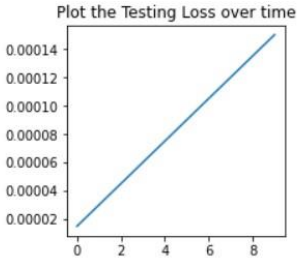
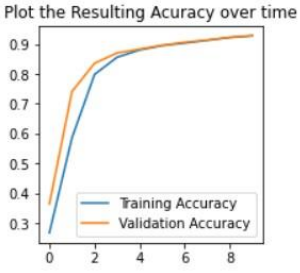
Sumbal Akram

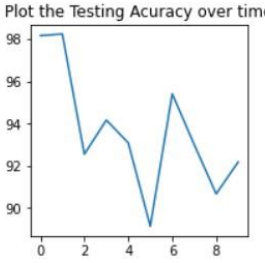
Report: Classification in Neural Network

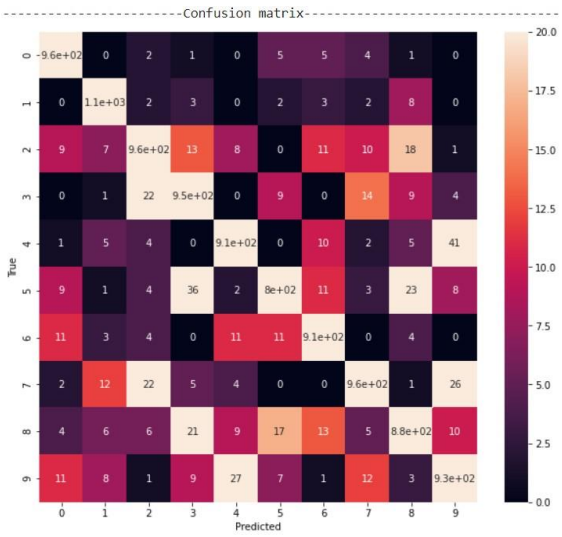
Plot loss and accuracy curves for both training and validation data with and without image normalization. Report the difference in their accuracy and loss curves. Also report accuracy and loss on test data.

Epochs = 10, Learning Rate = 0.001, Normalization = No, Batch Size= 64:









----------Training Results----------

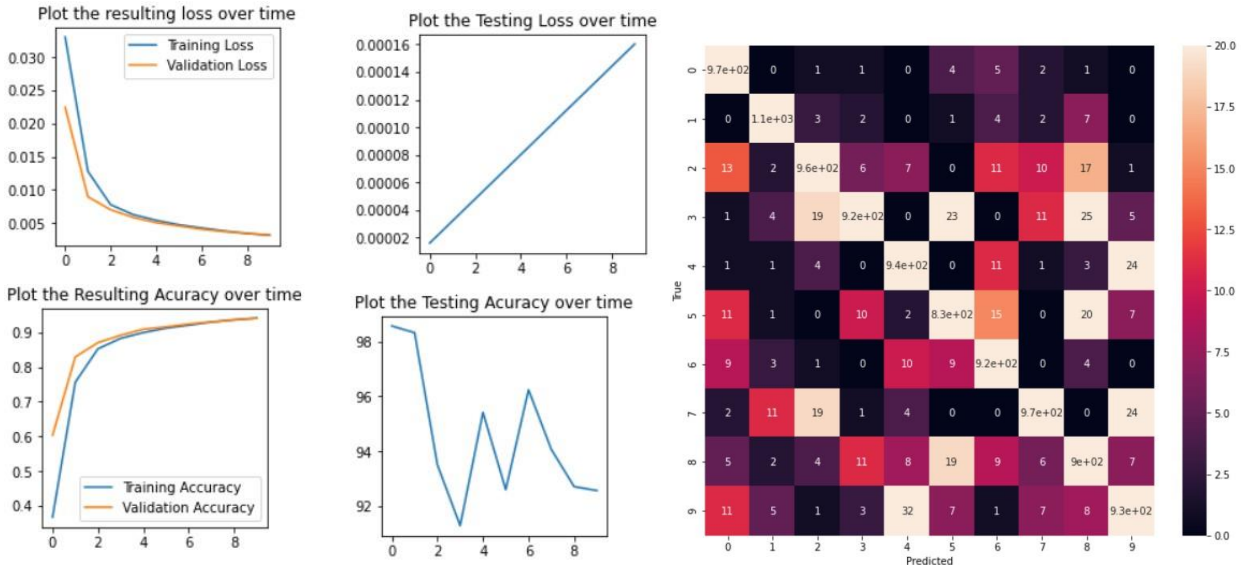
Total Training Loss = 0.0039
Total Training Accuracy = 44548/48000 (93%)
Training Time (in minutes) = 3.07

----------Testing Results----------

Total Testing Accuracy = (9375/10000) (93%)

```
--------------------------F1 & Accuracy Score-------------------------

Testing f1 score: 0.937359
Testing accuracy score: 0.937500
```

Epochs = 10, Learning Rate = 0.001, Normalization = No, Batch size = 64:



----------Training Results----------

Total Training Loss = 0.0031
Total Training Accuracy = 45183/48000 (94%)
Training Time (in minutes) = 3.99

----------Testing Results----------

Total Testing Accuracy = (9458/10000) (94%)

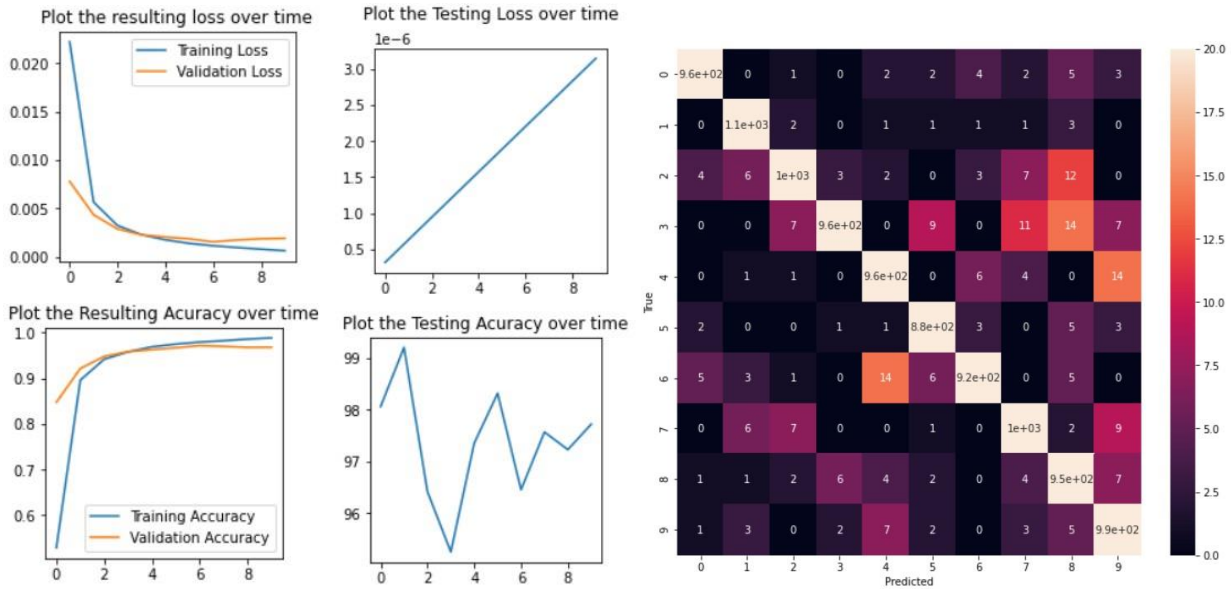
```
----------------------------F1 & Accuracy Score--------------------------

Testing f1 score: 0.945727
Testing accuracy score: 0.945800
```

Conclusion: Train-Test Accuracy increased when we normalize the images, but it takes more time to train the model when we normalize our data.

Report the accuracy by changing number of neurons in the hidden layers, or number of hidden layers or changing loss functions, batch size, learning rate and ratio of training and testing data etc.

(By Changing No of Hidden Layers and Neurons in hidden Layer)

- Hidden Layers = **4**
- Neurons in the hidden layers = **[512,256,128, 64, 10]**
- Loss Function = **Negative Log-Likelihood Loss(NLLLoss)**
- Batch Size = **64**
- Epochs = **10**
- Learning Rate = **0.005**
- Normalization = **Yes**
- Ratio of Training = **80%**
- Ratio of Validating = **20%**



----------Training Results----------

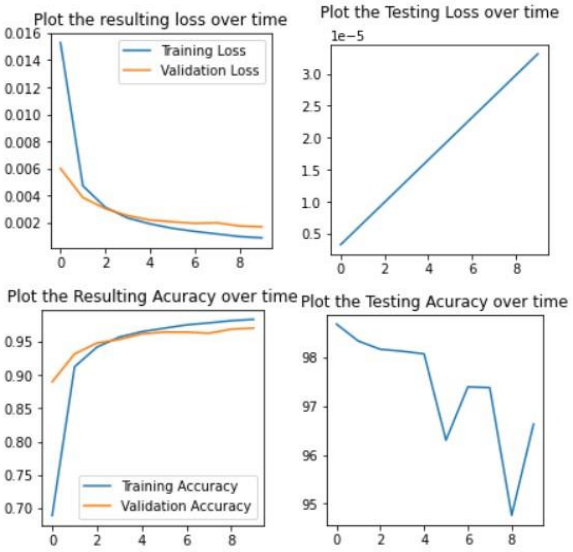
Total Training Loss = 0.0006
Total Training Accuracy = 47420/48000 (99%)
Training Time (in minutes) = 3.94

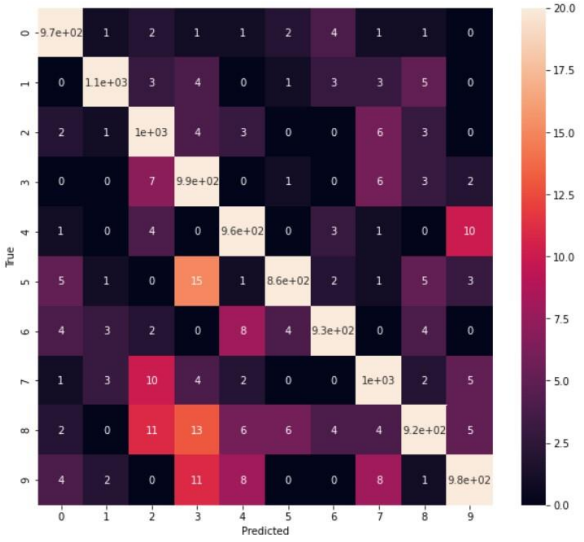
----------Testing Results

Total Testing Accuracy = (9737/10000) (97%)

```
----------F1 & Accuracy Score----------

Testing f1 score: 0.973707
Testing accuracy score: 0.973700
```

- Hidden Layers = **3**
- Neurons in the hidden layers = **[256,128, 64, 10]**
- Loss Function = **NLLLoss**
- Batch Size = **64**
- Epochs = **10**
- Learning Rate = **0.005**
- Normalization = **Yes**
- Ratio of Training = **80%**
- Ratio of Validating = **20%**





----------Training Results----------

Avg Training Loss = 0.0009
Total Training Accuracy = 47183/48000 (98%)
Training Time (in minutes) = 3.94

Testing Results----------

Test Accuracy (Overall): 97% (9741/10000)

```
----------F1 & Accuracy Score----------

Testing f1 score: 0.974100
Testing accuracy score: 0.974100
```

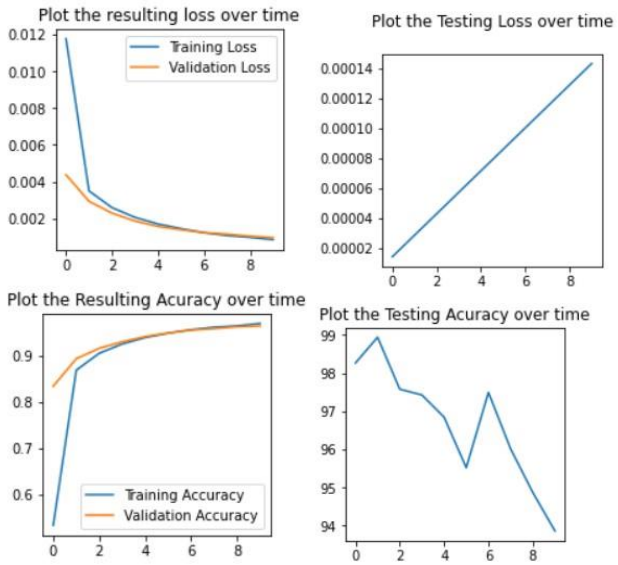
(By Changing Batch Size)

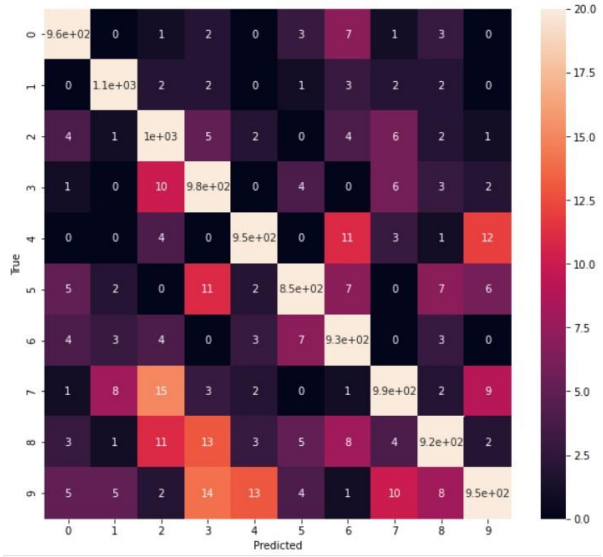
- Hidden Layers = **3**
- Neurons in the hidden layers = **[256,128, 64, 10]**
- Loss Function = **NLLoss**
- Batch Size = **128**
- Epochs = **10**
- Learning Rate = **0.005**
- Normalization = **Yes**
- Ratio of Training = **80%**
- Ratio of Validating = **20%**

----------Training Results----------

Avg Training Loss =
Total Training Accuracy
Training Time (

Testing Results----------





0.0008
= 46507/48000 (97%)
in minutes) = 3.85

Test Accuracy (Overall): 96% (9672/10000)

```
--------------------------F1 & Accuracy Score--------------------------

Testing f1 score: 0.967165
Testing accuracy score: 0.967200
```

(By Changing Learning Rate)

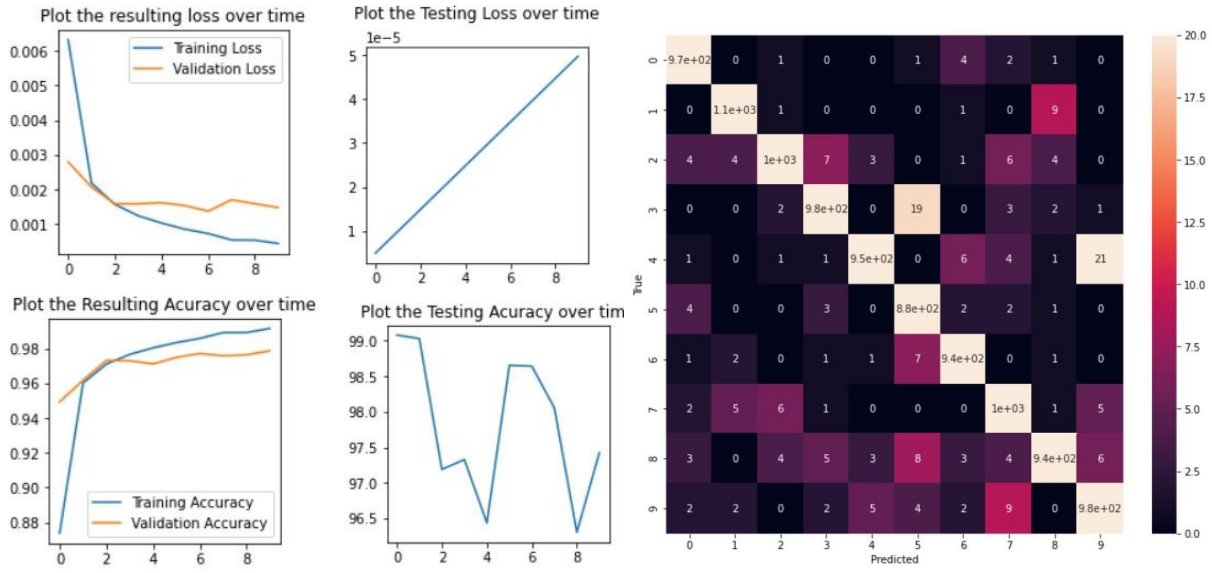
- Hidden Layers = **3**
- Neurons in the hidden layers = **[256,128, 64, 10]**
- Loss Function = **NLLLoss**
- Batch Size = **64**

----------Training Results----------

Avg Training Loss =
Total Training Accuracy
Training Time (

Testing Results----------

- Epochs = **10**
- Learning Rate = 0.05
- Normalization = Yes
- Ratio of Training = 80%
- Ratio of Validating = 20%



0.0004

= 47595/48000 (99%)

in minutes) = 4.029

Test Accuracy (Overall): 97% (9782/10000)

```
----------F1 & Accuracy Score----------

Testing f1 score: 0.978197
Testing accuracy score: 0.978200
```

(By Changing Training Size)

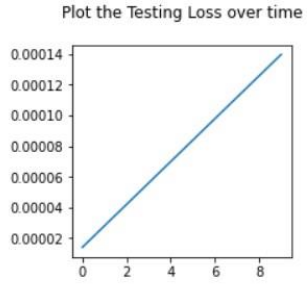
----------Training Results----------

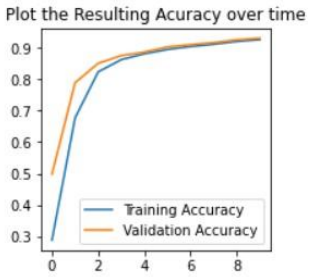
Avg Training Loss =
Total Training Accuracy
Training Time (

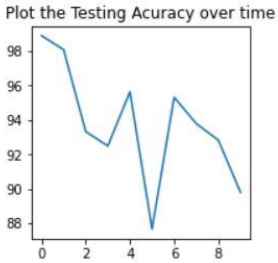
Testing Results----------

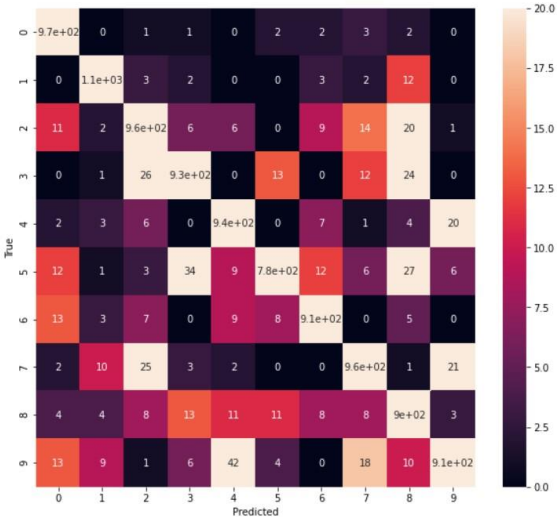
- Hidden Layers = 3
- Neurons in the hidden layers = [128, 64, 10]
- Loss Function = NLLLoss
- Batch Size = 64
- Epochs = 10
- Learning Rate = 0.005
- Normalization = Yes
- Ratio of Training = 70%
- Ratio of Validating = 30%

```
60000 Original Train Data
10000 Length of Test Data
42000 Length of Training Data
18000 Length of Validation Data
```









0.0040

= 38876/42000 (93%)

in minutes) = 4.08

----------Training Results----------

Avg Training Loss =
Total Training Accuracy
Training Time (

Testing Results----------

Test Accuracy (Overall): 93% (9387/10000)

```
----------F1 & Accuracy Score----------

Testing f1 score: 0.938532
Testing accuracy score: 0.938700
```

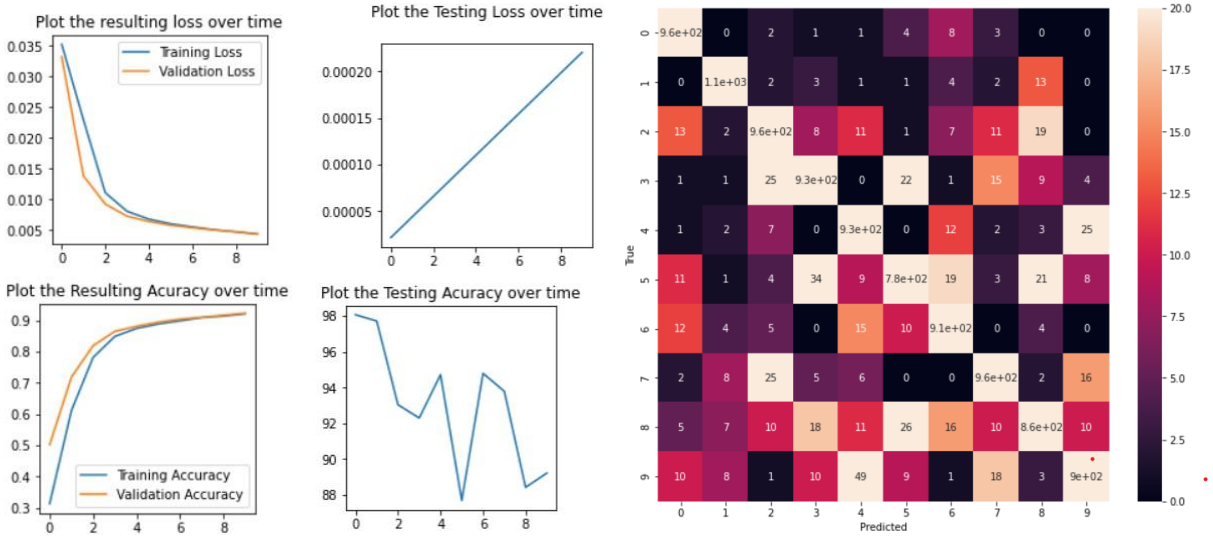
- Hidden Layers = 2
- Neurons in the hidden layers = [128, 64, 10]
- Loss Function = NLLLoss
- Batch Size = 64
- Epochs = 10
- Learning Rate = 0.005
- Normalization = Yes
- Ratio of Training = 60%
- Ratio of Validating = 40

```
60000 Original Train Data
10000 Length of Test Data
36000 Length of Training Data
24000 Length of Validation Data
```

----------Training Results----------

Avg Training Loss =
Total Training Accuracy
Training Time (

Testing Results----------



0.0043
= 3134/36000 (92%)
in minutes) = 3.90

Test Accuracy (Overall): 93% (9307/10000)

```
----------F1 & Accuracy Score----------

Testing f1 score: 0.930479
Testing accuracy score: 0.930700
```

For your final experiment display actual images and predicted labels for both cases if prediction is correct or wrong. If prediction is wrong show on what basis the model predicted it wrong and also show the worst prediction from the model. For example, if the model has predicted 3 as 8, prediction is wrong but both digits are similar in shape. But if the

----------Training Results----------

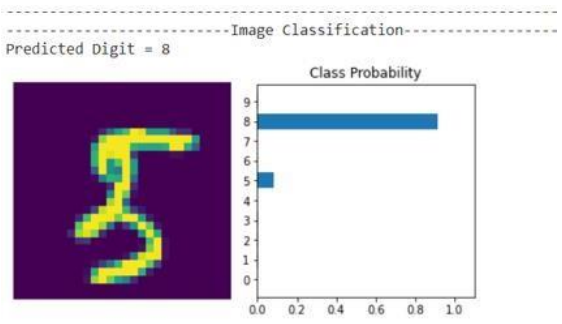
Avg Training Loss =
Total Training Accuracy
Training Time (

Testing Results----------

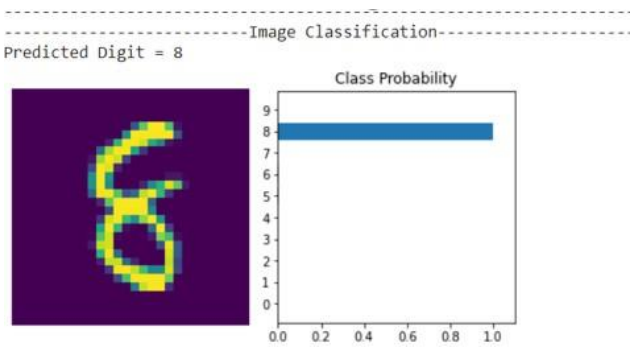
model has predicted 0 as 1 then this prediction can be worse as both digits do not have similar patterns in it. Also display confusion matrix on testing data and report Precision, Recall and F1 score align with loss and accuracy scores and curves.

No.of layers = 3, , learning rate = 0.005, Normalized, Epoch= 10

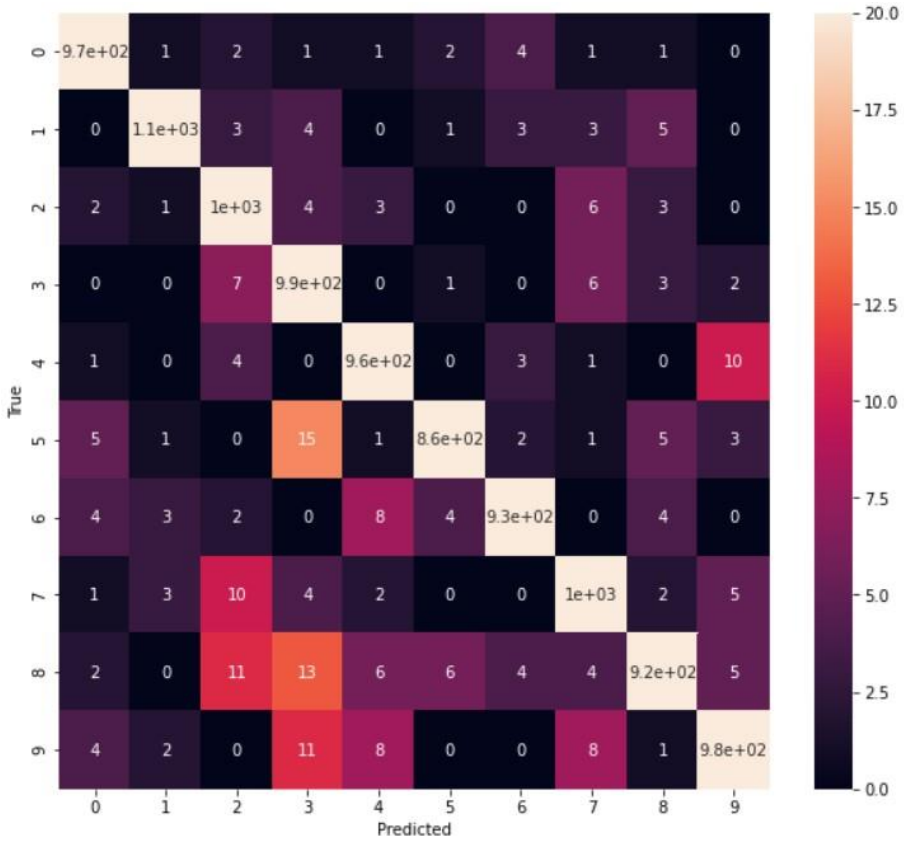
Wrong Predicted



True Predicted Image



Confusion Matrix Describe number of right predicted images of test data in each Epoch:



On the basis of Probability images are classified. Above matrix shows the true classified images in each Epoch.

Report what you learned from this assignment, your analysis and if you find something innovative or interesting in the conclusion section.

- I have learned how dynamic neural network works.
- How to split training data for training and validation.
- Factors effect the accuracy of the network i.e. (CPU, GPU, no of layers, no of neurons, batch size, learning rate and no of epochs).
- Classify the test data on basis of their probability.
- How to save and load the models.
- Pytorch and its use.

Discuss time effect on time taken by network in different experiments or when using CPU or GPU.

Each factor effects the training time but Device used (CPU or GPU) effects the most. Cpu takes couple of hours to train the data on the other hand GPU provides fast data training. Our data trained in between 3 to 4 min.

Epochs = 10, Learning Rate = 0.001, Normalization = No, Batch Size= 64: Training Time (in minutes) = 3.07
Epochs = 10, Learning Rate = 0.005, Normalized, Batch Size= 64: Training Time (in minutes) = 3.94
Epochs = 10, Learning Rate = 0.005, Normalized, Batch Size= 128: Training Time (in minutes) = 3.85
Epochs = 10, Learning Rate = 0.05, Normalized, Batch Size= 64: Training Time (in minutes) = 4.029
Epochs = 15, Learning Rate = 0.001, Normalized, Batch Size= 64: Training Time (in minutes) = 5.95