Homework 1: Perceptrons

Joe Arriaga
CS 445: Machine Learning

January 17, 2019

1 Description

The Python programming language was used to model ten instances of a perceptron; each perceptron was responsible for recognizing one of the ten digits (0–9). Together, the ten perceptrons should be able to identify the digit represented by an instance from the MNIST dataset of handwritten digits. Perceptrons were initialized with random weights, then the Perceptron Learning Algorithm (PLA) was used to train the perceptrons to recognize their respective digits. A Manager class was created to handle all ten perceptrons as a system, especially because the operations performed by each perceptron were almost identical and values produced by the individual perceptrons would need to be compared.

The experiment was run with three different learning rates, $\eta = 0.001, 0.01, 0.1$. For each learning rate, an initial accuracy was measured before any training, then the PLA was run for 50 epochs. After each epoch, accuracies with the updated weights were measured on both the training set and a test set. After 50 epochs, the results were the data of accuracy for both sets over the span of the 50 epochs, which are visualized in a plot, and a confusion matrix for the perceptron system on the test set using the final weights.

2 Results

2.1 $\eta = 0.001$

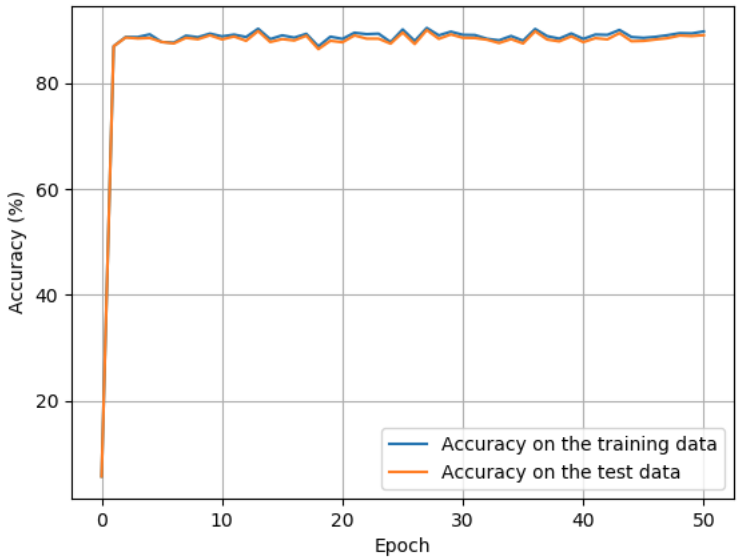
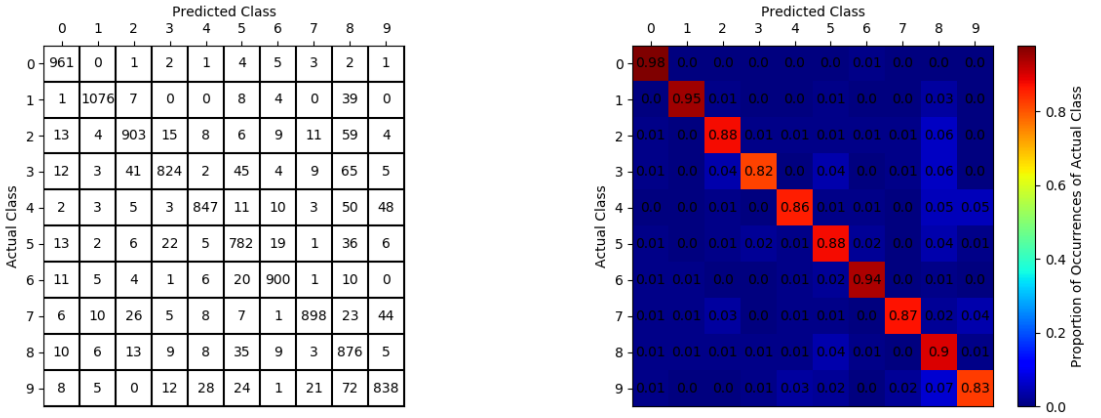


Figure 1: Accuracy with $\eta = 0.001$

The plots of accuracy show no sign of overfitting. The accuracy on the test set is almost equal to the accuracy on the training set for all epochs, indicating that the decision-making of the perceptrons generalizes well and is not specific to the training set data. The system reaches a high level of accuracy quickly, after only a few epochs, but after that shows signs of minor oscillations. The accuracy does indeed vary within a range but that range is only a few percentage points and the variation does not seem to have any regular periodicity. Therefore, although the system may exhibit some oscillation it is not significant and likely the result of small changes to a few weights as the system attempts to achieve a level of precision for which it is not designed. The final accuracy on the test set was 89%.



(a) Confusion matrix with values

(b) Heatmap representation of confusion matrix

Figure 2: Confusion matrix on test set after training has been completed with $\eta = 0.001$

The vast majority of instances were classified correctly. Digits 0, 1, and 6 achieved around 95% correct classification, digits 2, 4, 5, 7, and 8 achieved 85% correct classification or better, while digits 3 and 9 performed the worst with about 82% correct classification. Although there are instances of almost every digit being misclassified as almost every other digit there is no clear pattern as to what may have caused the misclassifications. These errors do not constitute a significant portion of any one class or the total number of instances, as evidenced by the high total accuracy. An intriguing pattern was that many of the misclassifications for every digit predicted 8, it would be interesting to investigate why 8 was predicted so often.

2.2 $\eta = 0.01$

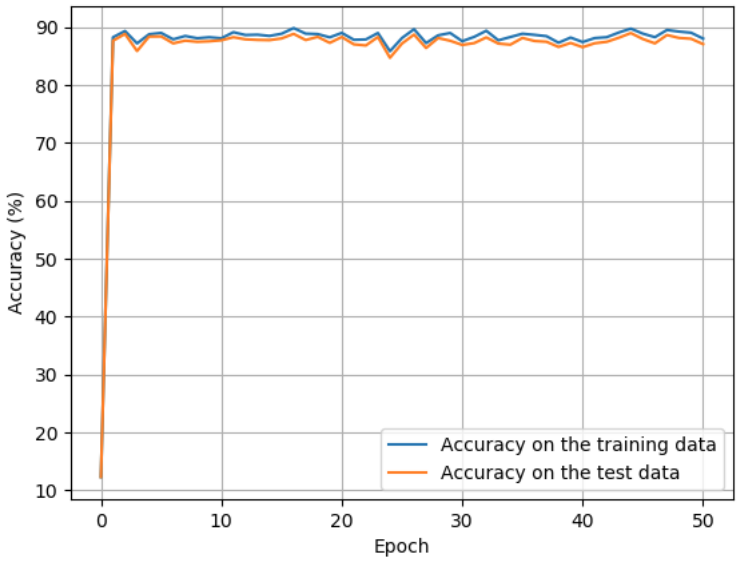
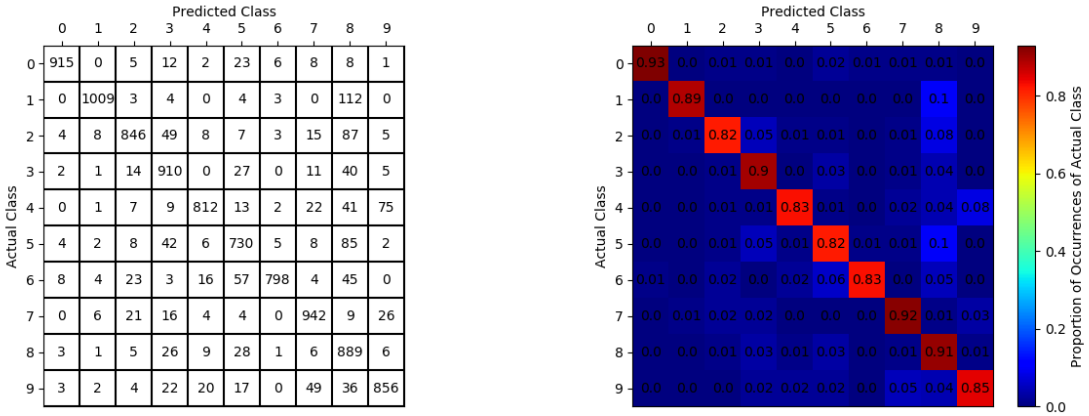


Figure 3: Accuracy with $\eta = 0.01$

Again, the plots show no sign of overfitting as the accuracies on the training set and the test set are almost equal. There are some signs of oscillation, especially between epochs 23 and 35, but these are again minor. The final accuracy is 87%.



(a) Confusion matrix with values

(b) Heatmap representation of confusion matrix

Figure 4: Confusion matrix on test set after training has been completed with $\eta = 0.01$

The results for this learning rate also prove similar to the previous learning rate when looking at the confusion matrix. The vast majority of examples are classified correctly and all at a rate of 82% or higher. The same pattern of incorrectly predicting 8's is also present. Most digits were classified less accurately than with the previous learning rate, which could be explained by a loss in precision with the larger learning rate, except that digits 3 and 7 improved. Is there something about the shape of these digits that makes them easier to identify with a coarse metric?

2.3 $\eta = 0.1$

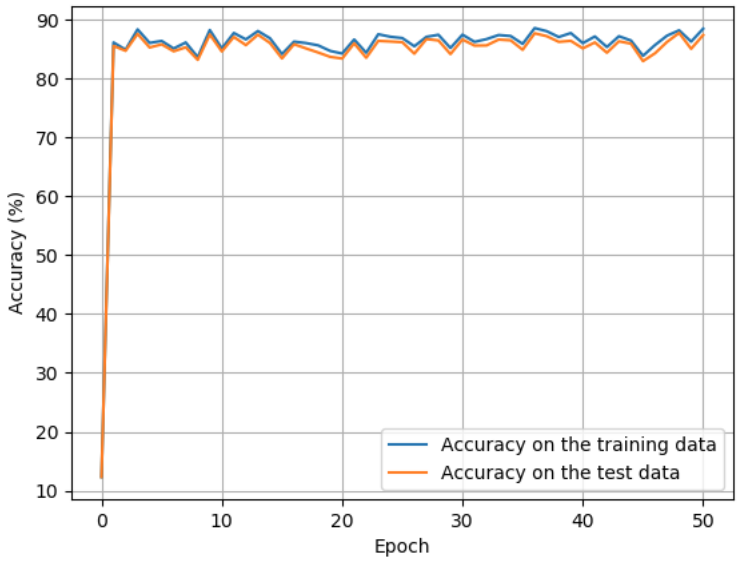
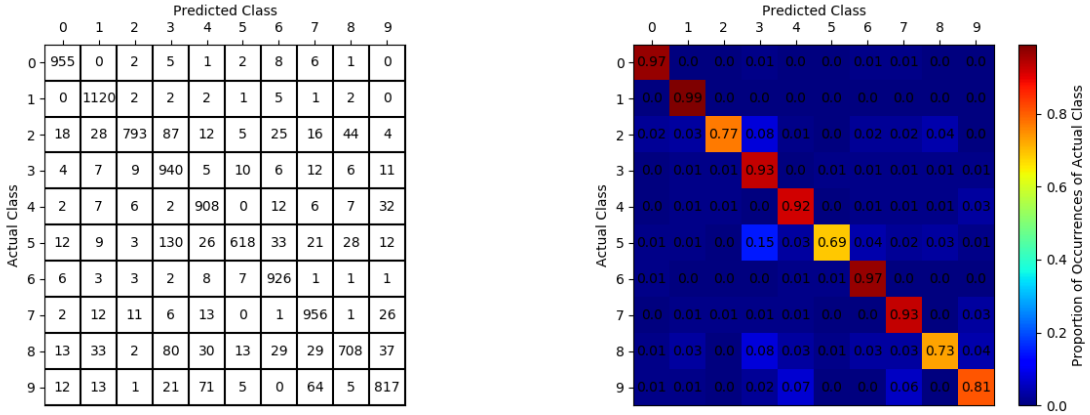


Figure 5: Accuracy with $\eta = 0.1$

Once again, there is no sign of overfitting as the accuracy on the training data and on the test data is almost equal throughout the experiment. These plots show more oscillation than the previous plots but it is still relatively minor. This can be easily explained by the larger learning rate. The larger learning rate means that any change to the perceptron weights is much greater, and thus more likely to overcompensate and not make any progress towards weights that accurately predict the examples. The final accuracy is 87%.



(a) Confusion matrix with values

(b) Heatmap representation of confusion matrix

Figure 6: Confusion matrix on test set after training has been completed with $\eta = 0.1$

The confusion matrix for this learning rate is characterized by a divergence of performance. Six of the ten digits achieved accuracies over 90%, many of which were improvements over previous learning rates. However, digits 2, 5, and 8 showed a drastic decrease in accuracy. The accuracy for digit 9 showed a slight

decrease but was still similar to previous learning rates and is not a member of either group. It is unclear what would cause such a discrepancy among certain digits, there are no obvious characteristics linking the digits that did well or those that did poorly, nor are there shared characteristics separating the correctly classified from the misclassified. There was no pattern is misclassifications as appeared with the previous learning rates, instead, misclassifications were spread evenly across the confusion matrix.

3 Conclusions

The smallest learning rate produced the best overall accuracy, likely because it was able to get closest to the "optimal" weights. As the learning rates became larger accuracy on some digits improved while accuracy on others declined. Even though the overall accuracy was similar on all three models, the models with larger learning rates simply were not consistent in being able to predict all digits. There was no evidence of overfitting in any of the models. There were also no signs of significant oscillation. All models exhibited some oscillation, and the larger the learning rate the more pronounced the oscillations were, however the greatest deviation due to oscillation seemed to be $\pm 3\%$ when $\eta = 0.1$, which is a variation of 3.5% of the total accuracy.

4 Areas for Further Study

The experiments indicate that smaller learning rates are more accurate overall but two questions arise: Would there be diminishing returns such that, below some value, reducing the learning rate would not yield a significant increase in accuracy and what would that value be? Also, would there be an associated need to increase the number of epochs? In these experiments all of the models appeared to reach their maximum accuracy in 50 epochs but would a much smaller learning rate need more epochs, and how many? Finally, could the overall accuracy be improved by decoupling the perceptrons and using the learning rate that worked best for each digit individually?