Q-1

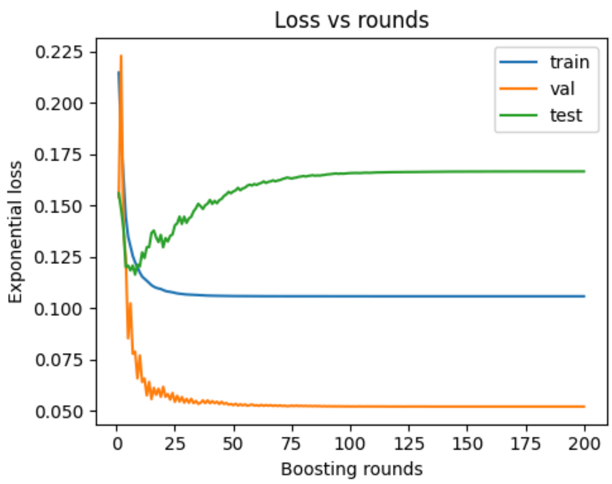
Terminal Results :

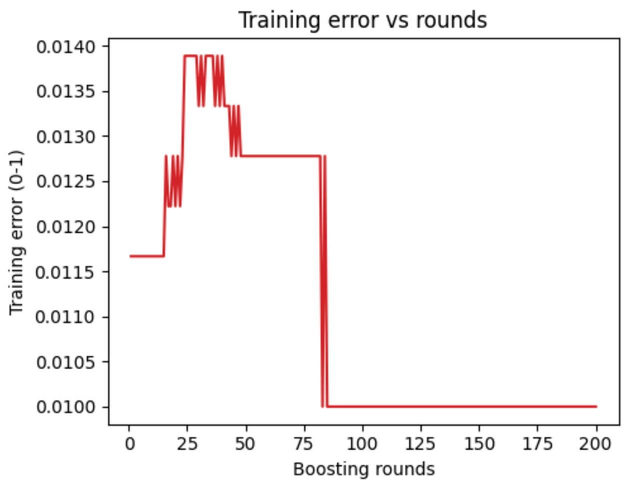
Round   1: eps=0.012, beta=2.220, train err=1.1666666666666674068%
Round 41: eps=0.488, beta=0.023, train err=1.3333333333333348136%
Round 81: eps=0.499, beta=0.003, train err=1.2777777777777790114%
Round 121: eps=0.500, beta=0.000, train err=1.0000000000000000000%
Round 161: eps=0.500, beta=0.000, train err=1.0000000000000000000%

Final test accuracy: 99.29%

The key metrics here are `eps` (the error rate of the current weak learner), `beta` (the weight assigned to the weak learner), and `train-err` (the training error after adding the weak learner).

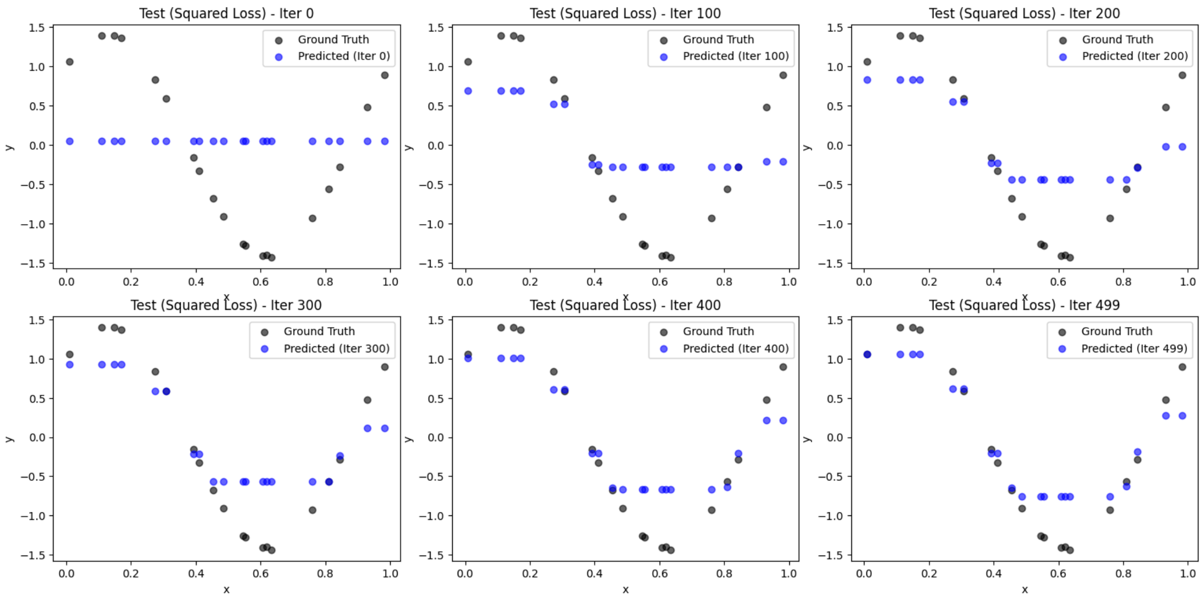
As boosting progresses, the error rate of individual weak learners increases, and their contribution (weight) to the final model decreases. However, the model still manages to fit the training data very well, with the training error dropping significantly, even though later learners are not adding much value. This suggests the model might be overfitting, as it becomes increasingly reliant on weak learners with almost no predictive power.



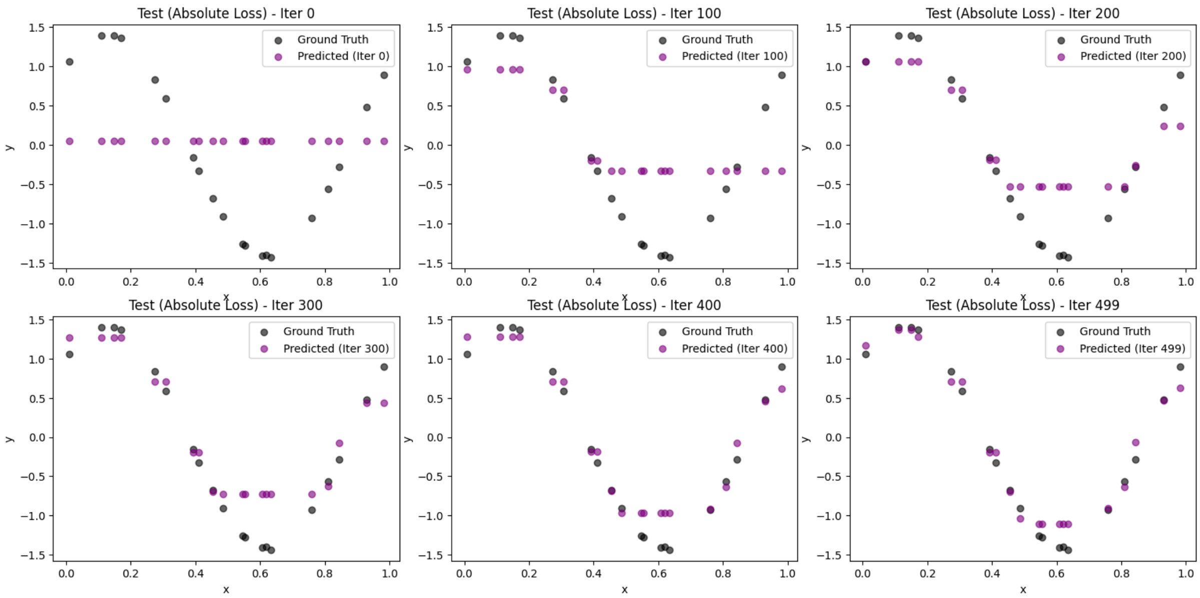


Q-2

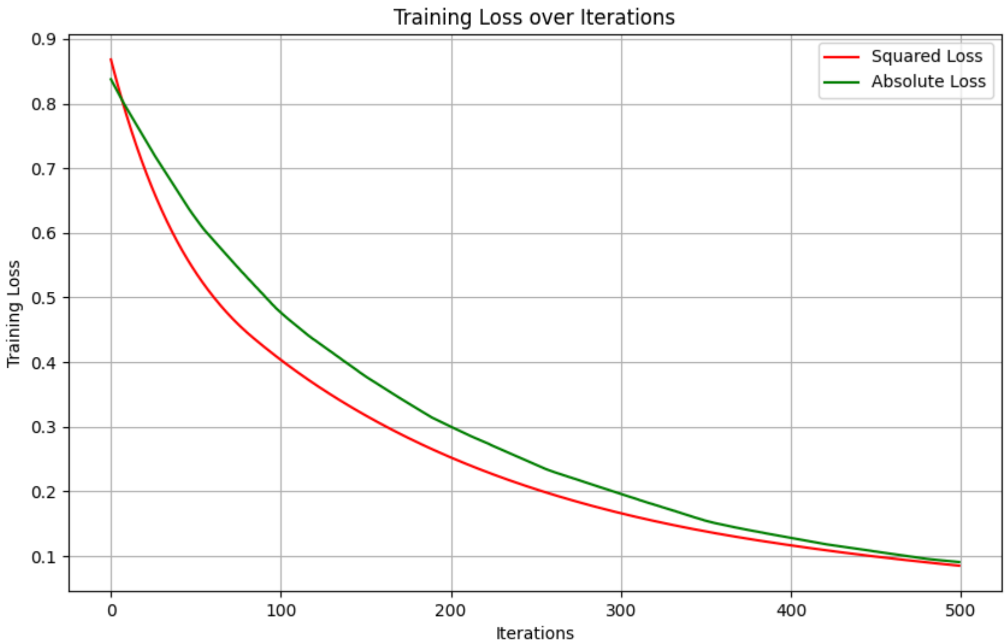
Test Error - Squared Loss



Test Error -Absolute Loss



Training LOSS over iterations :



Q-3

Terminal Output :

```
iteration 0: Training Loss = 0.9464
  w1 = [0.75562379 0.20731282], b1 = -0.1201
  w2 = [0.39548943], b2 = -0.1305

iteration 100: Training Loss = 0.0355
  w1 = [1.60454224 1.27348612], b1 = -0.7329
  w2 = [1.37791028], b2 = -0.0920

iteration 200: Training Loss = 0.0275
  w1 = [2.21525925 1.73738618], b1 = -1.1516
  w2 = [1.31642965], b2 = -0.0674

iteration 300: Training Loss = 0.0227
  w1 = [2.70055458 2.10332906], b1 = -1.4440
  w2 = [1.28243267], b2 = -0.0630

iteration 400: Training Loss = 0.0196
  w1 = [3.09058777 2.39772522], b1 = -1.6720
  w2 = [1.25809688], b2 = -0.0608

iteration 500: Training Loss = 0.0174
  w1 = [3.41744848 2.64325749], b1 = -1.8616
  w2 = [1.23940836], b2 = -0.0590

iteration 600: Training Loss = 0.0158
  w1 = [3.70055835 2.85424746], b1 = -2.0260
  w2 = [1.22454954], b2 = -0.0572
```

```
iteration 700: Training Loss = 0.0145
  w1 = [3.95182213 3.0397311 ], b1 = -2.1726
  w2 = [1.21242024], b2 = -0.0556

iteration 800: Training Loss = 0.0134
  w1 = [4.17890759 3.20566547], b1 = -2.3059
  w2 = [1.20230271], b2 = -0.0541

iteration 900: Training Loss = 0.0126
  w1 = [4.3869942  3.35615733], b1 = -2.4288
  w2 = [1.19370989], b2 = -0.0527

iteration 999: Training Loss = 0.0118

MSE over Test Samples : 0.0542
```

Interpretation :
The training log shows a healthy, steady convergence of the one-hidden-unit neural network on this very small, linearly separable toy problem. Starting from a near-random state (training MSE ≈ 0.95), gradient descent rapidly lowers the loss to ≈ 0.04 within the first 100 iterations and continues to fine-tune it to 0.012 by iteration 999. During this process the hidden-layer weights grow in magnitude while the bias becomes increasingly negative, indicating that the sigmoid is being sharpened and shifted so its transition zone lies between the two class centres at (-1,-1) and (1, 1). The output weight stabilises around 1.2 and its bias drifts toward zero, meaning that once the hidden unit has learned a discriminative feature, the output neuron mainly rescales it. The final test MSE of 0.054—very close to the training loss—confirms good generalisation and shows that a single-neuron hidden layer is sufficient to cleanly separate these two Gaussian clusters.