Logistic Regression for Digit Classification

Bao V. Nguyen

1.1.Logistic Regression Model Fitting

Figure 1 – Log Loss for 40 Iterations

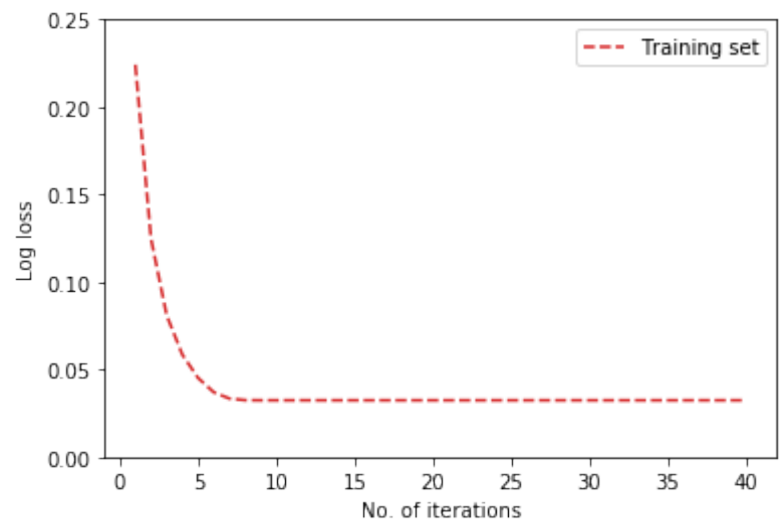
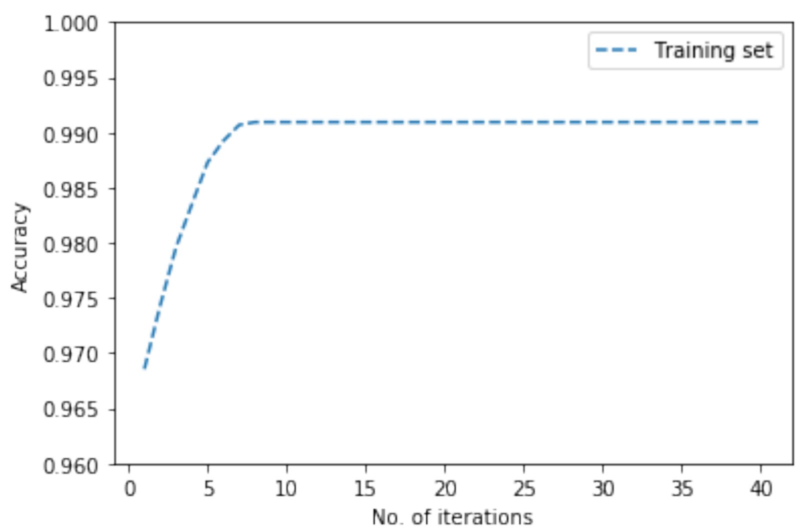


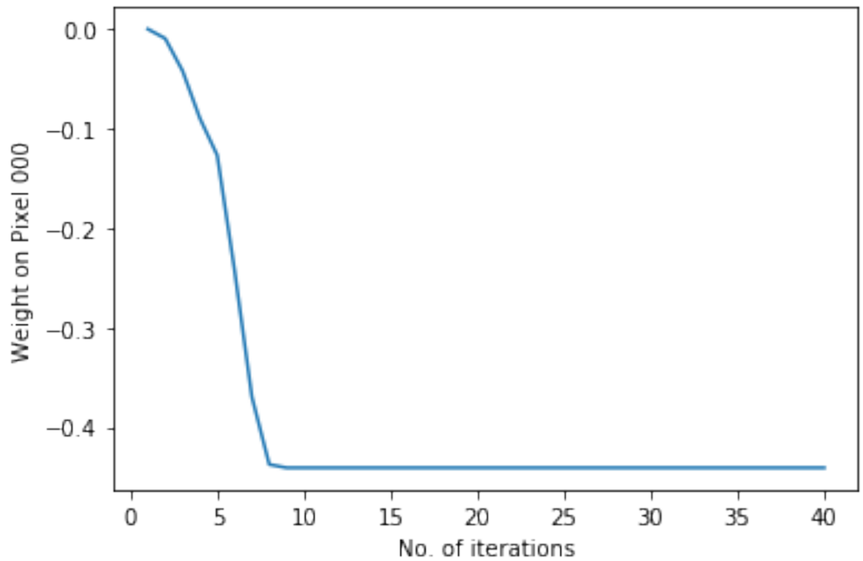
Figure 2 – Accuracy for 40 Iterations



As the number of iterations increases, the log loss reduces from 0.224 to 0.032. Log loss reduces when iterations increase. Similarly, the accuracy increases from 96.86% to 99.09% when iterations increases. This is because the model is using gradient descent to adjust the parameter weights. The more iterations we set, the more adjustments the model will make, which increases accuracy and decreases log loss. As we increase the number of iterations, the precision with which logistic regression tries to fit the data grows. Therefore, the regression algorithm modifies model parameters to account for noise induced fluctuations.

1.2.Access the Weights

Figure 3 – Weights of pixel000 for 40 Iterations



The weight of pixel000 goes from - 0.0005 to -0.4406. It decreases significantly until the 8th or 9th iteration, then remains stable until the 40th iteration. This is because few numbers of iterations will give different values for weights. As the number of iterations increases, the weight values will converge into a constant number.

1.3.Explore C Values

- Value that gives the least loss on the test data: 0.03162277660168379
- Accuracy score of the model: 0.9672213817448311

Figure 4 - Confusion matrix of model on the test data

Predicted	0	1
True		
0	923	51
1	53	956

1.4.Analyze Model Mistakes

Figure 5 – Analyze 9 False Positive Samples

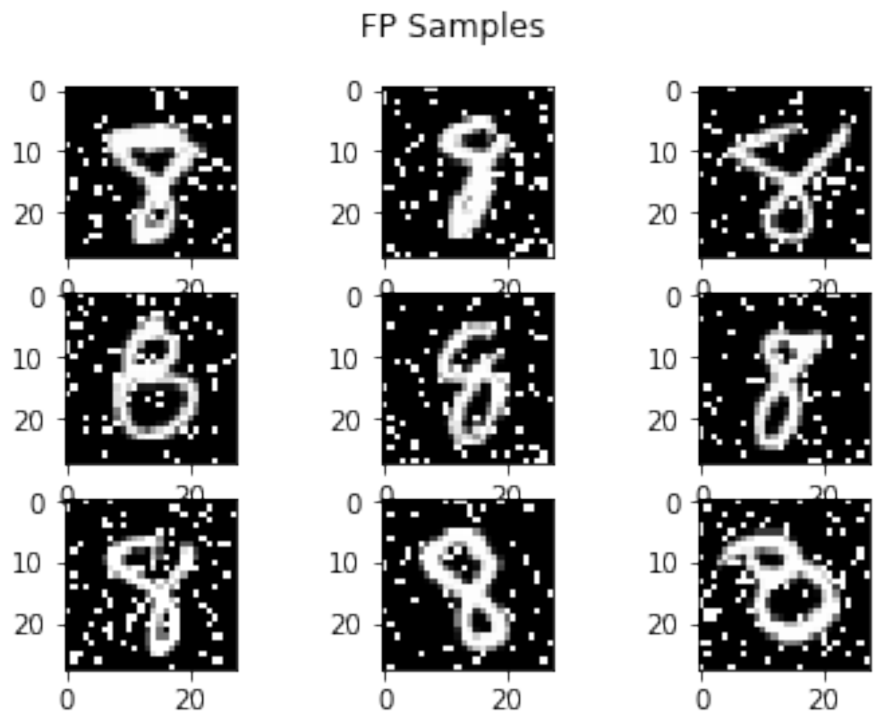
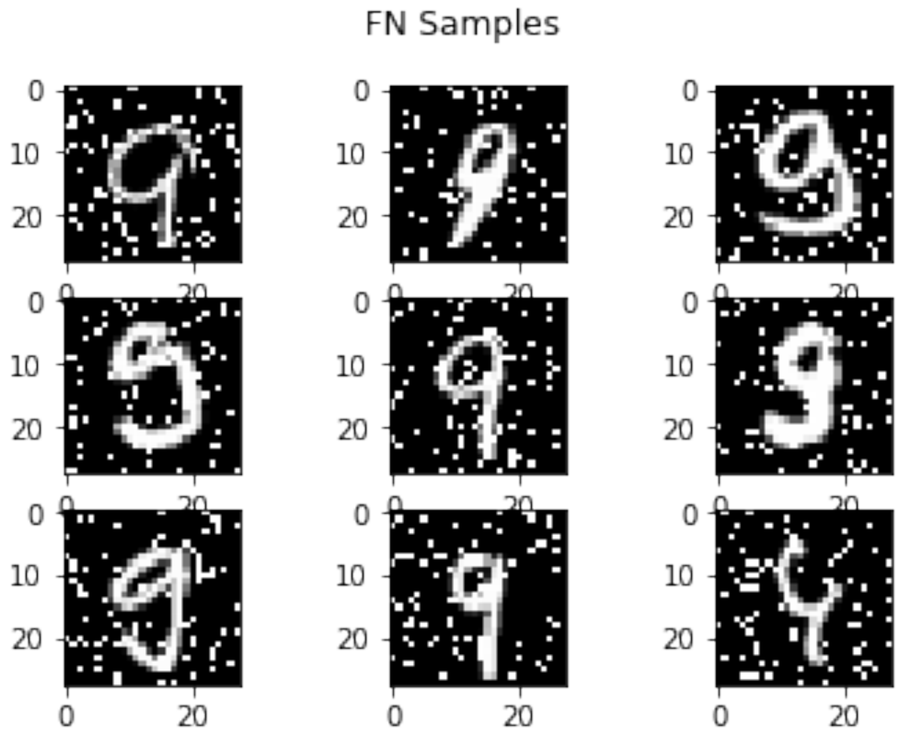


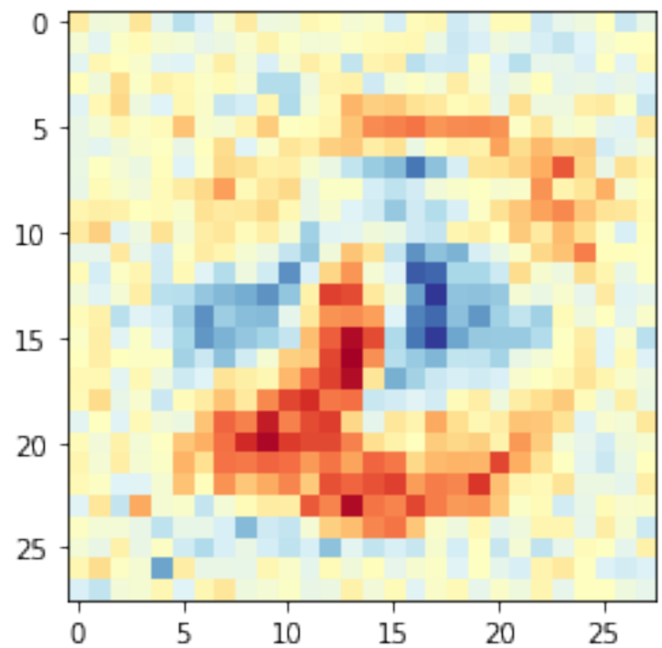
Figure 6 – Analyze 9 False Negative Samples



All examples above are even hard to recognize for human eyes; therefore, it's understandable for the classifier to make mistakes. For the false positives, some examples have a bigger upper curve in the number 8. For the false negatives, some images have bigger lower curve in the number 9. Also, as generally number 9 has fewer pixels than number 9, the false positives have fewer white pixels than black ones, making the classifier think that it’s a 9 while it’s actually an 8. The opposite logic applies for false negatives.

1.5. Analyze the Final Weights

Figure 7 – Heat Map of Final Weights



The red pixel corresponds to negative weights, and the blue pixel corresponds to positive weights. An 8 corresponds to 0 and a 9 corresponds to 1. Here, we can see that the number of red pixels dominate the number of blue pixels. If the pixel value is one (white) in the red areas, the probability of being an 8 increases. Similarly, if the pixel value is zero (black) in the blue areas, the probability of being a 9 increases.