CS-671

Deep Learning and its Applications

Assignment 2

Task 1

Foundations of Convolutional Neural Networks

submitted by

Team 5

Vishal Anand B16040

Aman Jain B16044

Yash Agrawal B16120

SCHOOL OF COMPUTING AND ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY MANDI, MANDI

March 2019

1 Objective

The objective of this assignment is to design deep convolutions neural networks for doing MNIST and Line image classification.

2 Part 1

In this part, the model architecture was already given and we had to just implement that model.

2.1 Architecture

Following is the model architecture:

1. 7x7 Convolutional Layer with 32 filters and stride of 1.
2. ReLU Activation Layer.
3. Batch Normalization Layer.
4. 2x2 Max Pooling layer with a stride of 2.
5. Fully connected layer with 1024 output units.
6. ReLU Activation Layer.
7. Output layer with Softmax activation.

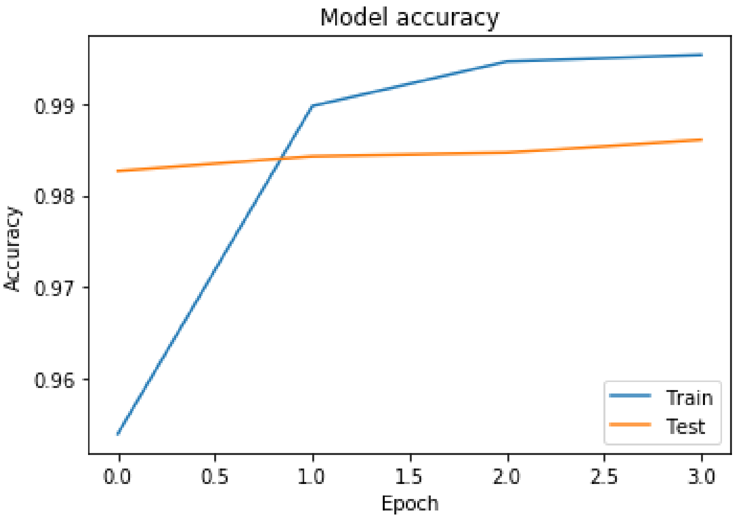
Use the model with adam optimizer and categorical crossentropy loss.

2.2 Results

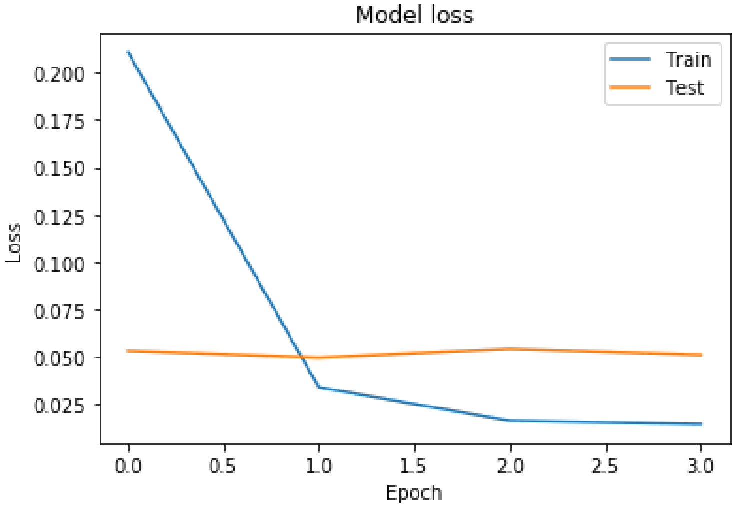
2.2.1 MNIST dataset

Learning Curves

- Accuracy



- Loss



FScores

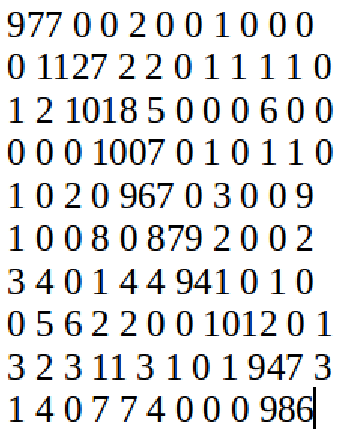
1. Accuracy = 0.9861

2. Precision = 0.9863057376293911

3. Recall = 0.9859680098092074

4. Fscore = 0.9860847180480686

Confusion matrix

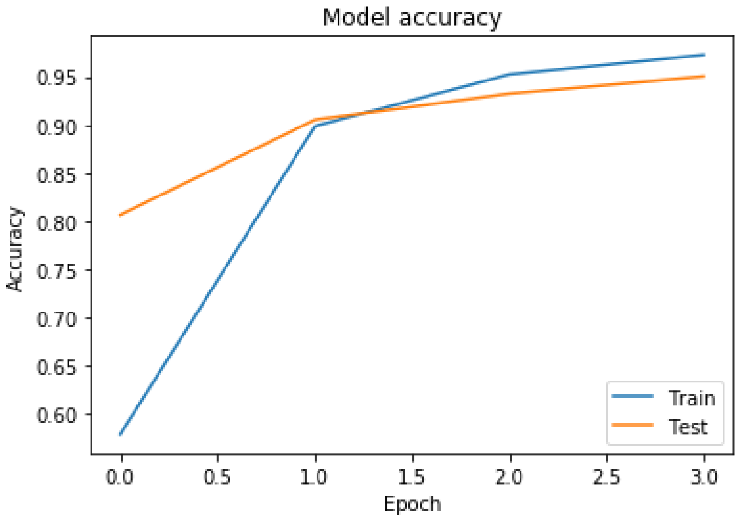


```
977 0 0 2 0 0 1 0 0 0
0 1127 2 2 0 1 1 1 1 0
1 2 1018 5 0 0 0 6 0 0
0 0 0 1007 0 1 0 1 1 0
1 0 2 0 967 0 3 0 0 9
1 0 0 8 0 879 2 0 0 2
3 4 0 1 4 4 941 0 1 0
0 5 6 2 2 0 0 1012 0 1
3 2 3 11 3 1 0 1 947 3
1 4 0 7 7 4 0 0 0 986
```

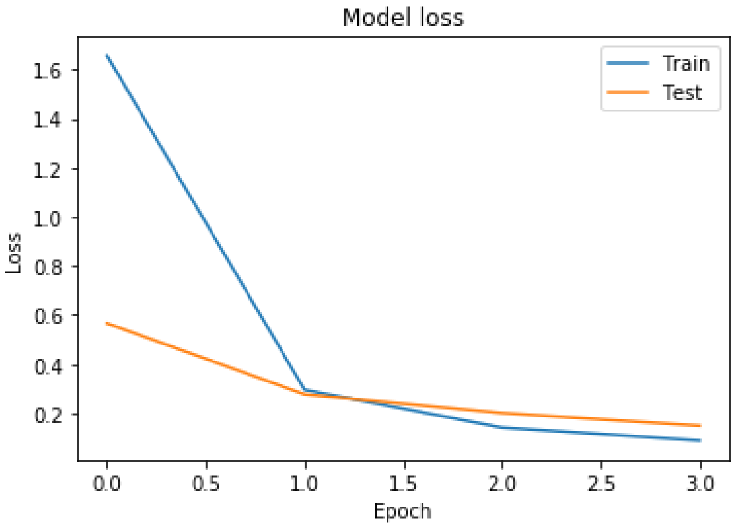
2.2.2 Line dataset

Learning Curves

- Accuracy



- Loss



FScores

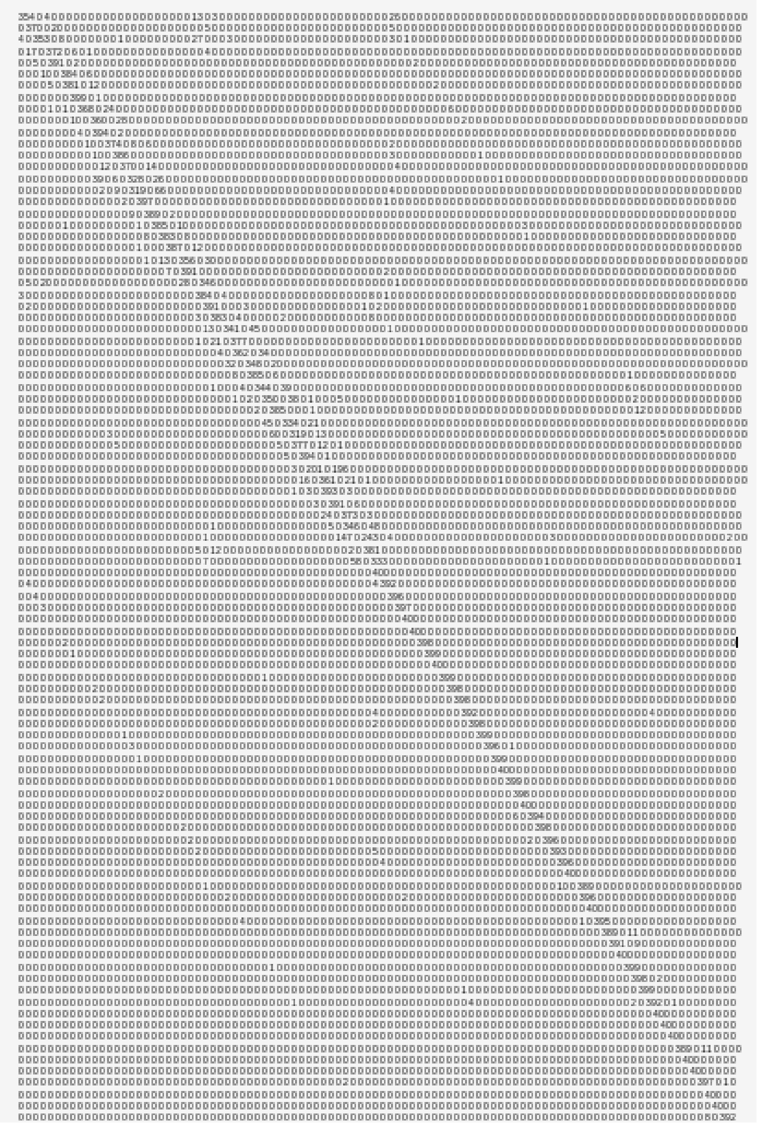
1. Accuracy = 0.9503645833333333

2. Precision = 0.9541497326709223

3. Recall = 0.9503645833333335

4. Fscore = 0.9499837093522262

Confusion matrix



2.3 Inferences

- The accuracy on the MNIST and line dataset seems to be decent though it could have been improved.

- In the given architecture, 7 x 7 filters are used, which is big enough. Smaller filters gives better results.

- Also, there is no regularization or drop outs used, so as to ensure not to overfit the model.

- The fully connected layer has 1024 which can also be tweaked to better the performance.

3 Part 2

In this part, we had to make our own architecture to attain greater accuracy. We are allowed to use any hyperparameters, loss functions, optimizers, etc.

3.1 VARIATION 1

3.1.1 Architecture

Following is the model architecture:

1. 3x3 Convolutional Layer with 32 filters and stride of 1.

2. ReLU Activation Layer.

3. Batch Normalization Layer.

4. 2x2 Max Pooling layer with a stride of 1.

5. Fully connected layer with 2048 output units.

6. ReLU Activation Layer.

7. Dropout layer with a rate of 0.4

8. Batch Normalization Layer.

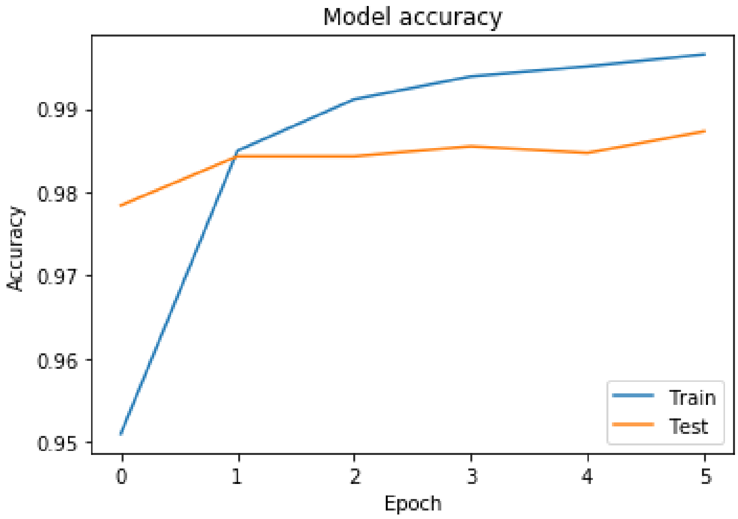
9. Output layer with Softmax activation.

Use the model with adam optimizer and categorical crossentropy loss.

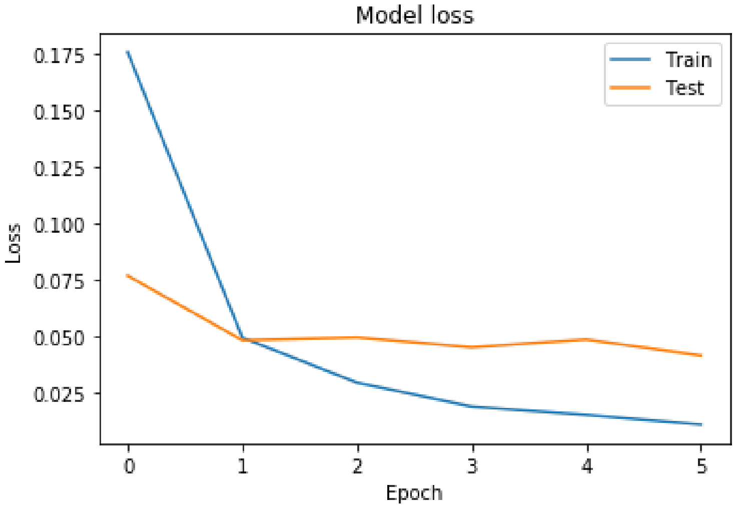
3.1.2 Results on MNIST Dataset

Learning Curves

- Accuracy



- Loss



FScores

1. Accuracy = 0.9873
2. Precision = 0.9872311451509317
3. Recall = 0.9873432144445424
4. Fscore = 0.9872720976326587

Confusion matrix

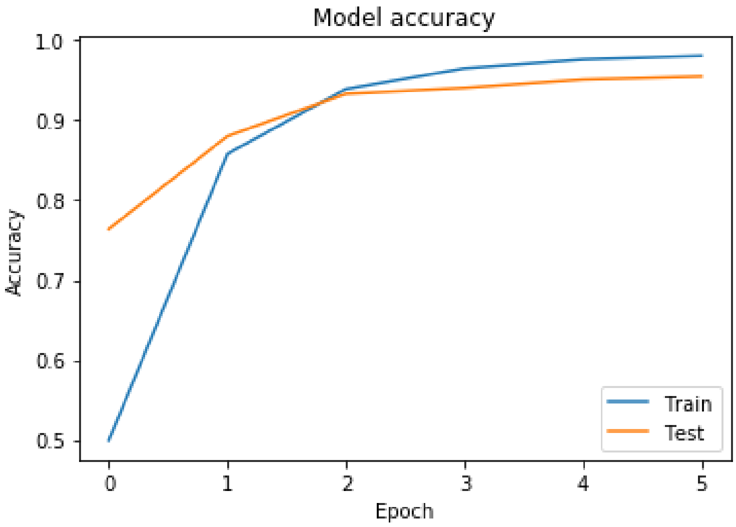
973 0 2 0 0 0 2 2 1 0
2 1120 3 2 0 1 4 1 2 0
1 0 1024 0 0 0 1 3 2 1
0 0 1 998 0 7 0 2 2 0
0 0 1 0 961 0 2 0 0 18
1 0 0 3 0 886 2 0 0 0

4 1 0 0 2 4 945 0 2 0
0 2 8 0 1 0 0 1015 1 1
2 0 3 1 0 0 1 2 959 6
0 1 1 2 4 2 0 5 2 992

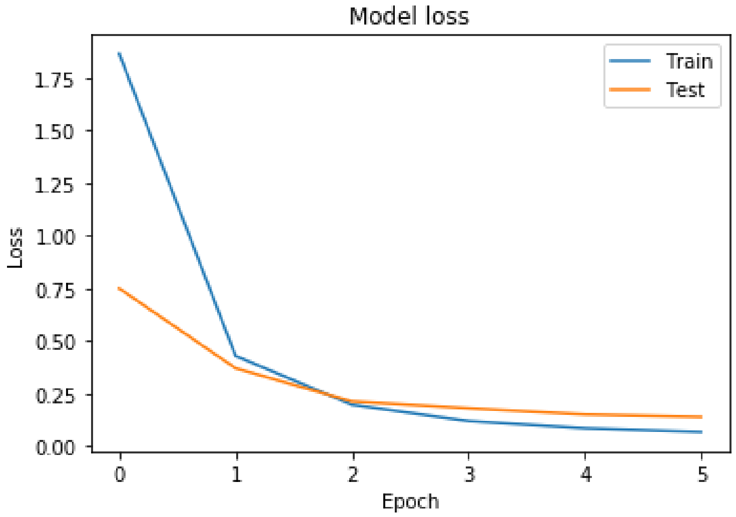
3.1.3 Results on Line Dataset

Learning Curves

- Accuracy



- Loss



FScores

1. Accuracy = 0.95484375
2. Precision = 0.9567343452253799
3. Recall = 0.9548437500000001

4. Fscore = 0.954348224422283

Confusion matrix

3.2 VARIATION 2

3.2.1 Architecture

Following is the model architecture:

1. 3x3 Convolutional Layer with 32 filters and stride of 1.

2. ReLU Activation Layer.

3. Batch Normalization Layer.

4. 2x2 Max Pooling layer with a stride of 1.

5. 1x1 Convolutional Layer with 8 filters and stride of 1.

6. ReLU Activation Layer.

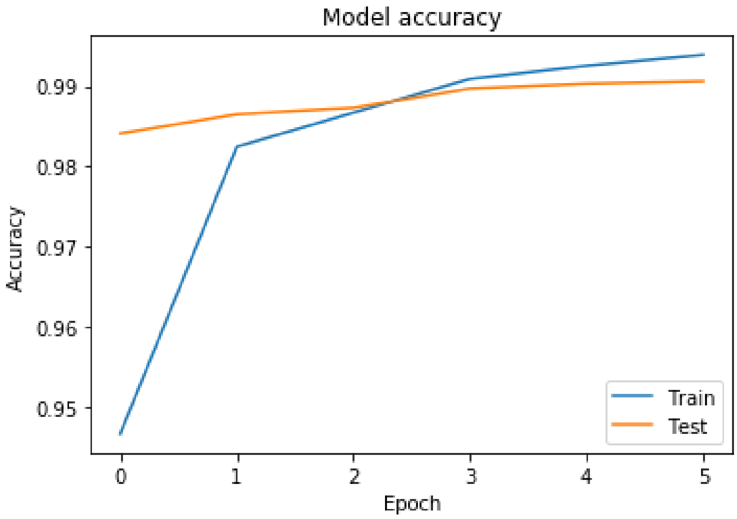
7. Batch Normalization Layer.
8. Dropout layer with rate of 0.4.
9. Fully connected layer with 512 output units.
10. ReLU Activation Layer.
11. Dropout layer with a rate of 0.4
12. Batch Normalization Layer.
13. Output layer with Softmax activation.

Use the model with adam optimizer and categorical crossentropy loss.

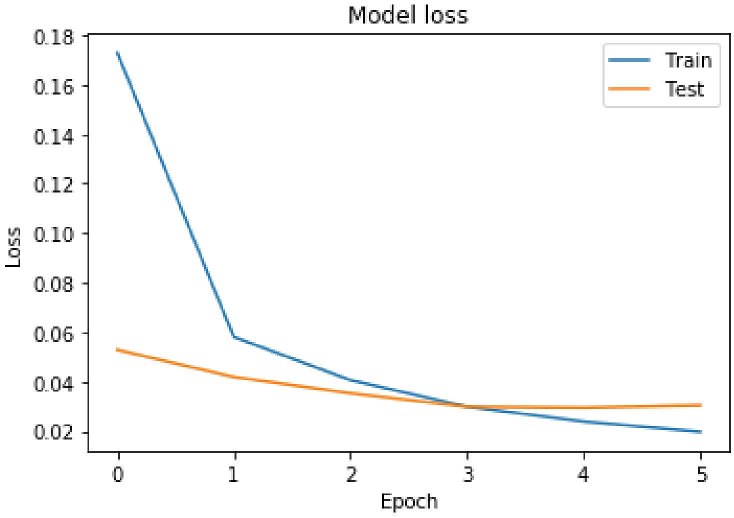
3.2.2 Results on MNIST Dataset

Learning Curves

- Accuracy



- Loss



FScores

1. Accuracy = 0.9906
2. Precision = 0.9905937167671777
3. Recall = 0.9905041041965077
4. Fscore = 0.9905347745005513

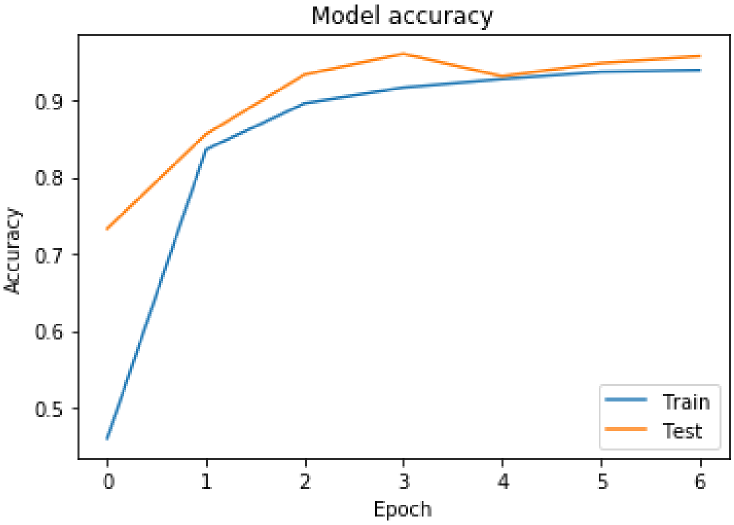
Confusion matrix

```
975 0 0 0 0 1 3 1 0 0
0 1134 0 0 0 0 1 0 0 0
1 0 1023 0 1 0 1 5 1 0
0 0 1 1003 0 3 0 2 0 1
0 0 0 0 974 0 4 0 0 4
1 0 0 2 0 886 2 1 0 0
3 2 0 0 1 1 950 0 1 0
0 1 8 1 0 0 0 1017 1 0
2 1 3 4 1 2 2 3 953 3
0 4 0 2 5 3 0 4 0 991
```

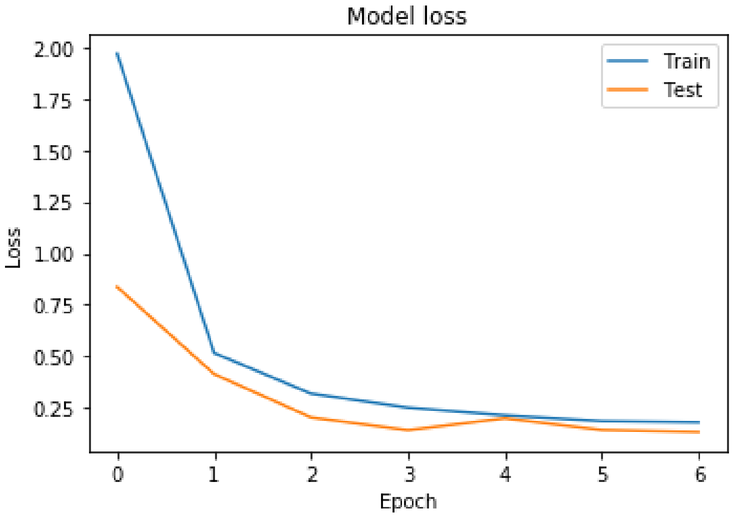
3.2.3 Results on Line Dataset

Learning Curves

- Accuracy



- Loss



Confusion matrix

[illegible]

FScores

1. Accuracy = 0.957109375

2. Precision = 0.961116913227459

3. Recall = 0.957109375

4. Fscore = 0.9565920925161743

3.3 VARIATION 3

3.3.1 Architecture

Following is the model architecture:

1. 3x3 Convolutional Layer with 32 filters and stride of 1.

2. ReLU Activation Layer.

3. Batch Normalization Layer.

4. 2x2 Max Pooling layer with a stride of 1.

5. 1x1 Convolutional Layer with 16 filters and stride of 1.

6. ReLU Activation Layer.

7. Batch Normalization Layer.

8. Dropout layer with rate of 0.3.

9. Fully connected layer with 512 output units.

10. ReLU Activation Layer.

11. Dropout layer with a rate of 0.3.

12. Batch Normalization Layer.

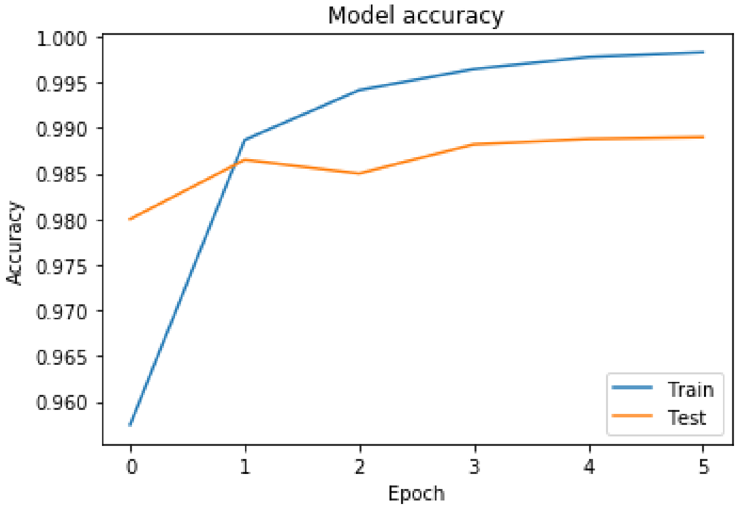
13. Output layer with Softmax activation.

Use the model with adam optimizer and categorical crossentropy loss.

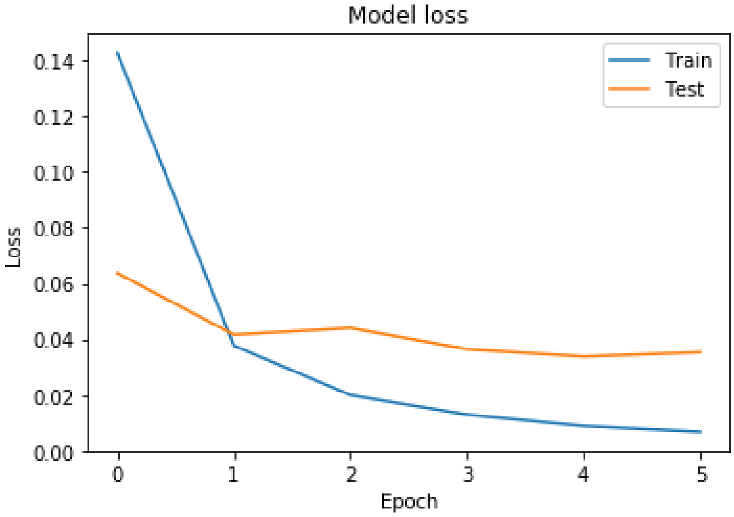
3.3.2 Results on MNIST Dataset

Learning Curves

- Accuracy



- Loss



FScores

1. Accuracy = 0.989
2. Precision = 0.9890279470195755
3. Recall = 0.9889174777911804
4. Fscore = 0.9889579945684706

Confusion matrix

975 0 3 0 0 0 0 1 1 0

1 1130 1 1 0 1 0 1 0 0

1 0 1022 0 1 0 2 6 0 0

0 0 1 1003 0 2 0 2 2 0

0 1 0 0 972 0 5 1 0 3

2 1 0 5 0 881 1 0 1 1

3 3 0 0 1 1 948 0 2 0

1 1 9 0 0 0 0 1016 0 1

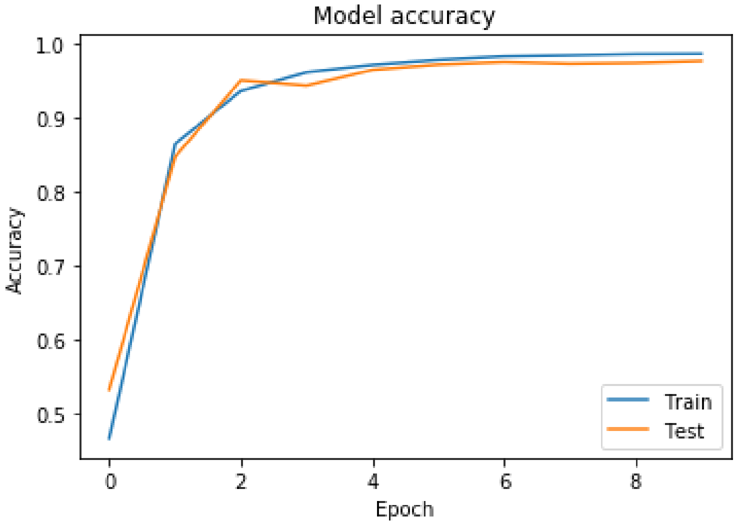
4 0 3 0 0 0 1 1 962 3

1 2 1 2 7 4 0 7 4 981

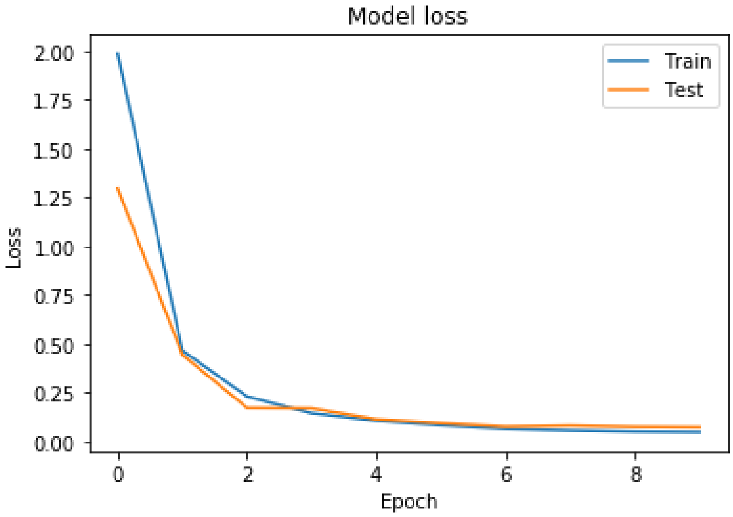
3.3.3 Results on Line Dataset

Learning Curves

- Accuracy



- Loss



FScores

1. Accuracy = 0.9769791666666666
2. Precision = 0.9777776372497188
3. Recall = 0.9769791666666667

4. Fscore = 0.976950191735133

Confusion matrix

3.4 Inferences

- The best results were obtained on MNIST Dataset in variation 2.
- The best results were obtained on Line Dataset in variation 3.
- Increasing one convolution layer with a dropout of 0.4 in architecture improved results for MNIST Dataset. Two small filters instead of one large filter proved to be a good variation and gave good results.
- Also, there is was no regularization or drop outs used in PART 1,therefore inserting dropout ensures the model does not over fit.
- Changing the fully connected layer with 1024 neurons to two fully connected layers with less number of output units increases the accuracy.