SML Report - Assignment 4

Q1.
Step 1 - read data from MNIST dataset.
Step 2

- For **5 iterations**, plot decision tree regressor models on the data.
- Reduce the sample space by the **learning rate (initialized to 0.1)** times the predicted value of the regressor on the training data.

Boosting algorithm for regression trees

1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set.
2. For $b = 1, 2, \ldots, B$, repeat:
   2.1 Fit a tree $\hat{f}^b$ with d splits ($d + 1$ terminal nodes) to the training data (X, r).
   2.2 Update $\hat{f}$ by adding in a shrunken version of the new tree:
   $$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x).$$
   2.3 Update the residuals,
   $$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i).$$
3. Output the boosted model,
$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x).$$

Step 3

- Calculate the value of the prediction as given in the pseudocode above. Here, B is M (number of iterations) and λ is the learning rate.
- Calculate for M = 1, 2, 3, 4, and 5 to get iteration-wise training and testing accuracy.

Q2.
Step 1 - Read data from FMNIST dataset.
Step 2

- Normalize the data from range (0,255) to (-0.5, 0.5).
- Create a neural network of 2 hidden layers of 64 nodes each and one output layer of 10 nodes.
- Initialize the kernel and biases to random values.
- Use SGD (Stochastic Gradient Descent) to learn the parameters using mini-batch sizes of 600.
- Train the neural network, for 50 epochs.
- Plot the epoch-wise training loss for the training data.

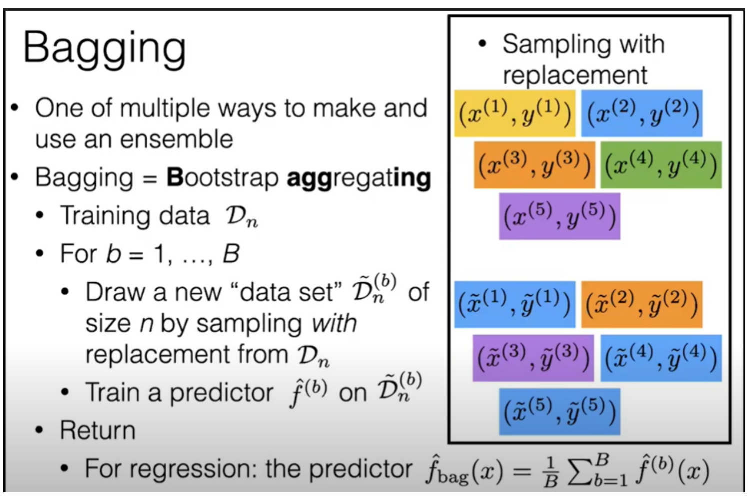
Step 3

- Predict the values of the classes using the testing data provided.
- Plot total and classwise accuracy.

Q3.
Step 1 - Read data from MNIST dataset.
Step 2
- Create an autoencoder with multiple layers as provided in the question.
- Train the data using input data.
- Minimize the training loss in the process.
- You now have the autoencoder required

Step 3
- Now drop the layers of the decoder and use the remaining layers of the autoencoder to find a good classifier for the data.
- Fix the trained layers, from the autoencoder so that their weights and biases do not change.
- Train the data and test.
- Plot the outputs

Q4.
Step 1 - Read data from MNIST dataset.
Step 2



- Implement the pseudocode using bag_size = 3
- Randomly choose values from the dataset such that the size of the bagged dataset remains the same.
- Train the different datasets using a decision tree classifier.

Step 3
- Test on the testing data
- Only choose the value for majority voting in case a clear majority prevails.