

REPORT - ASSIGNMENT 3

Q1

```
Data loaded !
(50000, 784, 2)
('Epoch :', 0, 'Accuracy : ', 91.86)
('Epoch :', 1, 'Accuracy : ', 93.93)
('Epoch :', 2, 'Accuracy : ', 95.05)
('Epoch :', 3, 'Accuracy : ', 95.8)
('Epoch :', 4, 'Accuracy : ', 96.00999999999999)
('Epoch :', 5, 'Accuracy : ', 96.2)
('Epoch :', 6, 'Accuracy : ', 96.22)
('Epoch :', 7, 'Accuracy : ', 96.56)
('Epoch :', 8, 'Accuracy : ', 96.6)
('Epoch :', 9, 'Accuracy : ', 96.56)
```

Best Accuracy for 'Sigmoid' function on The original MNIST Dataset: 96.56%

For epoc = 3 - Accuracy -> 92.11%
For epoc = 5 - Accuracy -> 93.78%
For epoc = 7 - Accuracy -> 96.32%
For epoc = 10 - Accuracy -> 96.56%

```
( 'Epoch :', 7, 'Accuracy : ', 99.92)
('Epoch :', 8, 'Accuracy : ', 99.76)
('Epoch :', 9, 'Accuracy : ', 99.7)
```

Accuracy for ReLU function on MNIST dataset: 99.7%

For epoc = 3 - Accuracy -> 98%
For epoc = 5 - Accuracy -> 99.11%
For epoc = 7 - Accuracy -> 99.32%
For epoc = 10 - Accuracy -> 99.7%

Subset MNIST data : 14251*28*28

b) Here I normalized the code by dividing it by 255 since the values were from 0-255.

Q2:

Using the following parameters:

```
mlp = MLPClassifier(activation='logistic',hidden_layer_sizes=(100,50), max_iter=100, alpha=0.0001,
                    solver='sgd', verbose=10, tol=1e-4, random_state=1,
                    learning_rate_init=0.1)
```

I got accuracy of 97.55%

```
Iteration 1, loss = 0.31675383
Iteration 2, loss = 0.12299209
Iteration 3, loss = 0.08981341
Iteration 4, loss = 0.06585255
Iteration 5, loss = 0.05440800
Iteration 6, loss = 0.04424590
Iteration 7, loss = 0.03760400
Iteration 8, loss = 0.02945605
Iteration 9, loss = 0.02682937
Iteration 10, loss = 0.01850703
Iteration 11, loss = 0.01635090
Iteration 12, loss = 0.01357203
Iteration 13, loss = 0.01431243
Iteration 14, loss = 0.01011230
Iteration 15, loss = 0.00669144
Iteration 16, loss = 0.00831085
Iteration 17, loss = 0.01159340
Iteration 18, loss = 0.00958737
Training loss did not improve more than tol=0
('Training set score: ', 0.99748000000000003)
('Test set score: ', 0.97550000000000003)
```

On the small MNIST dataset, accuracy came out to be 95.45%

Also there's no need to implement softmax since it's there by Default.

Q3:

Parameters :

```
mlp = MLPClassifier(activation='tanh',hidden_layer_sizes=(500,300), max_iter=100, alpha=1e-4,
                    solver='sgd', verbose=10, tol=1e-4,
                    learning_rate_init=0.1)
```

The following parameters were changed and experimented with:

- Activation
- hidden layers
- Solver
- K fold

Best accuracy came out to be at

- Activation = 'ReLU'

- hidden layers - (500,300)
- solver = 'sgd'
- k=6

Accuracy - 98.46%

```
Iteration 17, loss = 0.00066561
Training loss did not improve more than
('Training set score: ', 1.0)
('Test score: ', 0.98460000000000003)
```

EXP 1: activation - 'Sigmoid function', hidden layers - (100,50), solver = sgd

```
Iteration 18, loss = 0.00958737
Training loss did not improve more than tol=0
('Training set score: ', 0.99748000000000003)
('Test set score: ', 0.97550000000000003)
```

Accuracy - 97.55%

I also tried with Activation as Relu, and the hidden layers as 100,50 and the accuracy came out to be **97.8%**

```
Iteration 57, loss = 0.00386358
Iteration 58, loss = 0.00374585
Iteration 59, loss = 0.00362746
Iteration 60, loss = 0.00354737
Iteration 61, loss = 0.00342520
Iteration 62, loss = 0.00332993
Iteration 63, loss = 0.00324819
Iteration 64, loss = 0.00317867
Training loss did not improve more than to:
('Training set score: ', 1.0)
('Test set score: ', 0.97840000000000005)
```

Then Trying 'tanh' with number of hidden layers as 500,300 . The accuracy came out to be **98.24%**

```
Iteration 23, loss = 0.00130902
Iteration 24, loss = 0.00131180
Training loss did not improve more than t
('Training set score: ', 1.0)
('Test score: ', 0.98240000000000005)
```

Sigmoid's accuracy won't scale up much since it's output is limited to <1 . And reLU introduces non-Linearity due