A report on Convolutional Neural Network implemented with numpy

Ayan Antik Khan

1705036

CSE, BUET

CSE 472 Machine Learning Sessional

February 9th, 2023

Model Architecture: Lenet-5

1. Conv (6 filters of size 3x3 with stride = 1, padding = 0)

2. ReLU

3. Max-pooling (2x2 with stride = 2)

4. Conv(16 filters of size 3x3 with stride = 1, padding = 0)

5. ReLU

6. Max-pooling (2x2 with stride = 2)

7. Flatten

8. Fully-connected layer (120 dimensional output)

9. ReLU

10. Fully-connected layer (84 dimensional output)

11. ReLU

12. Fully-connected layer (10 dimensional output)

13. SoftMax layer

Results

Model 1

Learning Rate = 0.01, epochs = 30

Image preprocessing

1. Reshaped to (28x28)

2. Colors flipped by subtracting from 255

Results

Train Loss: **0.2768** | Train Accuracy: **0.9113**

Validation Loss: **0.3500** | Val Accuracy: **0.8877**
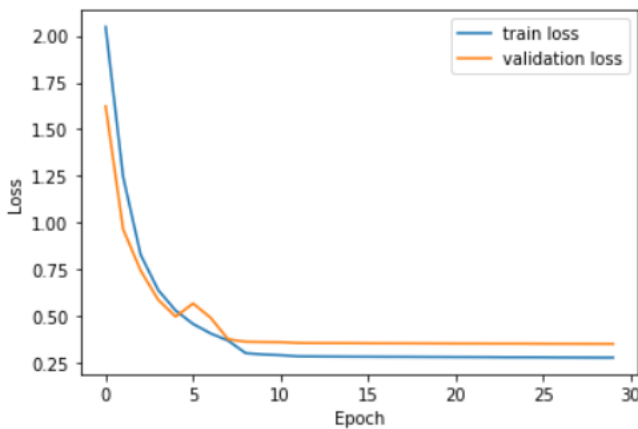
Validation f1(macro): **0.8877**
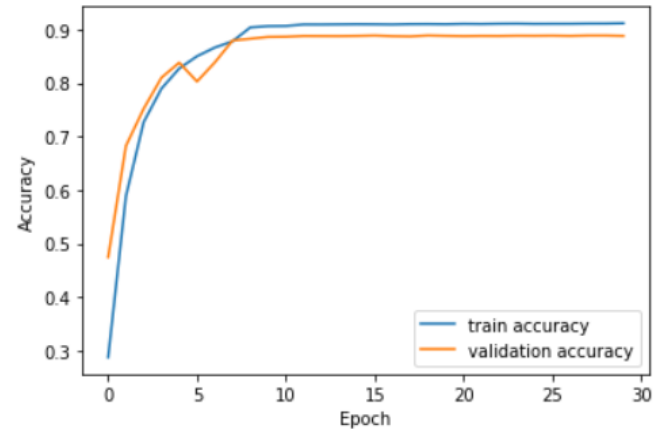

Fig: Loss Graph

Fig: Accuracy Graph

Performance on test set (*training-d)*

Accuracy: **0.8382**

F1 Score: **0.8379**

Model 2

Learning Rate = 0.001 from epochs(0-30), 0.0001 from epochs(30-60)

epochs = 60

batch size = 32

Image preprocessing

1. Reshaped to (28x28)

2. Colors flipped by subtracting from 255

3. Images dilated by using a (5x5) filter

Results

Train Loss: **0.5171** | Train Accuracy: **0.8294**

Validation Loss: **0.5984** | Validation Accuracy: **0.8069**
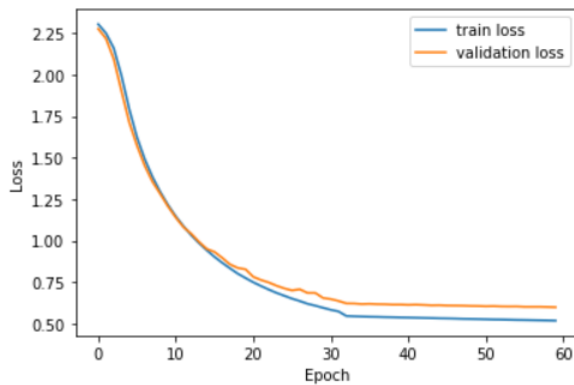
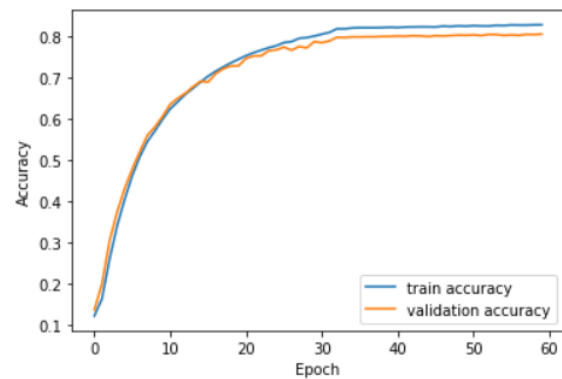Validation f1(macro): **0.8064**

Fig: Loss Graph

Fig: Accuracy Graph

Performance on test set (*training-d)*

Accuracy: **0.7648**

F1 Score: **0.7607**

Model 3

Learning Rate = 0.001 from epochs(0-30), 0.0001 from epochs(30-60)

epochs = 60

batch size = 32

Image preprocessing

1. Reshaped to (28x28)

2. Colors flipped by subtracting from 255

3. Images dilated by using a (3x3) filter

Results

Train Loss: **0.4691** | Train Accuracy: **0.8508**

Validation Loss: **0.5328** | Validation Accuracy: **0.8322**
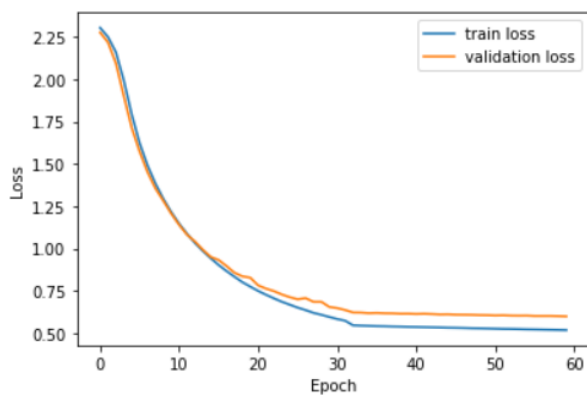
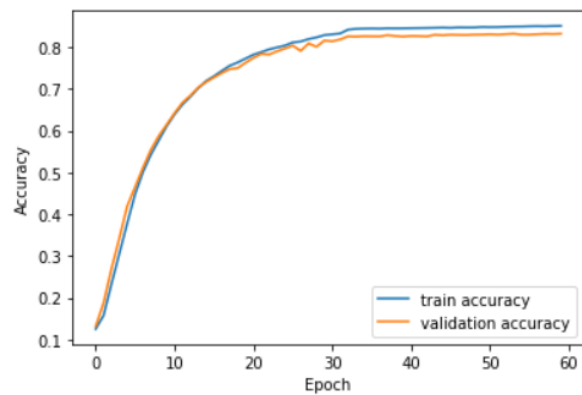Validation f1(macro): **0.8319**

Fig: Loss Graph   Fig: Accuracy Graph

Performance on test set (*training-d)*

Accuracy: **0.7921**

F1 Score: **0.7654**

Model 4

Learning Rate = 0.001 from epochs(0-30), 0.0001 from epochs(30-60)

epochs = 60

batch size = 32

Image preprocessing

1. Reshaped to (28x28)

2. Colors flipped by subtracting from 255

3. Images dilated by using a (5x5) filter

Results

Train Loss: **0.3611** | Train Accuracy: **0.8816**

Validation Loss: **0.4675** | Validation Accuracy: **0.8513**
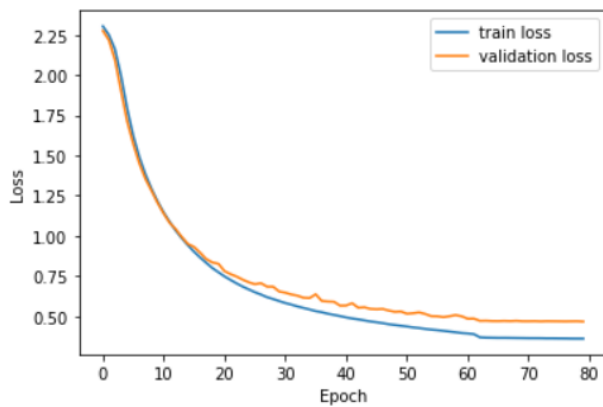
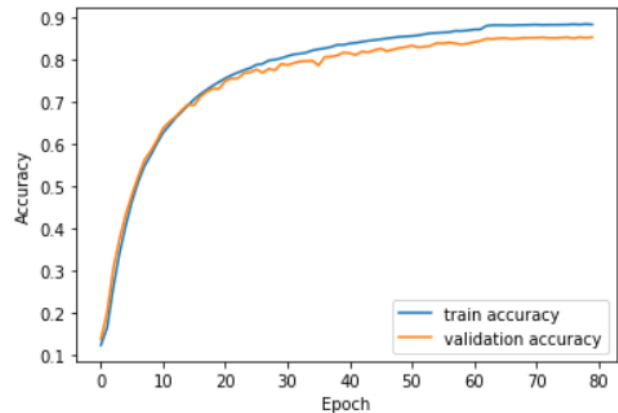Validation f1(macro): **0.8511**

Fig: Loss Graph

Fig: Accuracy Graph

Performance on test set (*training-d*)

Accuracy: **0.7925**

F1 Score: **0.7880**

Model 5

Learning Rate =0.01 from epochs(0-10), 0.001 from epochs(10-70), 0.005 from epochs(70-90)

Epochs: 90

Batch size: 32

Image preprocessing

1. Reshaped to (28x28)

2. Colors flipped by subtracting from 255

Results

```
Train Loss: 0.1133 | Train Accuracy: 0.9579

Validation Loss: 0.2604 | Validation Accuracy: 0.9180

              Validation f1(macro): 0.9187
```
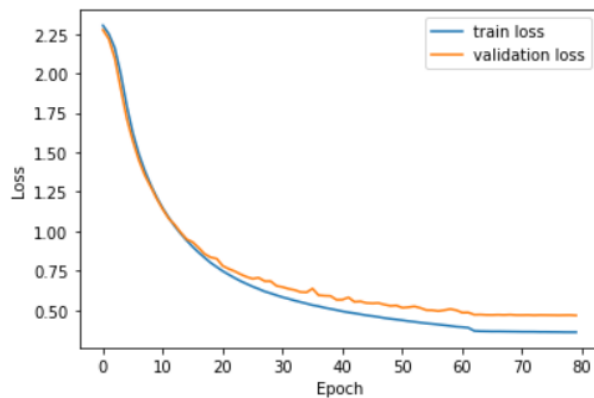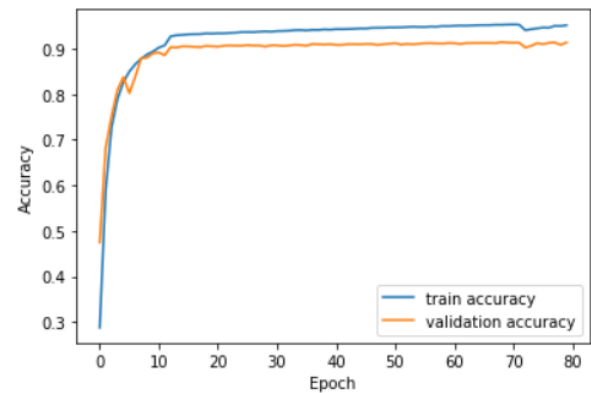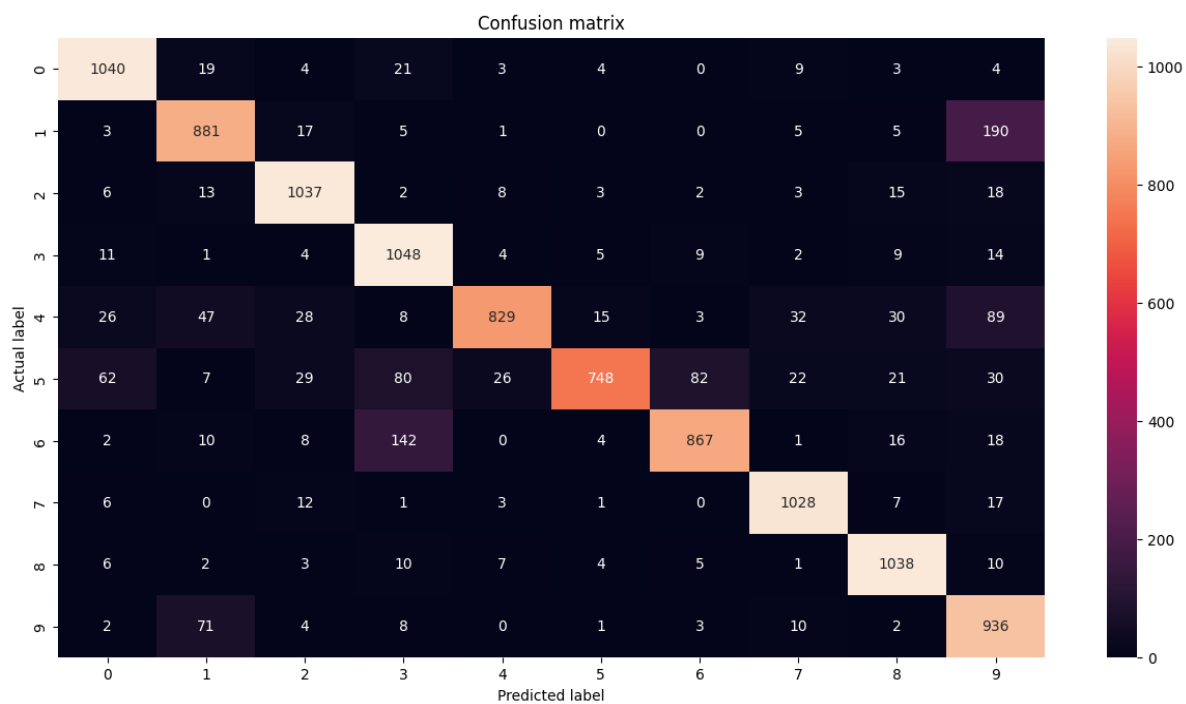

Fig: Loss Graph

Fig: Accuracy Graph

Fig: Confusion Matrix

Performance on test set (*training-d*)

Accuracy: **0.8665**

F1 Score: **0.8652**

Model 6 (Best Performing on test dataset)

Learning Rate =0.01 from epochs(0-20), 0.005 from epochs(20-25)

Epochs: 25

Batch size: 32

Image preprocessing

1. Reshaped to (28x28)

2. Colors flipped by subtracting from 255

Results

Train Loss: **0.1343** | Train Accuracy: **0.9563**

Validation Loss: **0.2745** | Validation Accuracy: **0.9188**
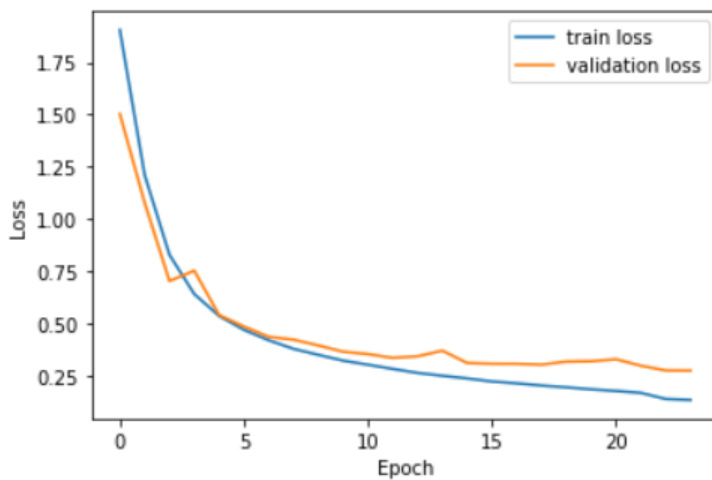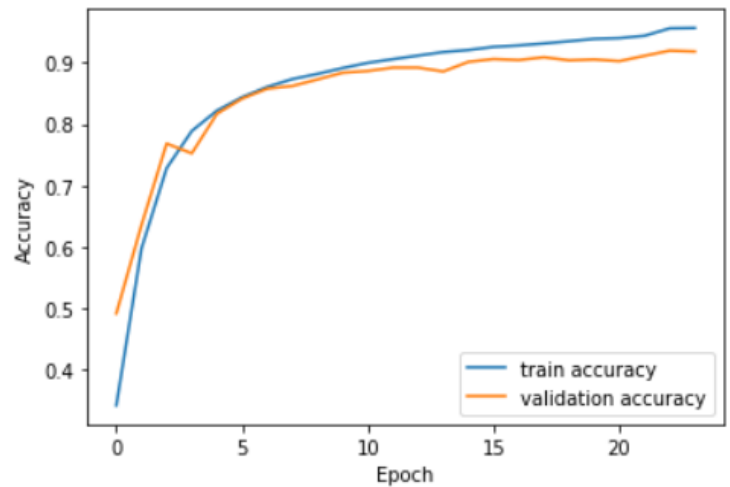
Validation f1(macro): **0.9182**

Fig: Loss Graph   Fig: Accuracy Graph

Fig: Confusion Matrix

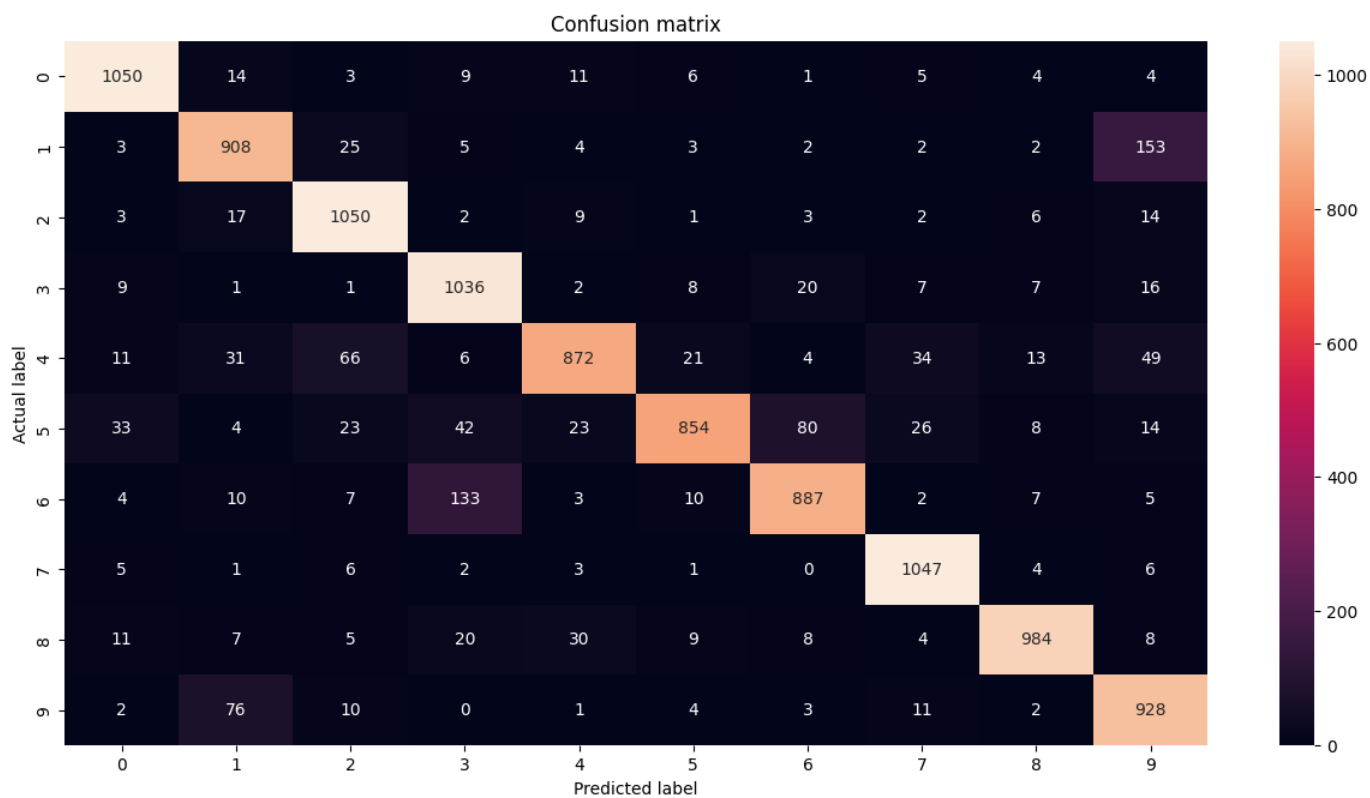## Performance on test set (*training-d*)

Accuracy: **0.8816**

F1 Score: **0.8808**