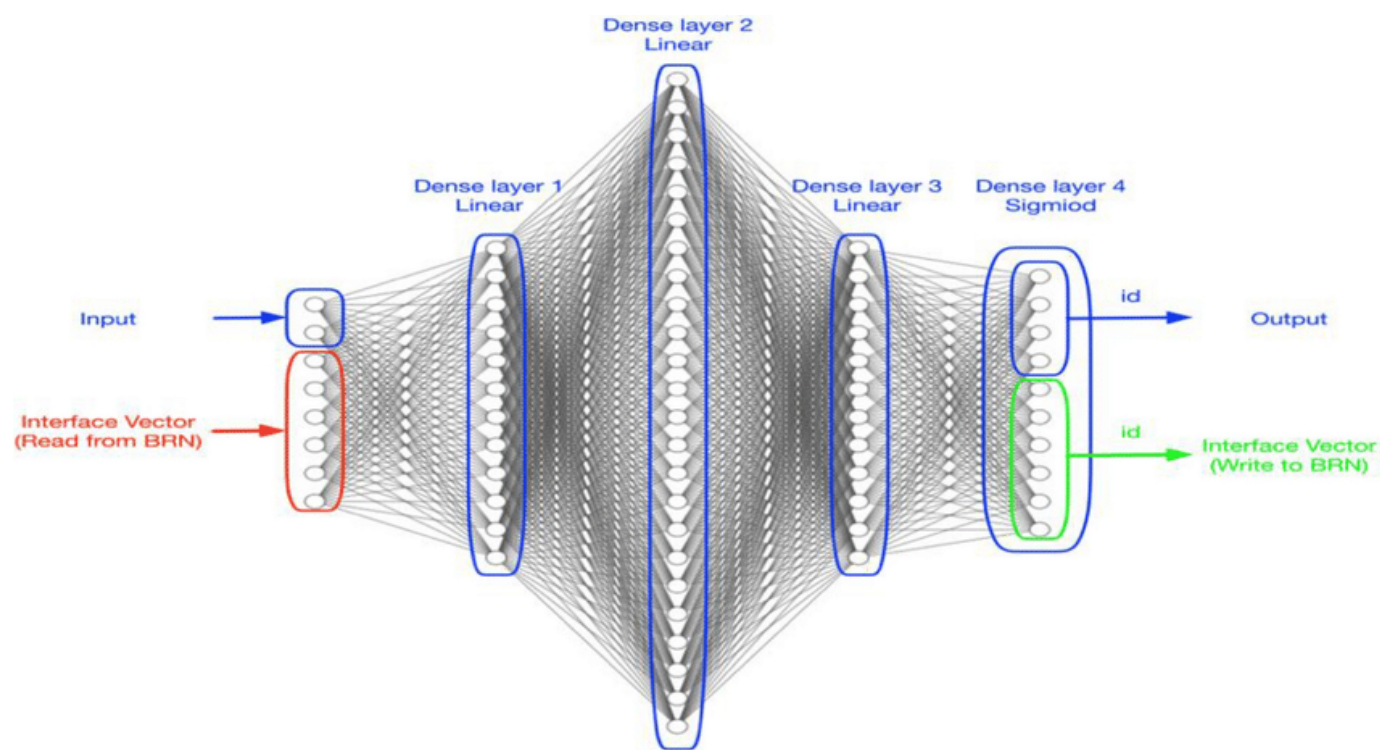


# Assignment 3: Function Approximation with Neural Network and Backpropagation

Name : Md. Al-Amin Sany

ID : 1905048

Section : A2



## Instructions to run the code

Run the following cells in the 1905048.ipynb file:

For the first model

```
for lr in learning_rates:
    run_diff_model('model_1',learning_rate=lr,dropout_rate=0.2,num_classes=10,num_epochs=15,batch_size=1024)
```

For the second model

```
for lr in learning_rates:
    run_diff_model('model_2',learning_rate=lr,dropout_rate=0.25,num_classes=10,num_epochs=15,batch_size=1024)
```

For the third model

```
▶ ▾  
    for lr in learning_rates:  
        | run_diff_model('model_3',learning_rate=lr,dropout_rate=0.3,num_classes=10,num_epochs=15,batch_size=1024)  
[70]
```

## Models

We have run three different models with 15 epochs, learning rates of 0.005, 0.0025, 0.00125, 0.000625 and batch sizes of 1024 . The model specifications are as follows:

The first model

```
model = NeuralNetwork(learning_rate=learning_rate)  
model.add(DenseLayer(512, 28*28))  
model.add(BatchNormalization(512))  
model.add(ReLU())  
model.add(Dropout(dropout_rate))  
model.add(DenseLayer(num_classes, 512))  
model.add(Softmax())  
return model
```

The Second Model

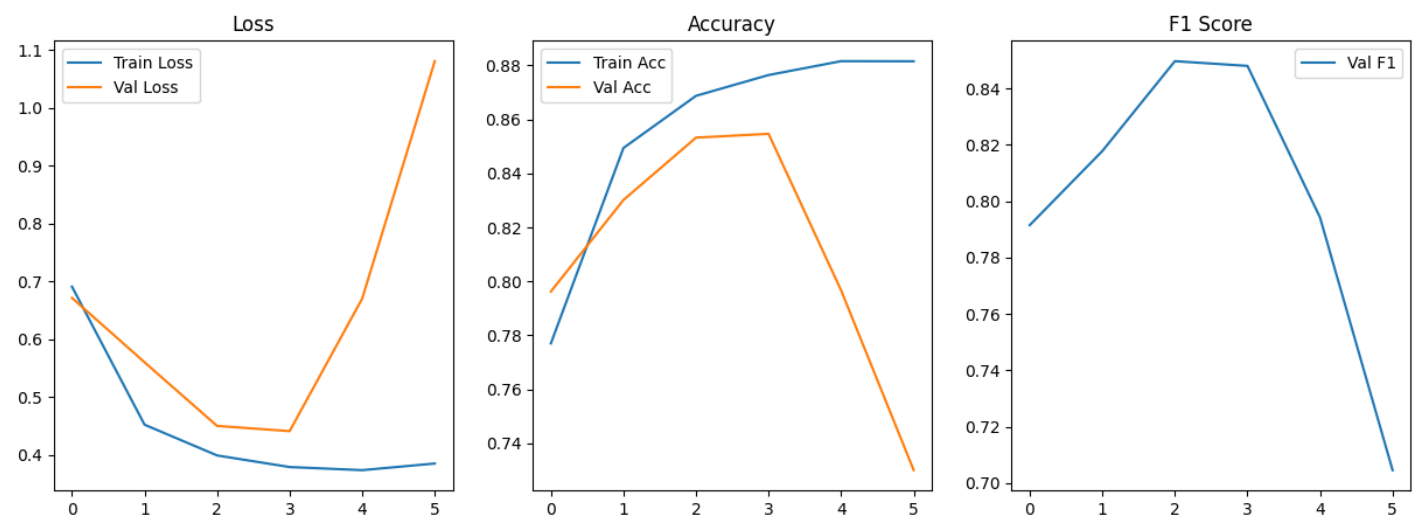
```
model = NeuralNetwork(learning_rate=learning_rate)  
model.add(DenseLayer(256, 28*28))  
model.add(BatchNormalization(256))  
model.add(ReLU())  
model.add(Dropout(dropout_rate))  
model.add(DenseLayer(num_classes, 256))  
model.add(Softmax())  
return model
```

The Third Model

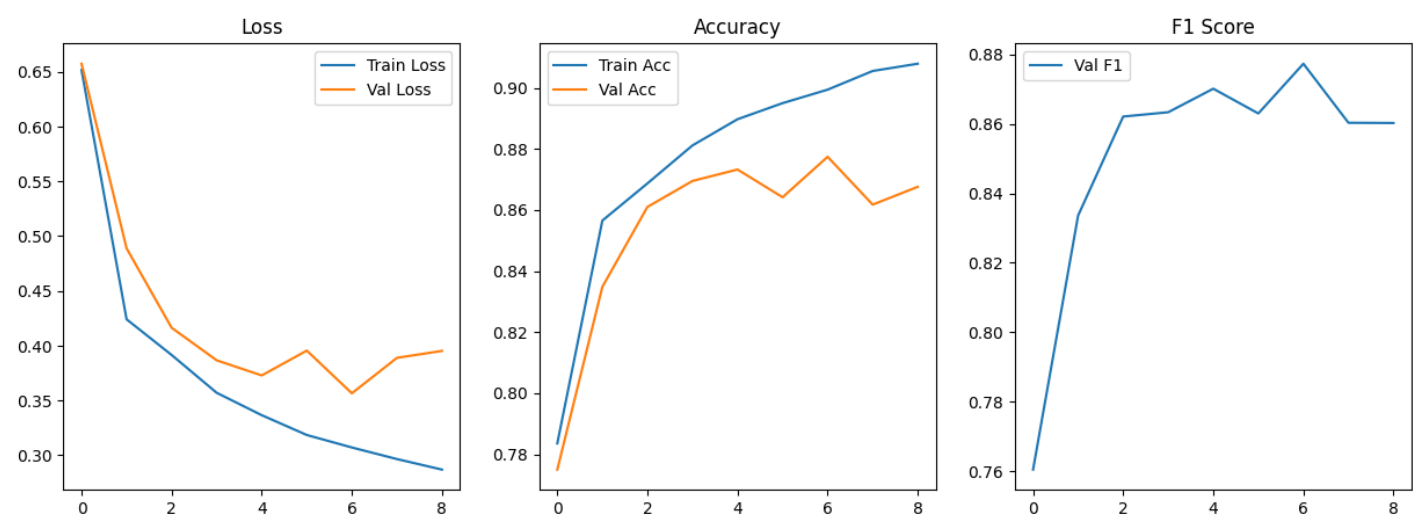
```
model = NeuralNetwork(learning_rate=learning_rate)  
model.add(DenseLayer(512, 28*28))  
model.add(BatchNormalization(512))  
model.add(ReLU())  
model.add(Dropout(dropout_rate))  
model.add(DenseLayer(256, 512))  
model.add(BatchNormalization(256))  
model.add(ReLU())  
model.add(Dropout(dropout_rate))  
model.add(DenseLayer(num_classes, 256))  
model.add(Softmax())  
return model
```

## Graphs

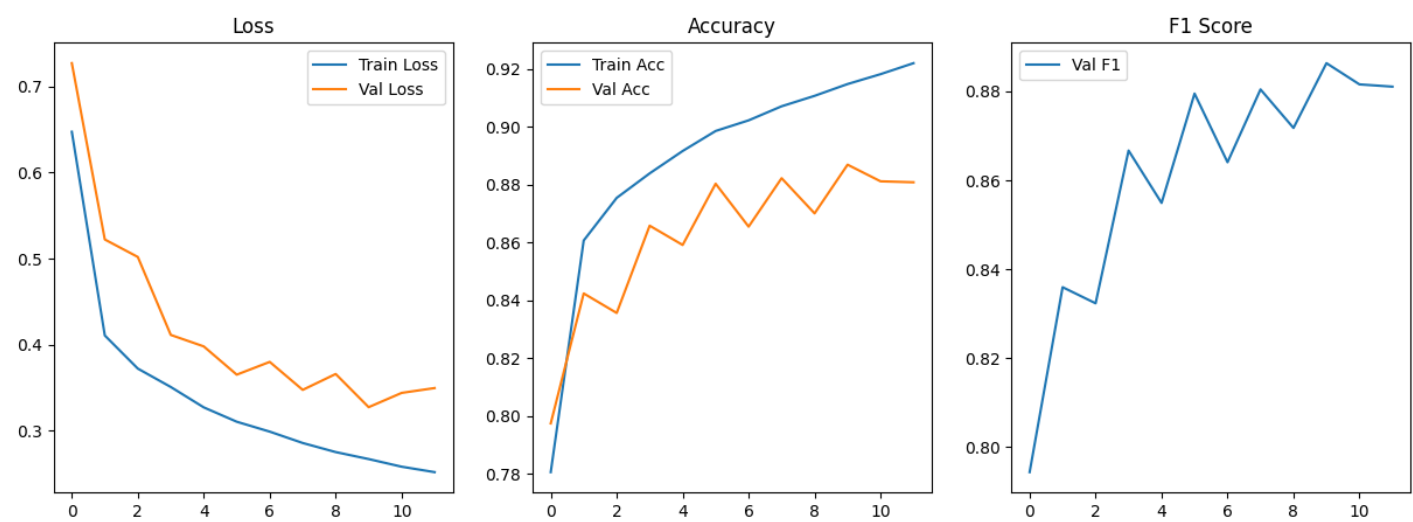
• **Model 1 (learning rate = 0.005)**



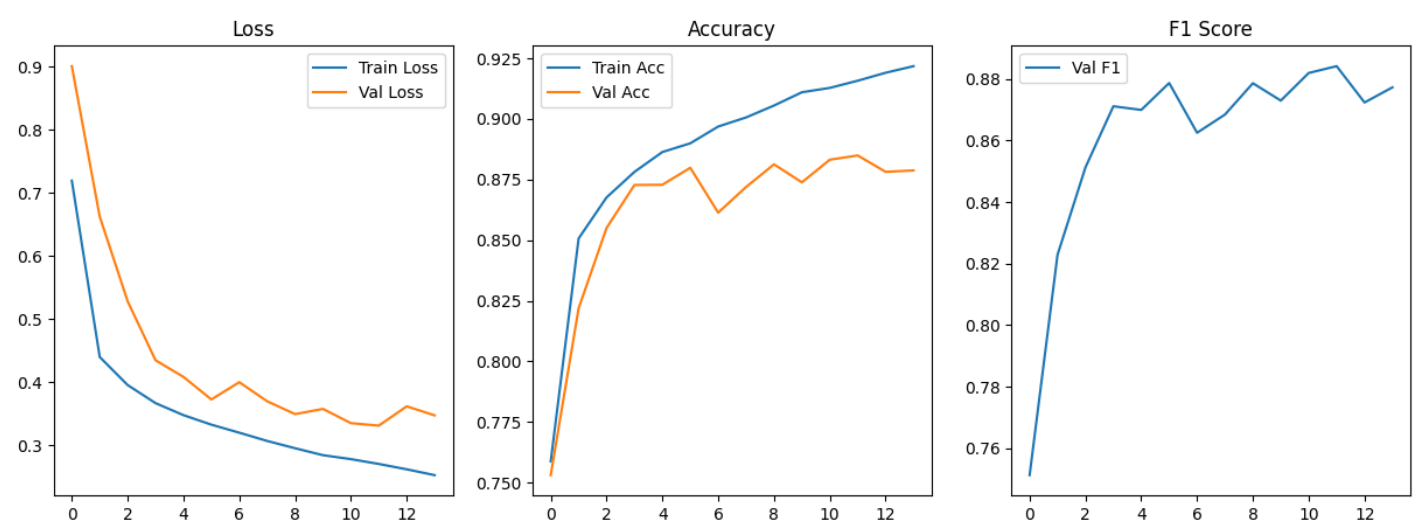
• **Model 1 (learning rate = 0.0025)**



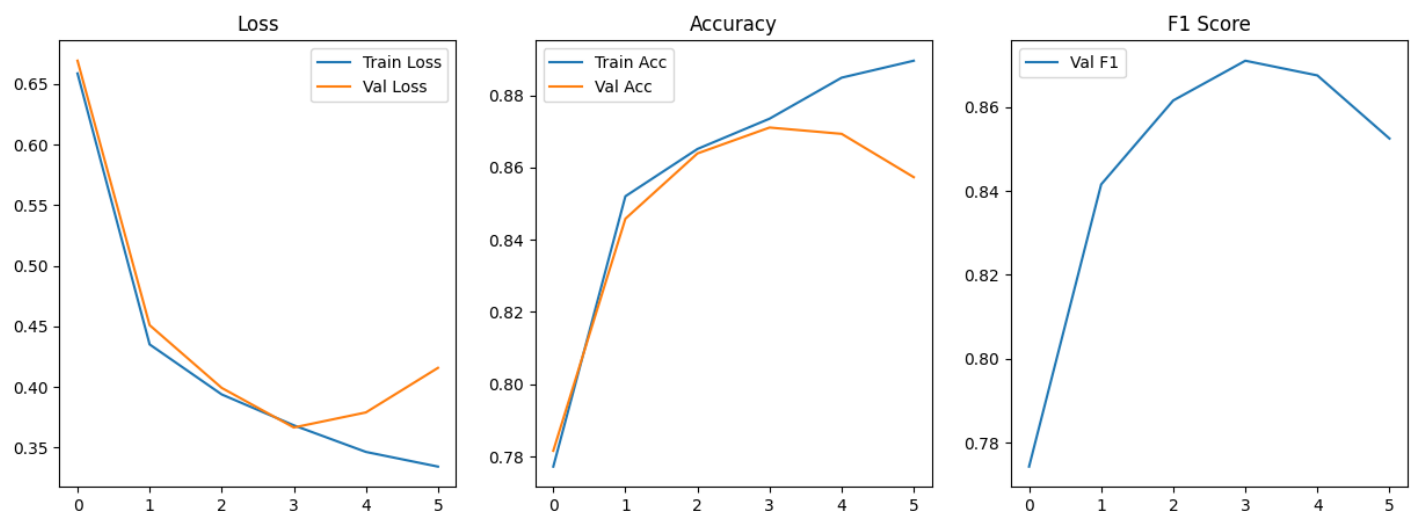
• **Model 1 (learning rate = 0.00125)**



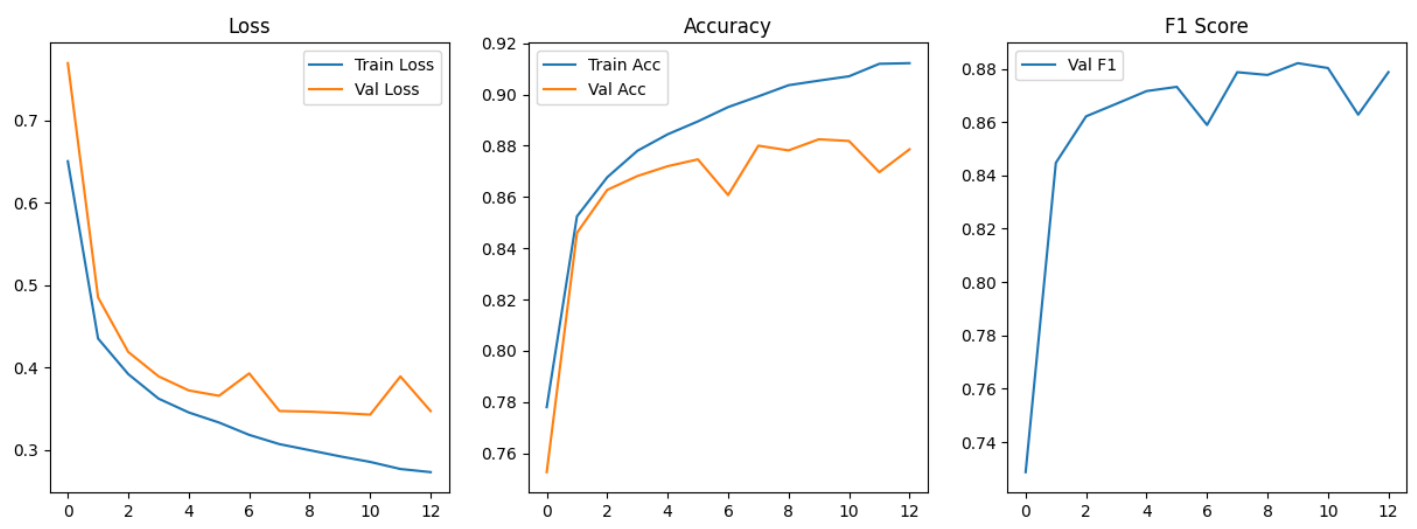
• **Model 1 (learning rate = 0.000625)**



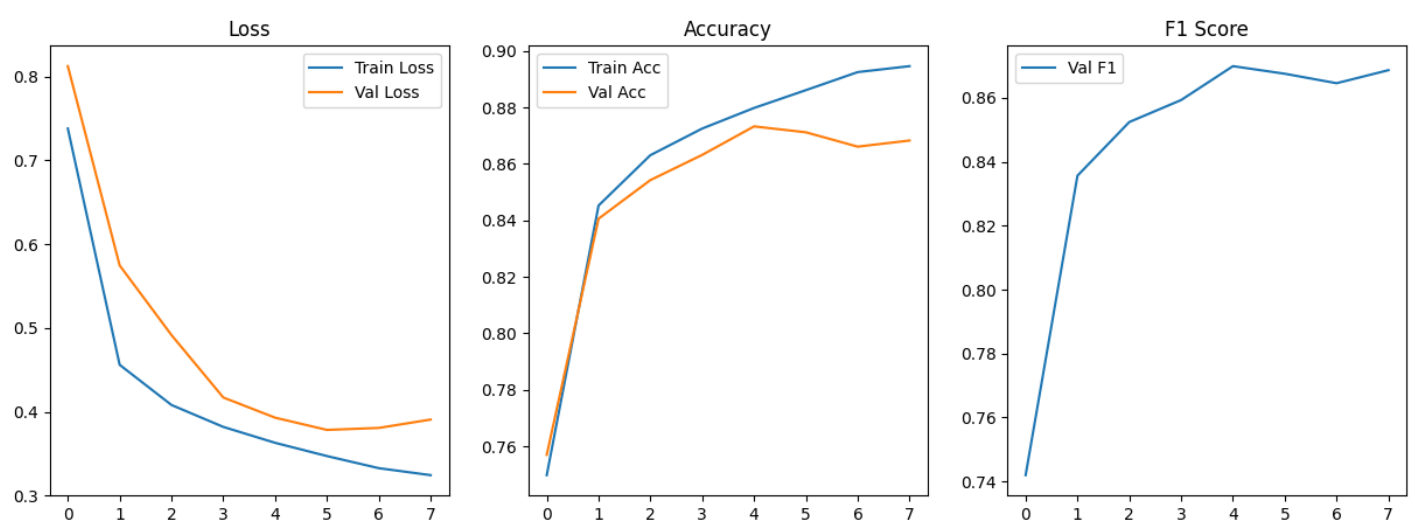
• **Model 2 (learning rate = 0.005)**



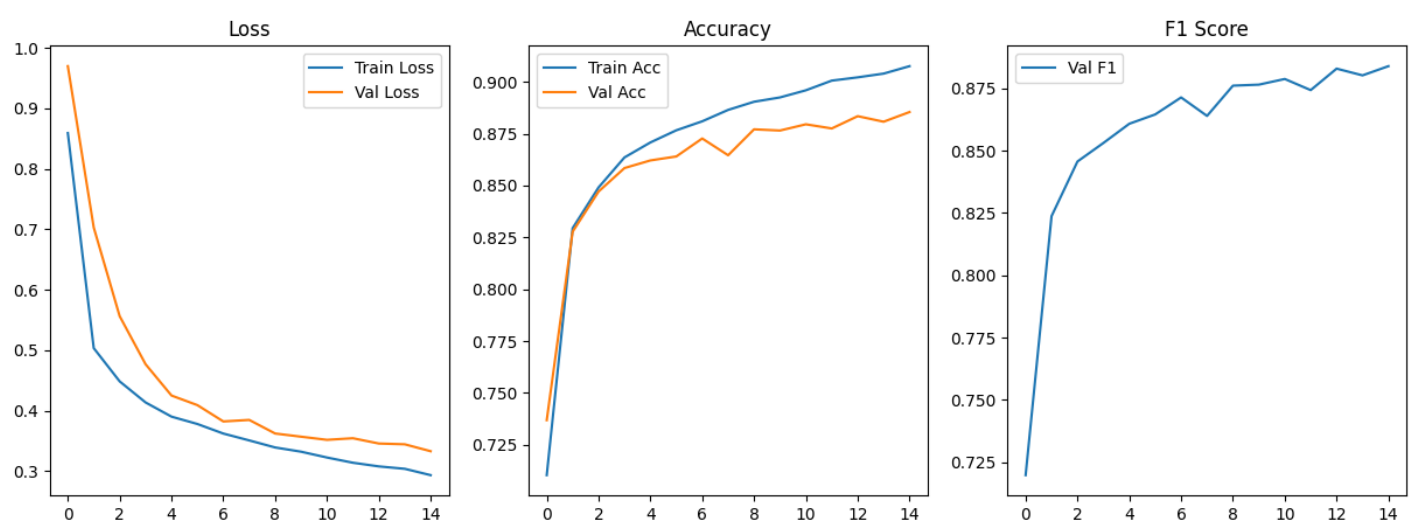
• **Model 2 (learning rate = 0.0025)**



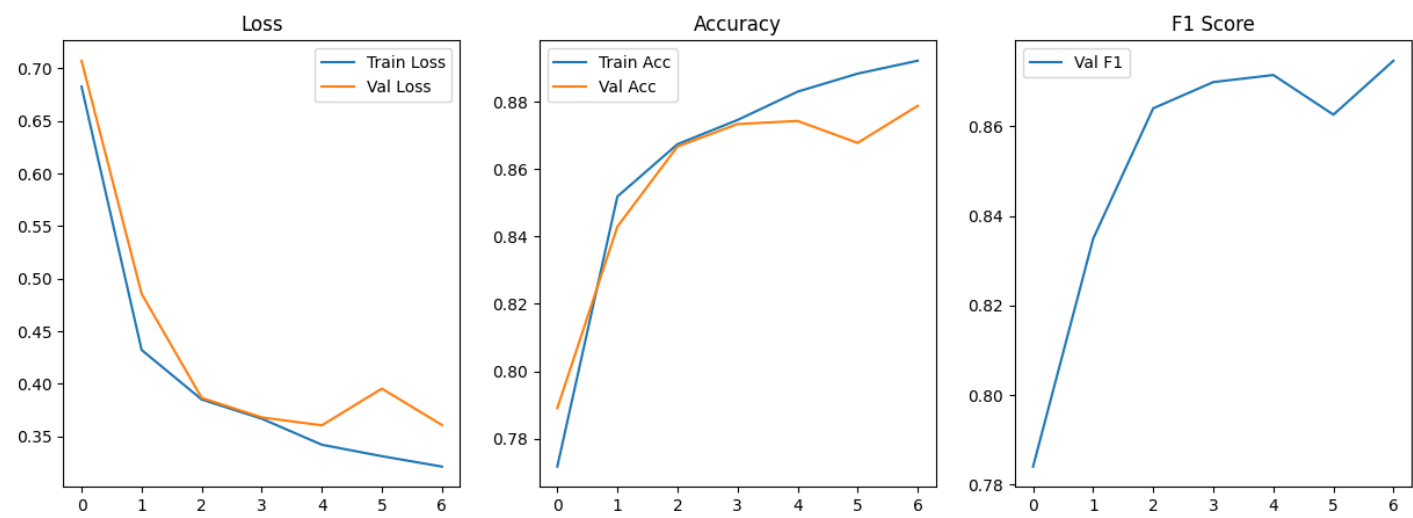
• **Model 2 (learning rate = 0.00125)**



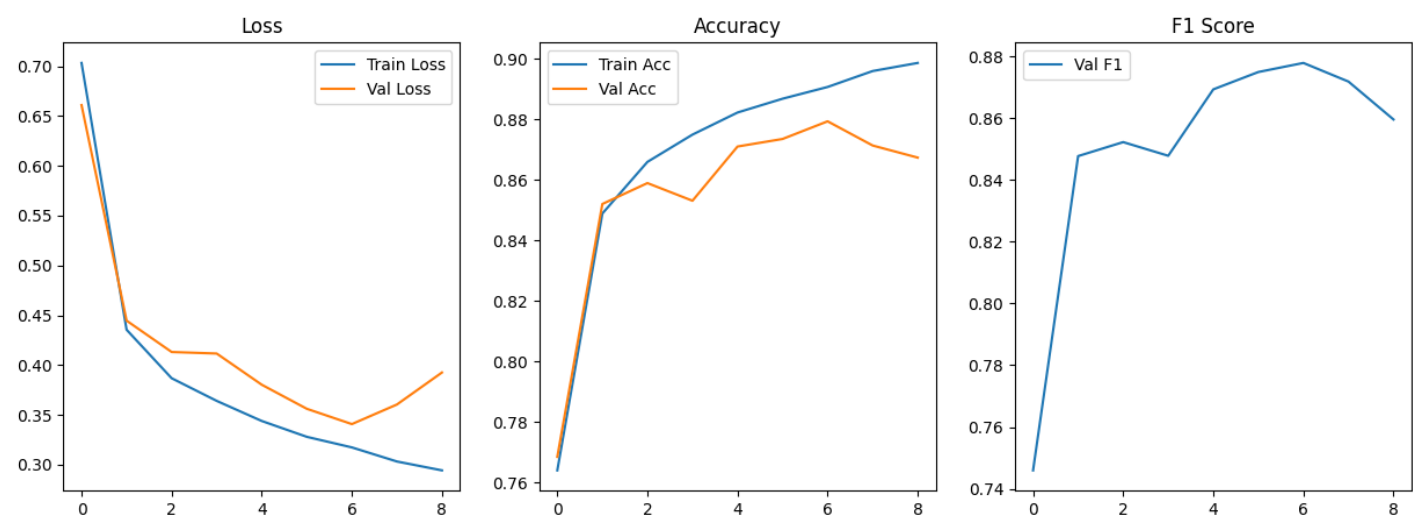
• **Model 2 (learning rate = 0.000625)**



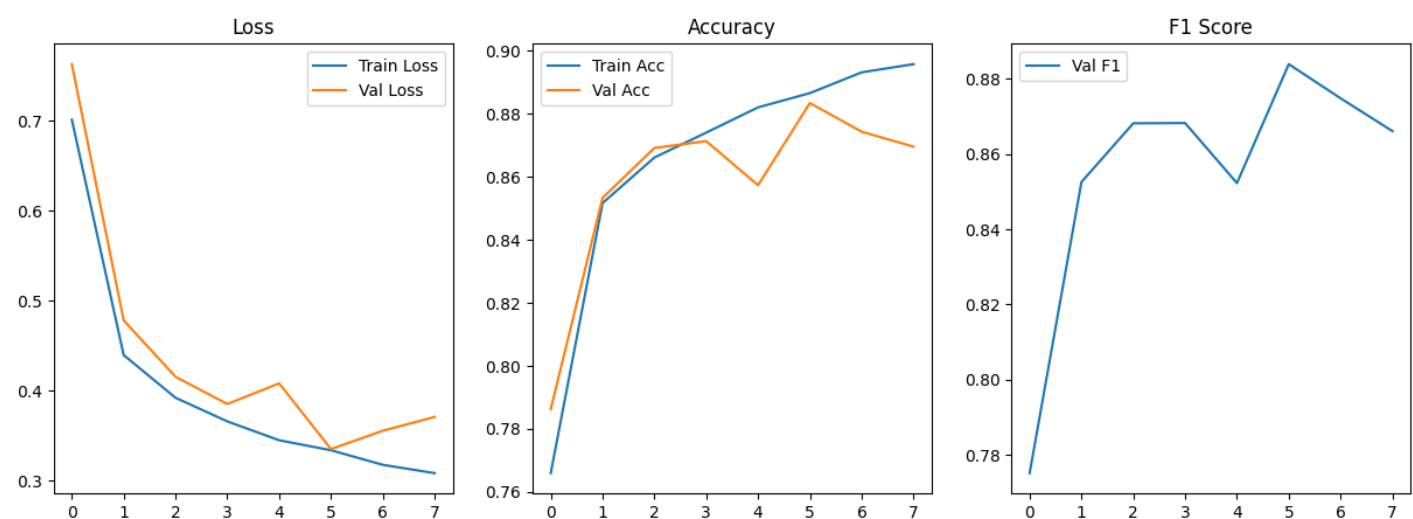
• **Model 3 (learning rate = 0.005)**



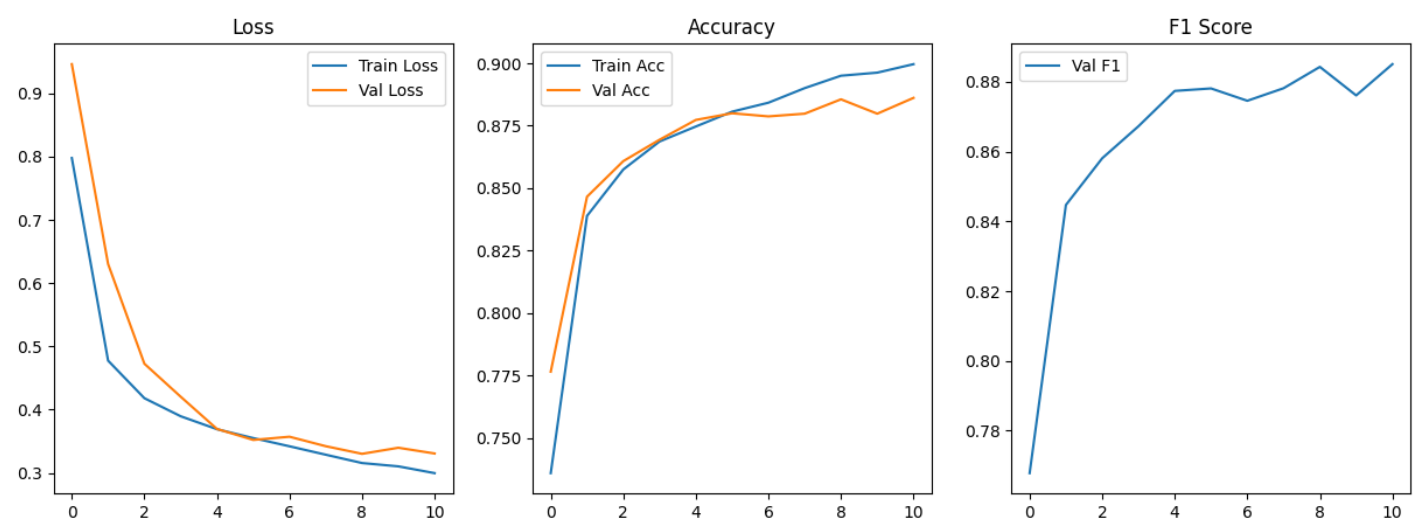
• **Model 3 (learning rate = 0.0025)**



• **Model 3 (learning rate = 0.00125)**

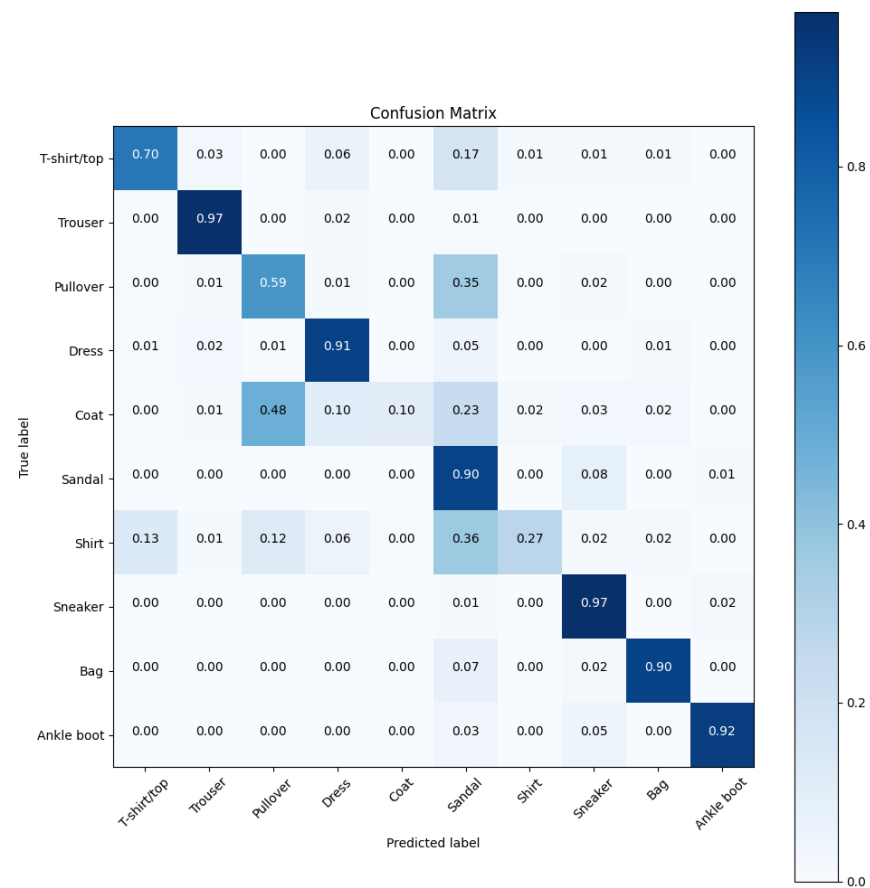


• **Model 3 (learning rate = 0.000625)**

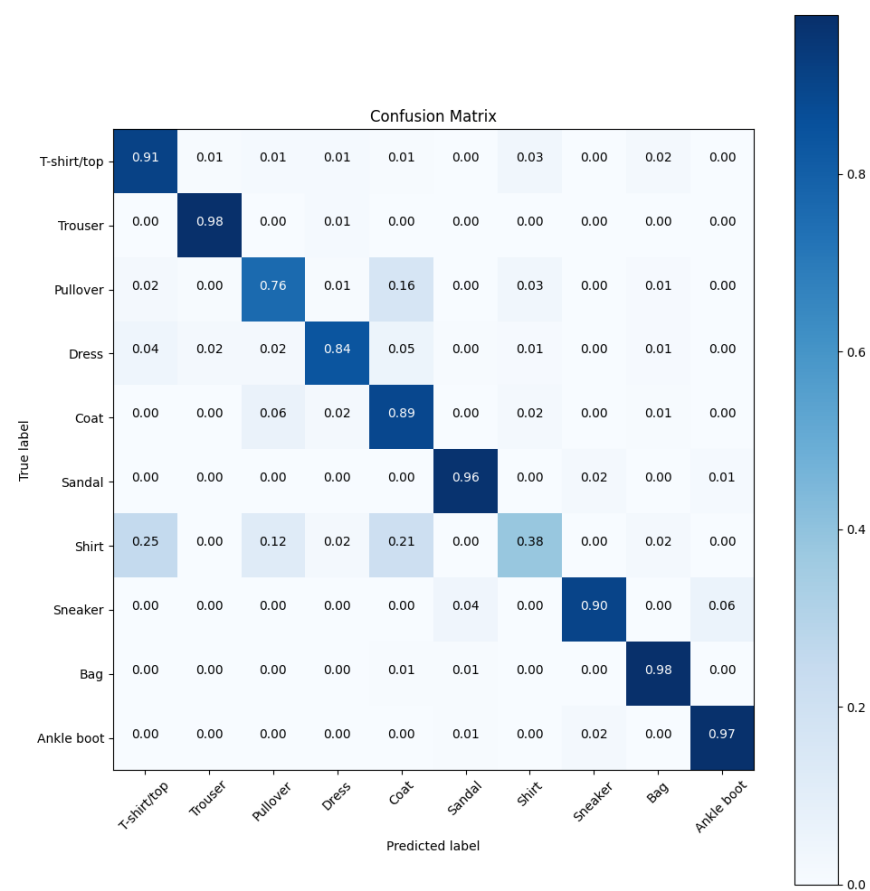


# Confusion Metrices

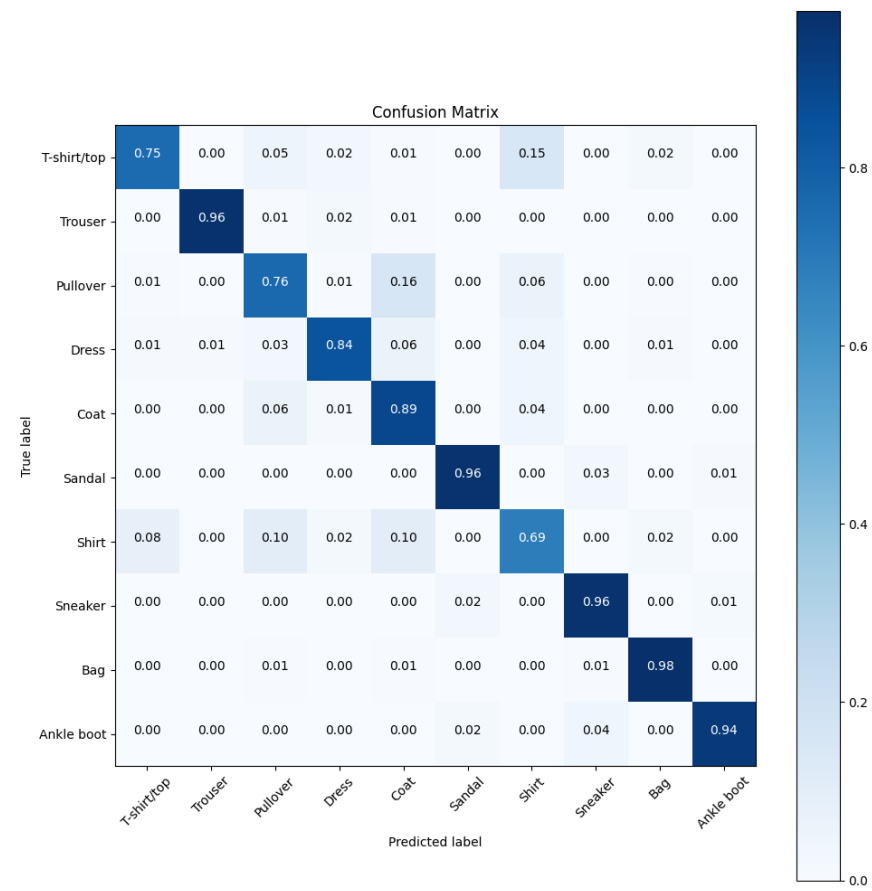
- Model 1 (learning rate = 0.005)



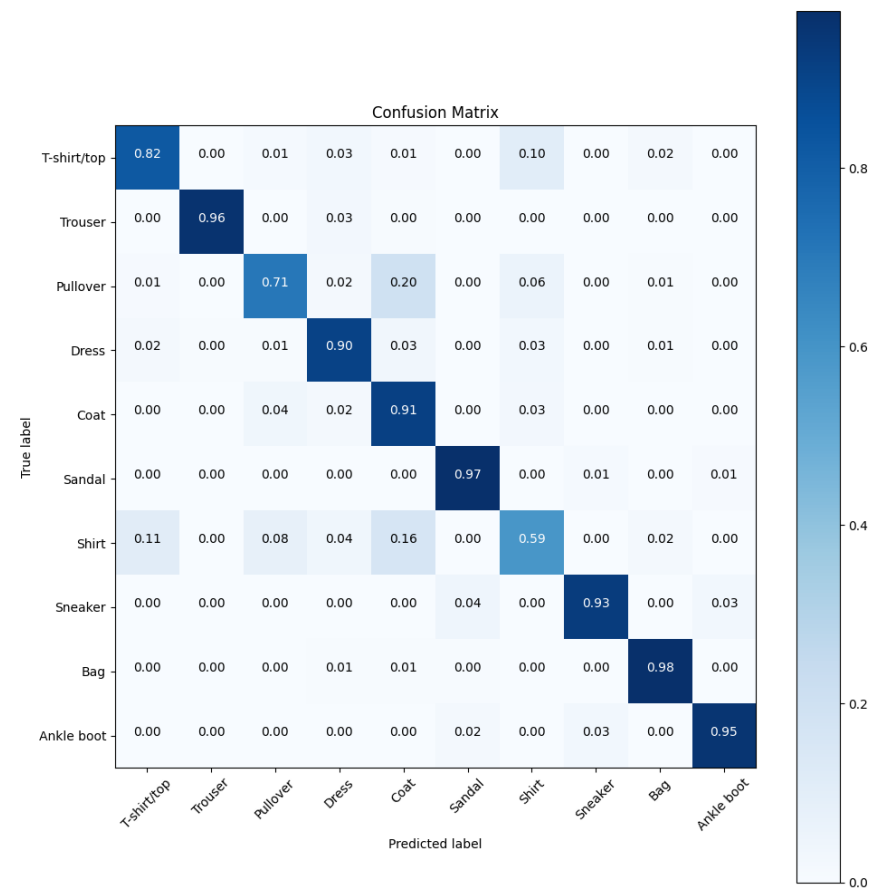
- Model 1 (learning rate = 0.0025)



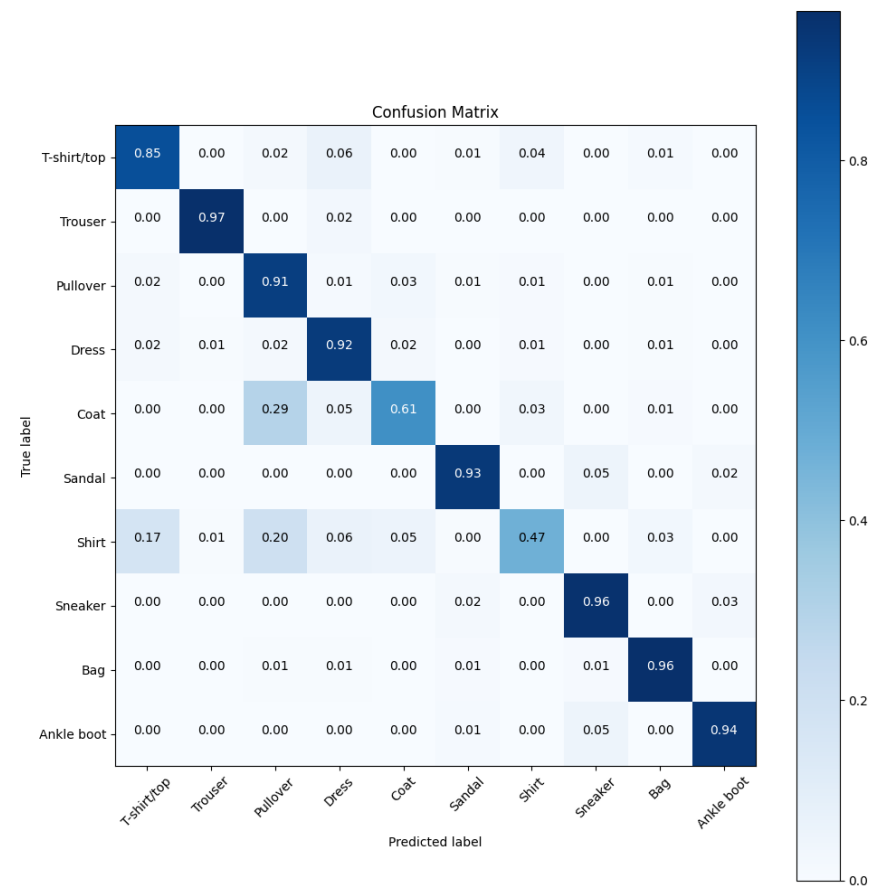
- Model 1 (learning rate = 0.00125)



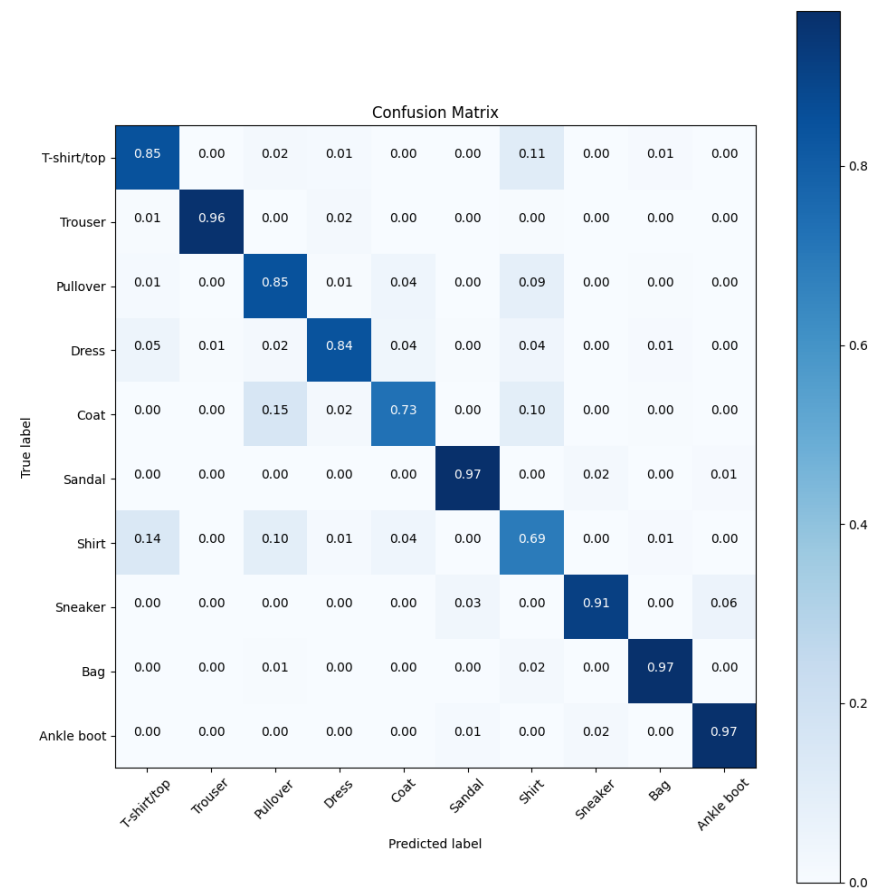
- **Model 1 (learning rate = 0.000625)**



- **Model 2 (learning rate = 0.005)**

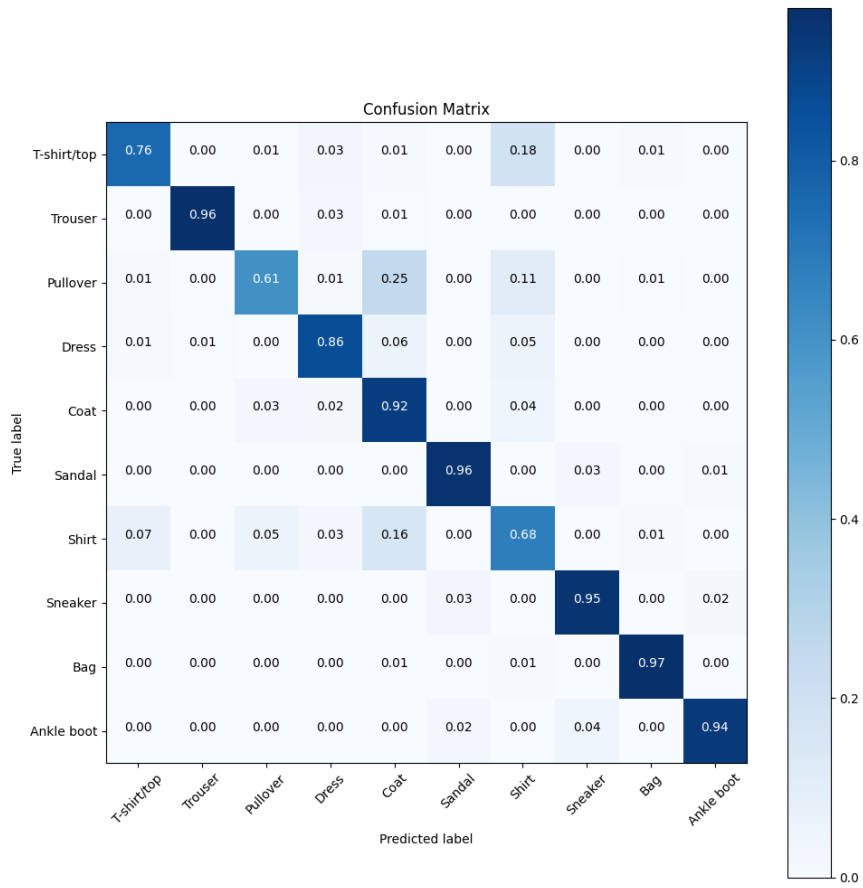


- Model 2 (learning rate = 0.0025)

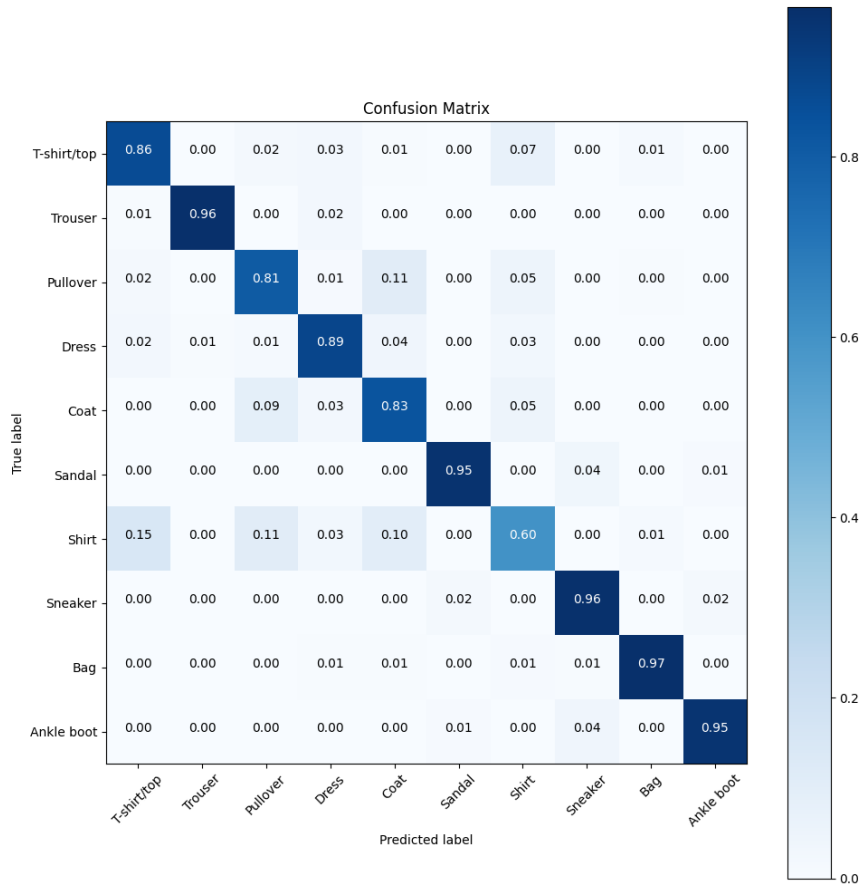


- Model 2 (learning rate = 0.00125)

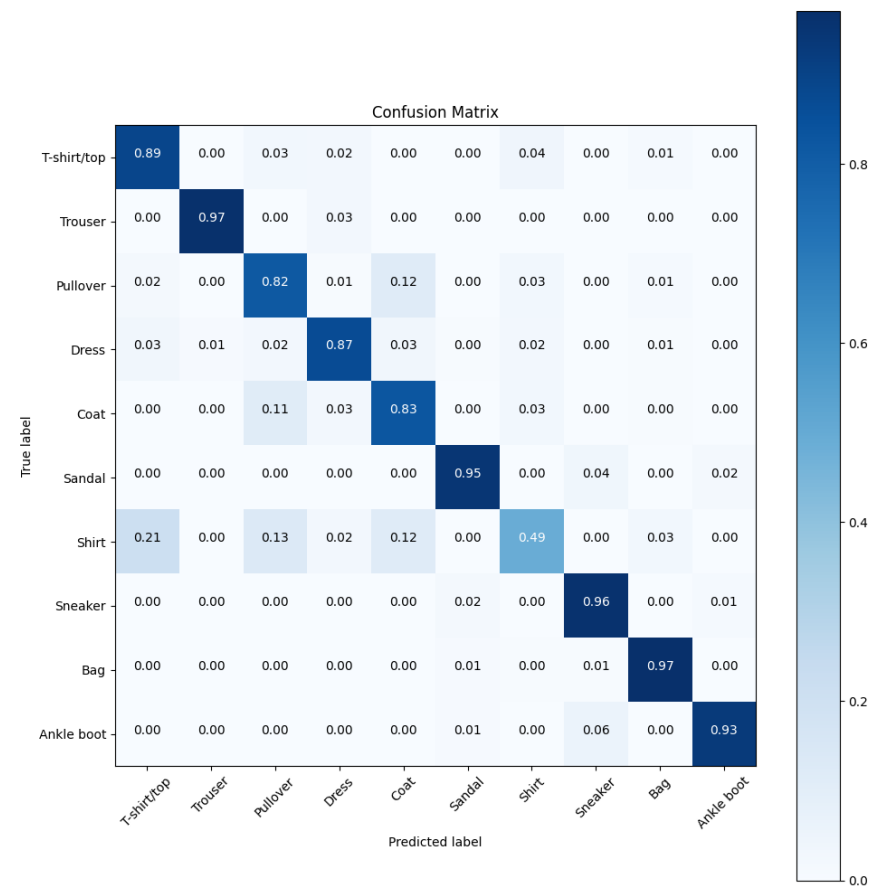




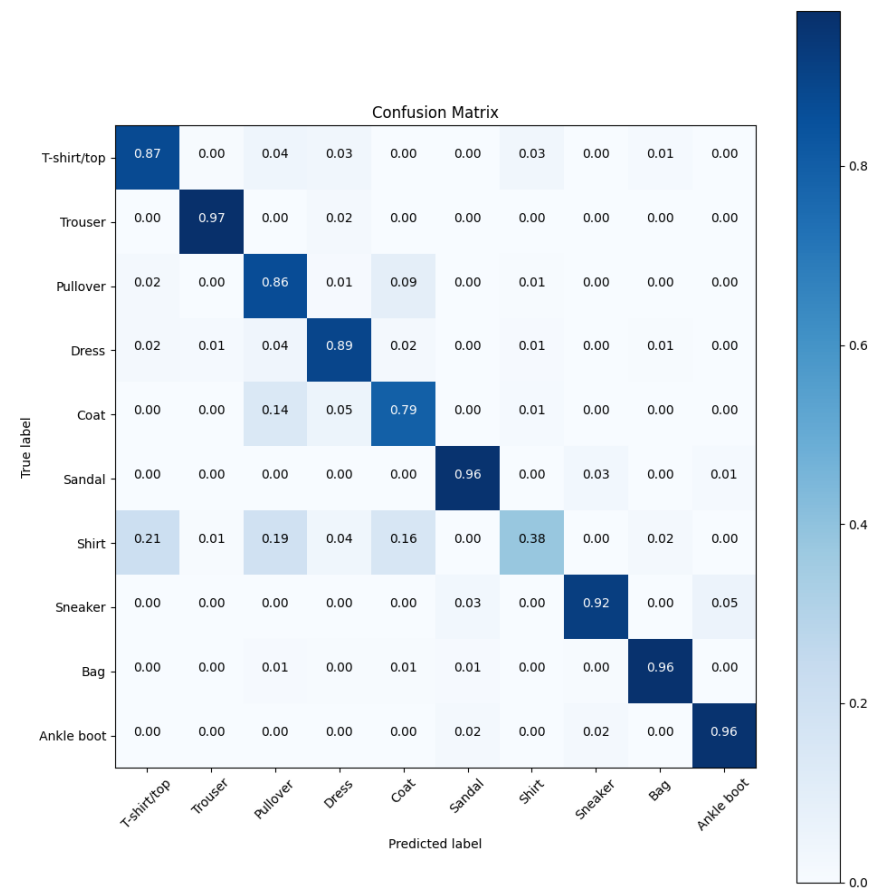
- Model 2 (learning rate = 0.000625)



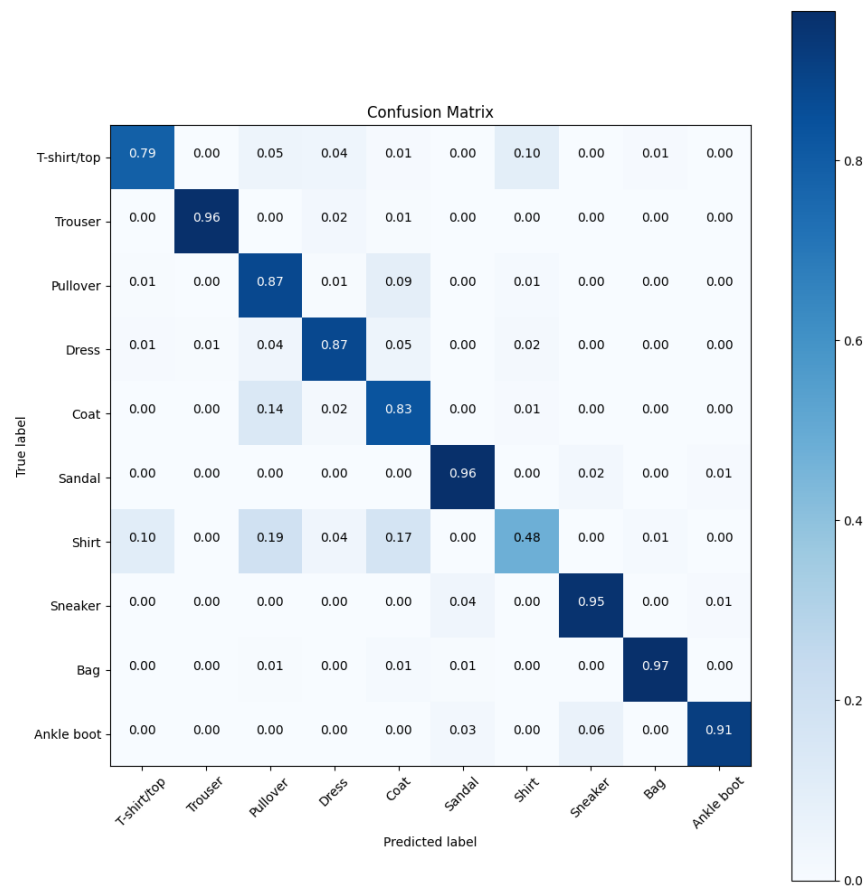
- Model 3 (learning rate = 0.005)



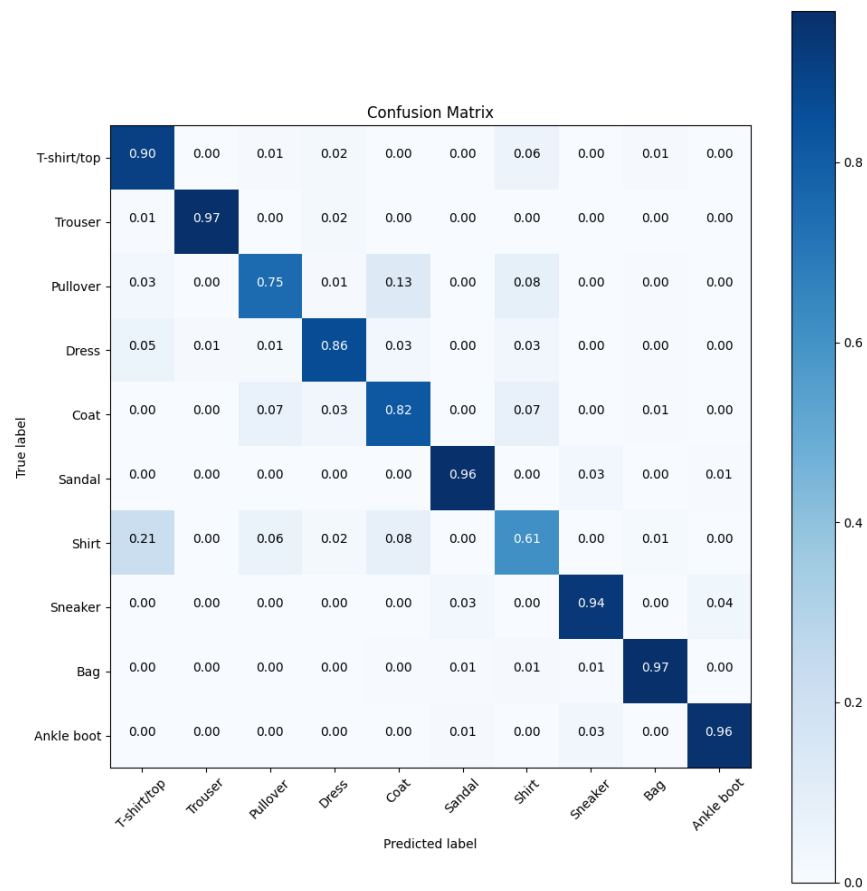
- Model 3 (learning rate = 0.0025)



- Model 3 (learning rate = 0.00125)



- **Model 3 (learning rate = 0.000625)**



## Chosen model and its average loss, accuracy and f1 score

I have chosen model 3 with learning rate of 0.000625 because it gave the best f1 score on validation set among all the models. The below table shows the loss, accuracy and f1 score of each iteration of training phase:

### Running model\_3 with learning rate 0.000625 and dropout rate 0.3

Epoch 1/15

100%|██████████| 46/46 [00:12<00:00, 3.78it/s]

Train Loss: 0.7976, Train Acc: 0.7359

Val Loss: 0.9460, Val Acc: 0.7765, Val F1: 0.7678

Epoch 2/15

100%|██████████| 46/46 [00:14<00:00, 3.22it/s]

Train Loss: 0.4773, Train Acc: 0.8389  
Val Loss: 0.6301, Val Acc: 0.8466, Val F1: 0.8447

Epoch 3/15

100%|██████████| 46/46 [00:11<00:00, 3.94it/s]  
Train Loss: 0.4180, Train Acc: 0.8575  
Val Loss: 0.4725, Val Acc: 0.8608, Val F1: 0.8581

Epoch 4/15

100%|██████████| 46/46 [00:11<00:00, 4.02it/s]  
Train Loss: 0.3895, Train Acc: 0.8687  
Val Loss: 0.4204, Val Acc: 0.8694, Val F1: 0.8673

Epoch 5/15

100%|██████████| 46/46 [00:11<00:00, 3.95it/s]  
Train Loss: 0.3690, Train Acc: 0.8747  
Val Loss: 0.3694, Val Acc: 0.8773, Val F1: 0.8774

Epoch 6/15

100%|██████████| 46/46 [00:12<00:00, 3.82it/s]  
Train Loss: 0.3550, Train Acc: 0.8806  
Val Loss: 0.3521, Val Acc: 0.8800, Val F1: 0.8781

Epoch 7/15

100%|██████████| 46/46 [00:16<00:00, 2.81it/s]  
Train Loss: 0.3420, Train Acc: 0.8842  
Val Loss: 0.3571, Val Acc: 0.8788, Val F1: 0.8746

Epoch 8/15

100%|██████████| 46/46 [00:14<00:00, 3.17it/s]  
Train Loss: 0.3287, Train Acc: 0.8901  
Val Loss: 0.3421, Val Acc: 0.8798, Val F1: 0.8782

Epoch 9/15

100%|██████████| 46/46 [00:12<00:00, 3.58it/s]  
Train Loss: 0.3155, Train Acc: 0.8950  
Val Loss: 0.3301, Val Acc: 0.8856, Val F1: 0.8843

Epoch 10/15

100%|██████████| 46/46 [00:14<00:00, 3.21it/s]  
Train Loss: 0.3103, Train Acc: 0.8963  
Val Loss: 0.3397, Val Acc: 0.8798, Val F1: 0.8761

Epoch 11/15

100%|██████████| 46/46 [00:11<00:00, 3.95it/s]  
Train Loss: 0.2996, Train Acc: 0.8997  
Val Loss: 0.3306, Val Acc: 0.8862, Val F1: 0.8851

Early stopping triggered at epoch 11

# Independent Test Performance of the best model

Model: model\_3  
Learning Rate: 0.000625  
Dropout Rate: 0.3  
Batch Size: 1024  
Number of Epochs: 15

Classification Report:

Category	Precision	Recall	F1-Score
T-shirt/top	0.75	0.90	0.82
Trouser	0.98	0.97	0.97
Pullover	0.83	0.75	0.79
Dress	0.89	0.86	0.88
Coat	0.78	0.82	0.80
Sandal	0.95	0.96	0.96
Shirt	0.71	0.61	0.66
Sneaker	0.93	0.94	0.94
Bag	0.96	0.97	0.96
Ankle boot	0.96	0.96	0.96

Test Accuracy: 0.8735  
Test F1 Score: 0.8723

