CSE 472
Machine Learning Sessional
Assignment 3 Report
Feed-forward Neural Networks

Name: A. H. M. Osama Haque
ID: 1805002
Section: A
Date of Submission: 03.01.2024

Hyperparameters:

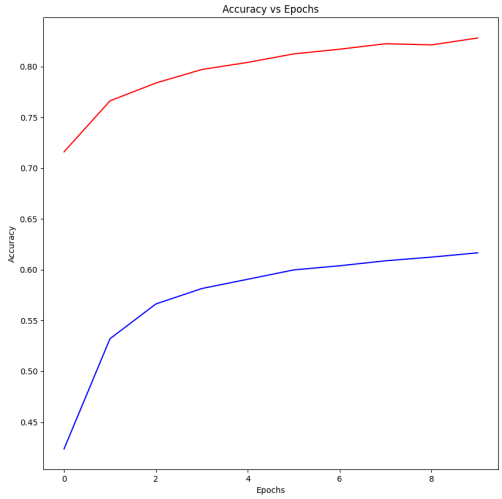
1. **Learning rates:** [0.005, 0.001, 0.0005, 0.0001]
2. **Batch_size:** 1024
3. **Epochs:** 10
4. **Optimizer:** Adam
5. **Initialization:** Xavier

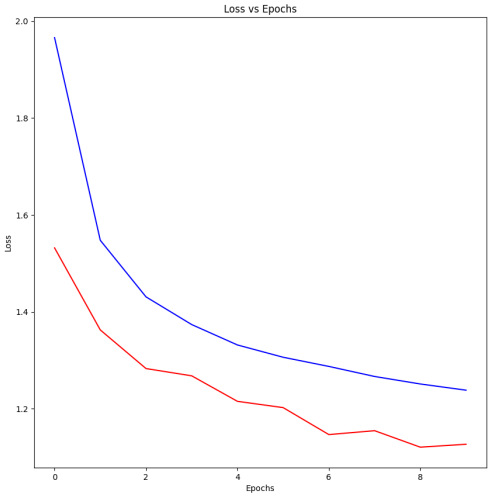
Model 1:

```
network = [
    DenseLayer(784, 256),
    ReLULayer(),
    dropout(0.15),
    DenseLayer(256, 26),
    SoftmaxLayer()
]
```

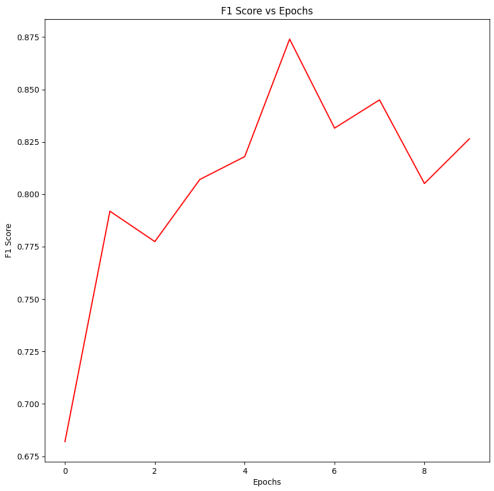
Learning Rate: 5e-3

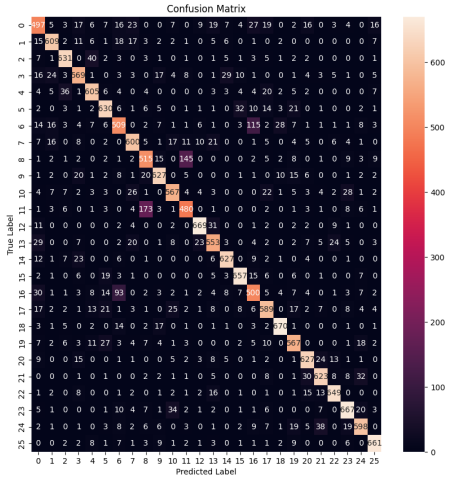
Accuracy vs Epoch & Loss vs Epoch:





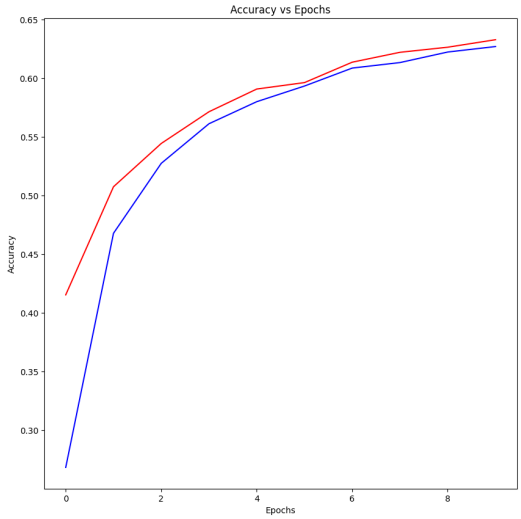
Validation F1 vs epoch & Confusion Matrix:

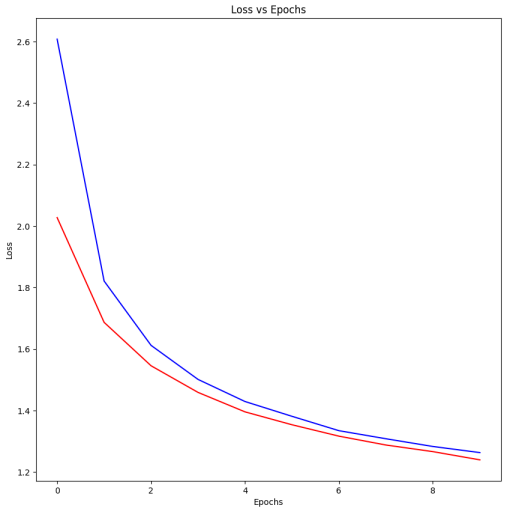




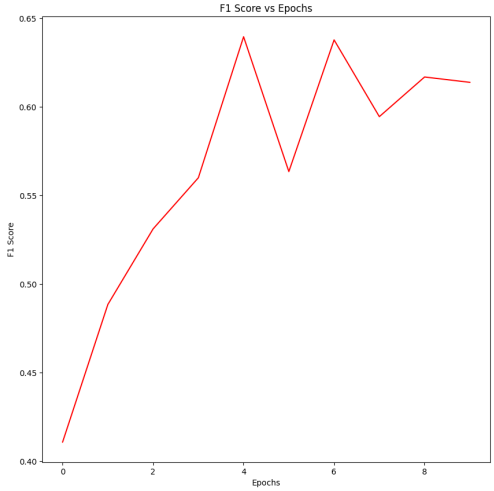
Learning Rate: 1e-3

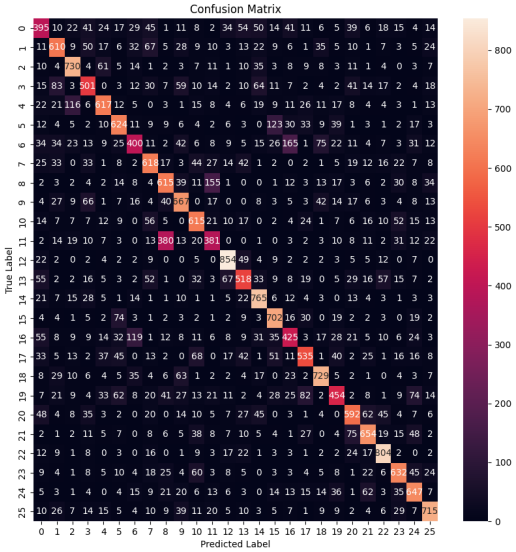
Accuracy vs Epoch & Loss vs Epoch:





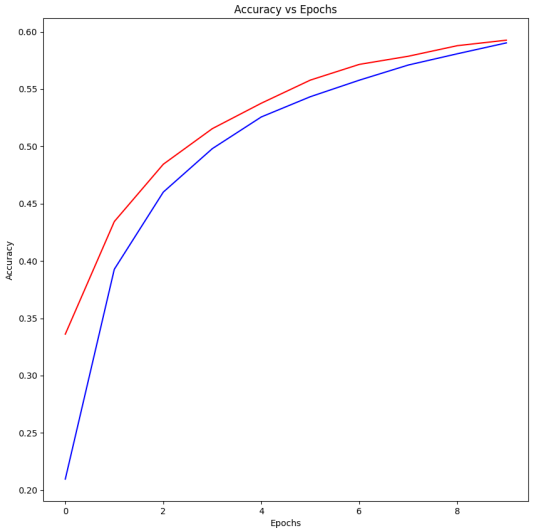
Validation F1 vs epoch & Confusion Matrix

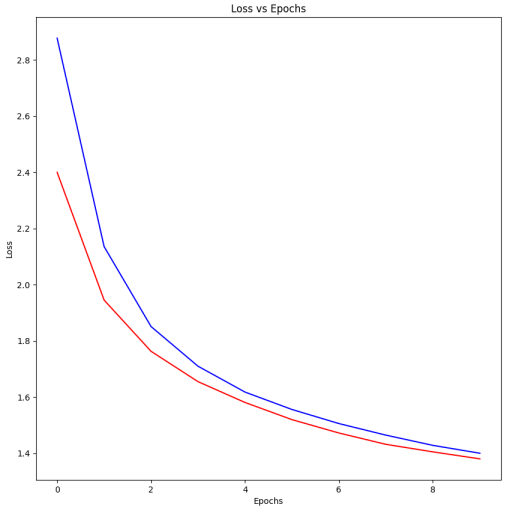




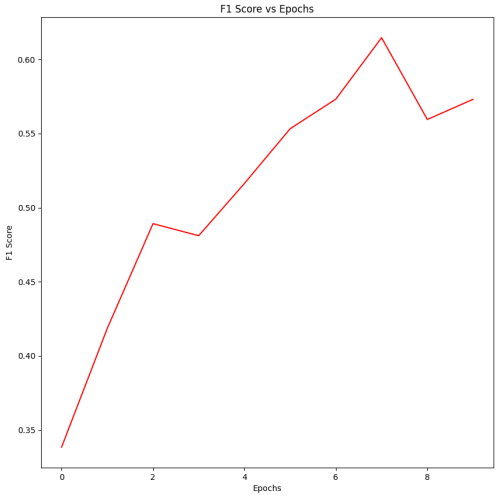
Learning Rate: 5e-4

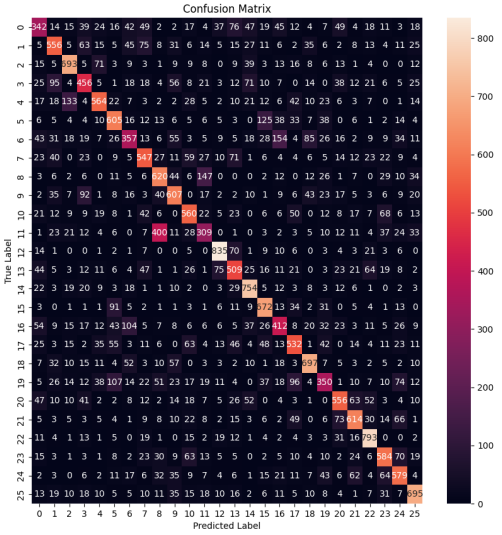
Accuracy vs Epoch & Loss vs Epoch:





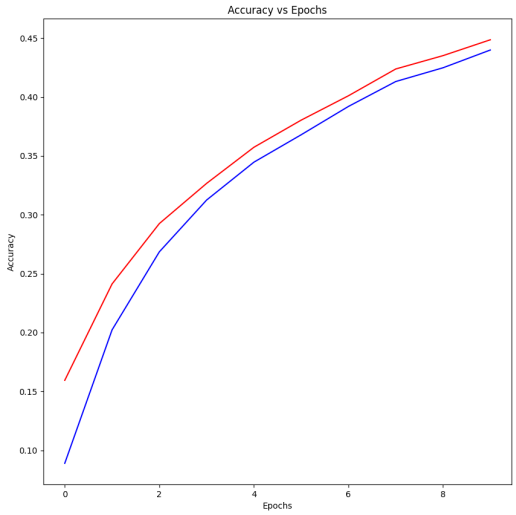
Validation F1 vs epoch & Confusion Matrix

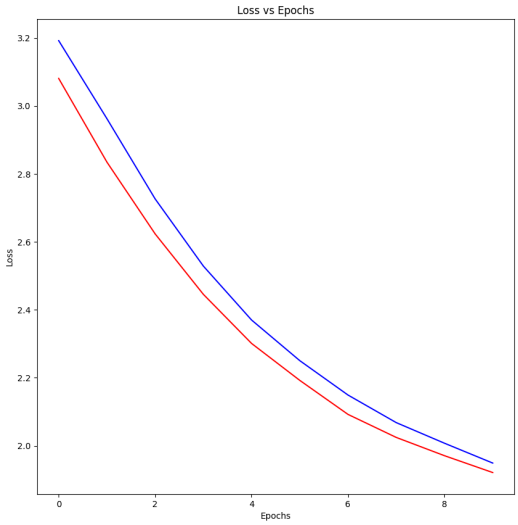




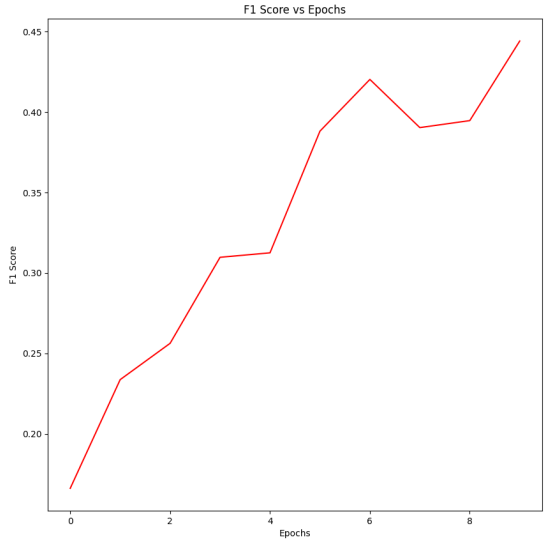
Learning Rate: 1e-4

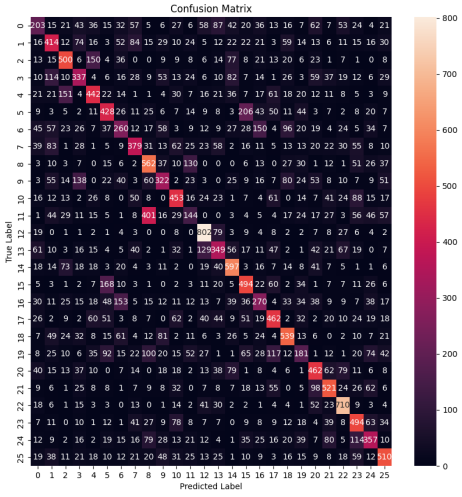
Accuracy vs Epoch & Loss vs Epoch:





Validation F1 vs epoch & Confusion Matrix



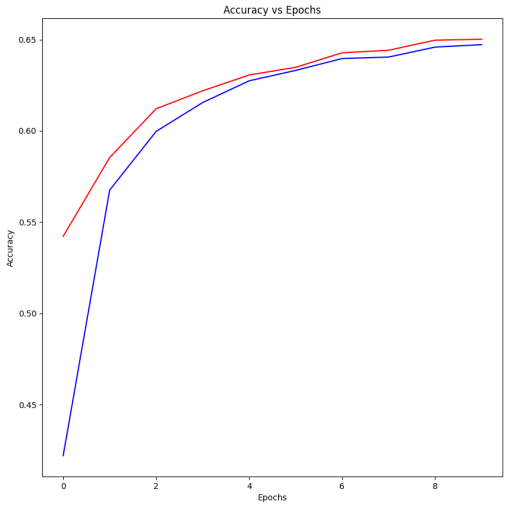


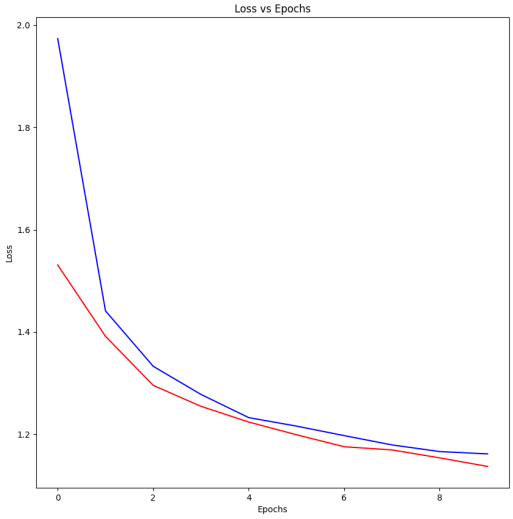
Model 2:

```
network2 = [
    DenseLayer(784, 512),
    ReLULayer(),
    dropout(0.15),
    DenseLayer(512, 26),
    SoftmaxLayer()
]
```

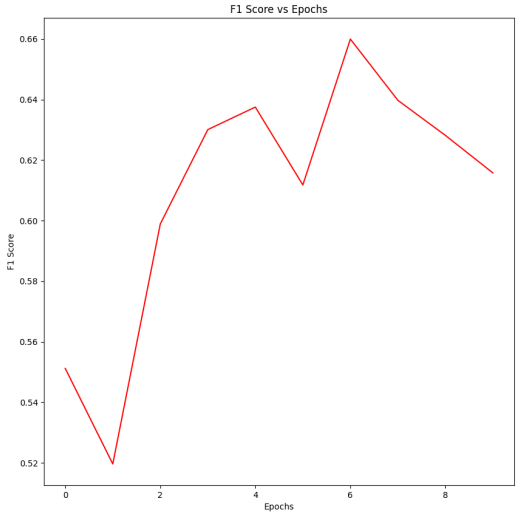
Learning Rate: 5e-3

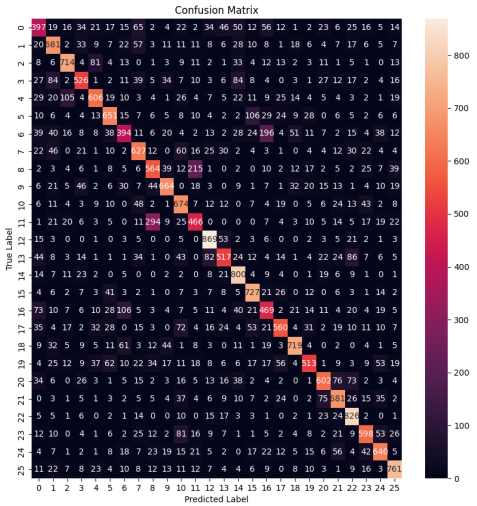
Accuracy vs Epoch & Loss vs Epoch:





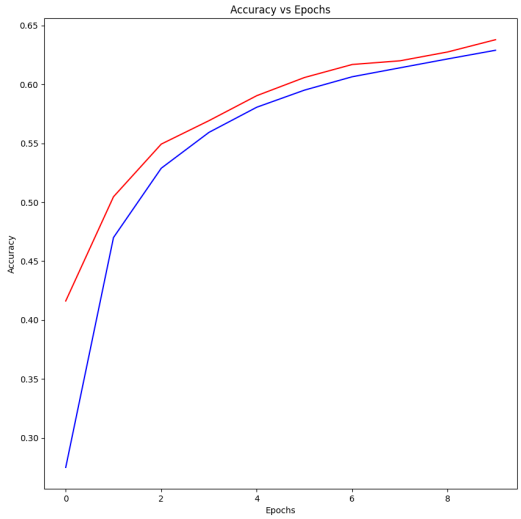
Validation F1 vs epoch & Confusion Matrix:

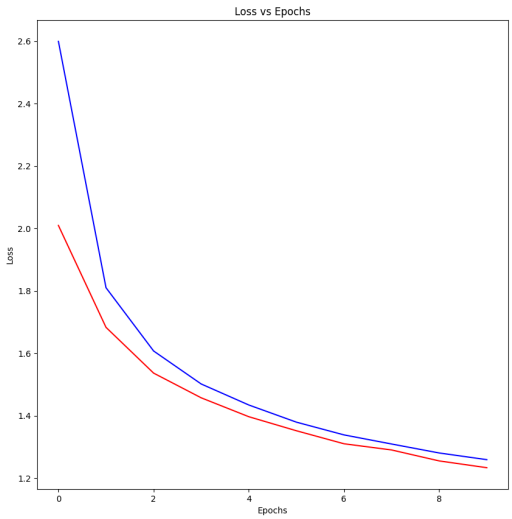




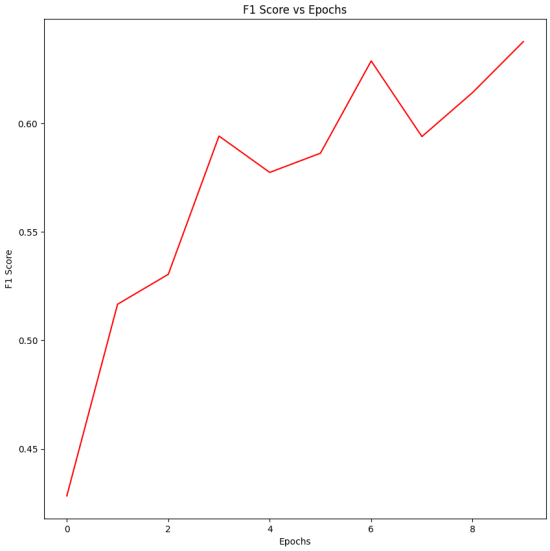
Learning Rate: 1e-3

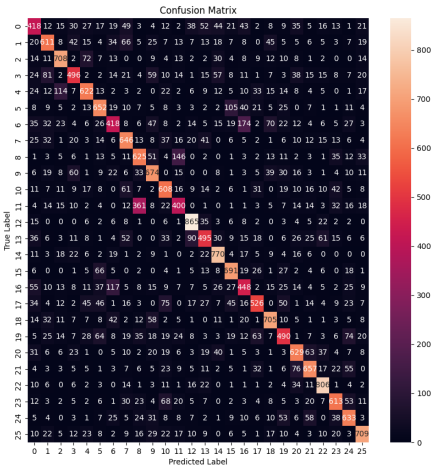
Accuracy vs Epoch & Loss vs Epoch:





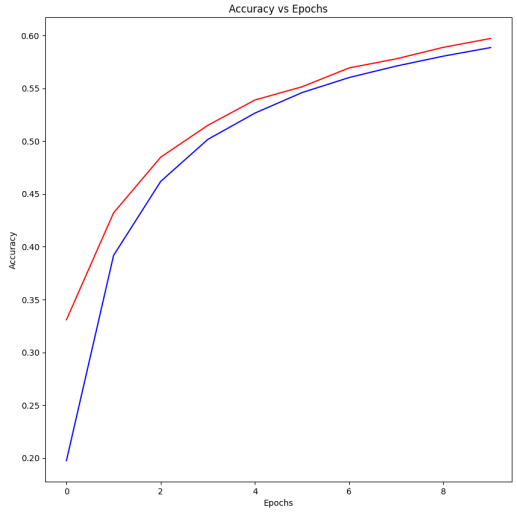
Validation F1 vs epoch & Confusion Matrix

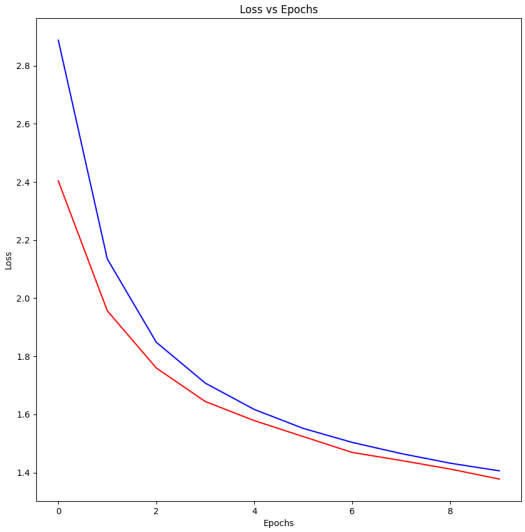




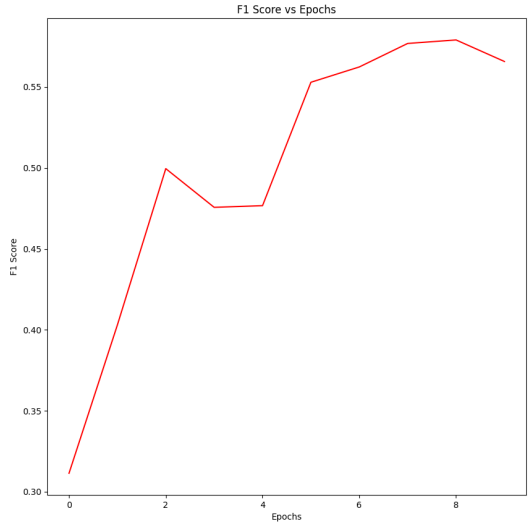
Learning Rate: 5e-4

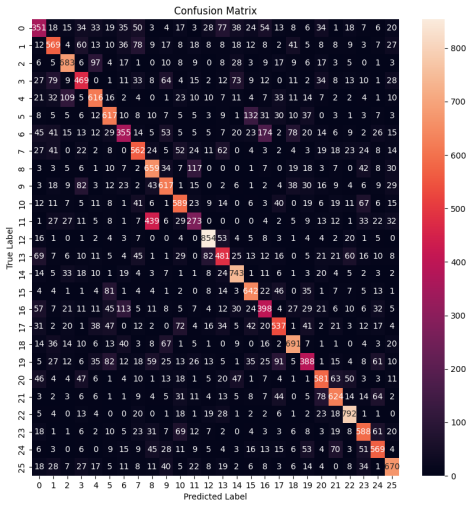
Accuracy vs Epoch & Loss vs Epoch:





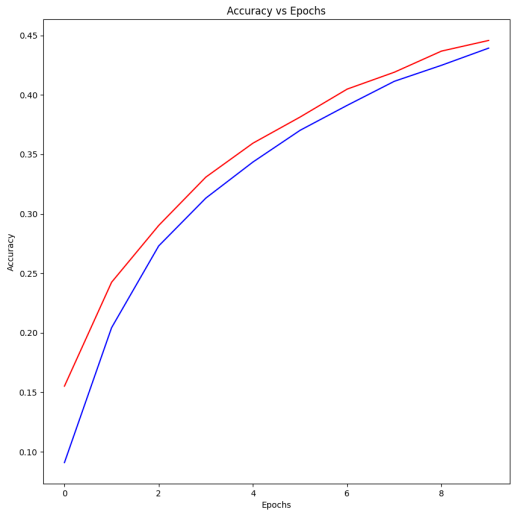
Validation F1 vs epoch & Confusion Matrix

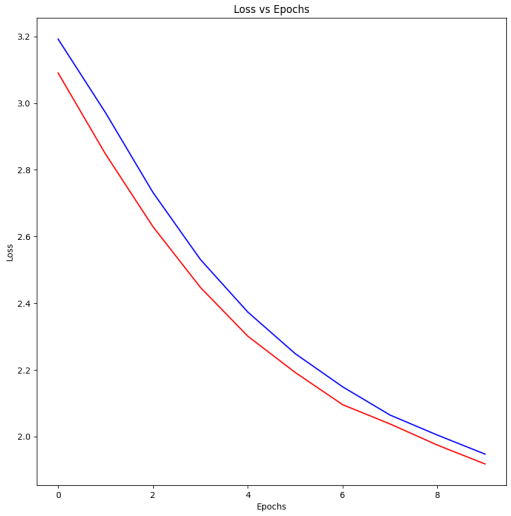




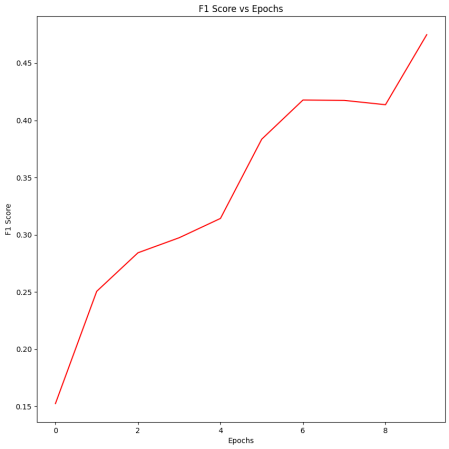
Learning Rate: 1e-4

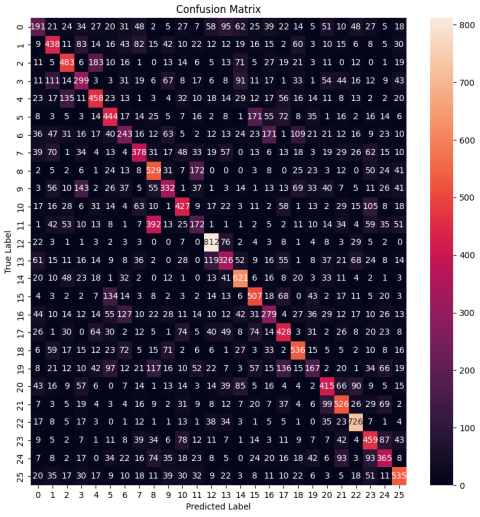
Accuracy vs Epoch & Loss vs Epoch:





Validation F1 vs epoch & Confusion Matrix



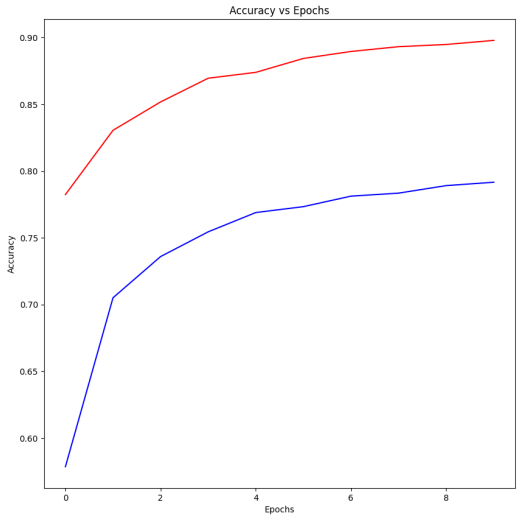


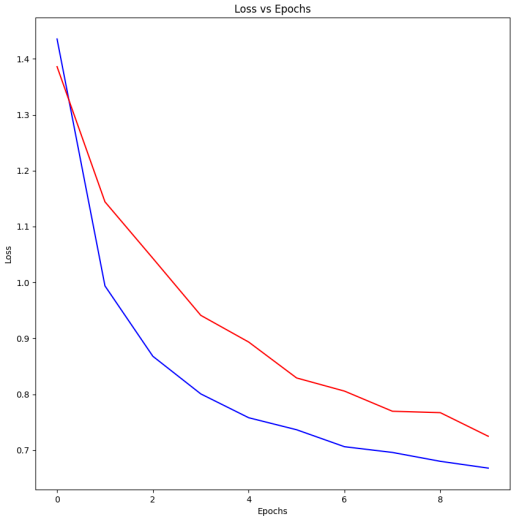
Model 3: (Best model)

```
network3 = [
    DenseLayer(784, 1024),
    ReLULayer(),
    dropout(0.15),
    DenseLayer(1024, 26),
    SoftmaxLayer()
]
```

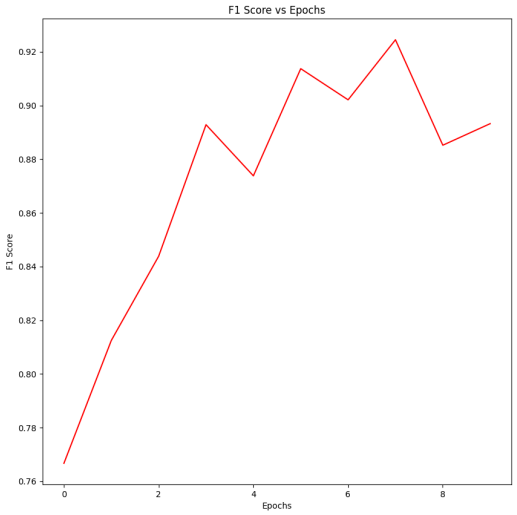
Learning Rate: 5e-3 (Best learning rate)

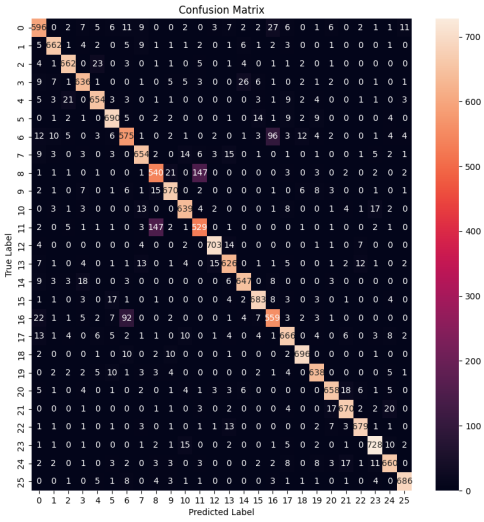
Accuracy vs Epoch & Loss vs Epoch:





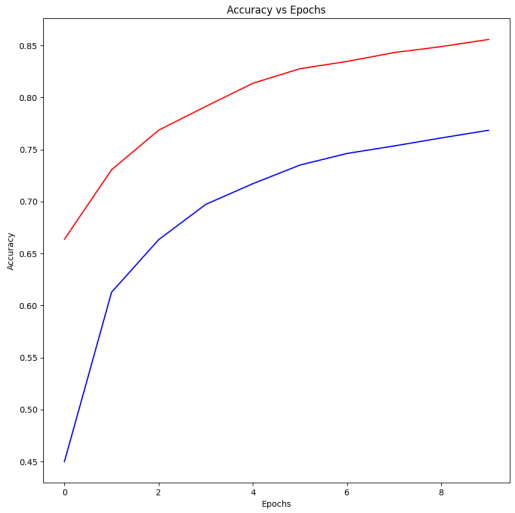
Validation F1 vs epoch & Confusion Matrix:

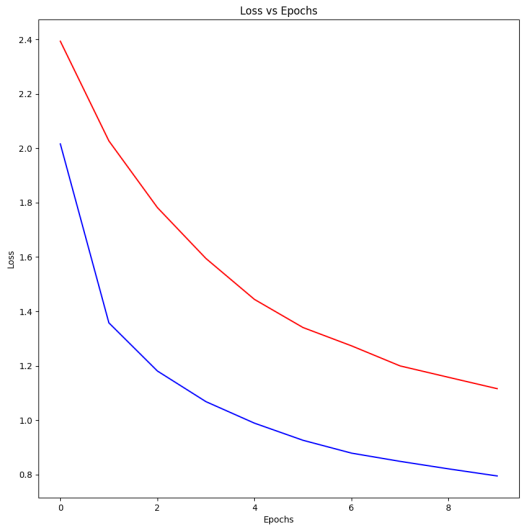




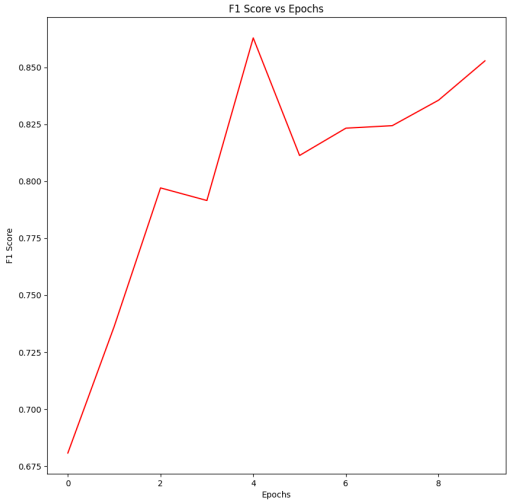
Learning Rate: 1e-3

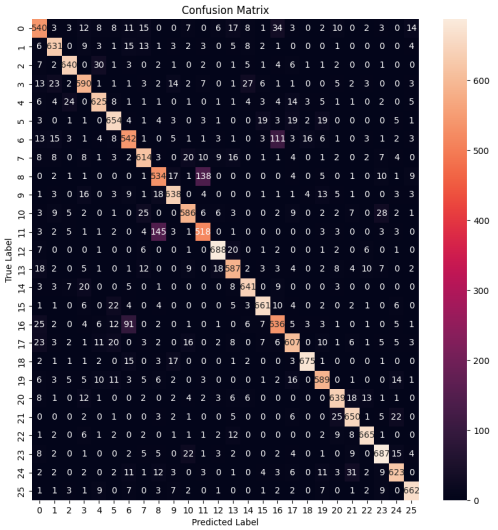
Accuracy vs Epoch & Loss vs Epoch:





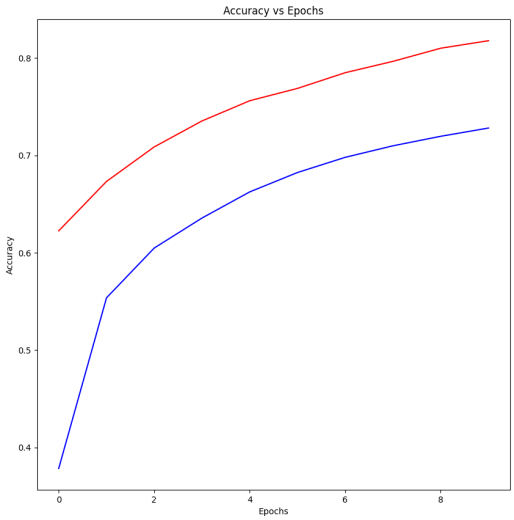
Validation F1 vs epoch & Confusion Matrix

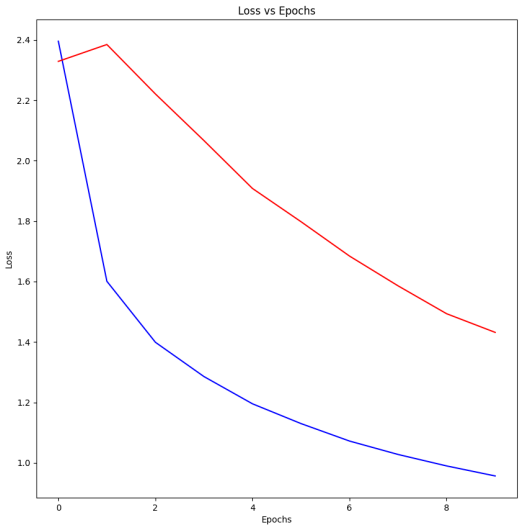




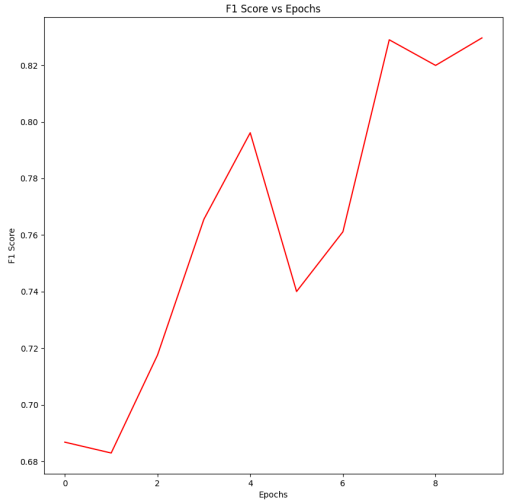
Learning Rate: 5e-4

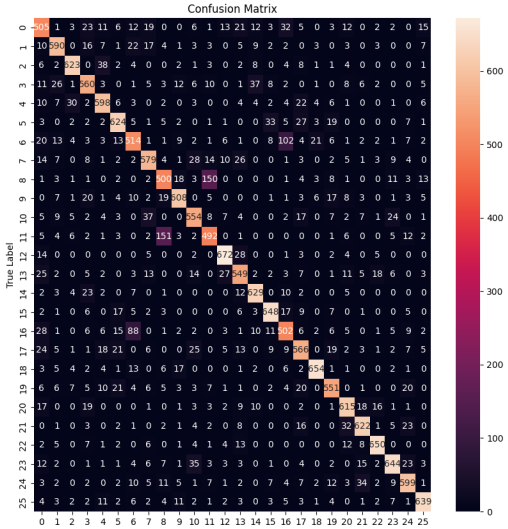
Accuracy vs Epoch & Loss vs Epoch:





Validation F1 vs epoch & Confusion Matrix

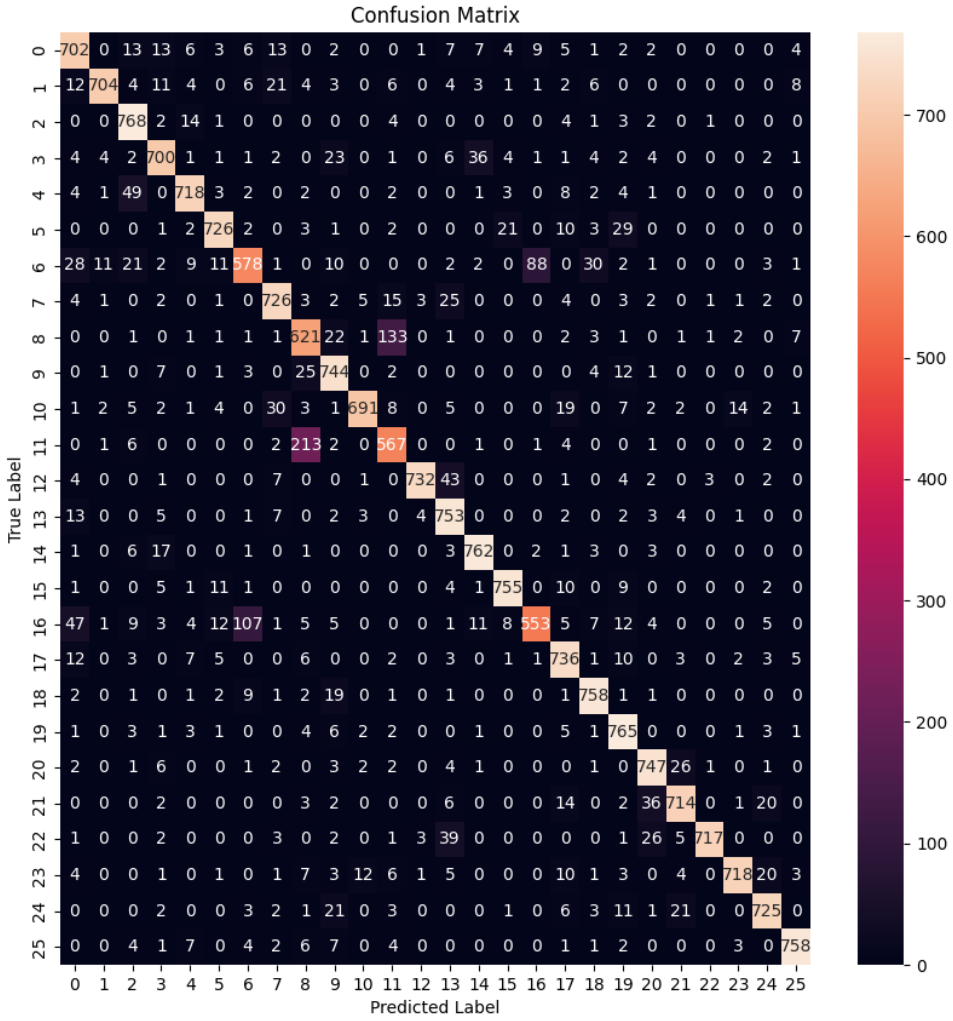




Test Data Performance for the best model

Test Accuracy: 88.64 %

Test F1 Score: 88.63 %



Observation:

1. Before applying “Adam” optimizer the loss tend to stuck in local optima (plateau)

2. For different learning rates, the learning convergences are varied. Lower learning rate can not converge perfectly, very high learning rate tends to overshoot from the direction of global optima. So, fine tuning is crucial.

3. Train_validation dataset was split in an 85-15 ratio. Train, validation and test dataset were loaded as batches of size 1024. Because we needed to train batch by batch, batch normalization could be an important method to try to achieve better performance.

4. Because the dataset is large, up to only 10 epochs are trained. From the plots, its is evident that loss and accuracy curves follow a usual trend. Training with more epochs can improve results.

5. The classes of the dataset started from 1 to 26. For proper one hot encoding, we transformed the labels to [0, 25].

6. For the loss function categorical cross entropy is used.

7. For multiclass classification, softmax activation was used in the final layer.

8. As seen from different models, the more units are present in the hidden layer, the more parameters are learnt. The validation score is observed to be improving as we increase the number of hidden layer units.

9. To prevent vanishing gradient problem, dropout is used.

10. There are a total of 48 plots generated. (3 models * 4 learning rate * 4 plots)