Report

Question 1:

Assumption:

1. As it is a classification problem, for initial predicted value used log(odds) and then converting it to a probabiltity
2. Used One-vs-rest approach

Steps

1. Load the dataset
2. Hot encoding the target
3. Implement the gradient boosting classifier
    1. Calculate the residuals
    2. Fit a regression tree
    3. Predict the model
    3. Update Fm with learning rate
4. Calculate the accuracy

```
Final testing accuracy : 0.2317
```

Question 2:

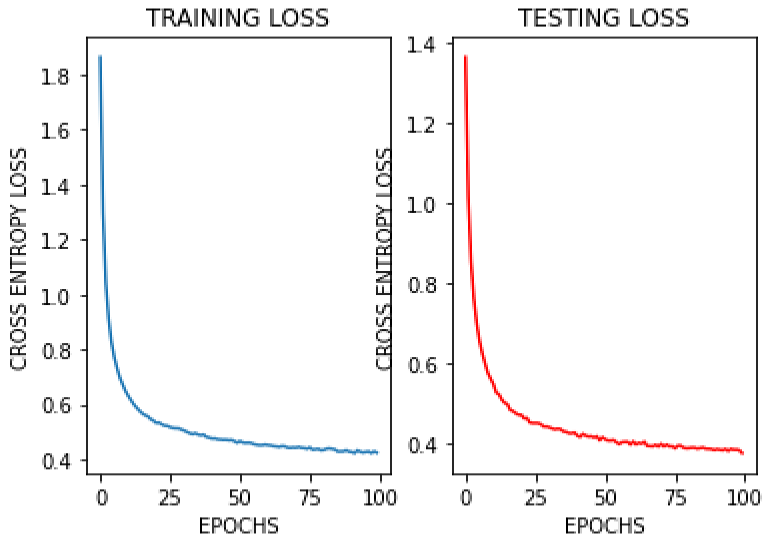
Assumption: Used TensorFlow keras Sequential model
Hyperparameters : Epochs = 100, Batch size = 128 , Learning rate = 0.01

Input layer : (input = 784, output = 256,activation = sigmoid)
Hidden layer : (input = 256, output = 128, activation = sigmoid)
Output layer : (input = 128, output = 10,activation = softmax)

Steps

1. Load the dataset
2. Apply oneHotEncoding
3. Define the 784-256-128-10 architecture
4. Compile the model
5. Plot epoch wise training loss

6. Calculate the accuracy



```
Testing Accuracy :  0.8656
```

Class-wise accuracy
Label 0 - 0.847
Label 1 - 0.97
Label 2 - 0.801
Label 3 - 0.89
Label 4 - 0.822
Label 5 - 0.92
Label 6 - 0.553
Label 7 - 0.926
Label 8 - 0.968
Label 9 - 0.959

Question 3:

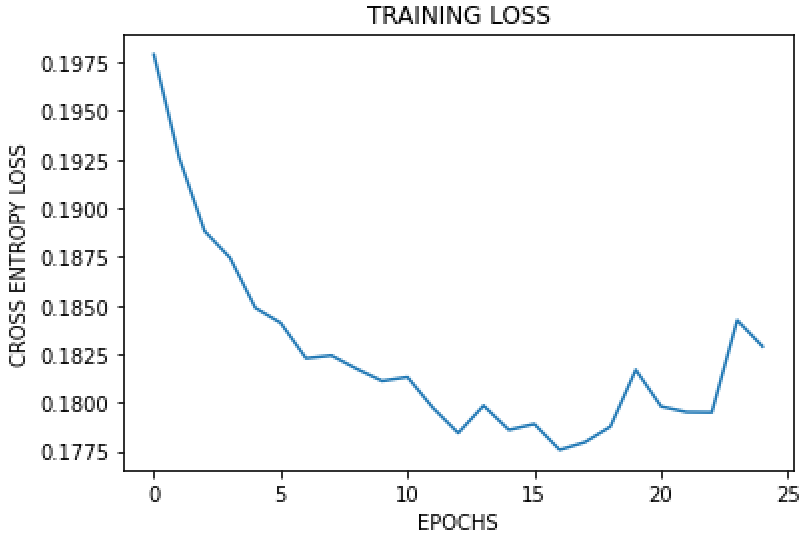
Assumption: Used TensorFlow keras Sequential model
For autoenocder:
The loss function used is Binary Cross Entropy as it is a classification problem and we have to compare each of the predicted probabilities of the hot encoded data to actual class output which can be either 0 or 1.
Hyperparameters : Epochs = 25, Batch size = 16, Learning rate = 0.0001

For MNIST Classifcation Model:

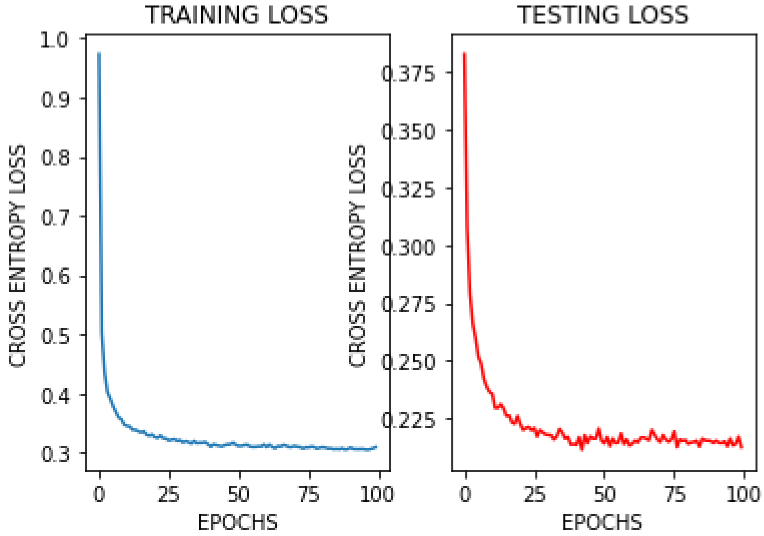
Hyperparameters : Epochs = 100, Batch size = 50, Learning rate = 0.001

Steps:

1. Load the dataset
2. Apply oneHotEncoding
3. Define the autoencoder architecture
4. Compile the autoencoder
5. Plot epoch wise training loss of autoencoder



6. Define the MNIST Classifcation Model
7. Plot epoch wise training loss



8. Calculate the accuracy

```
Accuracy :  0.9338
```

Class-wise accuracy
Label 0 - 0.9714285714285714
Label 1 - 0.9859030837004406
Label 2 - 0.9282945736434108
Label 3 - 0.9178217821782179
Label 4 - 0.9338085539714868
Label 5 - 0.9024663677130045
Label 6 - 0.9561586638830898
Label 7 - 0.9260700389105059
Label 8 - 0.9147843942505134
Label 9 - 0.8929633300297324

Question 4:

Steps:

1. Load the dataset
2. Implement the Bagging model and Majority voting techniques
3. Calculate the accuracy

```
Accuracy :  0.9066
```

Class-wise accuracy
Label 0 - 0.9816326530612245
Label 1 - 0.9814977973568282
Label 2 - 0.9147286821705426
Label 3 - 0.8871287128712871
Label 4 - 0.9134419551934827
Label 5 - 0.8598654708520179
Label 6 - 0.9144050104384134
Label 7 - 0.9221789883268483
Label 8 - 0.8151950718685832
Label 9 - 0.8602576808721506