Assignment 4 Report

Q1. AdaBoost on MNIST (Digits 0 and 1)

Objective

To implement the AdaBoost algorithm from scratch using decision stumps and evaluate it on a binary classification task (digits 0 and 1 from MNIST). The input features were reduced to 5 dimensions using PCA that was done using previous assignment code.

Implementation Details

- Used 1000 training samples per class (2000 total).
- PCA was applied to reduce the feature space from 784 to 5.
- For each decision stump, 3 uniformly spaced thresholds were chosen between the min and max of each feature dimension.
- Sample weights were updated after each boosting round based on weighted misclassification error.
- Classifier weight alpha was computed as done in class
- Final predictions were made using a weighted majority vote of weak learners.

Results on sample case got at that time:

Data Size:

```
x_train: (2000, 784)

y_train: (2000,)

x_test: (2115, 784)

y_test: (2115,)
```

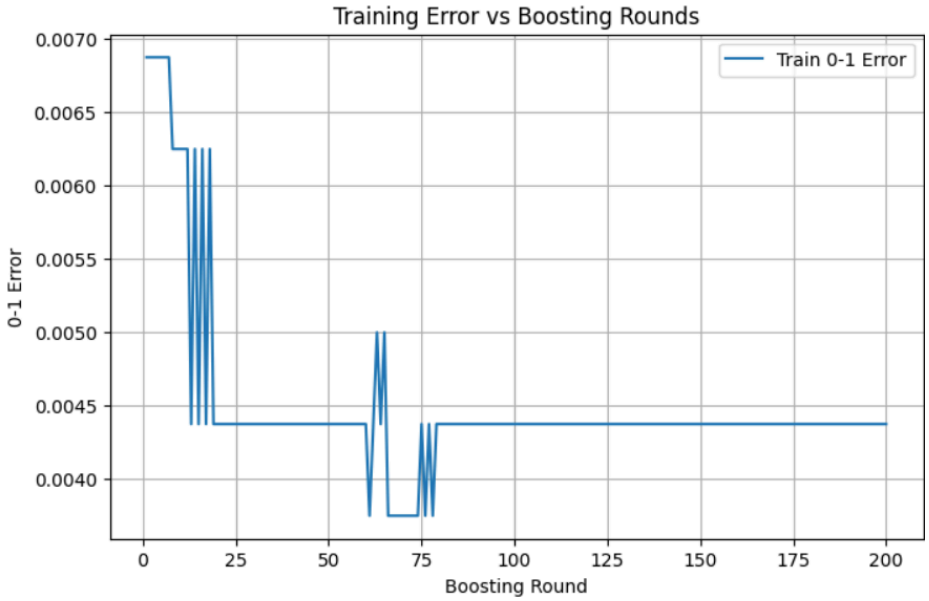
Some values:

```
Round 0, Error: 0.0069, Alpha: 2.4865
Round 10, Error: 0.3533, Alpha: 0.3024
Round 20, Error: 0.4282, Alpha: 0.1446
Round 30, Error: 0.4263, Alpha: 0.1485
Round 40, Error: 0.4451, Alpha: 0.1102
Round 50, Error: 0.4502, Alpha: 0.1000
Round 60, Error: 0.4595, Alpha: 0.0811
Round 70, Error: 0.4583, Alpha: 0.0836
Round 80, Error: 0.4716, Alpha: 0.0568
Round 90, Error: 0.4669, Alpha: 0.0664
Round 100, Error: 0.4696, Alpha: 0.0609
Round 110, Error: 0.4791, Alpha: 0.0419
Round 120, Error: 0.4735, Alpha: 0.0530
Round 130, Error: 0.4764, Alpha: 0.0473
Round 140, Error: 0.4827, Alpha: 0.0346
Round 150, Error: 0.4802, Alpha: 0.0396
Round 160, Error: 0.4818, Alpha: 0.0364
Round 170, Error: 0.4816, Alpha: 0.0367
Round 180, Error: 0.4820, Alpha: 0.0359
Round 190, Error: 0.4860, Alpha: 0.0279
```

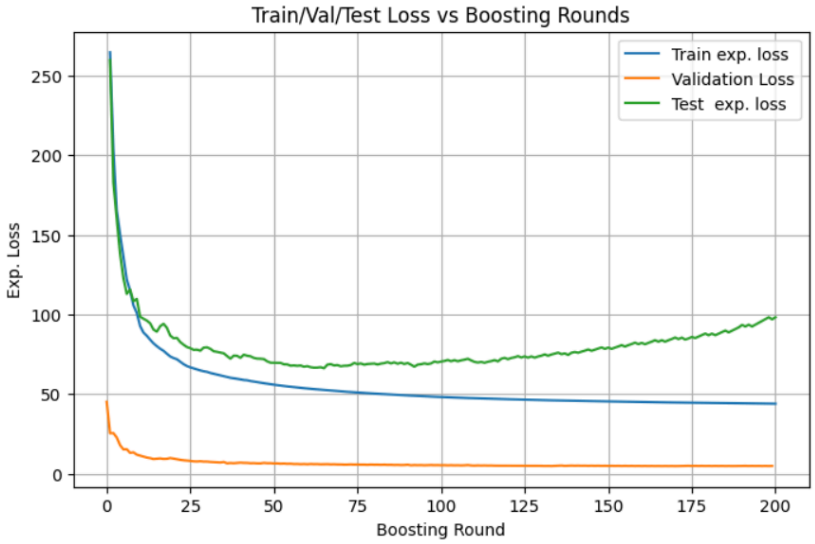
Test Accuracy: 99.57%

Plots

1. **Training Error vs Boosting Rounds**



2. **Train/Validation/Test Loss over Rounds**



Q2. Gradient Boosting for Regression

Implementation

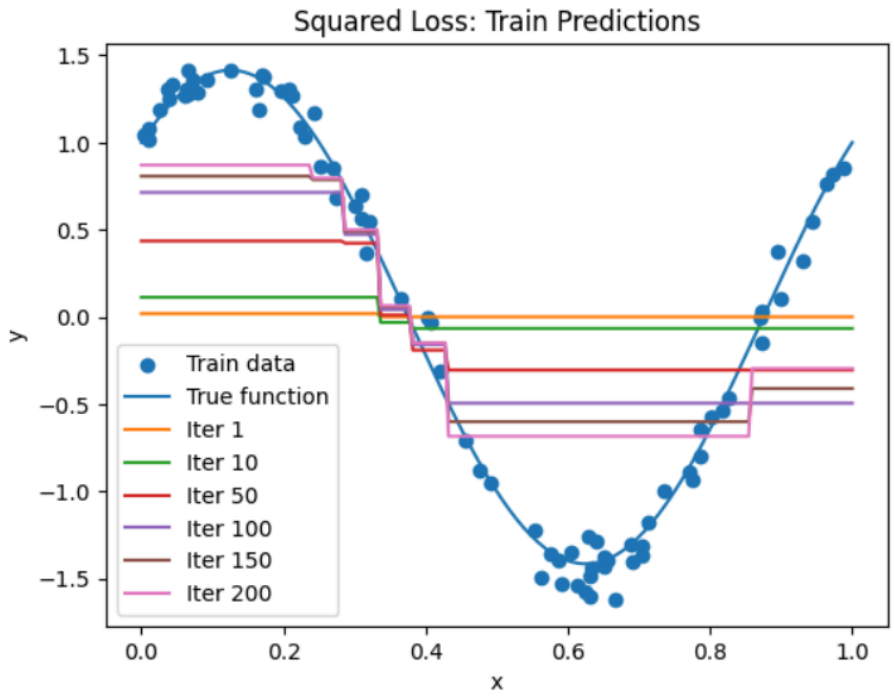
- Learning rate = 0.01
- 20 cuts uniformly spaced across [0, 1] for fitting decision stumps
- Iteratively fit weak learners on the negative gradient of the loss

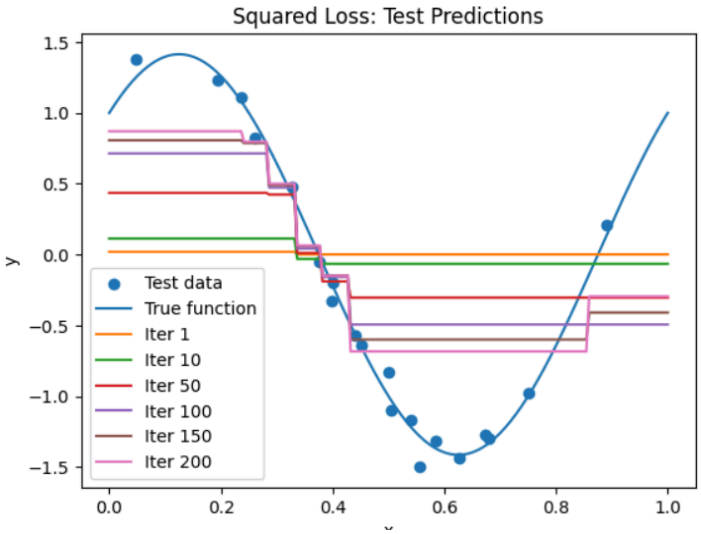
Below all are for a particular sample at that time

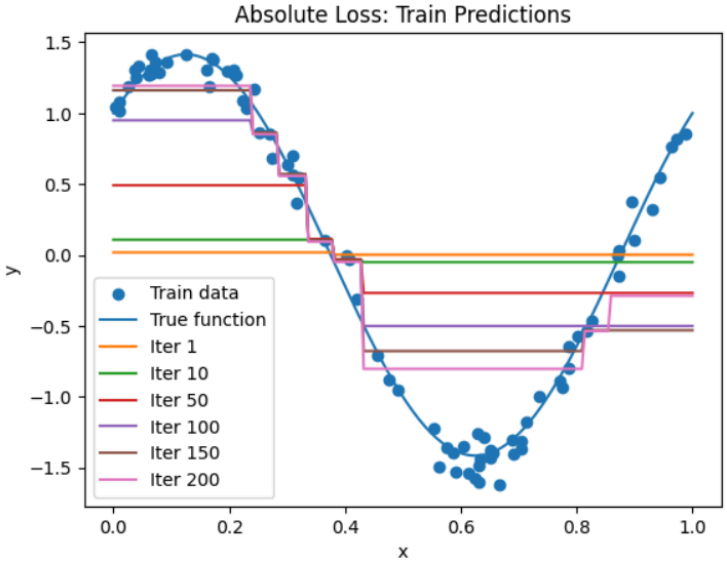
Results

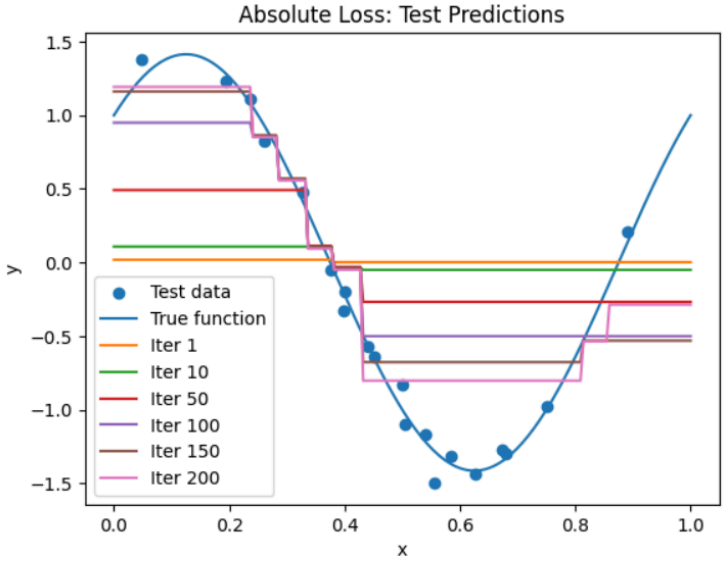
- Shown in plot

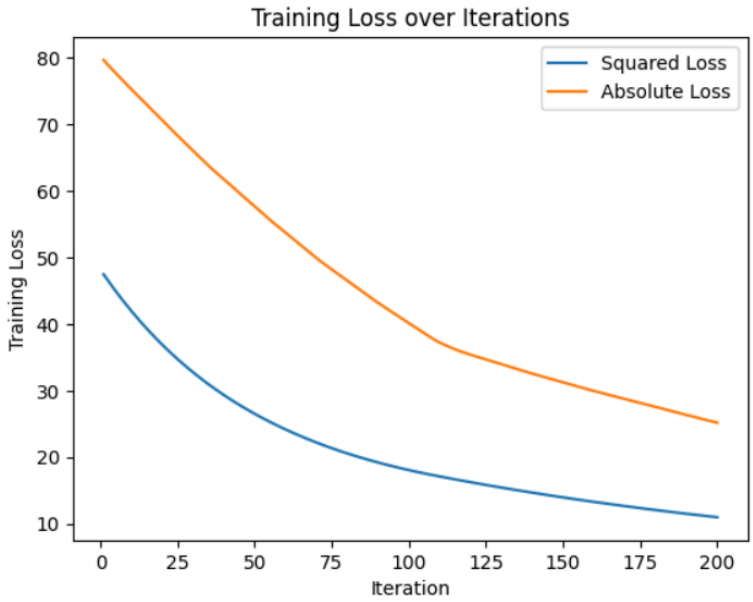
Plots




Squared Loss: Test Predictions
Test data
True function
Iter 1
Iter 10
Iter 50
Iter 100
Iter 150
Iter 200
y


Absolute Loss: Train Predictions
Train data
True function
Iter 1
Iter 10
Iter 50
Iter 100
Iter 150
Iter 200
y
x


Absolute Loss: Test Predictions
Test data
True function
Iter 1
Iter 10
Iter 50
Iter 100
Iter 150
Iter 200
y
x



Q3. Neural Network for Binary Classification

Implementation:

- As done in class first hidden layer before activation is a=(w1*x1 + w2*x2)
- z=sigmoid(a)
- Y_pred = w3*z

Results

For data:

```
X0=[[-1.40501115 -0.4416588 ]
 [-0.37298939  1.00832811]
 [-1.28854351  1.08060065]
 [-1.20878953  0.87098425]
 [-1.54035797 -1.04622186]
 [-0.04543345 -0.94785895]
 [-0.85628513  0.23234273]
 [ 1.48662165 -2.07147965]
 [-1.00917349  0.78995653]
 [-0.61831008 -2.25348316]]
y0=[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
X1=[[ 1.03178131  1.78244213]
```

```
 [-1.53455923  0.90080609]
 [ 0.66159655  2.93185008]
 [-0.32152172  0.07893366]
 [ 2.01383241  0.50394447]
 [ 1.90135571  0.77889113]
 [ 1.77349853  1.78346686]
 [ 0.32906053  1.25980153]
 [ 2.12666839 -0.11414793]
 [ 1.00060797  2.18238443]]
y1=[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]


x_train= [[-1.40501115 -0.4416588 ]
 [-0.37298939  1.00832811]
 [-1.28854351  1.08060065]
 [-1.20878953  0.87098425]
 [-1.54035797 -1.04622186]
 [ 1.03178131  1.78244213]
 [-1.53455923  0.90080609]
 [ 0.66159655  2.93185008]
 [-0.32152172  0.07893366]
 [ 2.01383241  0.50394447]]
y_train= [0. 0. 0. 0. 0. 1. 1. 1. 1. 1.]
x_test= [[-0.04543345 -0.94785895]
 [-0.85628513  0.23234273]
 [ 1.48662165 -2.07147965]
 [-1.00917349  0.78995653]
 [-0.61831008 -2.25348316]
 [ 1.90135571  0.77889113]
 [ 1.77349853  1.78346686]
 [ 0.32906053  1.25980153]
 [ 2.12666839 -0.11414793]
 [ 1.00060797  2.18238443]]
y_test= [0. 0. 0. 0. 0. 1. 1. 1. 1. 1.]


Initially:
w1 [1.32730033]
w2 [0.25824073]
w3 [0.38107874]
```

After iterations: **`Test MSE is  0.2`**