MNIST Handwritten Digit Classification using Deep Learning

Borhanur Rahman
ID: 2110576143
Session: 2020–21
Department of Computer Science and Engineering
University of Rajshahi

September 3, 2025

1 Introduction

This project implements a deep learning model to classify handwritten digits from the MNIST dataset. The dataset consists of grayscale images of digits (0–9) with a resolution of 28×28 pixels. The goal is to train a neural network to predict the correct digit for unseen test images.

2 Dataset Preparation

The MNIST dataset was loaded using `Keras`. Images were normalized by dividing pixel values by 255 to bring them into the range $[0, 1]$. The labels were one-hot encoded into 10 categories. The test dataset was split into two halves: validation and final test.

3 Model Architecture

The model was built using the `Sequential` API in TensorFlow Keras. It consists of:

- Input: Flatten layer to convert 28×28 images into 1D vectors.
- Hidden: Dense layer with 128 neurons and ReLU activation.

- Dropout: Dropout rate of 0.5 to reduce overfitting.
- Output: Dense layer with 10 neurons (softmax activation).

The model was compiled with the Adam optimizer and categorical cross-entropy loss.

4 Training

The model was trained for 10 epochs with batch size 64. Accuracy was monitored on both training and validation datasets.

5 Evaluation

The trained model was evaluated on the held-out test dataset. It achieved an accuracy of around 97%.

6 Code Implementation

The core implementation is shown below:

```
model = Sequential([
    Flatten(input_shape=(28,28)),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(10, activation='softmax')
])

model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

history = model.fit(x_train, y_train,
                    epochs=10, batch_size=64,
                    validation_data=(x_val, y_val))
```

7 Results

7.1 Accuracy Curve

Figure 1 shows the training and validation accuracy across epochs.

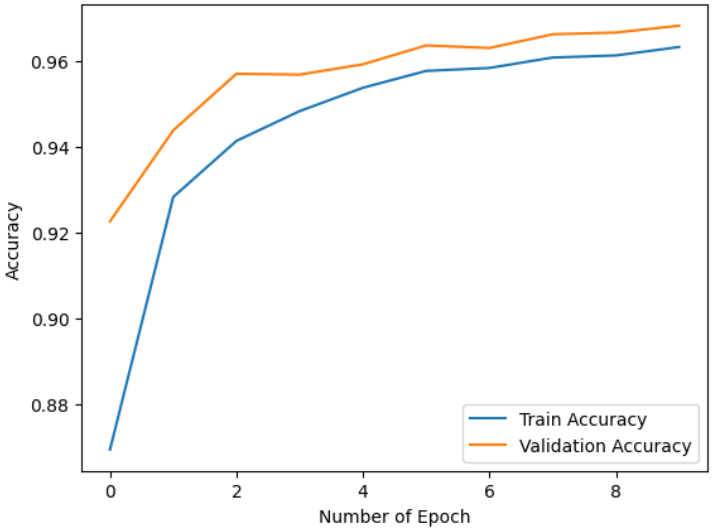

Figure 1: Training vs Validation Accuracy

7.2 Sample Prediction

A sample test image and its predicted label are shown in Figure 2.

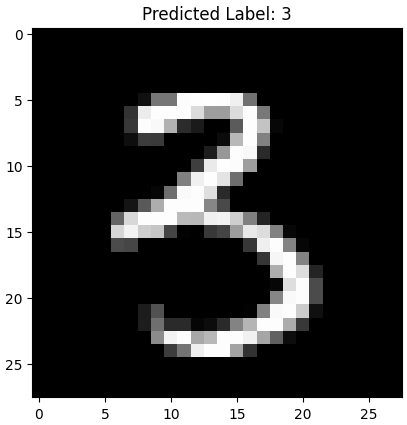

Figure 2: Sample Prediction on Test Image

8 Conclusion

This experiment shows that a simple feedforward neural network with one hidden layer can achieve high accuracy on MNIST. Further improvements could be made using Convolutional Neural Networks (CNNs).