EECE6036 HW3

Jack Fliegel

# Problem 1

**System Specification**

The following describes the results of a python program that simulates a perceptron that trains the weights 794 weights to represent a 28 x 28 pixel image. The learning rate that I used for the perceptron was **N = 0.01** and the simulation ran for **40 epochs**. The learning rate and total epoch numbers were found through trial and error where I looked for a negative exponential trend line and high enough learning rate such that the heatmap would have a visible difference between before and after training. The weights were initialized between -0.5 and 0.5 random and uniform.

**Results**

A graph with a line

Description automatically generatedA graph of a bar graph

Description automatically generated with medium confidence

Figure : Epoch vs Error Fraction

Figure : Precision, Recall, F1, and Error Fraction (Before and after training)

A diagram with a red line

Description automatically generated

Figure : Bias weight vs Metrics

We can see that the system is well trained at this point as the bias rate has little to no impact on the Error Fractions, Precisions, Recalls, and F1 Scores of the model. I would say the trained bias weight was the best fit for the given data.

A comparison of a comparison of weights

Description automatically generated with medium confidence

Figure : Before and After heatmaps of weights

A number in a rectangular box

Description automatically generated with medium confidence

Figure : Model ran with challenge set, shows result of each number

**Analysis**

The learning rate I chose took multiple tries to get correct and I had to settle for a more strongly fit model with a greater learning rate than a smaller learning rate but a weaker model. If I had a low learning rate the trend line in Figure 1 was not as exponentially negative and smoother but would not be enough to change the heatmap and therefore not be able to classify the challenge set as well.

We can see in the challenge set results in Figure 5 that numbers which overlap more with 0 have a false output and numbers that overlap with 1 more have a true output. The number 5 has a stronger false output because it would overlap with the dark (negative weights) areas on the heatmap in Figure 4. 8 has a higher true output because the way 8s are shaped they avoid the negative areas of the heatmap and hit the lightly colored (positive weights) areas of the heatmap.

# Problem 2

**Problem Statement**

Using the same learning rate and number of epochs as the previous problem, I trained the data on a dataset of 0’s and 9’s (0 = false, 9 = true). Weights were again set at random between -0.5 and 0.5 uniformly.

**Results**

**A graph with numbers and a line

Description automatically generatedA graph of a bar graph

Description automatically generated with medium confidence**

Figure : Epochs vs Error Fraction (0 and 9 training set)

Figure : Precision, Recall, F1, Error Fraction ((0 and 9 training set)

**A diagram of a weight loss

Description automatically generated with medium confidence**

Figure : Bias Weight vs Data Metrics

**A comparison of a weight scale

Description automatically generated with medium confidence**

Figure : Before and After Heat Maps of Weights

**A number in a box

Description automatically generated with medium confidence**

Figure : Table of Outputs for Challenge Set

**Discussion**

The performance when using 0’s and 9’s to train the model was very similar to Problem 1. A pattern I see within the Figure 10 output table is this model identifies numbers that have a similar structure to 9 as true and numbers that are rounder and look like 0s as false. This is as expected because the model is trying to linearly separate the numbers that it was not trained with, and it will categorize numbers that look more closely to a 9 as true and numbers that look more closely to a 0 as false. In both of my models, I had a low learning rate, but the learning seemed to happen quite fast as we can see with the steeply negative trend lines in Figure 1 and Figure 6. The initial values of the weights impacted the number of epochs it took to get to a low enough error fraction.