Single Threshold Neuron and Perceptron

Problem 1

The goal in this problem is to train a single threshold neuron to discriminate between two handwritten digits. The training will be done using the simple reinforcement paradigm.

System Specification

The number of epochs in this problem is 40 since it was recommended in the assignment that 40 epochs should be ok considering the value of the learning rate η is equal to 0.01. I decided to keep these values as “default” as possible to keep the neuron in a good position to not skew the results with an overly high learning rate or take too long to learn with a high epoch amount.

Results

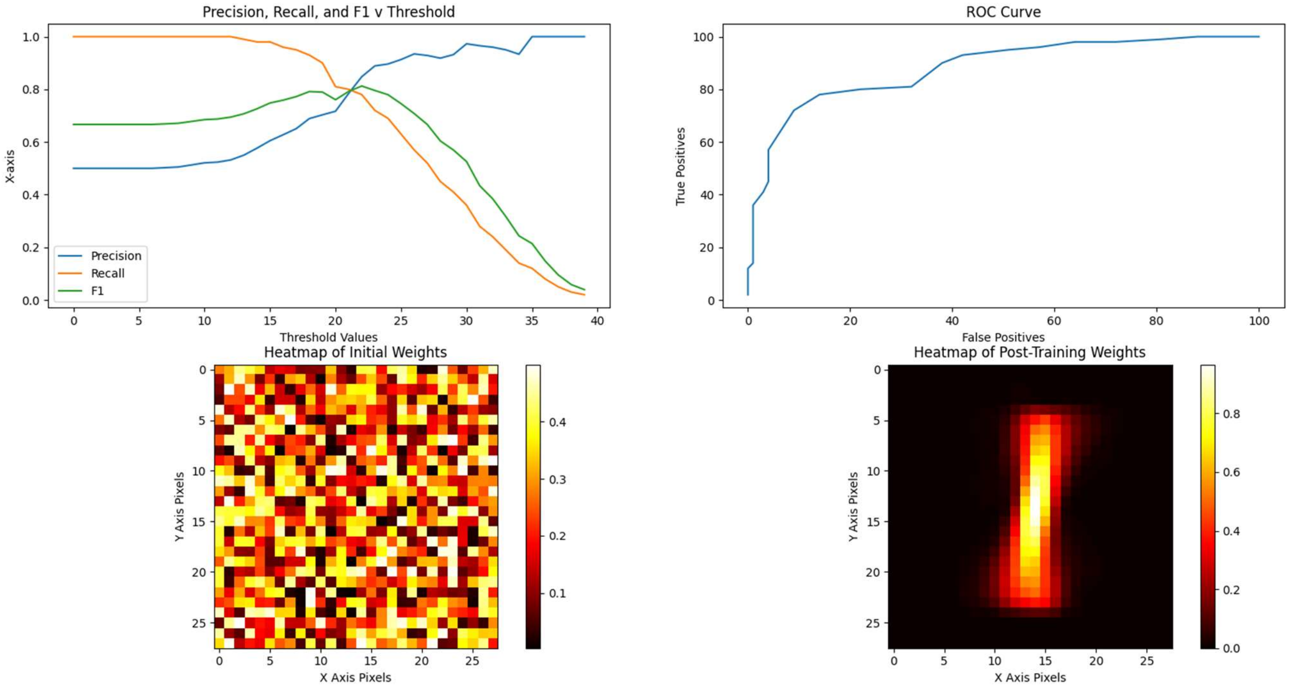
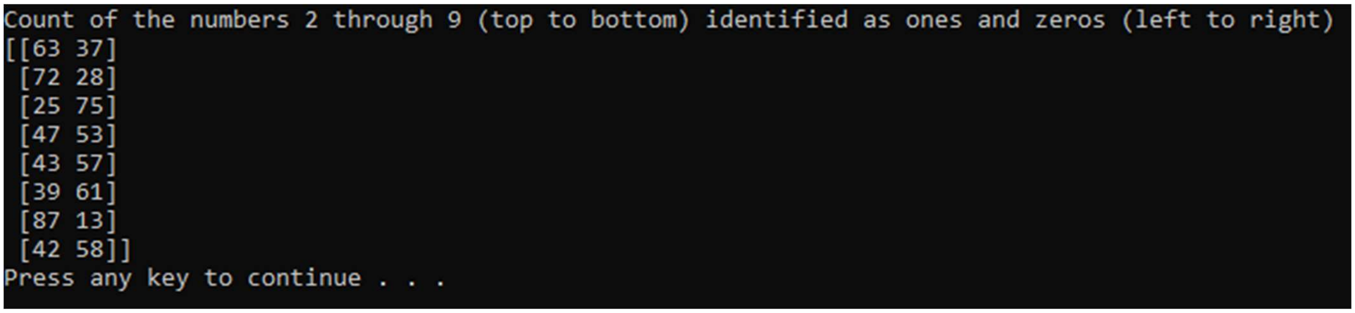


Figure 1 – Four required graphs

The optimal choice of θ according to the Precision, Recall, and F1 v Threshold graph would most likely be around 21. A threshold of 21 allows for a more balanced amount of the precision, recall and f1 scores for both the testing and challenge sets outputs.



```
Count of the numbers 2 through 9 (top to bottom) identified as ones and zeros (left to right)
[[63 37]
 [72 28]
 [25 75]
 [47 53]
 [43 57]
 [39 61]
 [87 13]
 [42 58]]
Press any key to continue . . .
```

Figure 2 – Challenge set output

Analysis of Results

The results (see figure 1) I achieved with this neuron are quite good in terms of differentiating between a handwritten 0 and 1 and good with the precision. It didn't seem to have a solid recall or f1 score by the end of the training, which meant that the model is not that great at detecting the specific category but is good at predicting a 1 or a 0. I believe this could be due to the fact that 1's and 0's can be drawn differently, which affects the model greatly since it is using pixels to determine the classification, and because precision and recall often reduce each other. So, since precision went up, recall went down.

The challenge numbers (per figure 2) most similarly looking to 1, such as 7 for example, should be more likely to be classified as a one. With seven, the only difference in a handwritten seven (in the given handwritten images) is an extra horizontal line at the top. However, in this case, because of how the weights are distributed, the 7 is actually more likely to mix with the zero due to the fact that the extra line correlates with the zeros line. The negative weights happen to be much larger than the positive weights and thus more 7's are identified as a 0. Thus, the neuron will have a hard time differentiating between other numbers as well.

Problem 2

The goal in this problem is the same as problem 1 except that a perceptron will be used instead of a reinforced neuron.

System Specification

The number of epochs in this problem is 40 since it was recommended in the assignment that 40 epochs should be ok considering the value of the learning rate η is equal to 0.01. I decided to keep these values as “default” as possible to keep the perceptron in a good position to not skew the results with an overly high learning rate or take too long to learn with a high epoch amount.

Results

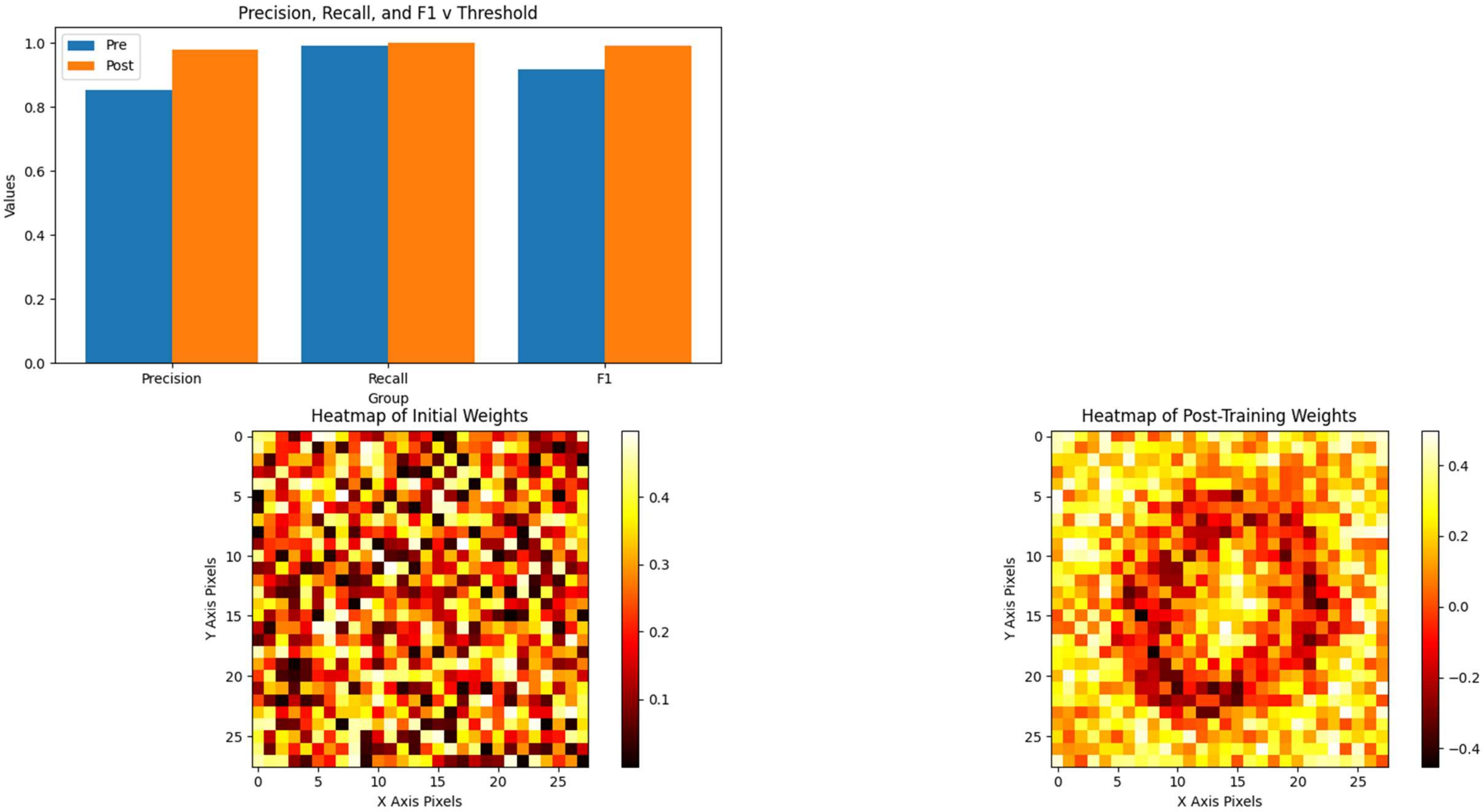


Figure 3: Three required graphs

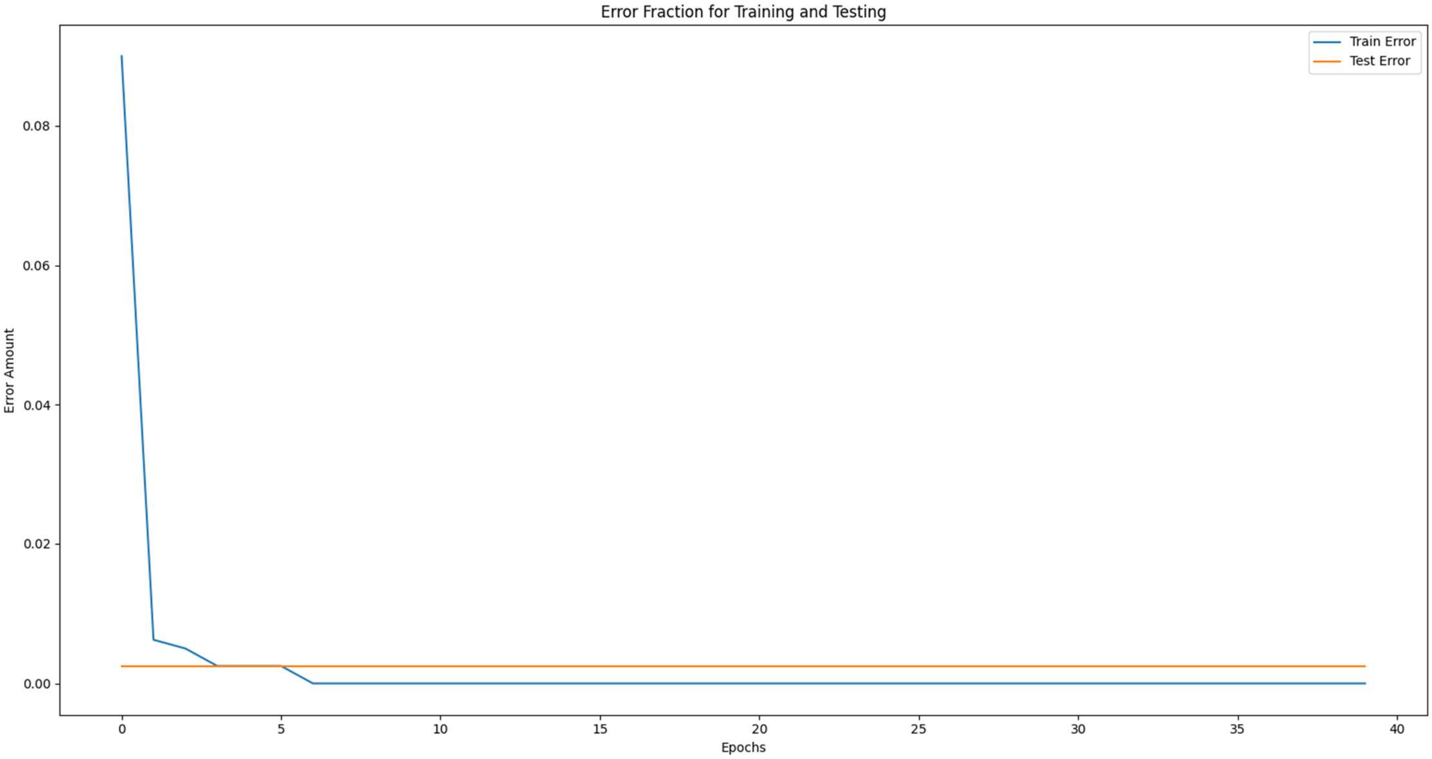
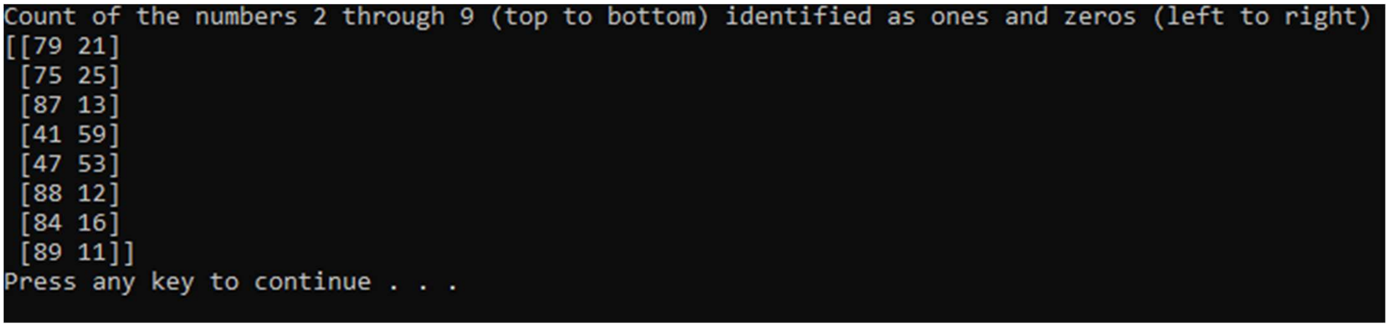


Figure 4: Error Fraction for Training and Testing vs Epochs



```
Count of the numbers 2 through 9 (top to bottom) identified as ones and zeros (left to right)
[[79 21]
 [75 25]
 [87 13]
 [41 59]
 [47 53]
 [88 12]
 [84 16]
 [89 11]]
Press any key to continue . . .
```

Figure 5: Challenge set output

Analysis of Results

The precision, recall, and F1 scores obtained with the perceptron after training are ultimately higher than the values obtained in Problem 1 with the best threshold value (Figure 3). All the scores are very close or equal to 1, unlike the neuron in which only the precision was close to 1. The perceptron also creates a negative 0 on the post-heatmap with average to high weights around it as well, unlike the neuron which creates a distinguishable positive 1 with very negative weights around it. This is due to how the perceptron updates each weight. Since the desired output minus the actual output is taken into account, zeros are pushed towards the negative scale and ones towards the positive side. The perceptron had a better result when handling the challenge set as one can see based on the number 7. The perceptron more accurately defined 7 as a 1, which a 7 is very close to a 1, meanwhile the neuron more likely classified a 7 as a 0, which is not likely (Figure 2, 5). I would much prefer to use a perceptron as I find it to result in a much better model that can be applied to other sets and not just the trained one.