Deep Learning Spring 2022
Assignment 2 Report

Ali Khalid
MSDS21001

May 11, 2022

1 Loss and Accuracy Curves for validation and training dataset.

1.1 Task1

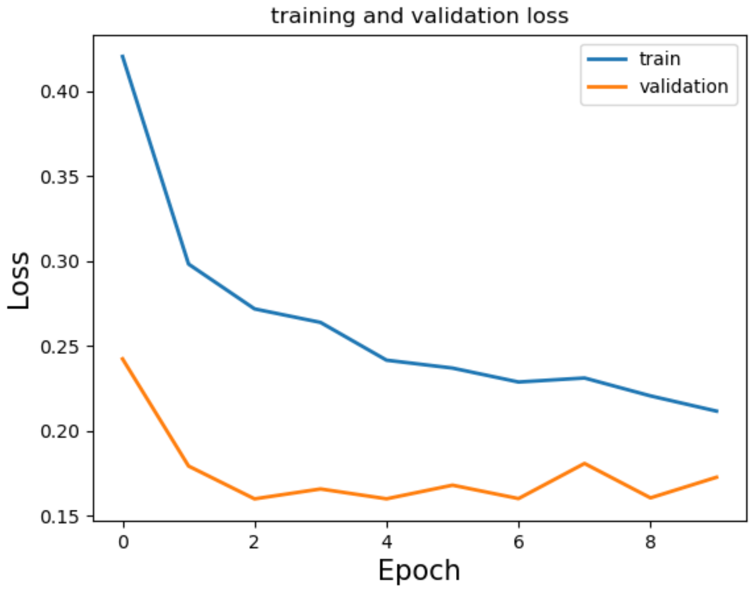


Figure 1: Training and Validation Loss Curve

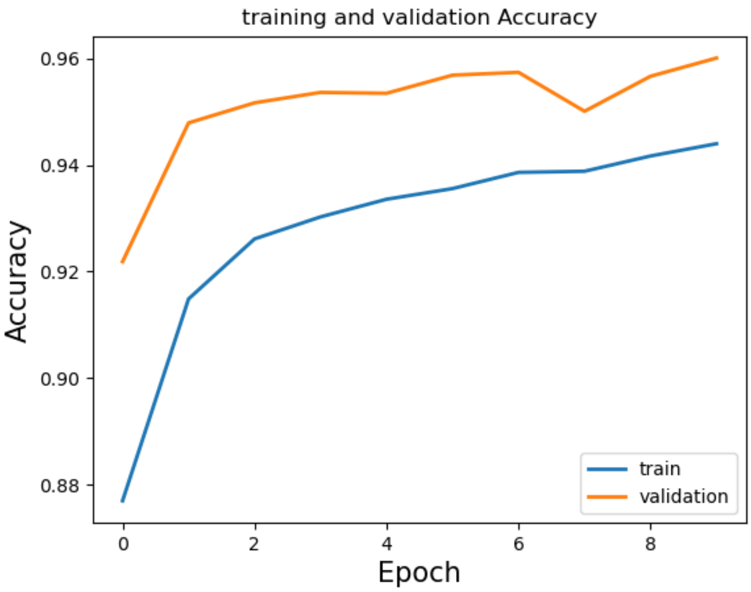


Figure 2: Training and Validation Accuracy Curve

1.2 Task2

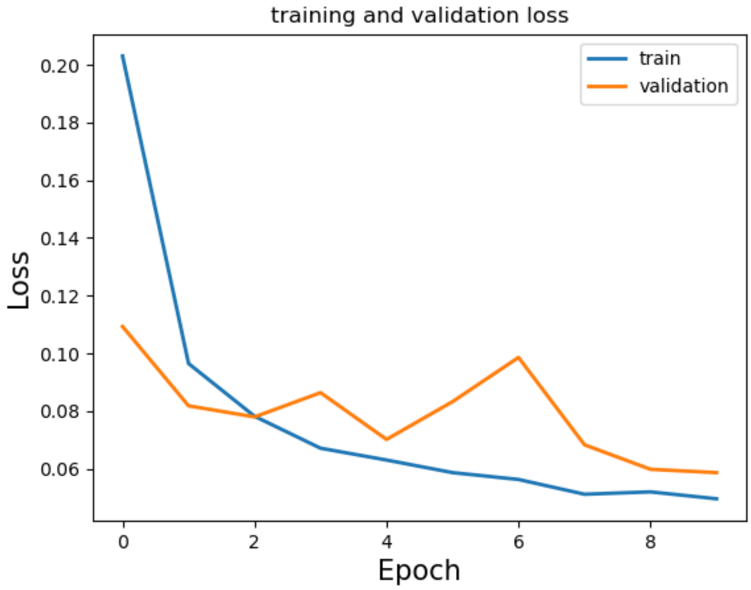


Figure 3: Training and Validation Loss Curve

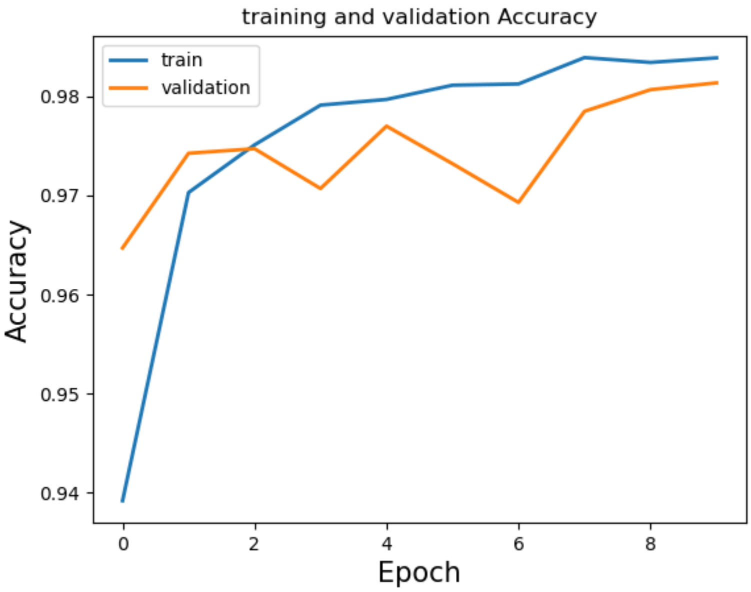


Figure 4: Training and Validation Accuracy Curve

2 Confusion matrices, Recall, and Accuracy for the testing set.

2.1 Task1

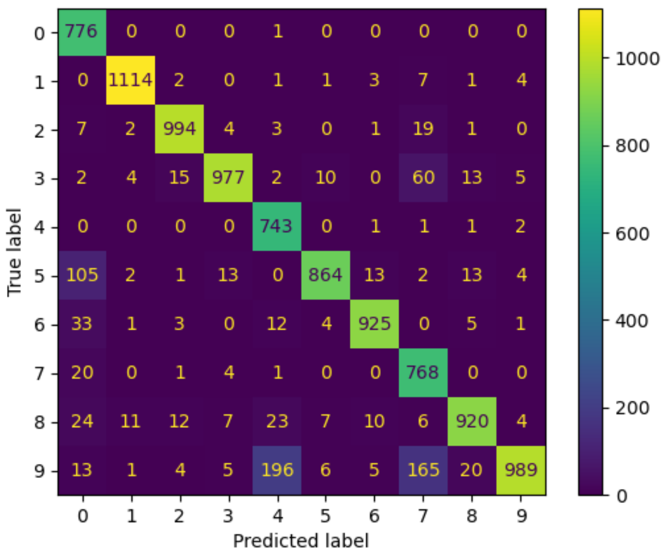
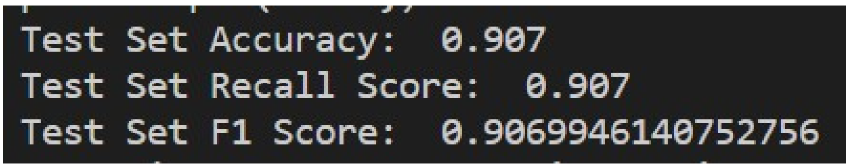


Figure 5: Confusion Matrix



```
Test Set Accuracy:  0.907
Test Set Recall Score:  0.907
Test Set F1 Score:  0.9069946140752756
```

Figure 6: Recall, Accuracy, f1 score

2.2 Task2

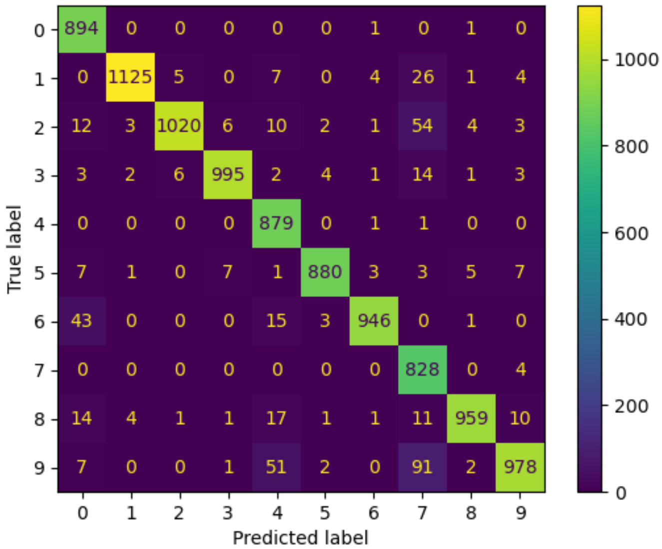


Figure 7: Confusion Matrix

```
Test Set Accuracy:  0.9504
Test Set Recall Score:  0.9504
Test Set F1 Score:  0.950939506568197
```

Figure 8: Recall, Accuracy, f1 score

3 Figures along with labels for correct predictions and wrong ones

3.1 Task1

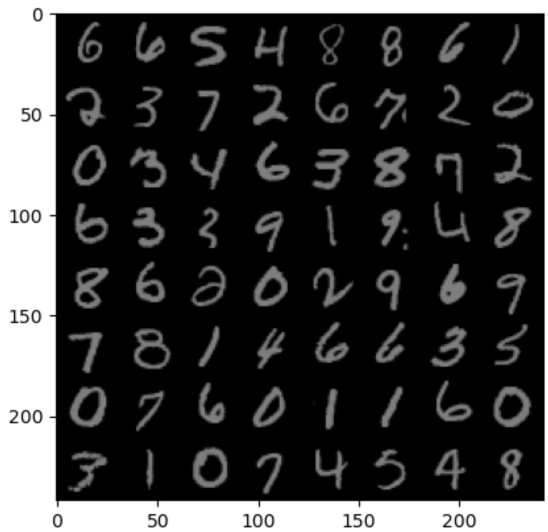


Figure 9: images

```
tensor([6, 6, 5, 4, 8, 8, 6, 1, 2, 3, 7, 2, 6, 7, 2, 0, 0, 3, 4, 6, 3, 8, 7, 2,
        6, 3, 3, 9, 1, 9, 4, 8, 8, 6, 2, 0, 2, 9, 6, 9, 7, 8, 1, 4, 6, 6, 3, 5,
        0, 7, 6, 0, 1, 1, 6, 0, 3, 1, 0, 7, 4, 5, 4, 8])
```

Figure 10: predicted labels

3.2 Task2

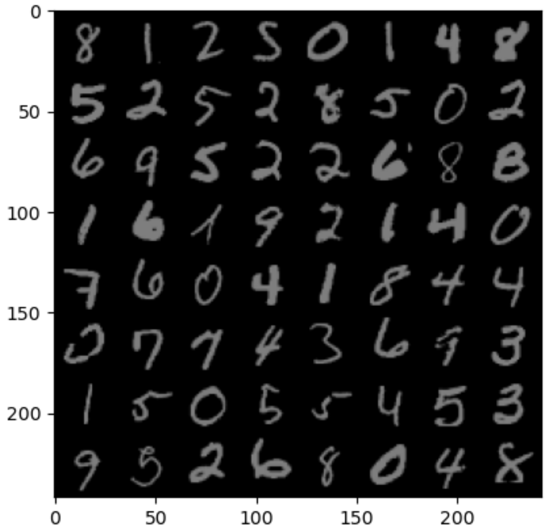
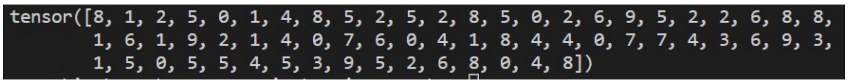


Figure 11: images



```
tensor([8, 1, 2, 5, 0, 1, 4, 8, 5, 2, 5, 2, 8, 5, 0, 2, 6, 9, 5, 2, 2, 6, 8, 8,
        1, 6, 1, 9, 2, 1, 4, 0, 7, 6, 0, 4, 1, 8, 4, 4, 0, 7, 7, 4, 3, 6, 9, 3,
        1, 5, 0, 5, 5, 4, 5, 3, 9, 5, 2, 6, 8, 0, 4, 8])
```

Figure 12: predicted labels

4 Plot learned filters of your last convolution layer using matplotlib.

4.1 Task2

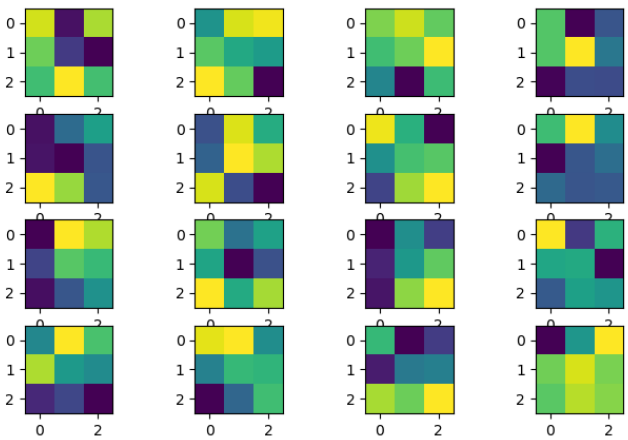


Figure 13: Last Layer filters

5 Analysis of your different experiments.

This is explained below with experimental results.

6 Show what happens when we do not use convolution layers and when we use convolution layers with neural networks. Show ROC curves, accuracy/loss curve, confusion matrix, and tsne plot.

6.1 Task1: Without convolutional Layer

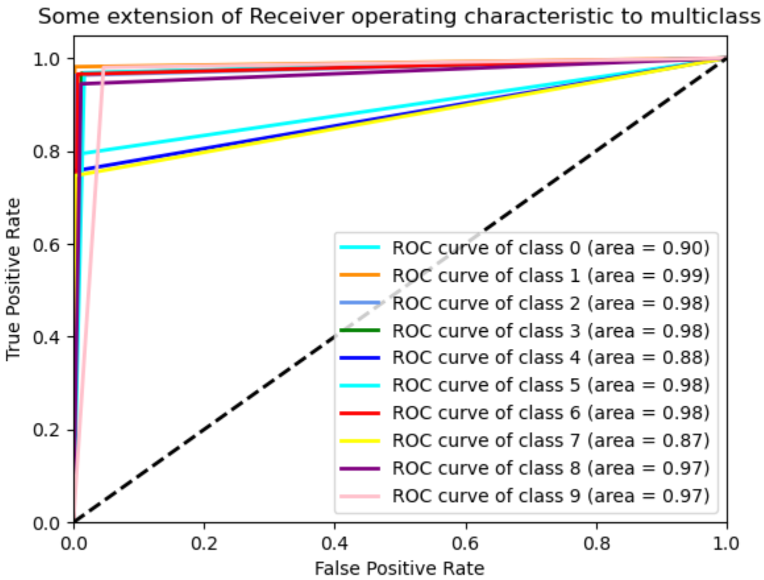


Figure 14: ROC curve

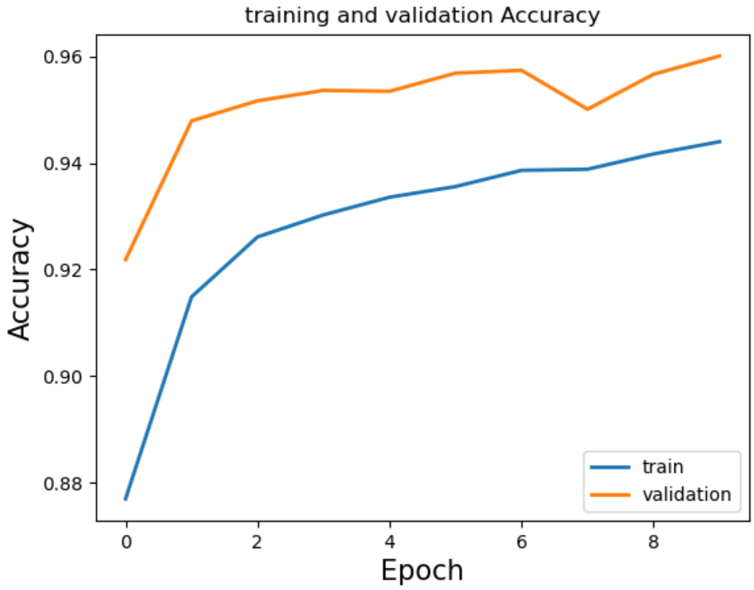


Figure 15: Accuracy curve

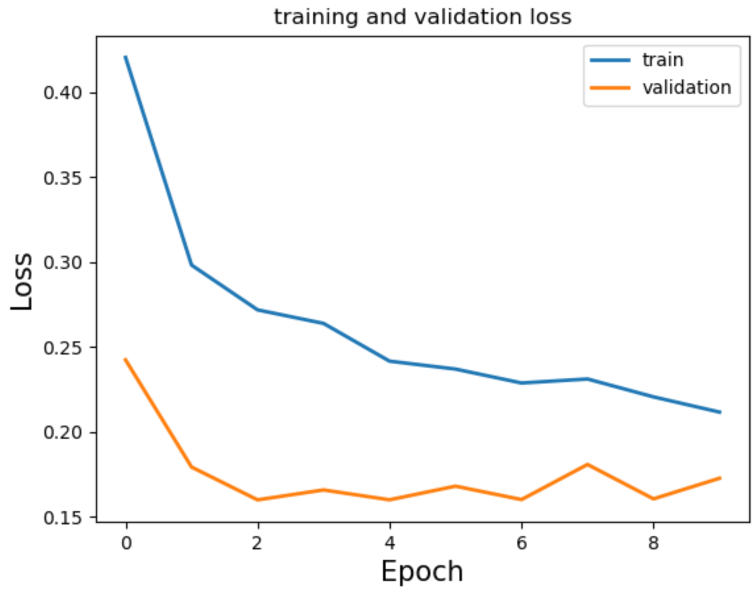


Figure 16: Loss curve

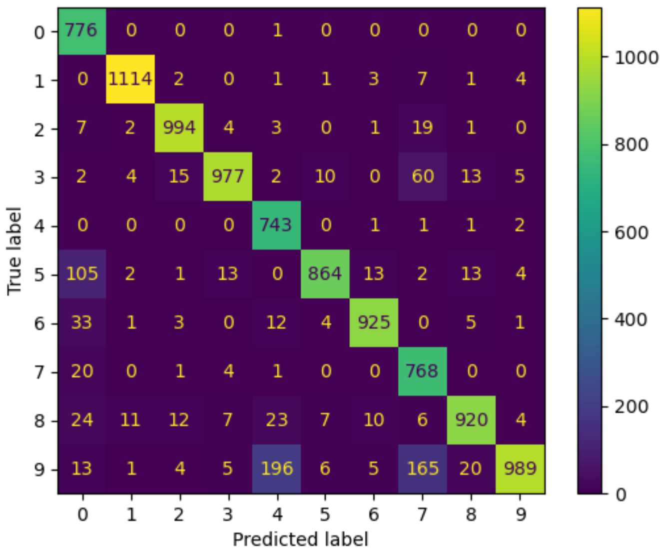


Figure 17: Confusion matrix

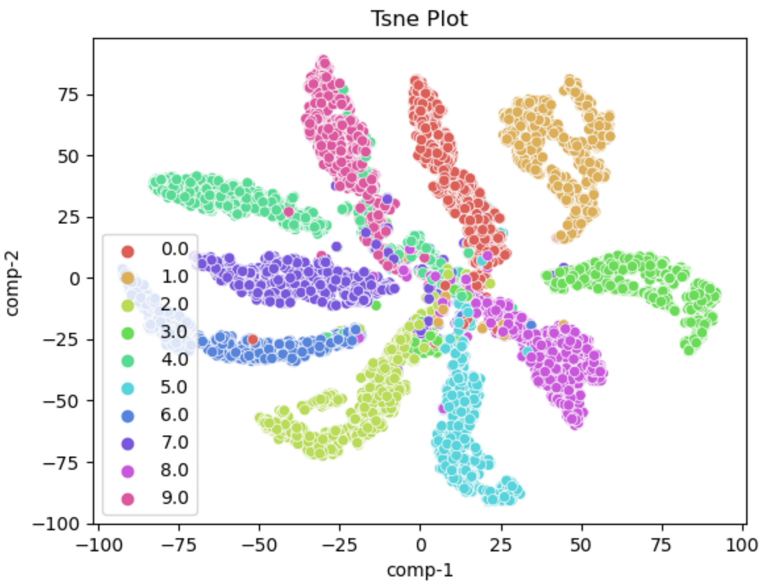


Figure 18: Tsne Plot

6.2 Task2: With convolutional Layer

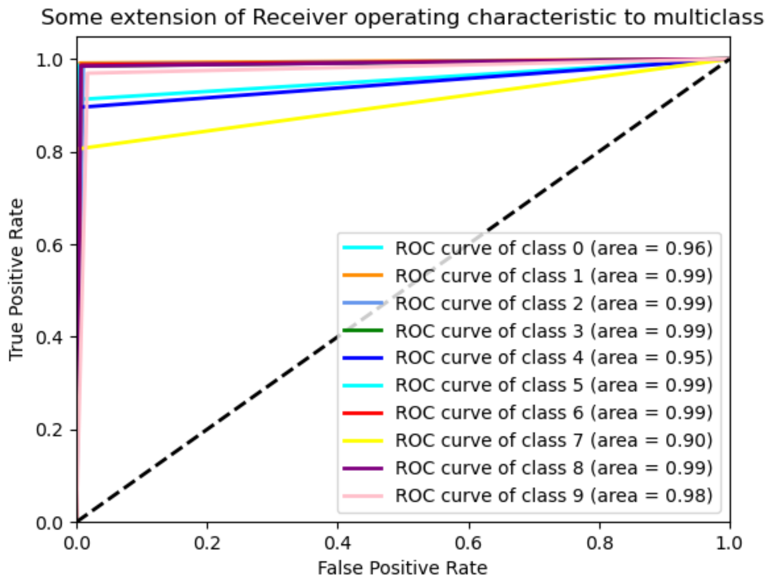


Figure 19: ROC curve

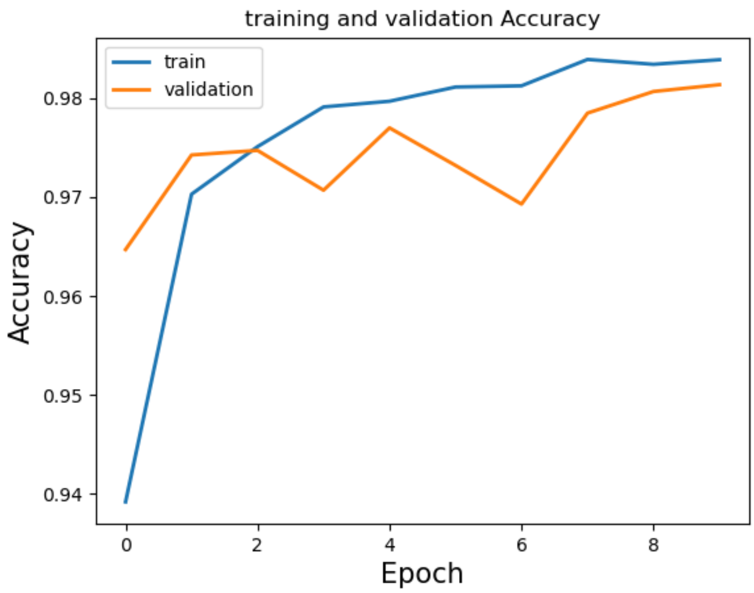


Figure 20: Accuracy curve

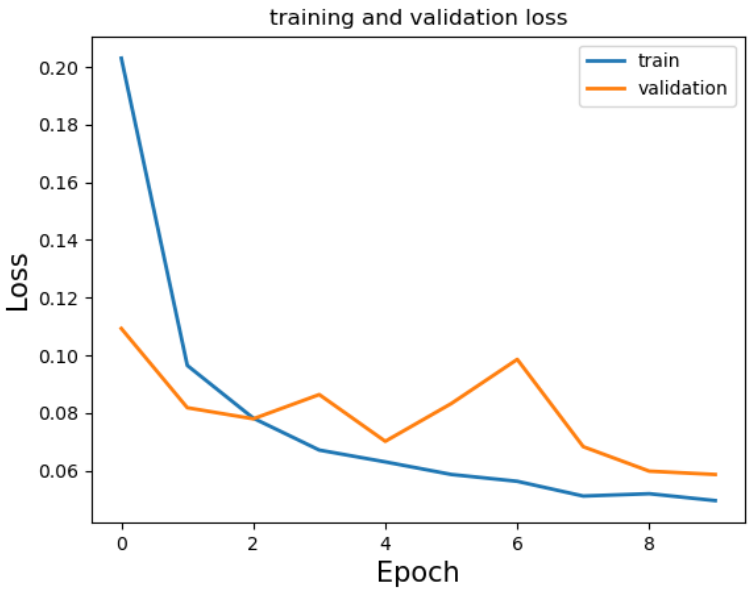


Figure 21: Loss curve

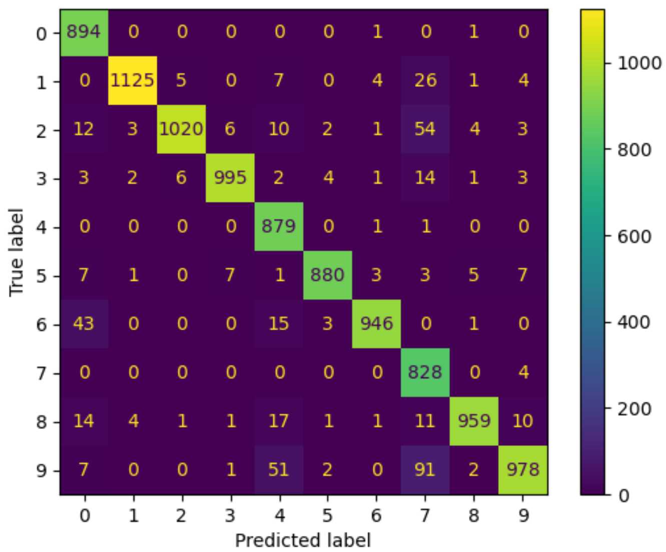


Figure 22: Confusion matrix

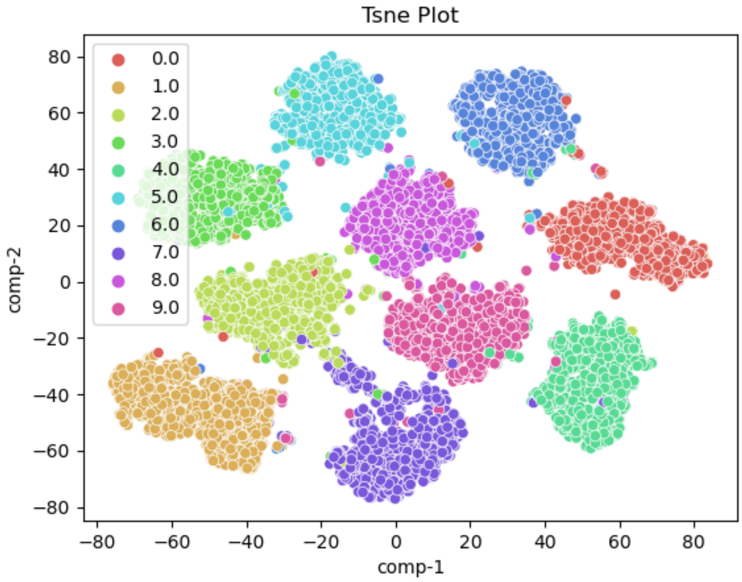


Figure 23: Tsne Plot

7 Show the tsne plot of the last fully connected layer after the first epoch and when training is done. Show the difference between both.

7.1 Task1

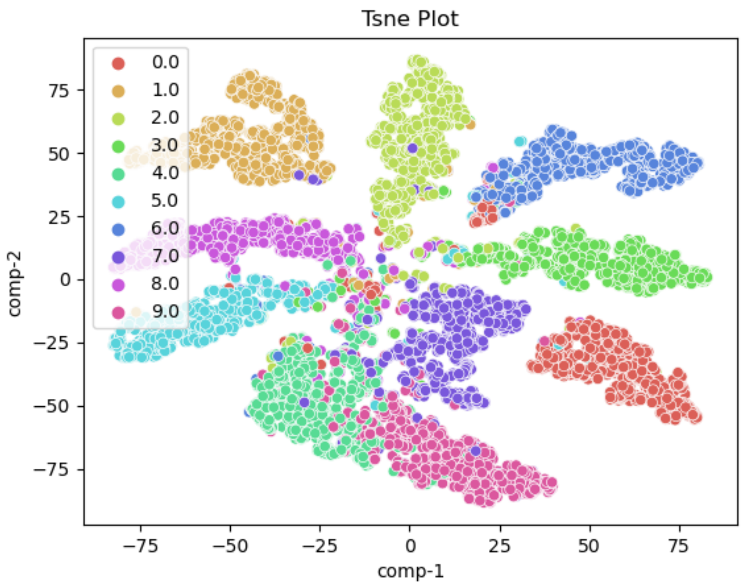


Figure 24: Tsne Plot after 1 epoch

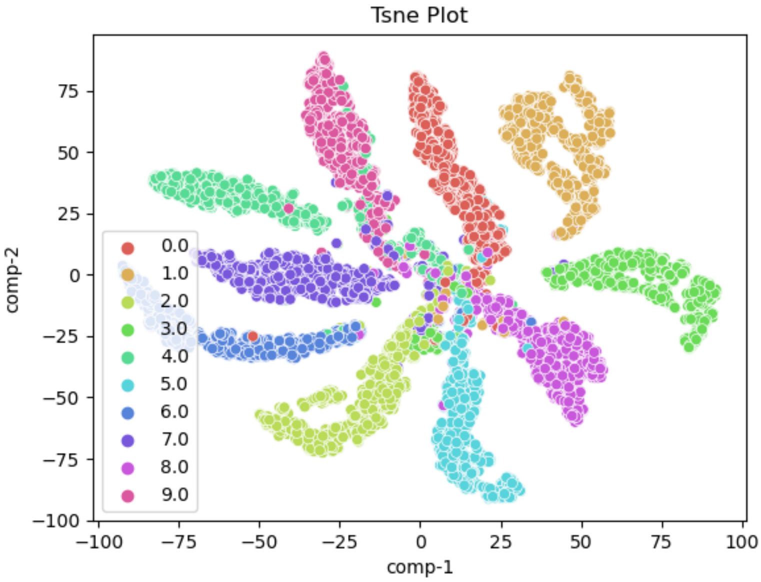


Figure 25: Tsne Plot after training

7.2 Task2

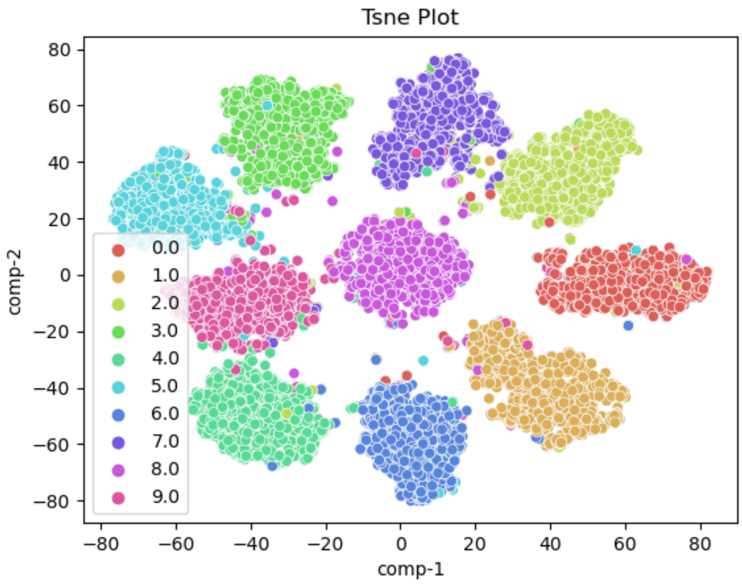


Figure 26: Tsne Plot after 1 epoch

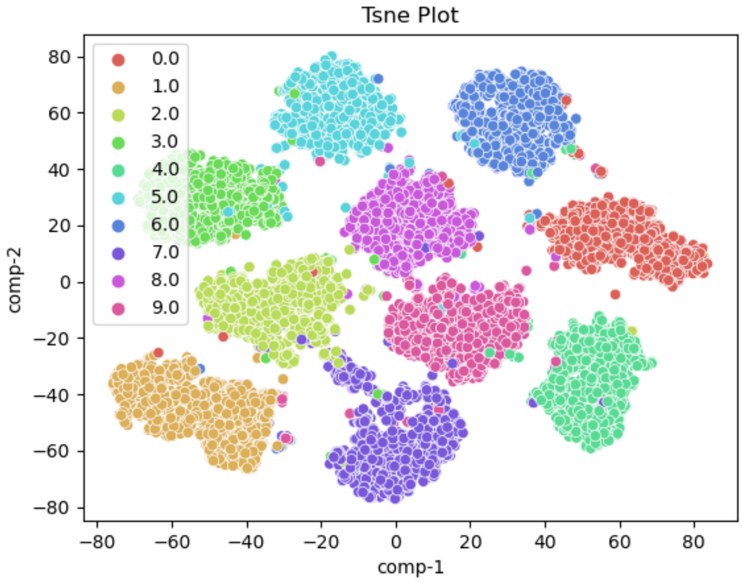


Figure 27: Tsne Plot after training

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

I have learnt following things from this assignment:

- Relu works better than sigmoid and tanh
- CNNs work better than NN even with less parametters
- Greater the batch size, better the training. This mean that with same number of epochs, having larger batch size leads to less test and training loss.
- using batch normalization improves the performance of CNN.
- Using dropout model can generalize better.

All these statements are supported by the experiment results in the later question.

9 Task 1 → Report the accuracy by changing the number of neurons in the hidden layers, or a number of hidden layers or changing loss functions, batch size learning rate, epochs, and the ratio of training and testing data, etc.

9.1 Changing Epochs

```
Test Set Accuracy:  0.9171
Test Set Recall Score:  0.9171
Test Set F1 Score:  0.9183934856675537
```

Figure 28: Scores after 5 epochs

```
Test Set Accuracy:  0.907
Test Set Recall Score:  0.907
Test Set F1 Score:  0.9069946140752756
```

Figure 29: Scores after 10 epochs

9.2 Changing batch Size

```
Test Set Accuracy:  0.8946
Test Set Recall Score:  0.8946
Test Set F1 Score:  0.8947306314069406
```

Figure 30: Scores with batch size = 32

```
Test Set Accuracy:  0.907
Test Set Recall Score:  0.907
Test Set F1 Score:  0.9069946140752756
```

Figure 31: Scores with batch size = 64

9.3 Changing hidden layers

```
Test Set Accuracy:  0.907
Test Set Recall Score:  0.907
Test Set F1 Score:  0.9069946140752756
```

Figure 32: Scores with two hidden layers of size 128 and 64

```
Test Set Accuracy:  0.9071
Test Set Recall Score:  0.9071
Test Set F1 Score:  0.9074249227204706
```

Figure 33: Scores with three hidden layers of size 128, 64 and 32

10 Task 2 → Report the accuracy by adding/subtracting batch norm or dropout layer or changing loss functions, batch size, learning rate, epochs, and the ratio of training and testing data, etc.

10.1 Changing Epochs

```
Test Set Accuracy:  0.9492
Test Set Recall Score:  0.9492
Test Set F1 Score:  0.9492307154907789
```

Figure 34: Scores after 5 epochs

```
Test Set Accuracy:  0.9504
Test Set Recall Score:  0.9504
Test Set F1 Score:  0.950939506568197
```

Figure 35: Scores after 10 epochs

10.2 Changing batch Size

```
Test Set Accuracy:  0.9384
Test Set Recall Score:  0.9384
Test Set F1 Score:  0.9387878788597365
```

Figure 36: Scores with batch size = 32

```
Test Set Accuracy:  0.9492
Test Set Recall Score:  0.9492
Test Set F1 Score:  0.9492307154907789
```

Figure 37: Scores with batch size = 64

10.3 With and without batchnorm layers

```
Test Set Accuracy:  0.9504
Test Set Recall Score:  0.9504
Test Set F1 Score:  0.950939506568197
```

Figure 38: Scores with batch normalization

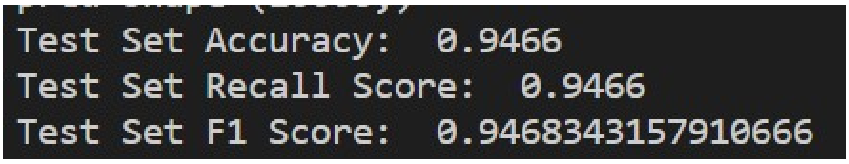


Figure 39: Scores without batch normalization

11 Describe the reasoning for choosing architecture for both tasks

11.1 Task1

Task 1 requires to perform a multiclass classification on MNIST dataset using neural networks. The input of the network was equal to the size of image after vectorizing it i.e 1x28x28= 784. After the input layer two hidden layers were used of size 128 and 64 respectively. These numbers were chosen randomly. As we studied in class that relu performs better than sigmoid, therefore I have used used relu as activation function in hidden layers and softmax at the output layer. Additionaly I have applied dropout = 0.3 on both hidden layers.

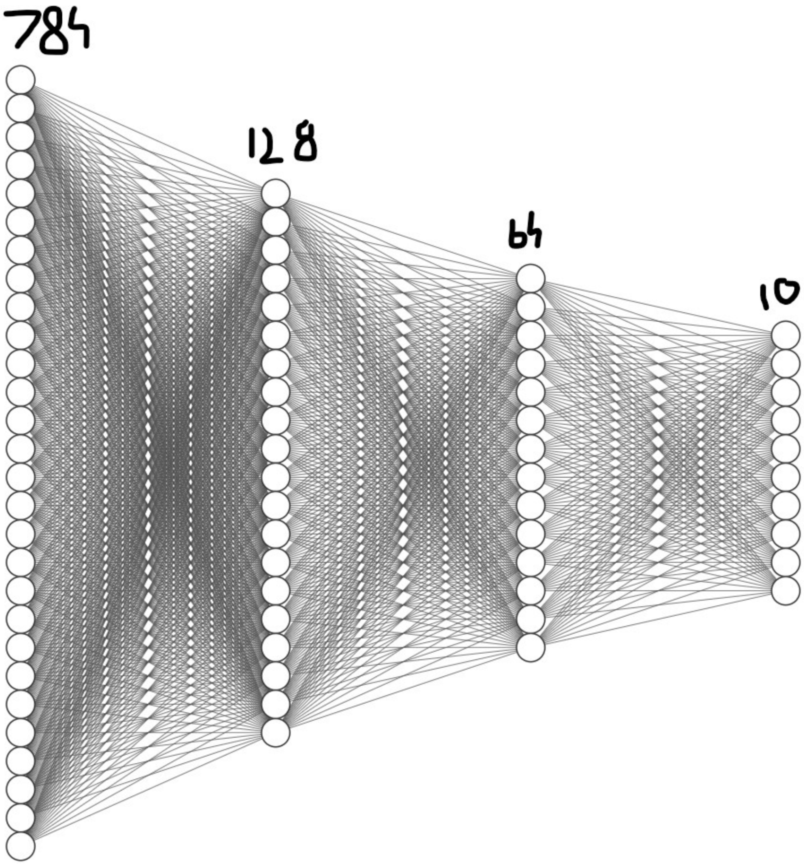


Figure 40: Task 1 Architecture

11.2 Task2

Task 2 requires to perform a multiclass classification on MNIST dataset using convolution neural networks. The input to the model was an image of size 28x28. After input I have a convolutional layer with filter size 3x3 and 4 output channels. This convolutional layer is followed by batch normalization layer and then by relu activation. After that I applied the max pooling on it. After that I repeat the convolutional layer, batch norm layer, relu layer and max pool layer. After that there is a fully connected layer with 10 neurons and softmax is applied on it s output. I didnt included the dropout layer as the network isn't much large and deep.

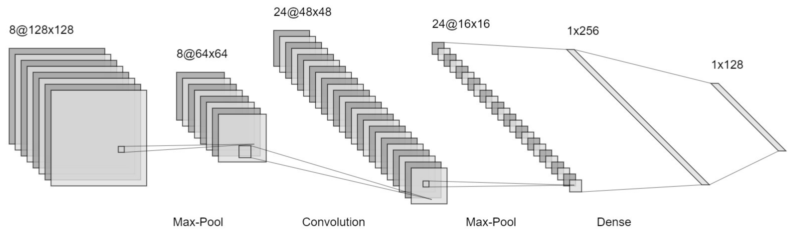


Figure 41: Task 2 Architecture

12 For task 1 Describe the reasoning for choosing the number of hiding layers and nodes.

The hidden layers and numbers were chosen randomly. I tried different combinations and the one with least complex architecture and having same performance was selected.

13 For task 2 describe the reason for choosing the number of filters for each Conv layer and why you choose a number of FC layers

As we have mostly seen in literature that filter size is usually odd. Therefore, the size of my filter is 3x3. I have also used batch normalization and max pooling in my architecture. I tried various combinations and then decided to go with the architecture explained above because of its simplicity and good performance.