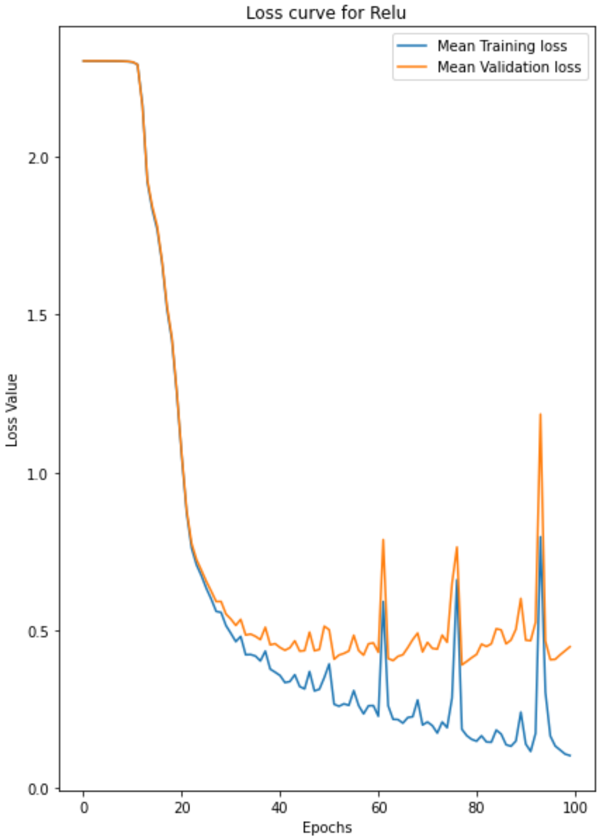
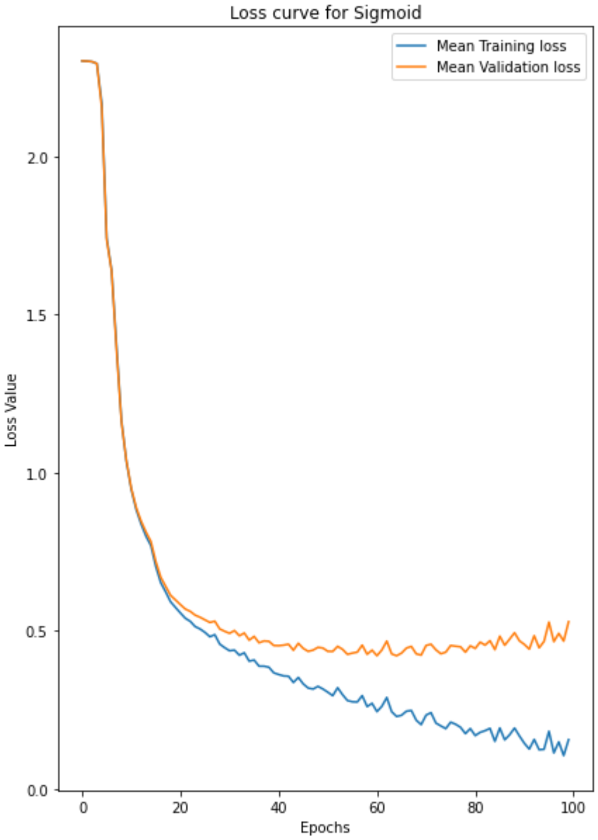
Q1)

train.csv, test.csv is first combined with each other which gives a total data of 70,000 rows and 784 features.
Since the pixel values are between 0-255, scaling is done after dividing them by 255, in order to get more accurate results.
Train,test,valid is split into 80:10:10 ratio.

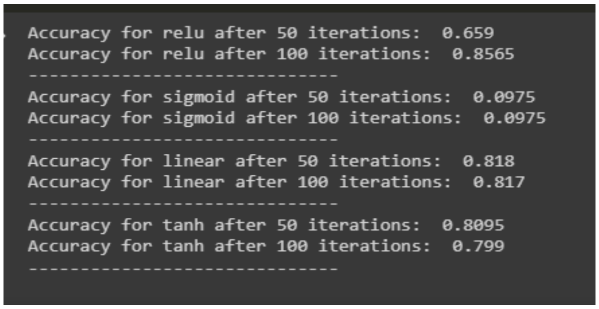
5)

Architecture of neural network=[input,256,128,64,output], with learning rate=0.1, epochs=100, normal weights initialization with optimizer as SGD with random shuffle.
b) Loss vs epoch





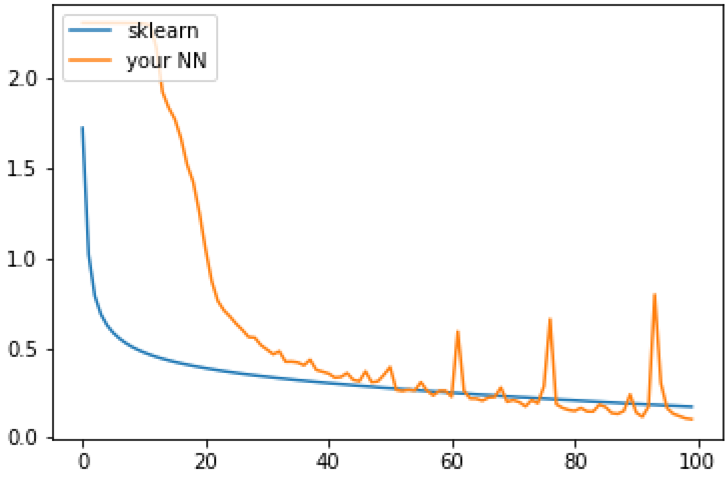
c) test set accuracies after 50 & 100 iterations



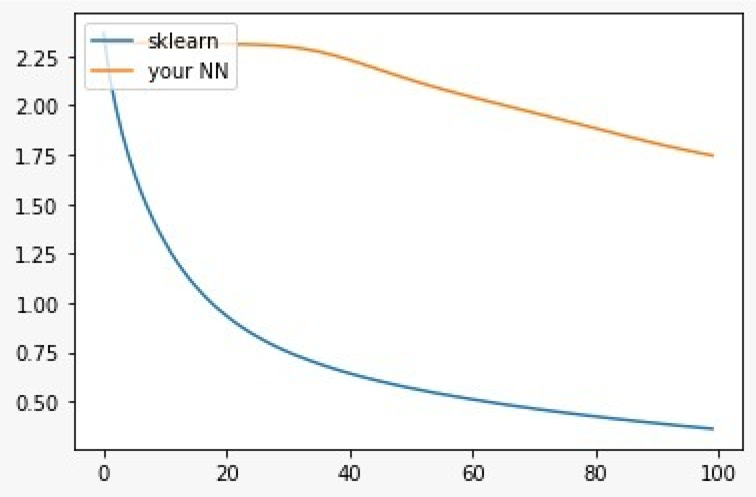
```
Accuracy for relu after 50 iterations:  0.659
Accuracy for relu after 100 iterations:  0.8565
------------------------------
Accuracy for sigmoid after 50 iterations:  0.0975
Accuracy for sigmoid after 100 iterations:  0.0975
------------------------------
Accuracy for linear after 50 iterations:  0.818
Accuracy for linear after 100 iterations:  0.817
------------------------------
Accuracy for tanh after 50 iterations:  0.8095
Accuracy for tanh after 100 iterations:  0.799
------------------------------
```

6)

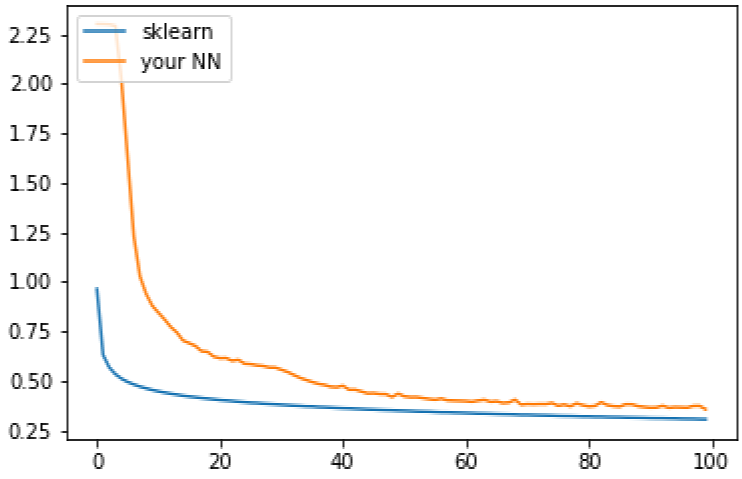
Relu :



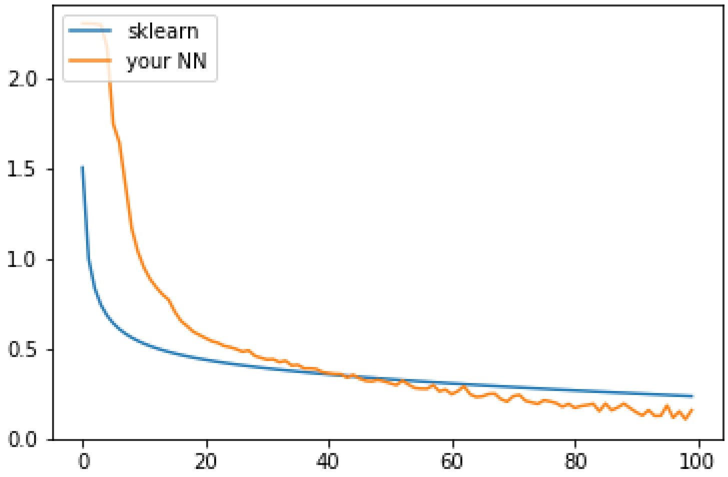
Sigmoid:

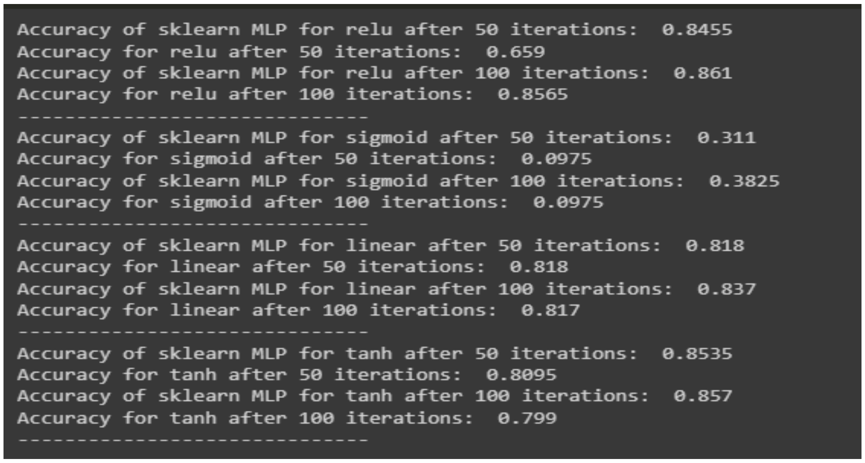


Linear:



Tanh:





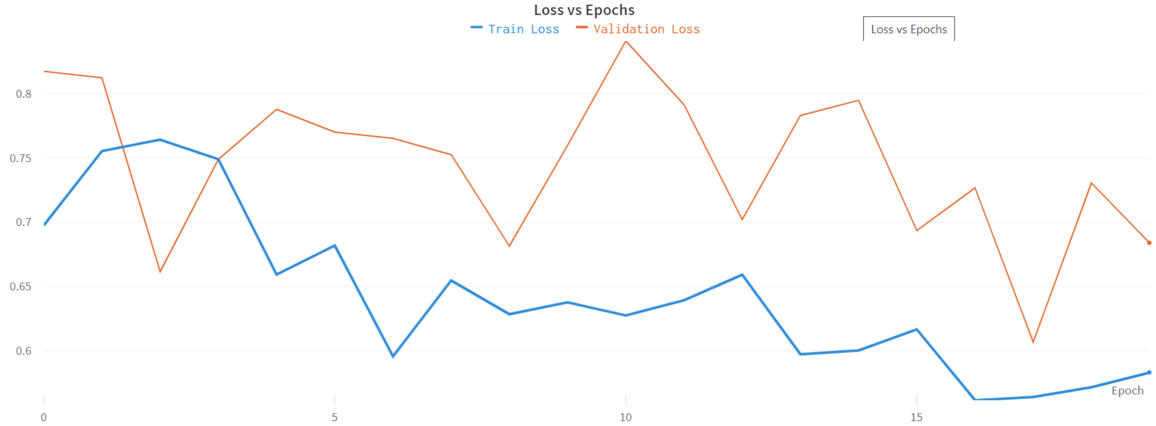
```
Accuracy of sklearn MLP for relu after 50 iterations:  0.8455
Accuracy for relu after 50 iterations:  0.659
Accuracy of sklearn MLP for relu after 100 iterations:  0.861
Accuracy for relu after 100 iterations:  0.8565
------------------------------
Accuracy of sklearn MLP for sigmoid after 50 iterations:  0.311
Accuracy for sigmoid after 50 iterations:  0.0975
Accuracy of sklearn MLP for sigmoid after 100 iterations:  0.3825
Accuracy for sigmoid after 100 iterations:  0.0975
------------------------------
Accuracy of sklearn MLP for linear after 50 iterations:  0.818
Accuracy for linear after 50 iterations:  0.818
Accuracy of sklearn MLP for linear after 100 iterations:  0.837
Accuracy for linear after 100 iterations:  0.817
------------------------------
Accuracy of sklearn MLP for tanh after 50 iterations:  0.8535
Accuracy for tanh after 50 iterations:  0.8095
Accuracy of sklearn MLP for tanh after 100 iterations:  0.857
Accuracy for tanh after 100 iterations:  0.799
------------------------------
```

Best model after checking accuracy on test set is Relu with sklearn MLP

Q2)

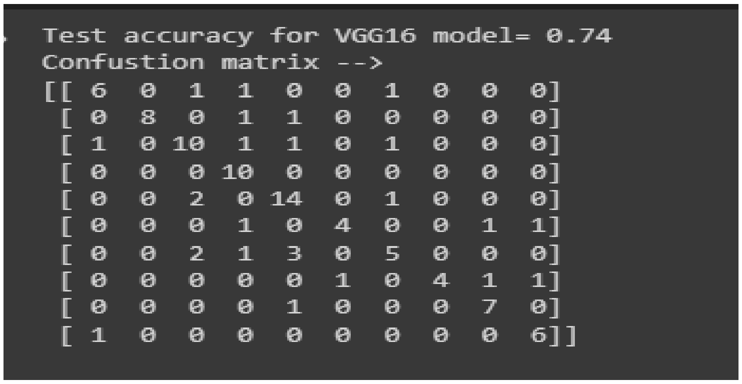
1) Preprocessing: Using pytorch transform resized the image to 224*224(since vgg16 model is pretrained on big sized images) on the and then normalized the values with mean and standard deviation as [0.485, 0.456, 0.406] and [0.229, 0.224, 0.225] respectively. Normalization ensures that each pixel value of input image is small and has same distribution so that model would converge faster.

   VGG16 Hyperparameters : "learning_rate": 0.001,
   "momentum": 0.9,
   "batch_size": 32,
   "epochs":20



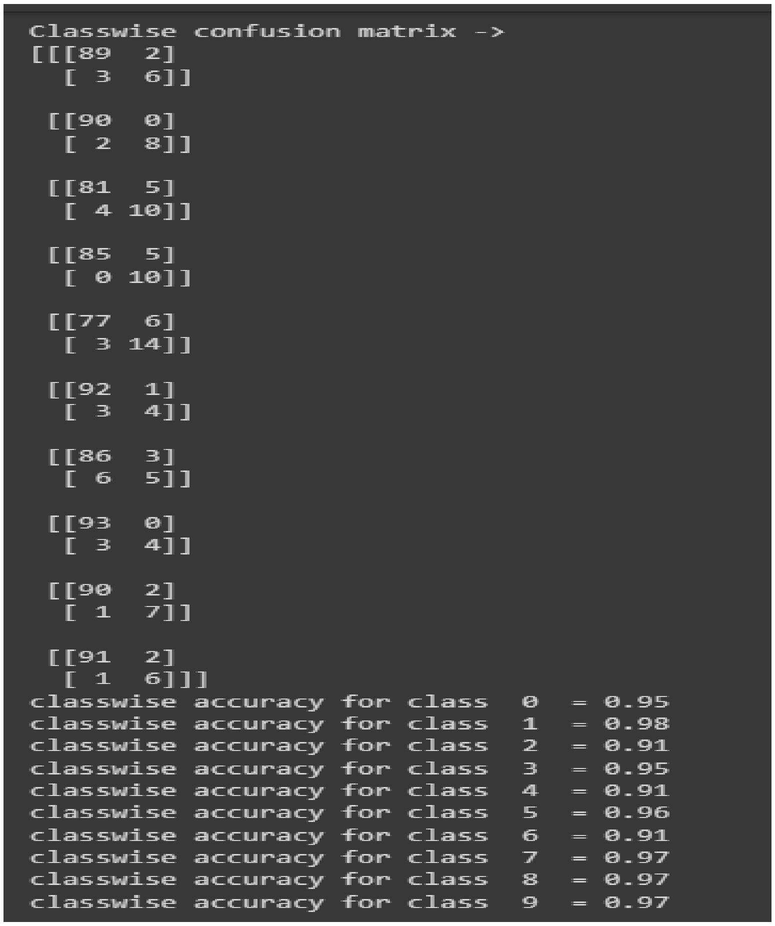
Plot Link

2) Test set accuracy and confusion matrix



```
Test accuracy for VGG16 model= 0.74
Confustion matrix -->
[[ 6  0  1  1  0  0  1  0  0  0]
 [ 0  8  0  1  1  0  0  0  0  0]
 [ 1  0 10  1  1  0  1  0  0  0]
 [ 0  0  0 10  0  0  0  0  0  0]
 [ 0  0  2  0 14  0  1  0  0  0]
 [ 0  0  0  1  0  4  0  0  1  1]
 [ 0  0  2  1  3  0  5  0  0  0]
 [ 0  0  0  0  0  1  0  4  1  1]
 [ 0  0  0  0  1  0  0  0  7  0]
 [ 1  0  0  0  0  0  0  0  0  6]]
```

3) Test set **class-wise** accuracies and confusion matrices



```
Classwise confusion matrix ->
[[[89  2]
  [ 3  6]]

 [[90  0]
  [ 2  8]]

 [[81  5]
  [ 4 10]]

 [[85  5]
  [ 0 10]]

 [[77  6]
  [ 3 14]]

 [[92  1]
  [ 3  4]]

 [[86  3]
  [ 6  5]]

 [[93  0]
  [ 3  4]]

 [[90  2]
  [ 1  7]]

 [[91  2]
  [ 1  6]]]
classwise accuracy for class  0  = 0.95
classwise accuracy for class  1  = 0.98
classwise accuracy for class  2  = 0.91
classwise accuracy for class  3  = 0.95
classwise accuracy for class  4  = 0.91
classwise accuracy for class  5  = 0.96
classwise accuracy for class  6  = 0.91
classwise accuracy for class  7  = 0.97
classwise accuracy for class  8  = 0.97
classwise accuracy for class  9  = 0.97
```