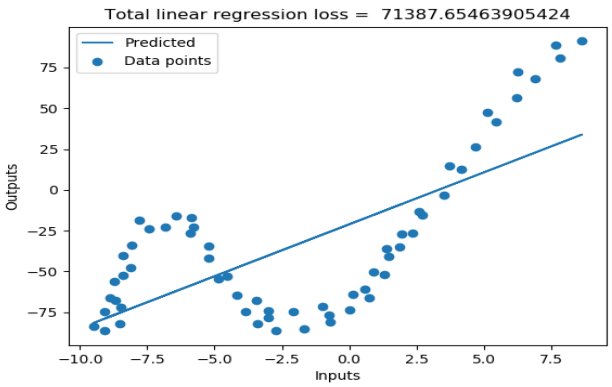
Hakan Gulcu
21702275
GE461-Project3
01.05.2022

a)



As can be seen from plot, it is not sufficient to use a linear regressor and it is necessary to use an ANN with a single layer. The reason is that linear regressor has big sum of square loss. For the first look, linear regression error is 71387 and ANN error is 165123, however, when we increase the number of hidden units it is clear that ANN is better choice for this dataset. The minimum number of units should be chosen by testing programs. I used 2,4,8,16 and 32 hidden units as stated in assignment and I will explain them at below.

# of Hidden Units	**Error**
2	165123.763
4	20590.707
8	21604.694
16	20473.215
32	14499.119

Having 2 hidden units creates much more errors than others. The best is to use which has less error so 32 hidden units. However, the minimum should be 4.

To find out good value for the learning rate, I chose 32 hidden units and 10000 epochs (for weight). After that I tested learning rates for 0.01, 0.001, 0.0001, 0.00001

Learning Rate	**Error**
0.01	60501.464
0.001	16038.965
0.0001	33642.995
0.00001	111031.792

As can be seen from table, the best rate is 0.001 because it has less error. Also, having it can be stated that having less learning rate may not be good.

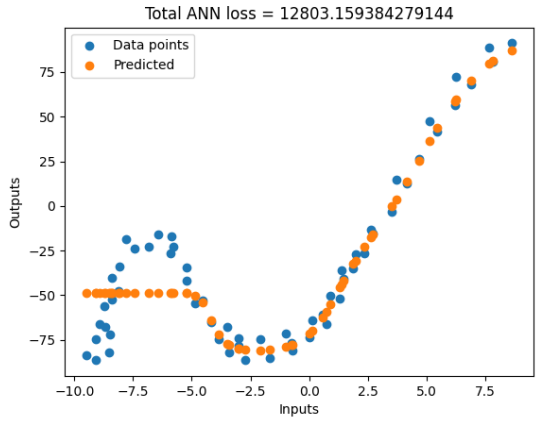
At the beginning, initializing weight is random. However, we can test it to choose to find best epoch value. To make experiment consistent, number of hidden unit chosen as 32 and learning rate chosen as 0.001. After that the experiment done with epoch values 10, 100, 1000, 10000, 100000, 1000000.

Epoch	**Error**
10	203160.616
100	91695.963
1000	34338.922
10000	17490.996
100000	12745.128
1000000	7600.996

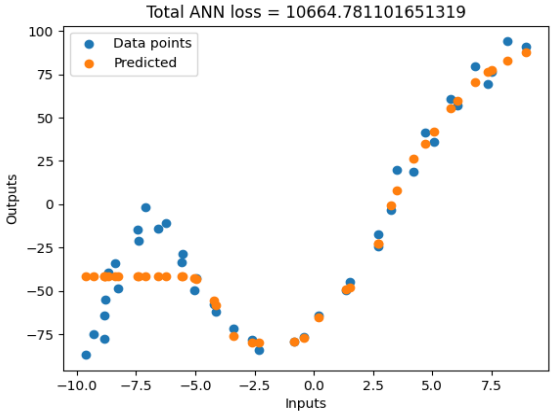
When we look at table 1000000 has less error than others and should be chosen. However, it takes lots of time than others. Therefore, I decided to chose it as 100000 and it is working accurately and fastly.

No, normalization does not affect the learning process a lot. It has an affect but cannot be said that it is significant.

b)



Train

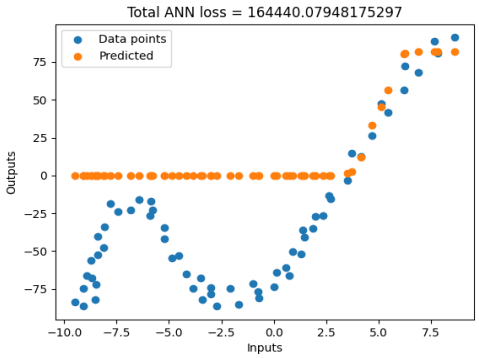


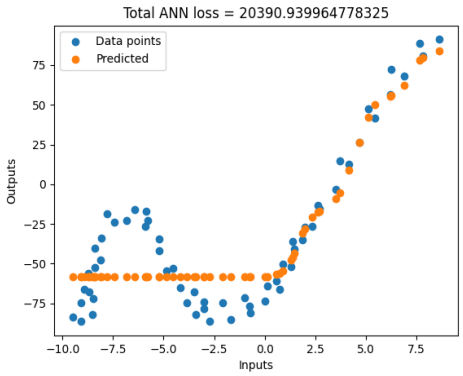
Test

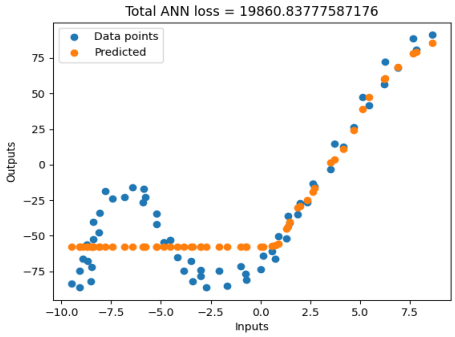
The best configuration is the best of each hidden units, learning rate and epoch.
ANN used: **32**
Learning rate: **0.001**
Range of initial weights: **0.1**
Number of epochs: **100000**
When to stop: **100000**
Is normalization used: **No**
Training loss (averaged over training instances): **12803.159384279144**
Test loss (averaged over test instances): **10664.781101651319**

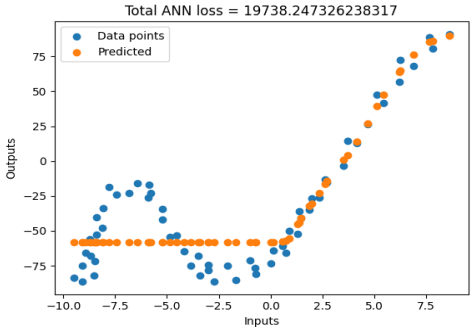
c)

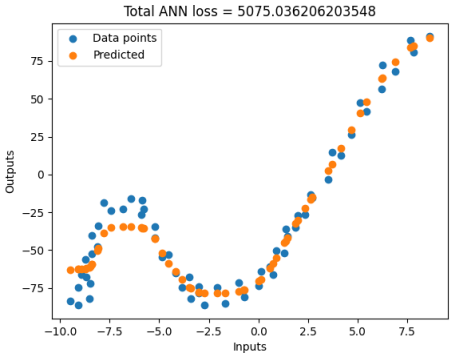
For this question, I made different experiments. Firstly, I did change hidden units and did not change learning rate and epoch for both training and test sets. And I used the same way for learning rate and epoch. By this way I can give better answers and have better understanding.











Those plots are nice to observe how the number of units of an ANN its ability to learn. When we look at the plots of ANNs with 2, 4, 8, 16 and 32 hidden units, respectively, from left to right, it is clear that having less hidden units is bad for learning process and having more is better. We can see that the tables with 4, 8 and 16 hidden units are very similar to each other and the total ANN losses are also very close to each other. However, when it is 32, we get much better results.

Hidden Unit	**Training Data Loss**	**Test Data Loss**	**Avg Training Loss**	**Avg Test Loss**	**Std Derivation Train**	**Std Derivation TEST**
2	164440.079	87102.651	2740.667	2750.562	2765.672	2775.650
4	20390.939	86579.046	339.848	2738.820	344.629	2763.809
8	19860.837	86533.668	331.013	2732.280	335.720	2757.214
16	19738.247	16342.715	328.970	379.109	333.660	384.216
32	5075.036	10894.624	84.583	263.624	87.331	267.773

This table is the result of above plots and also test data. For any error, it can be clearly seen that error rates are decreasing while hidden unit count is increasing. Therefore, it is clear that having more hidden units affects ability to learn positively. However, at some points, there are slight decreases. For example, for the 2 and 4 hidden units, results are very close.

Finally, I want to mention the epoch affect.

16 hidden units and 0.001 learning rate for epoch on training data:

Total loss = 20427.523, Average loss = 340.458, Standard Derivation = 345.244 for epoch=10000
Total loss = 19894.413, Average loss = 331.573, Standard Derivation = 336.285 for epoch=100000

16 hidden units and 0.001 learning rate for epoch on test data:

Total loss = 16518.431, Average loss = 399.672, Standard Derivation = 409.530 for epoch=10000
Total loss = 16386.558, Average loss = 402.888, Standard Derivation = 406.274 for epoch=100000

It is clear that having more epoch or weight has positive affect on learning rate like hidden units. However, it does not affect that much. Also, having less learning rate is better for learning affect. All results of mine can be seen from my source code.