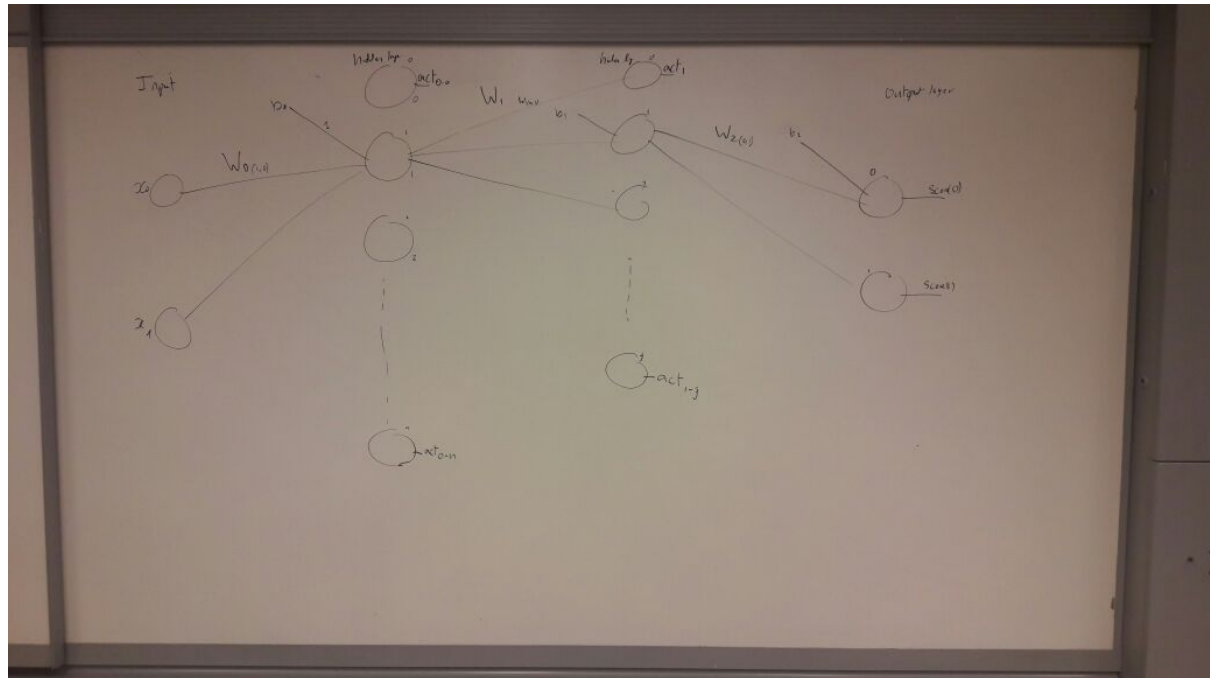


Assignment I
Neural Networks and Genetic Algorithms
Dr. Mohamed Mustafa

Mohamed Daw
900142433

1. 2-layer network architecture and derivation of Error of Backpropagation



Network Architecture

The handwritten notes show the derivation of backpropagation equations for a 2-layer network. The equations are as follows:

We need to compute the following:

$$\frac{\partial \text{loss}}{\partial w_0}, \frac{\partial \text{loss}}{\partial w_1}, \frac{\partial \text{loss}}{\partial b_0}, \frac{\partial \text{loss}}{\partial b_1}, \frac{\partial \text{loss}}{\partial a_1}, \frac{\partial \text{loss}}{\partial a_2}$$

Forward pass equations:

$$act_1 = \max(0, w_1 x_1 + b_1)$$

$$act_2 = \max(0, w_2 x_2 + b_2)$$

$$\text{Sum} = w_1 act_1 + b_1$$

$$\text{loss} = \log \frac{e^{\text{Sum}}}{2}$$

Backward pass equations (derivatives):

$$\frac{\partial \text{loss}}{\partial w_0} = \frac{\partial \text{loss}}{\partial \text{Sum}} \times \frac{\partial \text{Sum}}{\partial w_0}$$

$$\frac{\partial \text{loss}}{\partial w_1} = \frac{\partial \text{loss}}{\partial \text{Sum}} \times \frac{\partial \text{Sum}}{\partial act_1} \times \frac{\partial act_1}{\partial w_1}$$

$$\frac{\partial \text{loss}}{\partial w_2} = \frac{\partial \text{loss}}{\partial \text{Sum}} \times \frac{\partial \text{Sum}}{\partial act_2} \times \frac{\partial act_2}{\partial w_2}$$

$$\frac{\partial \text{loss}}{\partial b_0} = \frac{\partial \text{loss}}{\partial \text{Sum}} \times \frac{\partial \text{Sum}}{\partial b_0}$$

$$\frac{\partial \text{loss}}{\partial b_1} = \frac{\partial \text{loss}}{\partial \text{Sum}} \times \frac{\partial \text{Sum}}{\partial act_1} \times \frac{\partial act_1}{\partial b_1}$$

$$\frac{\partial \text{loss}}{\partial a_1} = \frac{\partial \text{loss}}{\partial \text{Sum}} \times \frac{\partial \text{Sum}}{\partial act_1} \times \frac{\partial act_1}{\partial a_1}$$

$$\frac{\partial \text{loss}}{\partial a_2} = \frac{\partial \text{loss}}{\partial \text{Sum}} \times \frac{\partial \text{Sum}}{\partial act_2} \times \frac{\partial act_2}{\partial a_2}$$

Derivation of Backpropagation 1

$$\begin{aligned}
 \frac{\partial \text{loss}}{\partial \text{net}_1} &= \frac{e^{-\text{net}_1}}{1 + e^{-\text{net}_1}} - 1 \quad (\text{for correct class}) \\
 \frac{\partial \text{loss}}{\partial \text{net}_1} &= \begin{cases} \text{act}_1 & (\text{for } \text{net}_1 \geq 0) \\ 0 & (\text{for } \text{net}_1 < 0) \end{cases} \\
 \frac{\partial \text{loss}}{\partial \text{net}_1} &= \begin{cases} W_1 & (\text{for } \text{net}_1 \geq 0) \\ 0 & (\text{for } \text{net}_1 < 0) \end{cases} \\
 \frac{\partial \text{act}_1}{\partial \text{net}_1} &= \begin{cases} 1 & (\text{for } \text{net}_1 \geq 0) \\ 0 & (\text{for } \text{net}_1 < 0) \end{cases} \\
 \frac{\partial \text{act}_1}{\partial b_0} &= 1 \\
 \frac{\partial \text{act}_1}{\partial b_1} &= 1 \\
 \frac{\partial \text{act}_1}{\partial b_2} &= 1
 \end{aligned}$$

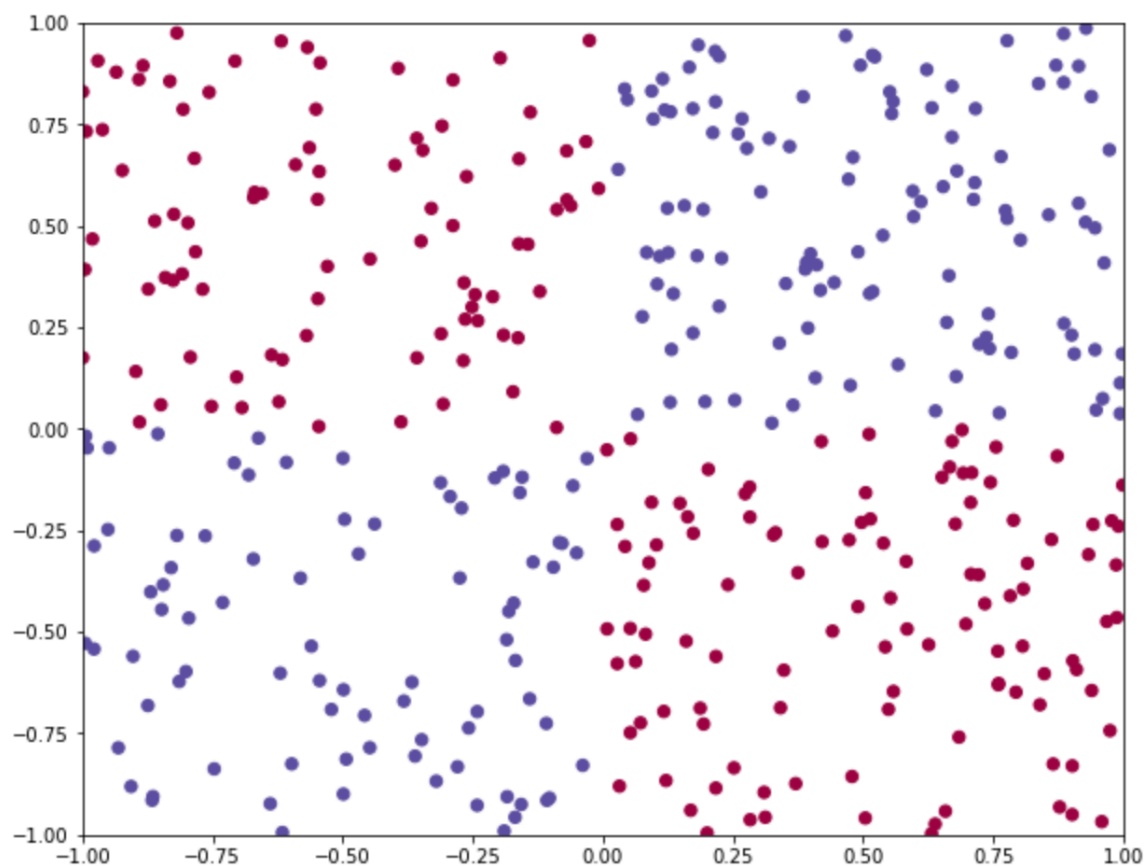
Derivation of Backpropagation 2

2. Network architecture differed depending upon the level of non linearity of the problem. The more non linear it was (face) the more layers and neurons were needed to conquer the problem. {ring, xor, circle, face}

 - Important note: I used regularization (L2) as to reduce over fitting ($\lambda = 1e-7$)
 - Learning Rate was set fixed at 0.1
 - I used Softmax loss as it was a classification problem.

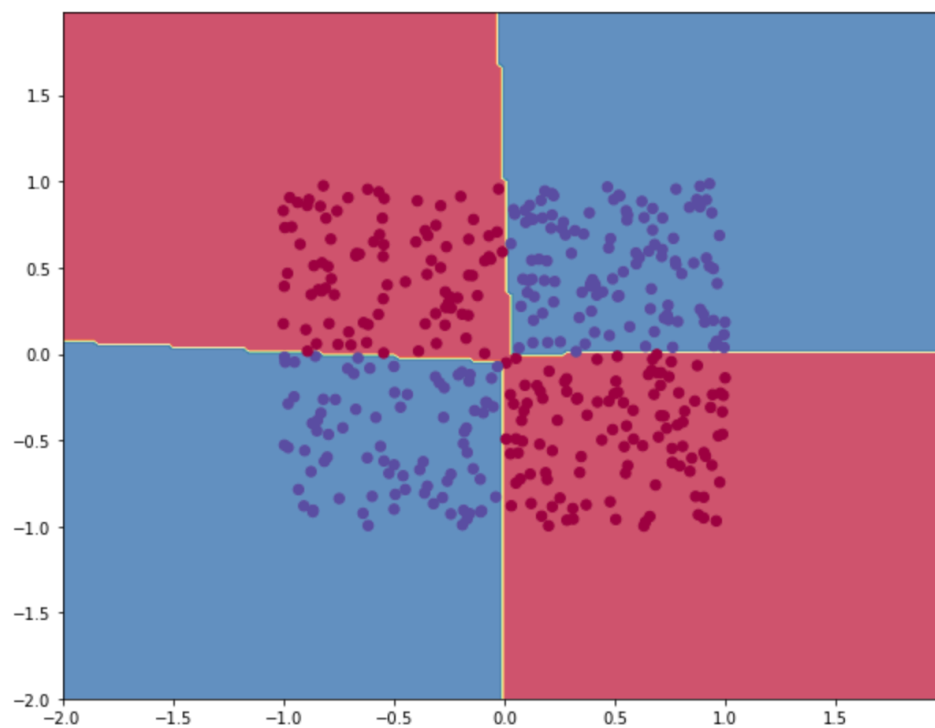
For the XOR problem:

The Architecture had one hidden layer with 20 neurons which was enough to conquer the problem and achieve a 100 % accuracy. Check the figure for data points and for the data boundaries with boundaries added.

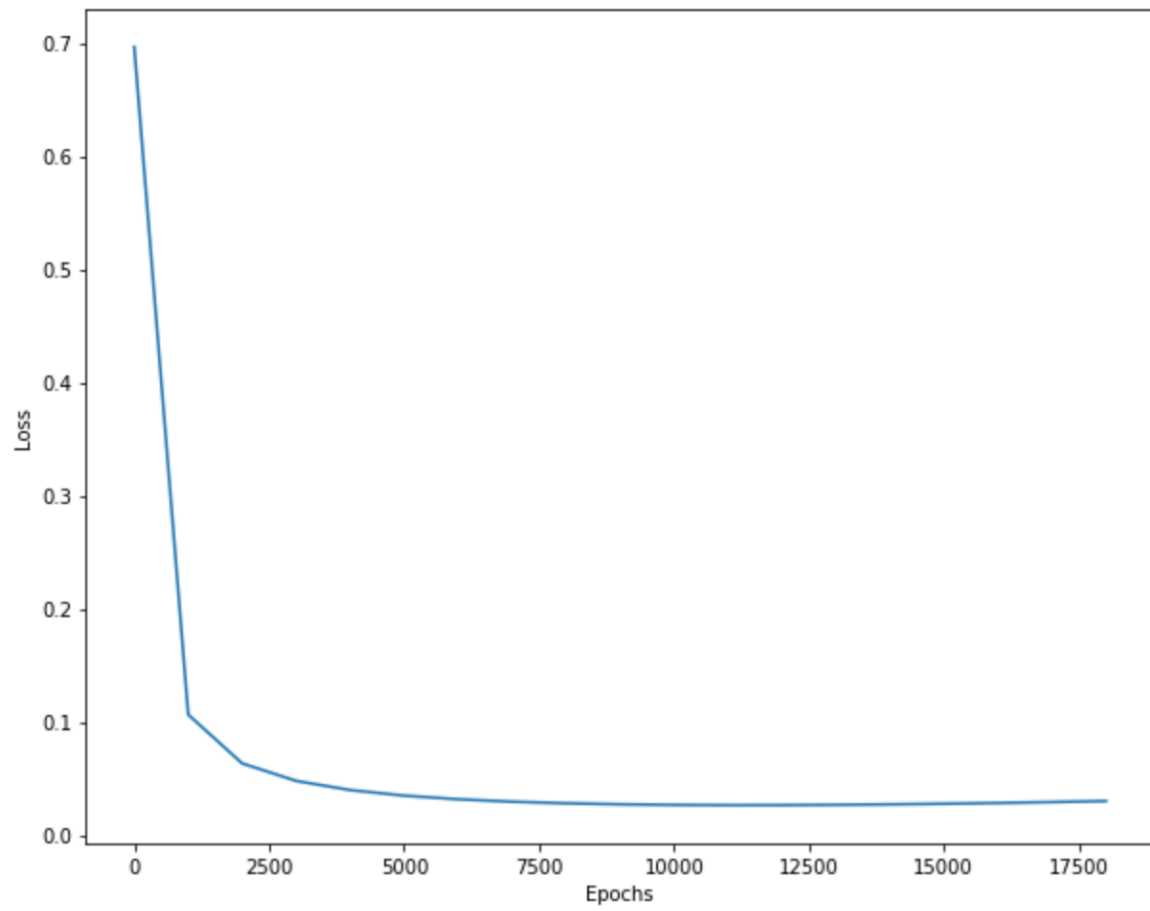


(200, 200)

Out[93]: (-1.9969999999999999, 1.98300000000000036)

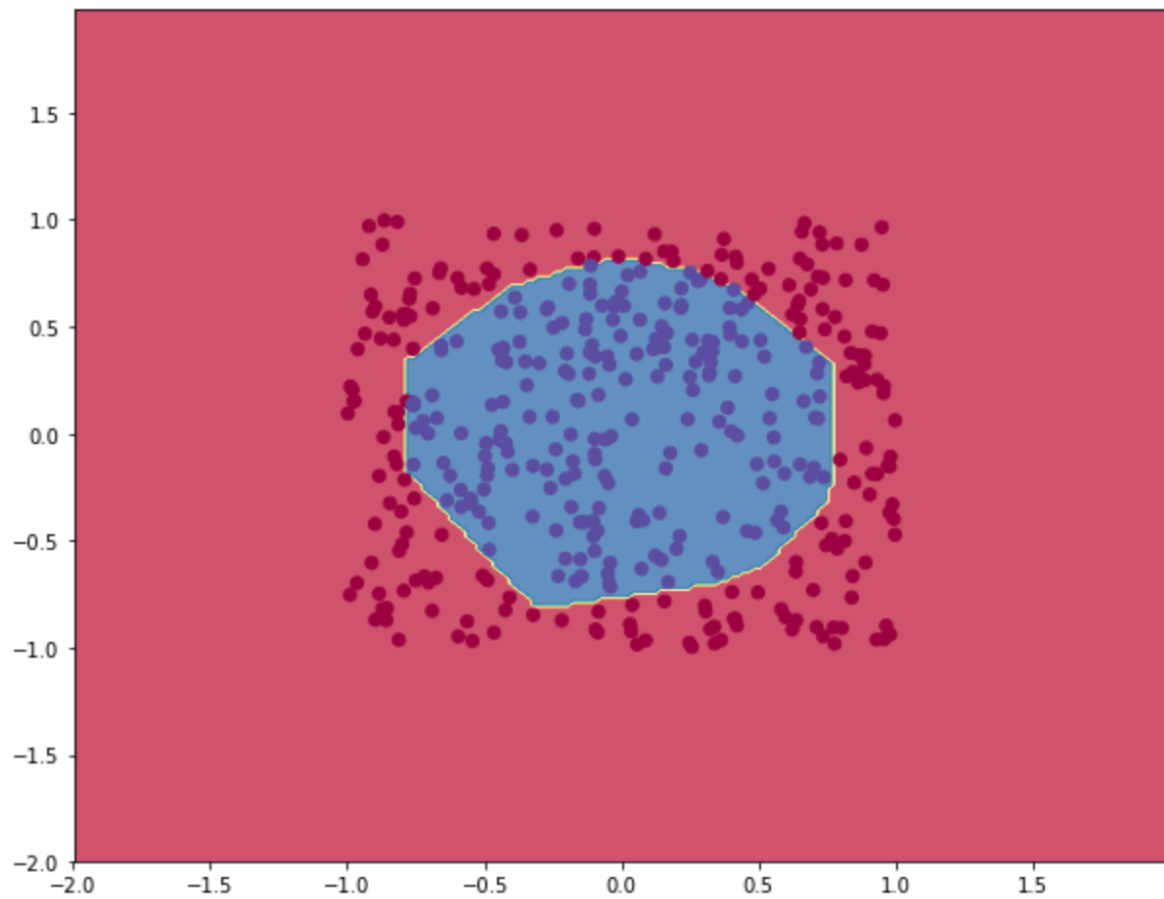


The Loss Vs epochs is shown below:

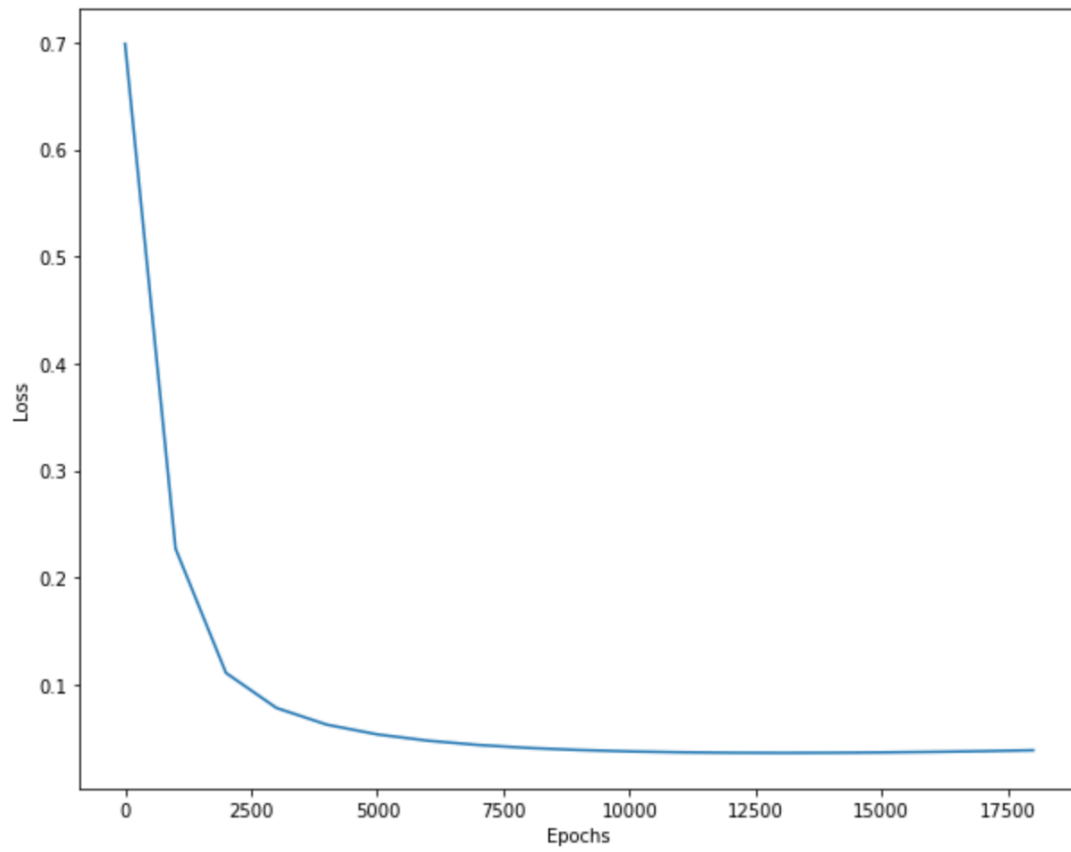


For the circle problem:

A NN with one hidden layer of 20 layers was sufficient to conquer the data set.

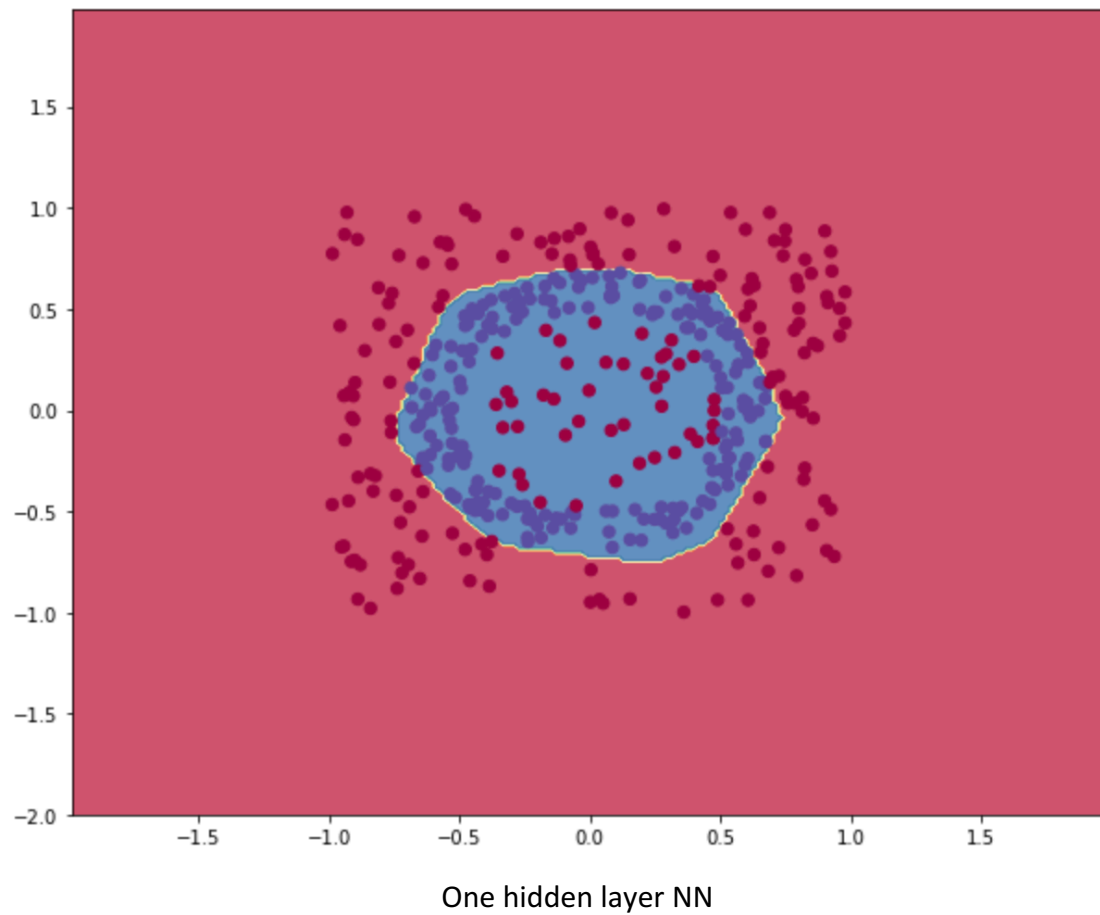


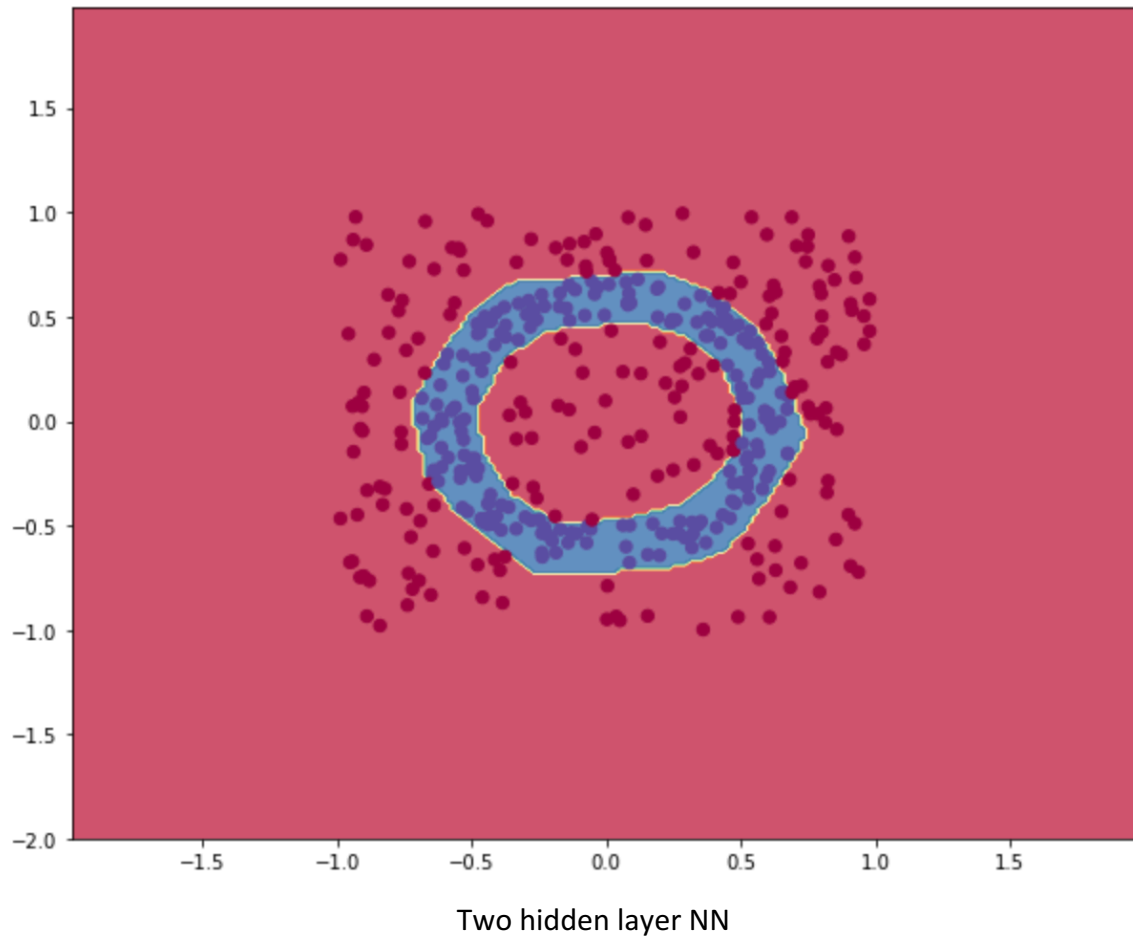
The Loss Vs epochs is shown below:



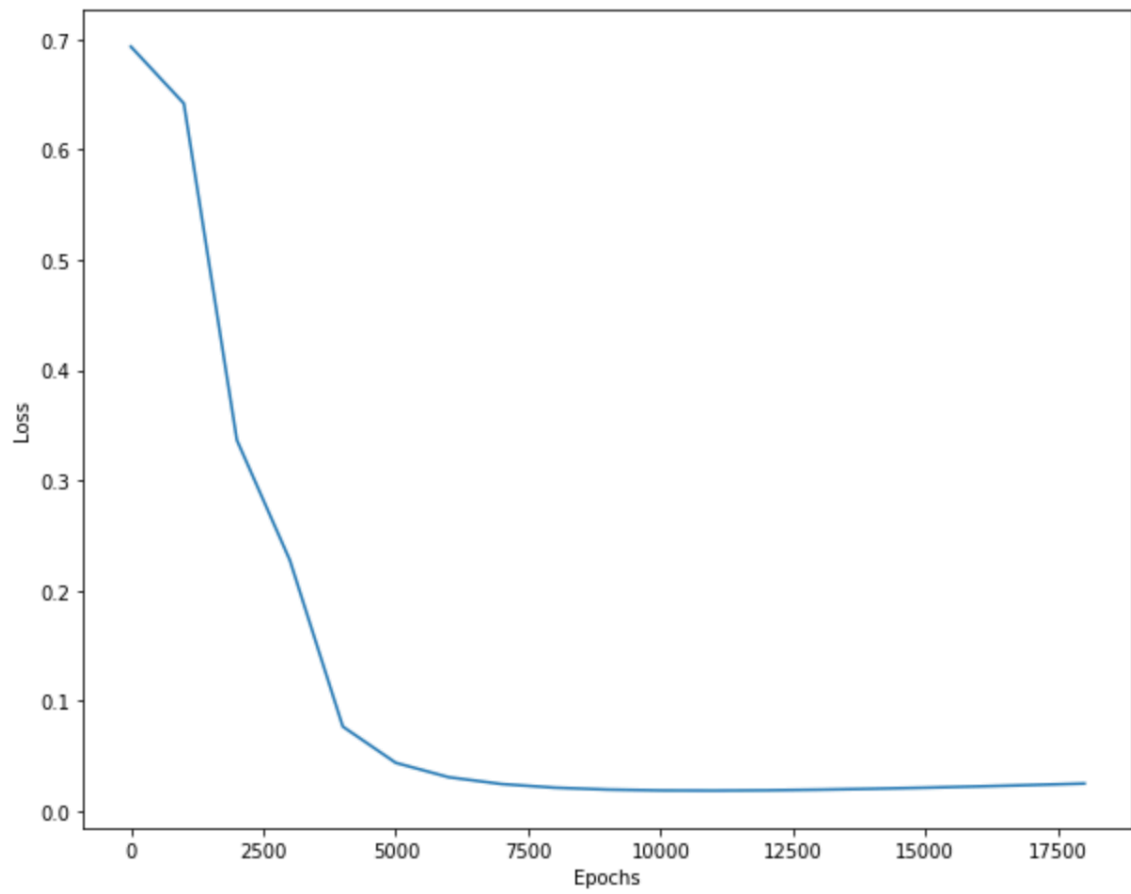
For the ring problem:

A neural Network with one hidden layer of 50 neurons was not sufficient to solve this problem. Thus, I used a neural Network with 2 hidden layers with 20 neurons in each. The following picture shows the outcome of using one hidden layer Vs Two hidden layers.



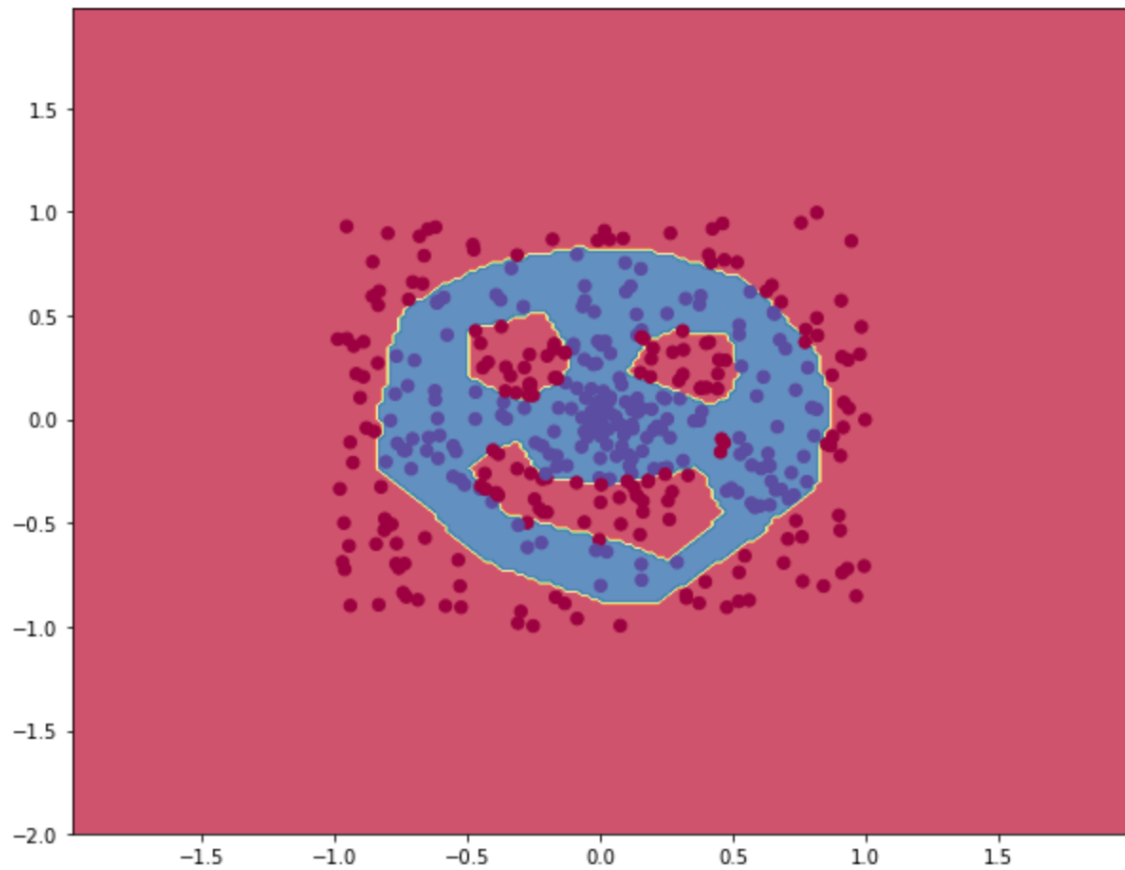


The Loss Vs epochs is shown below:

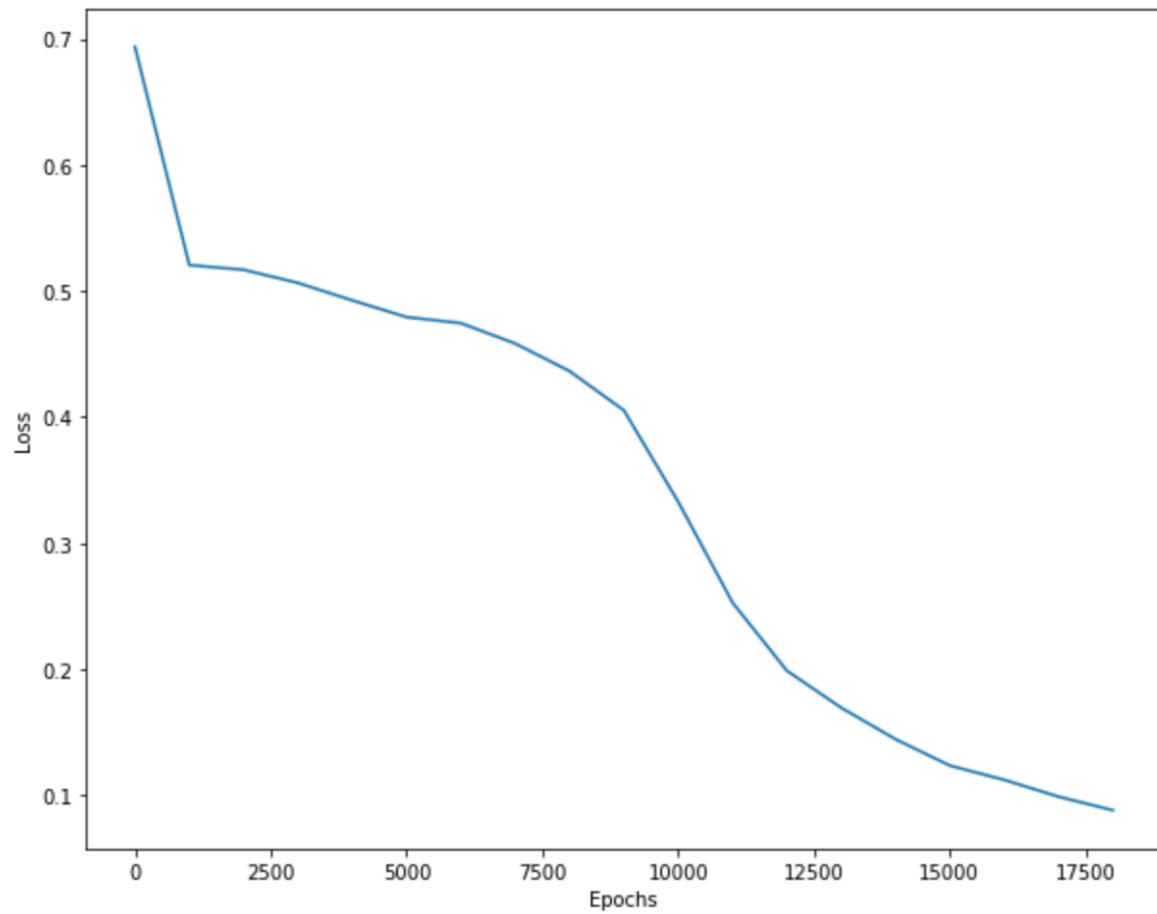


For the face problem:

This was the highest non linear problem in the assignment. I was able to achieve 97% accuracy with a 2 hidden layers each containing 25 neurons. The outcome of the NN is shown in the following figure



The Loss Vs epochs is shown below:



I was able to calculate MCCR for the face only, as it was the only problem that the NN was not able to achieve 100% classification accuracy.

MCCR1 = 97%

MCCR2 = 100%