Forward Propagation

Course Title: Artificial Intelligence

Course Code: cse-403

Khandaker Jannatul Ritu, Lecturer, Dept. of CSE [BAIUST]

Outlines:

1. Introduction to Forward Propagation in Neural Networks
2. Mathematical Explanation of Forward Propagation
3. Steps for Forward Propagation in Neural Network
4. Solve equation by apply forward propagation
   Ex:1,2,3,4,5,6,7,8,9,10,11,12
5. Solve Problems by apply forward propagation
   Problem:1,2,3,4,5,6,7,8,9,10,11,12

What is Forward Propagation in Neural Networks?

Feedforward neural networks stand as foundational architectures in deep learning. Neural networks consist of an input layer, at least one hidden layer, and an output layer. Each node is connected to nodes in the preceding and succeeding layers with corresponding weights and thresholds. In this article, we will explore what is forward propagation and it's working.

What is Forward Propagation in Neural Networks?

Forward propagation is the process in a neural network where the input data is passed through the network's layers to generate an output. It involves the following steps:

1. **Input Layer:** The input data is fed into the input layer of the neural network.
2. **Hidden Layers:** The input data is processed through one or more hidden layers. Each neuron in a hidden layer receives inputs from the previous layer, applies an activation function to the weighted sum of these inputs, and passes the result to the next layer.
3. **Output Layer:** The processed data moves through the output layer, where the final output of the network is generated. The output layer typically applies an activation function suitable for the task, such as softmax for classification or linear activation for regression.
4. **Prediction:** The final output of the network is the prediction or classification result for the input data.

Why Feed-forward network?

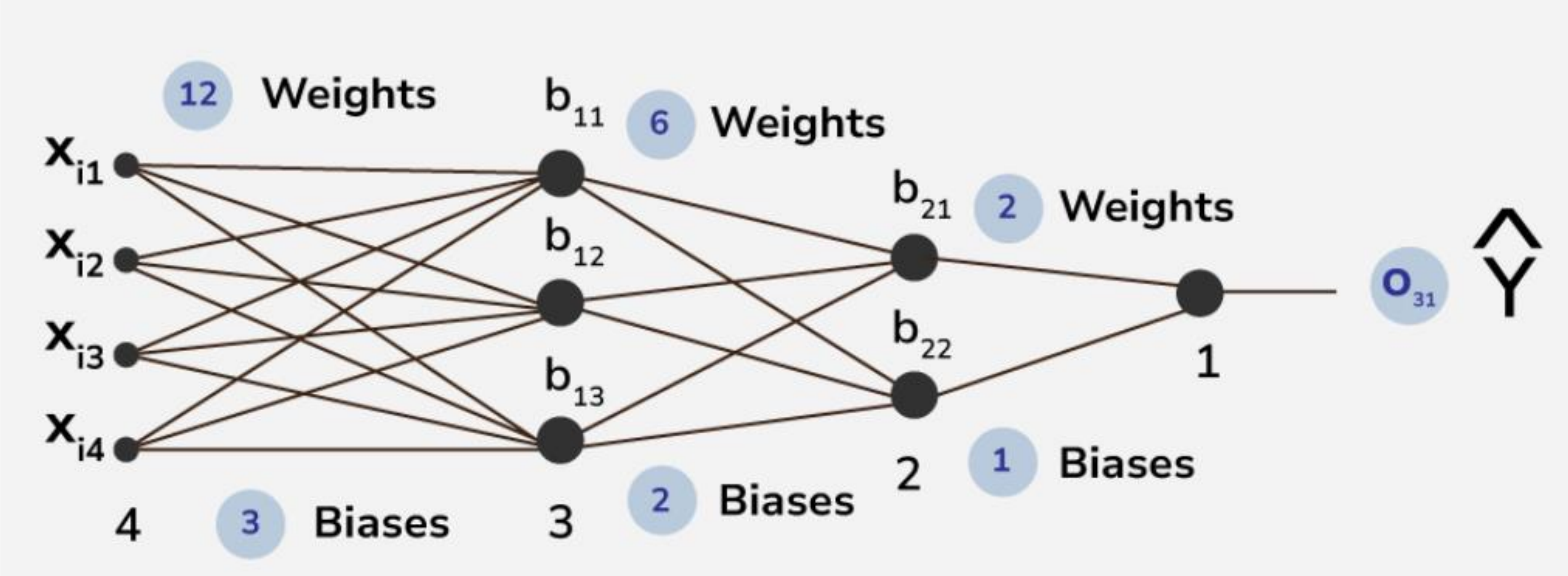
In order to generate some output, the input data should be fed in the forward direction only. The data should not flow in reverse direction during output generation otherwise it would form a cycle and the output could never be generated. Such network configurations are known as ***feed-forward network***. The *feed-forward network* helps in *forward propagation*.

Forward propagation is essential for making predictions in neural networks. It calculates the output of the network for a given input based on the current values of the weights and biases. The output is then compared to the actual target value to calculate the loss, which is used to update the weights and biases during the training process.

Mathematical Explanation of Forward Propagation

•In the above picture , the first layer of $Xi1,Xi2,Xi3,Xi4$Xi1,Xi2,Xi3,Xi4 are the input layer and last layer of $b31$b31 is the output layer .Other layers are hidden layers in this structure of ANN . this is a 4 layered deep ANN where first hidden layer consists of 3 neuron and second layer consists of 2 neuron .
•There are total 26 trainable parameters . here , in hidden layer 1 , the top to bottom biases are $b11,b12,b13$b11,b12,b13 and in hidden layer 2 , the top to bottom biases are $b21,b22$b21,b22 . the output layer contains the neuron having bias b31 . likewise the weights of corresponding connections are assigned like $W111,W112,W113,W121,W122$W111,W112,W113,W121,W122etc.
•Here, considering we are using sigmoid function as the activation function

The output prediction function is: $\sigma(w\mathrm{T}x+b)$σ(wTx+b)



Inside Layer 1

Here, from the previous output function , weights and biases are segmented as matrices and the dot product has been done along with matrix manipulation .

The procedure of the matrix operations for layer 1 are as follows:

The procedure of the matrix operations for layer 1 are as follows:

$$\sigma\left(\begin{bmatrix} W_{111} & W_{112} & W_{113} \\ W_{121} & W_{122} & W_{123} \\ W_{131} & W_{132} & W_{133} \\ W_{141} & W_{142} & W_{143} \end{bmatrix}^T \begin{bmatrix} X_{i1} \\ X_{i2} \\ X_{i3} \\ X_{i4} \end{bmatrix} + \begin{bmatrix} b_{11} \\ b_{12} \\ b_{13} \end{bmatrix}\right) = \sigma\left(\begin{bmatrix} W_{111}X_{i1} + W_{121}X_{i2} + W_{131}X_{i3} + W_{141}X_{i4} \\ W_{112}X_{i1} + W_{122}X_{i2} + W_{132}X_{i3} + W_{142}X_{i4} \\ W_{113}X_{i1} + W_{123}X_{i2} + W_{133}X_{i3} + W_{143}X_{i4} \end{bmatrix} + \begin{bmatrix} b_{11} \\ b_{12} \\ b_{13} \end{bmatrix}\right) :$$

$$\sigma\left(\begin{bmatrix} W_{111}X_{i1} + W_{121}X_{i2} + W_{131}X_{i3} + W_{141}X_{i4} + b_{11} \\ W_{112}X_{i1} + W_{122}X_{i2} + W_{132}X_{i3} + W_{142}X_{i4} + b_{12} \\ W_{113}X_{i1} + W_{123}X_{i2} + W_{133}X_{i3} + W_{143}X_{i4} + b_{13} \end{bmatrix}\right) = \begin{bmatrix} O_{11} \\ O_{12} \\ O_{13} \end{bmatrix} = a^{[1]}$$

After the completion of the matrix operations, the one-dimensional matrix has been formed as $[O11O12O13]$[O11O12O13]which is continued as $a^{[1]}$ known as the output of the first hidden layer.

Inside Layer 2

The output of the first hidden layer and the weights of the second hidden layer will be computed under the previously stated prediction function which will be matrix manipulated with those biases to give us the output of the second hidden layer i.e. the matrix $[O_{21}, O_{22}]$.The output will be stated as $a^{[2]}$here:

$$\sigma\left(\begin{bmatrix} W_{211} & W_{212} \\ W_{221} & W_{222} \\ W_{231} & W_{232} \end{bmatrix}^T \cdot \begin{bmatrix} O_{11} \\ O_{12} \\ O_{13} \end{bmatrix} + \begin{bmatrix} b_{21} \\ b_{22} \end{bmatrix}\right)$$
$$= \sigma\left(\begin{bmatrix} W_{211} \cdot O_{11} + W_{221} \cdot O_{12} + W_{231} \cdot O_{13} \\ W_{212} \cdot O_{11} + W_{222} \cdot O_{12} + W_{232} \cdot O_{13} \end{bmatrix} + \begin{bmatrix} b_{21} \\ b_{22} \end{bmatrix}\right)$$
$$= \sigma\left(\begin{bmatrix} W_{211} \cdot O_{11} + W_{221} \cdot O_{12} + W_{231} \cdot O_{13} + b_{21} \\ W_{212} \cdot O_{11} + W_{222} \cdot O_{12} + W_{232} \cdot O_{13} + b_{22} \end{bmatrix}\right)$$
$$= \begin{bmatrix} O_{21} \\ O_{22} \end{bmatrix}$$
$$= a^{[2]}$$

Inside Output Layer

$$\sigma\left(\left[\begin{array}{c} W_{311} \\ W_{321} \end{array}\right]^T \cdot \left[\begin{array}{c} O_{21} \\ O_{22} \end{array}\right] + [b_{31}]\right)$$
$$= \sigma\left([W_{311} \cdot O_{21} + W_{321} \cdot O_{22}] + [b_{31}]\right)$$
$$= \sigma\left(W_{311} \cdot O_{21} + W_{321} \cdot O_{22} + b_{31}\right)$$
$$= Y^i = [O_{31}]$$
$$= a^{[3]}$$

like in first and second layer , here also matrix manipulation along with the prediction function takes action and present us a prediction as output for that particular guessed trainable parameters in form of $[O_{31}]$which is termed as $a^{[3]} = Y^i$ here.

Now as for the complete equation , we can conclude as follows :

$\sigma(a[0]*W[1]+b[1])$ = $a[1]\sigma(a[1]*W[2]+b[2])$ = $a[2]\sigma(a[2]*W[3]+b[3])$=$a[3]$

which concludes us to :

$\sigma(\sigma((\sigma(a[0]*W[1]+b[1])*W[2]+b[2])*W[3]+b[3])$ = $a[3]$

Steps for Forward Propagation in Neural Network

The steps can be described at once as:

Step 1: Parameters Initialization

•We will first initialize the weight matrices and the bias vectors. It's important to note that we shouldn't initialize all the parameters to zero because doing so will lead the gradients to be equal and on each iteration the output would be the same, and the learning algorithm won't learn anything.

•Therefore, it's important to randomly initialize the parameters to a value between 0 and 1. The learning rate will be recommended as 0.01 to make the activation function active.

Step 2: Activation function

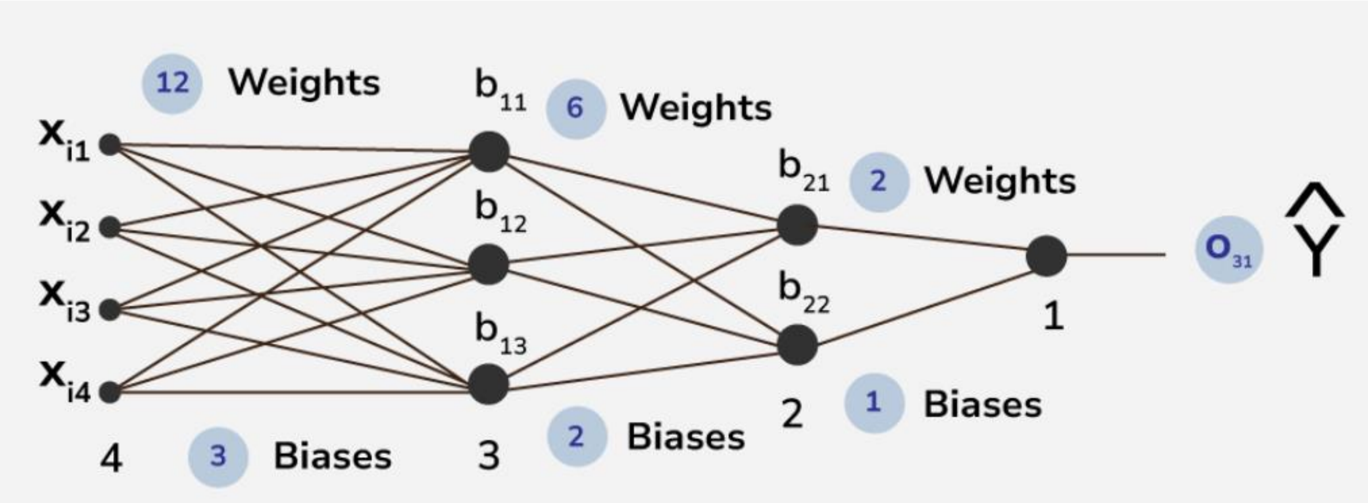
There is no definitive guide for which activation function works best on specific problems, based on specific requirements activation function must be chosen.

Step 3: Feeding forward

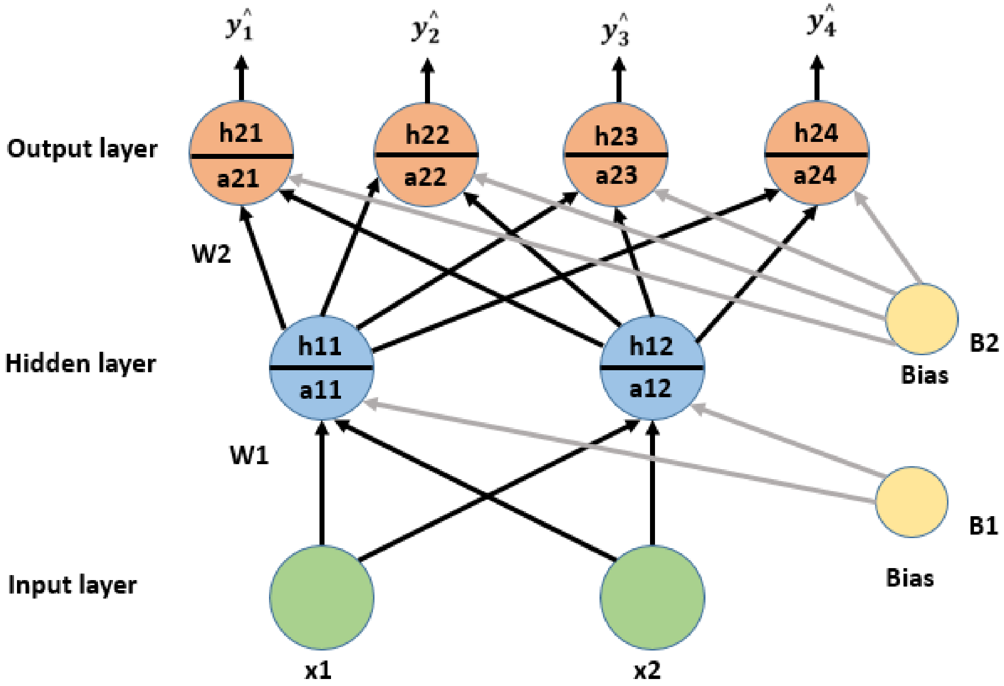
Given inputs from the previous layer, each unit computes an affine transformation $z=WTx+b$z=WTx+b and then applies an activation function $g(z)$g(z)such as ReLU element-wise. During this process, we will store all the variables computed and used on each layer to be used in backpropagation.

Solve equation by apply forward propagation:

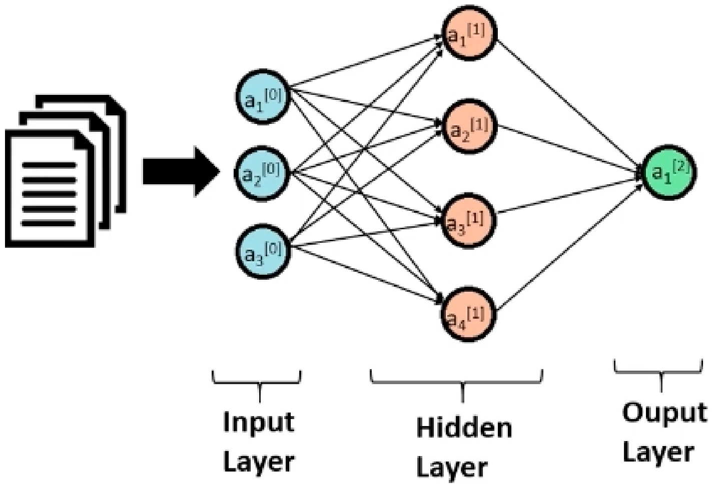
Ex-1: Obtain the output of the neuron Y



Ex-2: Obtain the output of the neuron Y



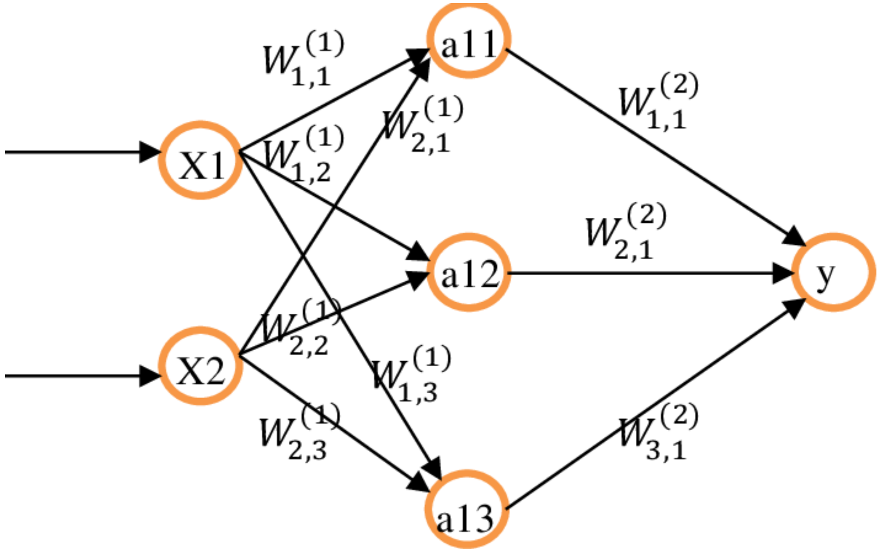
Ex-3: Obtain the output of the neuron Y



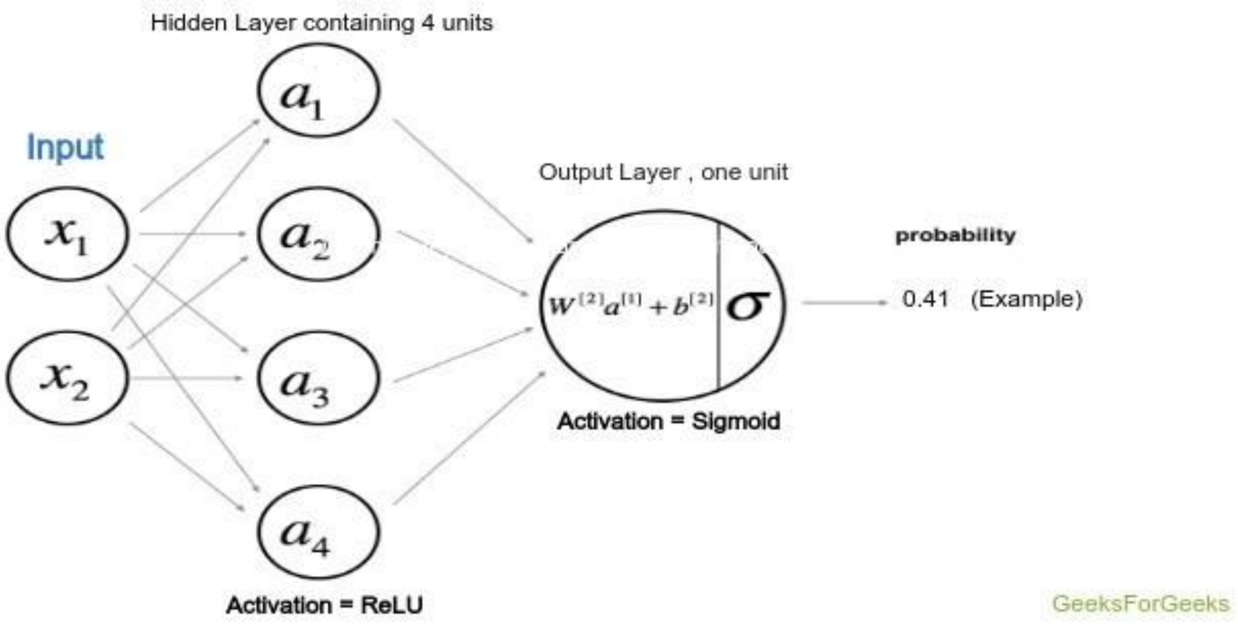
$a_1^{[1]}$ = activation_function($W_{11}^{[1]} * a_1^{[0]} + W_{12}^{[1]} * a_2^{[0]} + W_{13}^{[1]} * a_3^{[1]} + B1$)

$a_2^{[1]}$ = activation_function($W_{21}^{[1]} * a_1^{[1]} + W_{22}^{[1]} * a_2^{[1]} + W_{23}^{[1]} * a_3^{[1]} + B1$)

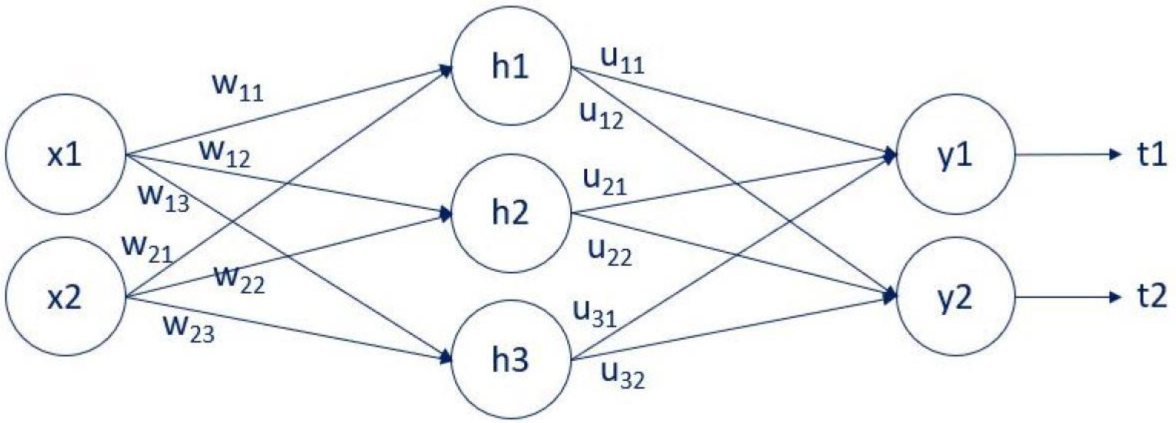
Ex-4: Obtain the output of the neuron Y



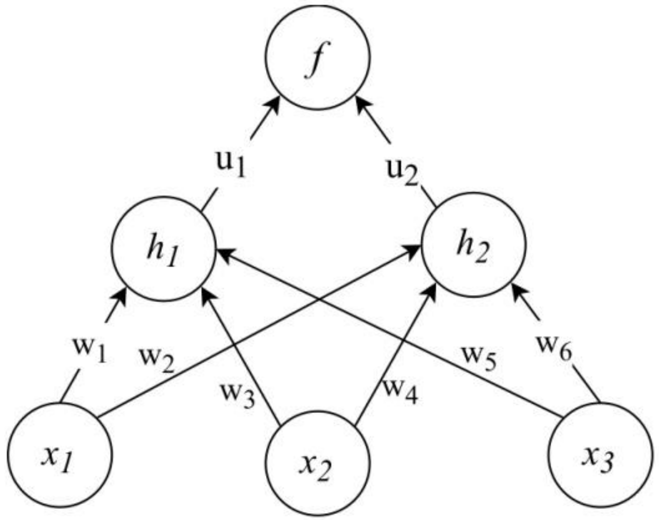
Ex-5: Obtain the output of the neuron Y



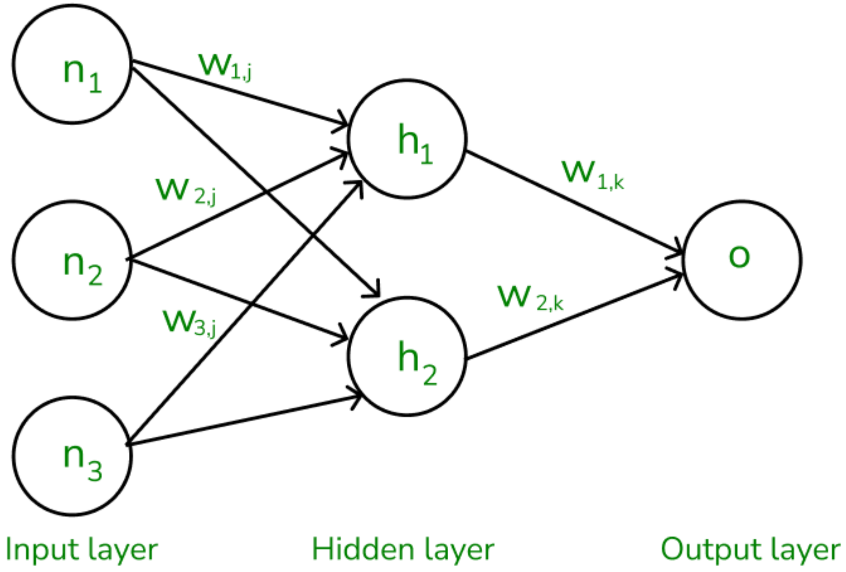
Ex-6: Obtain the output of the neuron Y



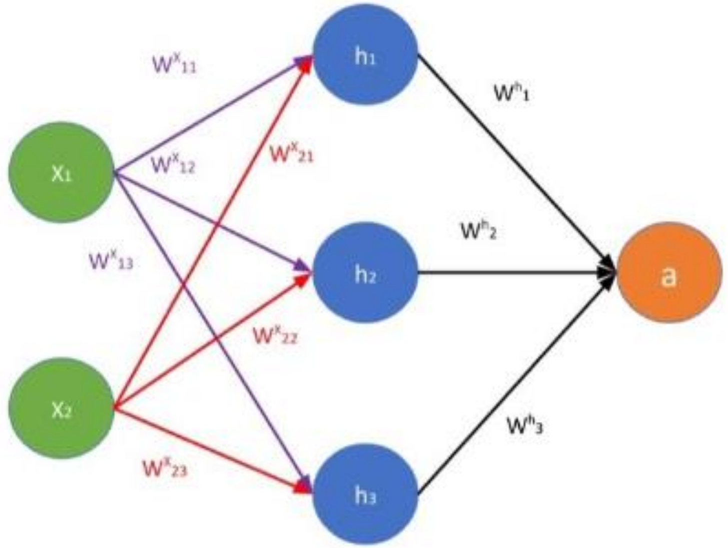
Ex-7: Obtain the output of the neuron Y



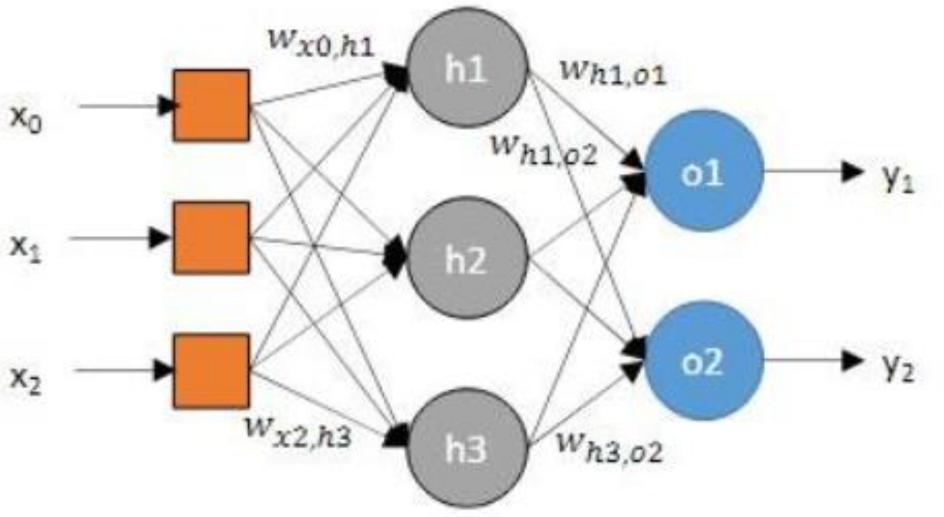
Ex-8: Obtain the output of the neuron Y



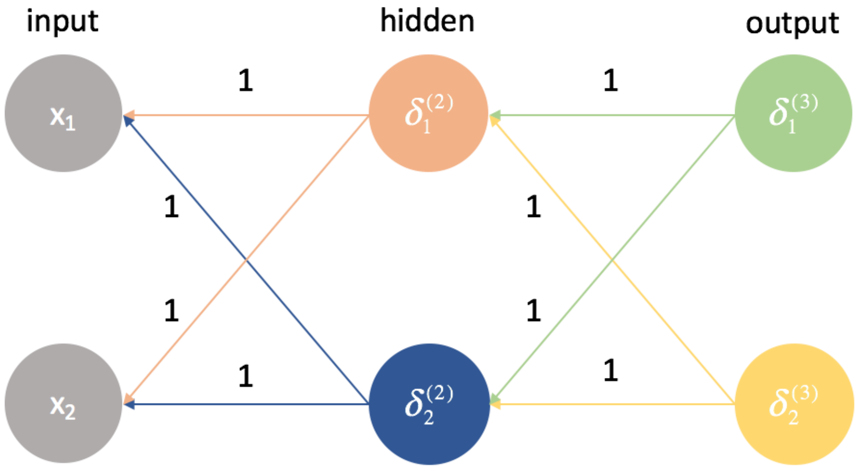
Ex-9: Obtain the output of the neuron Y



Ex-10: Obtain the output of the neuron Y



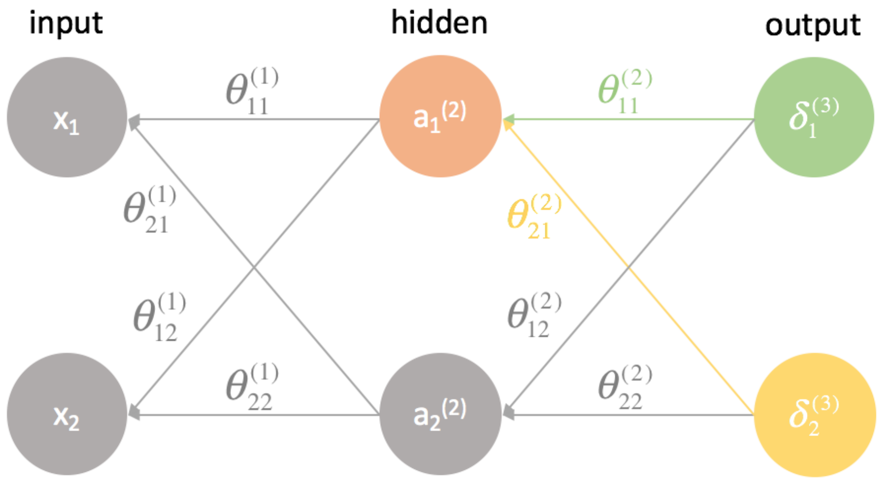
Ex-11: Obtain the output of the neuron Y



$$\delta_1^{(2)} = \left(\delta_1^{(3)} + \delta_2^{(3)}\right) f'\left(a^{(2)}\right) = \delta_2^{(2)}$$

Ex-12: Obtain the output of the neuron Y

$$\delta_1^{(2)} = \left(\delta_1^{(3)}\theta_{11}^{(2)} + \delta_2^{(3)}\theta_{21}^{(2)}\right) f'\left(a^{(2)}\right)$$



❑ Solve Problems by apply forward propagation:

Problem-01: Obtain the output of the neuron Y using

i) Binary sigmoidal and

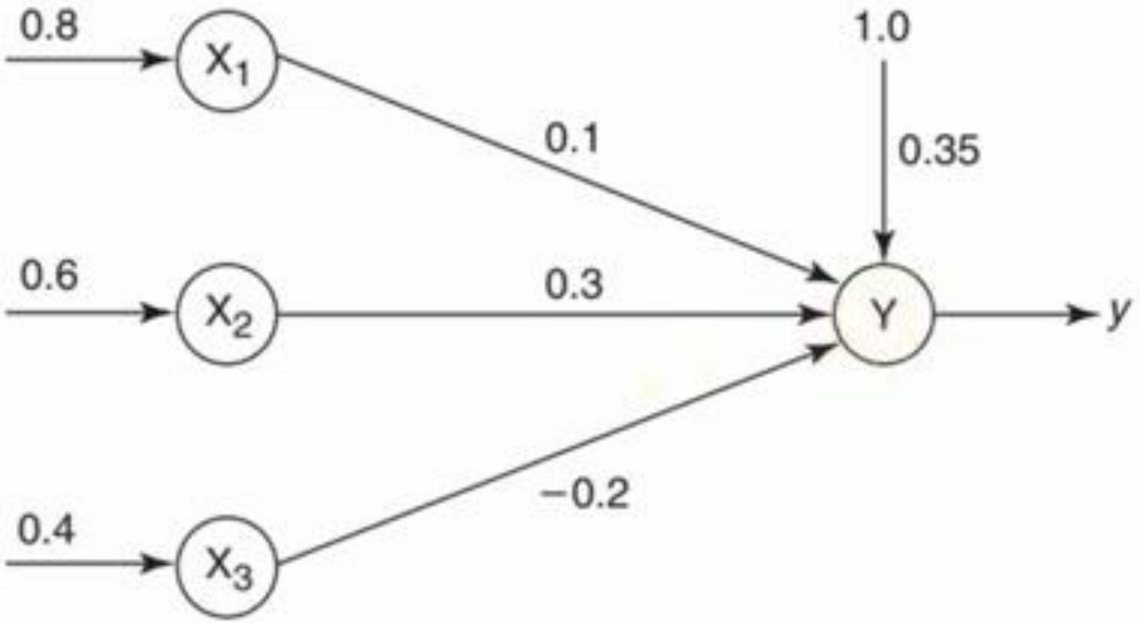
ii) Bipolar sigmoidal activation function

- Binary Sigmoid Activation Function

$$y = f(y_{in}) = \frac{1}{1+e^{-y_{in}}}$$

- Bipolar Sigmoid Activation Function

$$y = f(y_{in}) = \frac{2}{1+e^{-y_{in}}} - 1$$



- The net input to the output neuron is

$$y_{in} = b + \sum_{i=1}^{n} x_i w_i$$

$$= b + x_1 w_1 + x_2 w_2 + x_3 w_3$$

$$= 0.35 + 0.8 \times 0.1 + 0.6 \times 0.3 + 0.4 \times (-0.2)$$

$$= 0.35 + 0.08 + 0.18 - 0.08 = 0.53$$

- Binary Sigmoid Activation Function

$$y = f(y_{in}) = \frac{1}{1+e^{-y_{in}}} = \frac{1}{1+e^{-0.53}} = 0.625$$

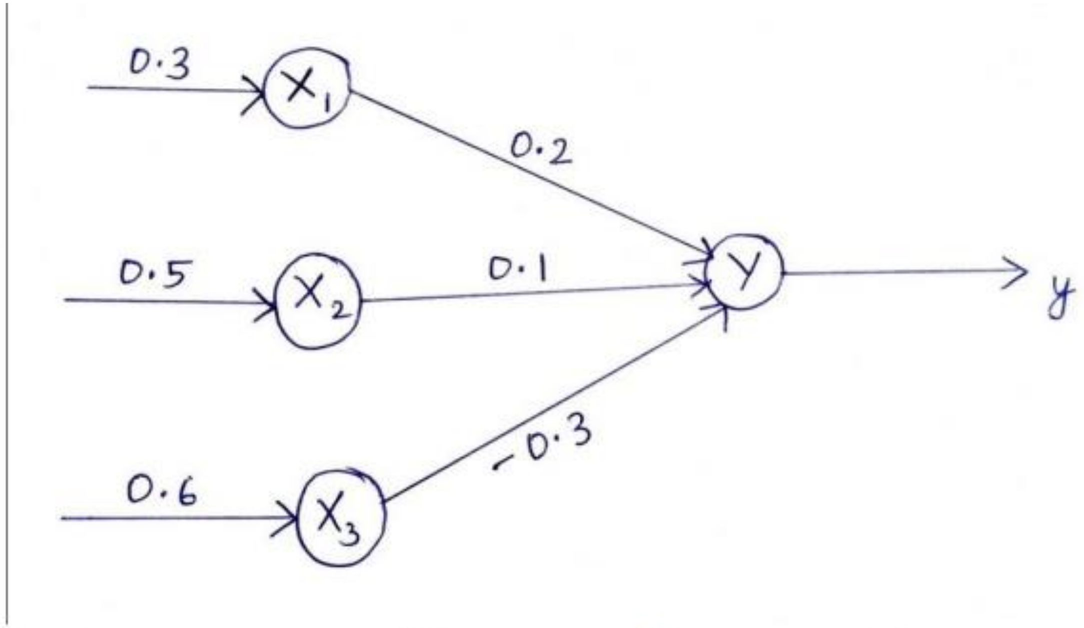
- Bipolar Sigmoid Activation Function

$$y = f(y_{in}) = \frac{2}{1+e^{-y_{in}}} - 1 = \frac{2}{1+e^{-0.53}} - 1 = 0.259$$

Problem-02: Obtain the output of the neuron Y using

i) Binary sigmoidal and

ii) Bipolar sigmoidal activation function



Inputs, $x_1=0.3$ $x_2=0.5$ $x_3=0.6$

Weights, $W_1=0.2$ $W_2=0.1$ $W_3=-0.3$

Net Input, $y_{in} = x_1 W_1 + x_2 W_2 + x_3 W_3$

$= 0.3*0.2 + 0.5*0.1 + 0.6*-0.3$

$= 0.06 + 0.05 - 0.18$

$=0.11 - 0.18$

$=$ **-0.07**

Binary Sigmoid Activation Function

$$y = f(y_{in}) = \frac{1}{1+e^{-y_{in}}}$$

Bipolar Sigmoid Activation Function

$$y = f(y_{in}) = \frac{2}{1+e^{-y_{in}}} - 1$$

Applying activation functions:

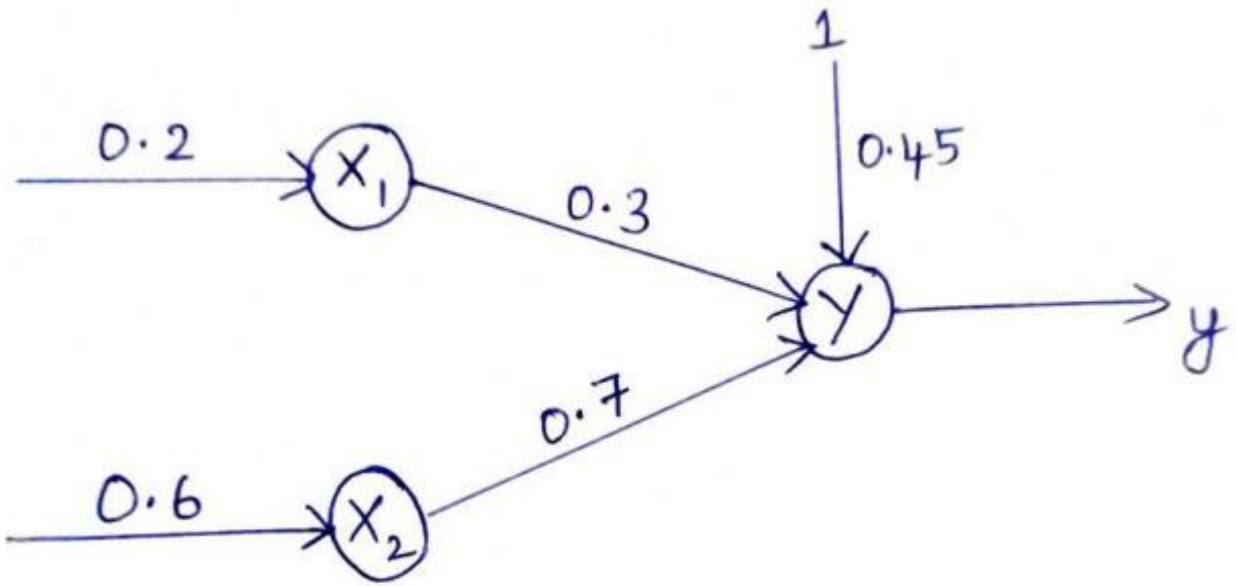
$$Y = f(-0.07) = \frac{1}{1+e^{0.07}} = 0.482$$

$$Y = f(-0.07) = \frac{2}{1+e^{0.07}} - 1 = -0.034$$

Problem-03: Obtain the output of the neuron Y using

i) Binary sigmoidal and

ii) Bipolar sigmoidal activation function



Binary Sigmoid Activation Function

$$y = f(y_{in}) = \frac{1}{1+e^{-y_{in}}}$$

Bipolar Sigmoid Activation Function

$$y = f(y_{in}) = \frac{2}{1+e^{-y_{in}}} - 1$$

Inputs, $x_1=0.2$ $x_2=0.6$

Weights, $W_1=0.3$ $W_2=0.7$

Net Input, $y_{in} = \text{bias} + x_1 W_1 + x_2 W_2$

$=0.45 + 0.2*0.3 + 0.6*0.7$

$=0.45 + 0.06 + 0.42$

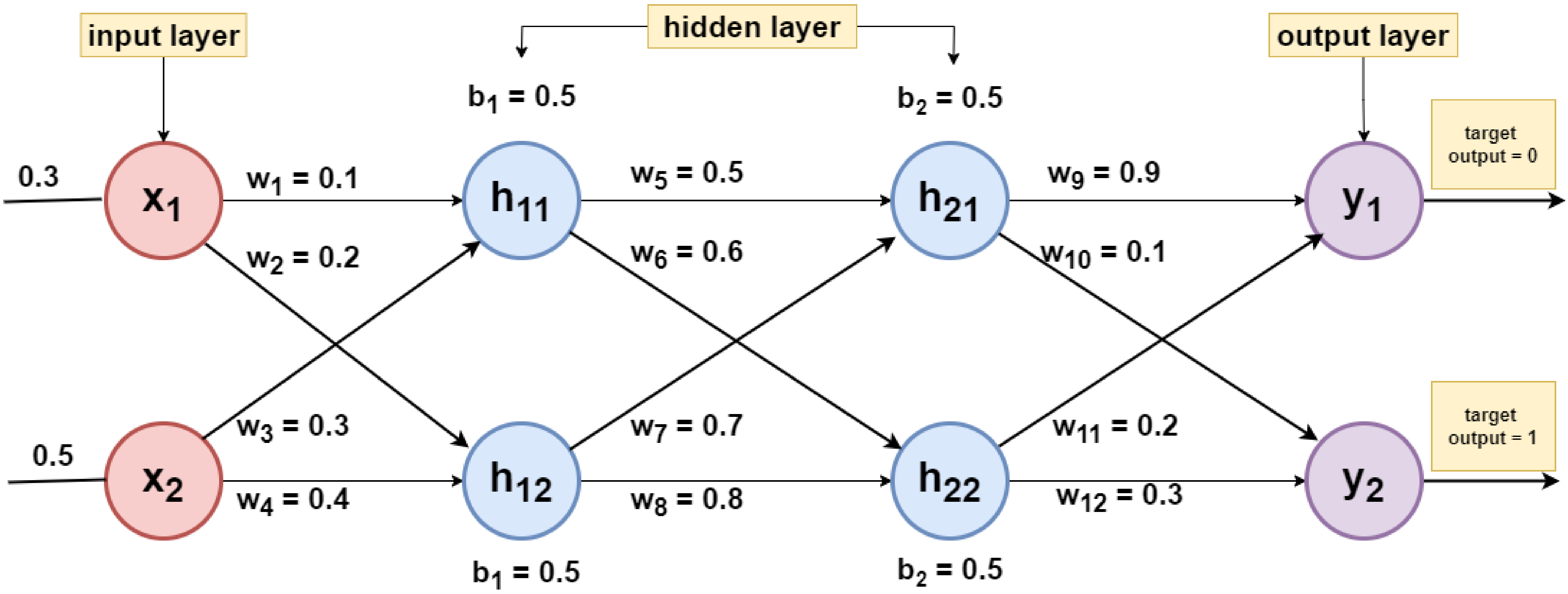
$=0.93$

Applying activation functions:

$$Y = f(0.93) = \frac{1}{1+e^{-0.93}} = 0.717$$

$$Y = f(0.93) = \frac{2}{1+e^{-0.93}} - 1 = 0.434$$

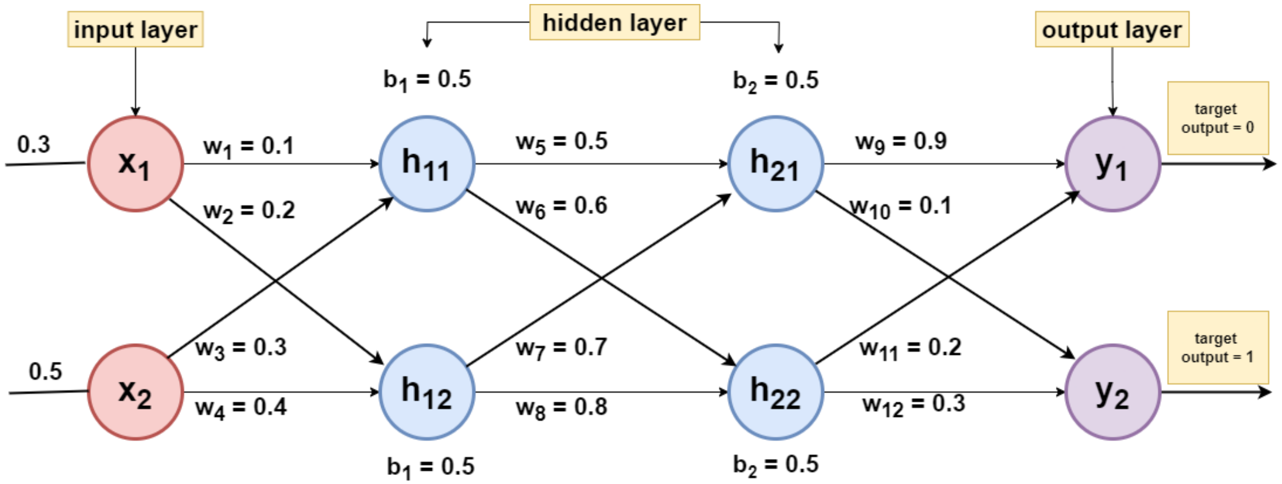
Problem-04: Obtain the output of the neuron Y using (Binary) sigmoidal activation function



Solution : Obtain the output of the neuron Y using Binary sigmoidal activation function.

Binary sigmoid activation function:

$$f(y) = \frac{1}{1+e^{-y}}$$



First hidden layer:

$$\begin{aligned} h11 &= x1*w1 + x2*w3 + b1 \\ &= 0.3*0.1 + 0.5*0.3 + 0.5 \\ &= 0.68 \end{aligned}$$

$$f(h11) = \frac{1}{1+e^{-h11}} = \frac{1}{1+e^{-0.68}} = 0.66$$

$$\begin{aligned} h12 &= x1*w2 + x2*w4 + b1 \\ &= 0.3*0.2 + 0.5*0.4 + 0.5 \\ &= 0.76 \end{aligned}$$

$$f(h12) = \frac{1}{1+e^{-h12}} = \frac{1}{1+e^{-0.76}} = 0.68$$

Second hidden layer:

$$\begin{aligned} h21 &= h11*w5 + h12*w7 + b2 \\ &= 0.66*0.1 + 0.68*0.3 + 0.5 \\ &= 1.306 \end{aligned}$$

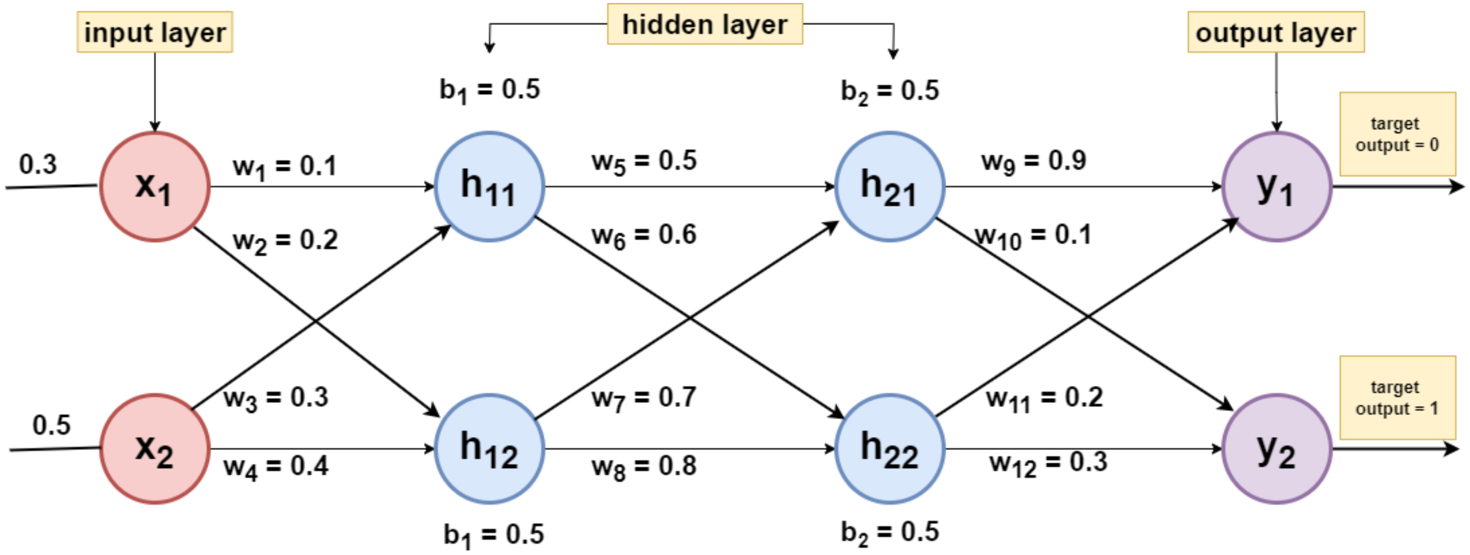
$$f(h21) = \frac{1}{1+e^{-h21}} = \frac{1}{1+e^{-1.306}} = 0.786$$

$$\begin{aligned} h22 &= h11*w6 + h12*w8 + b2 \\ &= 0.66*0.6 + 0.68*0.8 + 0.5 \\ &= 1.44 \end{aligned}$$

$$f(h22) = \frac{1}{1+e^{-h22}} = \frac{1}{1+e^{-1.44}} = 0.\ 808$$

Binary sigmoid activation function:

$$f(y) = \frac{1}{1+e^{-y}}$$



Output layer:

y11 = h21*w9 + h22*w11
= (0.786*0.9) + (0.808*0.2)
= 0.869

$$f(y11) = \frac{1}{1+e^{-y1}} = \frac{1}{1+e^{-0.869}} = \mathbf{0.704}$$

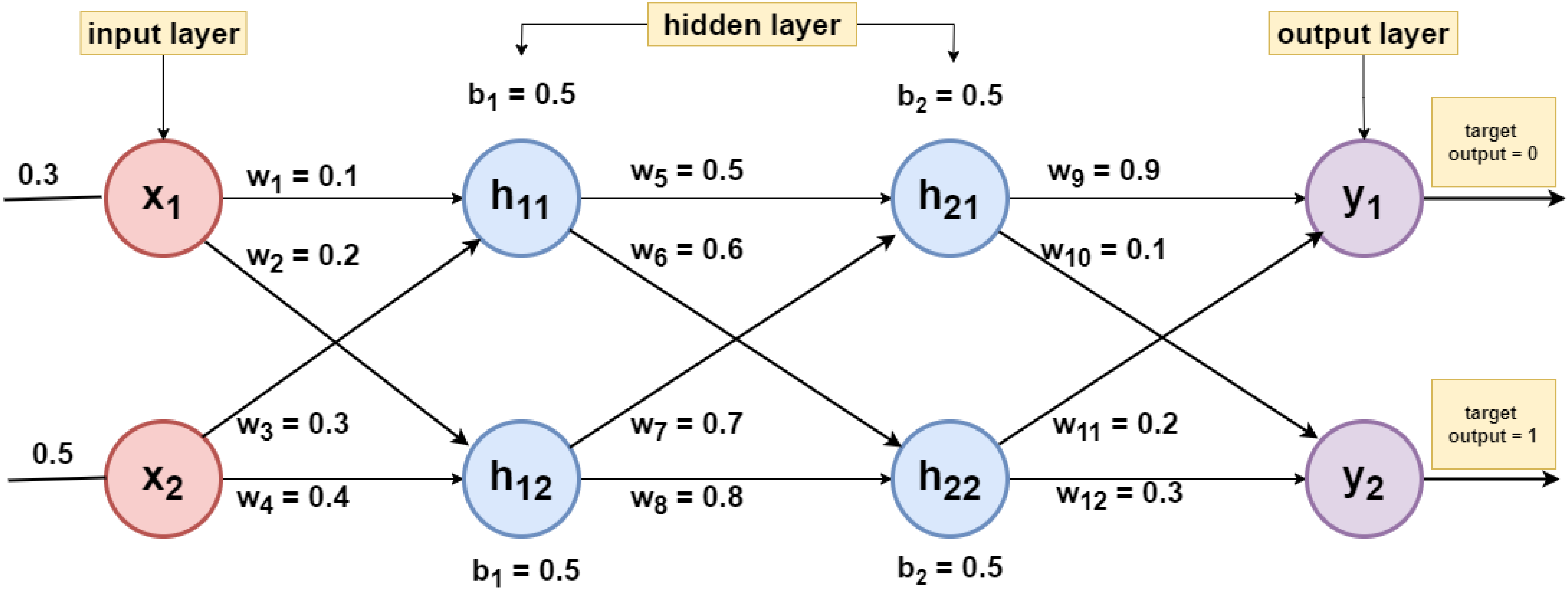
y2 = h21*w10 + h22*w12
= (0.786*0.1) + (0.808*0.3)
= 0.321

$$f(y2) = \frac{1}{1+e^{-y2}} = \frac{1}{1+e^{-0.321}} = \mathbf{0.579}$$

Calculation of Loss: (Cross entropy or log loss)

Mean squared error = ½ [$(y1_{actual} - y1_{target})^2 + (y2_{actual} - y2_{target})^2$]
= ½ [$(0.704 - 0)^2 + (0.579 - 1)^2$]
= ½ [0.495 + 0.177]
= ½ * 0.672
= 0.336

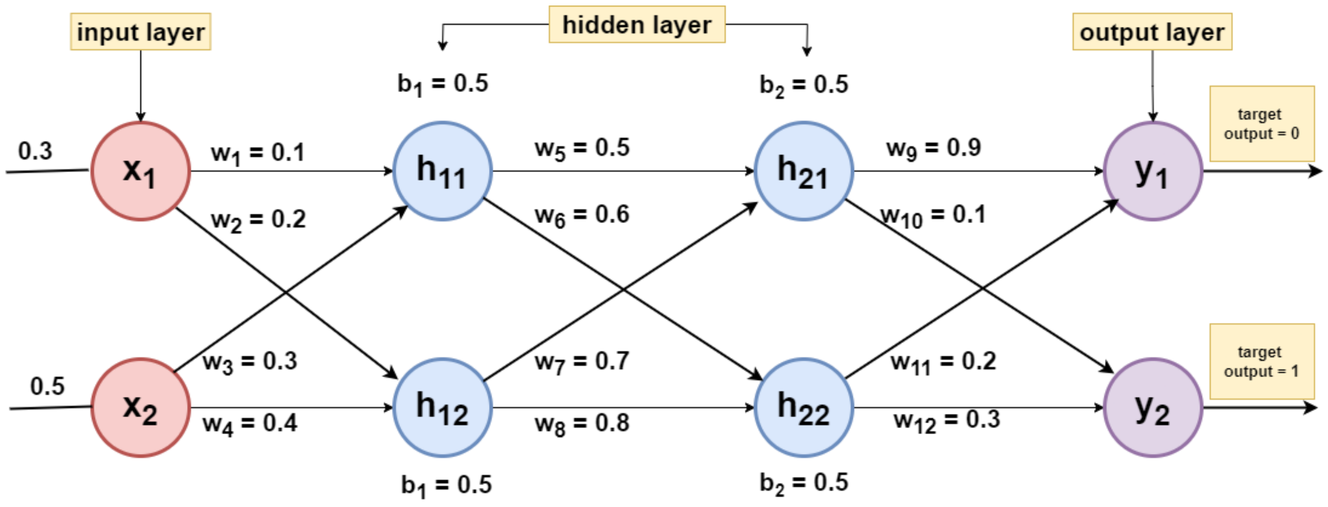
Problem-05: Obtain the output of the neuron Y using Bipolar sigmoidal activation function



Solution : Obtain the output of the neuron Y using Bipolar sigmoidal activation function.

Bipolar sigmoid activation function:

$$f(y) = \frac{2}{1+e^{-y}} - 1$$



First hidden layer:

h11 = x1*w1 + x2*w3 +b1
= 0.3*0.1 + 0.5*0.3 + 0.5
= 0.68

$$f(h11) = \frac{2}{1+e^{-h11}} - 1 = \frac{2}{1+e^{-0.68}} - 1 = 0.327$$

h12 = x1*w2 + x2*w4 +b1
= 0.3*0.2 + 0.5*0.4 + 0.5
= 0.76

$$f(h12) = \frac{2}{1+e^{-h12}} - 1 = \frac{2}{1+e^{-0.76}} - 1 = 0.362$$

Second hidden layer:

h21 = h11*w5 + h12*w7 +b2 = 0.66*0.1 + 0.68*0.3 + 0.5
= 1.306

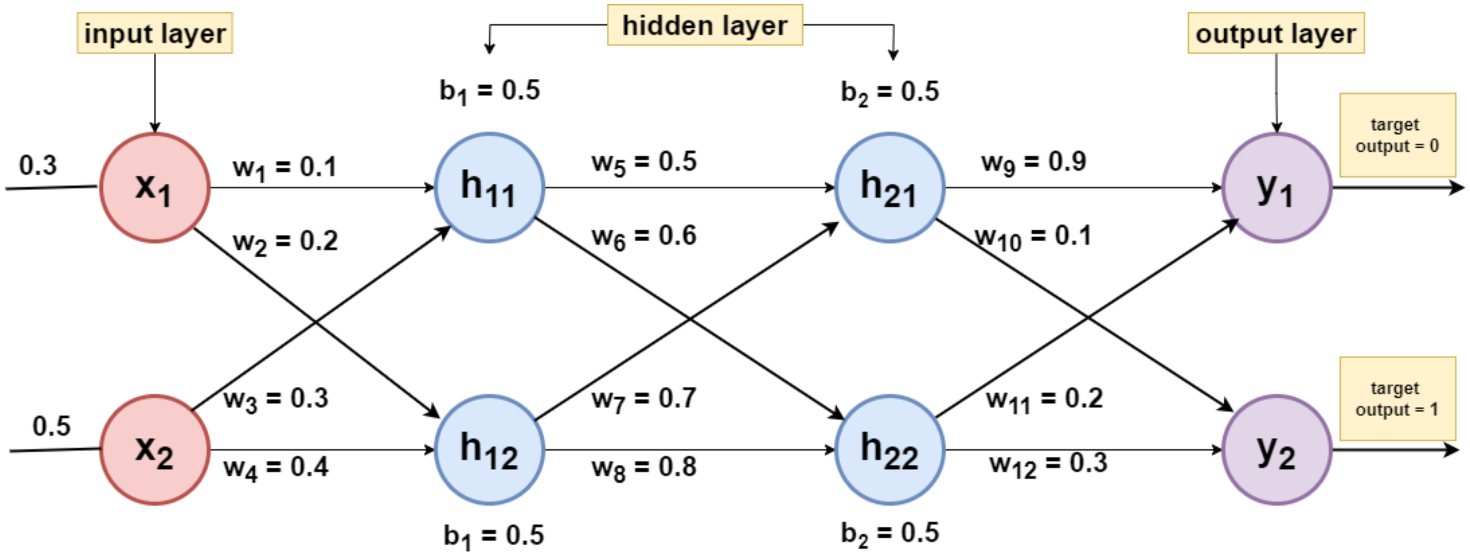
$$f(h21) = \frac{2}{1+e^{-h21}} - 1 = \frac{2}{1+e^{-1.306}} - 1 = 0.573$$

h22 = h11*w6 + h12*w8 +b2 = 0.66*0.6 + 0.68*0.8 + 0.5
= 1.44

$$f(h22) = \frac{2}{1+e^{-h22}} = \frac{2}{1+e^{-1.44}} - 1 = 0.\ 616$$

Bipolar sigmoid activation function:

$$f(y) = \frac{2}{1+e^{-y}} - 1$$



<u>Output layer:</u>

y11 = h21*w9 + h22*w11

= (0.573 * 0.9) + (0.616 * 0.2)

= 0.638

$$f(y11) = \frac{2}{1+e^{-y1}} - 1 = \frac{2}{1+e^{-0.638}} - 1 = 0.308$$

y2 = h21*w10 + h22*w12

= (0.573 * 0.1) + (0.616 * 0.3)

= 0.242

$$f(y2) = \frac{2}{1+e^{-y1}} - 1 = \frac{2}{1+e^{-0.242}} - 1 = 0.120$$

Mean squared error = ½ [$(y1_{actual} - y1_{target})^2 + (y2_{actual} - y2_{target})^2$]

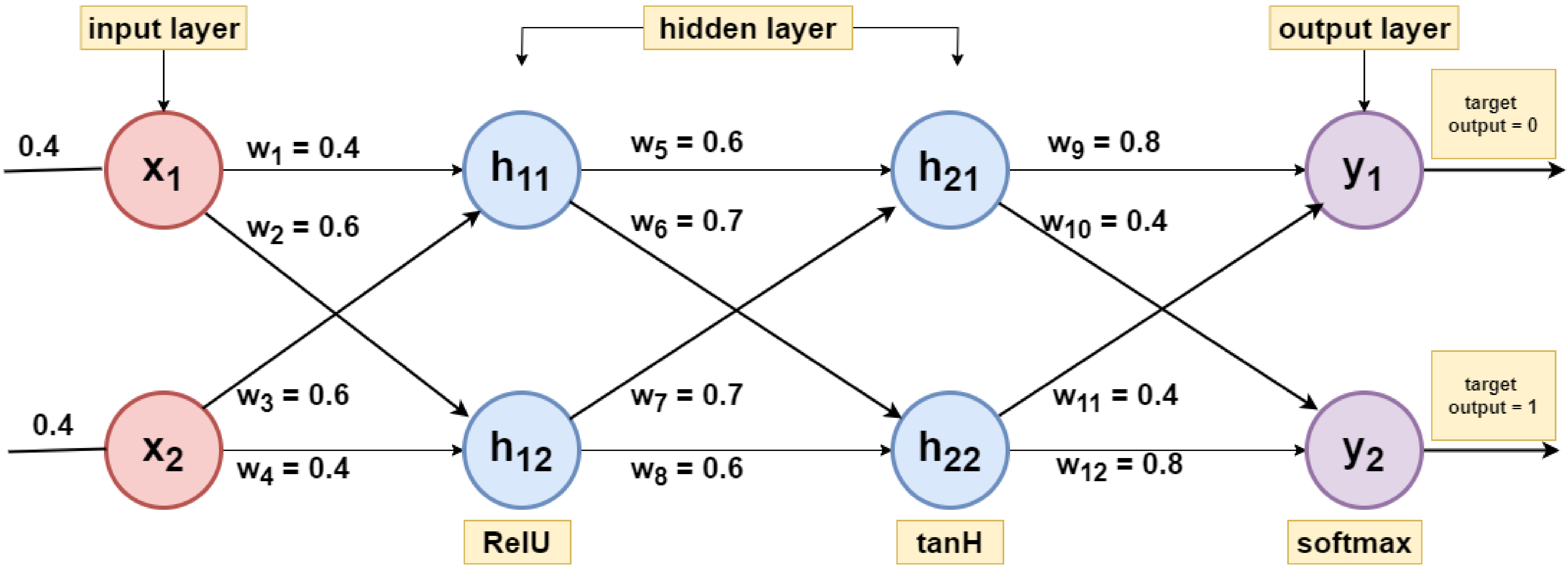
= ½ [$(0.308 - 0)^2 + (0.120 - 1)^2$]

= ½ [0.094 + 0.774]

= ½ * [0.868]

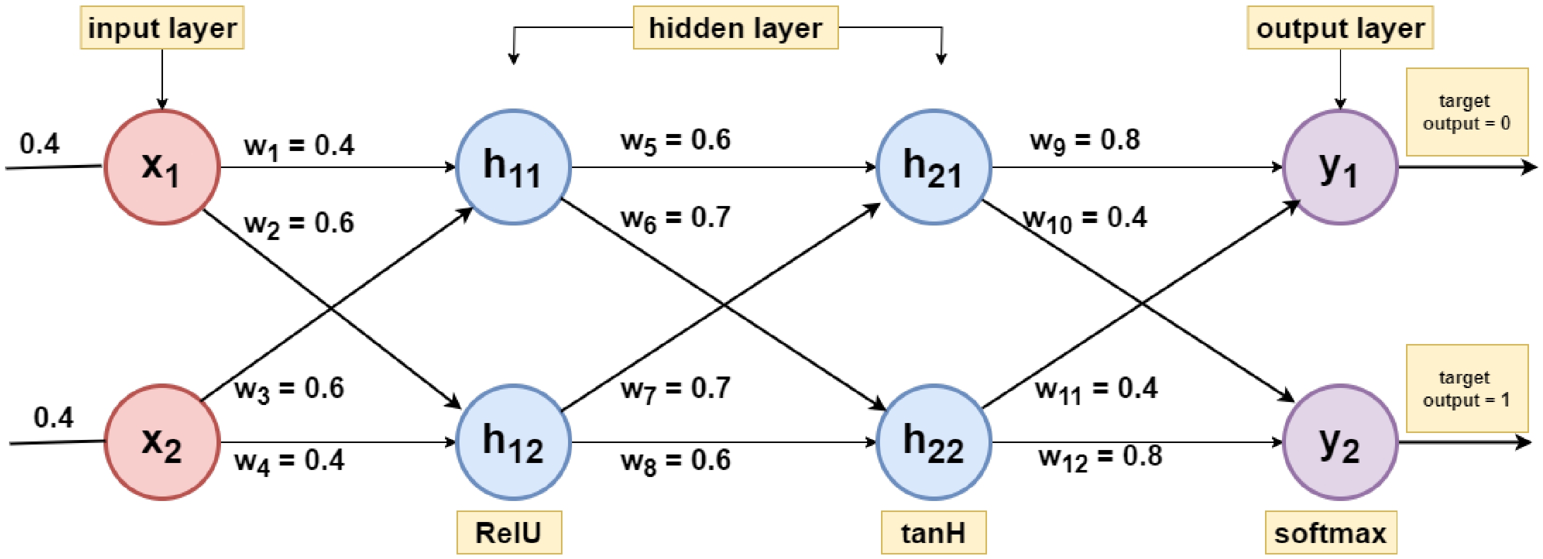
= 0.434

Problem-06: Obtain the output of the neuron Y



Solution :

Using RelU activation function:
f(y) = max(0, y)



First hidden layer:

h11 = x1*w1 + x2*w3
= (0.4*0.4) + (0.4*0.6)
= 0.24

f(h11) = max(0, h11) = (0, 0.24) = 0.24

h12 = x1*w2 + x2*w4
= (0.4*0.6) + (0.4*0.5)
= 0.2

f(h12) = max(0, h12) = (0, 0.2) = 0.2

tanH activation function: $f(y) = \dfrac{2}{1+e^{-2y}} - 1$

Second hidden layer:

h21 = h11*w5 + h12*w7 = (0.24 * 0.6) + (0.2 * 0.7) = 0.14

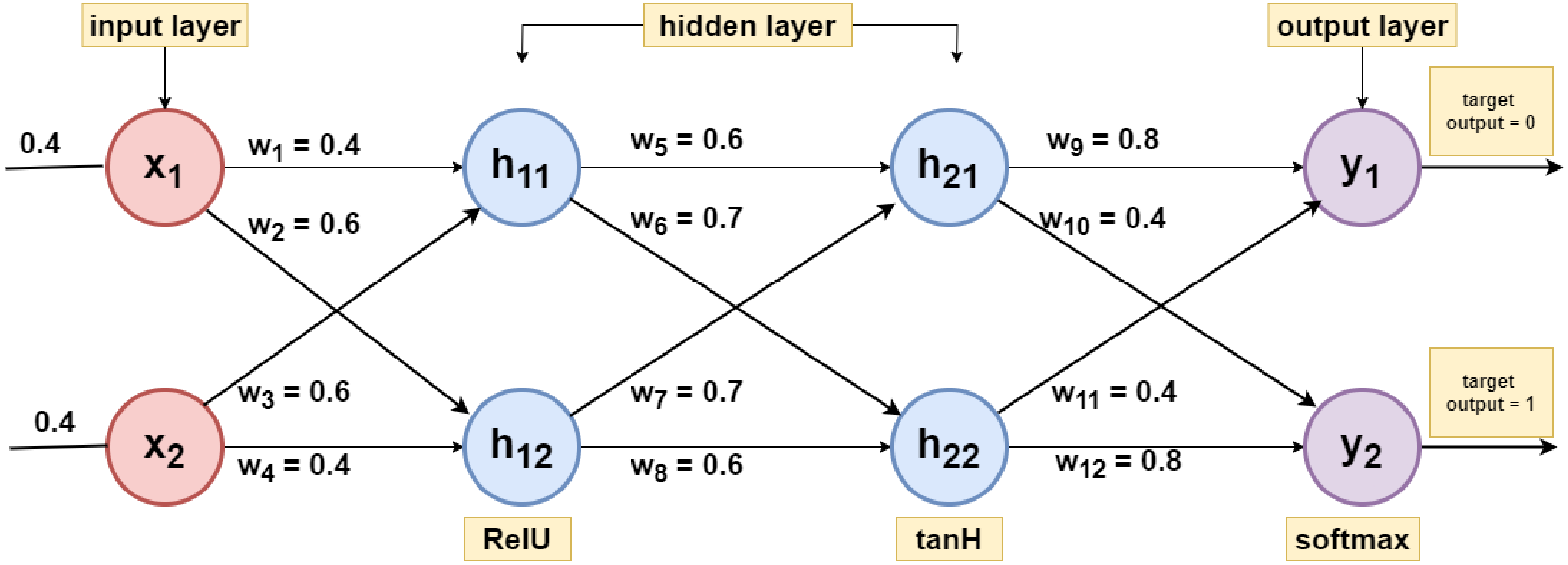
$$f(h21) = \frac{2}{1+e^{-2*h21}} - 1 = \frac{2}{1+e^{-2*0.14}} - 1 = 0.139$$

h22 = h11*w6 + h12*w8 = (0.24 * 0.7) + (0.2 * 0.6) = 0.12

$$f(h22) = \frac{2}{1+e^{-h22}} = \frac{2}{1+e^{-2*0.12}} - 1 = 0.119$$

SoftMax activation function:

$$s\left(x_i\right) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$



Output layer:

y1 = h21*w9 + h22*w11
= (0.139 * 0.8) + (0.119 * 0.4)
= 0.047

$$f(y1) = \frac{e^{-y1}}{e^{-y1}+e^{-y2}} = \frac{e^{-0.047}}{e^{-0.047}+e^{-0.095}} = \mathbf{0.511}$$

y2 = h21*w10 + h22*w12
= (0.139 * 0.4) + (0.119 * 0.8)
= 0.095

$$f(y2) = \frac{e^{-y1}}{e^{-y1}+e^{-y2}} = \frac{e^{-0.095}}{e^{-0.047}+e^{-0.095}} = \mathbf{0.488}$$

Mean squared error $= ½ \left[(y1_{actual} - y1_{target})^2 + (y2_{actual} - y2_{target})^2 \right]$

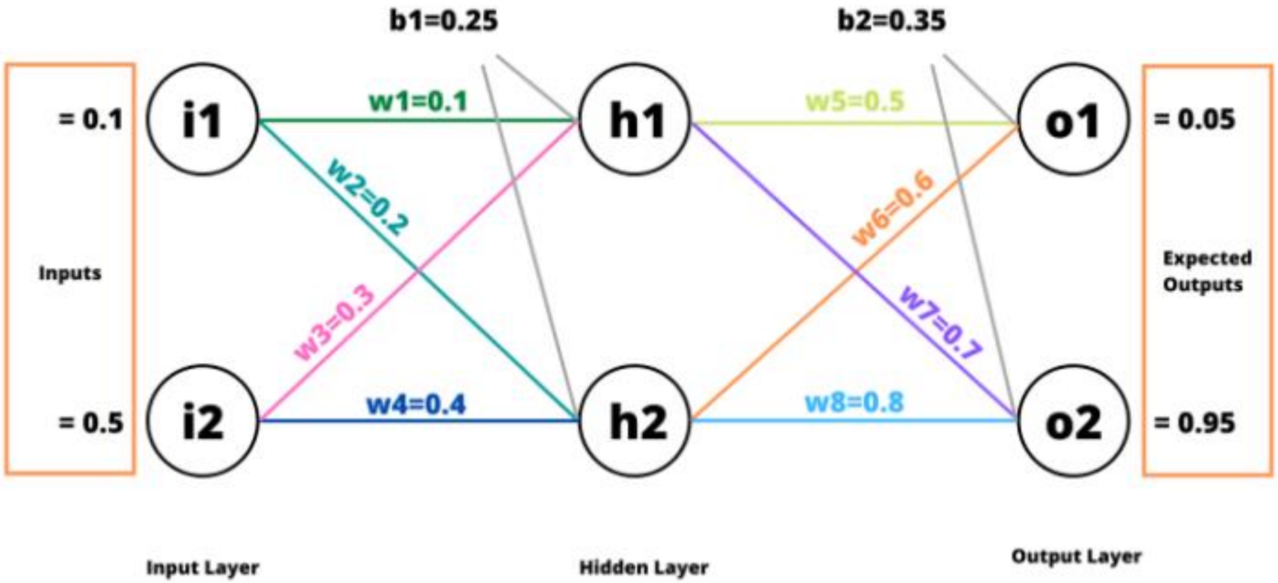
$= ½ \left[(\mathbf{0.511} - 0)^2 + (\mathbf{0.488} - 1)^2 \right]$

$= ½ \left[0.261 + 0.262 \right]$

$= ½ \; * \left[0.\,068 \right]$

$= 0.034$

Problem-07: Obtain the output of the neuron Y Using sigmoid activation function.



Let's get started with the forward pass.

For h1,

$$sum_{h1} = i_1 * w_1 + i_2 * w_3 + b_1$$

$$sum_{h1} = 0.1 * 0.1 + 0.5 * 0.3 + 0.25 = 0.41$$

Now we pass this weighted sum through the logistic function (sigmoid function) so as to squash the weighted sum into the range (0 and +1). The logistic function is an activation function for our example neural network.

$$output_{h1} = \frac{1}{1 + e^{-sum_{h1}}}$$

$$output_{h1} = \frac{1}{1 + e^{-0.41}} = 0.60108$$

Similarly for h2, we perform the weighted sum operation sum_{h2} and compute the activation value $output_{h2}$.

$$sum_{h2} = i_1 * w_2 + i_2 * w_4 + b_1 = 0.47$$

$$output_{h2} = \frac{1}{1 + e^{-sum_{h2}}} = 0.61538$$

Now, $output_{h1}$ and $output_{h2}$ will be considered as inputs to the next layer.

For o1,

$$sum_{o1} = output_{h1} * w_5 + output_{h2} * w_6 + b_2 = 1.01977$$

$$output_{o1} = \frac{1}{1 + e^{-sum_{o1}}} = 0.73492$$

Similarly for o2,

$$sum_{o2} = output_{h1} * w_7 + output_{h2} * w_8 + b_2 = 1.26306$$

$$output_{o2} = \frac{1}{1 + e^{-sum_{o2}}} = 0.77955$$

Computing the total error

We started off supposing the expected outputs to be 0.05 and 0.95 respectively for $output_{o1}$ and $output_{o2}$. Now we will compute the errors based on the outputs received until now and the expected outputs.

We'll use the following error formula,

$$E_{total} = \sum \frac{1}{2}(target - output)^2$$

To compute E_{total}, we need to first find out respective errors at $o1$ and $o2$.

$$E_1 = \frac{1}{2}(target_1 - output_{o1})^2$$

$$E_1 = \frac{1}{2}(0.05 - 0.73492)^2 = 0.23456$$

Similarly for E2,

$$E_2 = \frac{1}{2}(target_2 - output_{o2})^2$$

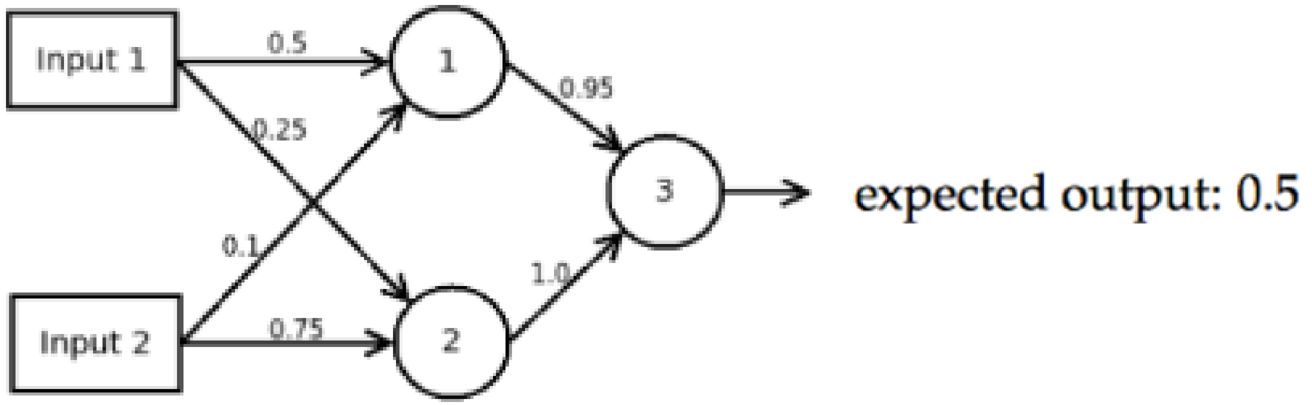
$$E_2 = \frac{1}{2}(0.95 - 0.77955)^2 = 0.01452$$

Therefore,

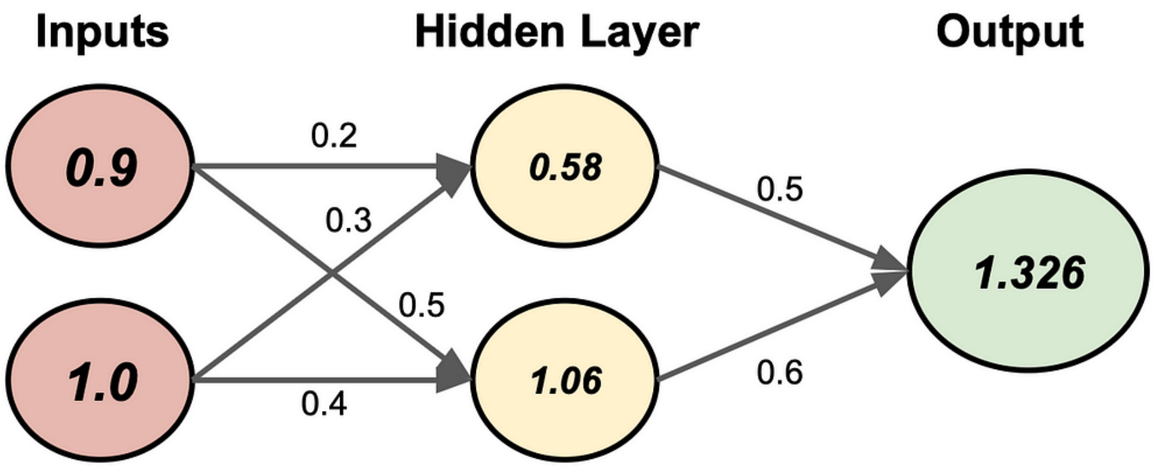
$$E_{total} = E_1 + E_2 = 0.24908$$

Problem-08: Obtain the output of the neuron Y Using sigmoid activation function.

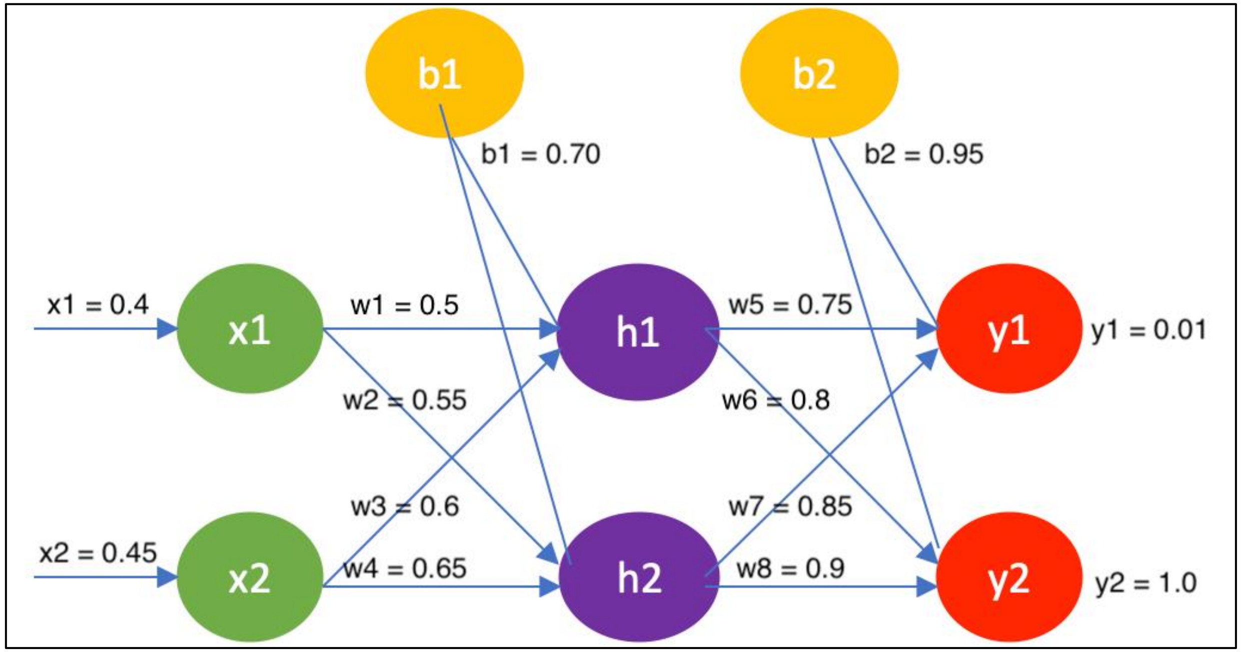
Given the following neural network (Assume each node uses a sigmoid activation function):



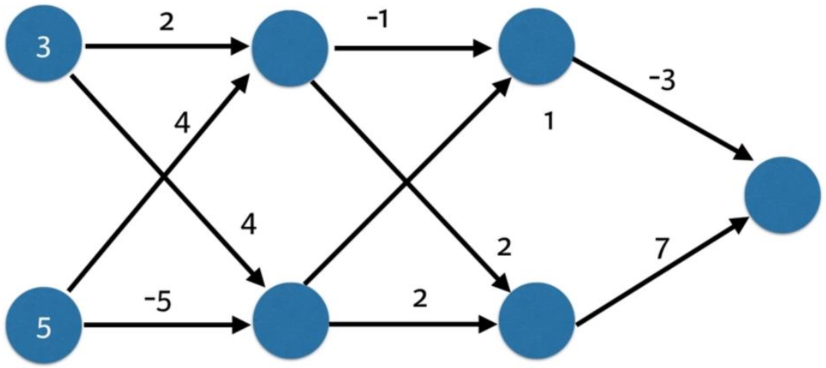
Problem-09: Obtain the output of the neuron Y Using sigmoid activation function.



Problem-10: Obtain the output of the neuron Y Using sigmoid activation function.

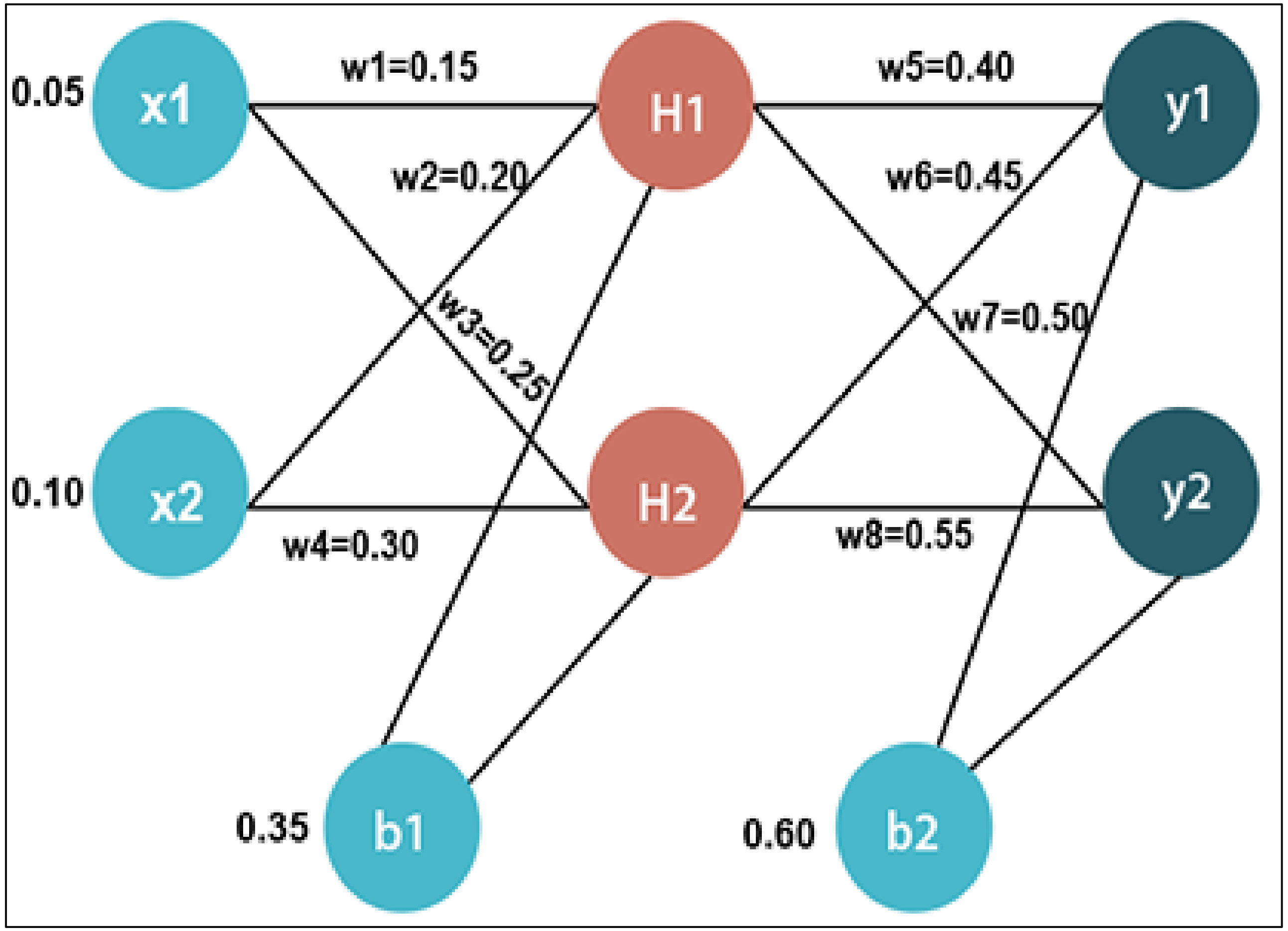


Problem-11: Obtain the output of the neuron Y



Calculate with ReLU Activation Function

Problem-12: Obtain the output of the neuron Y Using sigmoid activation function.



Target Value
T1=0.01
T2=0.99