CS671: Deep Learning and Applications
Programming Assignment I
Group 16

Anmol Bishnoi (B19069)
Rachita (B19258)
Anshul (B19150)

Classification Tasks

Perceptron

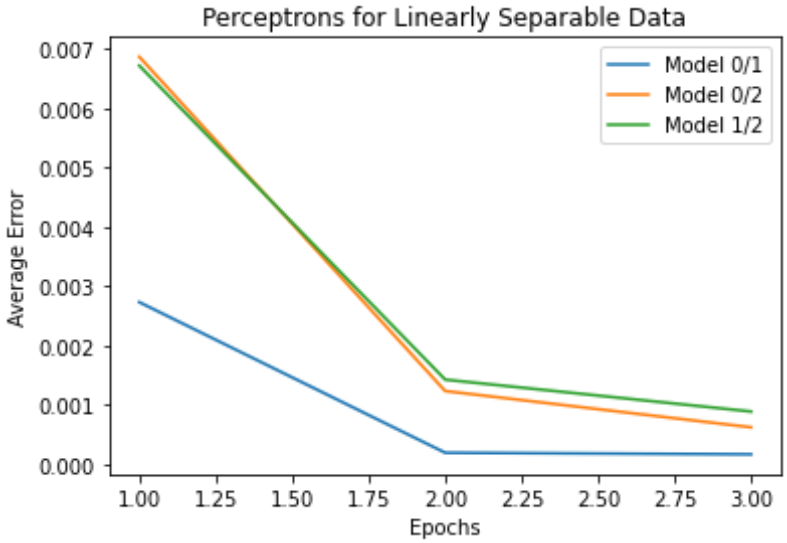
We wrote code for a sigmoid perceptron from scratch and used it to train on both the Linearly Separable and Non-Linearly Separable Data.
The parameters for both models were set as:

- Eta (Learning Rate Parameter) = 0.4
- Max Epochs = 10
- Error Threshold for successive epochs = 1 x 10^{-3}

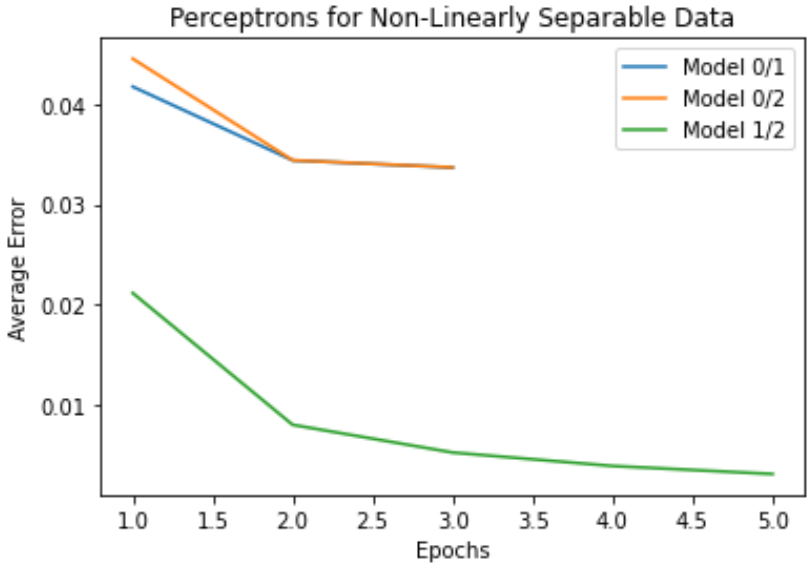
Train Test Split was done in the ratio 60:40 since the 20% from validation was also supposed to be considered as additional test data.

Both Linearly Separable Data and Non-Linearly Separable Data had three classes each.

Q1 Plot of average error vs epochs

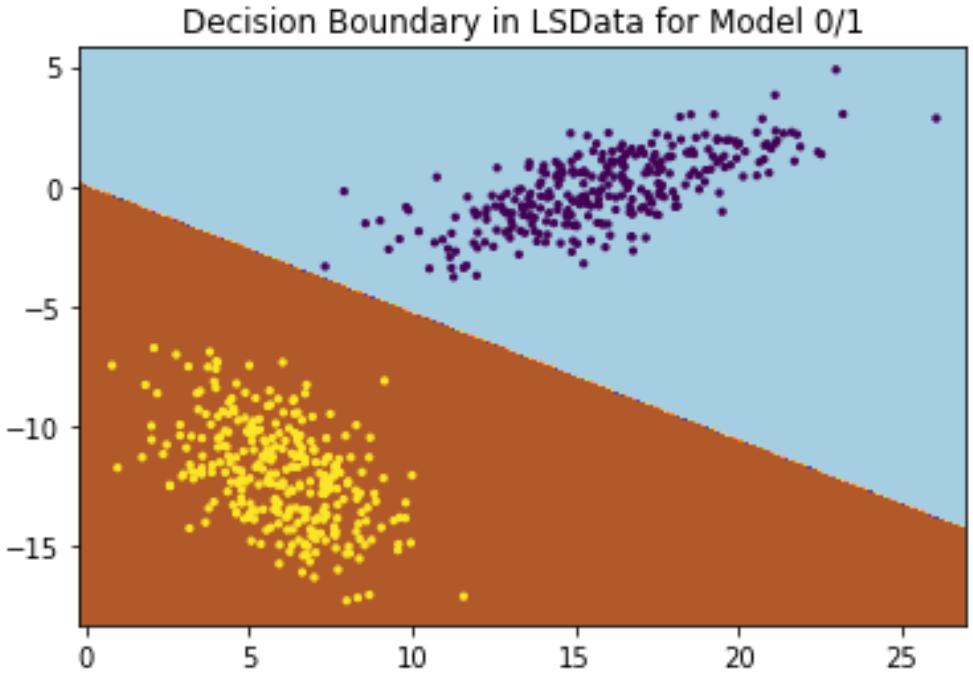


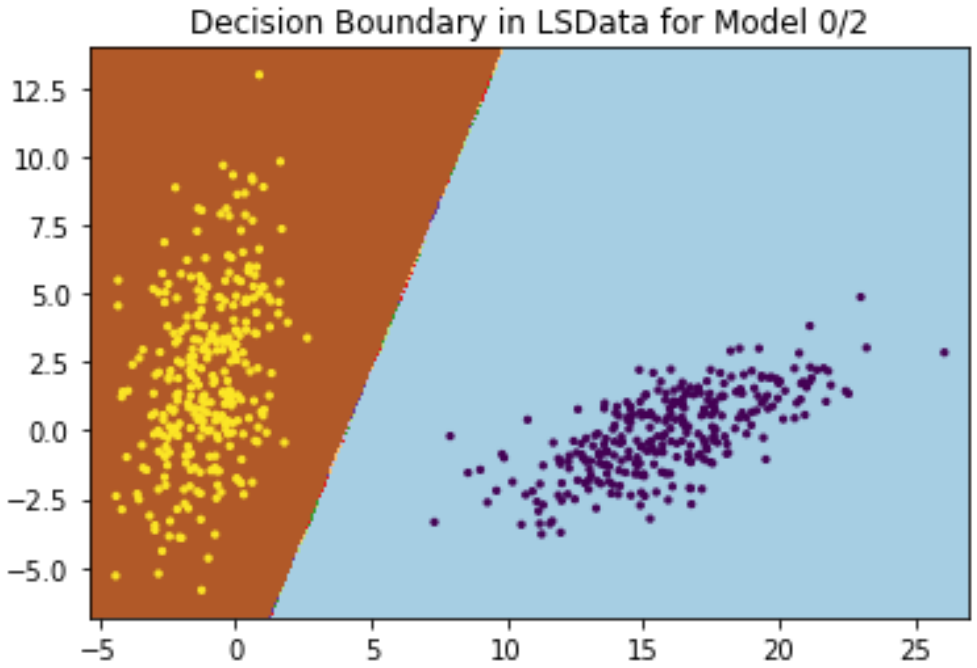
Since the one-vs-one approach was used, three different perceptrons were created to separate between classes 0/1, 1/2 & 2/0. All three perceptrons converged within 3 epochs each.

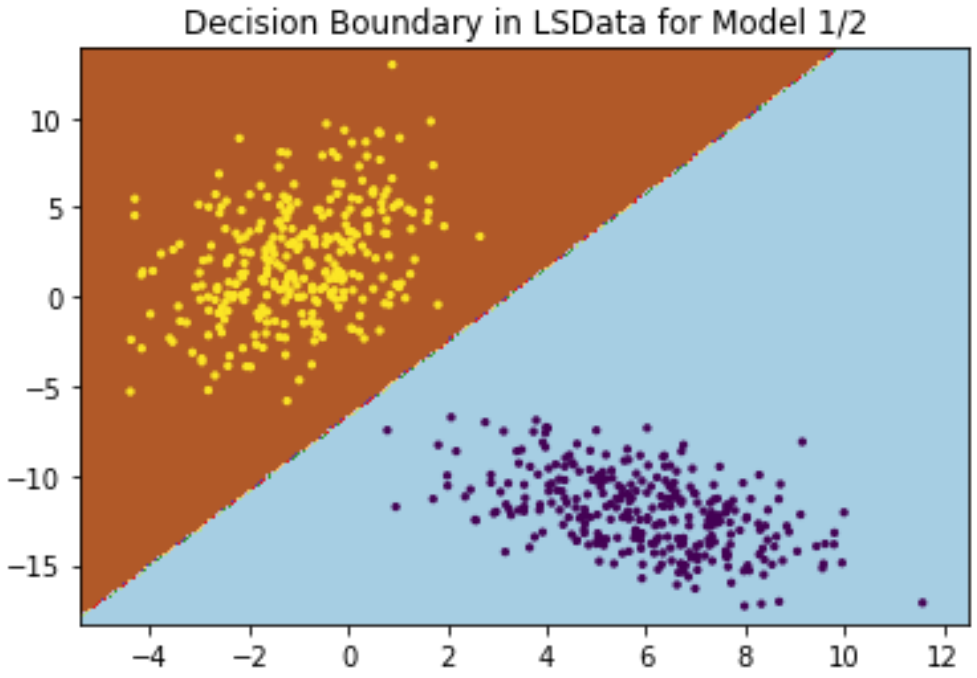


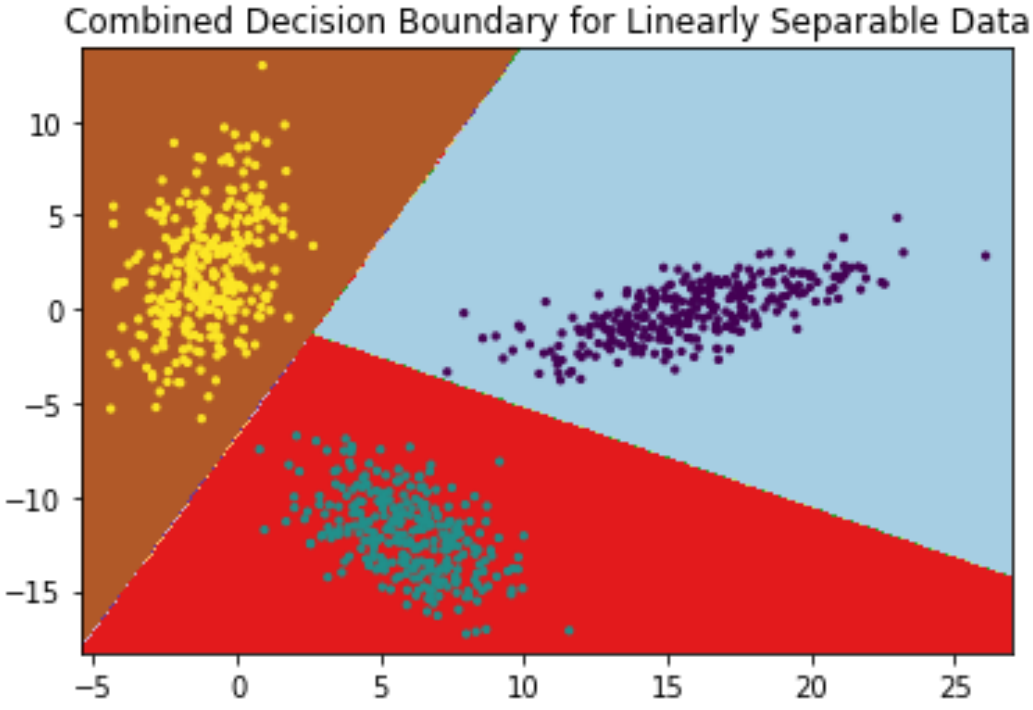
Once again three different perceptrons had to be created to classify among three classes but here the first two perceptrons i.e. 0/1 and 0/2 converged within 3 epochs while the last perceptron took 5 full epochs to converge.
Additionally, final average errors after model convergence for the first two models were also relatively higher than the third model.

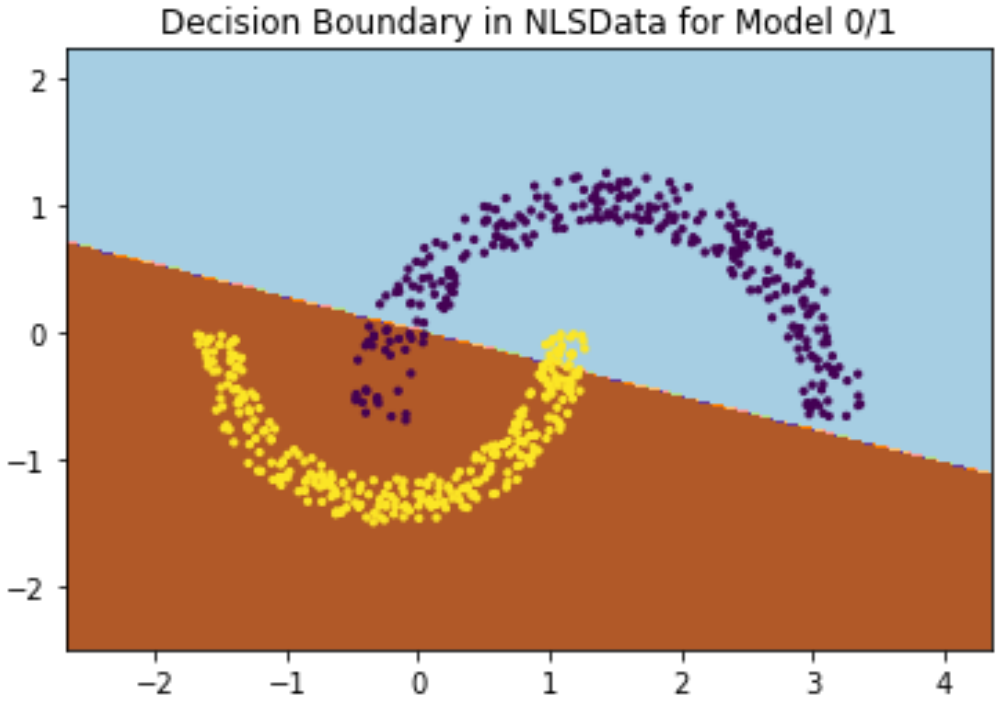
Q2 Decision Region Plot superimposed by training data

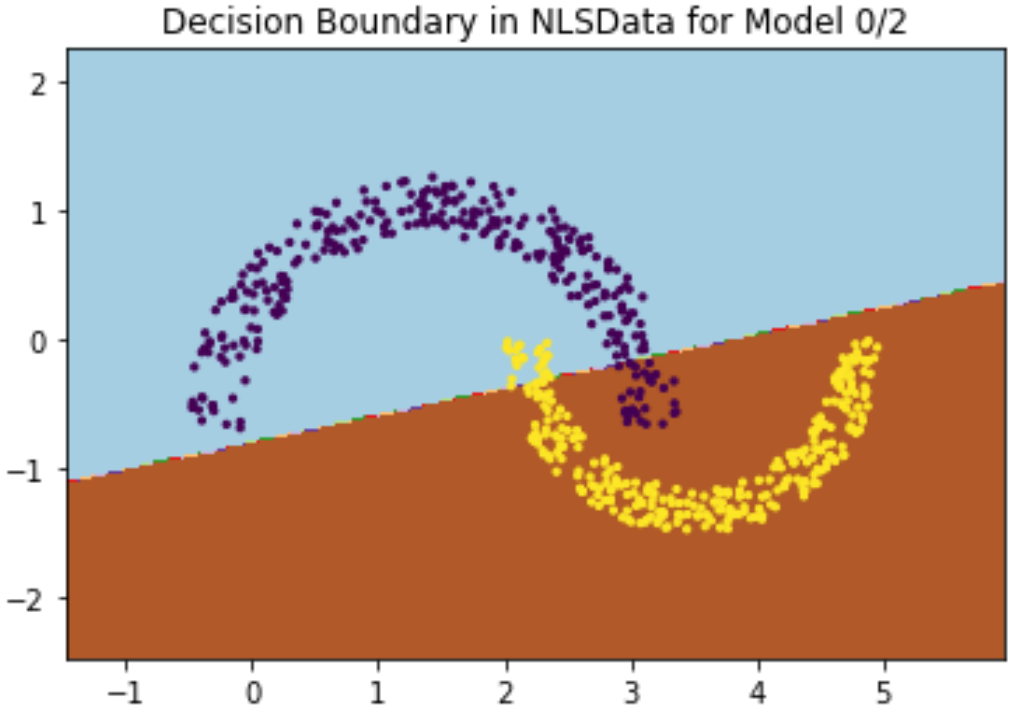


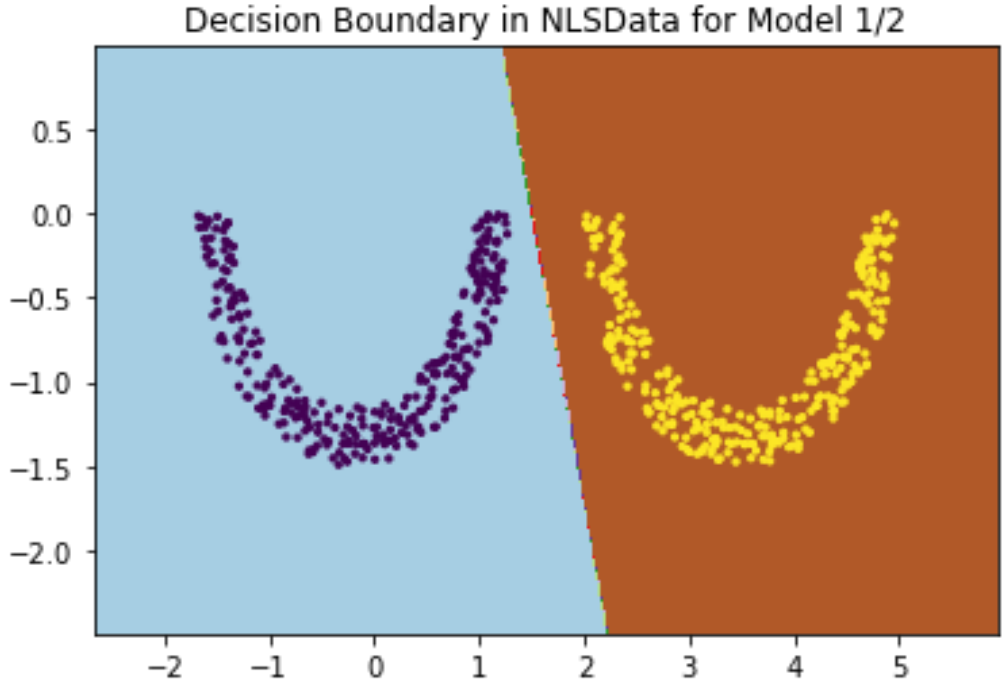

Decision Boundary in LSData for Model 0/2
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−5
0
5
10
15
20
25

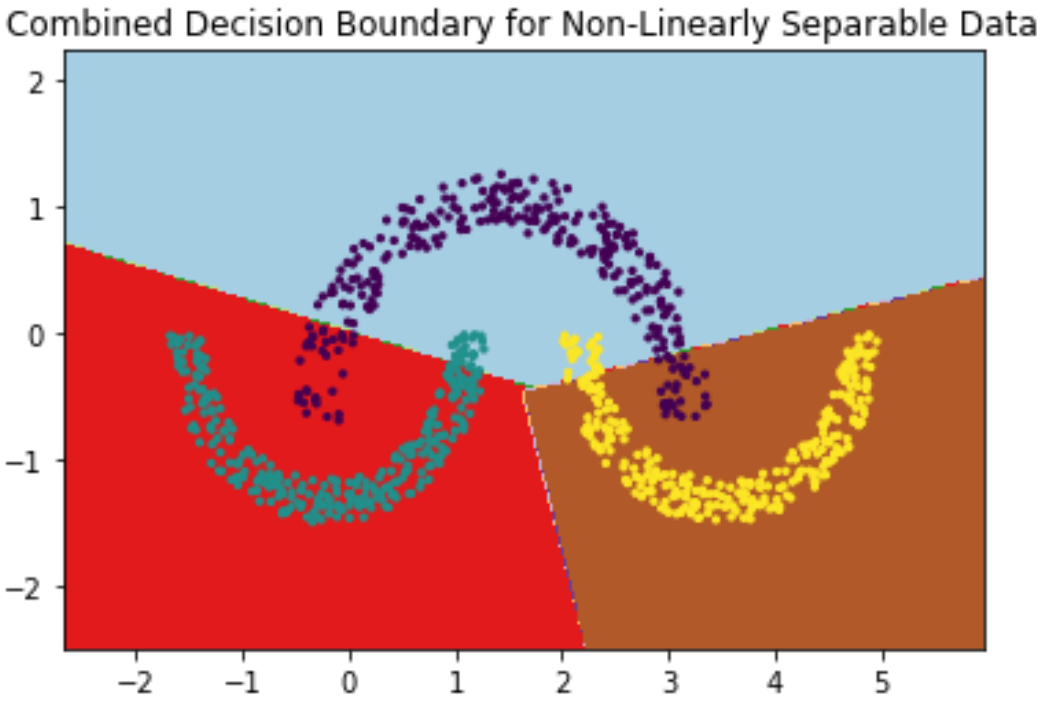

Decision Boundary in LSData for Model 1/2
10
5
0
−5
−10
−15
−4
−2
0
2
4
6
8
10
12


Combined Decision Boundary for Linearly Separable Data
10
5
0
−5
−10
−15
−5
0
5
10
15
20
25


Decision Boundary in NLSData for Model 0/1
2
1
0
−1
−2
−2
−1
0
1
2
3
4


Decision Boundary in NLSData for Model 0/2
2
1
0
−1
−2
−1
0
1
2
3
4
5


Decision Boundary in NLSData for Model 1/2
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2
−1
0
1
2
3
4
5

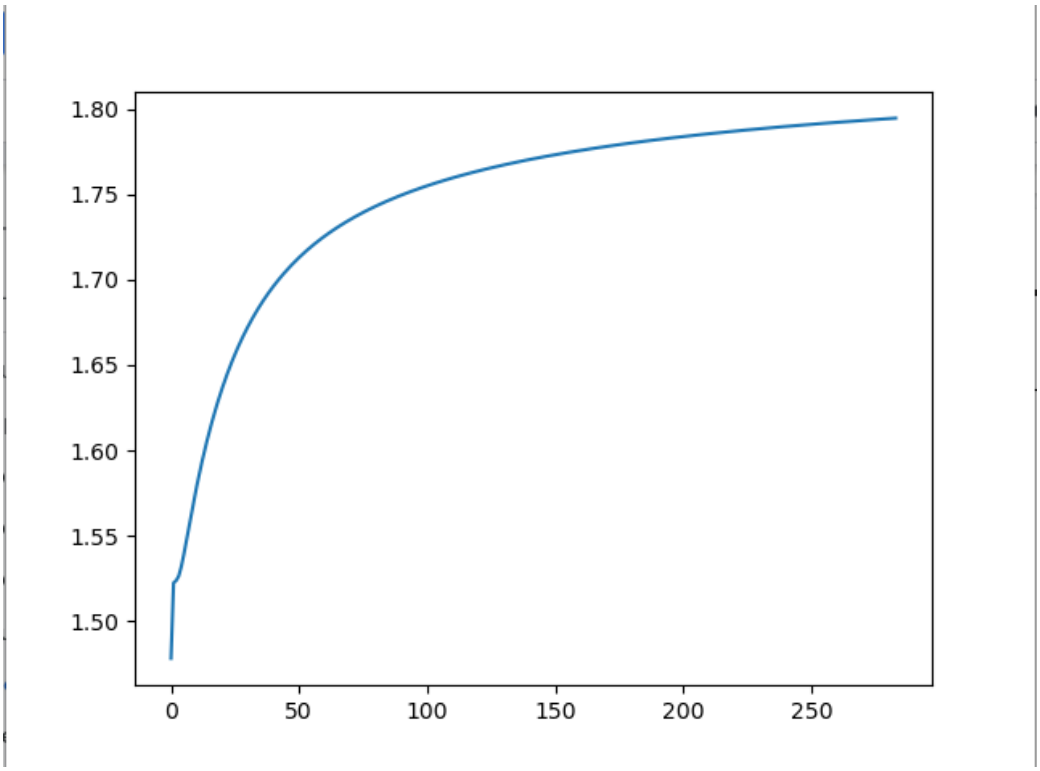


Q3 Confusion Matrices and Accuracy Scores

```
For Linearly Separable Data

Confusion Matrix:
[[204   0   0]
 [  0 197   0]
 [  0   0 199]]
Accuracy Score = 1.0


For Non-Linearly Separable Data

Confusion Matrix:
[[155  26  23]
 [  9 188   0]
 [ 19   0 180]]
Accuracy Score = 0.8716666666666667
```

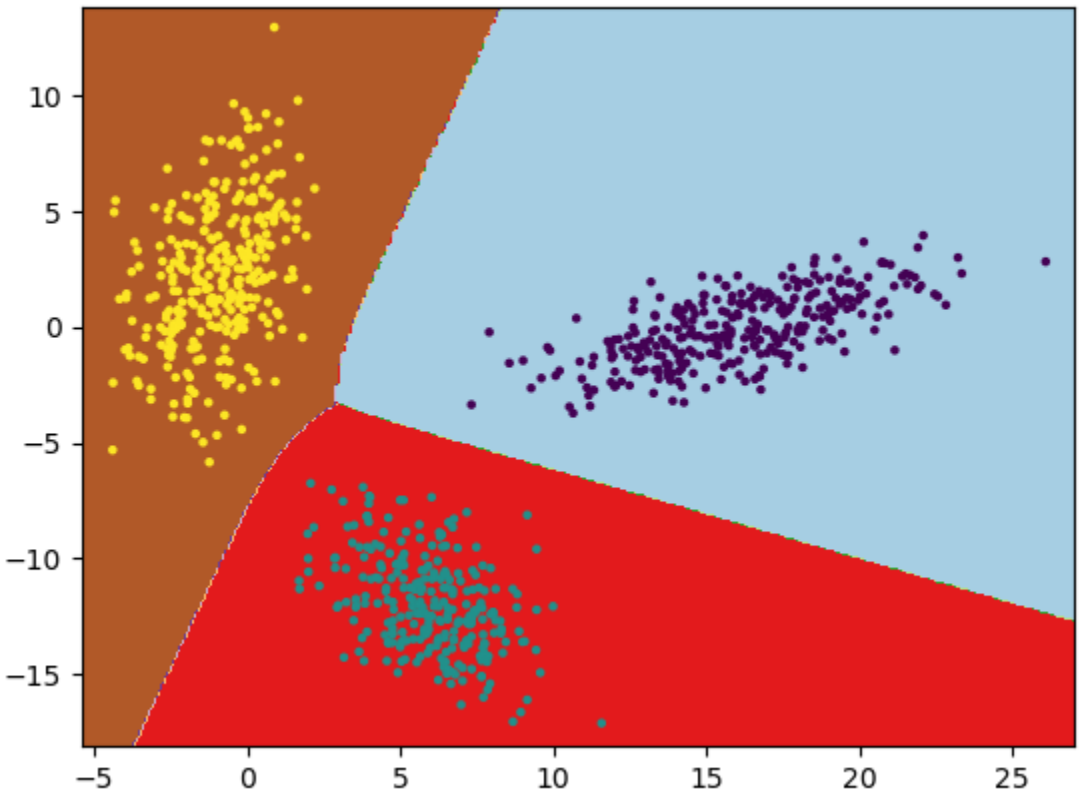
Fully Connected Neural Network

Que 1.

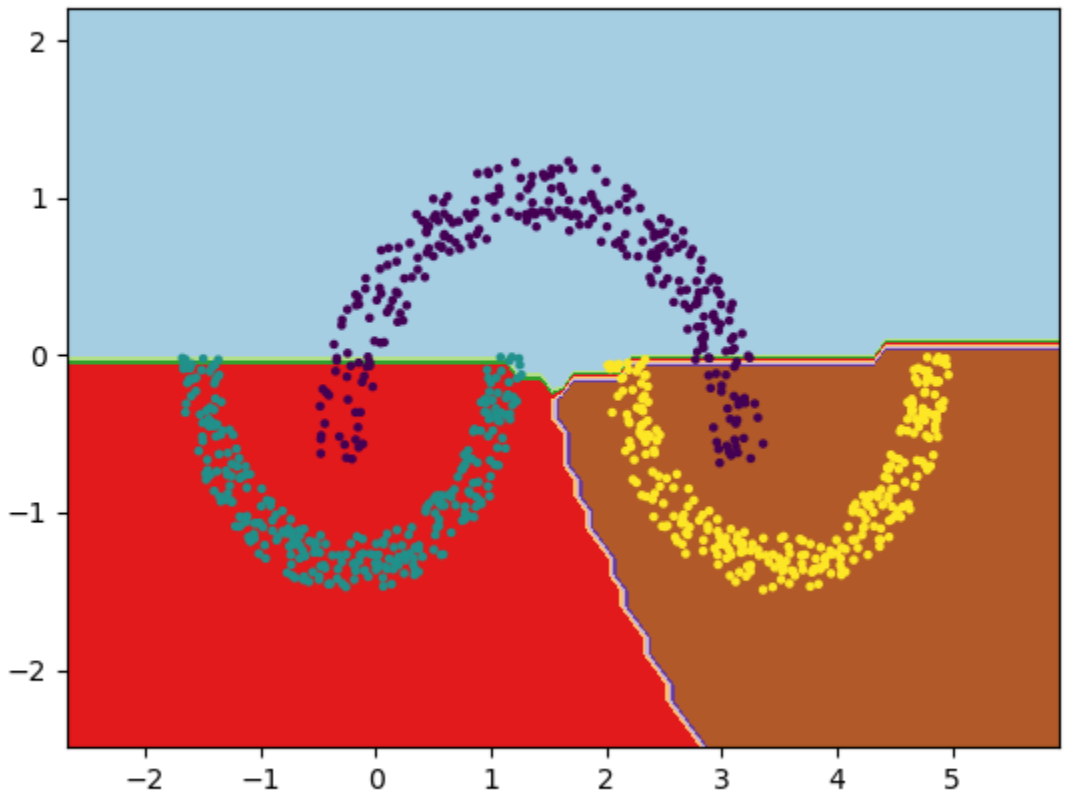


Que 2.

Decision boundary for LS data



Decision boundary for NLS data



Que 3.

Linearly Separable:

We have taken 3 nodes in the hidden layer.
When we had taken 1 node accuracy was 60%.
For nodes>=2 accuracy was 100%.

Confusion matrix:
[106 0 0]
[0 92 0]
[0 0 102]]
Accuracy = 100 %

Non-linearly Separable:

Confusion matrix :
[[73 13 14]

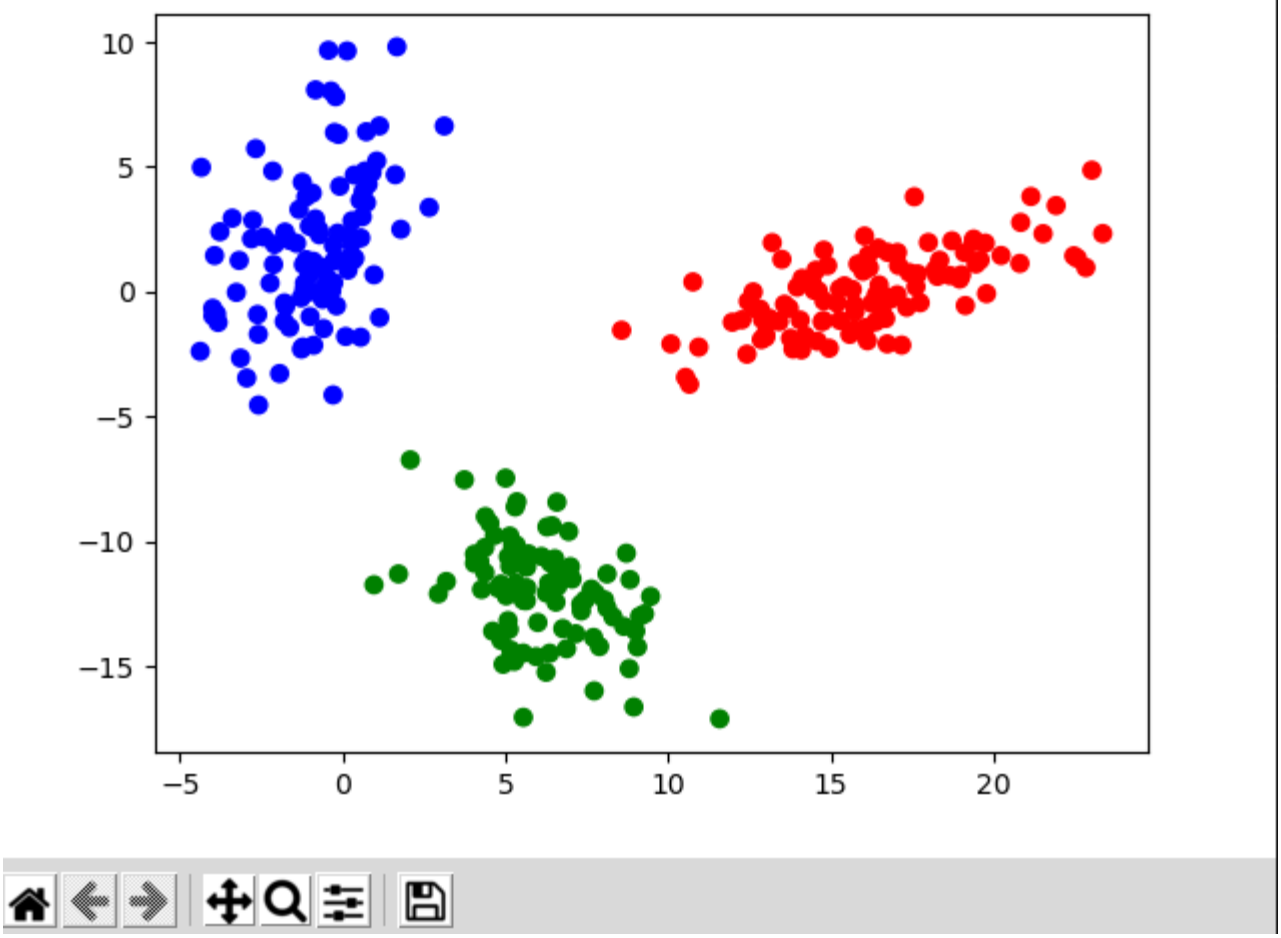
```
[  2  93   0]
[  4   0 101]]
```

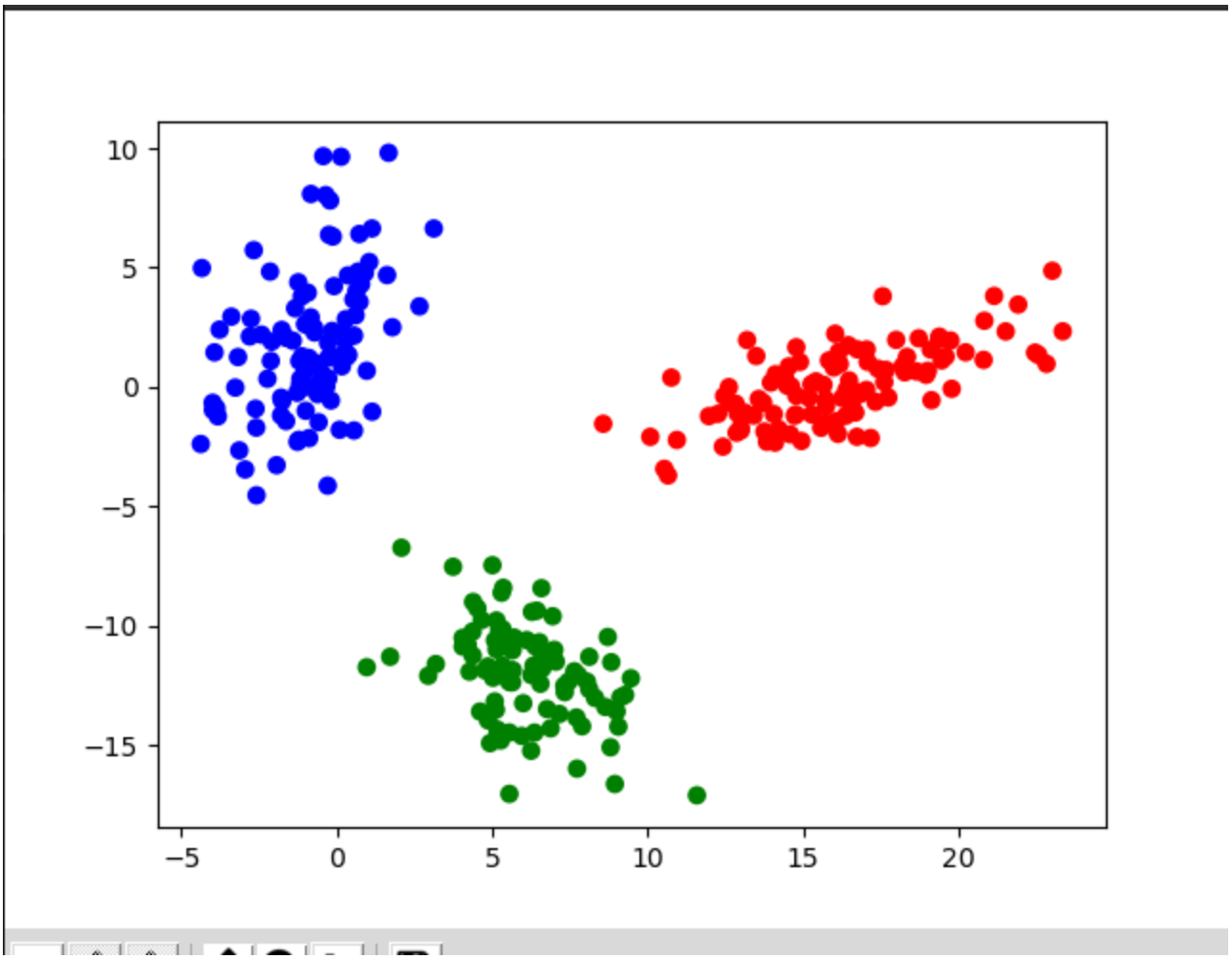
Accuracy : 0.89

Ques 4

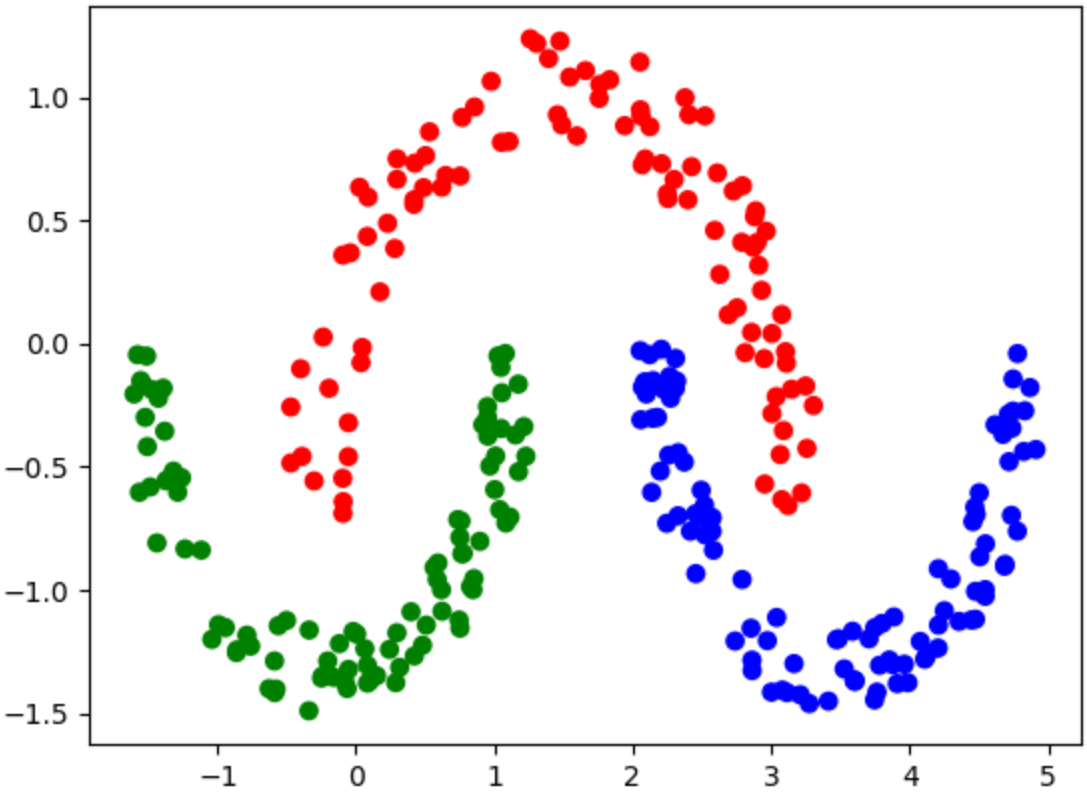
For linearly separable data, both perceptron and FCNN will perform well with an accuracy of 100%. However, in case of non linearly separable data, performance of FCNN will be better than perceptron:

A perceptron will just give a linear boundary whereas FCNN will stitch the linear boundaries obtained from all the neurons present in the model to give an approximated non-linear boundary. The performance further increases with an increase in the number of hidden layers.

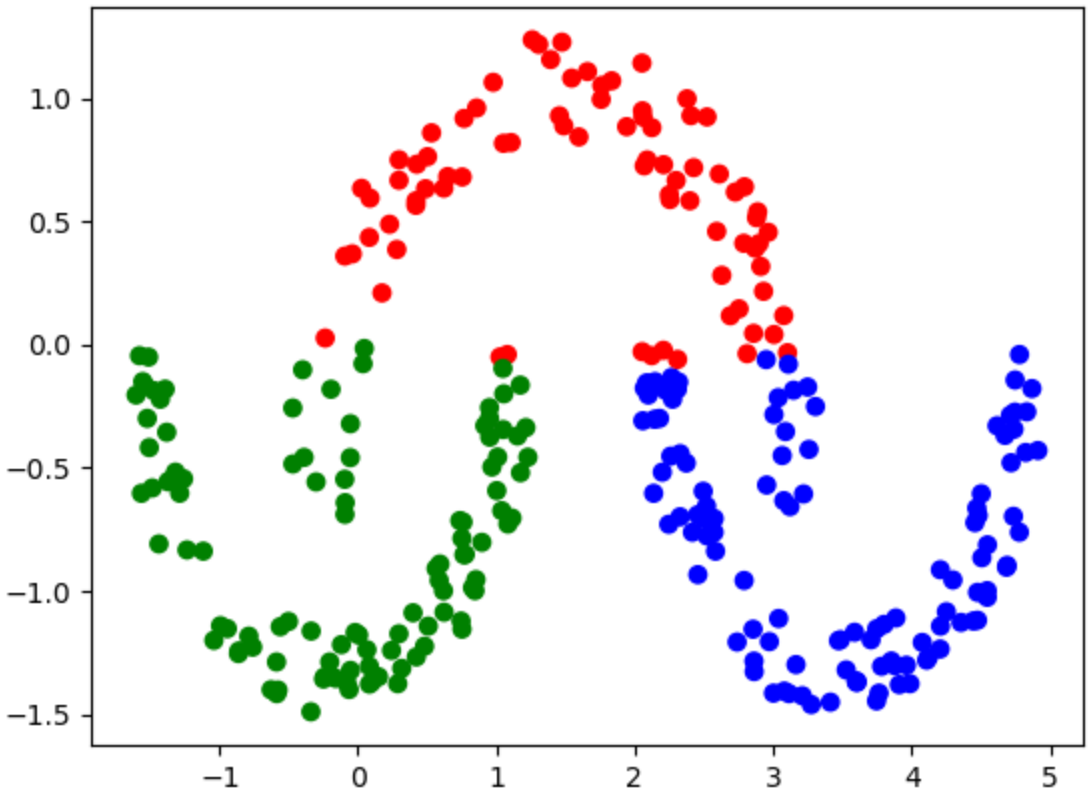

10
5
0
−5
−10
−15
−5
0
5
10
15
20



Predicted classes for LS test data



Actual classes for NLS test data



Predicted classes for NLS test data

Que 6.

$$\Delta w_{jk}^{(o)} = \eta \, \delta_{nk}^{(o)} \, h_{nj}$$

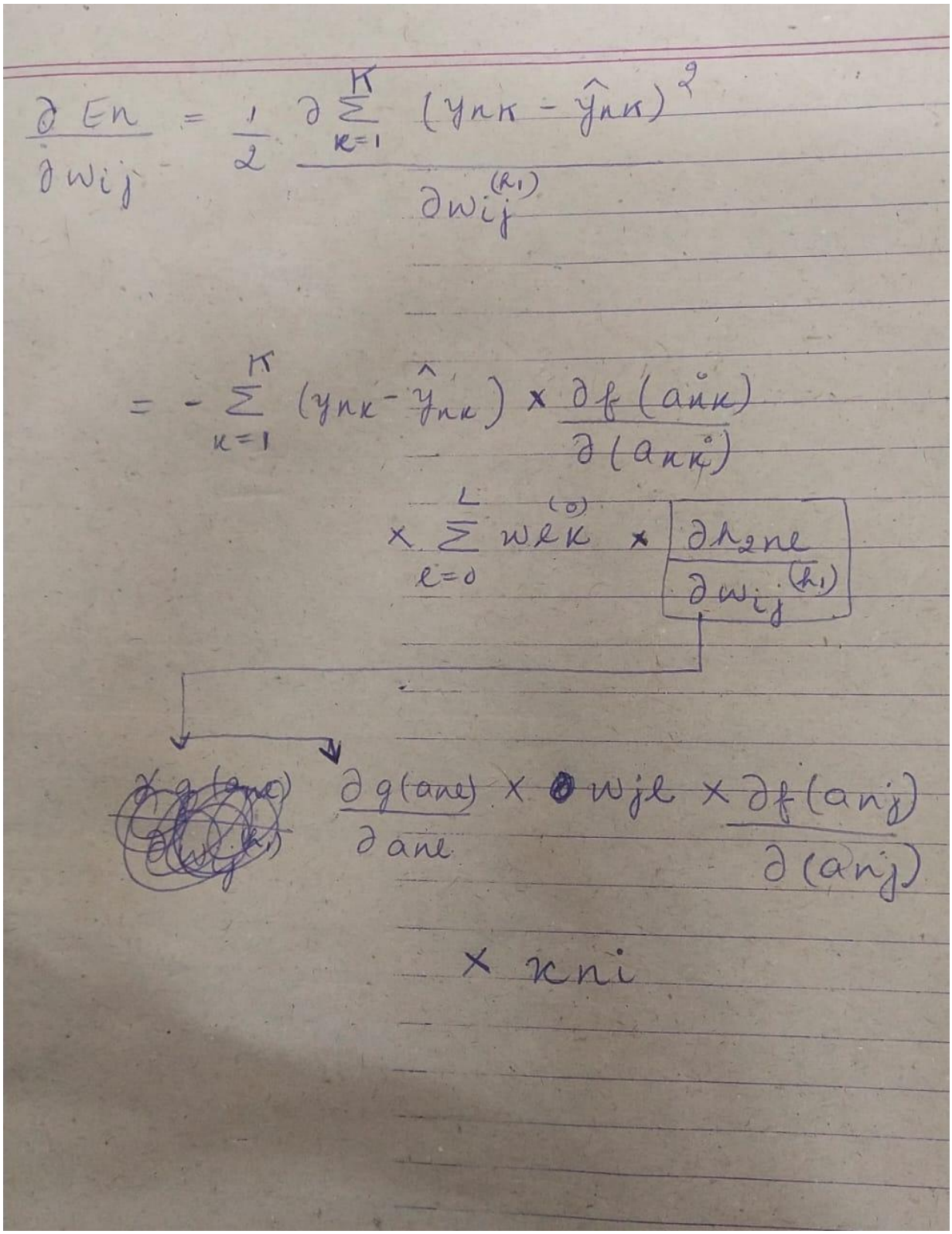
where $\delta_{nk}^{(o)} = (y_{nk} - \hat{y}_{nk}) \dfrac{\partial f(a_{nk}^{(o)})}{\partial a_{nk}^{(o)}}$

$$\Delta w_{ij}^{(h)} = -\eta \dfrac{\partial E_n}{\partial w_{ij}^{(h)}}$$

$$= \eta \, \delta_{nj}^{(h)} \, x_i$$

where $\delta_{nj}^{(h)} = \left[\sum_{k=1}^{K} \delta_{nk}^{(o)} w_{jk}^{(o)}\right] \dfrac{\partial g(a_{nj})}{\partial a_{nj}}$

Expressions for 1 hidden layer



Expression for 2 hidden layers

Regression Tasks

Perceptron

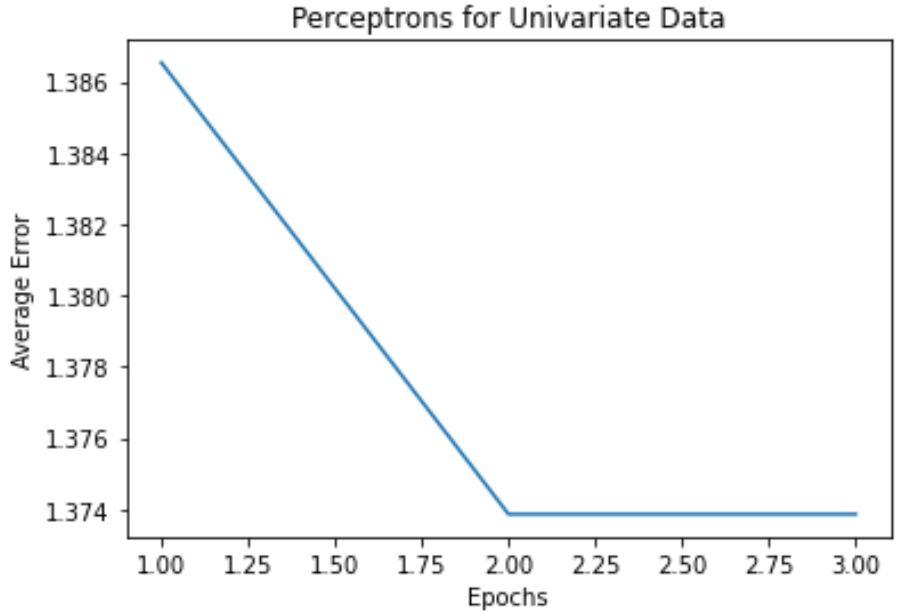
We wrote code for a linear perceptron from scratch and used it to train on both the Univariate and Bivariate Data.

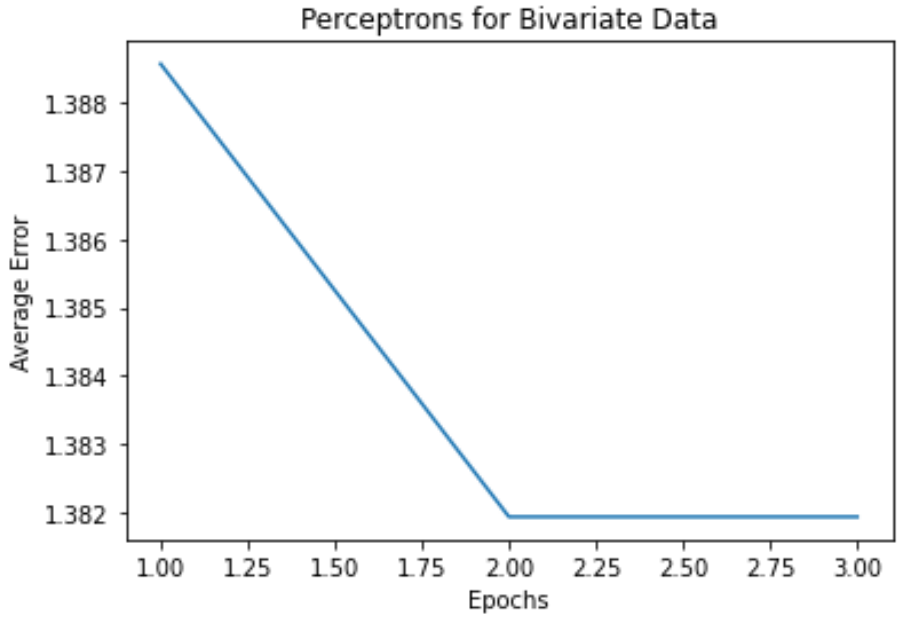
The parameters for both models were set as:

- Eta (Learning Rate Parameter) = 0.4
- Max Epochs = 10
- Error Threshold for successive epochs = 1×10^{-3}

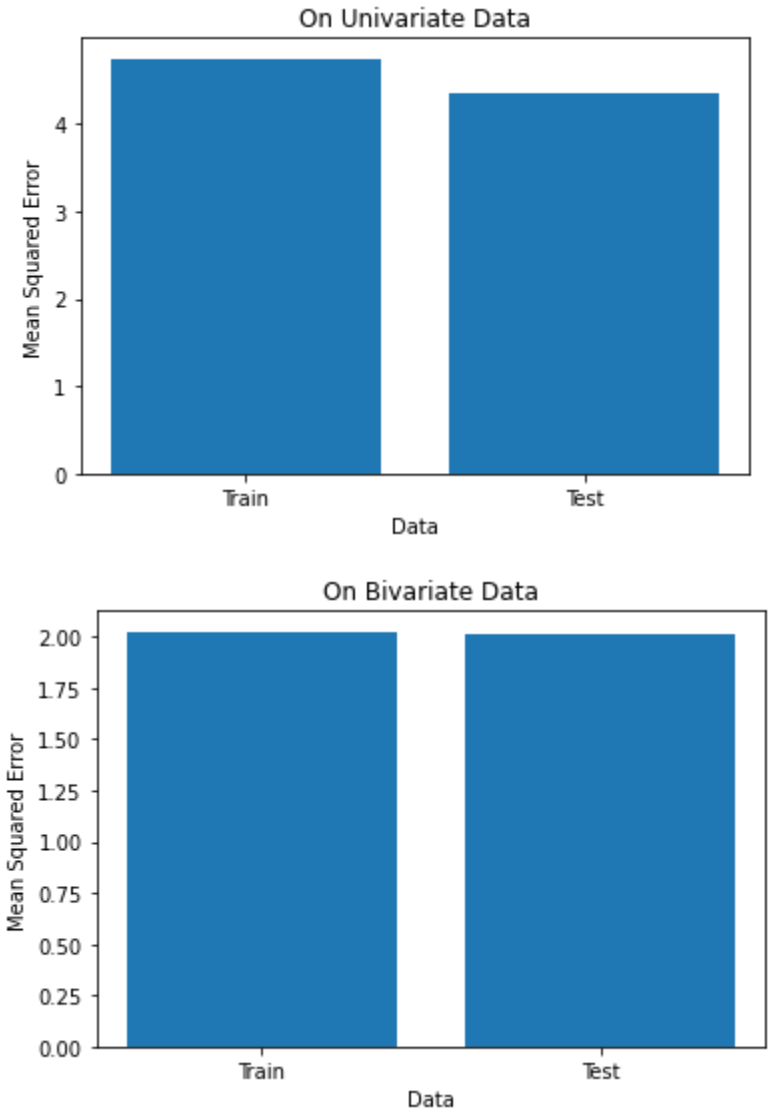
Train Test Split was done in the ratio 60:40 since the 20% from validation was also supposed to be considered as additional test data.

Q1 Plot of average error vs epochs

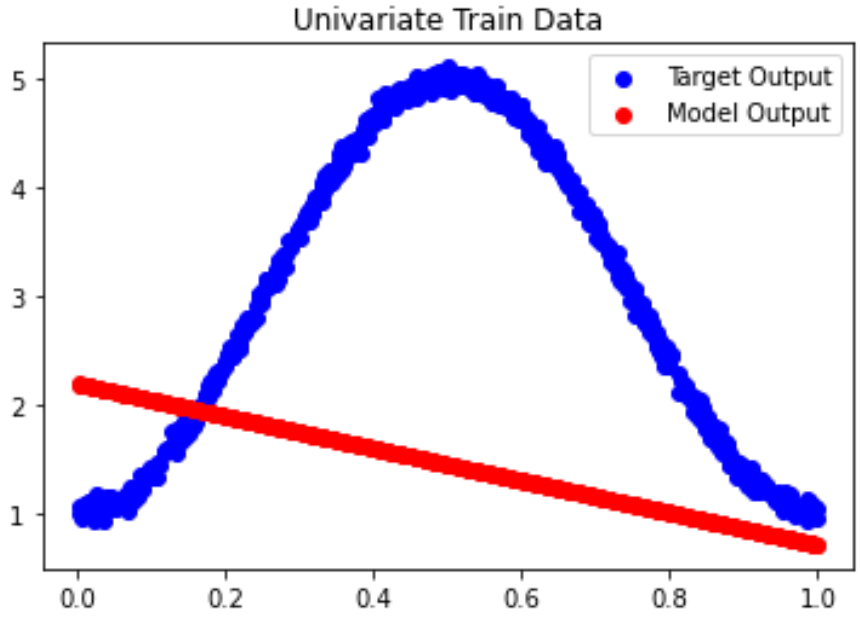


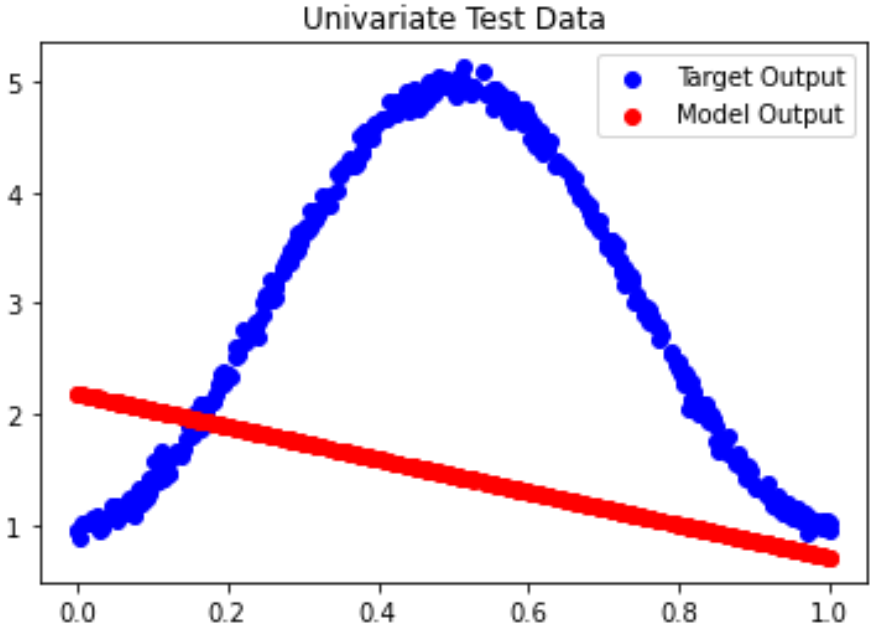


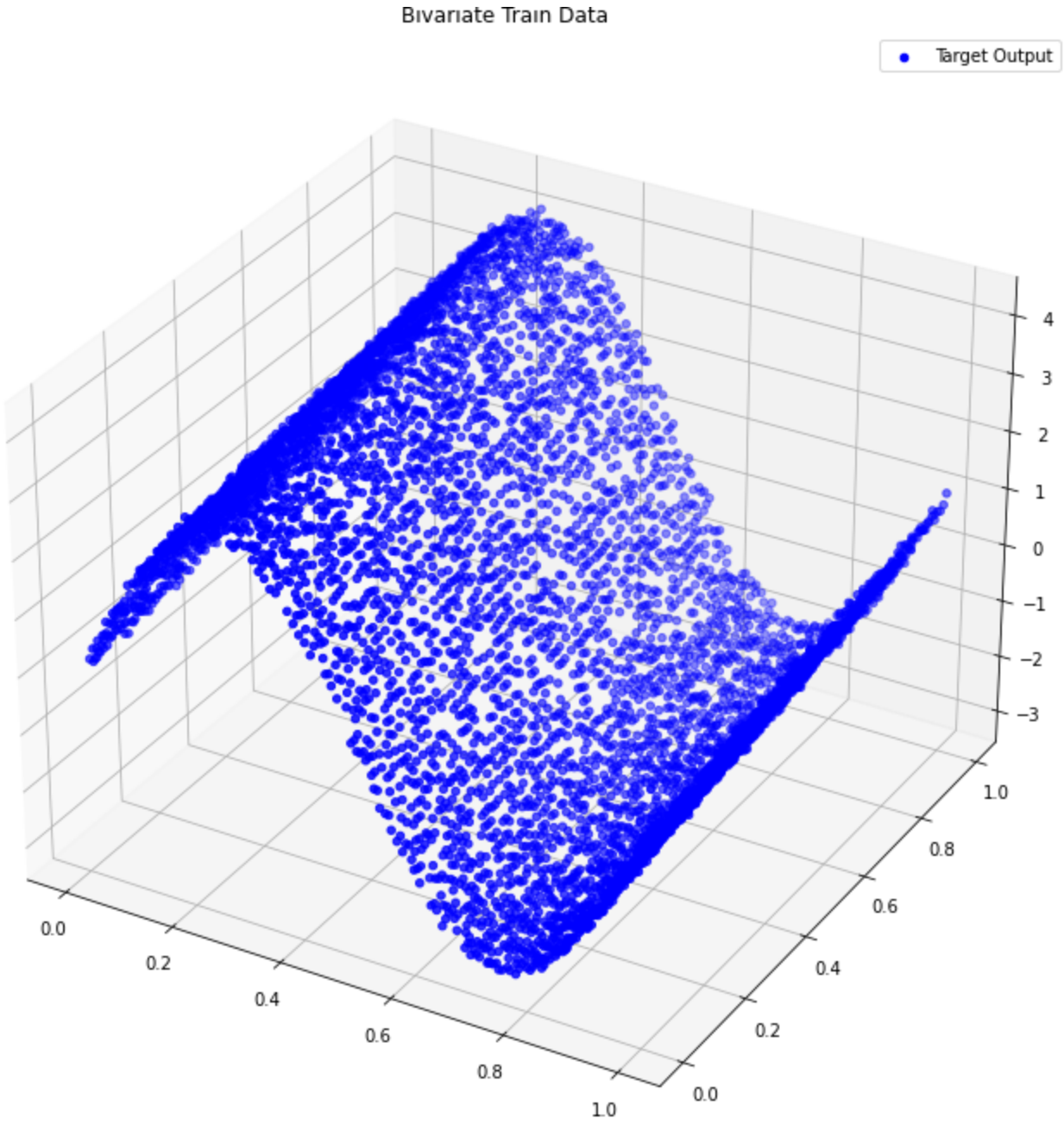
Q2 Plot of average error vs epochs

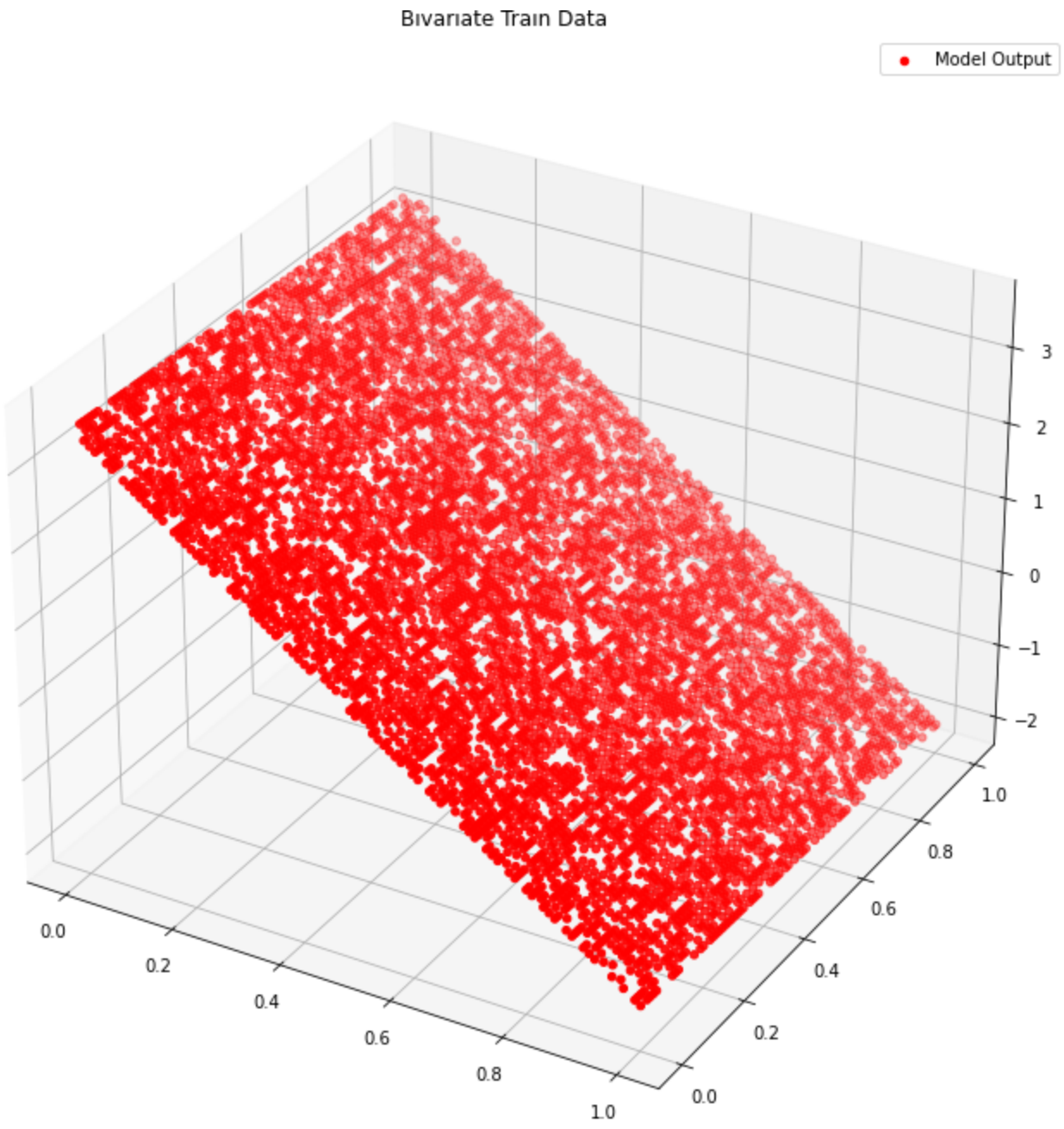


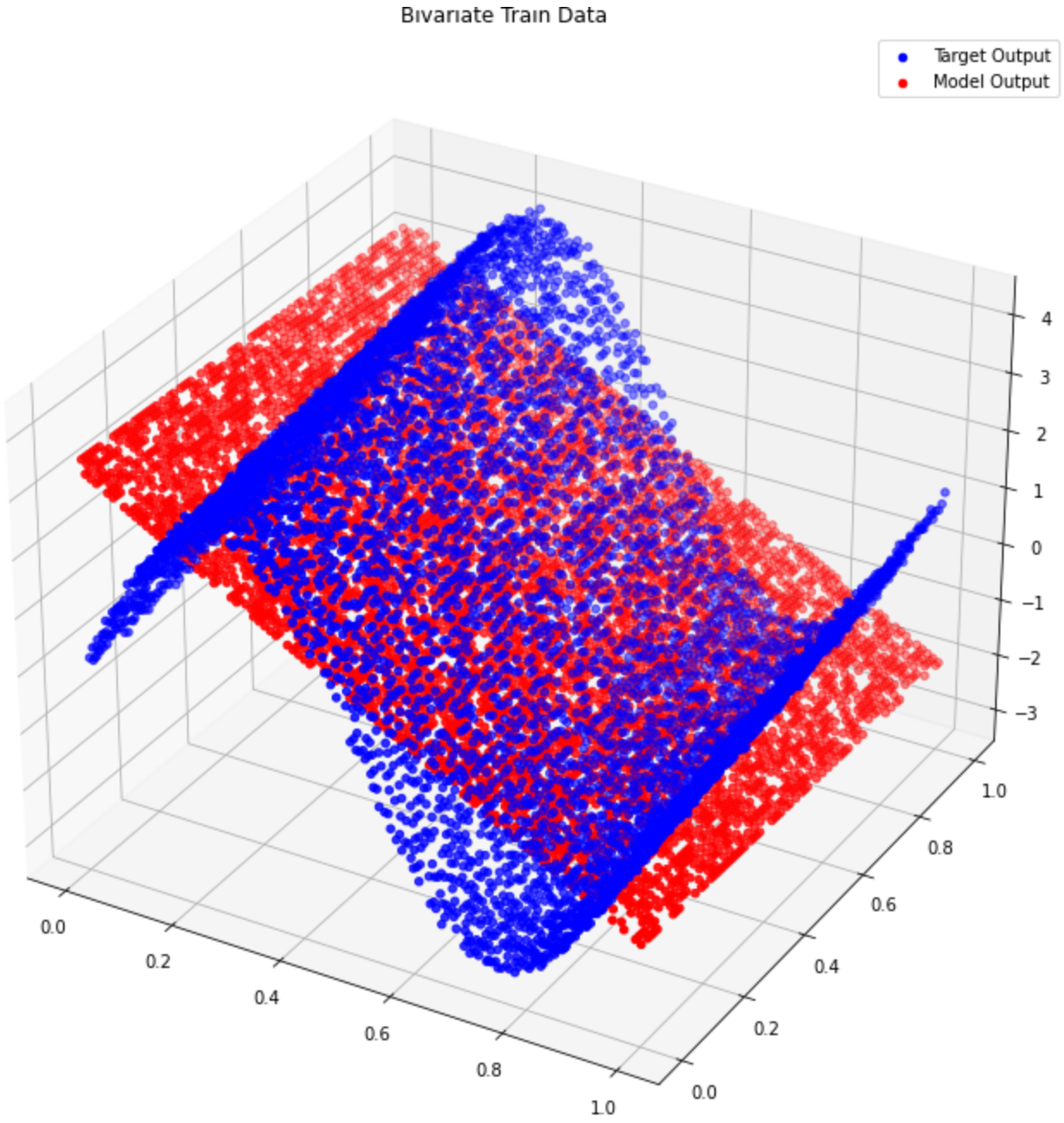
Q3 Model Output and Target Output

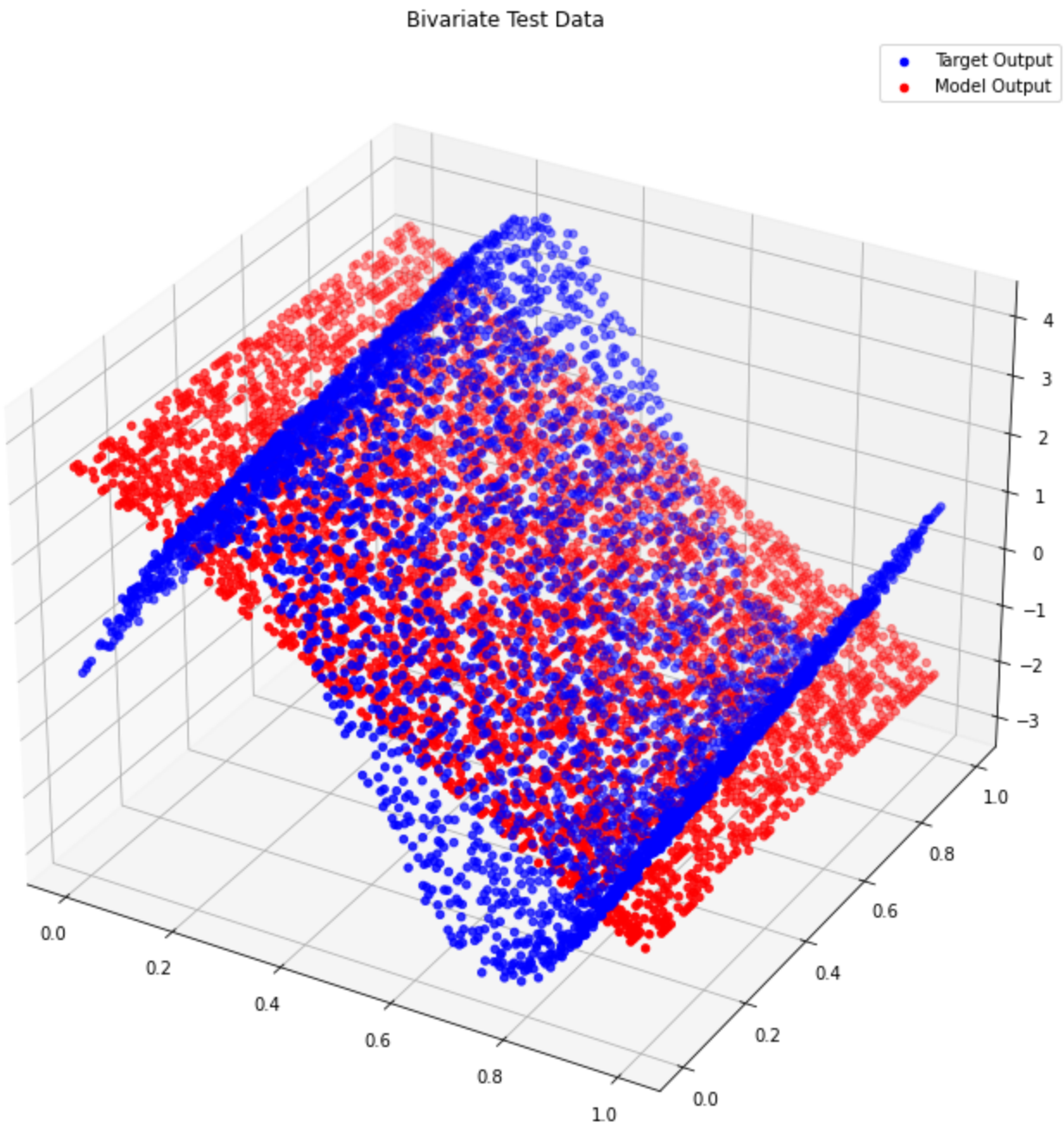

Univariate Train Data
Target Output
Model Output
5
4
3
2
1
0.0
0.2
0.4
0.6
0.8
1.0


Univariate Test Data
Target Output
Model Output
5
4
3
2
1
0.0
0.2
0.4
0.6
0.8
1.0

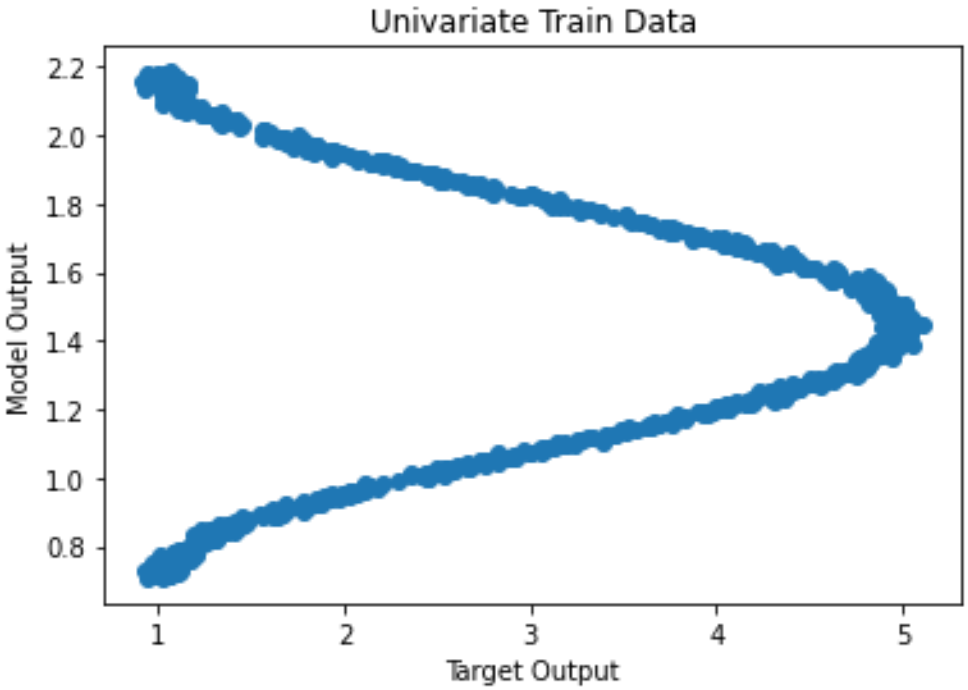

Bivariate Train Data
Target Output
4
3
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0


Bivariate Train Data
Model Output
3
2
1
0
−1
−2
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0

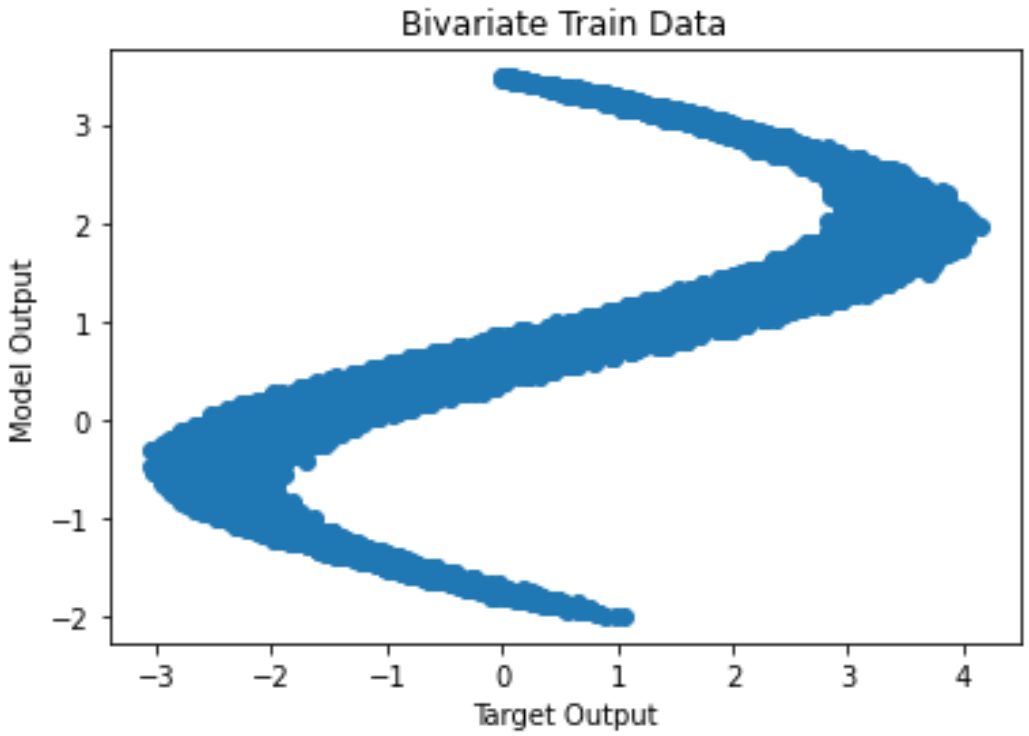

Bivariate Train Data
Target Output
Model Output
4
3
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0

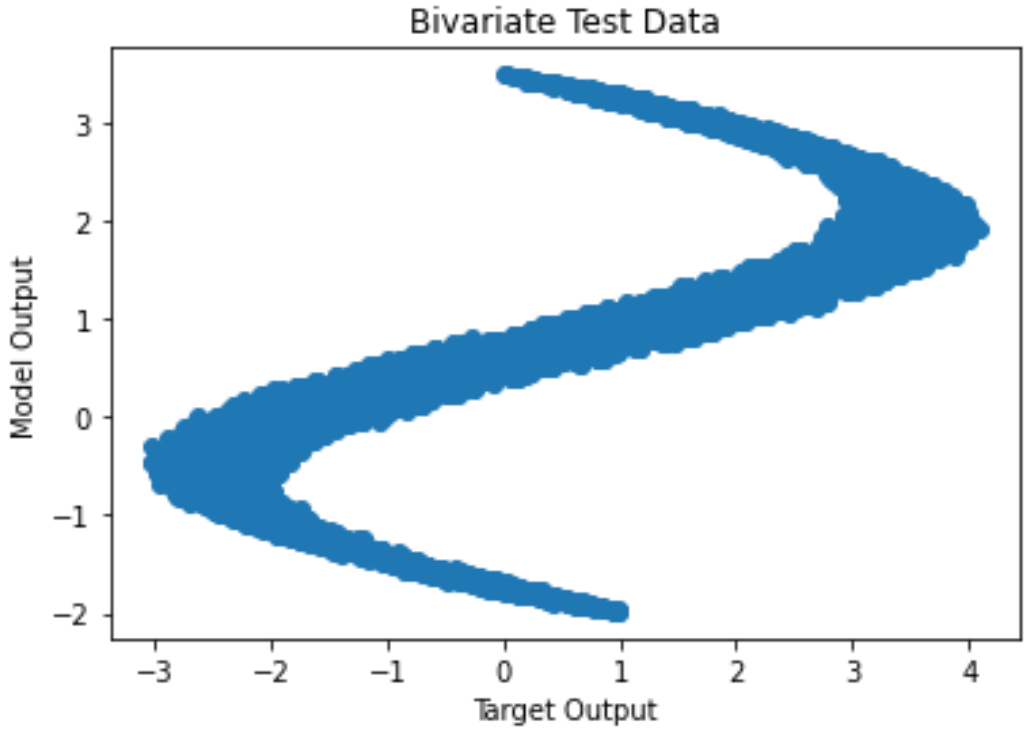


Q4 Model Output vs Target Output


Univariate Train Data
Model Output
Target Output
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
1
2
3
4
5


Univariate Test Data
Model Output
Target Output
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
1
2
3
4
5





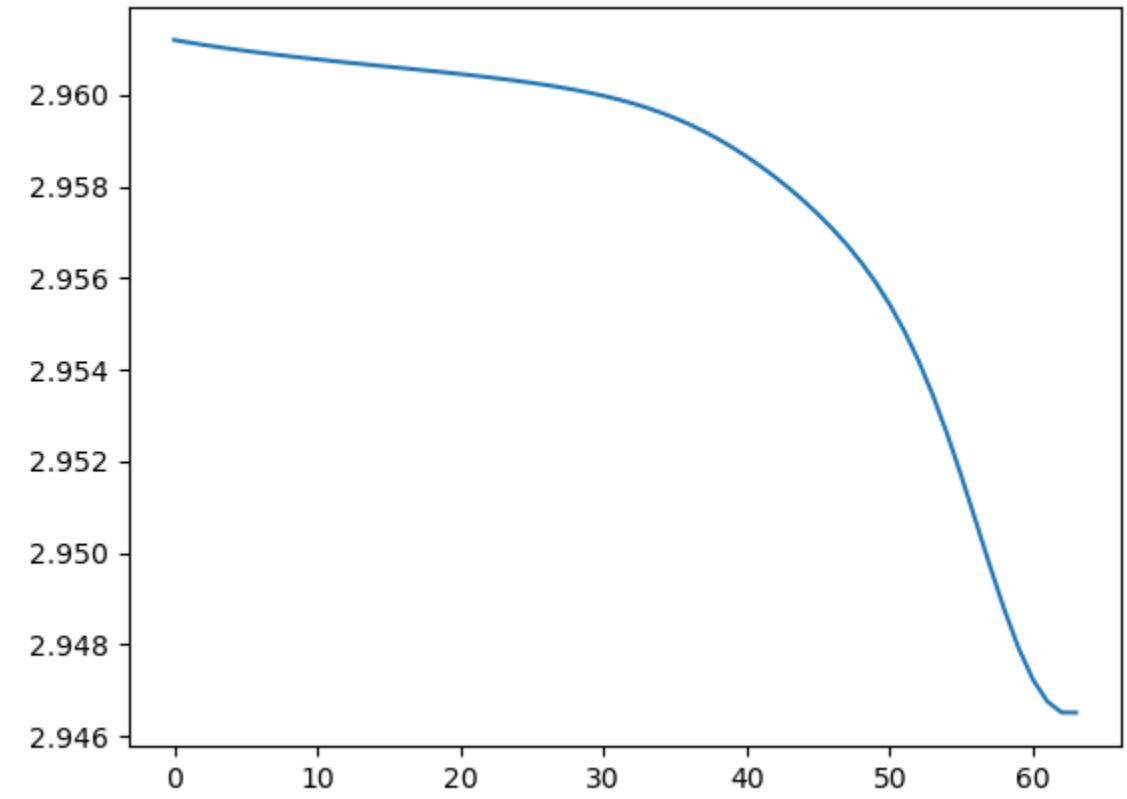
FCNN

We implemented multi layer perceptron with 1 hidden layer for univariate data.

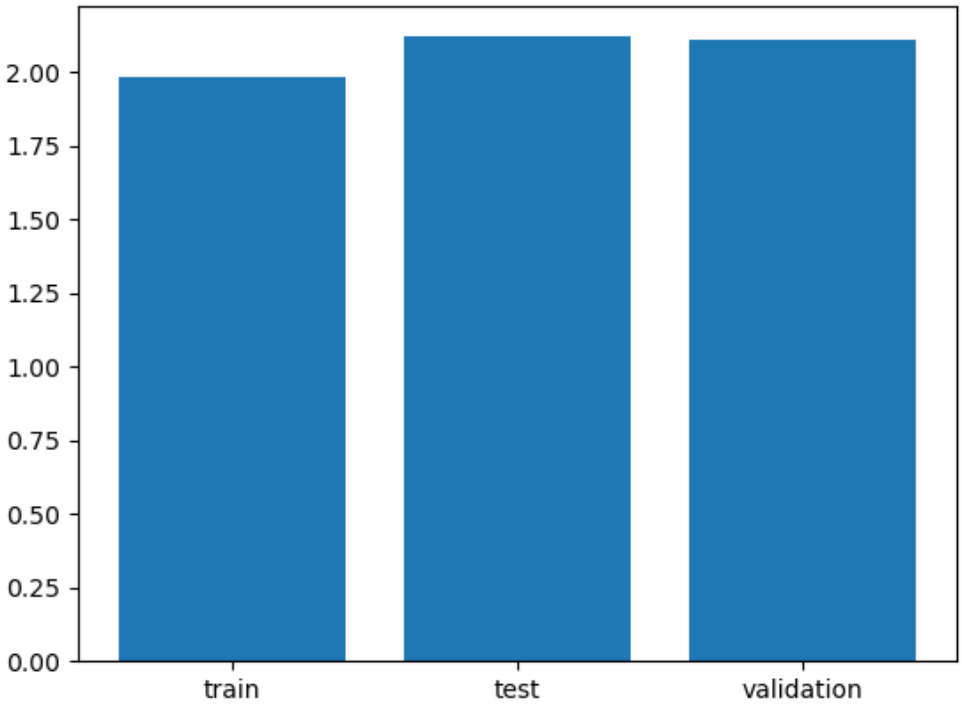
Parameters for this model were set as:

1) Eta = 0.01
2) Number of nodes in the hidden layer (n_hidden) = 4

Plot of average error vs epochs

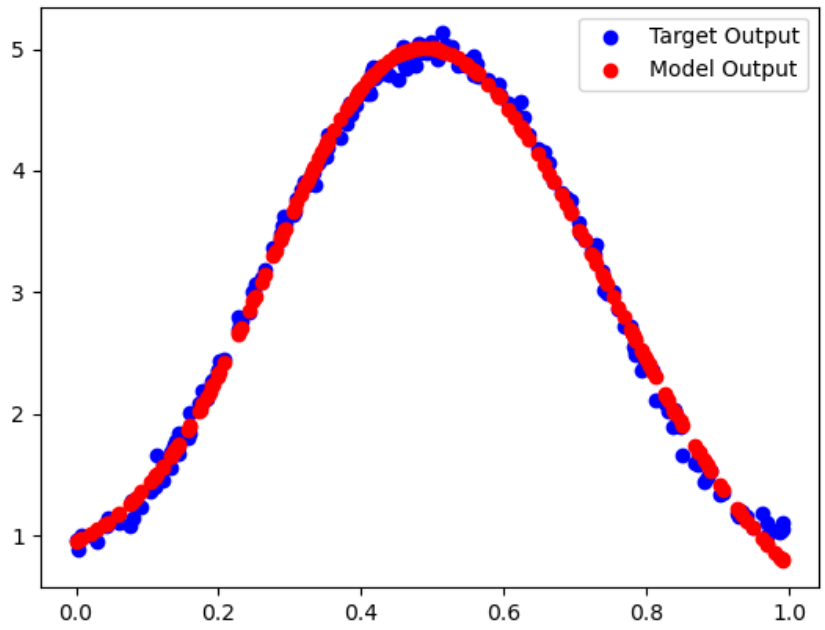


Plots of MSE

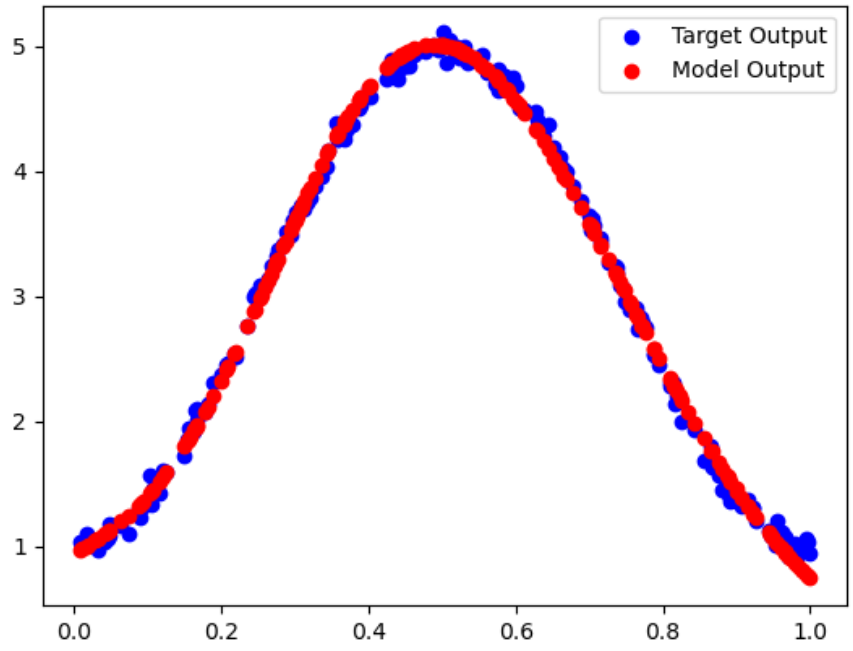


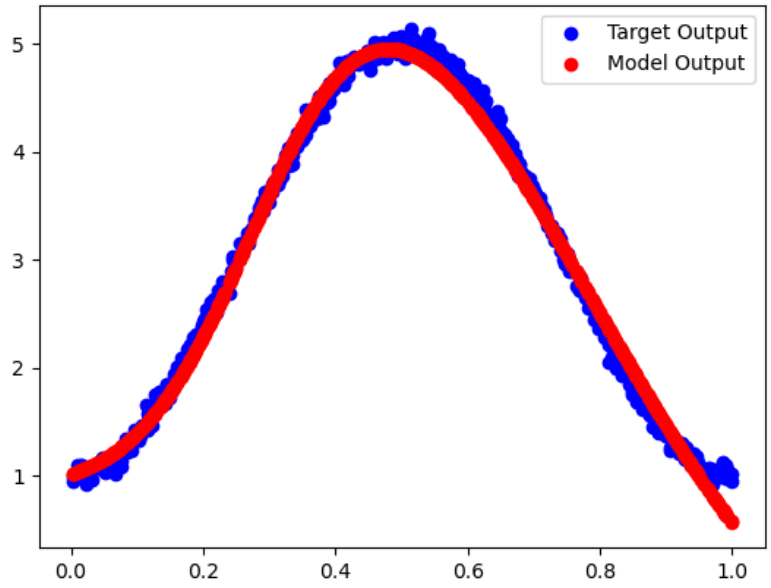
Q3.

Validation data



Test data

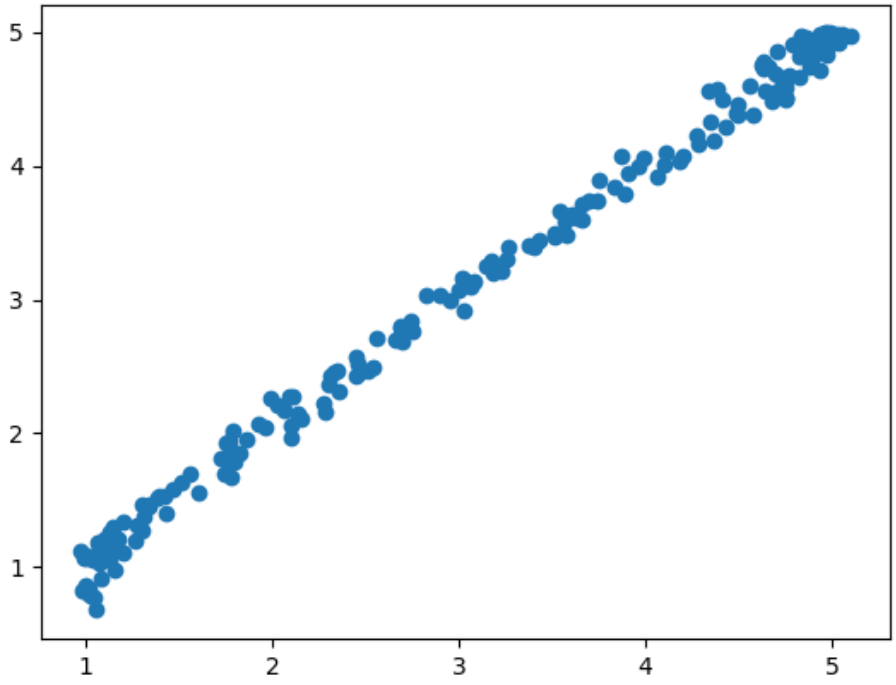




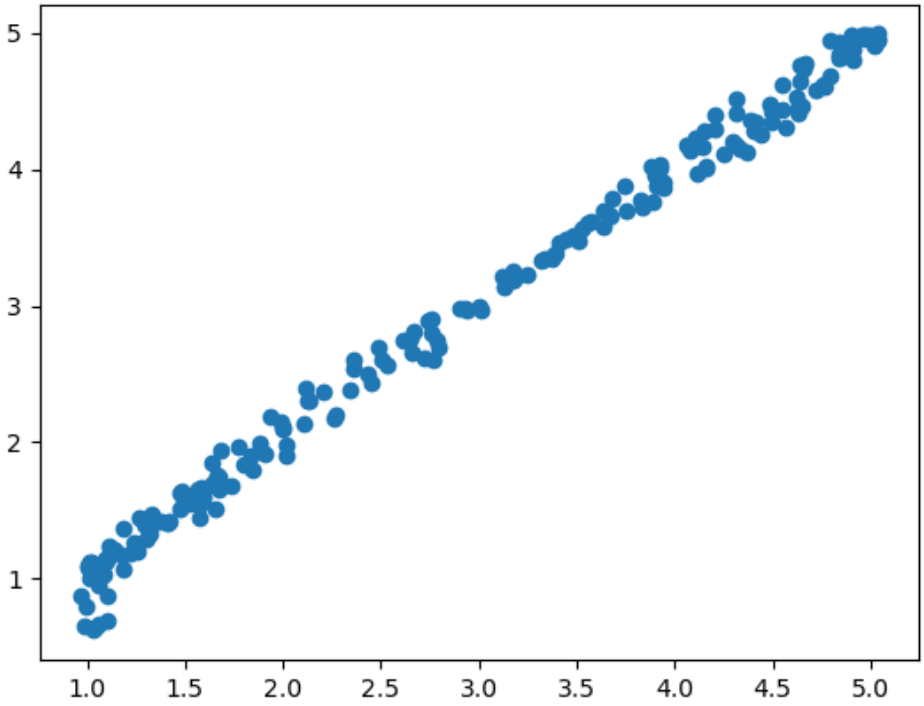
Training data

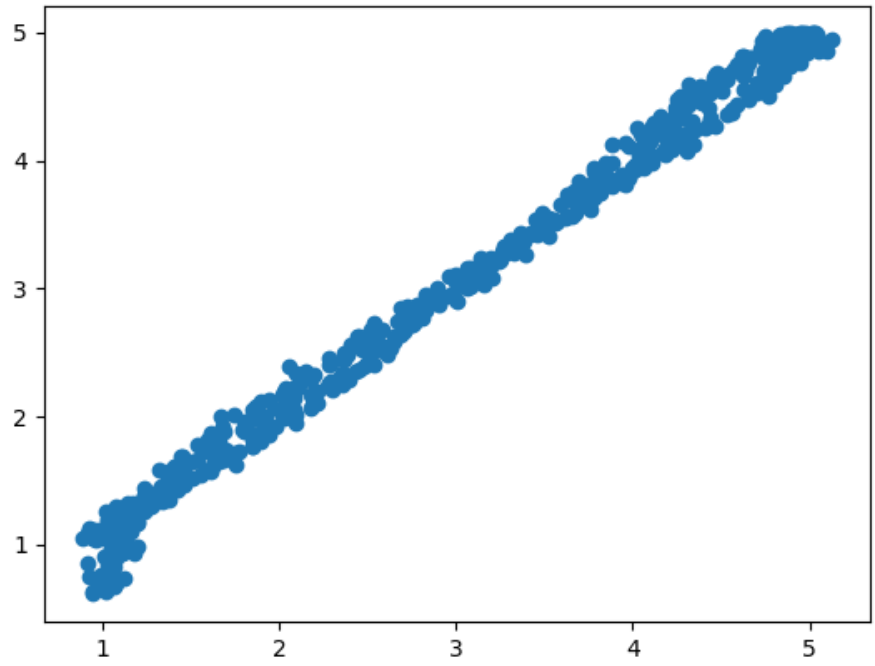
Q4.

Validation data



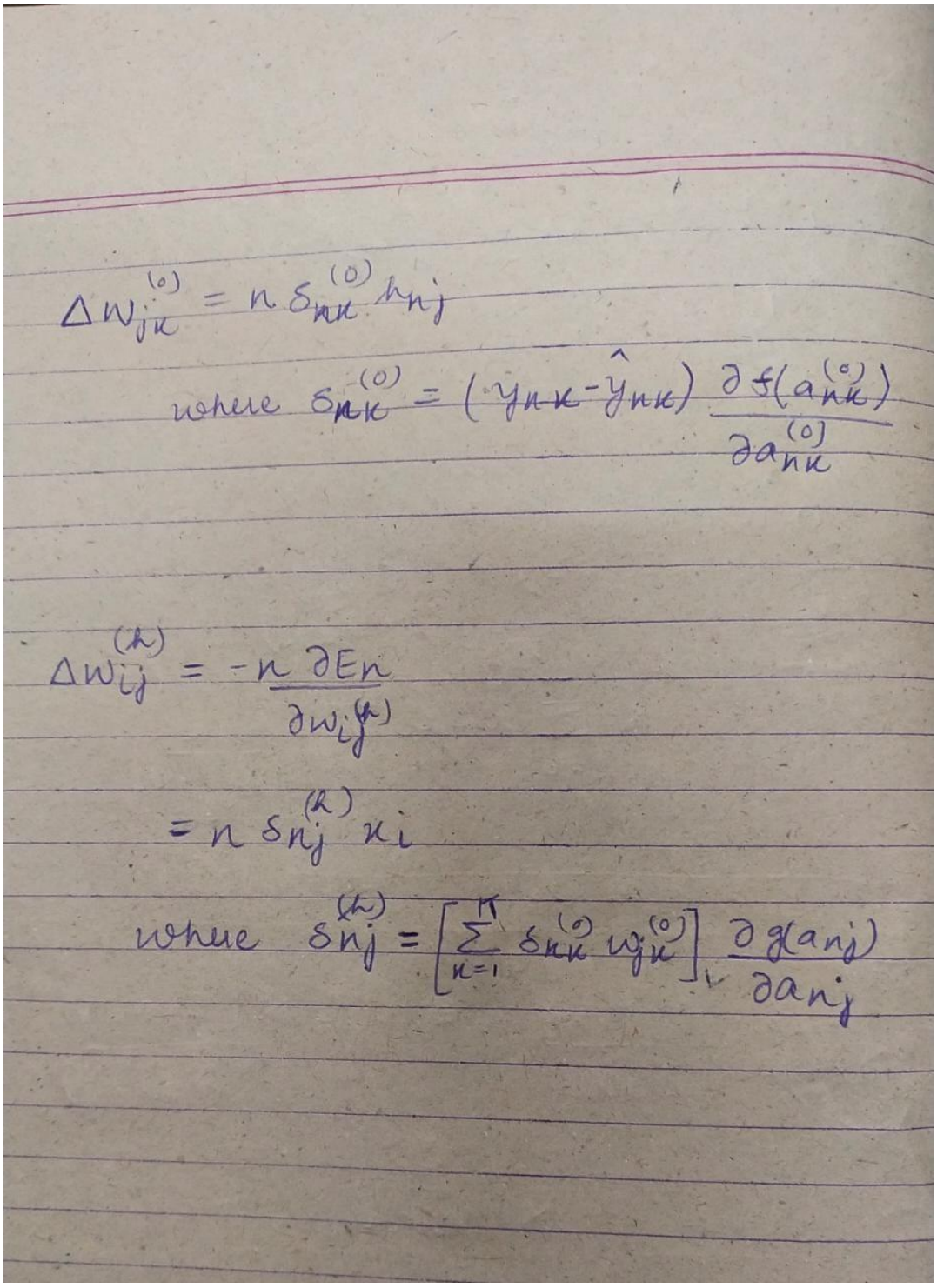
Test data



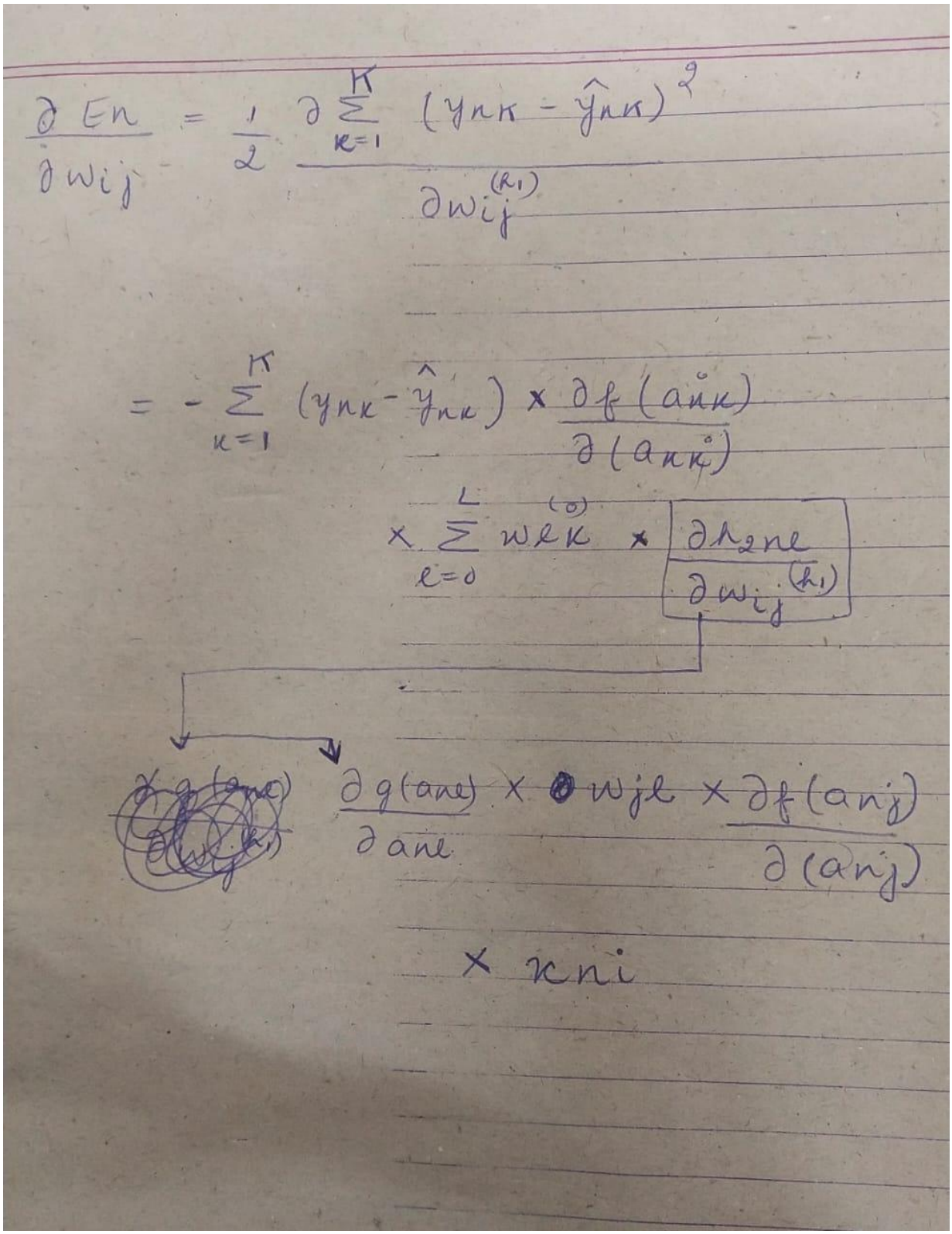


Training data

Que 6.



Expression for 1 hidden layer



Expression for 2 hidden layers