1 Output Layer

Assumption: In the question, both the weights with respect to X1 and X2 are labelled as W1 in the diagram. Assuming the weight with respect to X2 input is W2.

1)

$$X = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \qquad W = \begin{bmatrix} 2 \\ -1 \end{bmatrix} \qquad b_1 = 1$$

The output $y_1 = \phi(W^T X + b_1)$

where ϕ is the sigmoid function and b_1 is bias.

Lets say $W^T X + b_1 = V$. While performing the matrix multiplication we can account for the bias term b_1 by appending ~~adding~~ it to the weight vector and appending 1 to the corresponding position in the vector X to get V.

Thus, $V = \begin{bmatrix} 1 & 2 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = 1 + 4 - 1 = 4$

$$y_1 = \phi(V) = \frac{1}{1 + e^{-V}}$$

$$= \frac{1}{1 + e^{-4}} \qquad [\because V = 4]$$

$$= \frac{1}{1.0183} = 0.982$$

2)

Derivation of the gradient for wi using the chain rule:

$$J = \frac{1}{2}\sum_{n=1}^{N}(d_n - y_n)^2 \quad \text{[Replacing subscript } i \text{ with } n\text{]}$$

$$\frac{\partial J}{\partial w_i} = \frac{\partial}{\partial w_i}\left[\frac{1}{2}\sum_{n=1}^{N}(d_n - y_n)^2\right]$$

$$= \frac{1}{2}\sum_{n=1}^{N}\frac{\partial}{\partial w_i}(d_n - y_n)^2$$

$$= \frac{1}{2}\sum_{n=1}^{N}2(d_n - y_n)\frac{\partial}{\partial w_i}(d_n - y_n)$$

$$= \sum_{n=1}^{N}(d_n - y_n)\left(-\frac{\partial}{\partial w_i}y_n\right)$$

Now, $y_n = \phi(V_n) = \phi(W^T X_n)$

$$\therefore \quad \frac{\partial J}{\partial w_i} = \sum_{n=1}^{N} -(d_n - y_n)\frac{\partial y_n}{\partial V_n}\cdot\frac{\partial V_n}{\partial w_i}$$

Since we have a Sigmoid activation function,

$$y_n = \frac{1}{1+e^{-\alpha V_n}}$$

$$\Rightarrow \quad \frac{\partial y_n}{\partial V_n} = \frac{\partial \phi(V_n)}{\partial V_n}$$

$$= \frac{\partial}{\partial V_n}\cdot\frac{1}{1+e^{-\alpha V_n}}$$

$$= \frac{(1+e^{-\alpha V_n}) - (1)(e^{-\alpha V_n})}{(1+e^{-\alpha V_n})^2}$$

$$= \frac{-\frac{\partial}{\partial V_n}(1+e^{-\alpha V_n})}{(1+e^{-\alpha V_n})^2}$$

$$= \frac{-e^{-\alpha V_n}(-\alpha)}{(1+e^{-\alpha V_n})^2}$$

$$= \frac{1}{1+e^{-\alpha V_n}} \cdot \frac{e^{-\alpha V_n}}{1+e^{-\alpha V_n}}$$

$$= y_n(1-y_n)$$

Using this value of $\frac{\partial y_n}{\partial V_n}$, we get,

$$\frac{\partial J}{\partial w_i} = \sum_{n=1}^{N} -(d_n - y_n)\, y_n (1-y_n) \frac{\partial V_n}{\partial w_i}$$

$$= \sum_{n=1}^{N} -(d_n - y_n)\, y_n (1-y_n) \frac{\partial}{\partial w_i} W^T X_n$$

$$= \sum_{n=1}^{N} -(d_n - y_n)\, y_n (1-y_n)\, x_{ni}$$

3)

Using the derived formula for gradient of w_i we get,

$$\frac{\partial J}{\partial w_1} = -(d_n - y_n)\, y_n\, (1-y_n)\, x_1$$

$$= -(1-0.982)(0.982)(1-0.982)(2) \quad [\because x_1 = 2,\ d_n = d = 1,\ y_1 = 0.982 \text{ (from question 1)}]$$

$$= -(0.000318)(2)$$

$$= -0.000636$$

$\therefore$ Using stochastic gradient descent,
The update for w_1, $w_1' = w_1 - \eta \frac{\partial J}{\partial w_1}$

$$= 2 - (1)(-0.000636) \quad [\because \eta = 1]$$

$$= 2.000636$$

Similarly $\frac{\partial J}{\partial w_2} = -(d_n - y_n)(y_n)(1-y_n)\, x_2$

$$= -(0.000318)(1) \quad [\because x_2 = 1,\ d_n = d = 1,\ y_1 = 0.982]$$

$$= -0.000318$$

$\therefore$, The update for w_2, $w_2' = w_2 - \eta \frac{\partial J}{\partial w_2}$

$$= (-1) - (1)(-0.000318) \quad [\because \eta = 1]$$

$$= -1 + 0.000318$$

$$= -0.99968$$

For the bias b_1, we will have the $x = 1$

∴ the gradient will be $= -(d_n - y_n)\, y_n (1 - y_n)(1)$

$$= -(0.000318)(1) \quad [\because d_n = d = 1,\ y_n = y_1 = 0.982]$$

$$= -0.000318$$

∴ The update for b_1, $b_1' = b_1 - \eta(-0.000318)$

$$= 1 - (1)(-0.000318) \quad [\because b_1 = 1,\ \eta = 1]$$

$$= 1 + 0.000318$$

$$= 1.000318$$

2 Single Hidden Layer

1)

Like we computed for part 1, for the 1st neuron in the hidden layer,

$$V_1 = W^T X + b_1$$

$$= \begin{bmatrix} b_1 & W_1 & W_3 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$$

[appending the bias term to weight vector and 1 to corresponding position in Vector X]

$$= \begin{bmatrix} 3 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$$

$$= 3 + 1 - 2 = 2$$

Similarly for the 2nd neuron in the hidden layer,

$$V_2 = W^T X + b_2$$

$$= \begin{bmatrix} b_2 & W_2 & W_4 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$$

$$= \begin{bmatrix} 4 & -1 & -2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$$

$$= 4 - 1 + 2 = 5$$

∴ output from the 1st neuron in hidden layer will be $= \phi(V_1) = \dfrac{1}{1+e^{-V_1}}$ [∵ we are using a sigmoid activation function]

$$= \frac{1}{1+e^{-2}} = 0.8808$$

Similarly, output from the 2nd neuron in the hidden layer $= \phi(V_2) = \dfrac{1}{1+e^{-V_2}}$

$$= \frac{1}{1+e^{-5}}$$

$$= 0.9933$$

$\therefore$ For the neuron in the output layer,

$$V = W^T X + b_3$$

$$= \begin{bmatrix} b_3 & W_5 & W_6 \end{bmatrix} \begin{bmatrix} 1 \\ \phi(V_1) \\ \phi(V_2) \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0.8808 \\ 0.9933 \end{bmatrix}$$

$$= 0 - 0.8808 + 0.9933$$

$$= 0.1125$$

$$\therefore \; y_1 = \phi(V) = \frac{1}{1 + e^{-0.1125}}$$

$$= \frac{1}{1.893597}$$

$$= 0.5281$$

2)

Derivation of the gradient for wj using the chain rule:

$$J = \frac{1}{2}\sum_{i=1}^{N} (d_i - y_i)^2$$

$$\Rightarrow \frac{\partial J}{\partial \omega_{jl}} = \frac{\partial J}{\partial e_l} \cdot \frac{\partial e_l}{\partial y_l} \cdot \frac{\partial y_l}{\partial V_l} \cdot \frac{\partial V_l}{\partial \omega_{jl}}$$

$$= [e_l][-1][\phi'(V_l)][y_l]$$

A local gradient $\delta_l = -\frac{\partial J}{\partial V_l} = e_l \phi'(V_l)$

Similarly, $\delta_n = -\frac{\partial J}{\partial y_n} \cdot \frac{\partial y_n}{\partial V_n}$

$$= -\frac{\partial J}{\partial y_n} \cdot \phi'(V_n)$$

$$= -\left[\sum_n e_m [-\phi'(V_m)][\omega_{jn}]\right] \phi'(V_n)$$

$$= \phi'(V_n) \sum_m \delta_m \omega_{jn}$$

$\therefore$ The gradient at a hidden neuron in terms of the local gradient $= \delta\omega_{jn} = \eta\, \delta_n y_j$

3)

$$W_5' = W_5 - \eta \frac{\partial J}{\partial W_5}$$

~~From previous derivation~~

Using chain rule,

$$\frac{\partial J}{\partial W_5} = \frac{\partial J}{\partial \phi_1} \cdot \frac{\partial \phi_1}{\partial V_1} \cdot \frac{\partial V_1}{\partial W_5} \quad \text{—— (1)}$$

Now, $$\frac{\partial J}{\partial \phi_1} = \frac{\partial}{\partial \phi_1}\left(\frac{1}{2}(d - y_1)^2\right) \qquad [\phi_1 = y_1]$$

$$= -(d - y_1)$$

$$= -(0 - 0.5281) = 0.5281$$

$$\frac{\partial \phi_1}{\partial V_1} = \frac{\partial}{\partial V_1}\left(1/1+e^{-V_1}\right) \qquad [\phi_1 = y_1]$$

$$= y_1(1 - y_1)$$

$$= (0.5281)(1 - 0.5281)$$

$$= 0.2492$$

$$\frac{\partial V_1}{\partial W_5} = \frac{\partial}{\partial W_5}\left(W_5\phi_2 + W_6\phi_3 + b_3\right)$$

$$= \phi_2 = 0.8808$$ [Calculated in the 1st question]

$$\therefore \quad W_5' = (-1) - (1)\left(0.5281 \times 0.2492 \times 0.8808\right)$$

$$= -1 - 0.1159$$

$$= -1.1159$$

$$W_1' = W_1 - \eta \frac{\partial J}{\partial W_1}$$

Now, $$\frac{\partial J}{\partial W_1} = \frac{\partial J}{\partial \phi_2} \cdot \frac{\partial \phi_2}{\partial V_2} \cdot \frac{\partial V_2}{\partial W_1}$$

we will have $$\frac{\partial J}{\partial \phi_2} = \frac{\partial J}{\partial V_1} \cdot \frac{\partial V_1}{\partial \phi_2}$$

$$= \frac{\partial J}{\partial \phi_1} \cdot \frac{\partial \phi_1}{\partial V_1} \cdot \frac{\partial V_1}{\partial \phi_2}$$

$$= (0.5281)(0.2492)\frac{\partial}{\partial \phi_2}\left(W_5\phi_2 + W_6\phi_3 + b_3\right)$$

$$= (0.1316)(W_5) = -0.1316$$

$$\frac{\partial \phi_2}{\partial V_2} = \frac{\partial}{\partial V_2}\left(1/1+e^{-V_2} \right)$$

$$= \phi_2 (1-\phi_2)$$

$$= (0.8808)(1-0.8808)$$

$$= 0.10499$$

$$\frac{\partial V_2}{\partial W_1} = \frac{\partial}{\partial W_1}(w_1 x_1 + W_3 x_2 + b_1)$$

$$= x_1 = 1$$

$$\therefore W_1' = W_1 - \eta\left((-0.1316)\cdot(0.10499)\cdot 1 \right)$$

$$= 1 + 1(0.0138) = 1.0138$$

Similarly,

$$W_2' = W_2 - \eta \frac{\partial J}{\partial W_2}$$

Now, $\frac{\partial J}{\partial W_2} = \frac{\partial J}{\partial \phi_3} \cdot \frac{\partial \phi_3}{\partial V_3} \cdot \frac{\partial V_3}{\partial W_2}$

we have, $\frac{\partial J}{\partial \phi_3} = \frac{\partial J}{\partial V_1} \cdot \frac{\partial V_1}{\partial \phi_3}$

$$= \frac{\partial J}{\partial \phi_1} \cdot \frac{\partial \phi_1}{\partial V_1} \cdot \frac{\partial V_1}{\partial \phi_3}$$

$$\Rightarrow \frac{\partial J}{\partial \phi_3} = (0.1316) \cdot \frac{\partial V_1}{\partial \phi_3}$$

$$= (0.1316) \frac{\partial}{\partial \phi_3}(W_5 \phi_2 + W_6 \phi_3 + b_3)$$

$$= (0.1316) \; W_6$$

$$= 0.1316 \qquad [\because W_6 = 1]$$

$$\frac{\partial \phi_3}{\partial V_3} = \phi_3 (1 - \phi_3)$$

$$= (0.9933)(1 - 0.9933) \qquad \text{[Calculated in question 1]}$$

$$= 0.006655$$

$$\frac{\partial V_3}{\partial W_2} = \frac{\partial}{W_2}(W_2 x_1 + W_4 x_2 + b_2)$$

$$= x_1 = 1$$

$$\therefore W_2' = W_2 - \eta(0.1316 \times 0.006655 \times 1)$$

$$= -1 - 1(0.000876)$$

$$= -1.000876$$

3 UF Network

1)

Number of units in the first hidden layer = 3
Number of units in the second hidden layer = 2

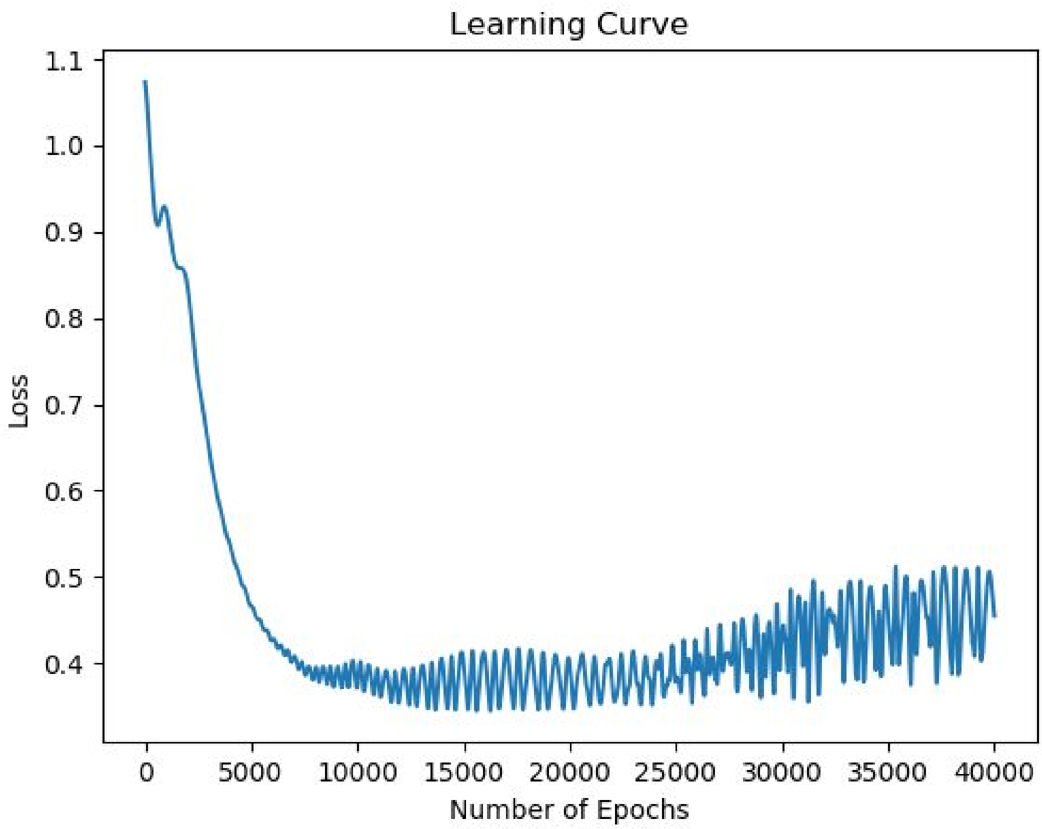
There is no general rule of thumb as to the exact number of units required for a neural network. But for the given dataset, 3 neurons in the first hidden layer and 2 neurons in the second hidden layer gives the best results. Having too many units in the hidden layer leads to overfitting. If we start off with a minimal number of units and keep increasing the units in the hidden layers gradually, we see the Loss keeps decreasing till a certain point. After that if we further increase the number of units the Loss value increases. The number of nodes at this point which gives the minimum Loss value is the optimal number of units needed in the hidden layers.

2)

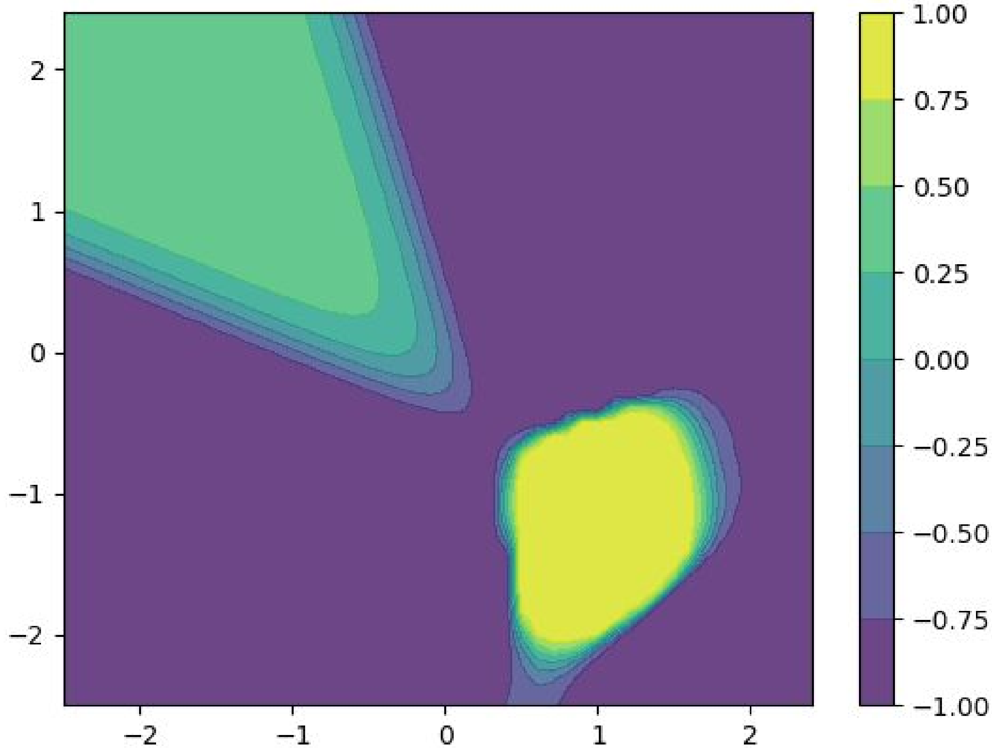
Combination 1:

Learning rate = 0.0002
Epochs = 40000

Learning Curve:



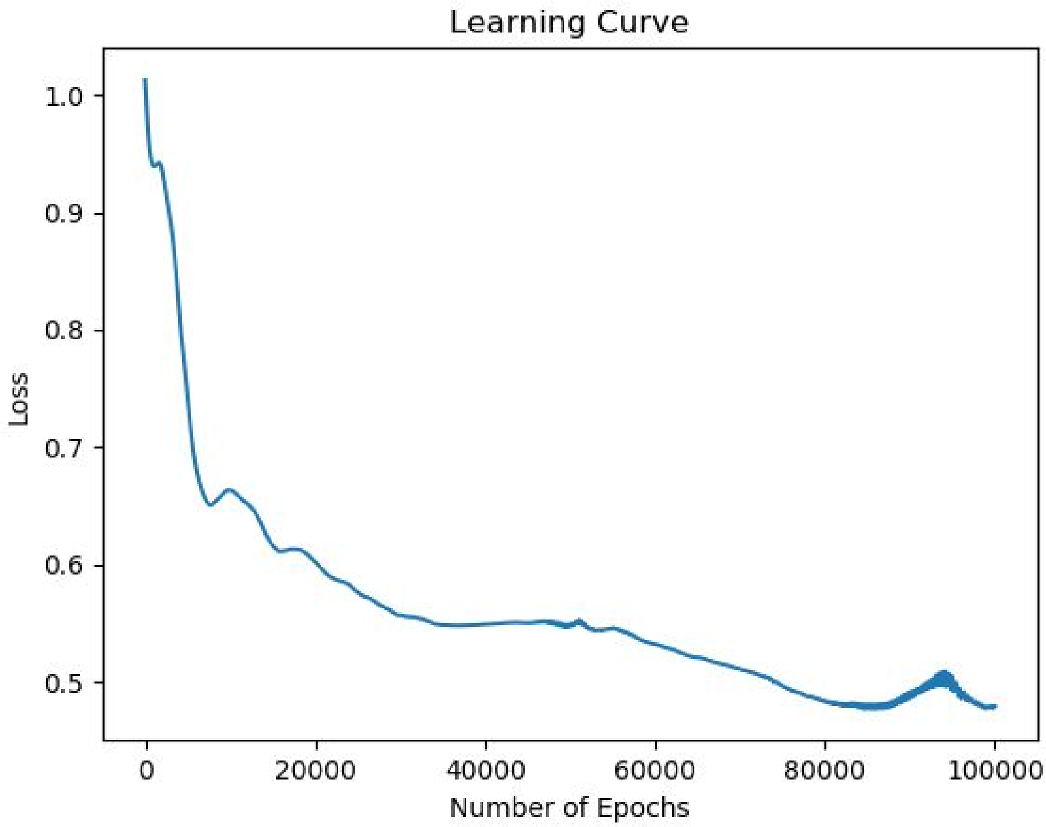
Decision Boundary:



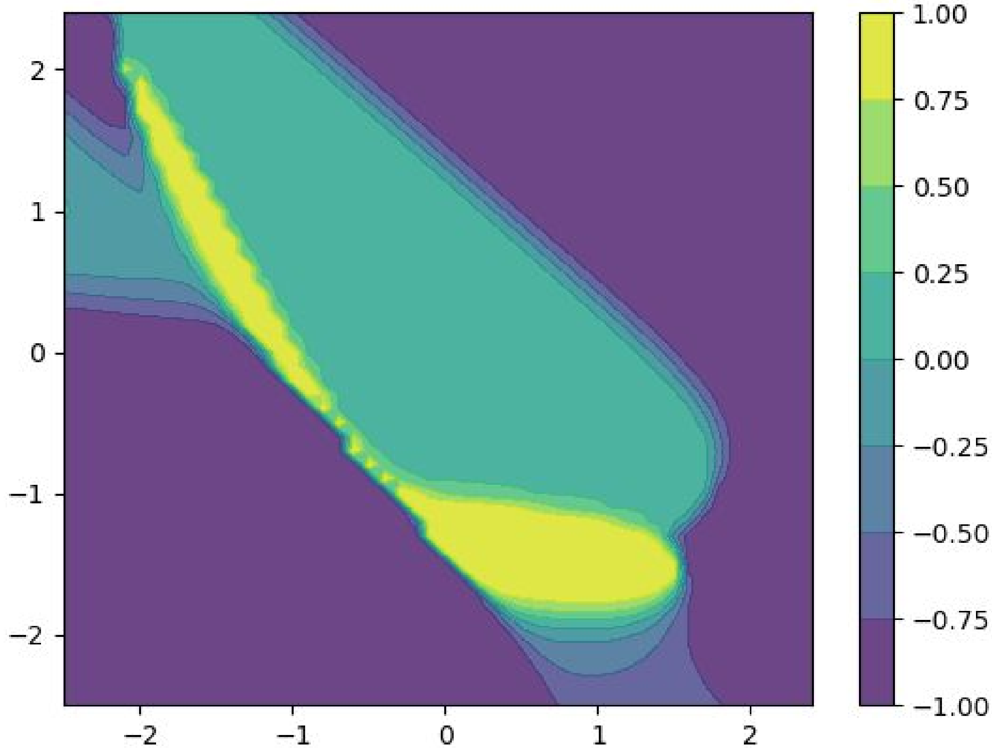
Combination 2:

Epochs = 100000
Learning rate = 0.000085

Learning Curve:



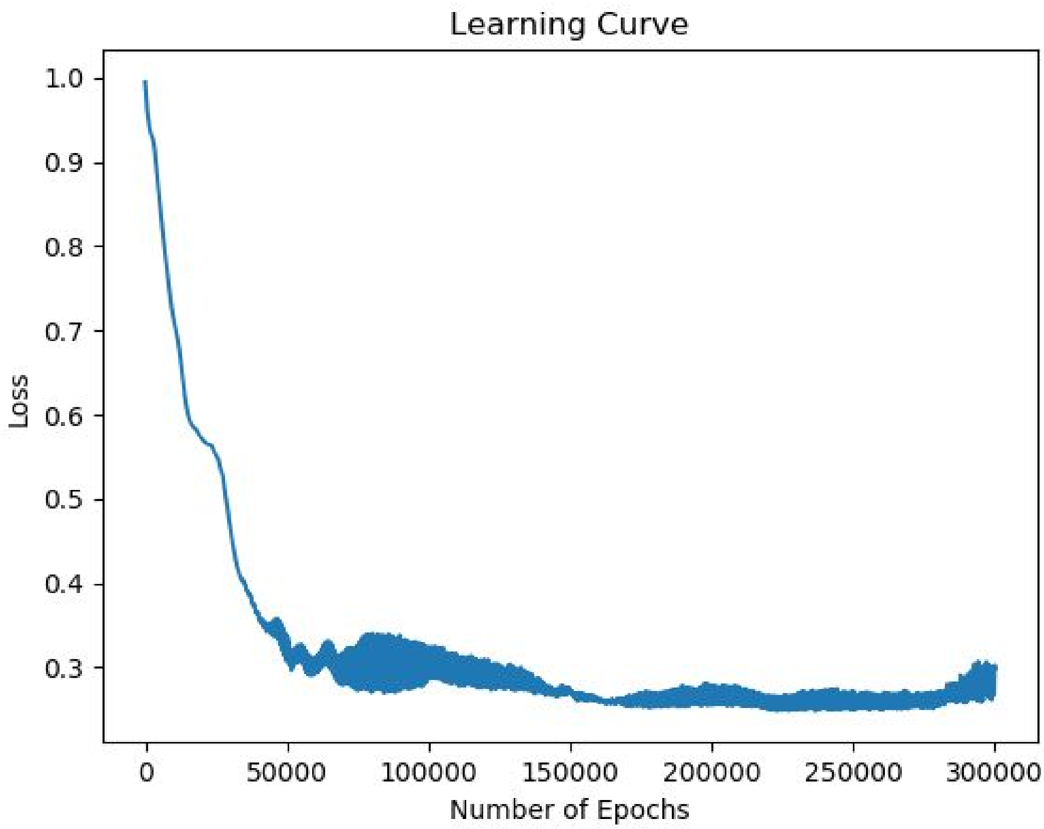
Decision Boundary:



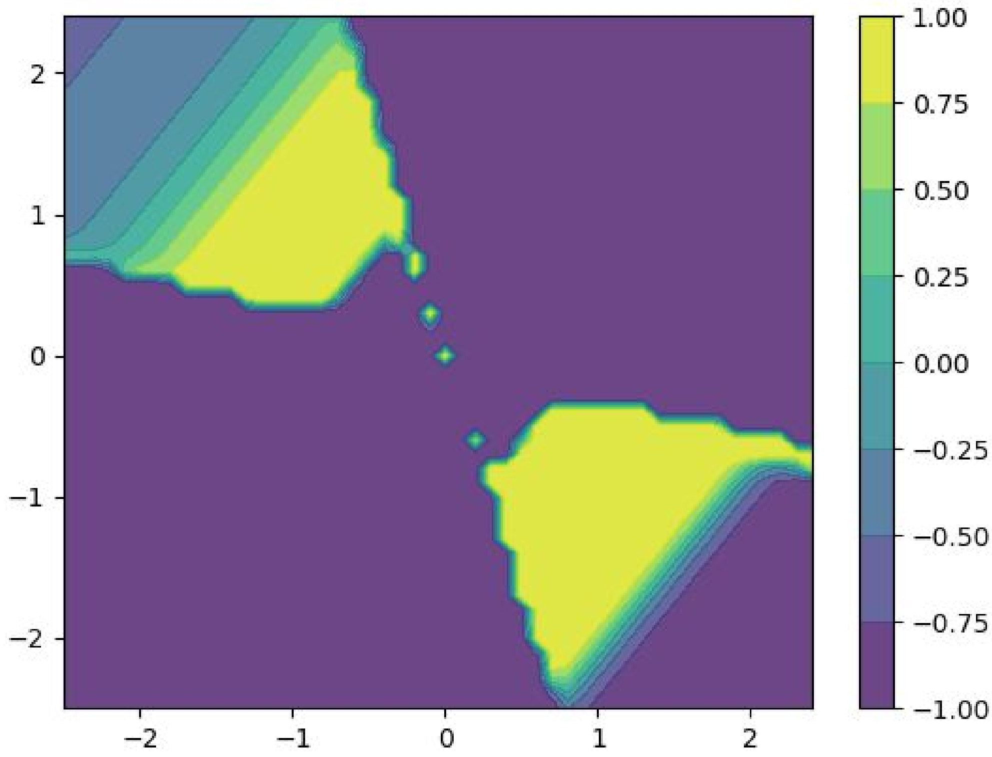
Combination 3:

epochs = 300000
Learning rate = 0.000057

Learning Curve:



Decision Boundary:



3)

a)

Having less number of epochs is resulting in underfitting while having too many epochs pose the risk overfitting. An optimal number of epochs is which gives us the least Loss value with an optimal learning rate. With less than optimal number of epochs, even with an ideal learning rate we might not get the desired weights for our network. In this case 300000 is the optimal number of epochs which give the least Loss value of 0.298893 with learning rate of 0.000057

Having a very high learning rate leads to unstable learning. As we can see from the learning curves with high learning rates, the graph is not as smooth as the ones with low learning rates and it tends to have more spikes. Although it might reach an optima faster, it can completely jump over an optima which might well be the global optima and eventually give a greater error. On the other hand having a low learning rate takes a lot of time to reach an optima. However, it gets there steadily. This prevents it from jumping over an optima. If we have the right number of epochs, a low learning rate will eventually reach the optima. But then again, it runs the risk of getting stuck in a local optima.

b)

The selected values of learning rate 0.000057 and number of epochs 300000 in combination 3 was ideal in this case. These values worked well because the learning rate is not too high so that it gives unstable results and not too low so that the model remains under-trained with the selected number of epochs. It is in optimal value where the model learns steadily provides the least Loss value. Also, with more number of epochs than this, we overfit the data while with lesser number of epochs we underfit. Thus, this is an ideal combination which gave us the least Loss value out of all the combinations.