/400; 98%) and digit 9 having the lowest accuracy (353/400; 88%). The accuracy is roughly 85% when compared to the test set's confusion matrix, with the digits 0 and 1 scoring the highest (95%) and the digit 8 scoring the lowest (79%) respectively. The system operates as expected in that the network can continue to learn to improve and classify the training set more accurately after each epoch, despite the fact that this only slightly improves test accuracy. Overall, the vast majority of data points had accurate classifications.

The error rate did level off after about 150 epochs and then fluctuated, eventually getting close to 0 after 600 epochs. Finally, the numbers that were incorrectly identified can be explained by the fact that, when considering some of the digits, they will have a similar draw to one another. For instance, 7 and 1 can occasionally be misunderstood if we write them too quickly.

2 Problem 2

2.1 System Description

This autoencoder consists of 784 input layers, each corresponding to a single pixel in a 28 x 28 MNIST dataset image, 180 hidden neurons, 784 output neurons, a momentum (alpha) of 0.5, a high output threshold of 0.75, a low output threshold of 0.25, a learning rate of 0.01, and 1000 epochs. Each of the 10 output neurons is assigned a unique number (from 0 - 9).
I use the same neural network system for the autoencoder to ensure that the autoencoder's goal is to compress the 784 input neurons to 180 hidden neurons and then have the same output as the input, especially when compressing and decompressing the data value. The change for the system in the autoencoder would be to set the bias with the value of 1 and unchange for the whole system which significantly decreases the error rate when training.

2.2 Results

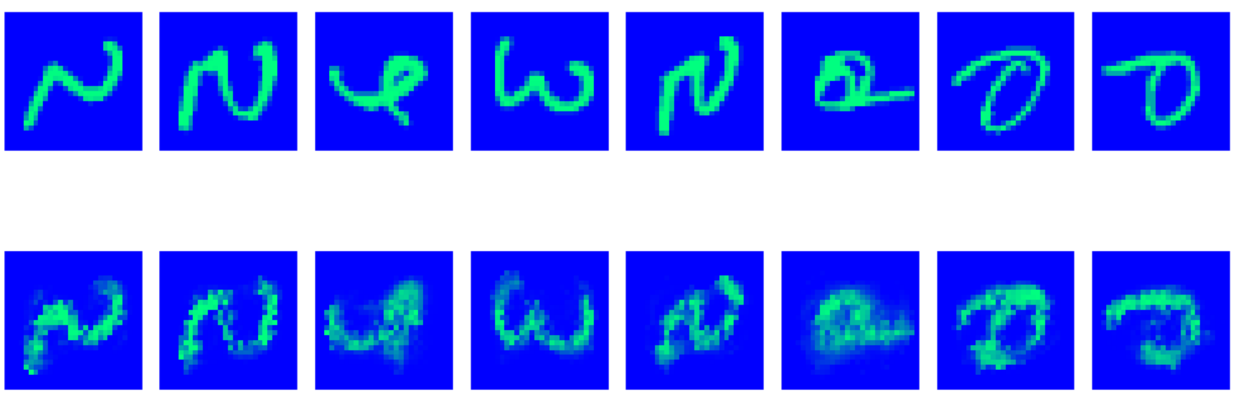

Figure 2.1. Inputs from Test Set vs Output from AutoEncoder

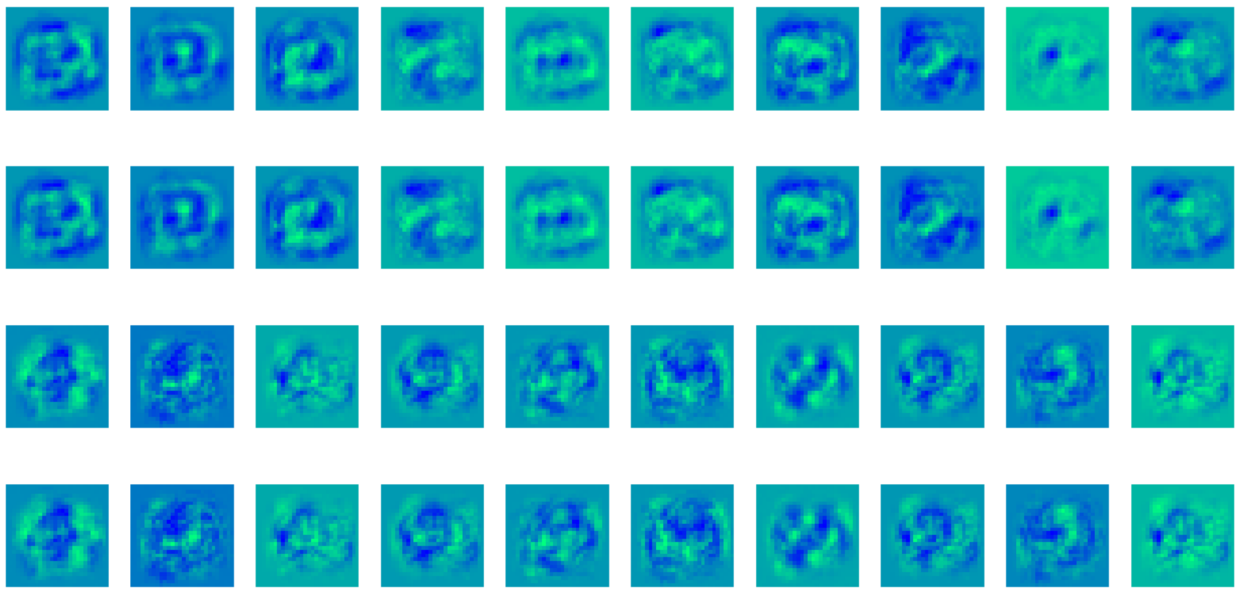

Figure 2.2. Random features images of neural network and autoencoder

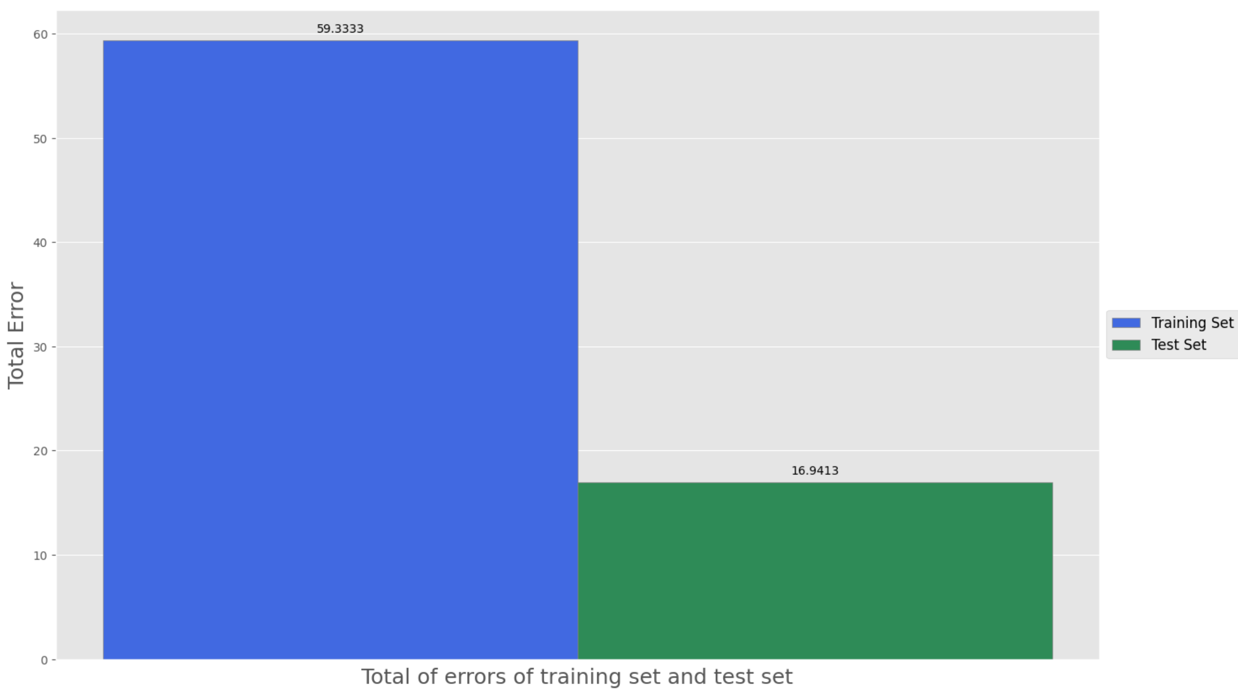

Figure 2.3. Total loss of training set and test set after training

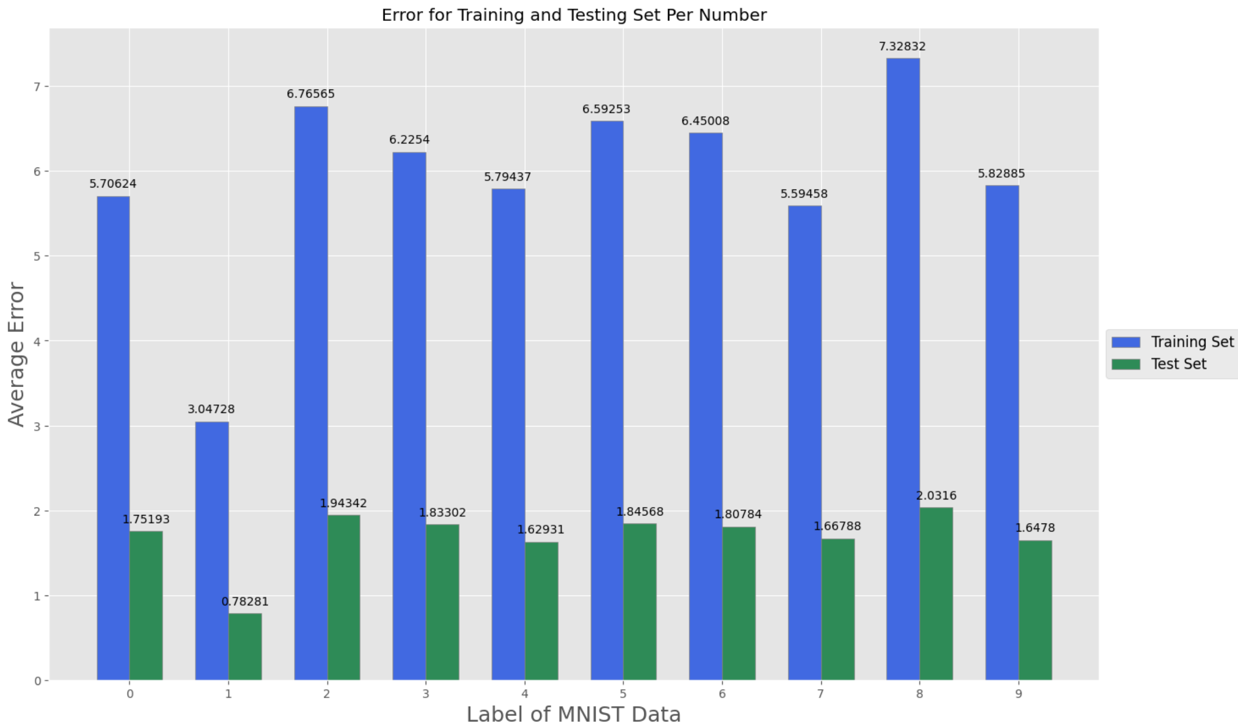

Figure 2.4. Average error for each digit of training set and test set

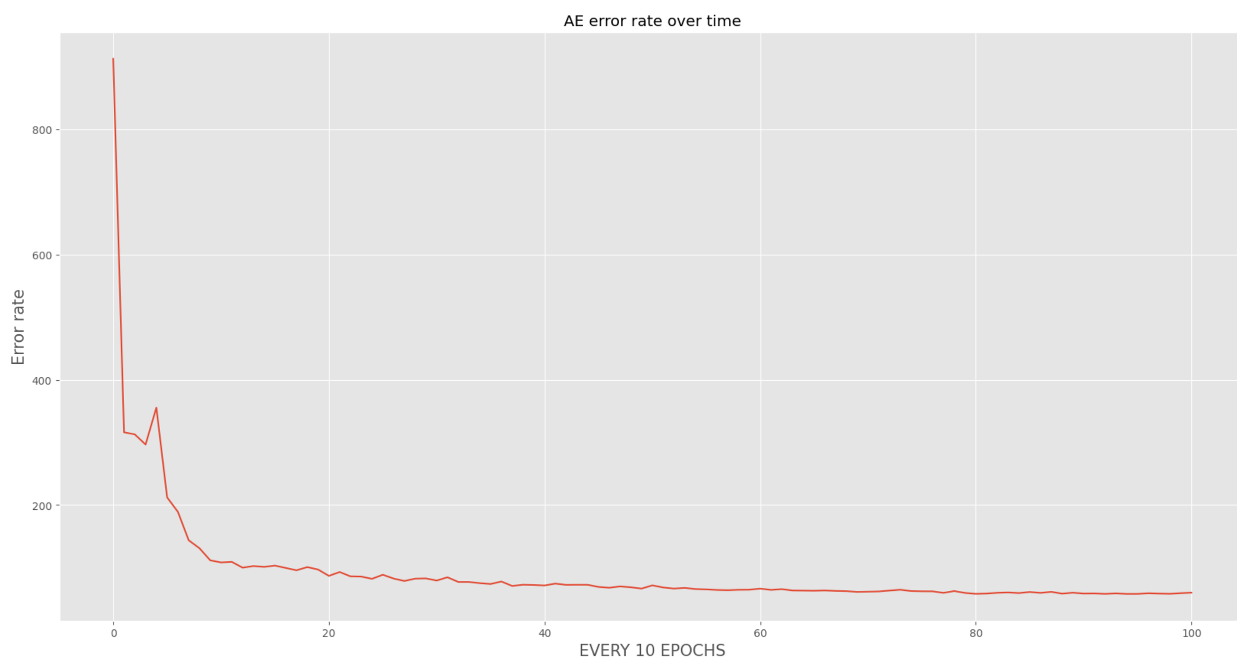

Figure 2.5. Training set loss error over time

2.3 Analysis of Results

Although it is unclear from the first figure (figure 2.1) what is to be classified, we can see that the numbers to be identified are 2, 3, 5, and 6. The error rate caused a higher loss of data points because the feature sets were complex, making it difficult to identify or generalize, which ultimately resulted in a less-than-accurate result. A total of 40 trained features from the hidden neuron output of the neural network and the hidden neuron output of the autoencoder are displayed in Figure 2.2. However, it cannot be said with certainty that any of the features show traits unique to individual digits. The system can use overlaps of these digits collectively to decode the compressed input and produce the full output. Figures 2.3 and 2.4 show that the test's overall loss is equal to one-third of the training set, indicating that higher error may result from the assumption that unseen data will be provided. Last but not least, figure 2.5 shows the autoencoder's error rate over time. This error function exhibits the same pattern over time, making sharp corrections in the early epochs, followed by decreases in error rate that are more difficult to explain.