CSE472 (Machine Learning Sessional)

Assignment 3: Function Approximation with Neural Network and Backpropagation

Report
Submitted By
MD Sadik Hossain Shanto
ID: 1905101
Section: B2

To **train** the code, run this cell and adjust the parameters as required:

- The trainer function will automatically save the model, the default path is "model/model.pkl"

```
# preprocess the data
train_data, val_data, test_data, train_labels, val_labels, test_labels = preprocess_data(train_dataset, test_dataset)

input_dim = train_data.shape[1] * train_data.shape[1]
output_dim = len(np.unique(train_labels))

layers = [
    Dense(input_dim, 512),
    Batchnorm(512),
    ReLU(),
    Dropout(0.5),
    Dense(512, output_dim),
    SoftMax()
]
# train the model
train_loss, val_loss, train_acc, val_acc, val_macro_f1, val_conf_matrix = trainer(
    layers=layers,
    train_data=train_data,
    train_labels=train_labels,
    val_data=val_data,
    val_labels=val_labels,
    learning_rate=0.005,
    nepochs=60,
    batch_size=64
)
```

To **tes**t the code, Run the following cell and adjust the params as required:

- First, define the layers (should be the same as the model you want to load)
- Then specify the path from where your saved model should be loaded.

```
layers = [
    Dense(input_dim, 512),
    Batchnorm(512),
    ReLU(),
    Dropout(0.5),
    Dense(512, 256),
    Batchnorm(256),
    ReLU(),
    Dropout(0.5),
    Dense(256, output_dim),
    SoftMax()
]

# test the model
accuracy, precision, recall, f1, conf_matrix = tester(
    layers=layers,
    path='model/model.pkl',
    test_data=test_data,
    test_labels=test_labels,
    batch_size=64
)
```

Model Architechtures

Model:01

```
layers = [
    Dense(input_dim, 512),
    Batchnorm(512),
    ReLU(),
    Dropout(0.5),
    Dense(512, output_dim),
    SoftMax()
]
```

Model:02

```
layers = [
    Dense(input_dim, 512),
    Batchnorm(512),
    ReLU(),
    Dropout(0.5),
    Dense(512, 256),
    Batchnorm(256),
    ReLU(),
    Dropout(0.5),
    Dense(256, output_dim),
    SoftMax()
]
```

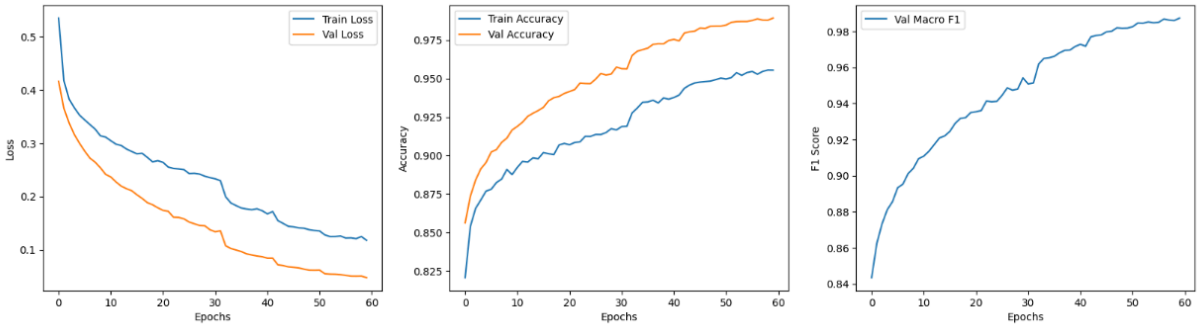
Model:03

```
layers = [
    Dense(input_dim, 512),
    Batchnorm(512),
    ReLU(),
    Dropout(0.5),
    Dense(512, 256),
    Batchnorm(256),
    ReLU(),
    Dropout(0.5),
    Dense(256, 128),
    Batchnorm(128),
    ReLU(),
    Dropout(0.5),
    Dense(128, output_dim),
    SoftMax()
]
```

Model-1

lr: 0.005

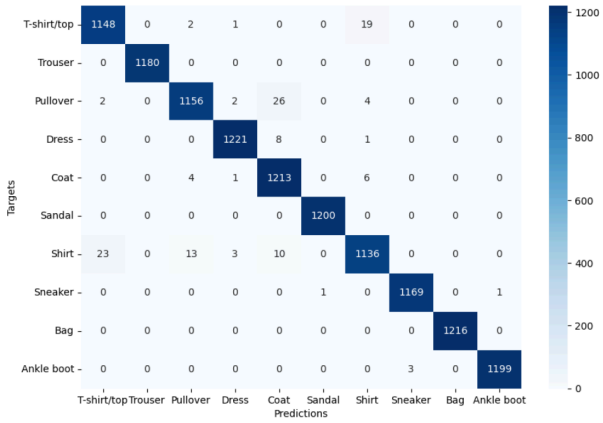
epoch: 60



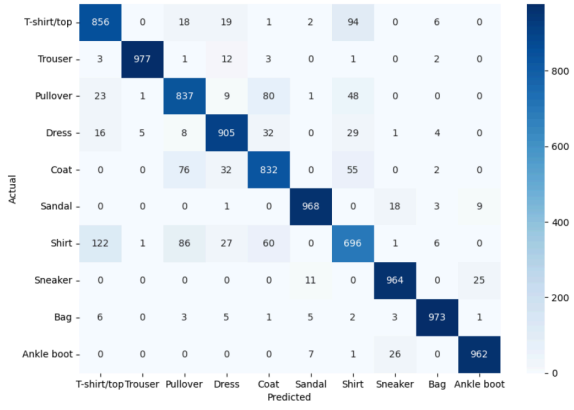
Training and validation performance:

Train Loss: 0.1177, Val Loss: 0.0478,
Train Acc: 0.9554, Val Acc: 0.9891,
Val Macro F1: 0.9873

Validation Confusion Matrix



Test Confusion Matrix



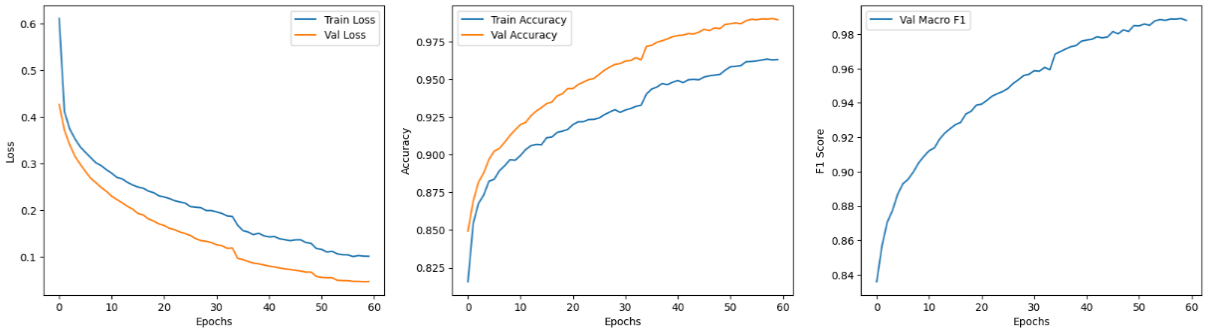
Test Performance:

Test Accuracy: 0.8984,
Precision: 0.8980,
Recall: 0.8984,
F1 Score: 0.8981

Model-1

lr: 0.001

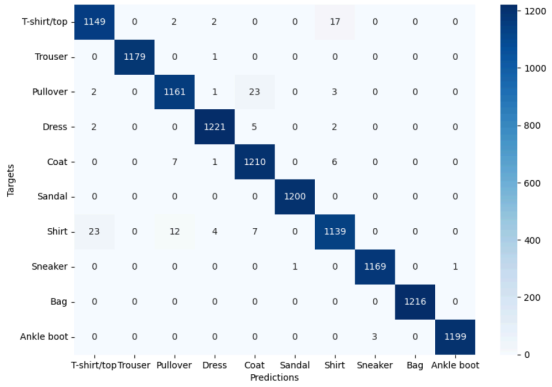
epoch: 60



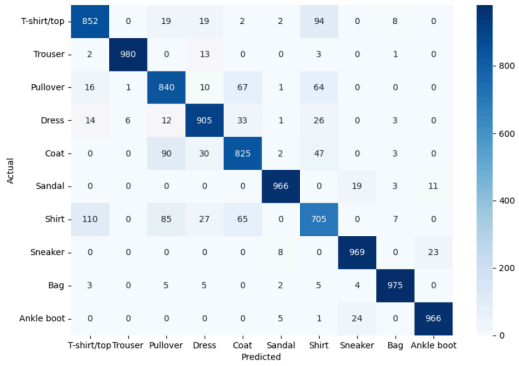
Training and validation performance:

Train Loss: 0.1016, Val Loss: 0.0474,
Train Acc: 0.9631, Val Acc: 0.9896,
Val Macro F1: 0.9880

Validation Confusion Matrix



Test Confusion Matrix



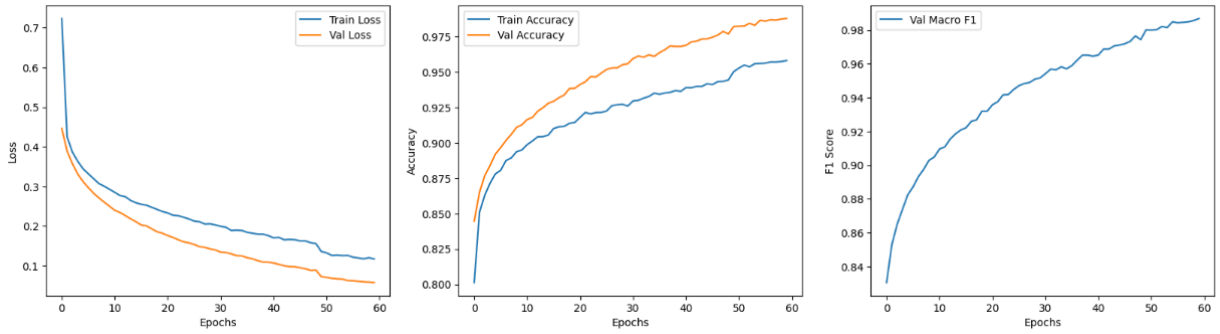
Test Performance:

Test Accuracy: 0.8997,
Precision: 0.8995,
Recall: 0.8997,
F1 Score: 0.8995

Model-1

lr: 0.0005

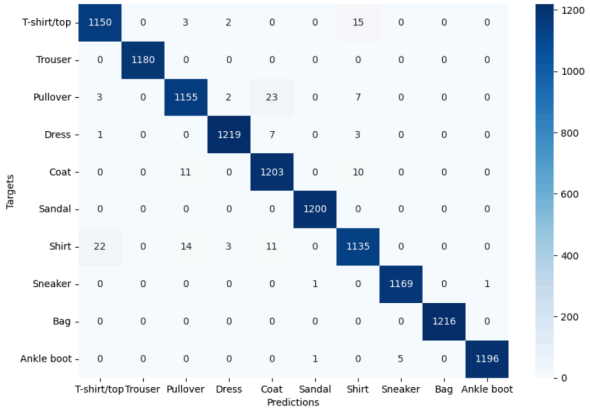
epoch: 60



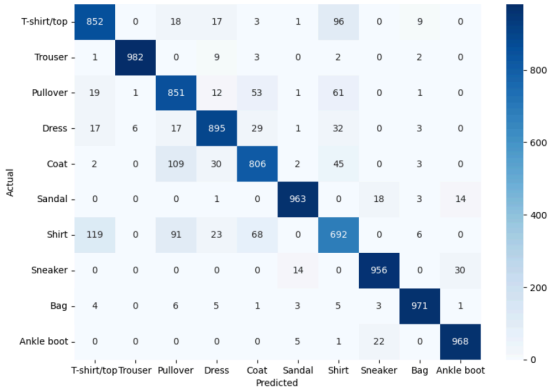
Training and validation performance:

Train Loss: 0.1174, Val Loss: 0.0580,
Train Acc: 0.9581, Val Acc: 0.9879,
Val Macro F1: 0.9868

Validation Confusion Matrix



Test Confusion Matrix



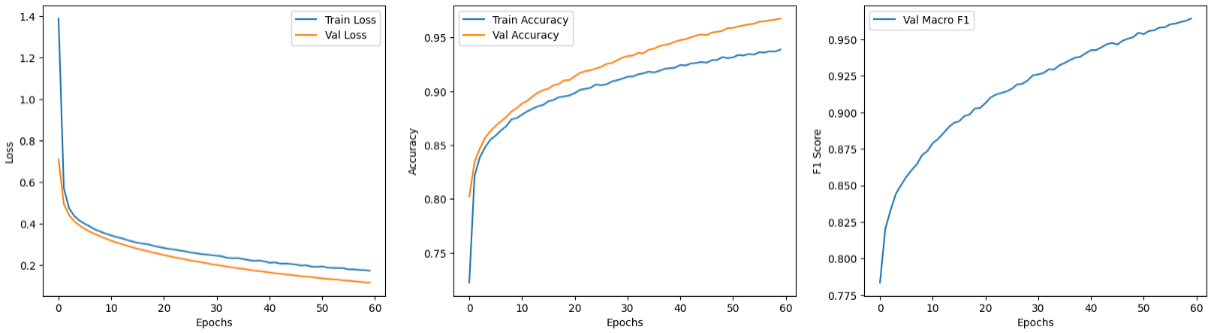
Test Performance:

Test Accuracy: 0.8950,
Precision: 0.8951,
Recall: 0.8950,
F1 Score: 0.8948

Model-1

lr: 0.0001

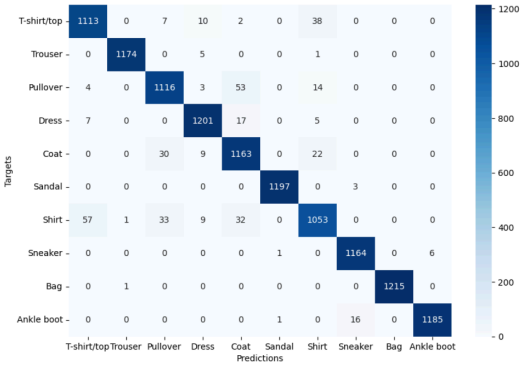
epoch: 60



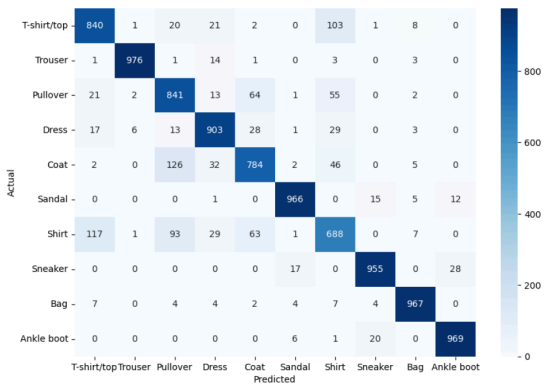
Training and validation performance:

Train Loss: 0.1742, Val Loss: 0.1167,
Train Acc: 0.9388, Val Acc: 0.9677,
Val Macro F1: 0.9643

Validation Confusion Matrix



Test Confusion Matrix



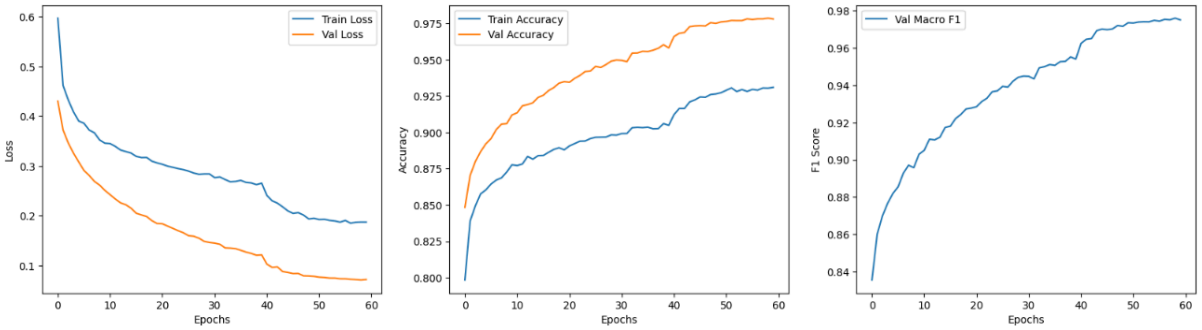
Test Performance:

Test Accuracy: 0.8903,
Precision: 0.8903,
Recall: 0.8903,
F1 Score: 0.8900

Model-2

lr: 0.005

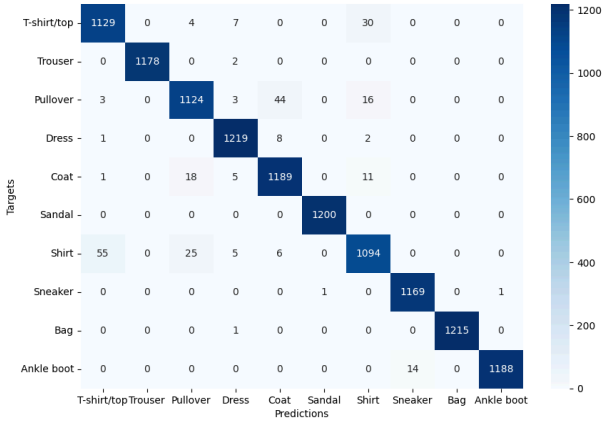
epoch: 60



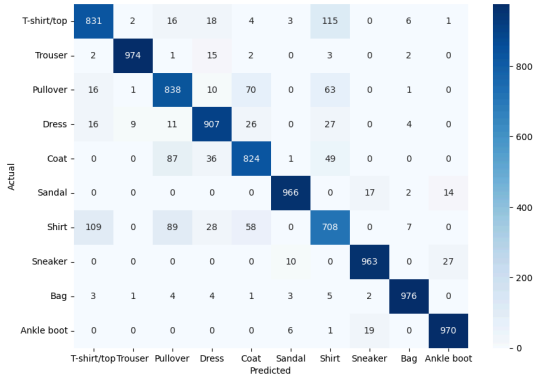
Training and validation performance:

Train Loss: 0.1872, Val Loss: 0.0717,
Train Acc: 0.9310, Val Acc: 0.9780,
Val Macro F1: 0.9751

Validation Confusion Matrix



Test Confusion Matrix



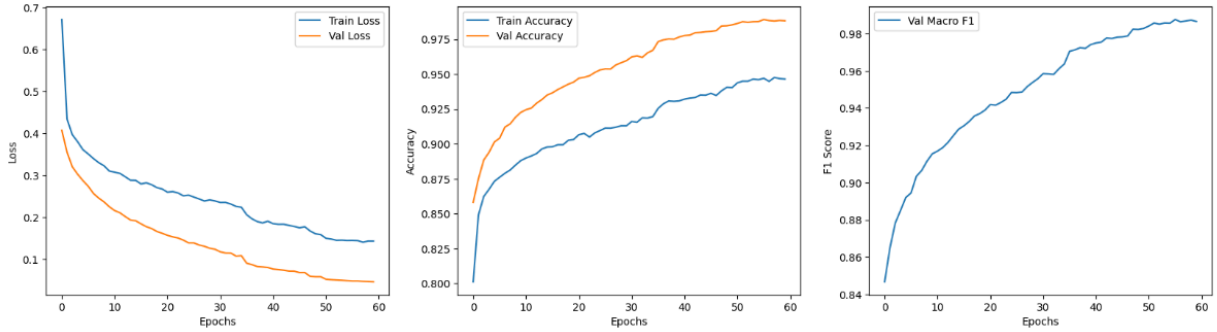
Test Performance:

Test Accuracy: 0.8971,
Precision: 0.8970,
Recall: 0.8971,
F1 Score: 0.8970

Model-2

lr: 0.001

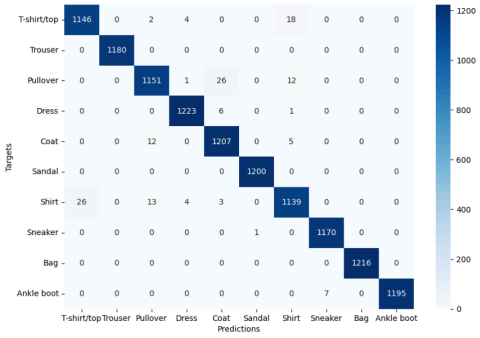
epoch: 60



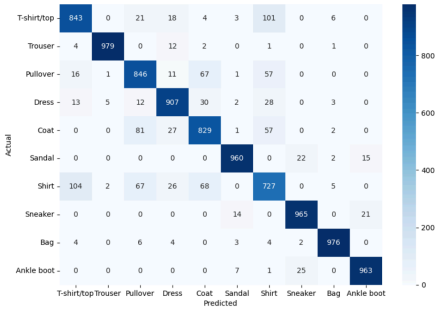
Training and validation performance:

Train Loss: 0.1427, Val Loss: 0.0456,
Train Acc: 0.9465, Val Acc: 0.9882,
Val Macro F1: 0.9865

Validation Confusion Matrix



Test Confusion Matrix



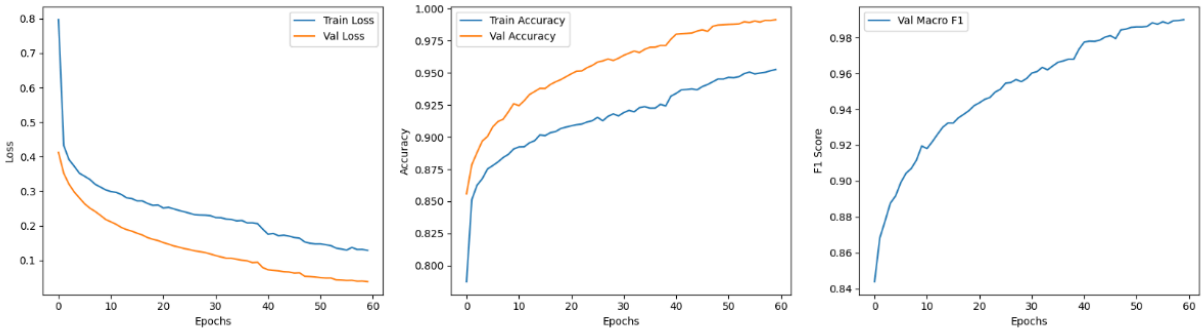
Test Performance:

Test Accuracy: 0.9009,
Precision: 0.9009,
Recall: 0.9009,
F1 Score: 0.9009

Model-2

lr: 0.0005

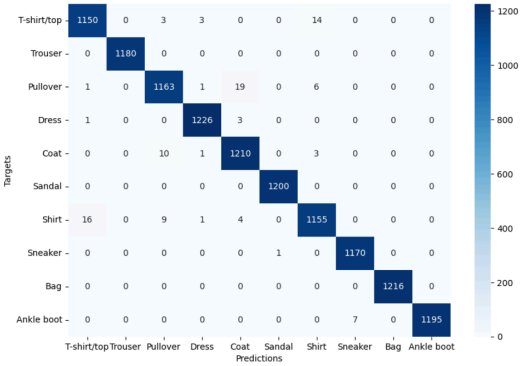
epoch: 60



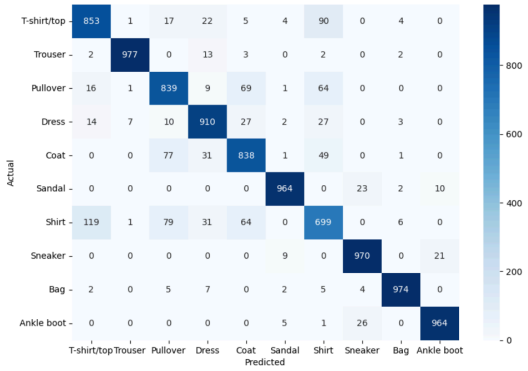
Training and validation performance:

Train Loss: 0.1289, Val Loss: 0.0384,
Train Acc: 0.9526, Val Acc: 0.9914,
Val Macro F1: 0.9899

Validation Confusion Matrix



Test Confusion Matrix



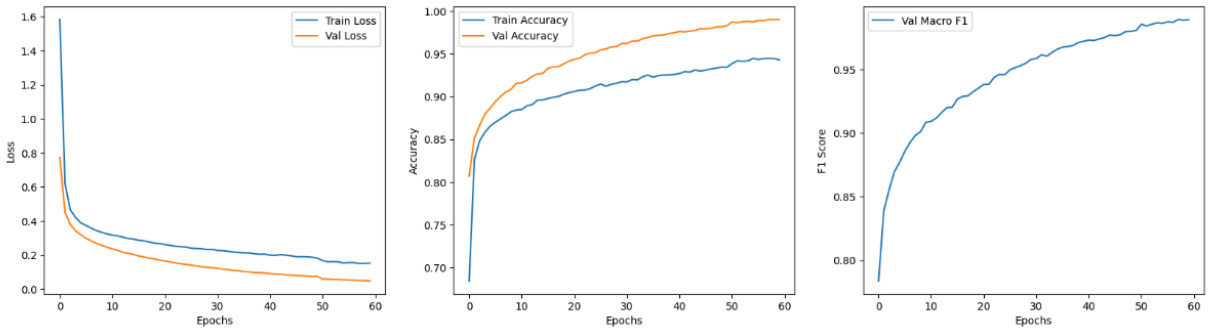
Test Performance:

Test Accuracy: 0.9002,
Precision: 0.8998,
Recall: 0.9002,
F1 Score: 0.8999

Model-2

lr: 0.0001

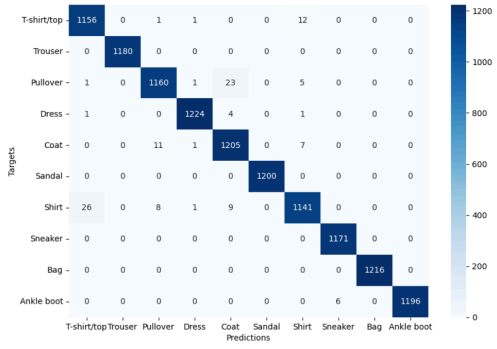
epoch: 60



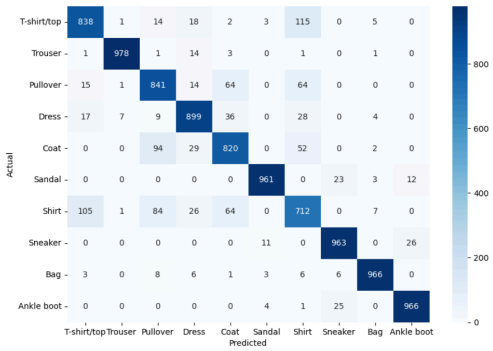
Training and validation performance:

Train Loss: 0.1524, Val Loss: 0.0480,
Train Acc: 0.9429, Val Acc: 0.9901,
Val Macro F1: 0.9889

Validation Confusion Matrix



Test Confusion Matrix



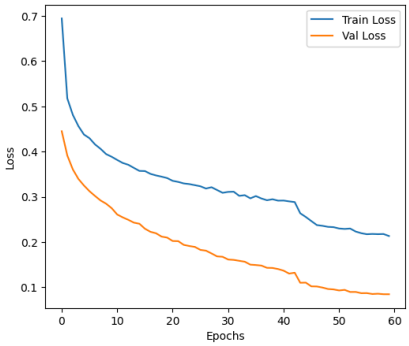
Test Performance:

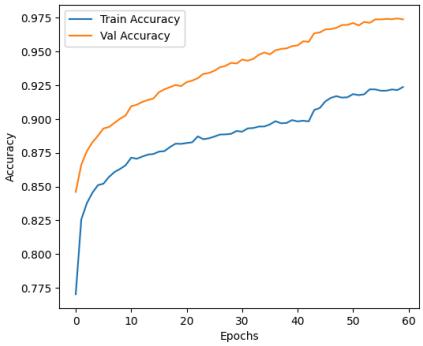
Test Accuracy: 0.8958,
Precision: 0.8961,
Recall: 0.8958,
F1 Score: 0.8959

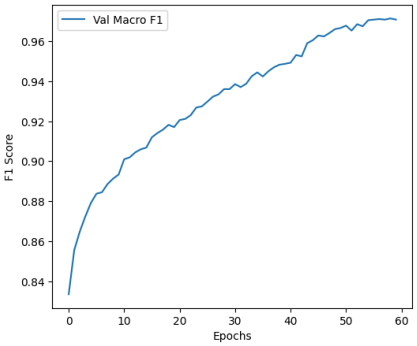
Model-3

lr: 0.005

epoch: 60



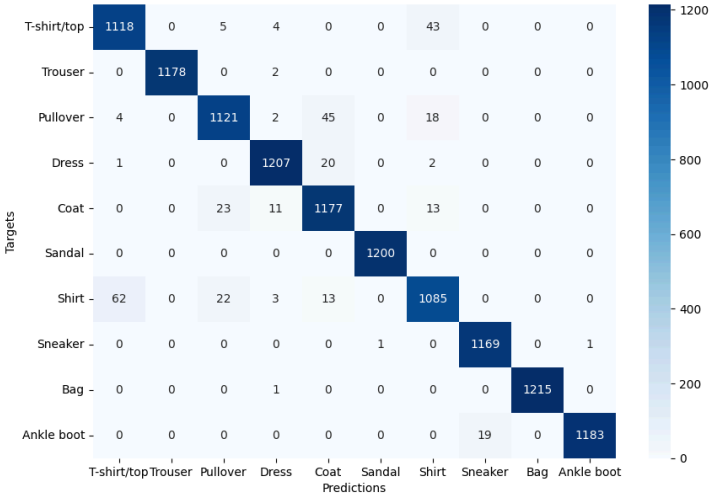




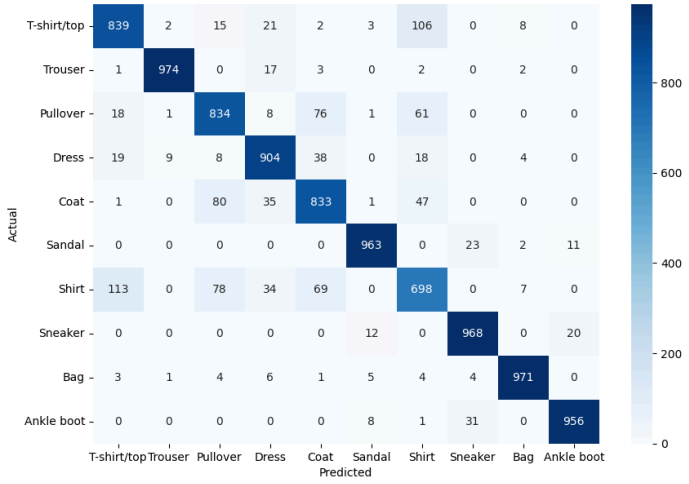
Training and validation performance:

Train Loss: 0.2133, Val Loss: 0.0846,
Train Acc: 0.9236, Val Acc: 0.9737,
Val Macro F1: 0.9709

Validation Confusion Matrix



Test Confusion Matrix



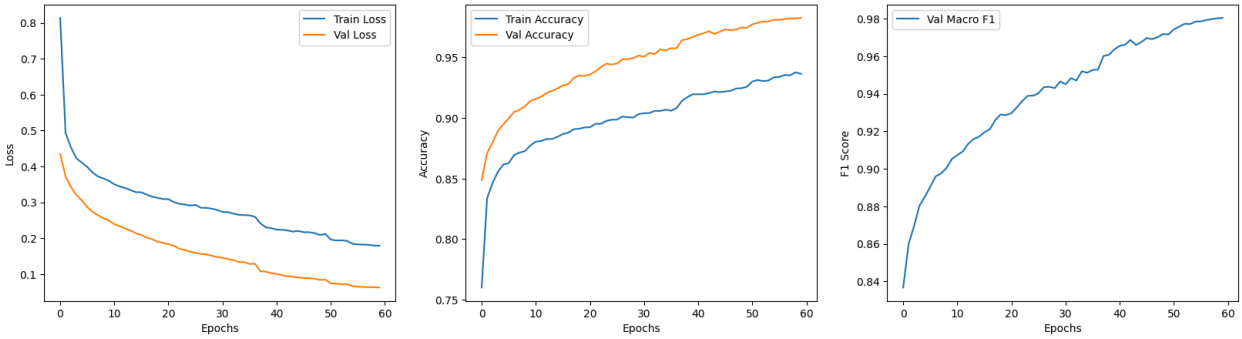
Test Performance:

Accuracy: 0.8954,
Precision: 0.8950,
Recall: 0.8954,
F1 Score: 0.8951

Model-3

lr: 0.001

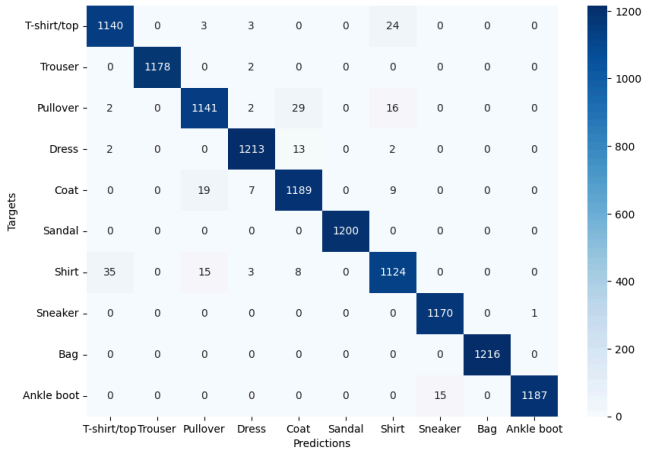
epoch: 60



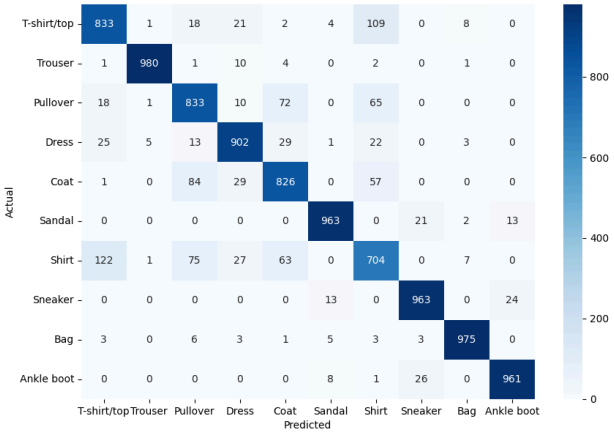
Training and validation performance:

Train Loss: 0.1791, Val Loss: 0.0624,
Train Acc: 0.9364, Val Acc: 0.9825,
Val Macro F1: 0.9804

Validation Confusion Matrix



Test Confusion Matrix



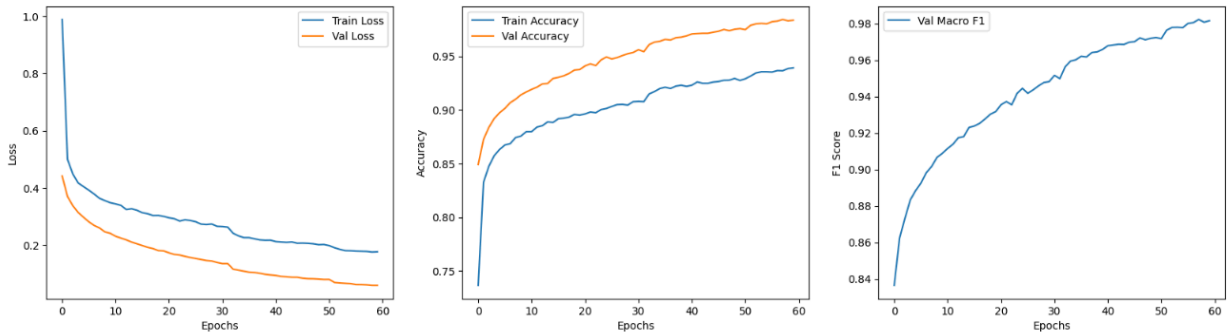
Test Performance:

Test Accuracy: 0.8954,
Precision: 0.8952,
Recall: 0.8954,
F1 Score: 0.8953

Model-3

lr: 0.0005

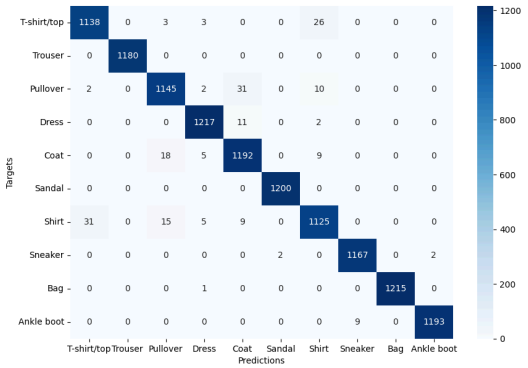
epoch: 60



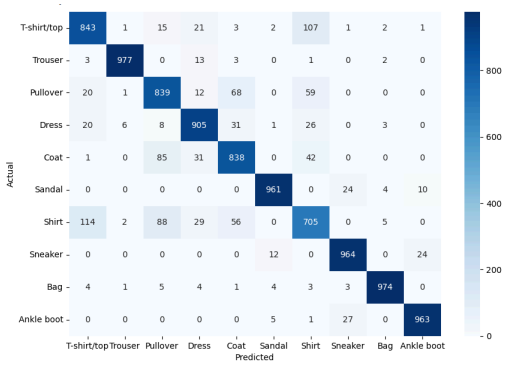
Training and validation performance:

Train Loss: 0.1769, Val Loss: 0.0596,
Train Acc: 0.9393, Val Acc: 0.9836,
Val Macro F1: 0.9815

Validation Confusion Matrix



Test Confusion Matrix



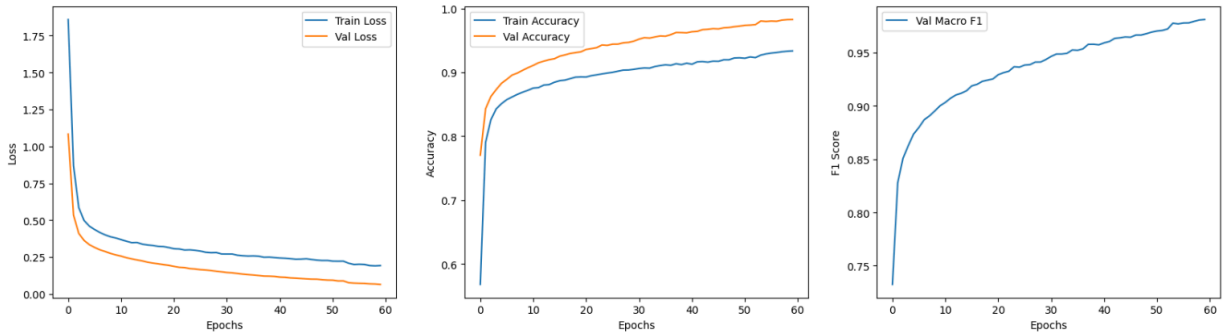
Test Performance:

Test Accuracy: 0.8983,
Precision: 0.8981,
Recall: 0.8983,
F1 Score: 0.8981

Model-3

lr: 0.0001

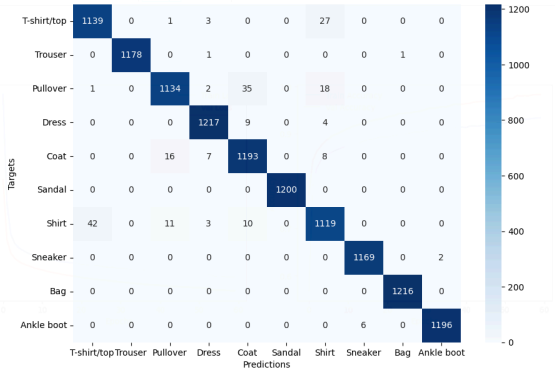
epoch: 60



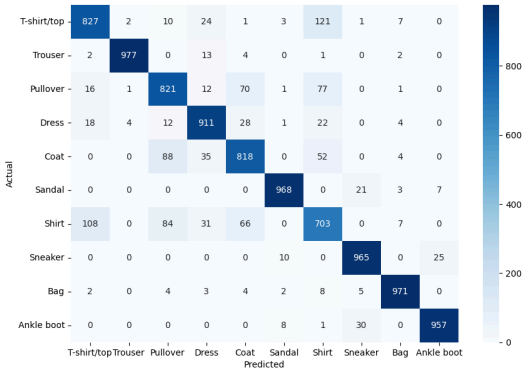
Training and validation performance:

Train Loss: 0.1927, Val Loss: 0.0650,
Train Acc: 0.9335, Val Acc: 0.9827,
Val Macro F1: 0.9812

Validation Confusion Matrix



Test Confusion Matrix



Test Performance:

Test Accuracy: 0.8932,
Precision: 0.8932,
Recall: 0.8932,
F1 Score: 0.8931

Best Model

(Based on the macro f1 score on the validation set)

Model-2

lr: 0.0005

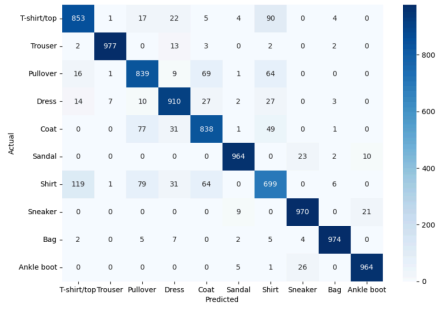
epoch: 60

Training and validation performance:

Train Loss: 0.1289, Val Loss: 0.0384,
Train Acc: 0.9526, Val Acc: 0.9914,
Val Macro F1: 0.9899

```
layers = [
    Dense(input_dim, 512),
    Batchnorm(512),
    ReLU(),
    Dropout(0.5),
    Dense(512, 256),
    Batchnorm(256),
    ReLU(),
    Dropout(0.5),
    Dense(256, output_dim),
    SoftMax()
]
```

Test Confusion Matrix



Test Performance:

Test Accuracy: 0.9002,
Precision: 0.8998,
Recall: 0.9002,
F1 Score: 0.8999