DEEP-LEARNING ASSIGNMENT-2
MANASVI AGGARWAL (MTECH. (RES.))
SR. NO. 16223

Part1: Train a neural network on Fashion MNIST dataset.

My model specifications are as follows:

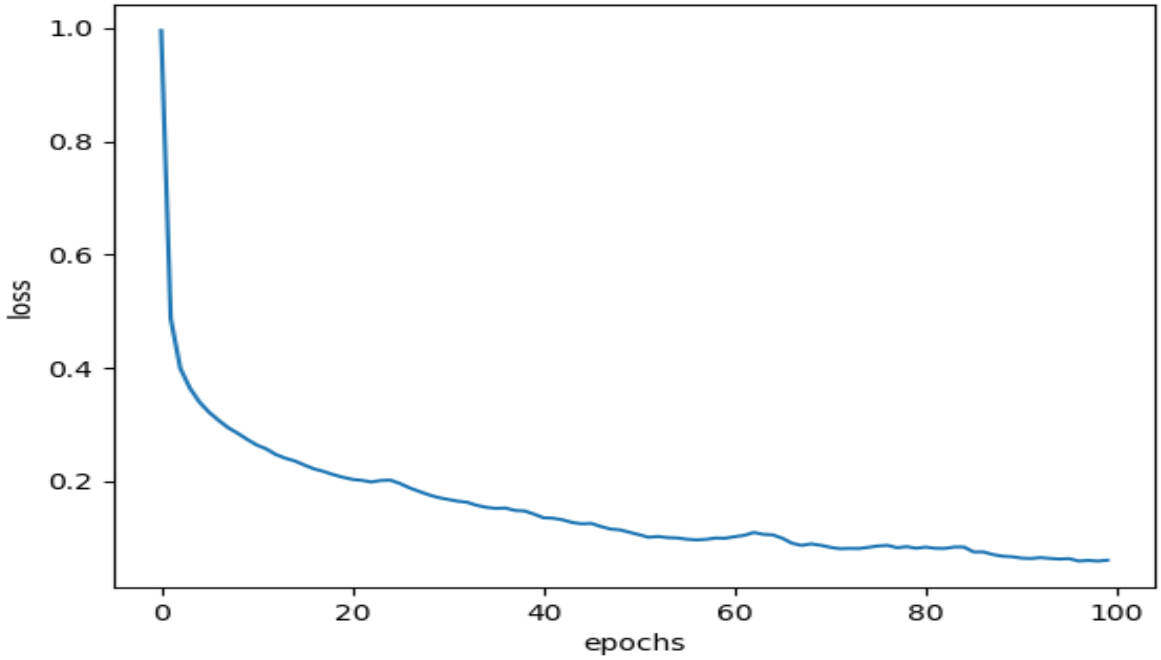
#layers 4
#neurons in each layer [512,128,16,10]
activation used in each (relu)
batch size 512
#epochs 100
dropout rate 0.

I came to these specifications based on the validation loss and accuracy. And the validation loss was minimum for this configuration so I saved this model. Validation set contains 10% of the train set. 90% of the train set is used for training. At the end I tested my saved model on the test set. I testes various configurations. For #layers I tested for 2,3,4 and for #neurons I tesetd for 16,64,256. Training accuracy of the saved model is 96% and test accuracy is 89.35%. I choose current architecture as validation loss was minimum on this architechture. Also, more complex model requires more data therefore, I didn't choose more than 4 layers. Also, Learning rate is set to 0.001 as higher learning rate will lead to oscillations during training.
Confusion Matrix for NN model is (on test set):

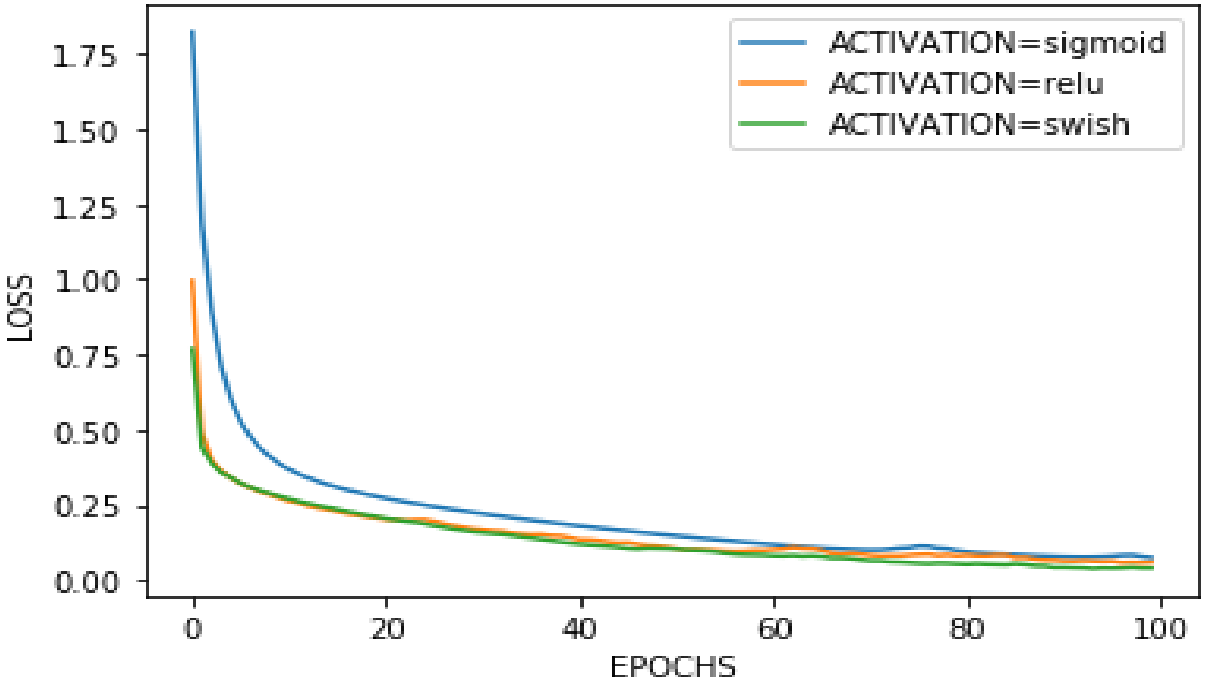
```
[[879   1  29  21   9   0  50   0  11   0]
 [  2 986   0   7   4   0   0   0   1   0]
 [ 24   1 851  11  75   0  36   0   2   0]
 [ 25   6  12 885  46   0  20   0   5   1]
 [  1   1  81  21 870   0  24   0   2   0]
 [  0   0   0   0   0 960   0  24   1  15]
 [167   3 108  23  81   0 602   0  16   0]
 [  0   0   0   0   0   6   0 978   0  16]
 [  3   0   3   1   5   1   3   5 979   0]
 [  0   0   0   0   0   8   1  46   0 945]]
```

Given are some plots for some configurations which I tested: (All plots on y axis denotes the train accuracy/train loss) :

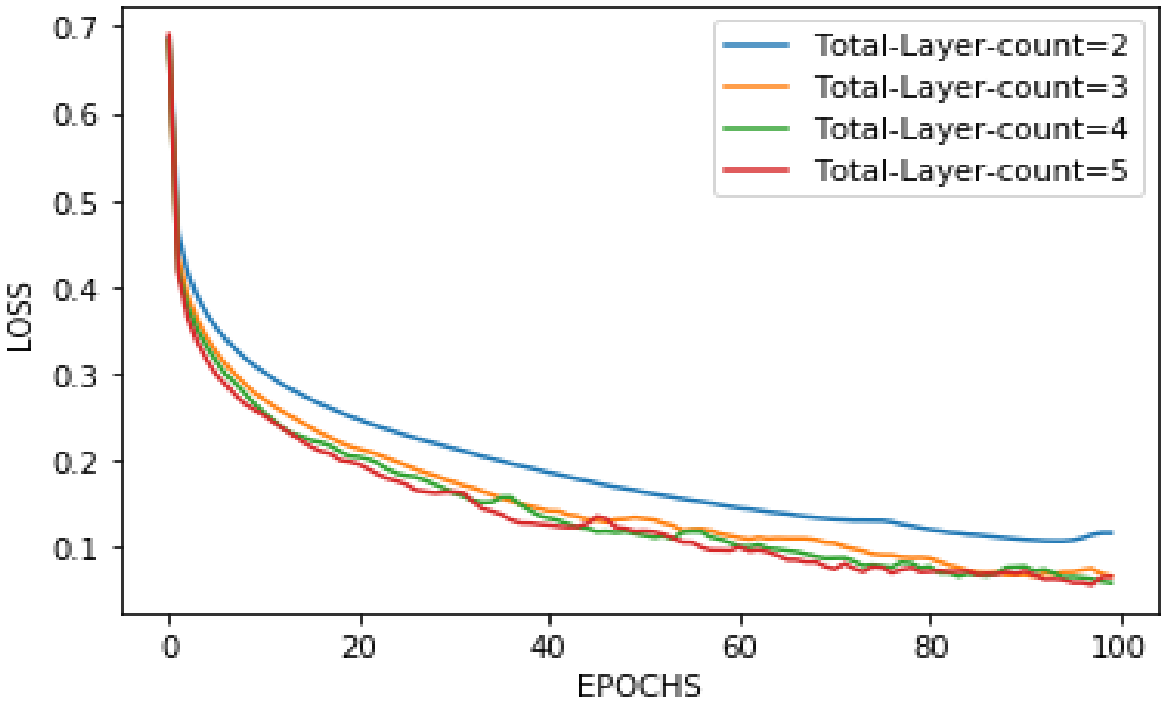
Plot 1: EPOCHS VS LOSS



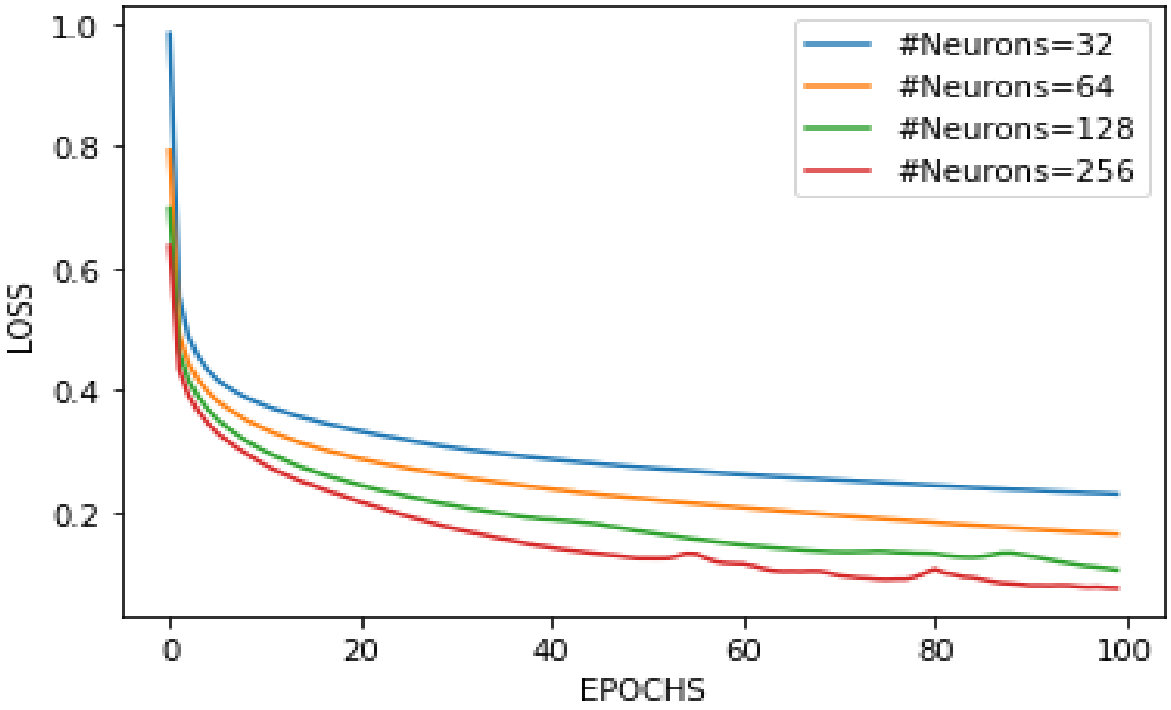
Plot 2: VARYING ACTIVATION FUNCTIONS



Plot 3: DIFFERENT NUMBER OF LAYERS



Plot 4: VARYING NUMBER OF NEURONS



PART 2: Train a CNN on Fashion MNIST dataset.

My model specifications are as follows:

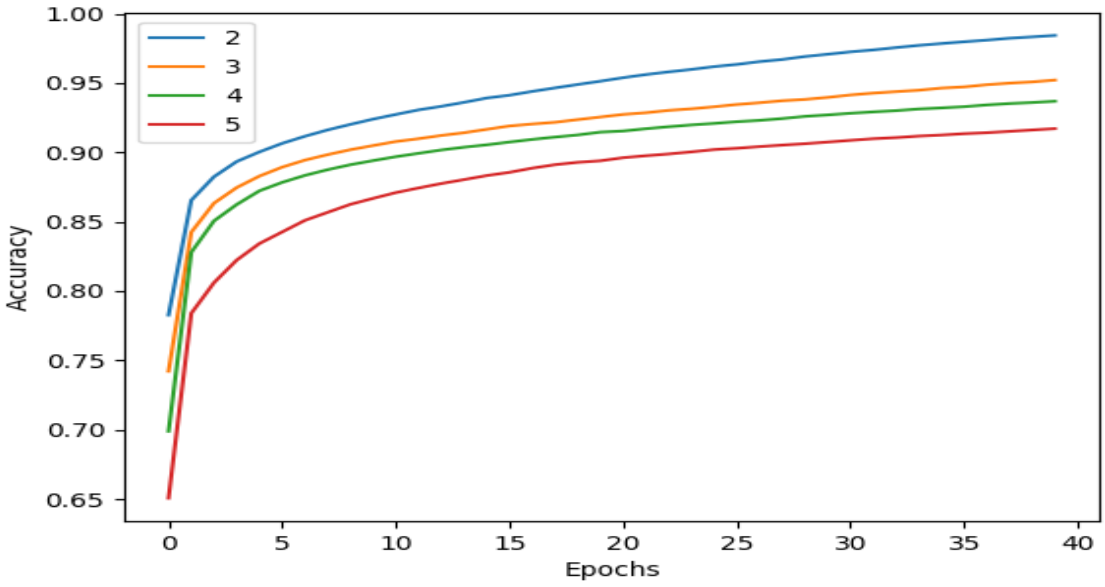
#CNN layers 2
Filter size 5,2
activation used in each layer relu
batch size 128
#epochs 30
#filters in each CNN layer 32,64
#FCC layers 3
#neurons in each FCC layer 512,10

I came to these specifications based on the validation loss and accuracy. I keep doubling the #channels in each CN layer as input to a CNN layer is decreasing after every CNN layer. And the validation loss was minimum for this configuration so I saved this model. Validation set contains 10% of the train set. 90% of the train set is used for training. At the end I tested my saved model on the test set. I testes various configurations. For #CNN/FC layers I tested for 2,3,4 and for #neurons I tested for 16,32,64,128,256 for each FC layer. Training accuracy of the saved model is 97% and test accuracy is 91.55%. More complex model requires more data therefore, I didn't choose more than 2 CNN layers and 2 FCL. Also, model is converging and hence these configurations are used. The early stopping is used while training CNN and when validation loss starts increasing for some epochs I stop the training. Also, Learning rate is set to 0.001 as higher learning rate will lead to oscillations during training. Confusion matrix is (on test set):

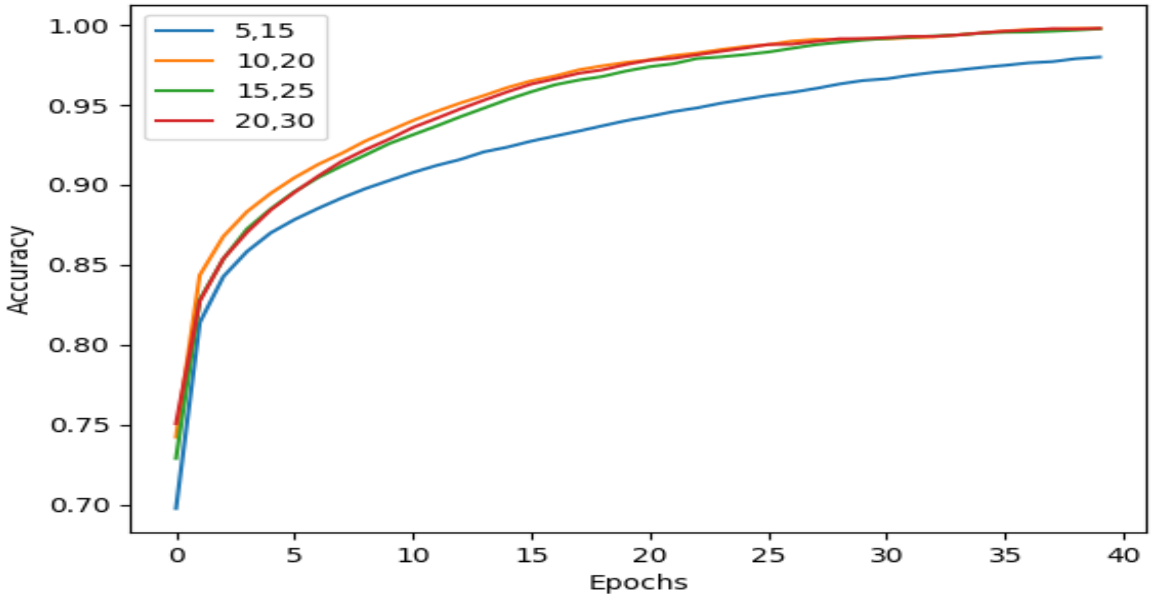
```
[[847   0  18  15   7   1 107   0   4   1]
 [  3 979   3   9   4   0   2   0   0   0]
 [ 15   1 853   7  71   0  53   0   0   0]
 [ 13   2   9 911  37   0  28   0   0   0]
 [  3   0  36  15 916   0  30   0   0   0]
 [  0   0   0   0   0 981   0  13   0   6]
 [101   1  59  24  97   1 714   0   3   0]
 [  0   0   0   0   0   4   0 979   0  17]
 [  3   0   6   3   3   2   3   4 976   0]
 [  0   0   0   0   0   5   1  29   1 964]]
```

Given are some plots for some configurations which I tested: (All plots on y axis denotes the train accuracy)

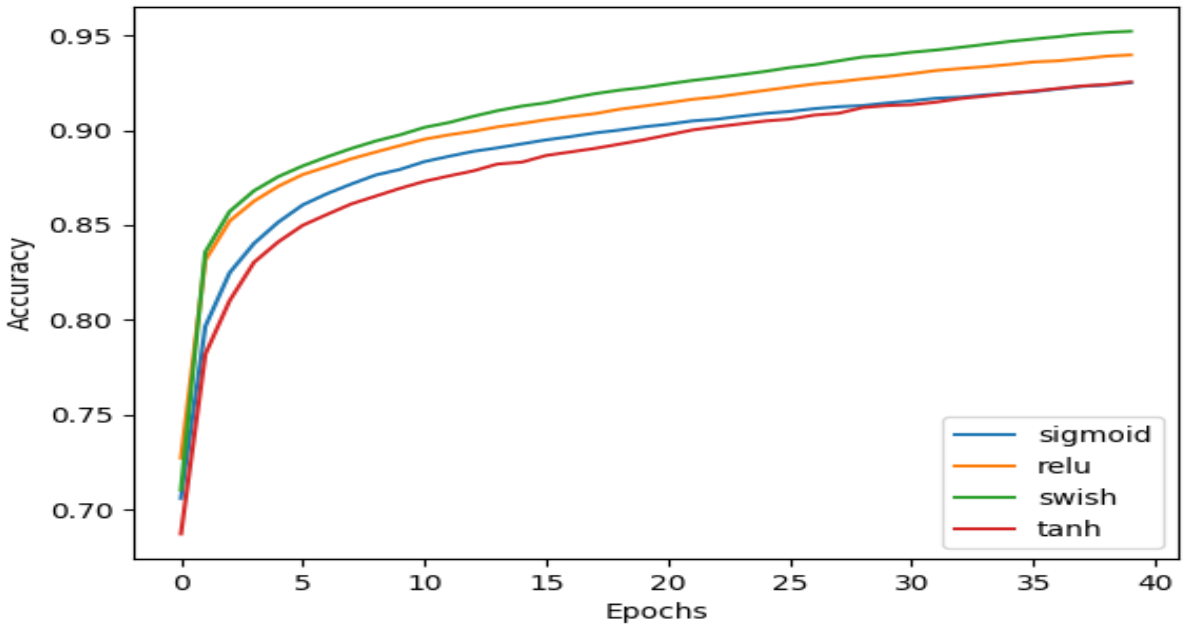
Plot 1: DIFFERENT NUMBER OF CNN LAYERS



Plot 2: VARYING NUMBER OF NEURONS



Plot 3: VARYING ACTIVATION FUNCTIONS


0.95
0.90
0.85
0.80
0.75
0.70
Accuracy
0
5
10
15
20
25
30
35
40
Epochs
sigmoid
relu
swish
tanh