I certify that the code and data in this assignment were generated independently,using only the tools and resources defined in the course and that I did not receive any external help, coaching or contributions during the production of this work.

1. Income Dataset Analysis:

The income dataset gives the description about the income of the people based on different factors such as occupation, education,workclass, relationship,native country e.t.c

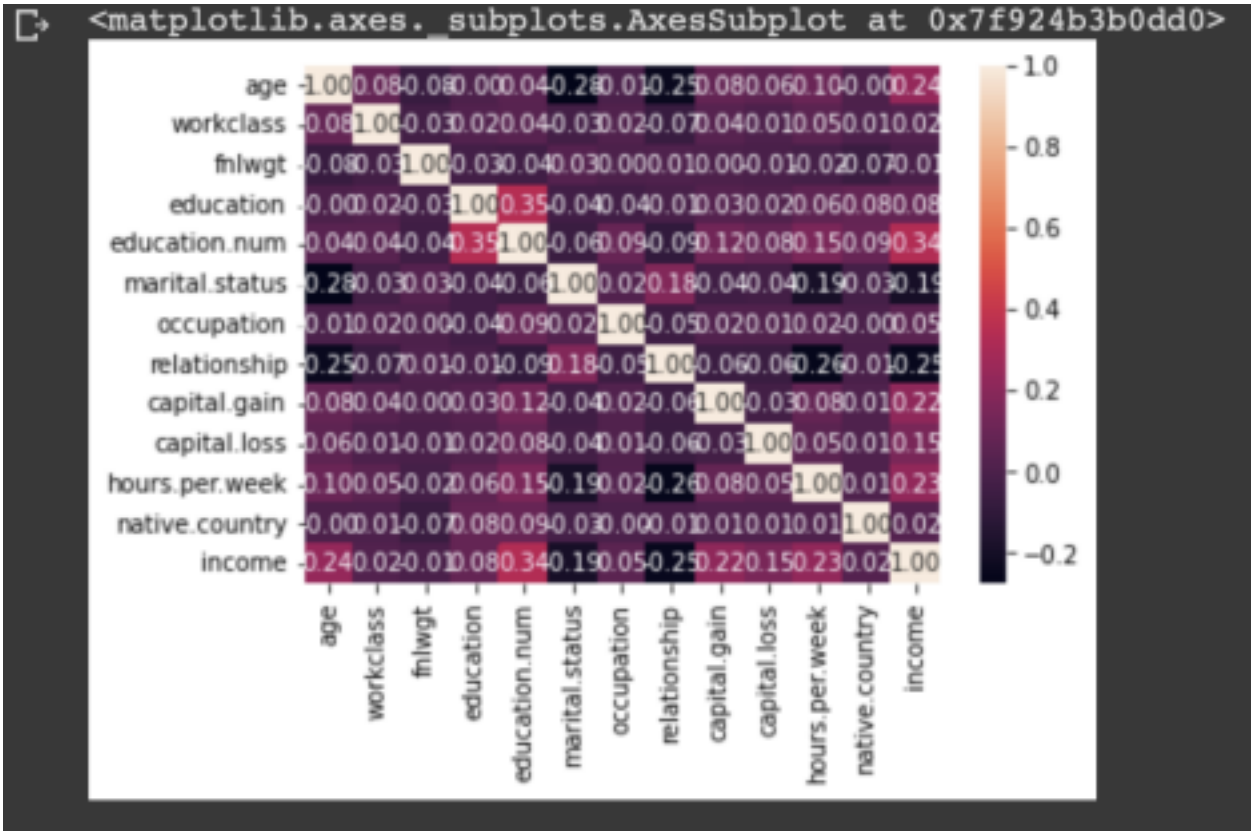
- Shape of Dataset : (32561,13)
- Total rows: 32561
- Total columns : 13
- Null Values : 0
- Categorical Columns: [Occupation,Workclass,Marital status, Native Country]
- Most of the columns are int data type

Statistical Data Analysis :

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	r
count	30162.000000	30162.000000	3.016200e+04	30162.000000	30162.000000	30162.000000	30162.000000	
mean	38.437902	2.199324	1.897956e+05	10.333764	10.121312	2.580134	5.959850	
std	13.134665	0.953925	1.056525e+05	3.812292	2.549995	1.498016	4.029566	
min	17.000000	0.000000	1.378900e+04	0.000000	1.000000	0.000000	0.000000	
25%	28.000000	2.000000	1.176272e+05	9.000000	9.000000	2.000000	2.000000	
50%	37.000000	2.000000	1.784300e+05	11.000000	10.000000	2.000000	6.000000	
75%	47.000000	2.000000	2.376285e+05	12.000000	13.000000	4.000000	9.000000	
max	90.000000	6.000000	1.484705e+06	15.000000	16.000000	6.000000	13.000000	

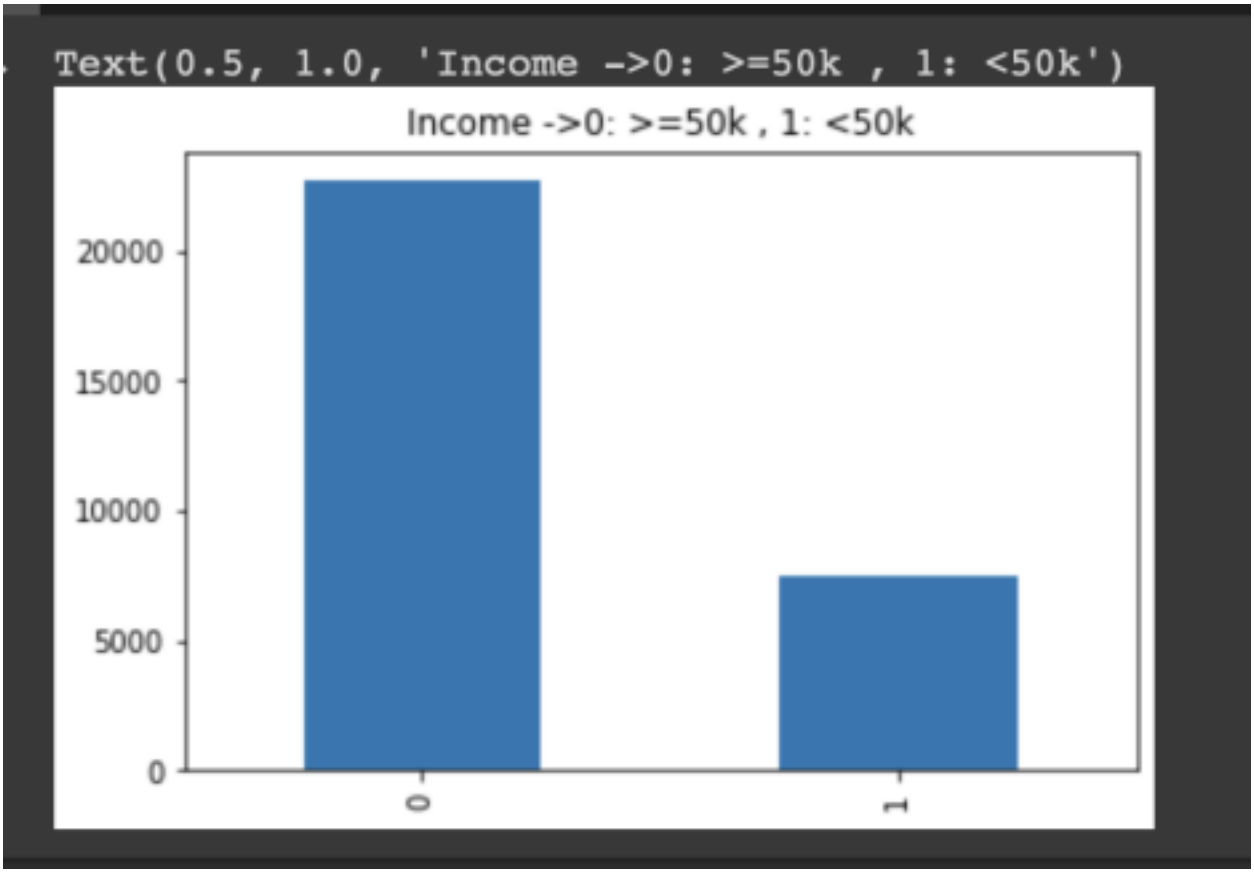
Visualization Graph :

1. Correlation Graph



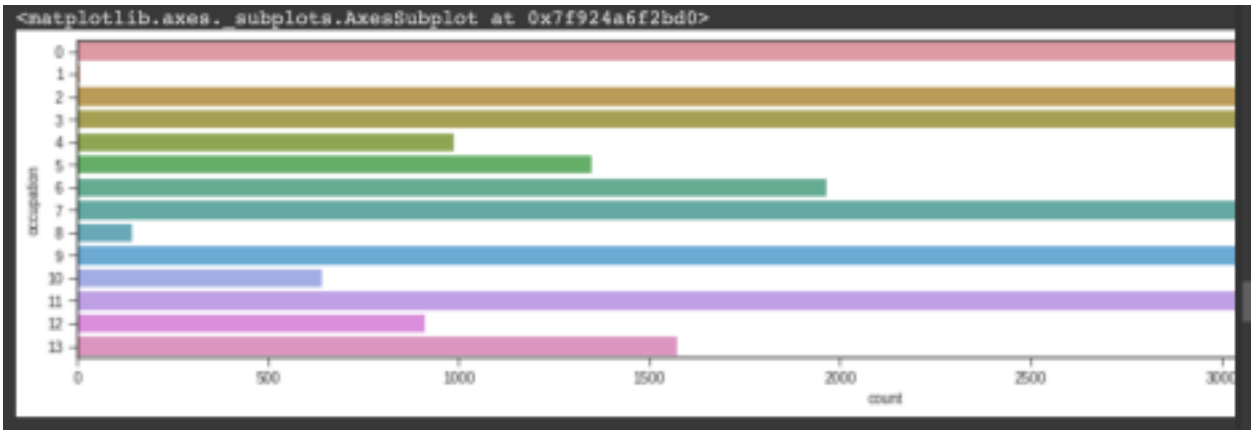
This correlation graph tells about the relationship between two features. The closer the values the higher the correlation. In this occupation has a strong correlation with age and age has a strong correlation with marital status.

2. Bar Plot



This bar plot tells us about the count of people having income less than or greater than 50K according to given dataset.

3. Occupation Bar Plot



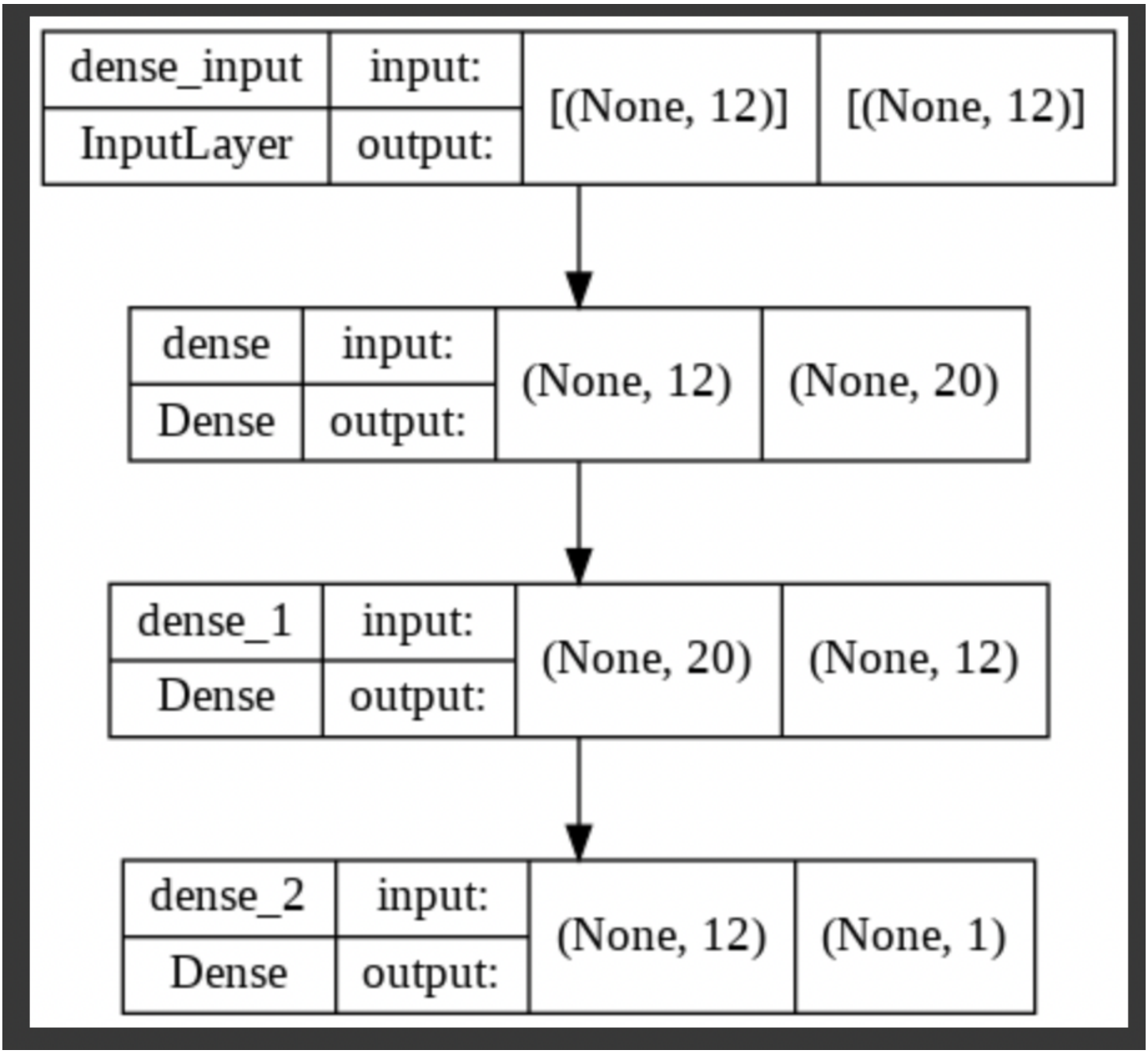
This bar plot tells us about the count of different occupation in the dataset.

3. For the Preprocessing I have used sklearn.preprocessing standard scaler fit transform function to normalize the dataset.

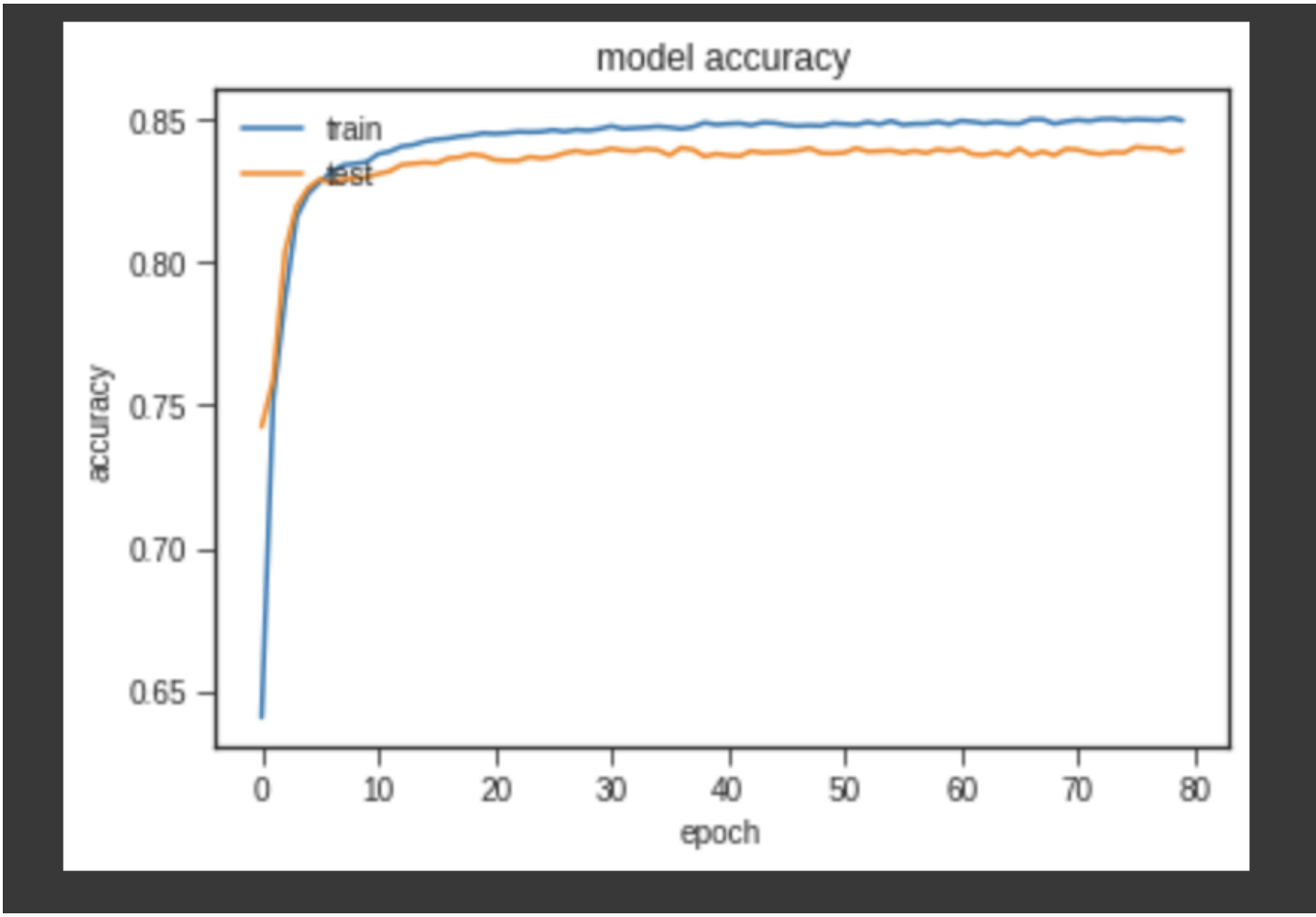
4. Architecture of NN

- For training I have used 80% of the data and for the test remaining 20% ●

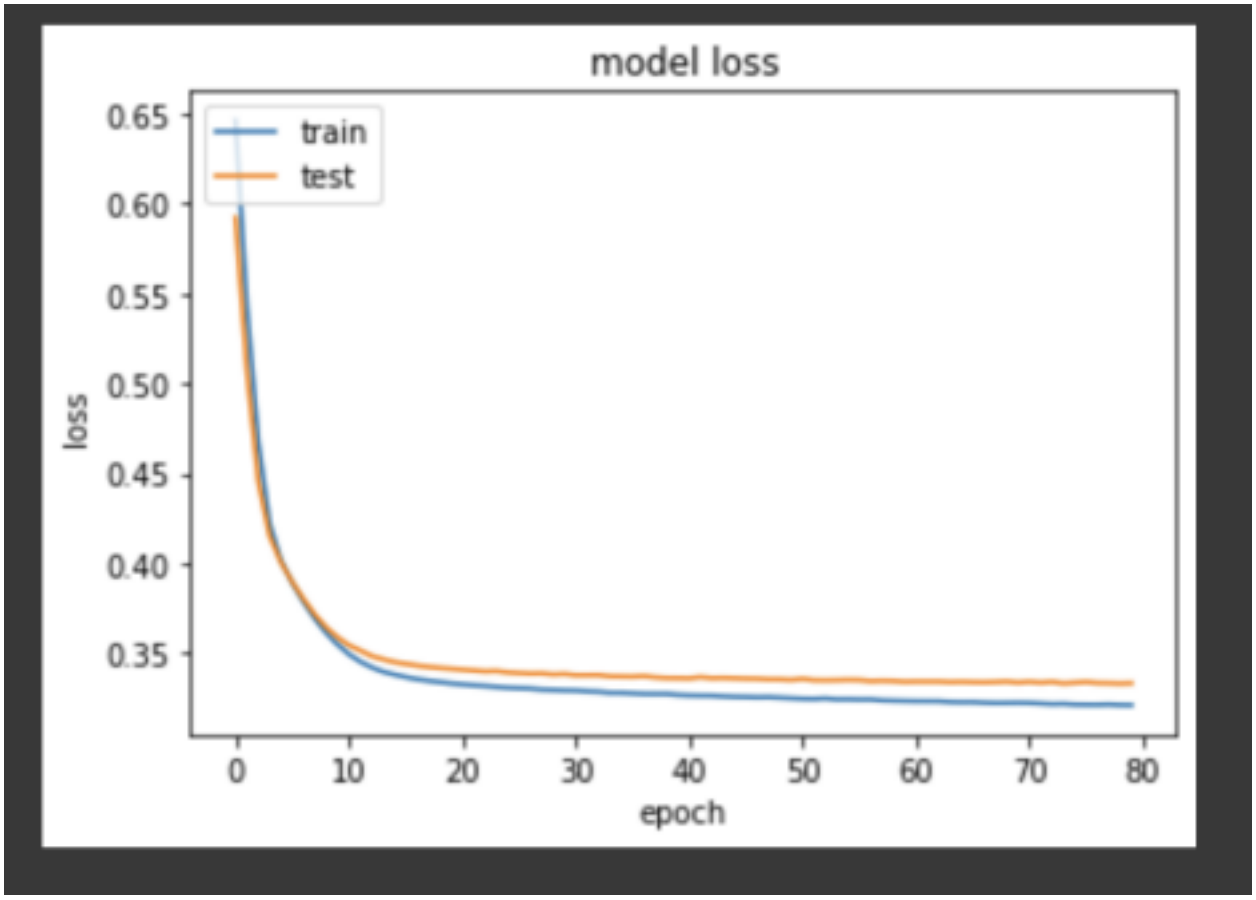
Input Dimension for the first hidden layer is 12 as we have 12 features. ●
First hidden layer output is 20 and activation function is Relu
- For second hidden layer output is 12 and activation function is Relu
- For the output layer the output is obviously 1 and activation function is Sigmoid.
- Optimizer = Adam
- Loss function = Binary Crossentropy
- Accuracy = 85%
- Loss = 0.320
- Batch size = 500
- Epochs = 80



5. Training and Test Accuracy:



2. Test and Training loss:


model loss
train
test
loss
epoch
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0
10
20
30
40
50
60
70
80

PART -2

1. Changing Optimizers

Optimizer	Accuracy
SGD	81.29
Adadelta	48.36
RMSprop	84.02

2. Changing Dropout

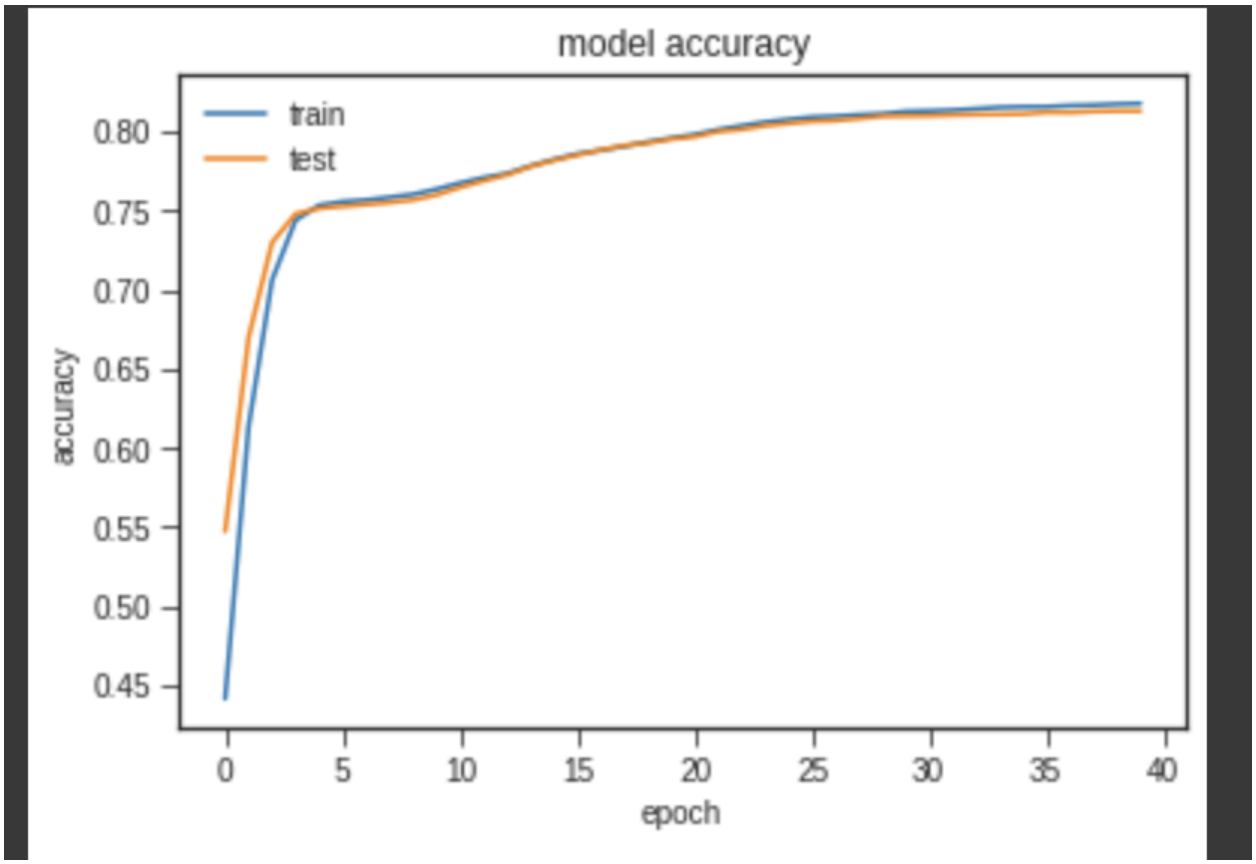
Dropout_Value	Accuracy
0.3	82.93
0.5	83.44
0.7	82.24

3. Changing Activation Function

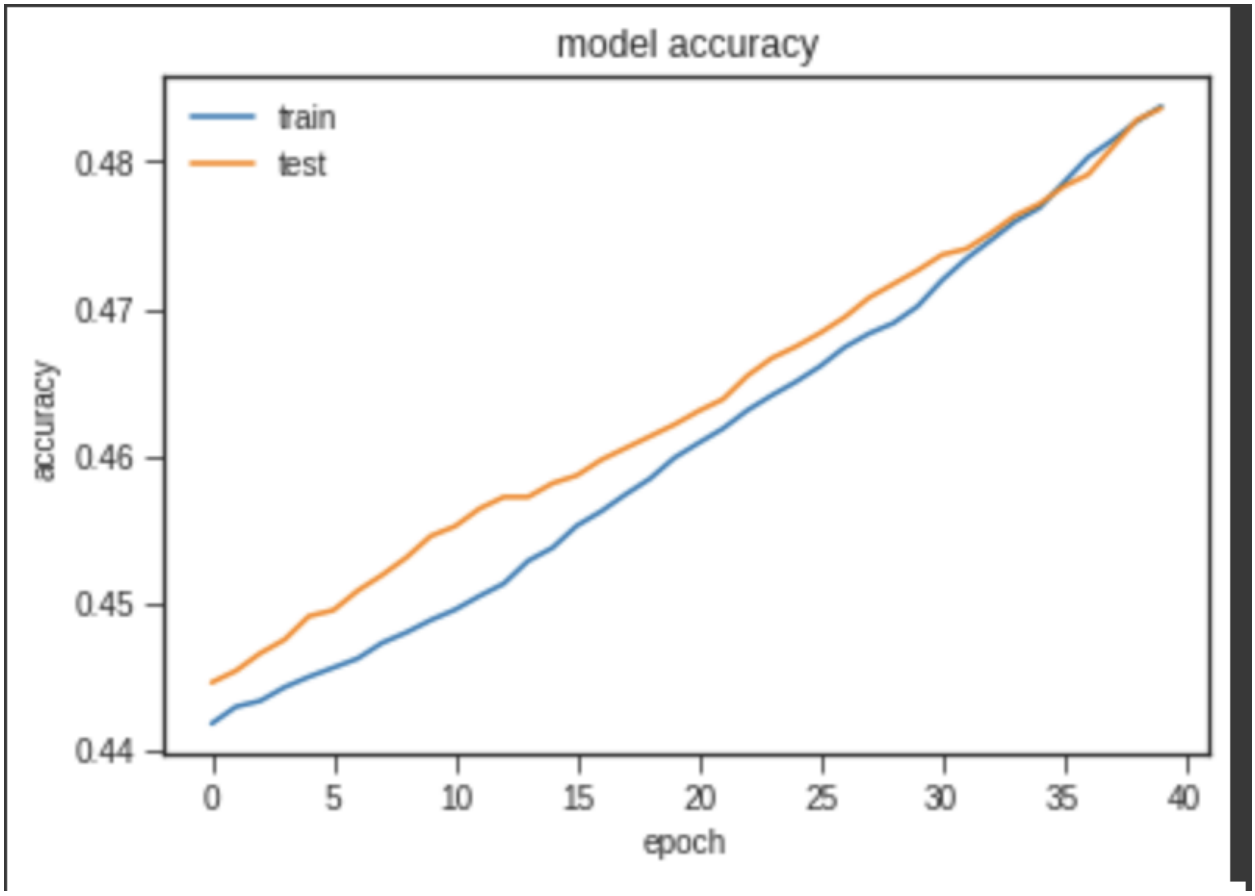
Activation Function	Accuracy
Selu	83.68
Elu	83.81
Tanh	83.62

Q2) Training and Test accuracy graphs

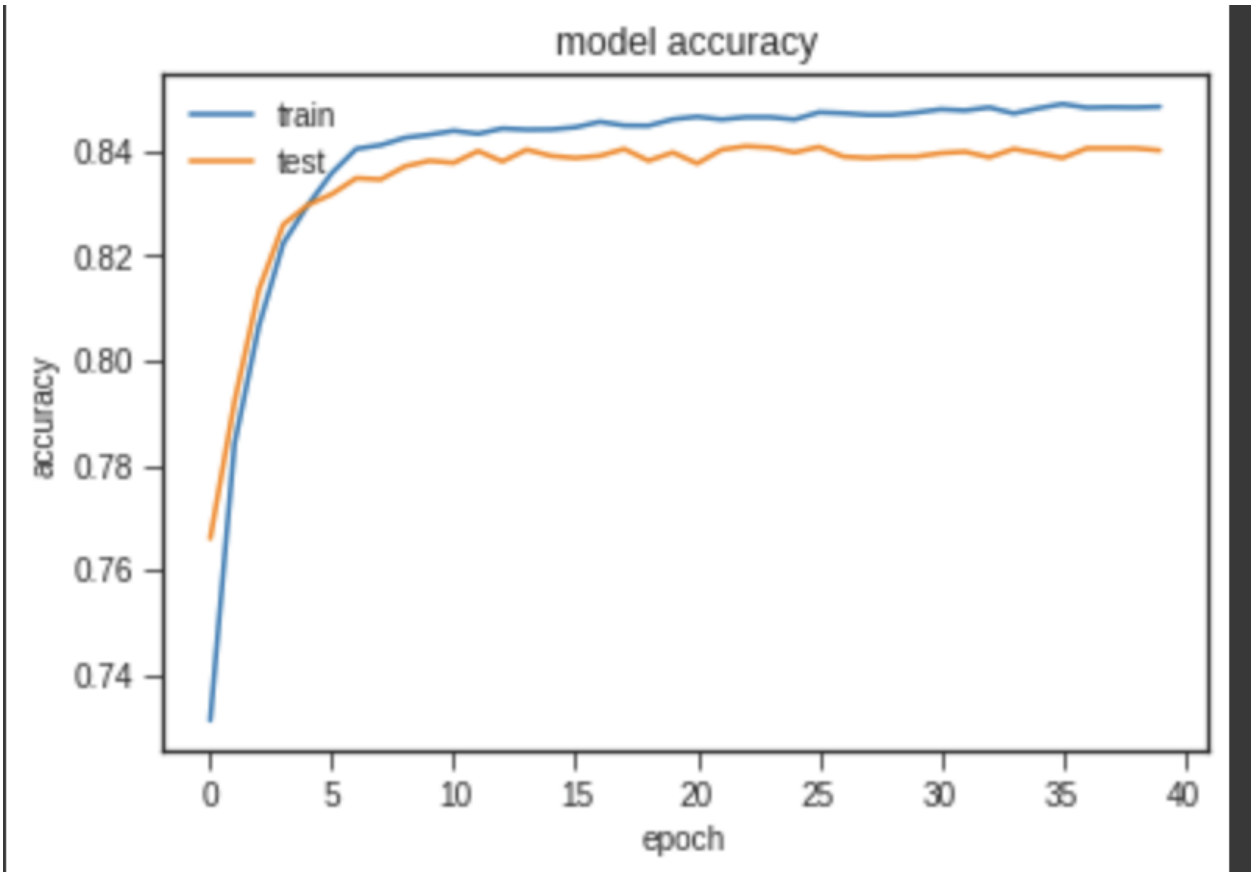
1. Optimizer = SGD



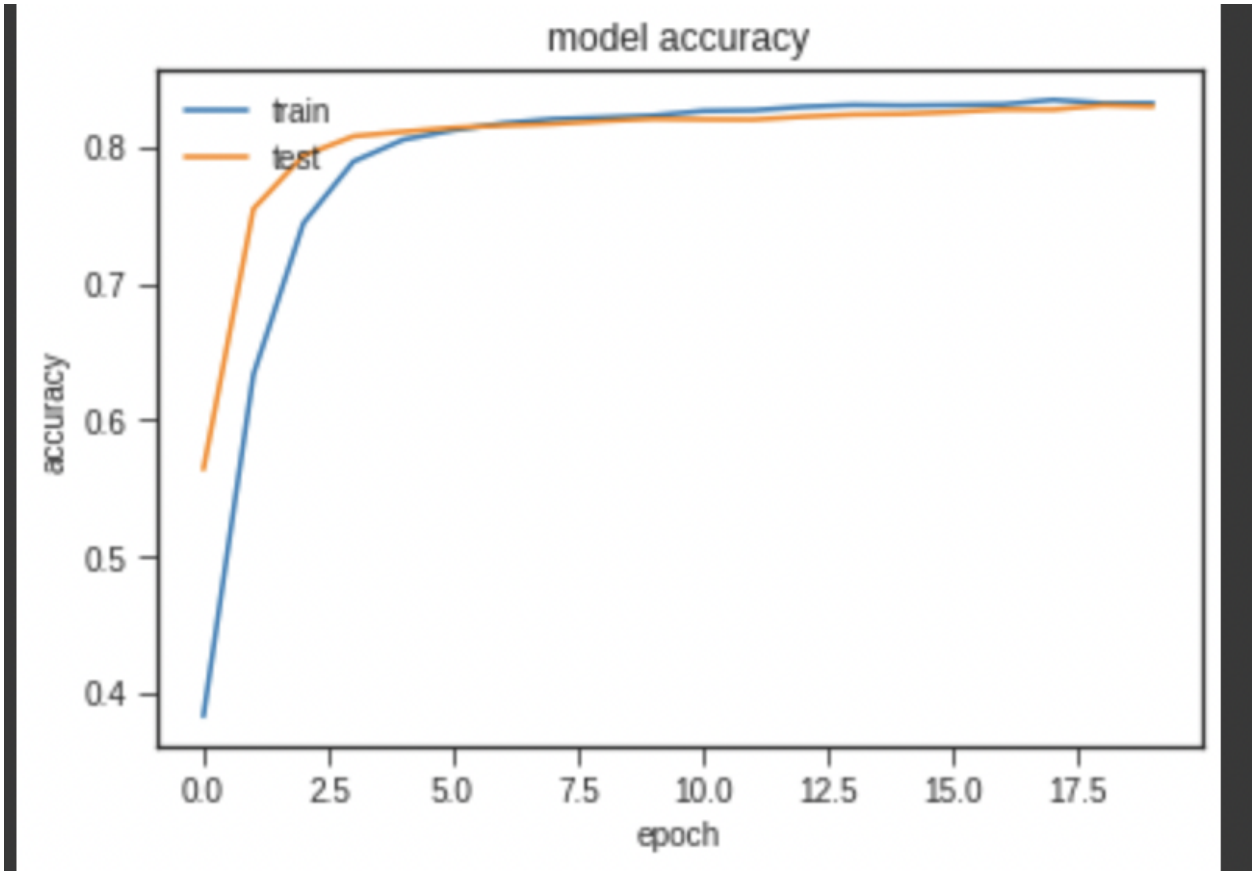
2. Optimizer = Adadelta


model accuracy
train
test
0.48
0.47
0.46
0.45
0.44
accuracy
0
5
10
15
20
25
30
35
40
epoch

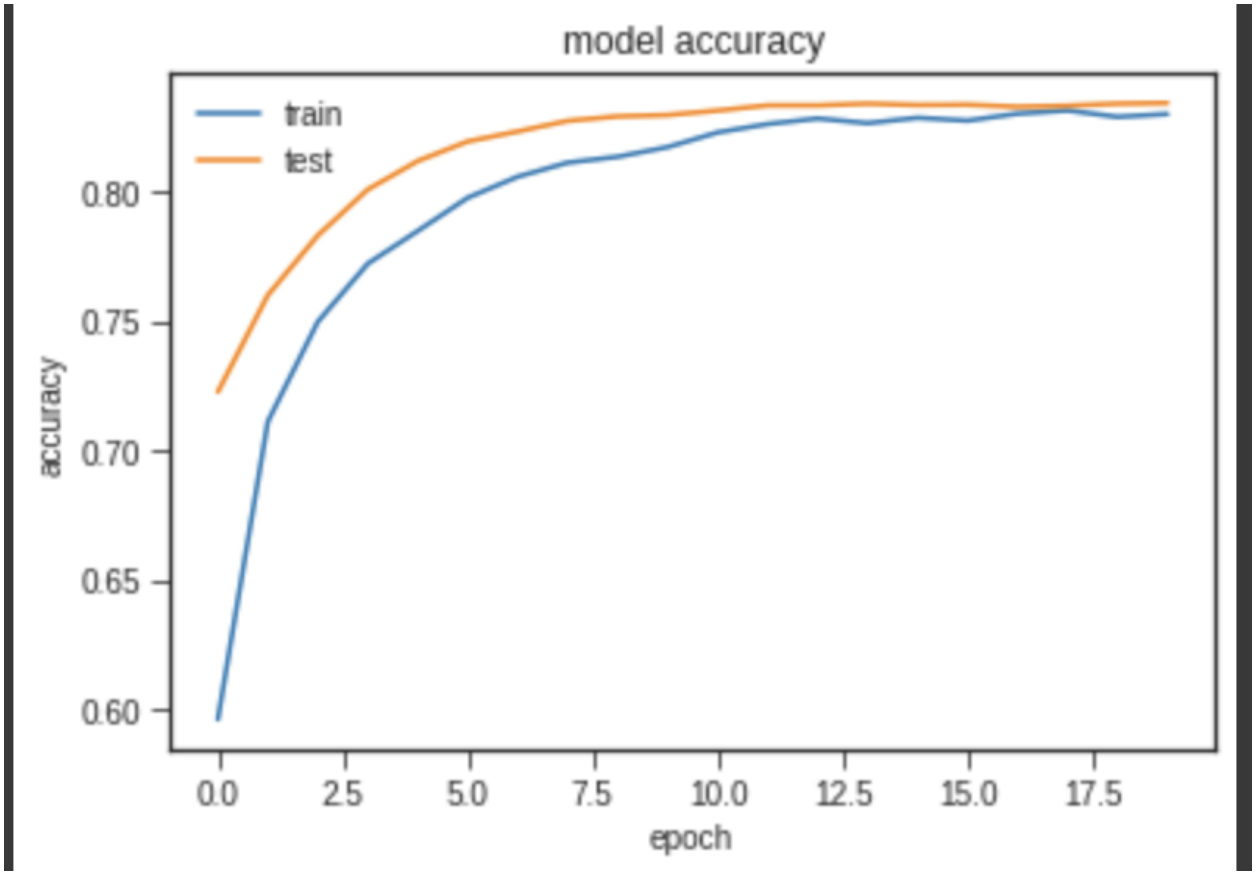
3. Optimizer = RMSprop



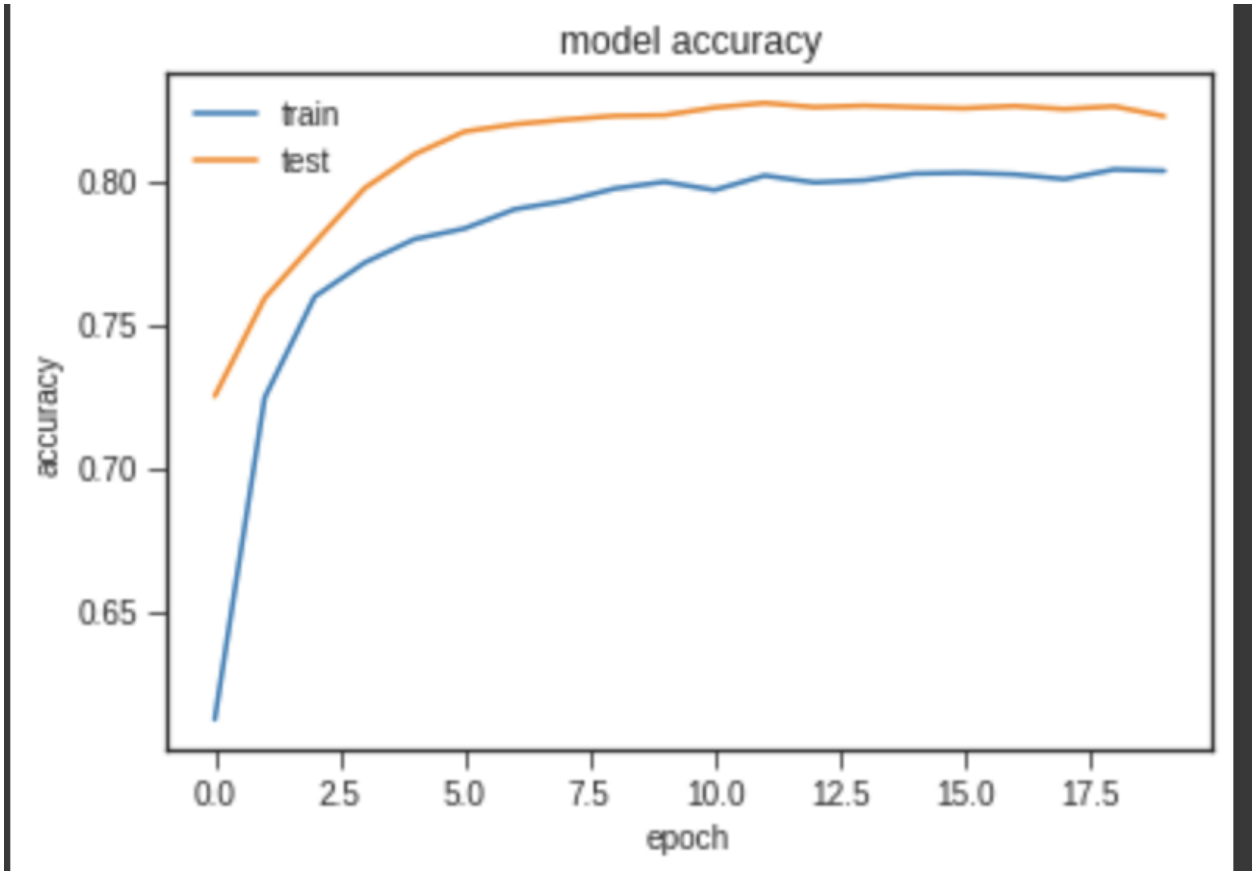
4. Dropout = 0.3


model accuracy
train
test
accuracy
0.8
0.7
0.6
0.5
0.4
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
epoch

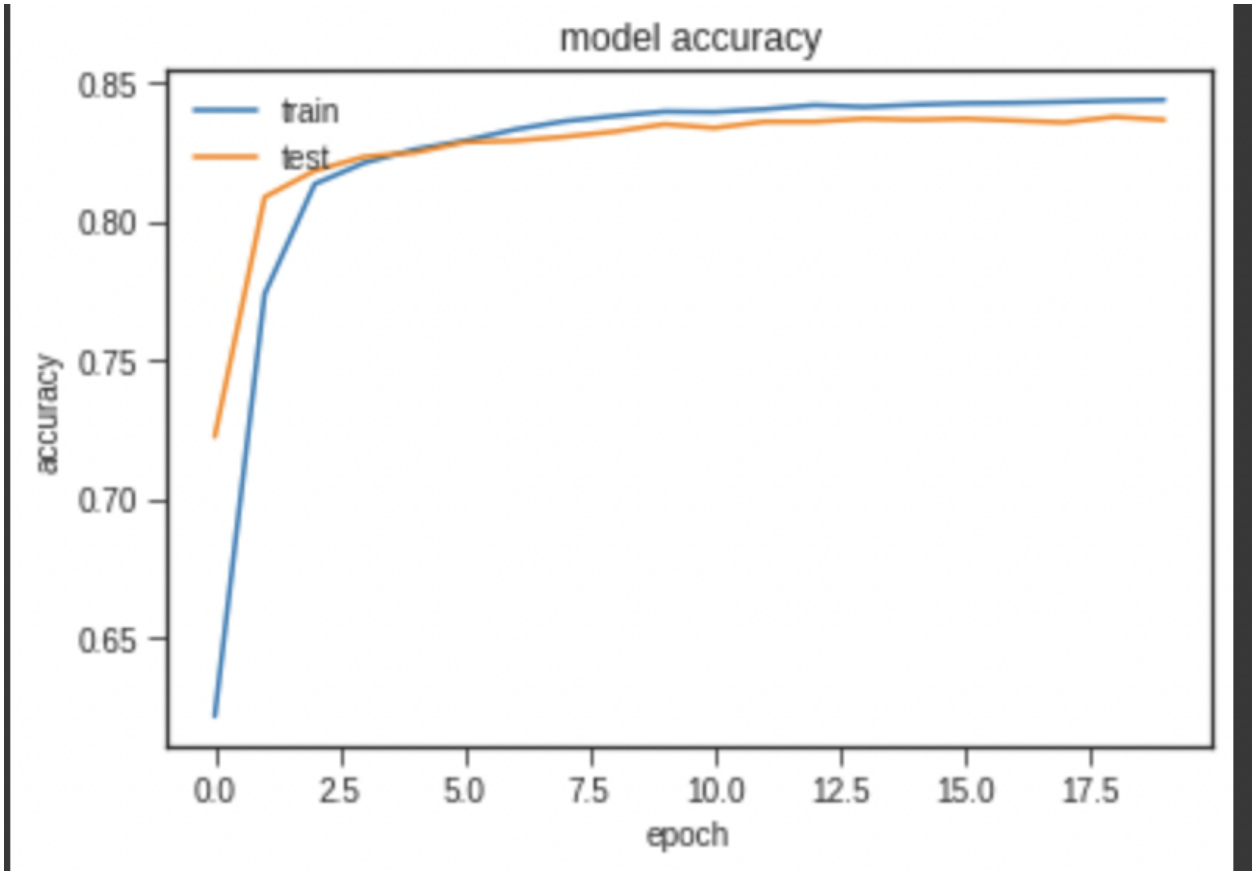
5. Dropout = 0.5


model accuracy
train
test
0.80
0.75
0.70
0.65
0.60
accuracy
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
epoch

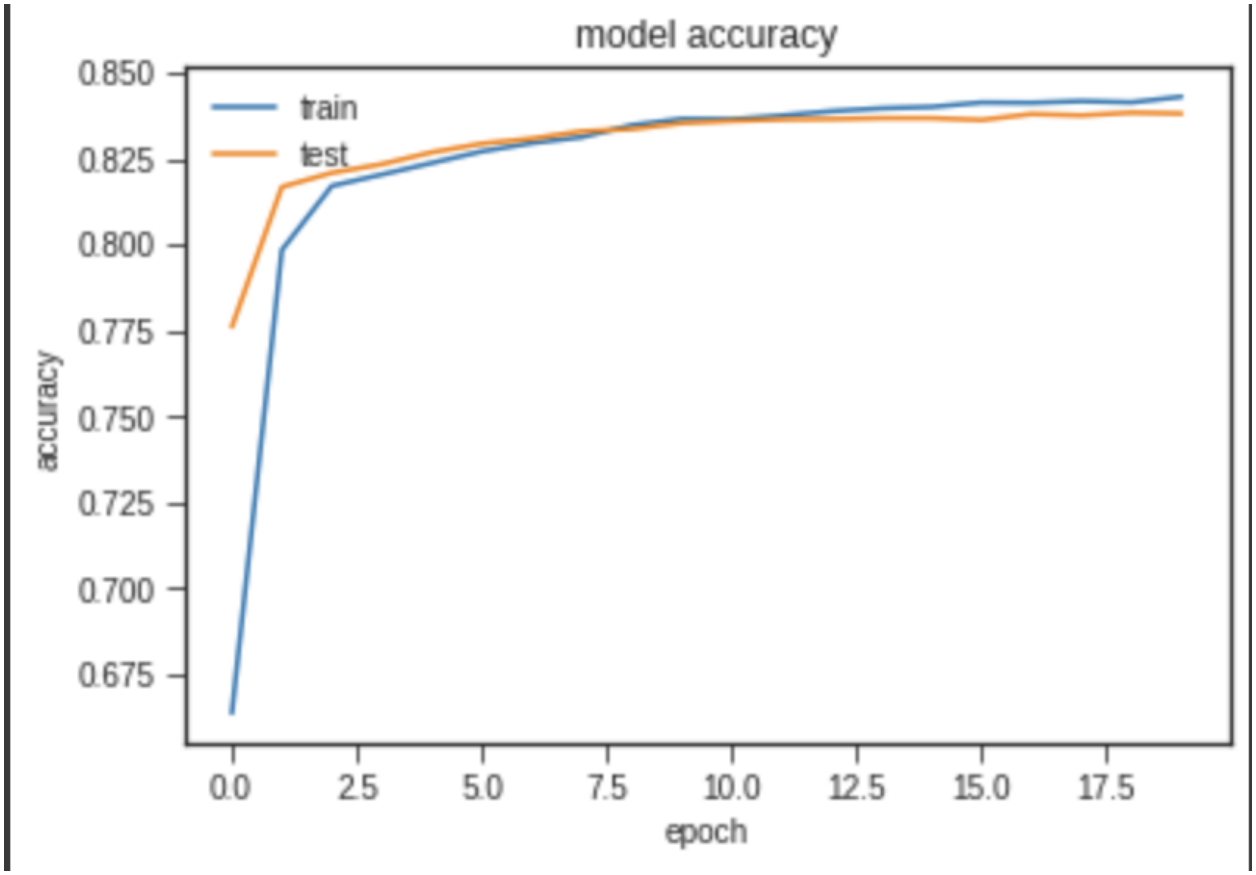
6. Dropout = 0.7



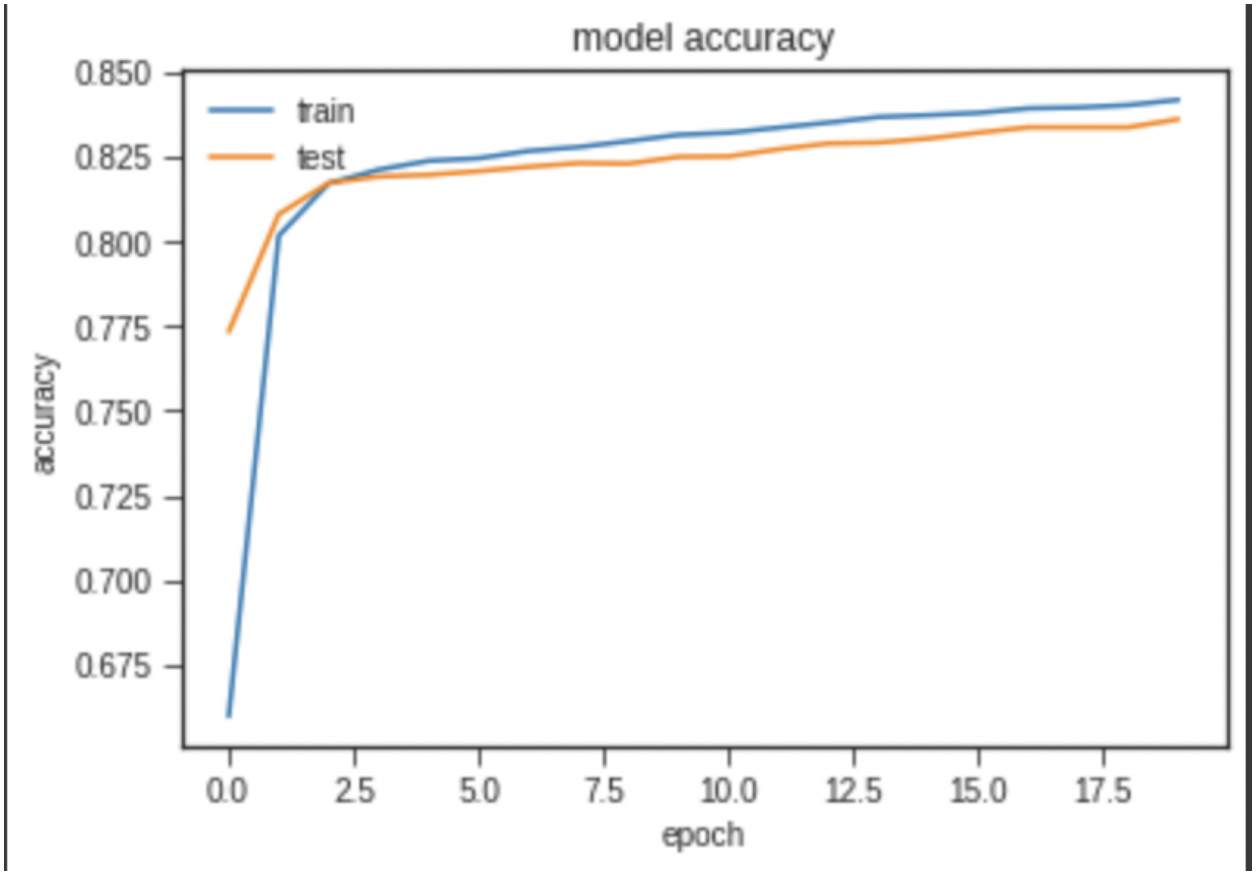
7. Activation function = Selu



8. Activation function = Elu



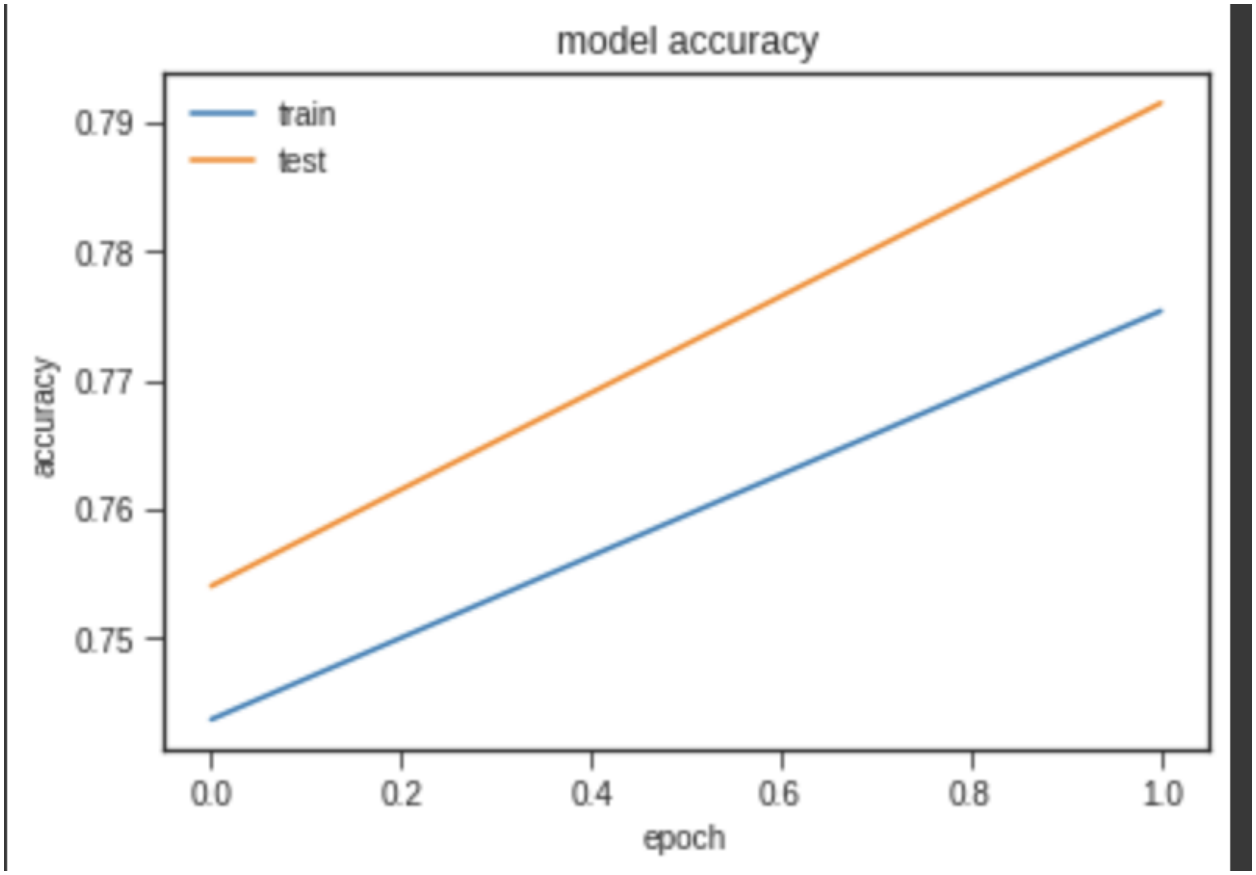
9. Activation function = Tanh



Q3) I have used different dropout values in my neural network to prevent overfitting of my model and to stop the model from memorizing the outputs. I also tried different optimizers and activation functions. Sigmoid is the common activation I have used in all the setups as it is a binary classification problem. Each activation function performed differently and have different range of outputs which helps to modify the efficiency of the model. Elu activation function gives the best accuracy out of the three setups while tanh gives the lowest. For optimizers I have first used SGD as it first measures the gradient and then makes a correction which helps in low overshooting. Second I have used Adadelta which tends to remove the decaying learning rate. In the third setup I have used RMSprop as the optimizer which helps the model to converge faster by taking large steps hence less time for training.

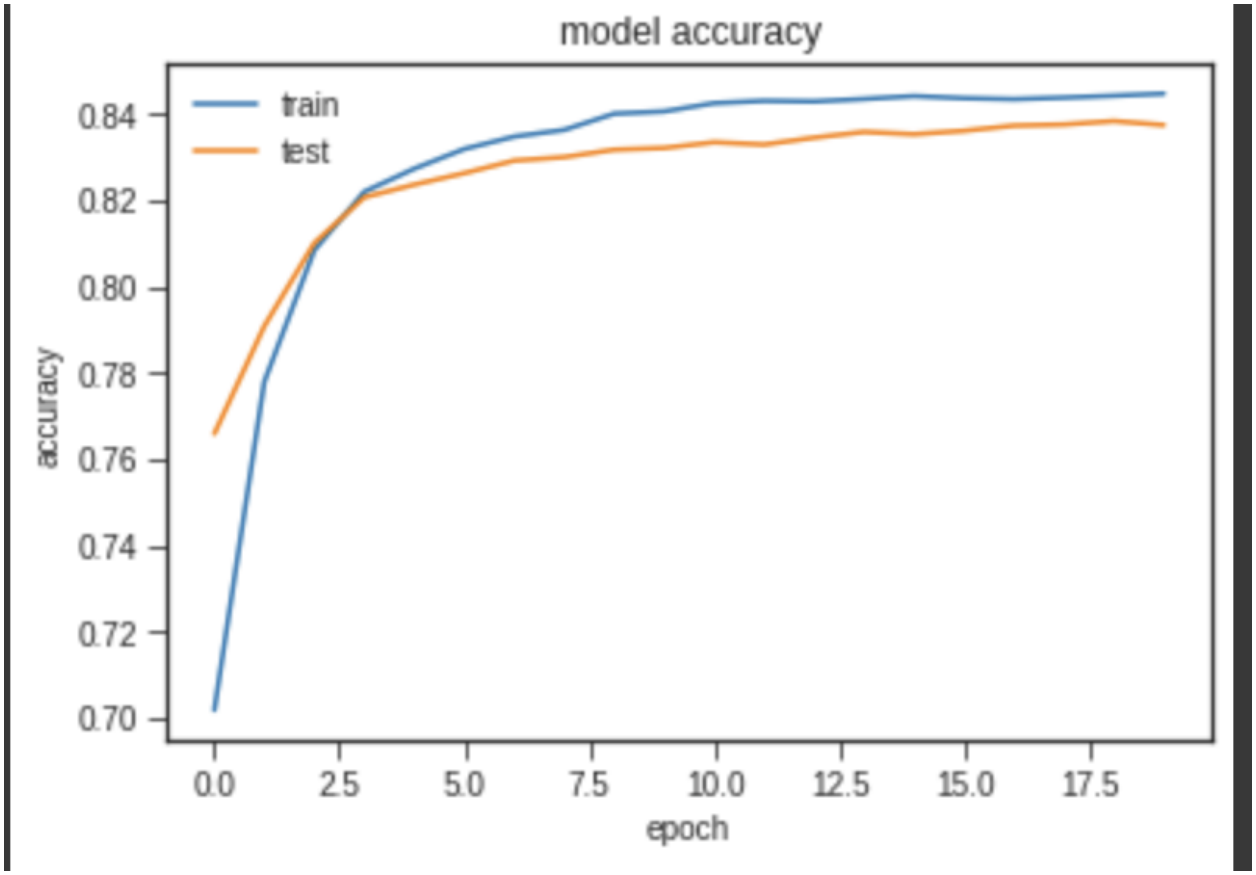
Q4) Methods used to improve the accuracy of our model :

1. Early Stopping : The goal of early stopping is to minimize the loss at every epoch it will check whether the loss is decreasing or not. Accuracy graph while using early stopping:



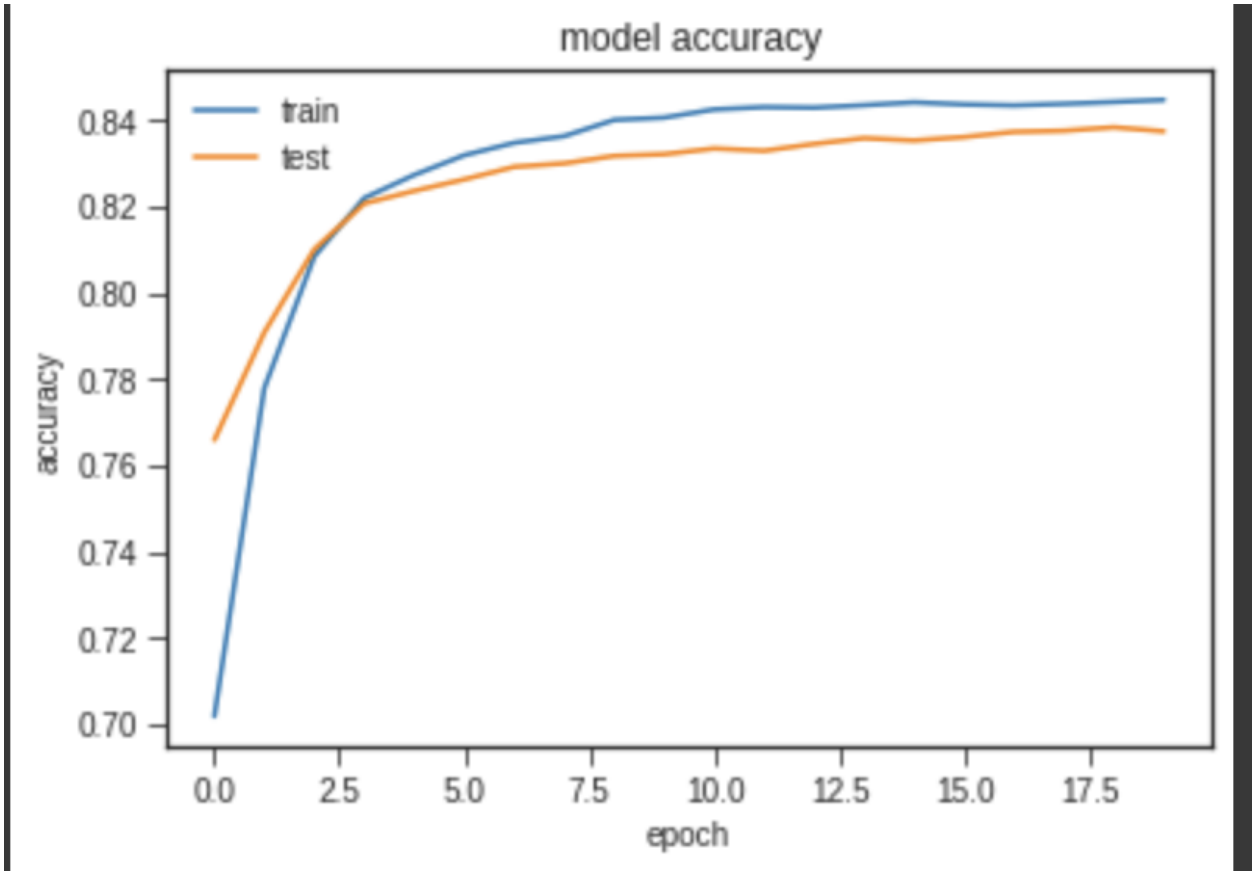
2. KFold : It is basically used for cross validation of our model on new data. In my

model I have taken the value of k as 2 as the goal of the model is to predict two classes. Accuracy while using KFold is 84.52.



3. Batch Normalization: The main motive behind using batch normalization is to

normalize the output of the previous layer and it also allows each layer of the network to learn more independently.



Part -3

In this part we are provided with Fashion_MNIST dataset which have images of size (28*28). In this dataset we 10 unique labels which are

```
['T-shirt/Top',
                'Trouser',
                'Pullover',
                'Dress',
                'Coat',
                'Sandal',
                'Shirt',
                'Sneaker',
                'Bag',
                'Ankle boot']
```

The Dataset is divided in Training and testing datasets.

Training_dataset = (60000,28,28)

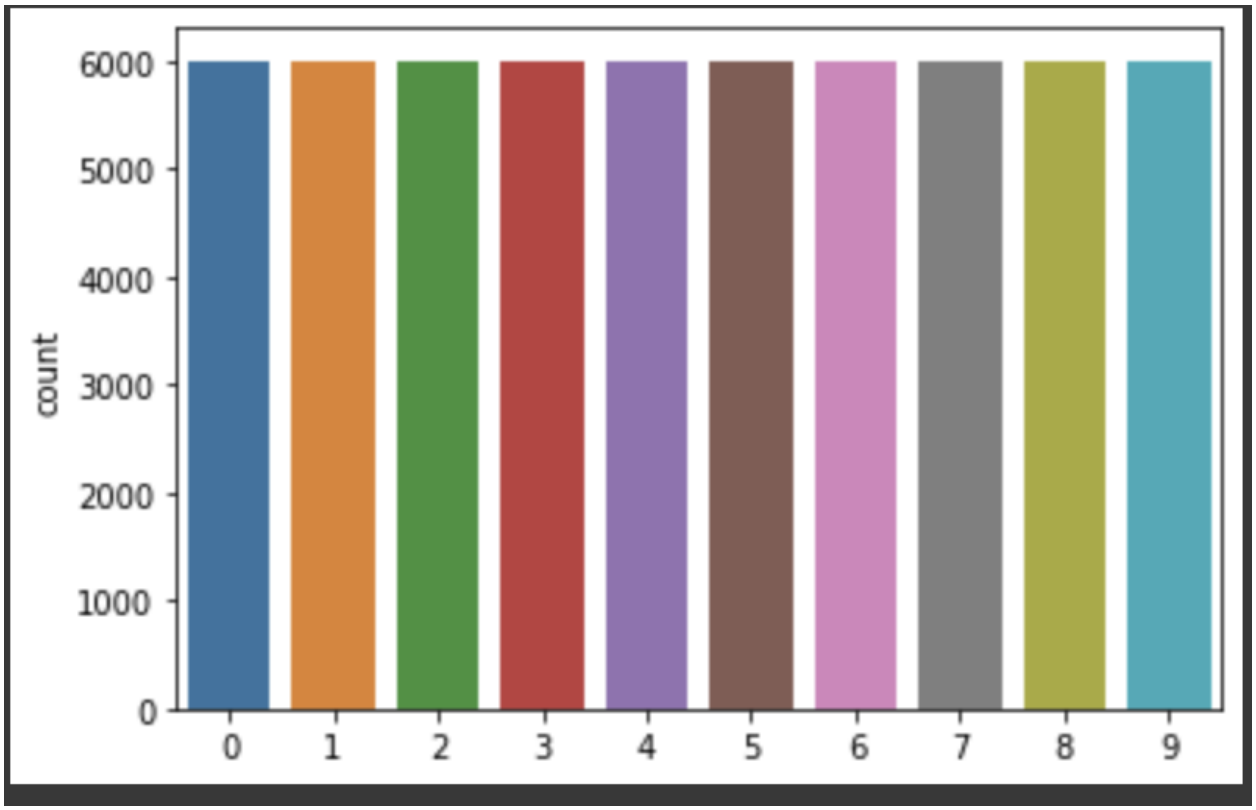
Testing_dataset = (10000,28,28)

Data type = Int type

Unique_Labels = [0,1,2,3,4,5,6,7,8,9]

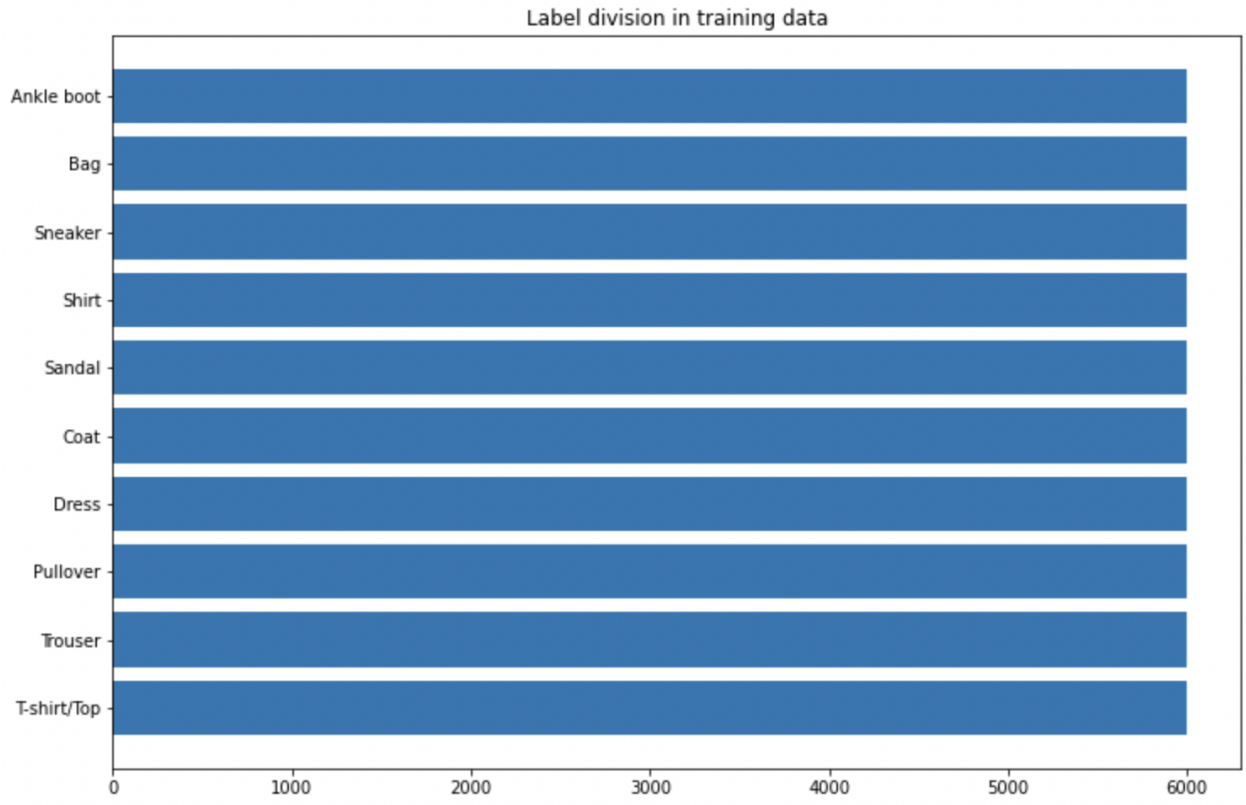
#. Visualization Graphs

1. Count Plot

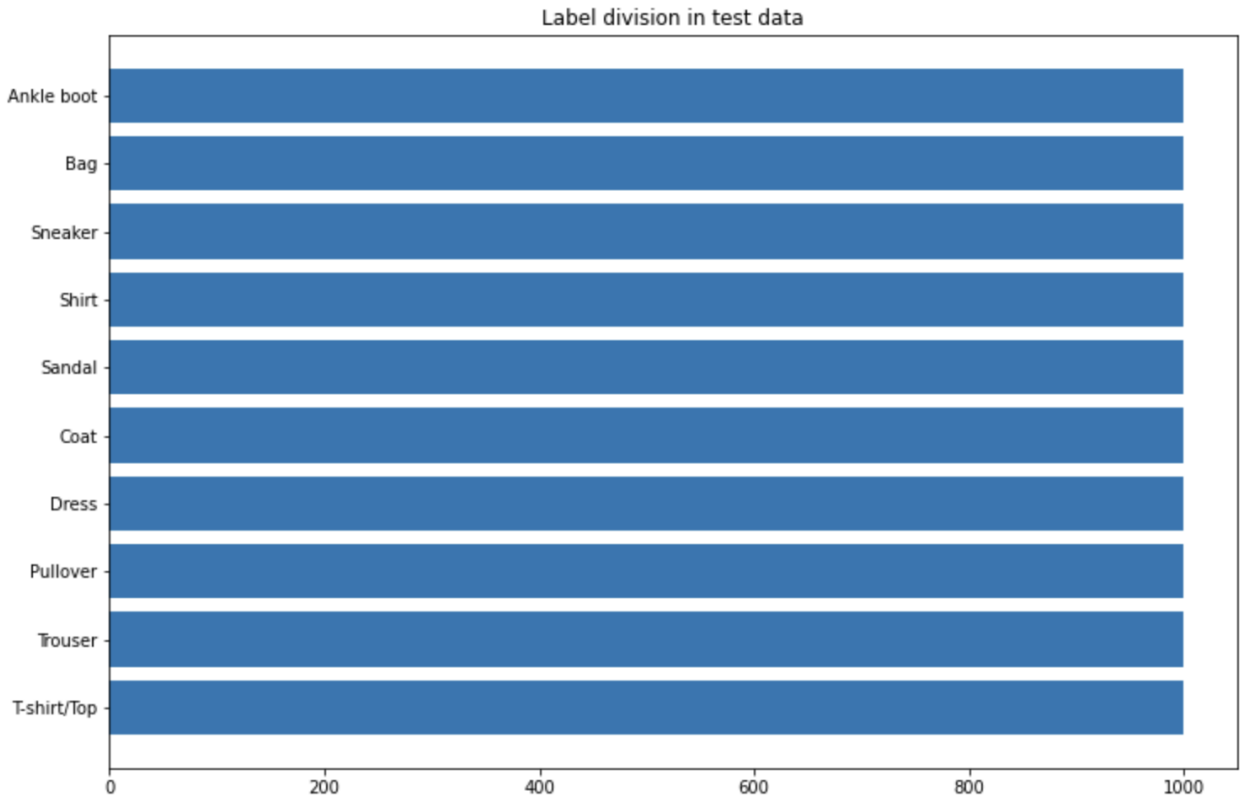


This graph depicts the count of each class in the training dataset.

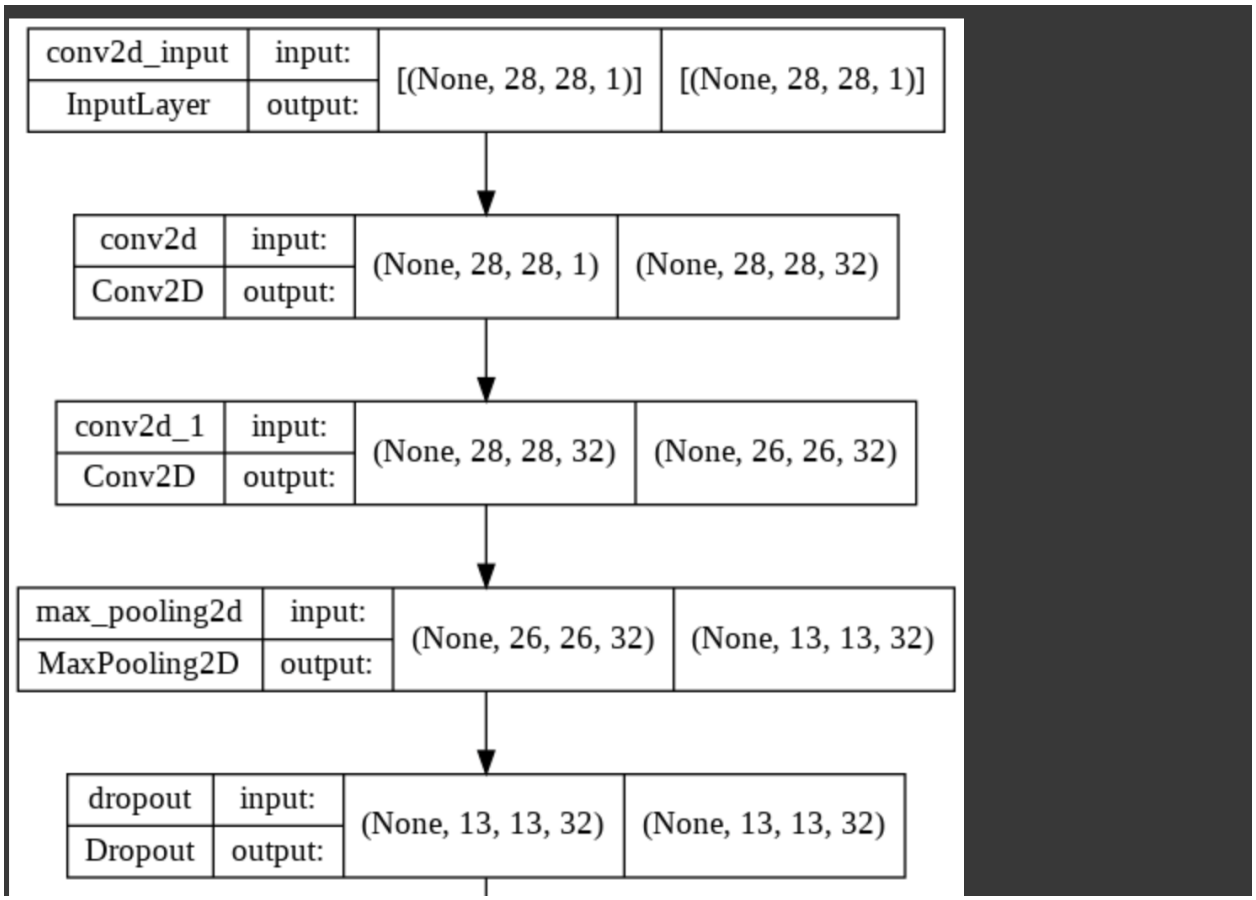
2. Count Plot with labels in training dataset

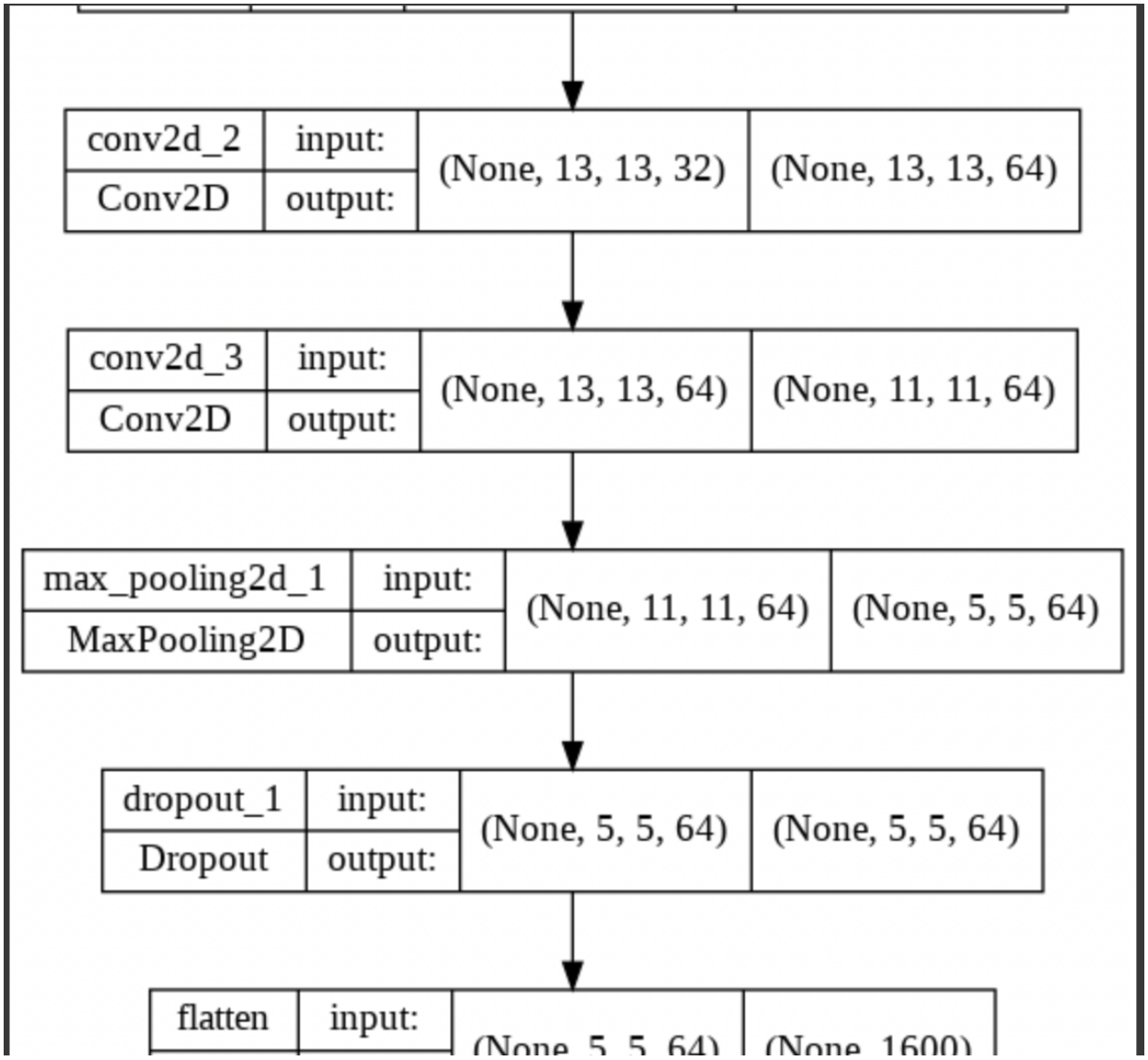


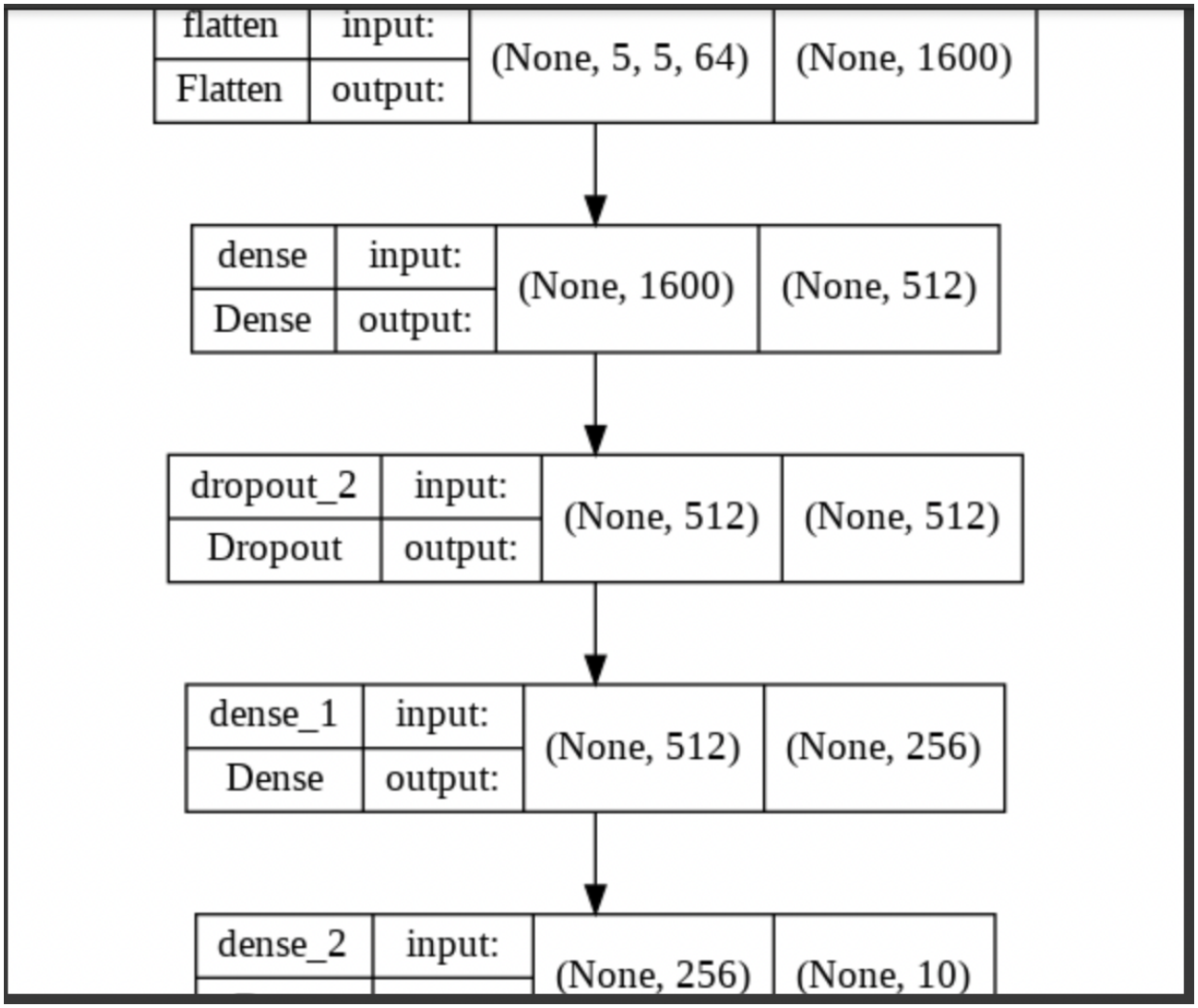
3. Count Plot with labels in testing dataset



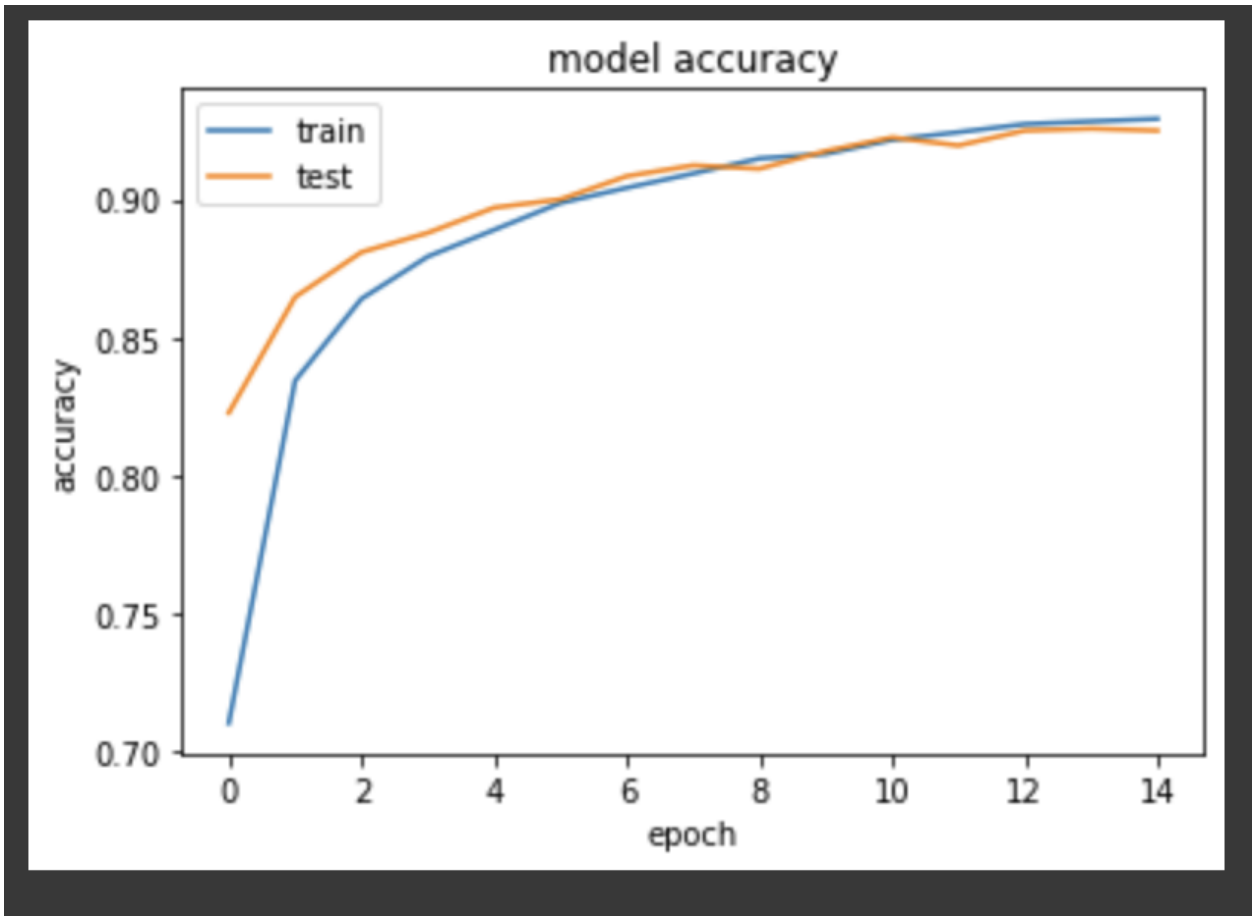
Q3) Architecture of CNN


conv2d_input
InputLayer
input:
output:
[(None, 28, 28, 1)]
[(None, 28, 28, 1)]
conv2d
Conv2D
input:
output:
(None, 28, 28, 1)
(None, 28, 28, 32)
conv2d_1
Conv2D
input:
output:
(None, 28, 28, 32)
(None, 26, 26, 32)
max_pooling2d
MaxPooling2D
input:
output:
(None, 26, 26, 32)
(None, 13, 13, 32)
dropout
Dropout
input:
output:
(None, 13, 13, 32)
(None, 13, 13, 32)

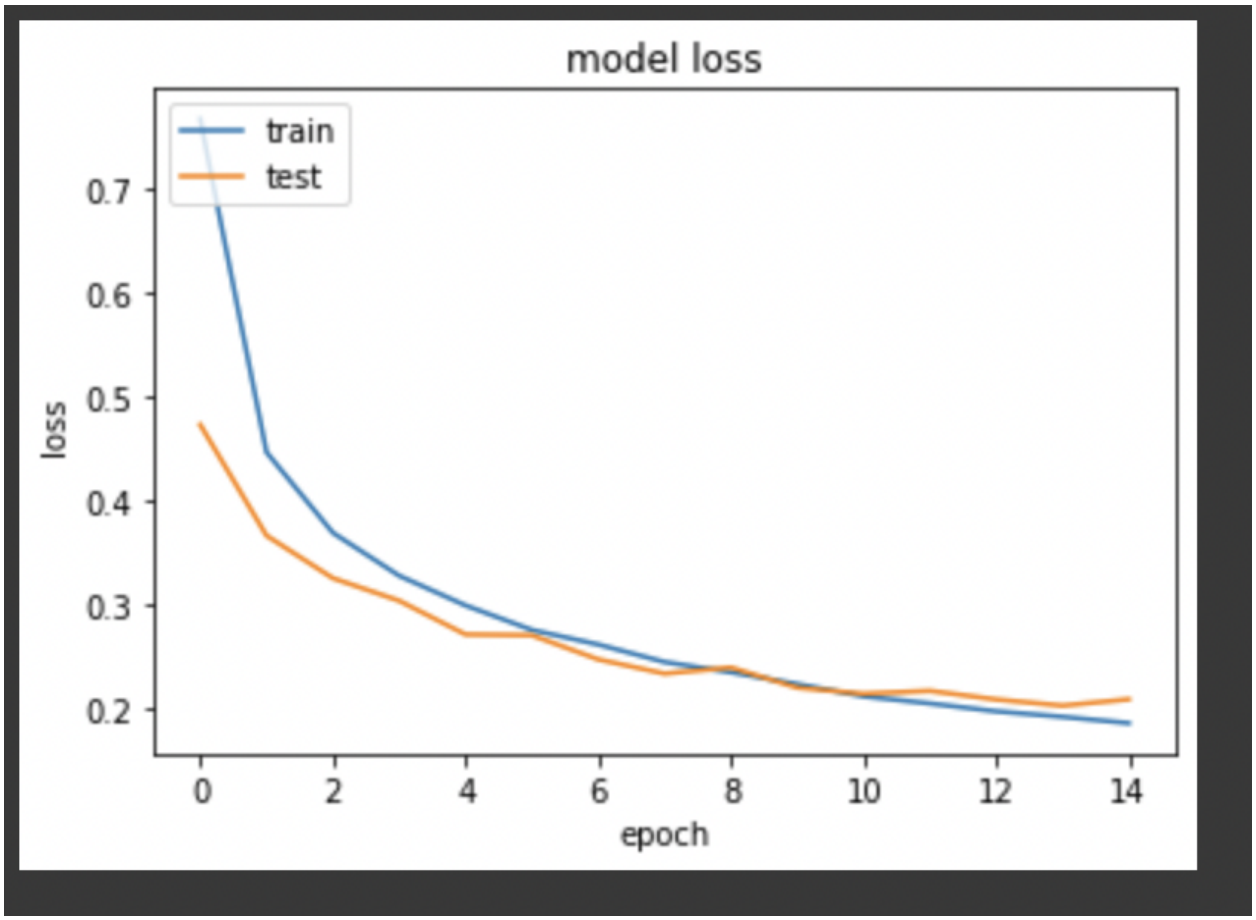

conv2d_2
input:
Conv2D
output:
(None, 13, 13, 32)
(None, 13, 13, 64)
conv2d_3
input:
Conv2D
output:
(None, 13, 13, 64)
(None, 11, 11, 64)
max_pooling2d_1
input:
MaxPooling2D
output:
(None, 11, 11, 64)
(None, 5, 5, 64)
dropout_1
input:
Dropout
output:
(None, 5, 5, 64)
(None, 5, 5, 64)
flatten
input:
(None, 5, 5, 64)
(None, 1600)


flatten
input:
Flatten
output:
(None, 5, 5, 64)
(None, 1600)
dense
input:
Dense
output:
(None, 1600)
(None, 512)
dropout_2
input:
Dropout
output:
(None, 512)
(None, 512)
dense_1
input:
Dense
output:
(None, 512)
(None, 256)
dense_2
input:
(None, 256)
(None, 10)

5. Training and Test accuracy



Train and Test loss



Part-4

1. Changing Dropout

Dropout_value	Accuracy
0.3	98.68
0.5	95.64
0.7	92.91

2. Changing Optimizer

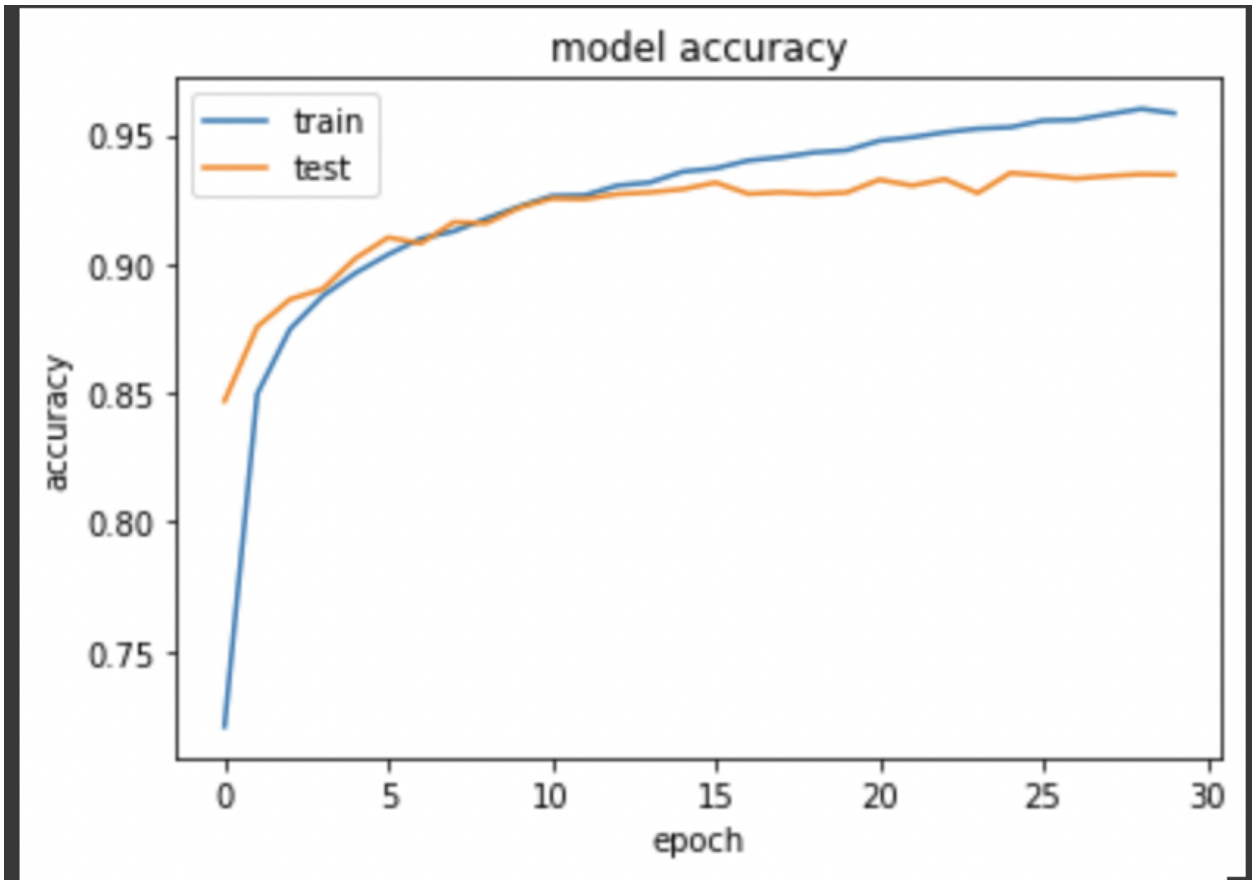
Optimizer	Accuracy
SGD	72.15
Adadelta	20.63
RMSprop	92.47

3. Changing Activation Function

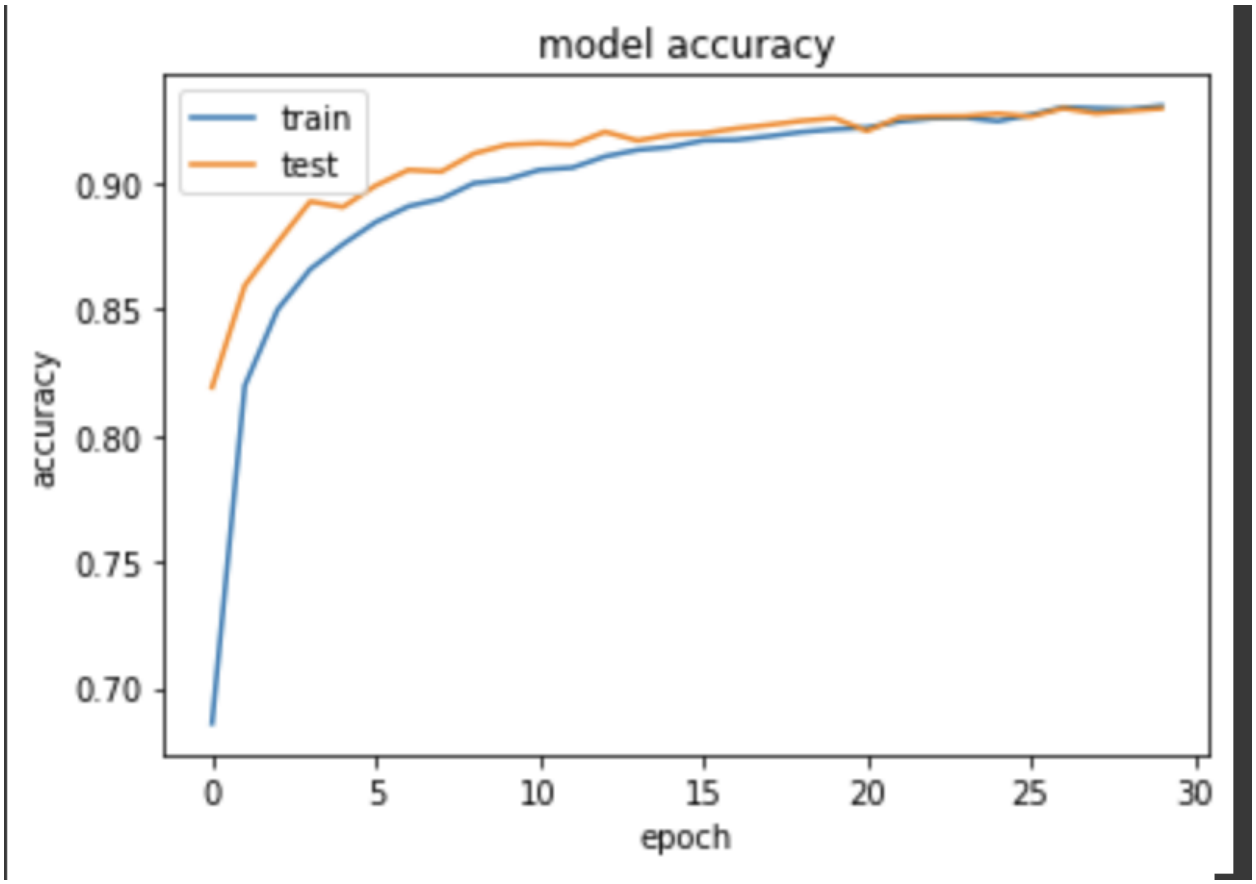
Activation Function	Accuracy
Selu	91.68
Elu	91.98
Tanh	92.12

Q2) Training and test accuracy

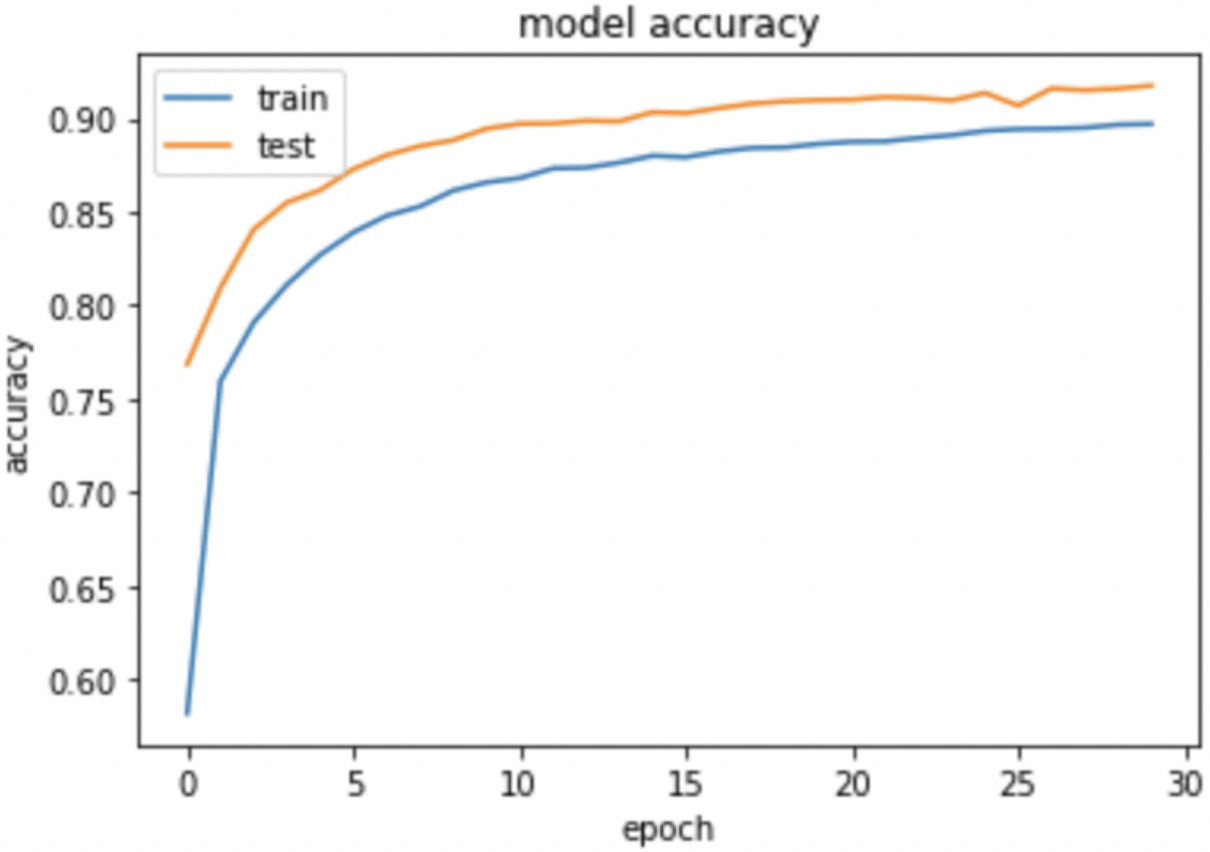
1. Dropout = 0.3


model accuracy
train
test
accuracy
0.95
0.90
0.85
0.80
0.75
0
5
10
15
20
25
30
epoch

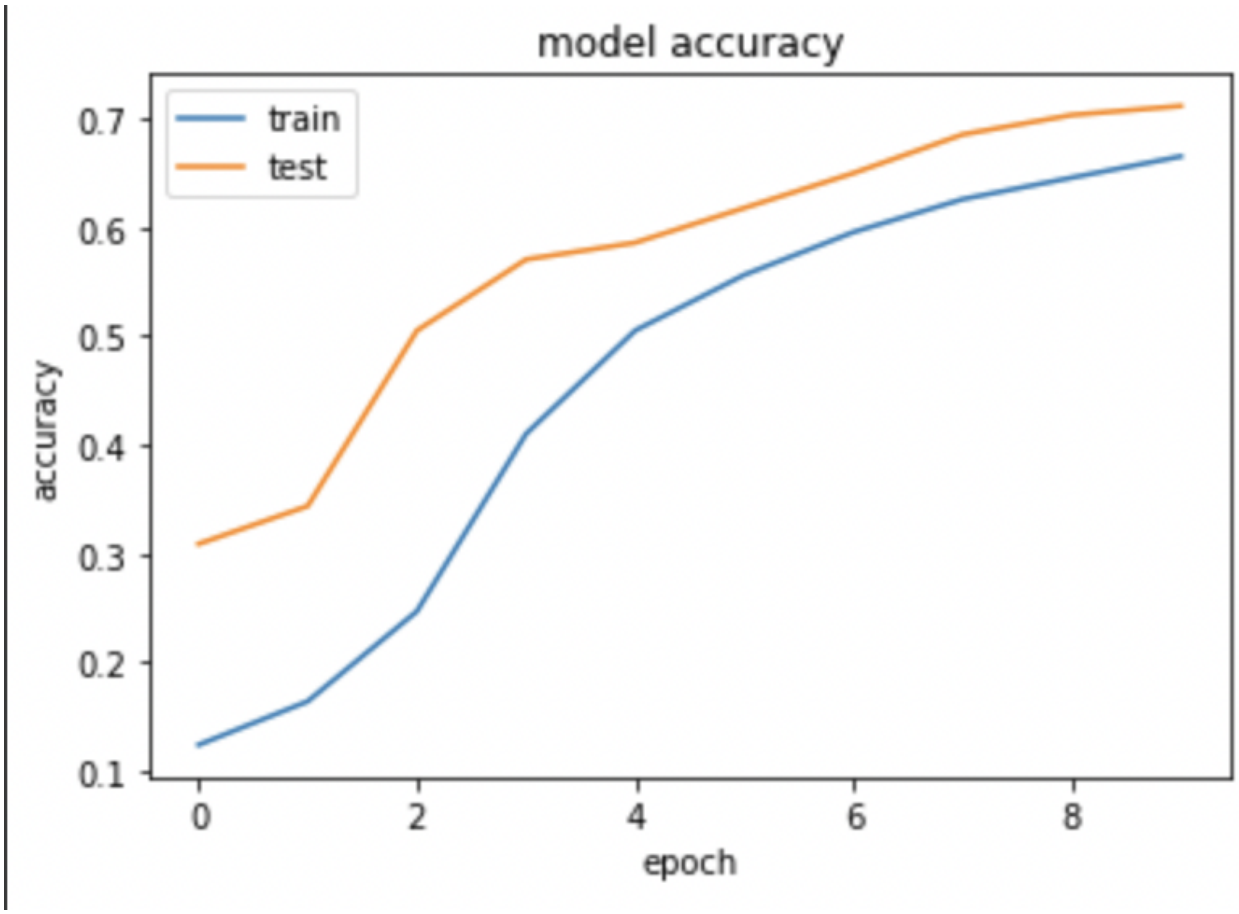
2. Dropout = 0.5


model accuracy
train
test
accuracy
0.90
0.85
0.80
0.75
0.70
0
5
10
15
20
25
30
epoch

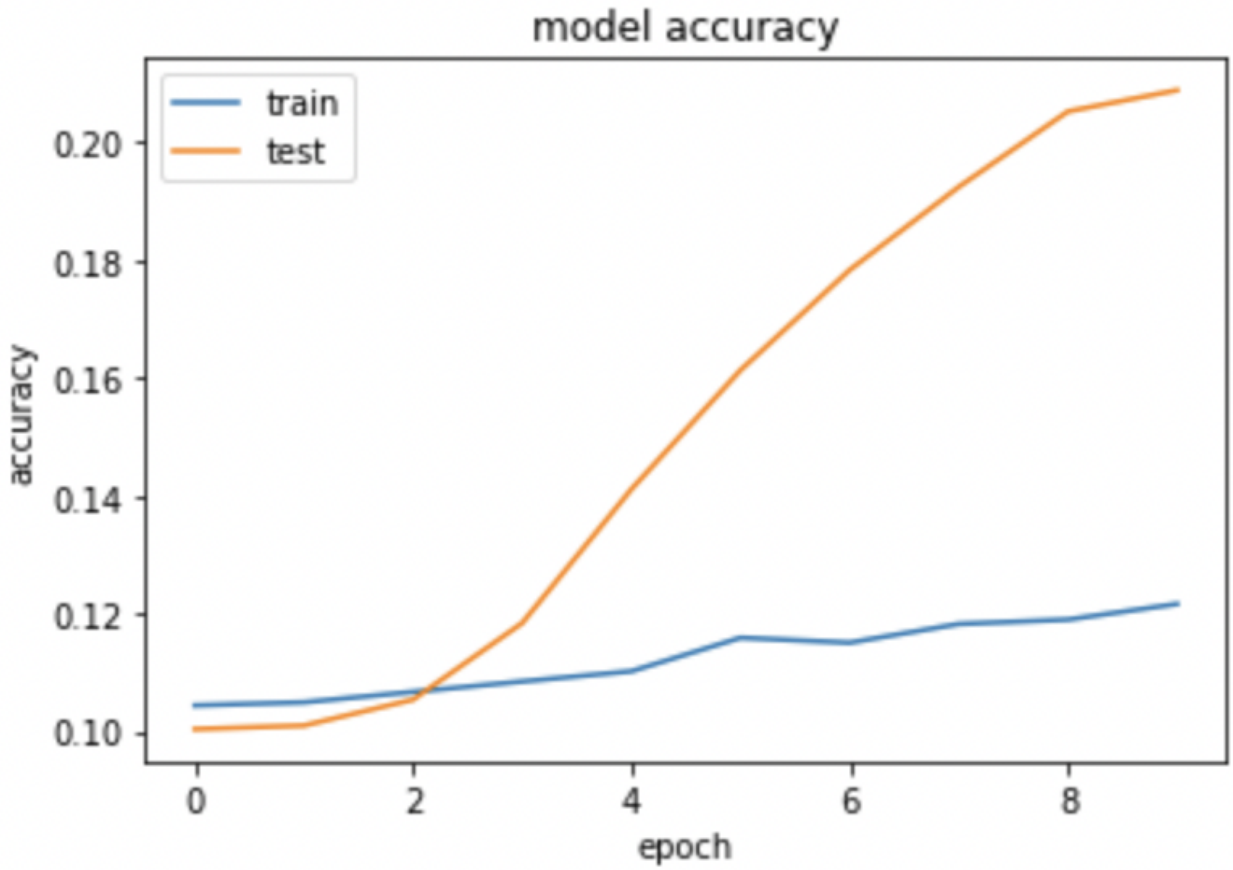
3. Dropout = 0.7


model accuracy
train
test
accuracy
epoch
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0
5
10
15
20
25
30

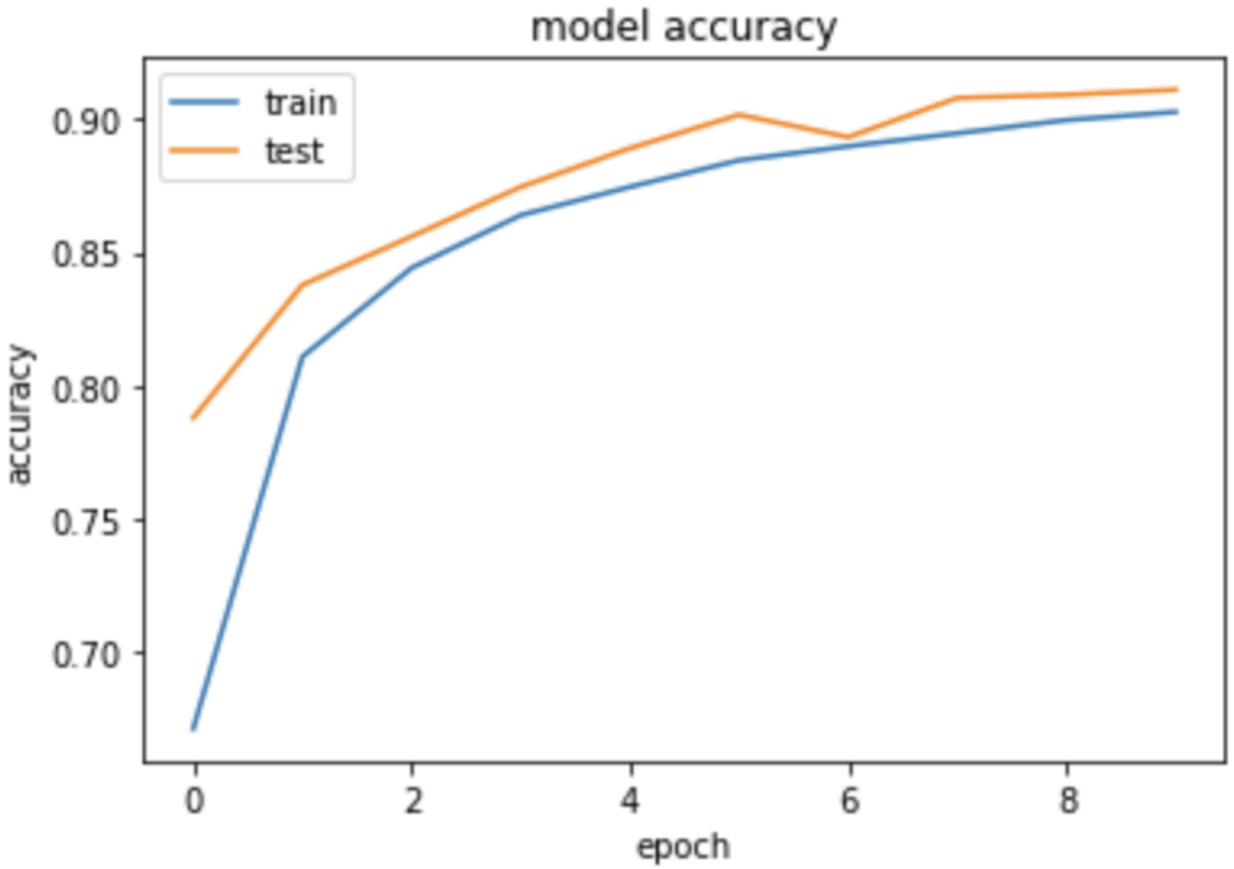
4. Optimizer = SGD



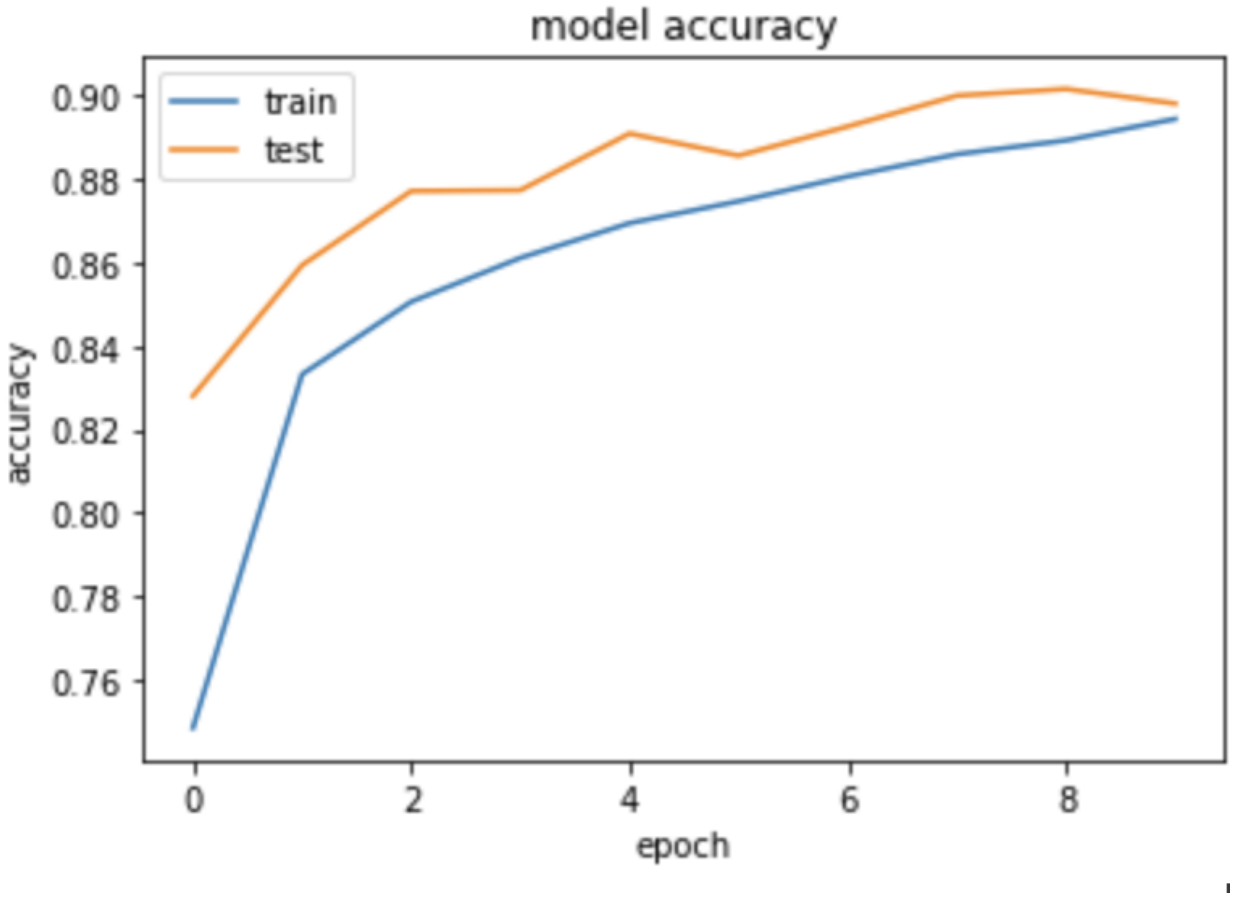
5. Optimizer = Adadelta



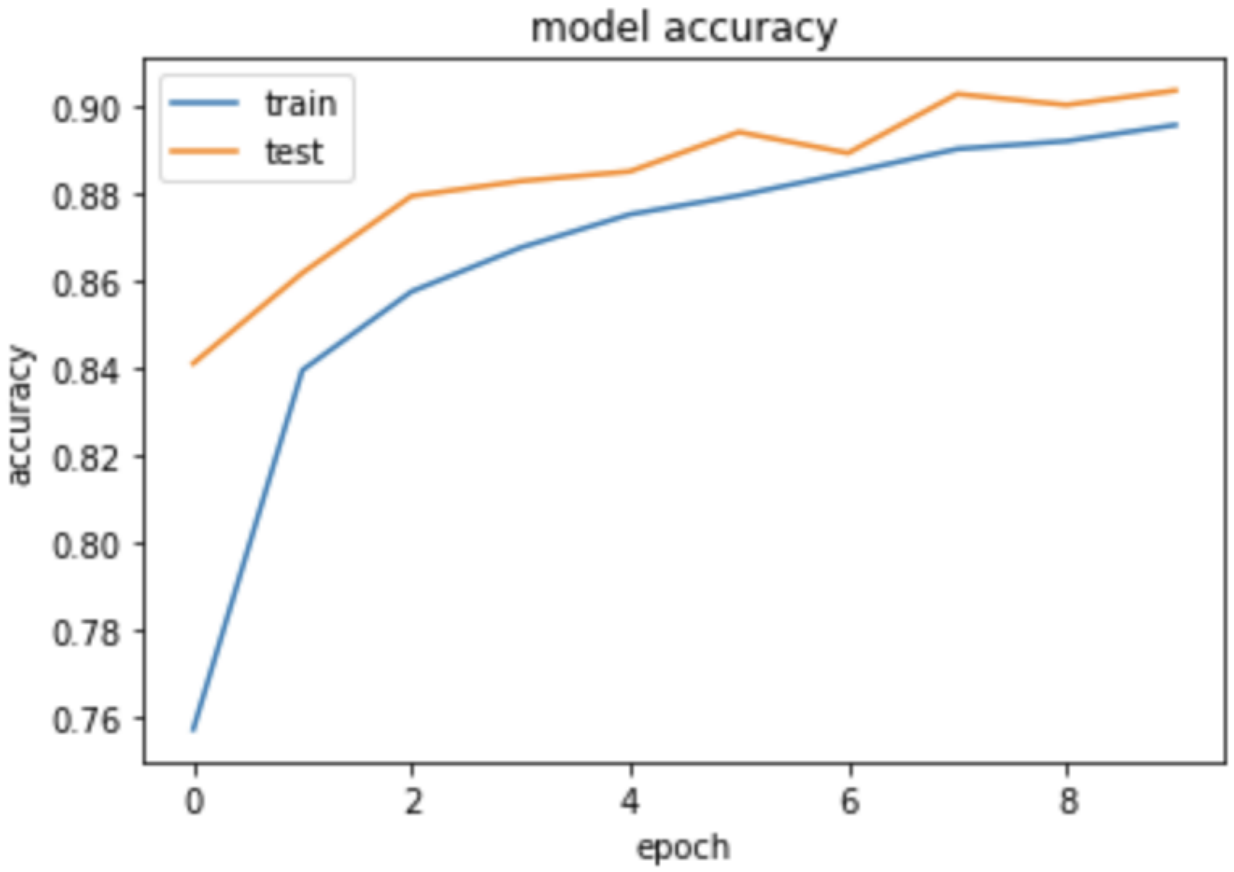
6. Optimizer = RMSprop



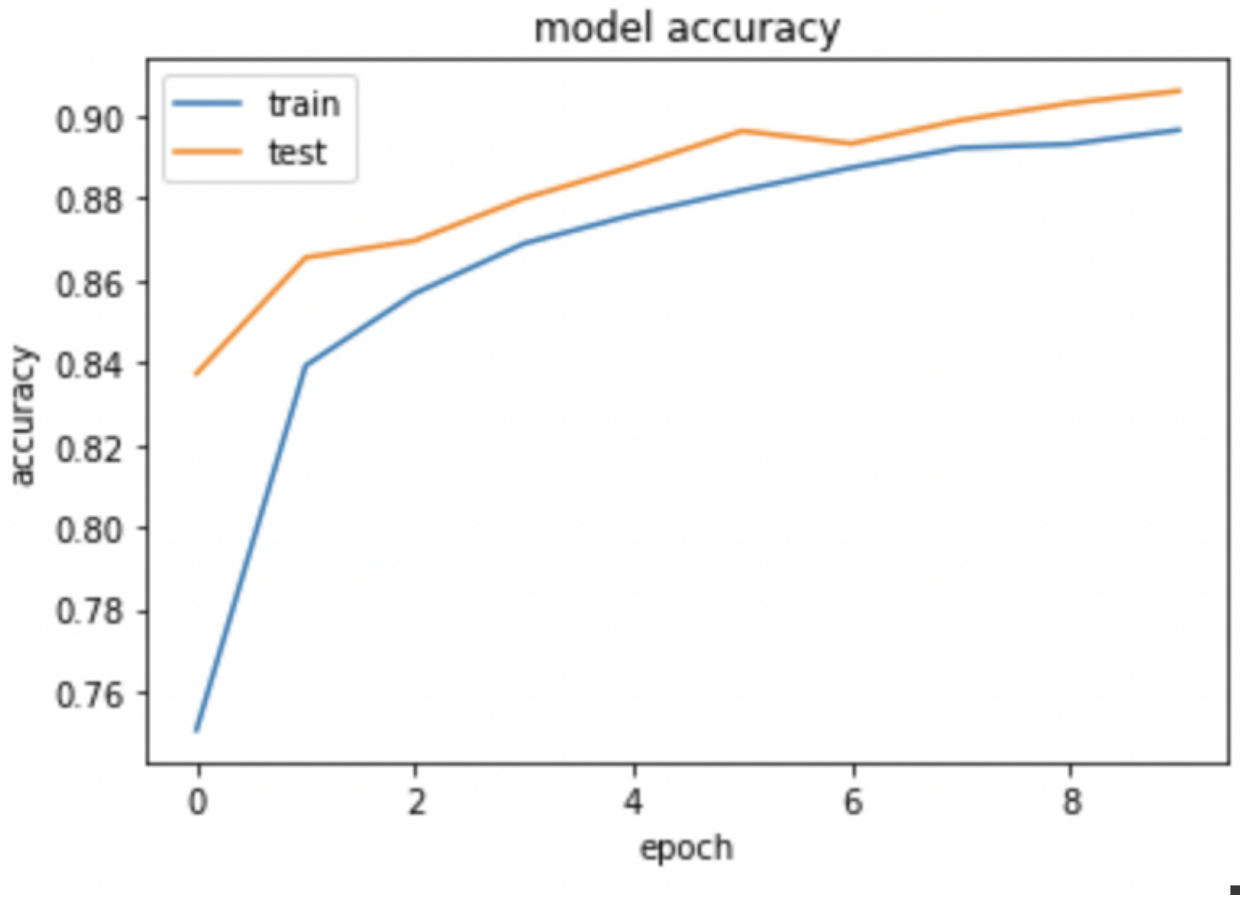
7. Activation Function = Selu



8. Activation Function = Elu



9. Activation Function = Tanh



Q3) I have used different dropout values in my neural network to prevent overfitting of my model and to stop the model from memorizing the outputs. I also tried different optimizers and activation functions. Sigmoid is the common activation I have used in all the setups as it is a binary classification problem. Each activation function performed differently and have different range of outputs which helps to modify the efficiency of the model. Elu activation function gives the best accuracy out of the three setups while tanh gives the lowest. For optimizers I have first used SGD as it first measures the gradient and then makes a correction which helps in low overshooting. Second I have used Adadelta which tends to remove the decaying learning rate. In the third setup I have used RMSprop as the optimizer which helps the model to converge faster by taking large steps hence less time for training.

Q4) For data augmentation I have used random rotation, horizontal flip and random zoom. The benefit of using data augmentation is that it helps our model to learn what an object generally looks rather than memorizing it in a specific way as it appears in our training data.