Bangladesh University of Engineering and Technology

Machine Learning Sessional

CSE 472

Assignment 3

Function Approximation with Neural Network and Backpropagation

Report

Submitted By

Md. Huzzatun Ali

1905027

Date

November 8, 2024

Contents

1. Introduction 3
2. Running the Code 3
3. Data Loading and Preprocessing 4
4. Model Architectures 5
5. Training and Validation Metrics 5
6. Results for Different Models and Learning Rates 6
7. Best Model Selection 19
8. Best Model Performance Statistics 20
9. Best Model Performance on an Independent Test Set 21
10. Conclusion 21

1 Introduction

This report covers the training and evaluation of three neural network architectures on the Fashion-MNIST dataset using four learning rates. Performance is measured by training/validation loss, accuracy, and F1 scores. The model with the highest validation macro-F1 score is chosen and tested on an independent set.

2 Running the Code

To run the code provided, please follow these steps:

1. Install Necessary Libraries
2. Set Up the Dataset
3. Execute the Training Script
4. Save the Best Model
5. Test the Best Model on Independent Dataset

All these steps can be completed by running the 1905027.ipynb file following the order of the cells in the ipynb file.

3 Data Loading and Preprocessing

The dataset for this report is the Fashion-MNIST, a common benchmark for computer vision tasks. It is loaded and preprocessed using torchvision, with normalization applied to the input images. The images are flattened into 28×28 = 784-dimensional vectors for use in the fully connected neural networks.

```
# Data Loading and Preprocessing
def preprocess(dataset):
    images = []
    labels = []
    for image, label in dataset:
        image = np.array(image).flatten() / 255.0  # Flatten and normalize to [0, 1]
        label_onehot = np.zeros(10)  # One-hot encoding
        label_onehot[label] = 1
        images.append(image)
        labels.append(label_onehot)

    # Convert lists to NumPy arrays
    images = np.array(images)
    labels = np.array(labels)
    return images, labels
✓ 0.0s
```

```
# Load Data Function
def load_data():
    train_dataset = ds.FashionMNIST(root='./data', train=True, transform=transforms.ToTensor(), download=True)
    test_dataset = ds.FashionMNIST(root='./data', train=False, transform=transforms.ToTensor(), download=True)

    X_train, Y_train = preprocess(train_dataset)
    X_test, Y_test = preprocess(test_dataset)

    return X_train, Y_train, X_test, Y_test
✓ 0.0s
```

4 Model Architectures

We implemented three different neural network architectures to compare performance:

1. **Model 1:** A simple two-layer network with 256 neurons in the hidden layer, followed by a ReLU activation and a 10-class Softmax output layer.
2. **Model 2:** A deeper network with two hidden layers, first with 128 neurons and then 256 neurons, both using ReLU activation, followed by a Softmax output.
3. **Model 3:** The most complex model with three hidden layers, progressively increasing in size (64, 128, and 256 neurons), each followed by ReLU, and a Softmax output layer for classification.

Each model aims to classify the 28x28 input images into 10 categories.

5 Training and Validation Metrics

For each training epoch, the following metrics are recorded:

- Train Loss
- Train Accuracy
- Val Loss (Validation Loss)
- Val Accuracy (Validation Accuracy)

For each learning rate of a model, the following metrics are recorded:

- Accuracy
- Validation Macro-F1 Score

6 Results for Different Models and Learning Rates

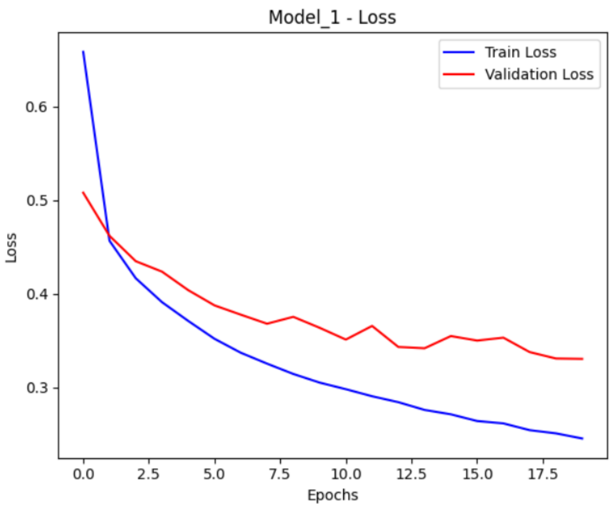
For each model, we tested four learning rates: 0.005, 0.001, 0.0005, and 0.0001. Below are the results showing the effect of these learning rates on performance metrics.

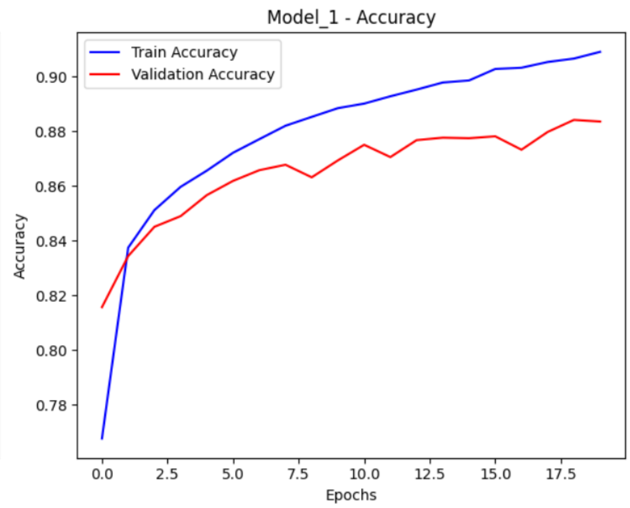
The graphs show:

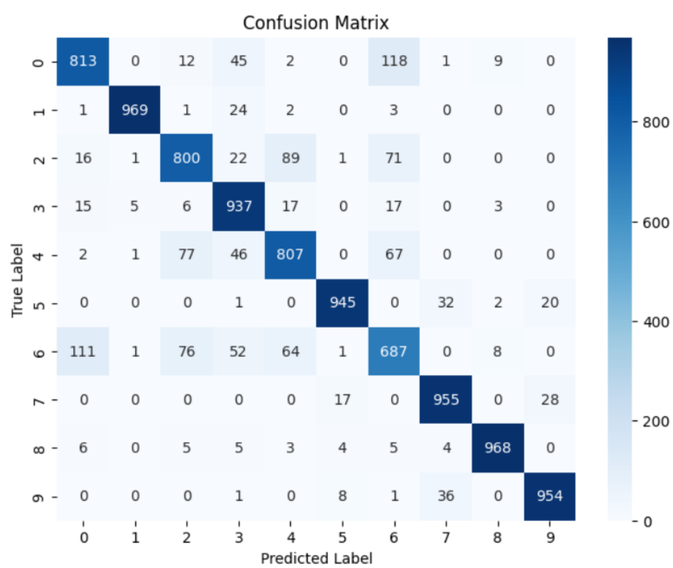
- Training and validation loss against number of epochs
- Training and validation accuracy against number of epochs
- Confusion matrix

Model 1 (One Layer, Hidden Dimension = 256)

Learning Rate = 0.005





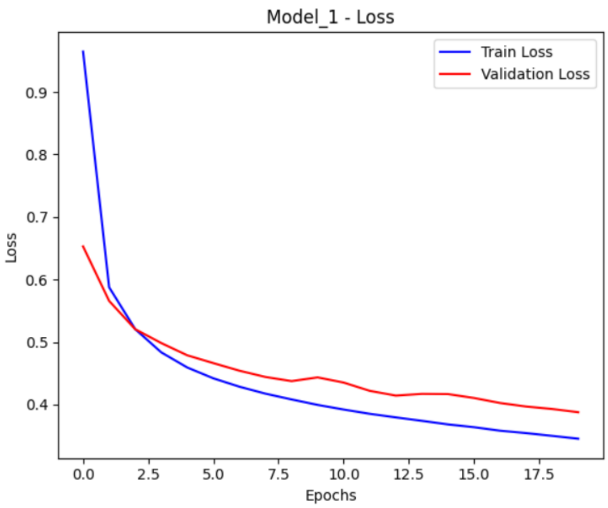


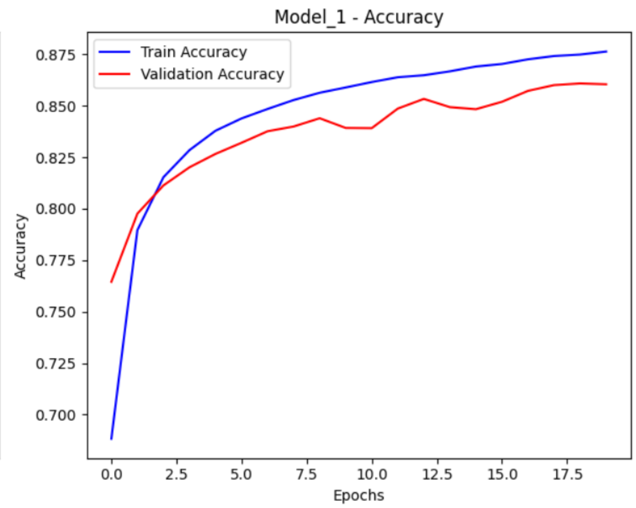
True \ Predicted	0	1	2	3	4	5	6	7	8	9
0	813	0	12	45	2	0	118	1	9	0
1	1	969	1	24	2	0	3	0	0	0
2	16	1	800	22	89	1	71	0	0	0
3	15	5	6	937	17	0	17	0	3	0
4	2	1	77	46	807	0	67	0	0	0
5	0	0	0	1	0	945	0	32	2	20
6	111	1	76	52	64	1	687	0	8	0
7	0	0	0	0	0	17	0	955	0	28
8	6	0	5	5	3	4	5	4	968	0
9	0	0	0	1	0	8	1	36	0	954

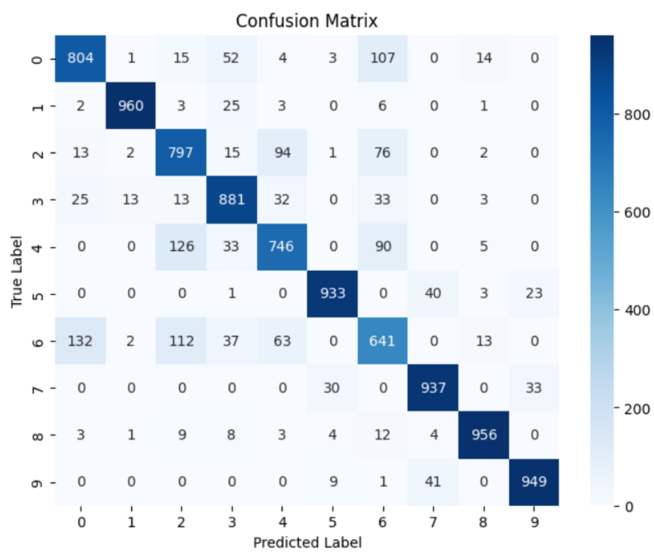
Accuracy: 88.35%

Validation Macro-F1 Score: 0.8832

Learning Rate = 0.001



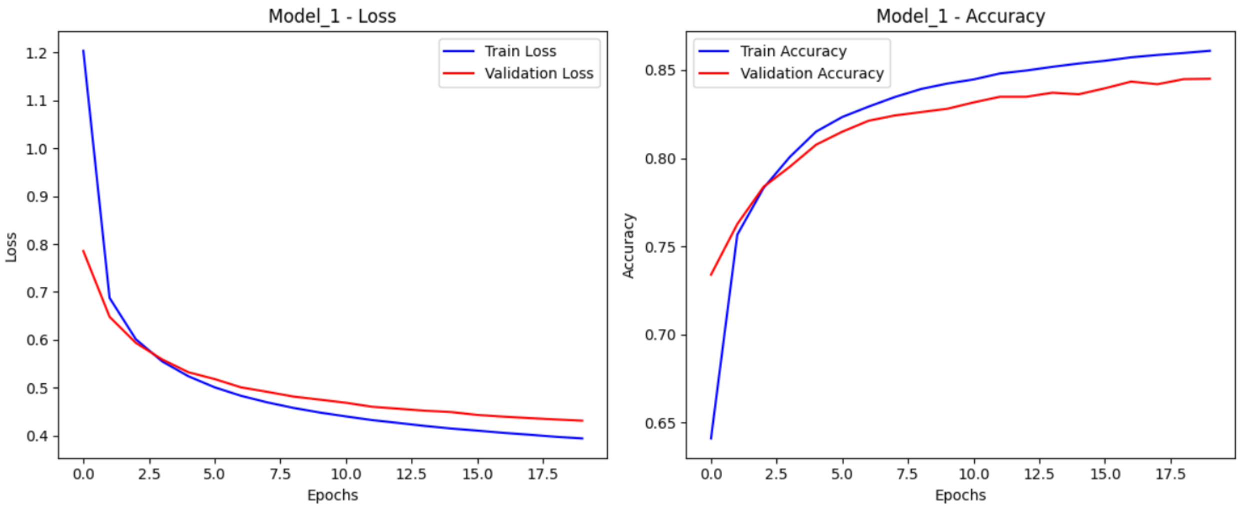


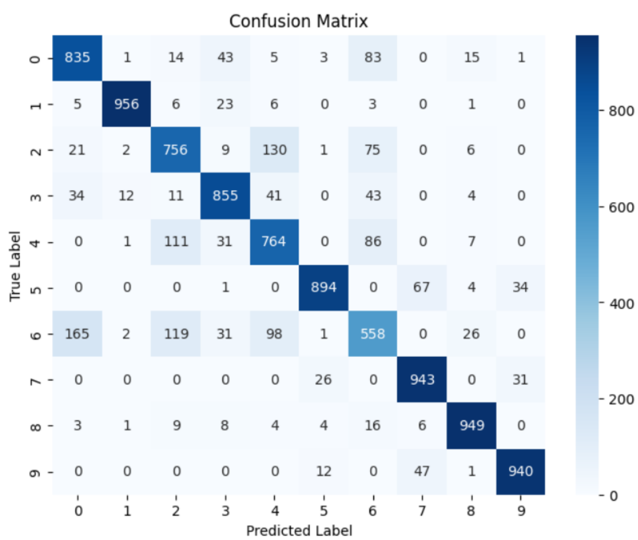


Accuracy: 86.04%

Validation Macro-F1 Score: 0.8602

Learning Rate = 0.0005

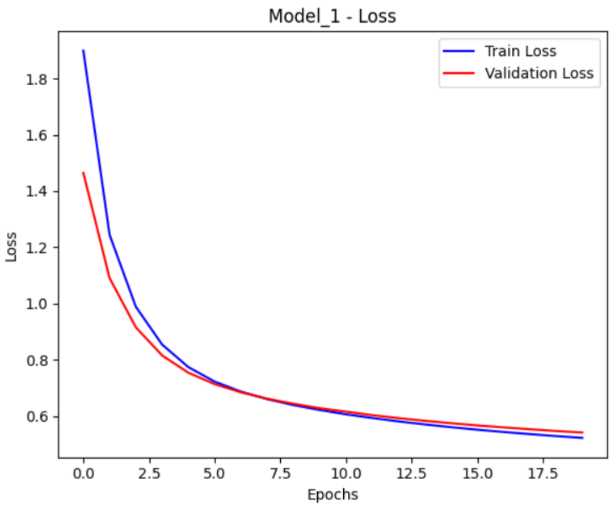


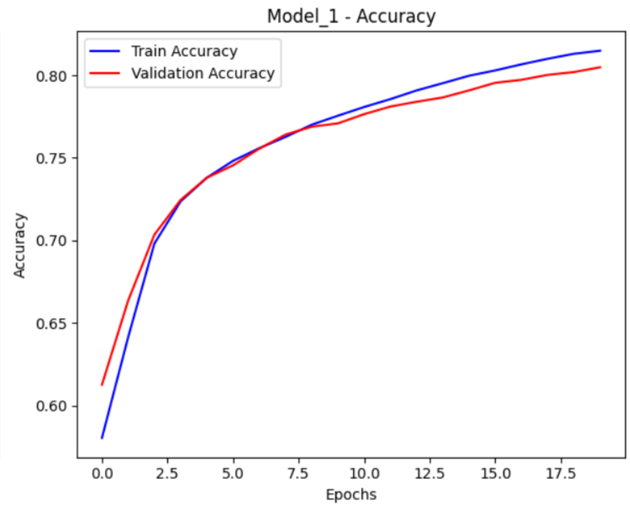


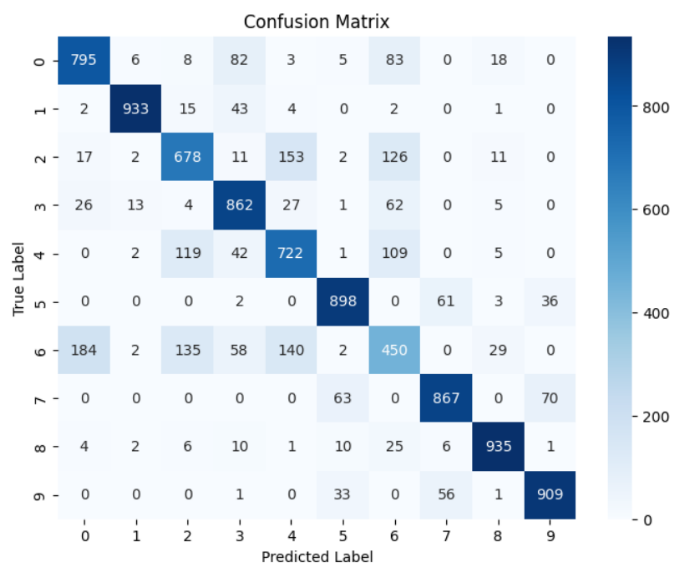
Accuracy: 84.50%

Validation Macro-F1 Score: 0.8439

Learning Rate = 0.0001





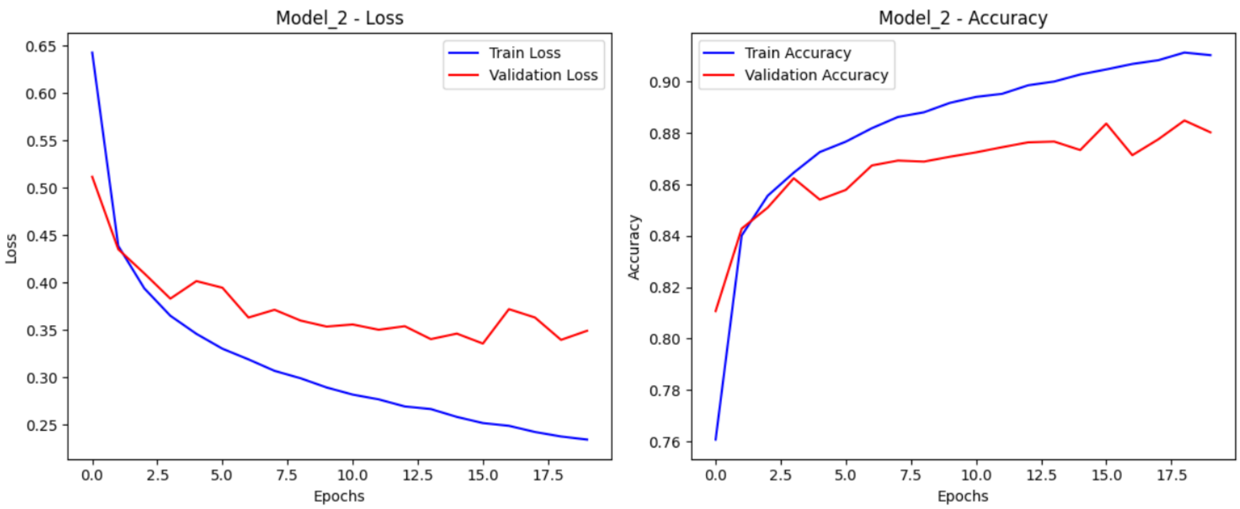


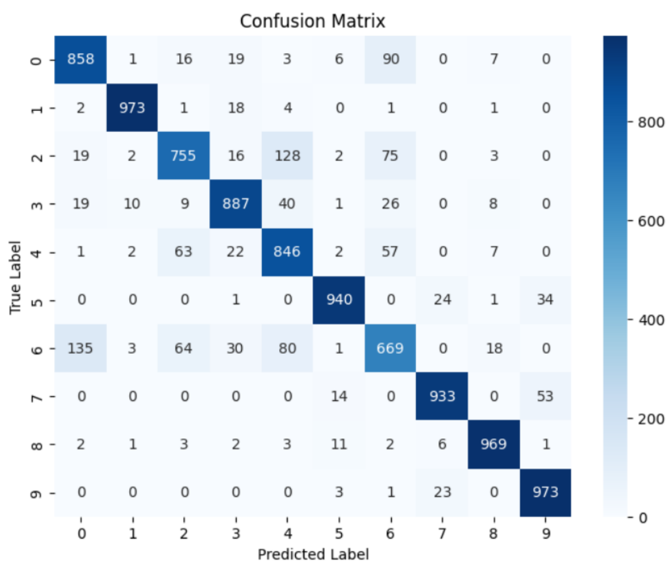
Accuracy: 80.49%

Validation Macro-F1 Score: 0.8028

Model 2 (Two Layer, Hidden Dimension-1 = 128, Hidden Dimension-2 = 256)

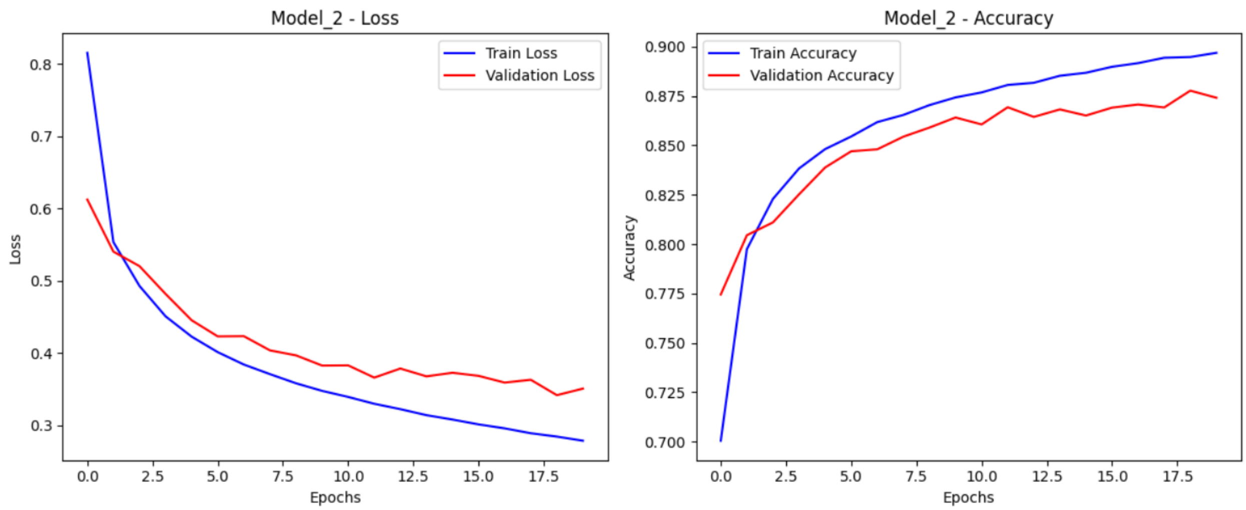
Learning Rate = 0.005

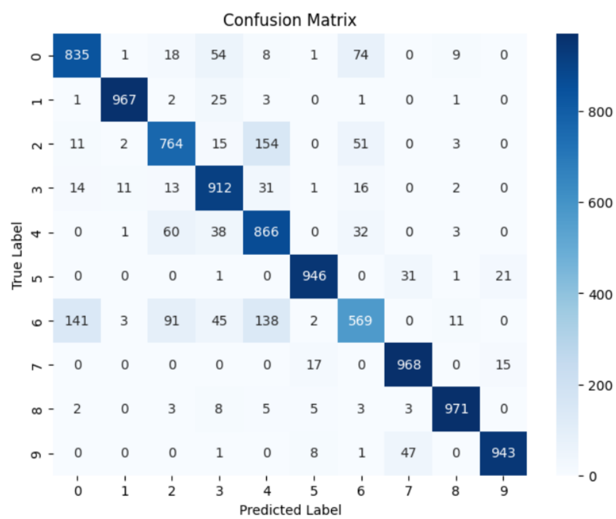




Accuracy: 88.03% Validation Macro-F1 Score: 0.8795

Learning Rate = 0.001

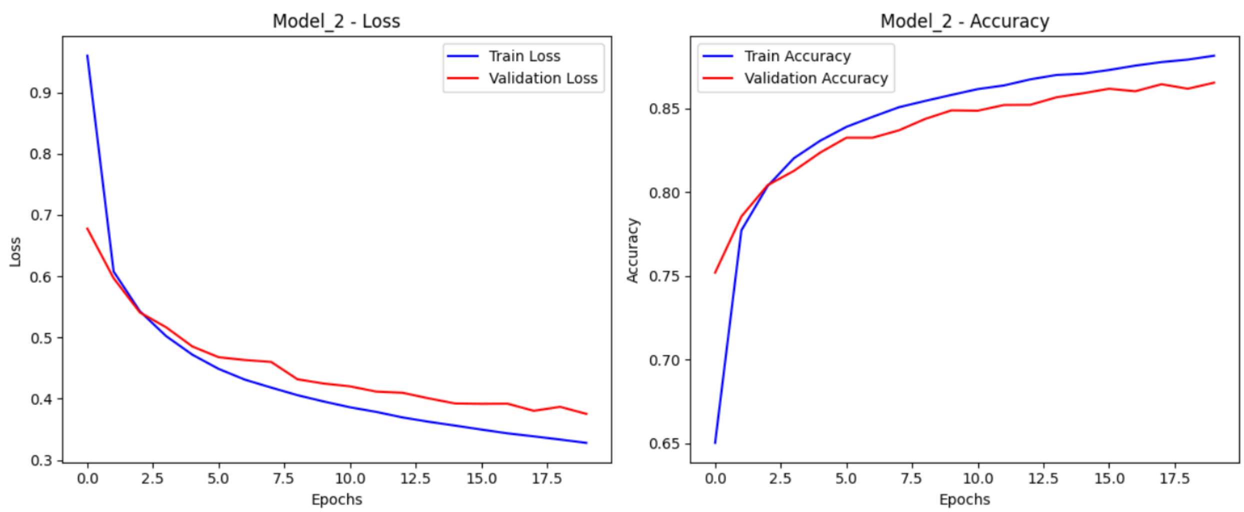


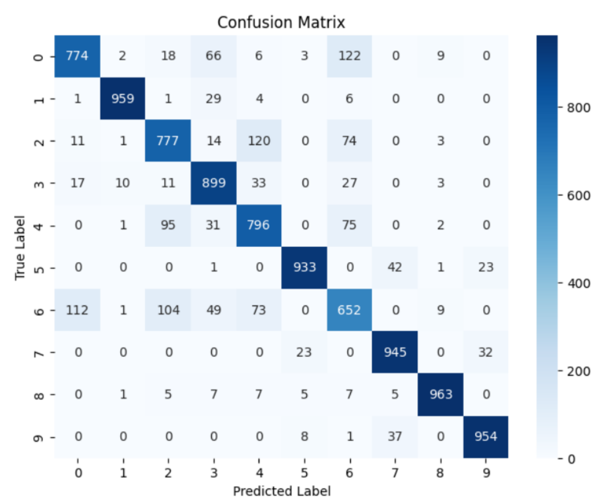


Accuracy: 87.41%

Validation Macro-F1 Score: 0.8721

Learning Rate = 0.0005

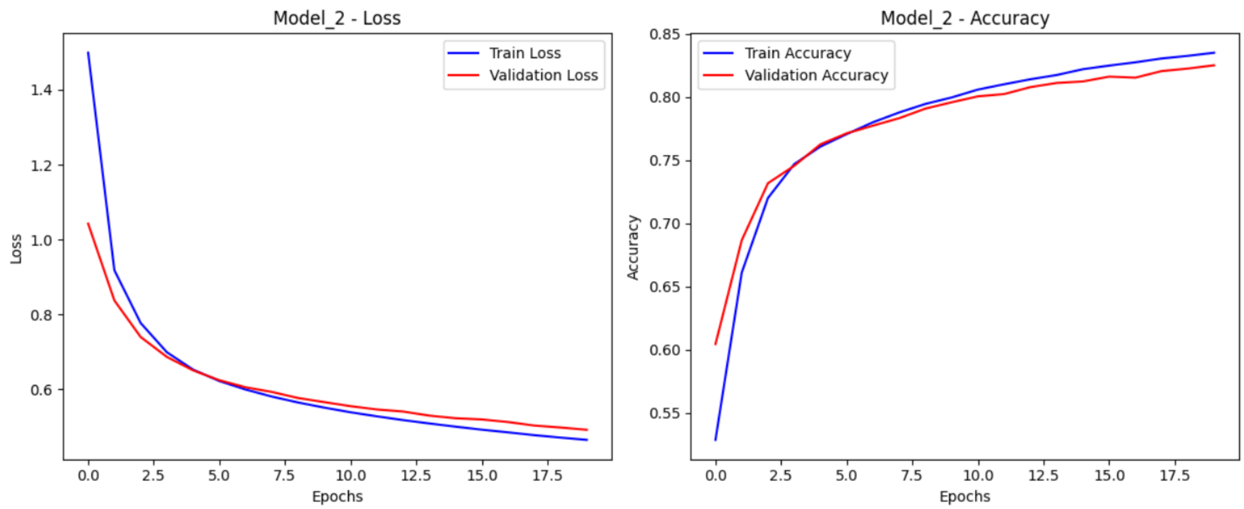


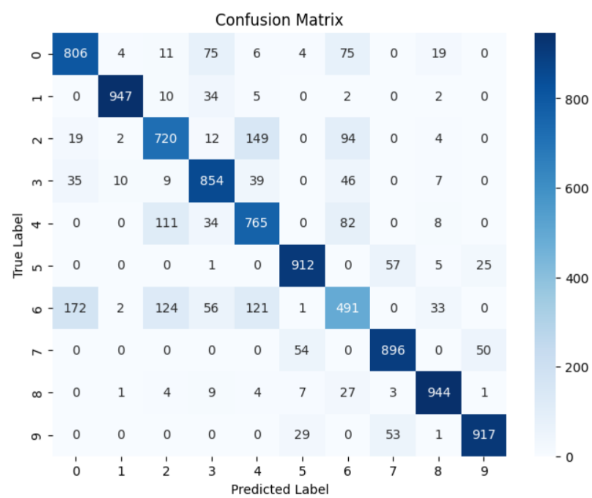


Accuracy: 86.52%

Validation Macro-F1 Score: 0.8650

Learning Rate = 0.0001





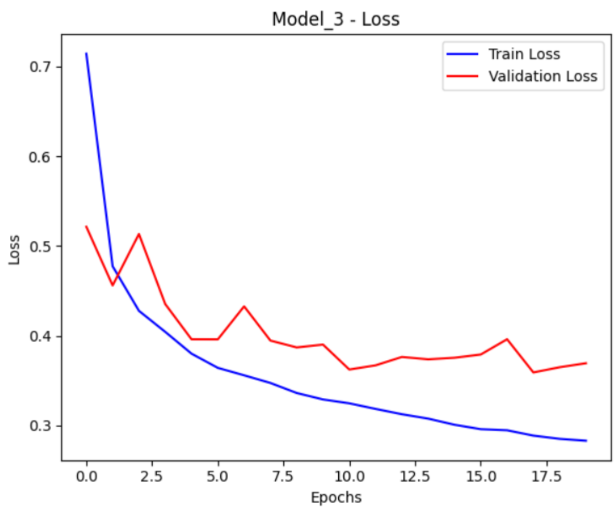
Accuracy: 82.52%

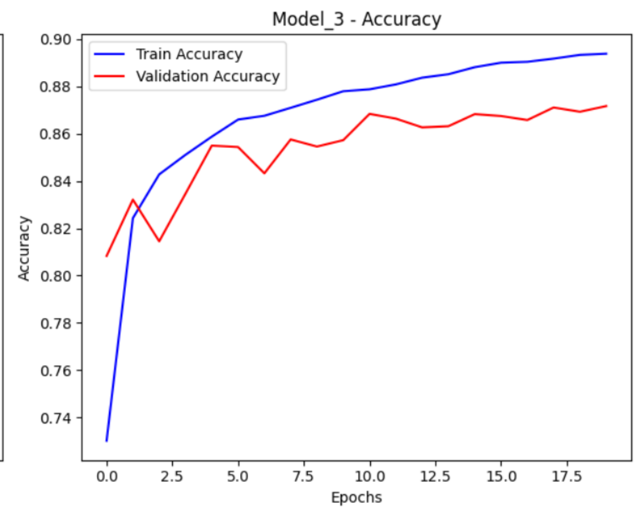
Validation Macro-F1 Score: 0.8231

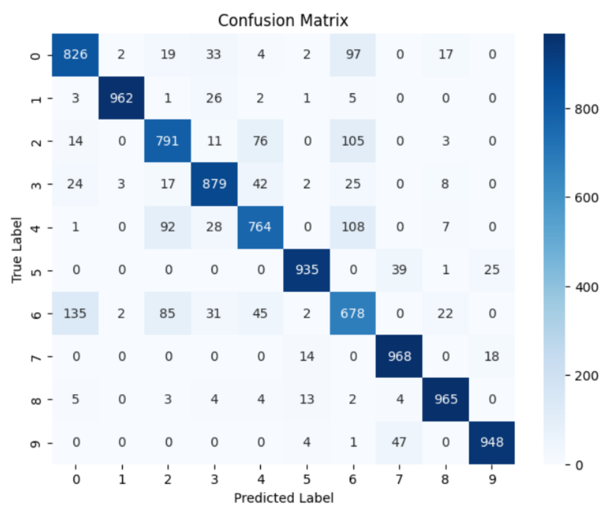
Model 3 (Three Layer)

(Hidden Dimension-1 = 64, Hidden Dimension-2 = 128, Hidden Dimension-3 = 256)

Learning Rate = 0.005



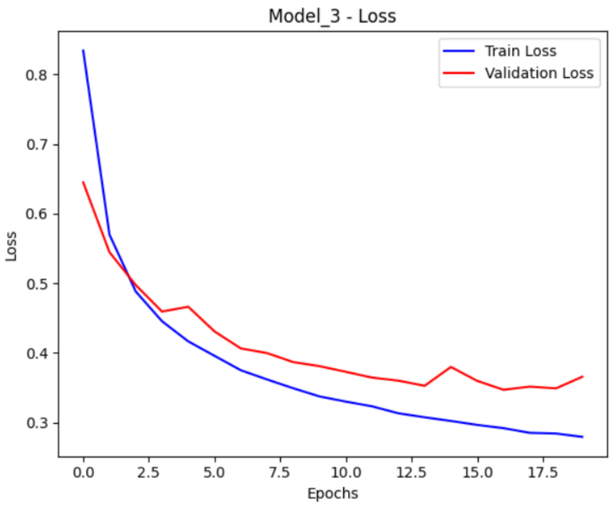


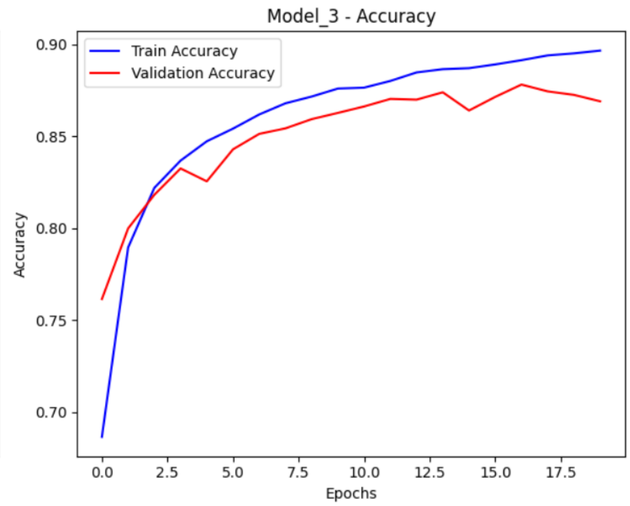


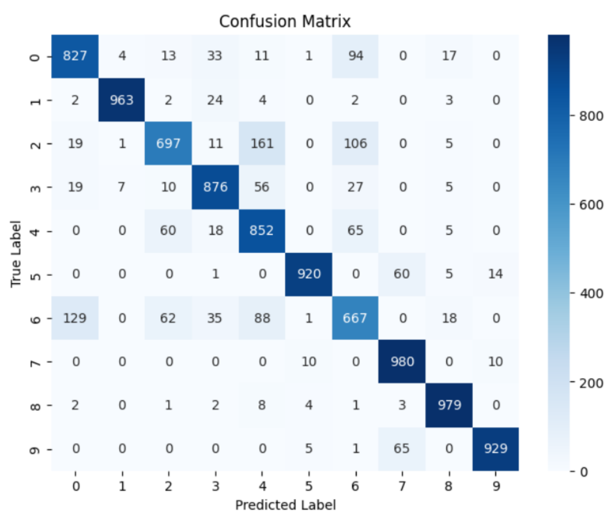
Accuracy: 87.16%

Validation Macro-F1 Score: 0.8716

Learning Rate = 0.001

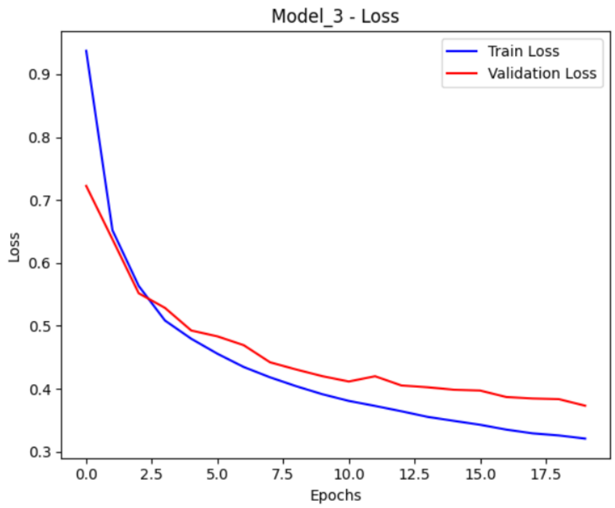


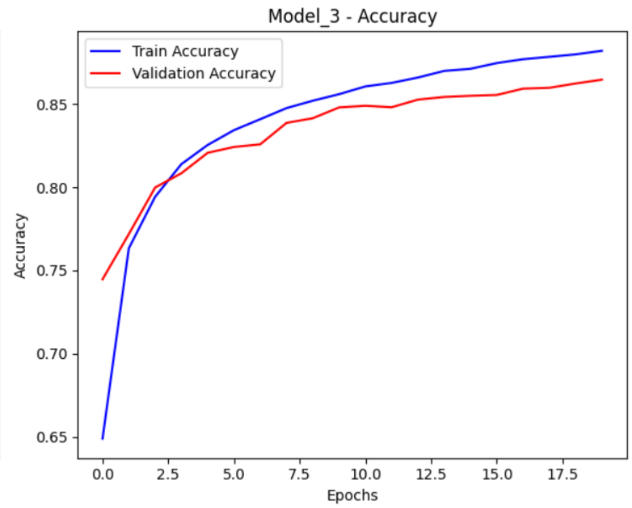


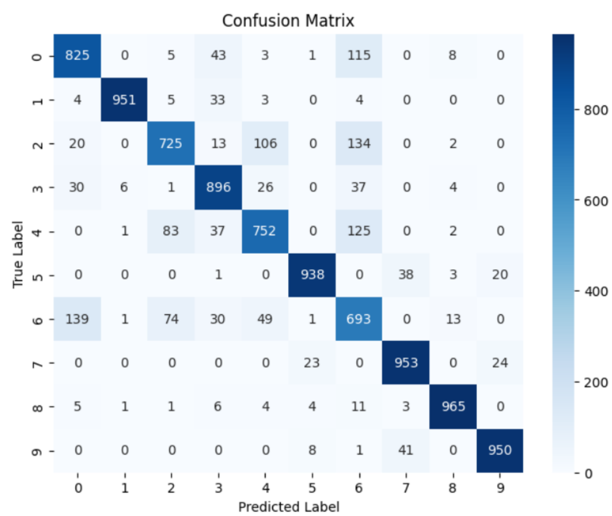


Accuracy: 86.90% Validation Macro-F1 Score: 0.8686

Learning Rate = 0.0005



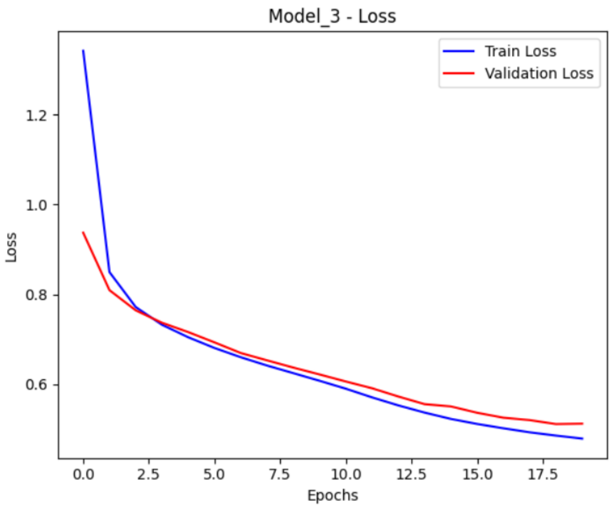


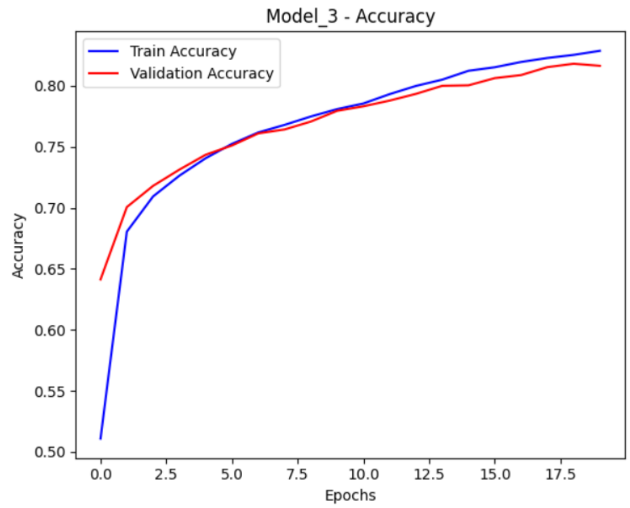


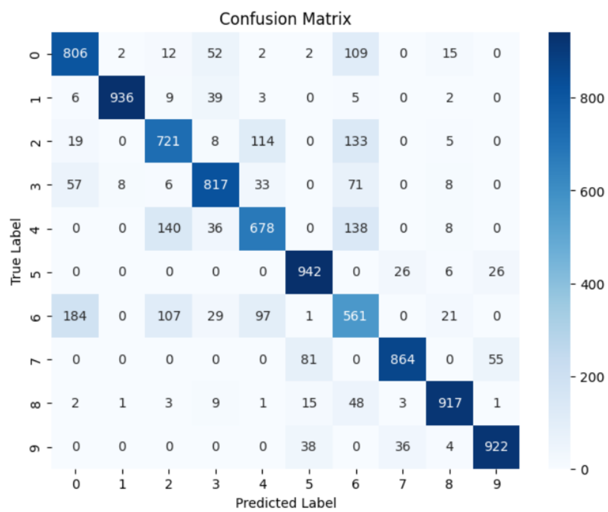
Accuracy: 86.48%

Validation Macro-F1 Score: 0.8656

Learning Rate = 0.0001







Accuracy: 81.64%

Validation Macro-F1 Score: 0.8174

7 Best Model Selection

The process tracks the model with the highest F1 score during training. If a model's F1 score surpasses the previous best, the model, its name, and the learning rate are updated to ensure the top-performing model is selected.

Initialization:

```
best_score = 0.0
best_model = None
best_model_name = None
best_model_learning_rate = None
```

Update:

```
if f1 > best_score:
    best_score = f1
    best_model = model
    best_model_name = model_name
    best_model_learning_rate = lr
```

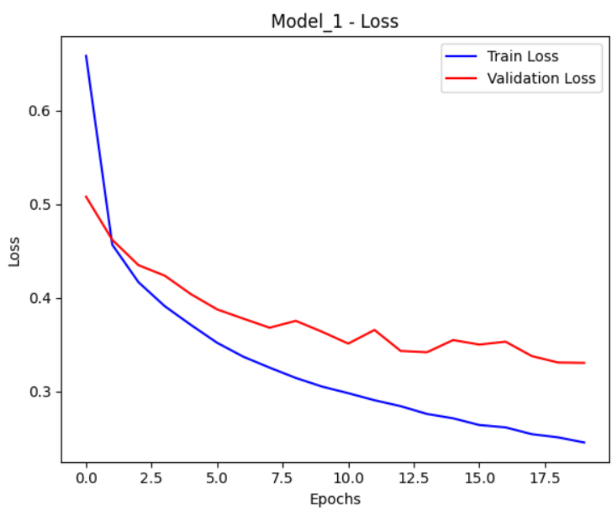
8 Best Model Performance Statistics

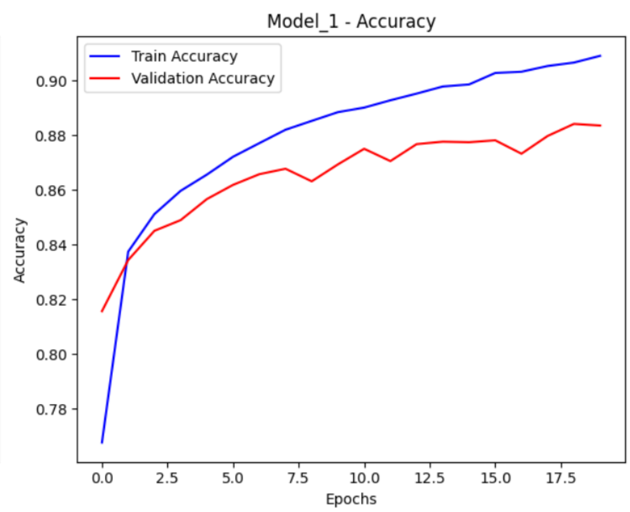
The best model is **model-1 (One Layer, Hidden Dimension = 256)** with a **learning rate of 0.005**. Its **validation micro-F1 score is 0.8832.**

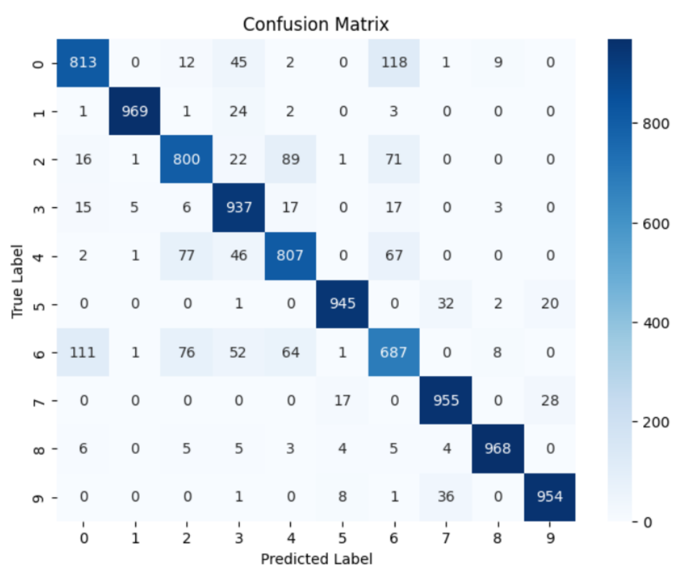
```
Best_Model: Model_1, Learning_Rate: 0.005
```

Model 1 (One Layer, Hidden Dimension = 256)

Learning Rate = 0.005

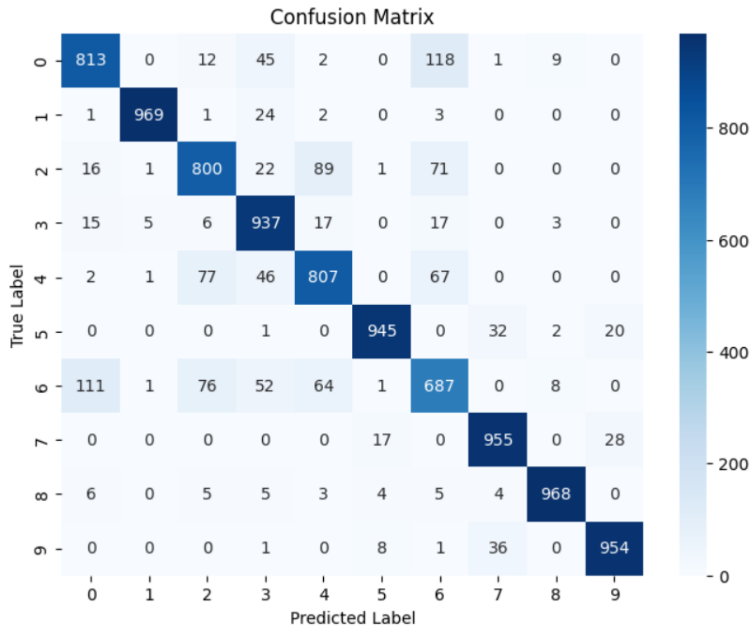






Accuracy: 88.35% Validation Macro-F1 Score: 0.8832

9 Best Model Performance on an Independent Test Set




```
Accuracy: 88.35%, F1 Score: 0.8832
```

10 Conclusion

This report compares three neural network models across four learning rates. **model-1 (One Layer, Hidden Dimension = 256)** with a **learning rate of 0.005** performed the best, **achieving a validation macro-F1 score of 0.8832**. Future improvements could focus on further tuning, testing different architectures, increasing number of epochs and exploring alternative optimization techniques.