MNIST Classifier using Pytorch

Key Findings:

- *Achieved over 97% accuracy in digit classification on the MNIST test dataset.*
- *The digit '7' was the most commonly misclassified, as observed in the confusion matrix.*
- *On inspecting misclassified examples, it was found that some images (e.g., distorted ‘0’ or ‘6’) were difficult to recognize even for humans, validating the challenge for the model.*

Model Architecture:

A feedforward neural network (Multilayer Perceptron) was used with the following structure:

- Input Layer: 784 neurons (28 × 28 pixels flattened)
- Hidden Layer 1: 500 neurons, with ReLU activation
- Hidden Layer 2: 125 neurons, with ReLU activation
- Output Layer: 10 neurons (corresponding to digits 0–9)

  Summary of the architecture: 784 → 500 → 125 → 10

Optimizer : Adam

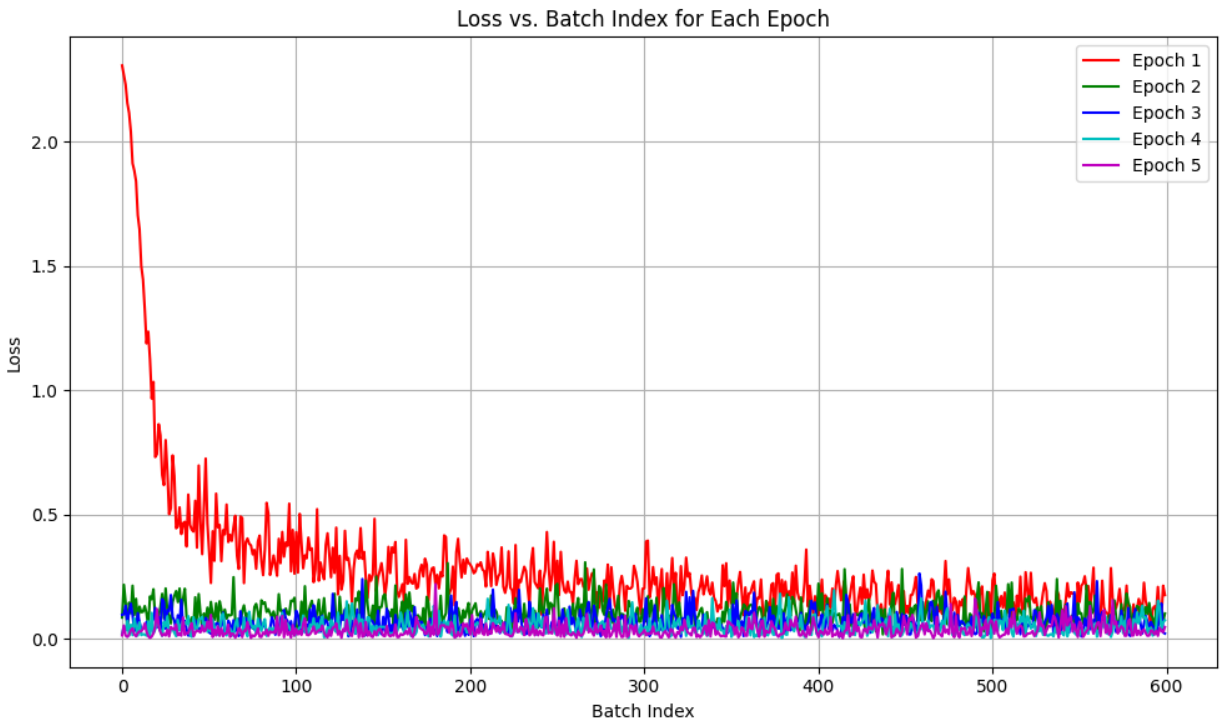
Loss function: CrossEntroppyLoss

Learning rate: 0.001

Number of epochs: 5

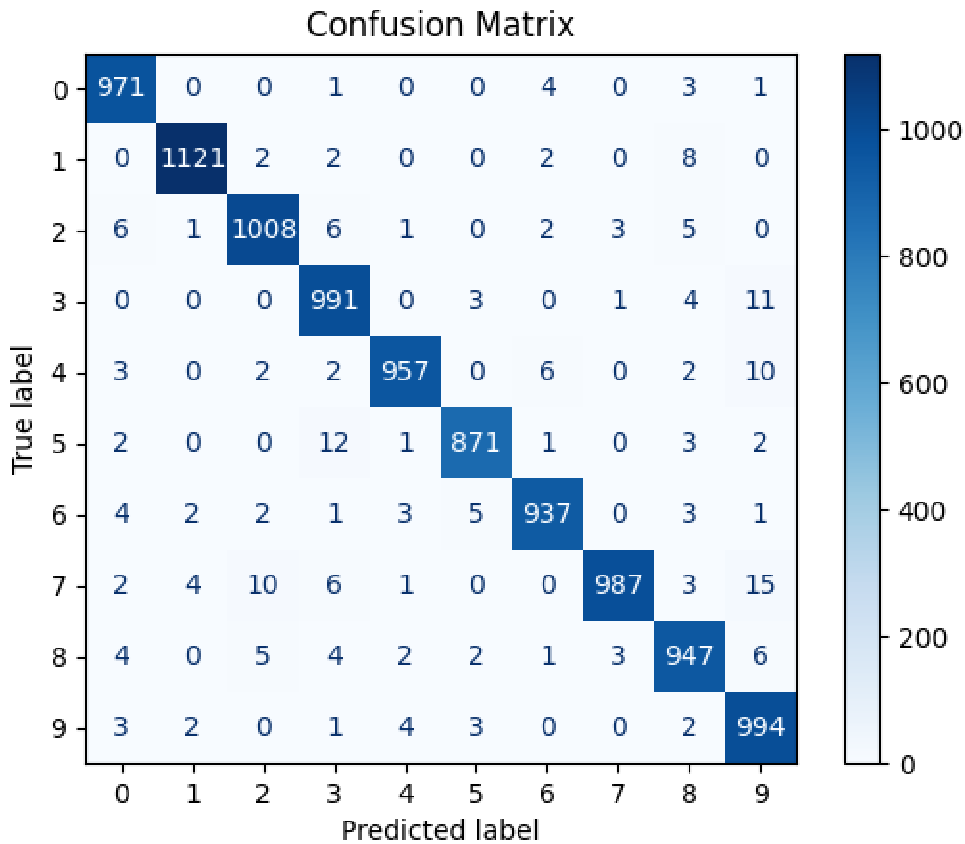
Batch size: 100

Plots:



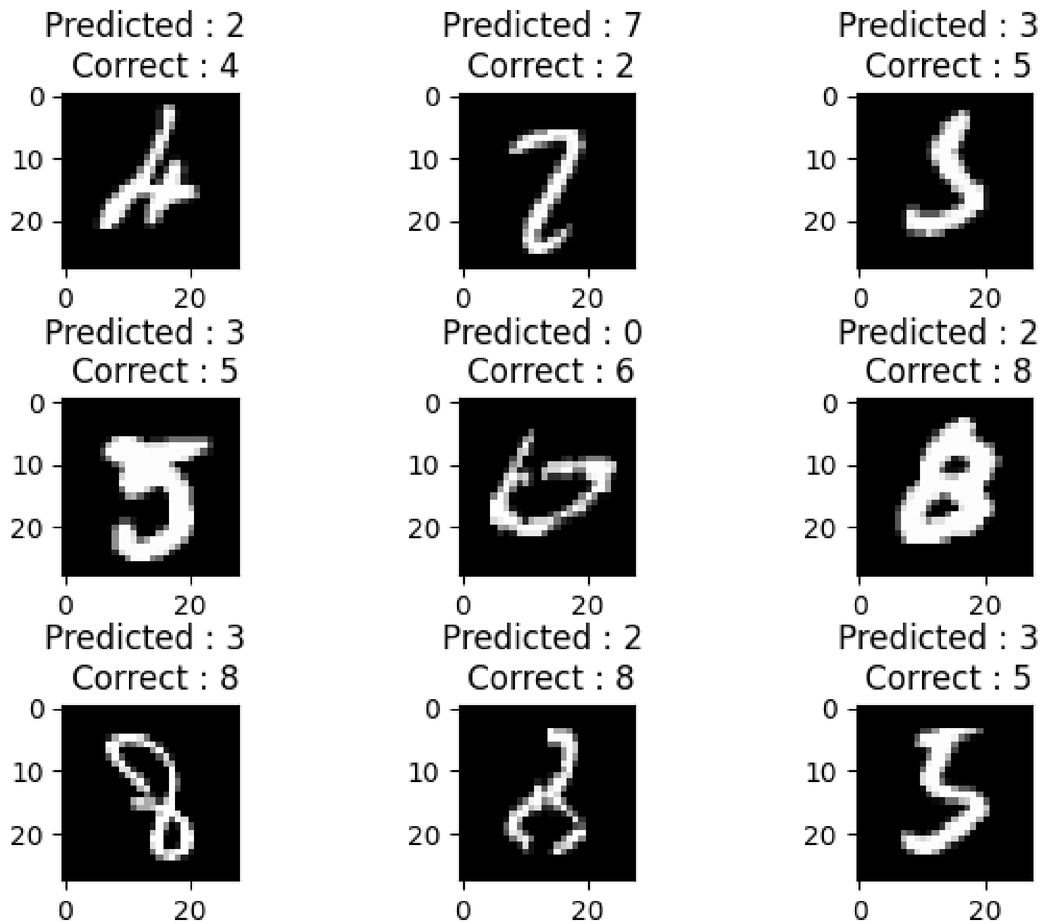
The graph depicts loss vs. batch index during training for each epoch. As training progresses, the loss values across batches consistently decrease — with the loss curve in each subsequent epoch moving closer to the x-axis (Loss = 0).

By Epoch 4 and Epoch 5, the loss curves become almost identical, indicating that the model has largely converged and is no longer improving significantly. This observation is the reason why the number of training epochs was set to 5 — adding more epochs would likely result in minimal improvement and increased training time without additional benefits.



The confusion matrix shows the performance of the model across all digit classes (0–9). As observed, the diagonal elements have significantly higher values than the off-diagonal elements. This indicates that the model correctly classifies most digits, validating its overall efficiency and high accuracy.

The most confused digit is ‘7’, which is likely due to the variation in handwriting styles. Some individuals write the digit ‘7’ with a horizontal line through the middle, while others do not. This inconsistency can lead the model to misinterpret '7’



This image displays a selection of digits that the model misclassified during testing. On closer inspection, many of these digits appear to be ambiguous or poorly written, making them difficult to interpret — even for humans.

Such challenging samples highlight the limitations of a feedforward neural network when dealing with heavily distorted or unclear handwriting, and suggest that future improvements (e.g., using convolutional neural networks or data augmentation) could help handle these edge cases more effectively.