Back Propagation

Course Title: Artificial Intelligence

Course Code: cse-403

Khandaker Jannatul Ritu, Lecturer, Dept. of CSE(BAIUST)

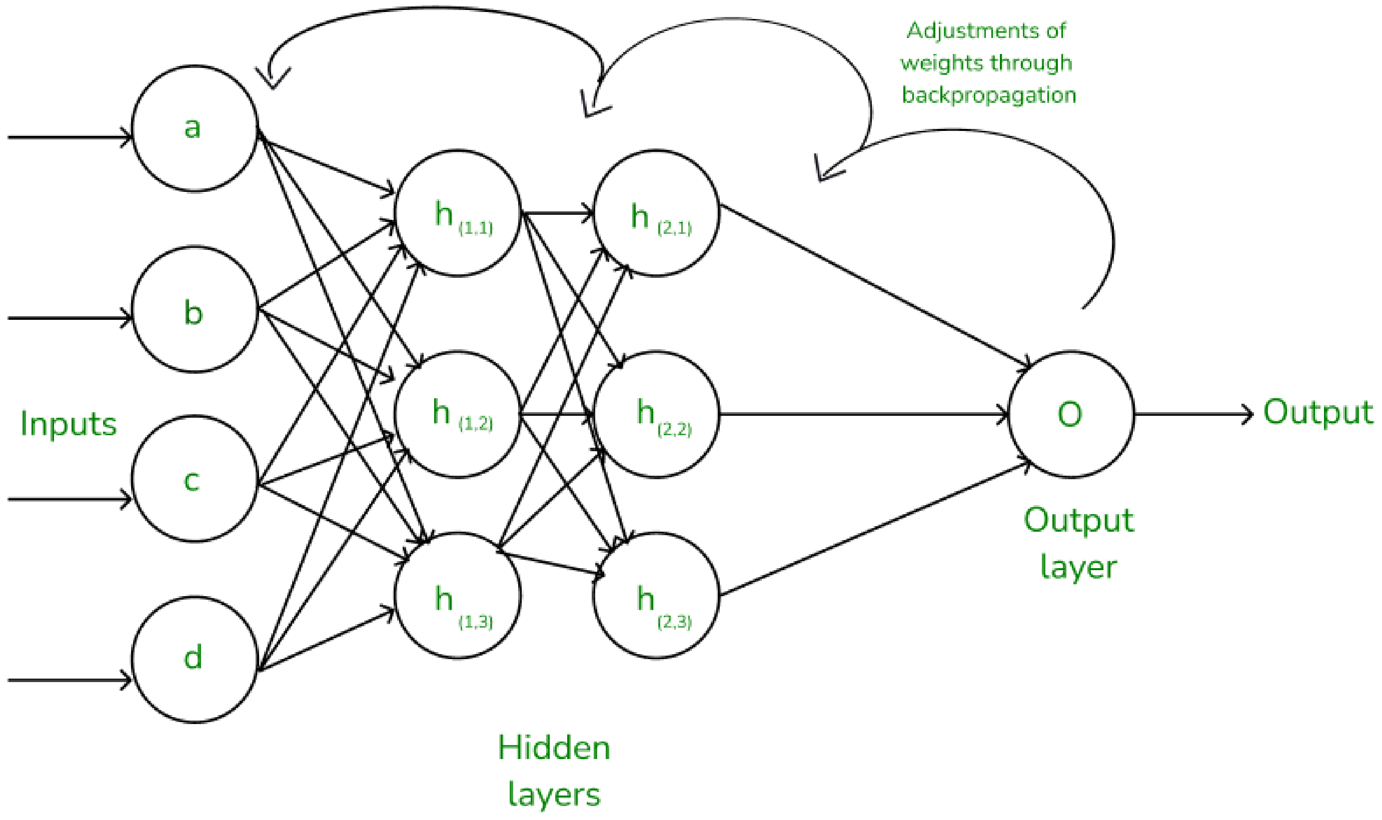
Outlines:

- What is backpropagation?
- Why the Backpropagation Algorithm?
- Why We Need Backpropagation?
- Backpropagation Key Points
- Backpropagation
- The algorithm works as follows:
- Types of Backpropagation Networks
- Disadvantages of using Backpropagation
- Forward propagation vs backward propagation in neural network
- Example: 01, 02, 03, 04, 05, 06, 07, 08

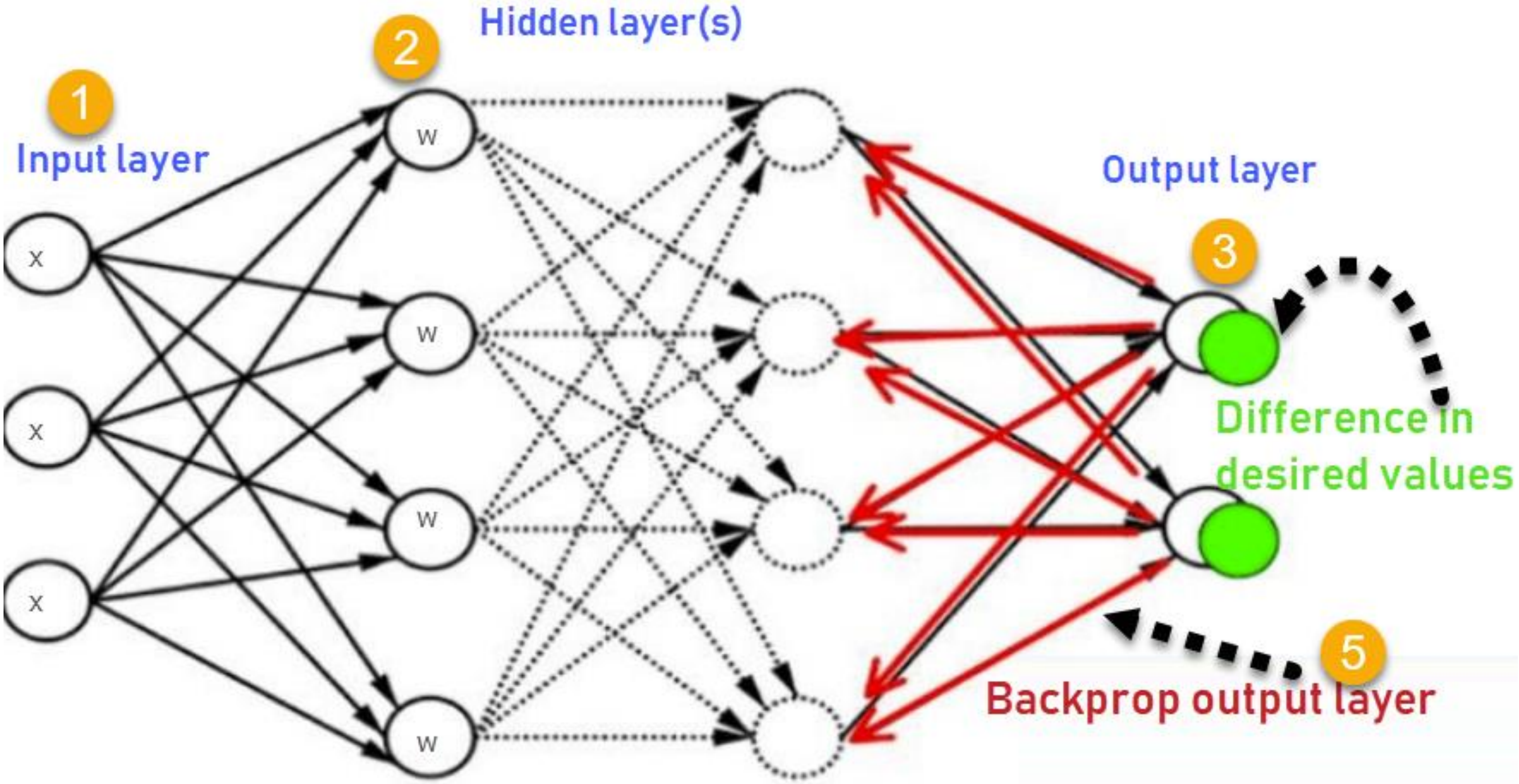
What is backpropagation?

•Backpropagation is an iterative algorithm, that helps to minimize the cost function by determining which weights and biases should be adjusted. During every epoch, the model learns by adapting the weights and biases to minimize the loss by moving down toward the gradient of the error. Thus, it involves the two most popular optimization algorithms, such as gradient descent or stochastic gradient descent.

•Computing the gradient in the backpropagation algorithm helps to minimize the cost function and it can be implemented by using the mathematical rule called chain rule from calculus to navigate through complex layers of the neural network.



fig(a) A simple illustration of how the backpropagation works by adjustments of weights



How Backpropagation Algorithm Works

- ✓ **Backpropagation** is one of the important concepts of a neural network. Our task is to classify our data best. For this, we have to update the weights of parameter and bias, but how can we do that in a deep neural network? In the linear regression model, we use gradient descent to optimize the parameter. Similarly here we also use gradient descent algorithm using Backpropagation.
- ✓ For a single training example, **Backpropagation** algorithm calculates the gradient of the **error function**. Backpropagation can be written as a function of the neural network. Backpropagation algorithms are a set of methods used to efficiently train artificial neural networks following a gradient descent approach which exploits the chain rule.
- ✓ The main features of Backpropagation are the iterative, recursive and efficient method through which it calculates the updated weight to improve the network until it is not able to perform the task for which it is being trained. Derivatives of the activation function to be known at network design time is required to Backpropagation.
- ✓ Now, how error function is used in Backpropagation and how Backpropagation works? Let start with an example and do it mathematically to understand how exactly updates the weight using Backpropagation.

Why the Backpropagation Algorithm?

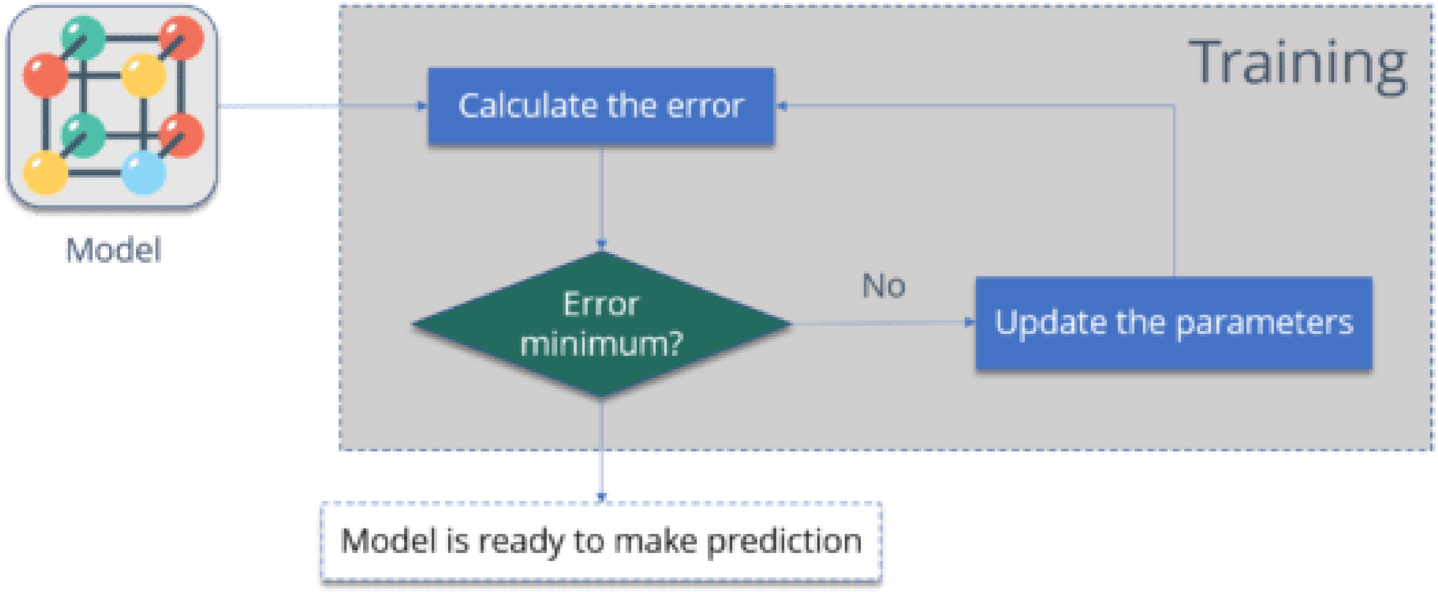
Backpropagation Algorithm works faster than other neural network algorithms. If you are familiar with data structure and algorithm, backpropagation is more like an advanced greedy approach. The backpropagation approach helps us to achieve the result faster. Backpropagation has reduced training time from month to hours. Backpropagation is currently acting as the backbone of the neural network.

Why We Need Backpropagation?

- Backpropagation is fast, simple and easy to program
- It has no parameters to tune apart from the numbers of input
- It is a flexible method as it does not require prior knowledge about the network
- It is a standard method that generally works well
- It does not need any special mention of the features of the function to be learned.

Why We Need Backpropagation?

- ✓ While designing a Neural Network, in the beginning, we initialize weights with some random values or any variable for that fact. Now obviously, we are not *superhuman*. So, it's not necessary that whatever weight values we have selected will be correct, or it fits our model the best.
- ✓ Okay, fine, we have selected some weight values in the beginning, but our model output is way different than our actual output i.e. the error value is huge. Now, how will you reduce the error?
- ✓ Basically, what we need to do, we need to somehow explain the model to change the parameters (weights), such that error becomes minimum. Let's put it in an another way, we need to train our model.
- ✓ One way to train our model is called as Backpropagation. Consider the diagram below:



✓ **Let me summarize the steps for you:**

•**Calculate the error** – How far is your model output from the actual output.

•**Minimum Error** – Check whether the error is minimized or not.

•**Update the parameters** – If the error is huge then, update the parameters (weights and biases). After that again check the error. Repeat the process until the error becomes minimum.

•**Model is ready to make a prediction** – Once the error becomes minimum, you can feed some inputs to your model and it will produce the output.

✓ **Backpropagation Key Points**

•Simplifies the network structure by elements weighted links that have the least effect on the trained network

•You need to study a group of input and activation values to develop the relationship between the input and hidden unit layers.

•It helps to assess the impact that a given input variable has on a network output. The knowledge gained from this analysis should be represented in rules.

•Backpropagation is especially useful for deep neural networks working on error-prone projects, such as image or speech recognition.

•Backpropagation takes advantage of the chain and power rules allows backpropagation to function with any number of outputs.

Backpropagation

- ✓ Backpropagation is a supervised learning algorithm and is mainly used by Multi-Layer - Perceptron's to change the weights connected to the net's hidden neuron layer(s).
- ✓ The backpropagation algorithm uses a computed output error to change the weight values in backward direction.
- ✓ To get this net error, a forward propagation phase must have been done before. While propagating in forward direction, the neurons are being activated using the sigmoid activation function.
- ✓ The formula of sigmoid activation is:

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

- ✓ **The algorithm works as follows:**

1. Perform the forward propagation phase for an input pattern and calculate the output error
2. Change all weight values of each weight matrix using the formula weight(old) + learning rate * output error * output(neurons i) * output(neurons i+1) * (1 - output(neurons i+1))
3. Go to step 1
4. The algorithm ends, if all output patterns match their target patterns

✓ Types of Backpropagation Networks

1. Static back-propagation

It is one kind of backpropagation network which produces a mapping of a static input for static output. It is useful to solve static classification issues like optical character recognition.

2. Recurrent Backpropagation

Recurrent Back propagation in data mining is fed forward until a fixed value is achieved. After that, the error is computed and propagated backward.

The **main difference** between both of these methods is: that the mapping is rapid in static back-propagation while it is non-static in recurrent backpropagation.

✓ Disadvantages of using Backpropagation

- The actual performance of backpropagation on a specific problem is dependent on the input data.
- Back propagation algorithm in data mining can be quite sensitive to noisy data
- You need to use the matrix-based approach for backpropagation instead of mini-batch.

✓ Forward propagation vs backward propagation in neural network

Aspect	Forward Propagation	Backward Propagation
Purpose	Compute the output of the neural network given inputs	Adjust the weights of the network to minimize error
Direction	Forward from input to output	Backwards, from output to input
Calculation	Computes the output using current weights and biases	Updates weights and biases using calculated gradients
Information flow	Input data -> Output data	Error signal -> Gradient updates
Steps	1. Input data is fed into the network. 2. Data is processed through hidden layers. 3. Output is generated.	1. Error is calculated using a loss function. 2. Gradients of the loss function are calculated. 3. Weights and biases are updated using gradients.
Used in	Prediction and inference	Training the neural network

What to Update?

- Weights
- Bias

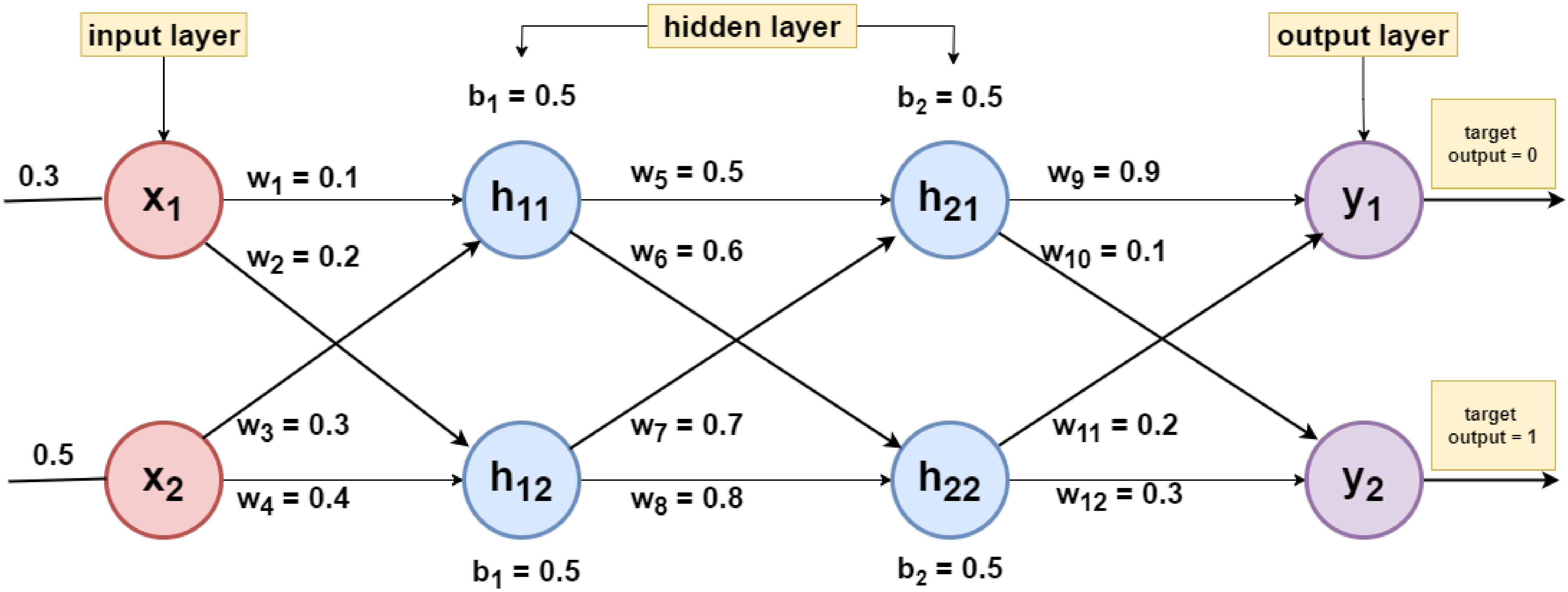
How to Update?

- Backpropagation

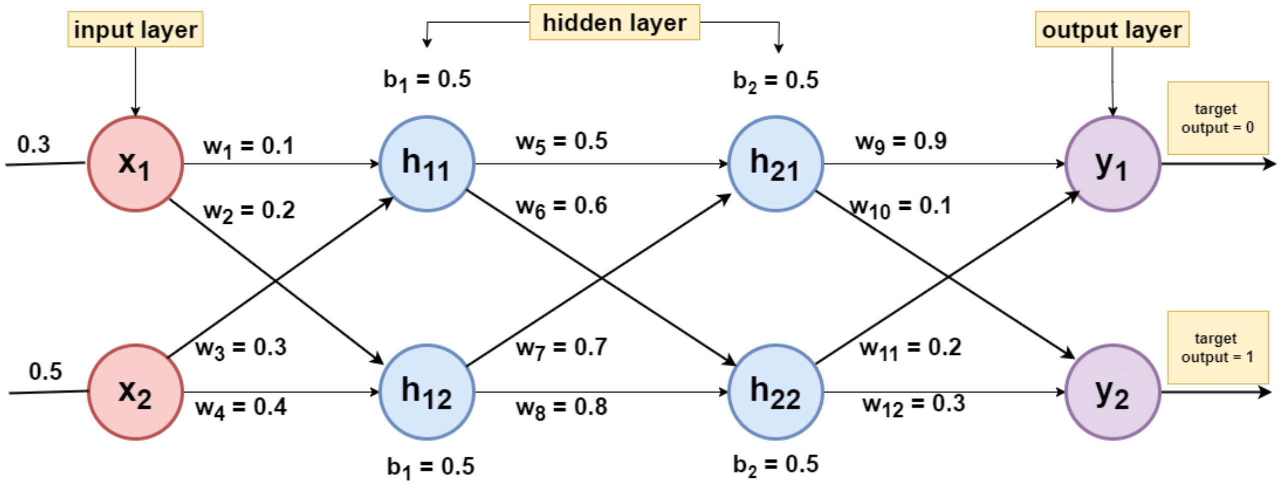
Problem-01: assume that the neurons have a sigmoid activation function, perform a forward pass and a backward pass on the network. assume that the target output is $t1 = 0$, $t2 = 1$ and learning rate(η) = 0.1

a. calculate error

b. update the weight of w12



Solution : Forward Pass



Binary sigmoid activation function:

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

First hidden layer:

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

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

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

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

Second hidden layer:

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

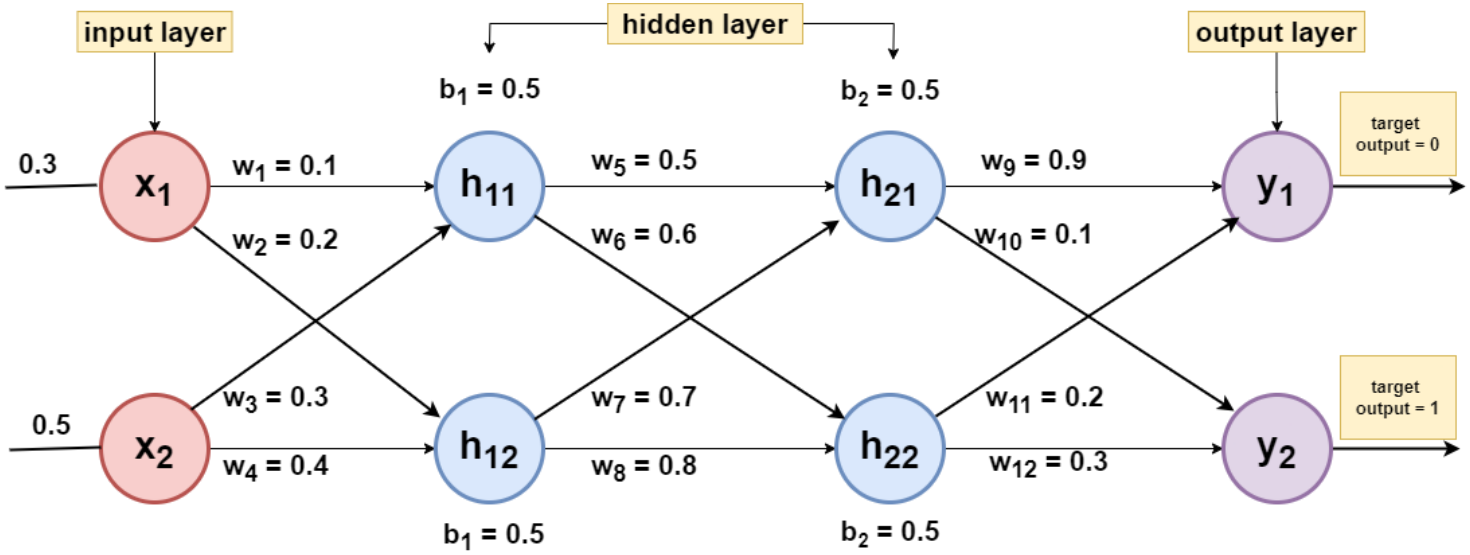
$$f(h21) = \frac{1}{1+e^{-h21}} = \frac{1}{1+e^{-1.306}} = 0.786$$

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

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

Binary sigmoid activation function:

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



Output layer:

y1 = h21*w9 + h22*w11
= (0.786*0.9) + (0.808*0.2)
= 0.869

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

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

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

Calculation of Loss: (Cross entropy or log Loss)

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

E_{total} = ½ [$(0.704 - 0)^2 + (0.579 - 1)^2$]
= ½ [0.495 + 0.177]
= ½ * 0.672
= 0.336

Backward Pass : How to Update Weights?

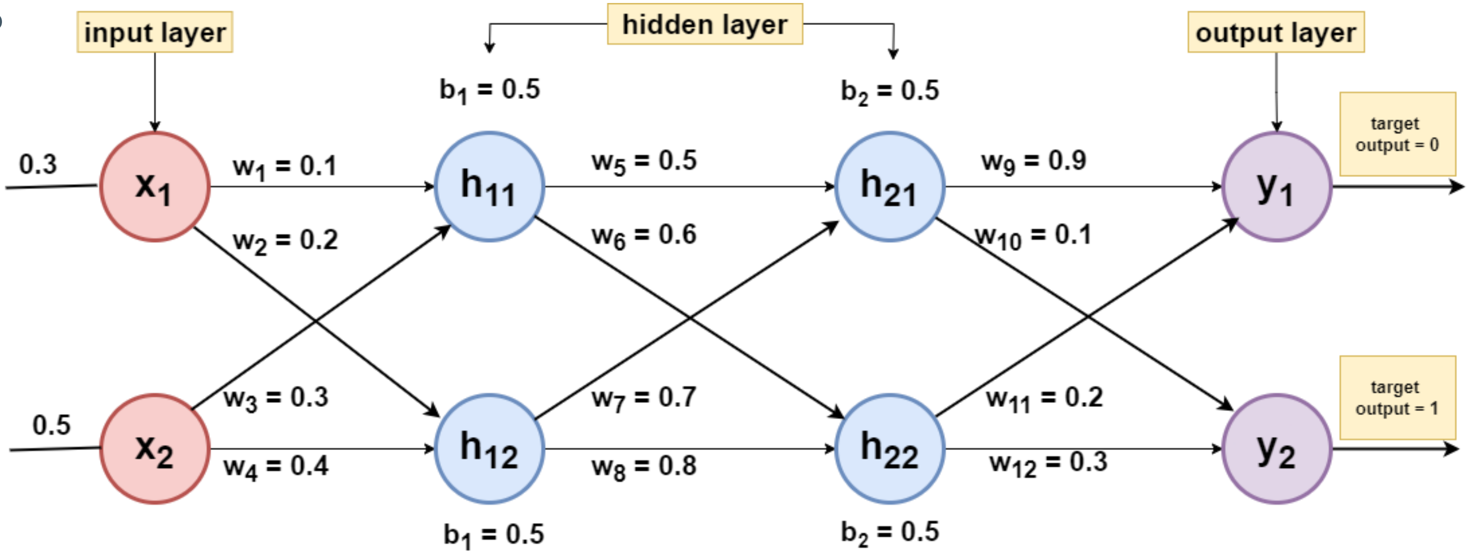
Given,

→ f(y) = sigmoid

→ Learning rate (η) = 0.1

→ Total error(E) = 0.336

→ Target $T_1 = 0$ and $T_2 = 1$



Formula to update weights:-

$$W_{new} = W_{old} - \eta * \frac{\partial E(w)}{\partial w_{old}}$$

For update W_9:-

$$W_{9'} = W_9 - \eta * \frac{\partial E(w)}{\partial w_9}$$

For update W_{11}:-

$$W_{10'} = W_{10} - \eta * \frac{\partial E(w)}{\partial w_{10}}$$

For update W_{11}:-

$$W_{11'} = W_{11} - \eta * \frac{\partial E(w)}{\partial w_{11}}$$

For update W_{12}:-

$$W_{12'} = W_{12} - \eta * \frac{\partial E(w)}{\partial w_{12}}$$

Rough:

$$\frac{\partial E}{\partial w_{12}} = W12 \rightarrow y2 \rightarrow f(y_2) \rightarrow E$$

Calculus formulas:

$$\frac{d}{dx}(x^n) = nx^{n-1}$$

$$\frac{d}{dx}(x) = 1.x^{1-1} = 1.x^0 = 1\ .\ 1 = 1$$

$$\frac{d}{dx}(x^2) = 2.x^{2-1} = 2.x^1 = 2x$$

For update W_{10}: $$W_{10'} = W_{10} - \eta * \frac{\partial E(w)}{\partial w_{10}} \quad ----------------(i)$$

First Calculate:

$$\frac{\partial E}{\partial w_{10}} = \frac{\partial E}{\partial f(y_2)} \times \frac{\partial f(y_2)}{\partial y_2} \times \frac{\partial y2}{\partial w_{10}} \quad --------------------(ii)$$

$$\frac{\partial E}{\partial f(y_2)} = \frac{\partial}{\partial f(y_2)} \times E$$
$$= \frac{\partial}{\partial f(y_2)} \times ½ \,[\,(f(y1) - T_1)^2 + (f(y2) - T_2)^2\,]$$
$$= ½ \times \frac{\partial}{\partial f(y_2)} \times [\,(f(y1) - T_1)^2 + (f(y2) - T_2)^2\,]$$
$$= ½ \times [\,0 + 2 \times (f(y2) - T_2)^{2-1}\,]$$
$$= ½ \times [2 \times (f(y2) - T_2)\,]$$
$$= ½ \times 2 \times (f(y2) - T_2)$$
$$= f(y2) - T_2$$
$$= 0.579 - 1$$
$$\frac{\partial E}{\partial f(y_2)} = -\,0.421$$

$$\frac{\partial f(y_2)}{\partial y_2} = \frac{\partial}{\partial y_2} \times f(y_2)$$
$$= f(y_2) \times [\,1 - f(y_2)\,]$$
$$= 0.579 \times [1 - 0.579]$$
$$= 0.243$$

$$\frac{\partial y2}{\partial w_{10}} = \frac{\partial}{\partial w_{10}} \times y_2$$
$$= \frac{\partial}{\partial w_{12}} \times [f(h21)*w10 + f(h22)*w12]$$
$$= 0 + f(h21) = 0.786$$

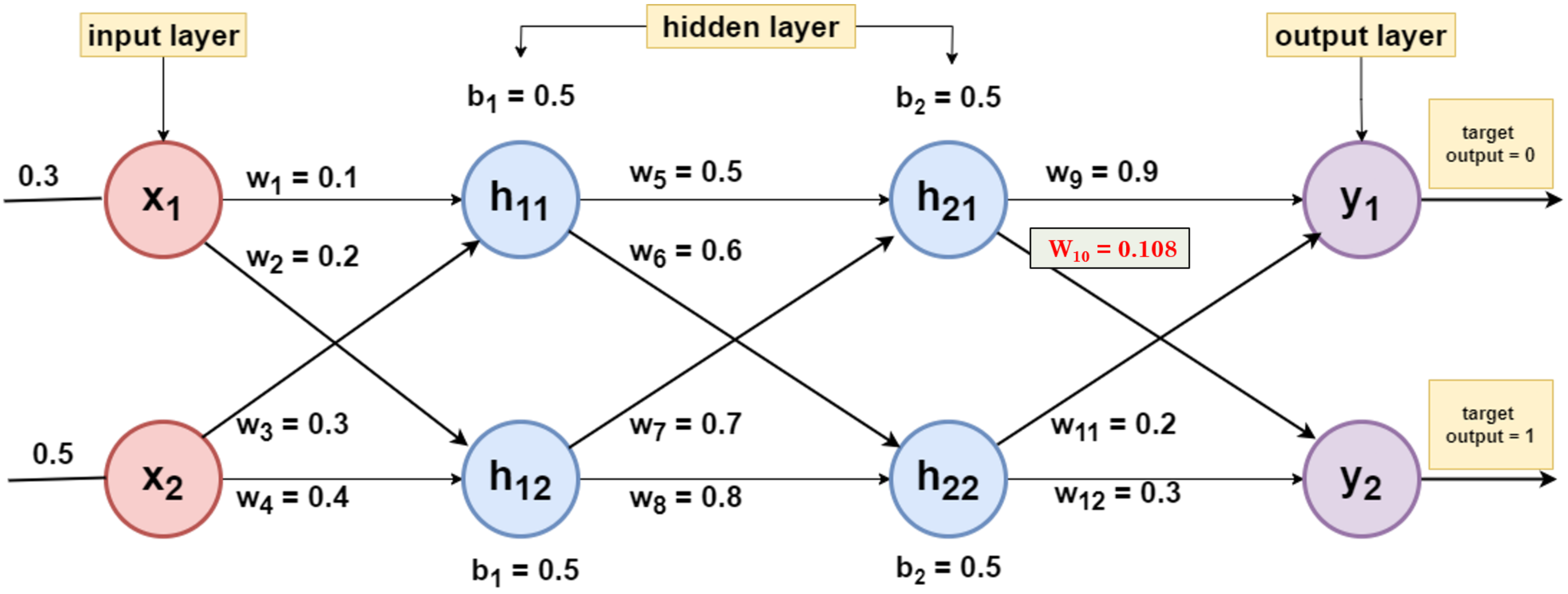
Putting all the values in Equation (ii) :

$$\frac{\partial E}{\partial w_{10}} = \frac{\partial E}{\partial f(y_2)} \times \frac{\partial f(y_2)}{\partial y_2} \times \frac{\partial y2}{\partial w_{10}}$$
$$= (-0.421 * 0.243 * 0.786) = -\,0.08041$$

Putting all the values in Equation (i) :

$$W_{10'} = W_{10} - \eta * \frac{\partial E(w)}{\partial w_{10}} = 0.1 - 0.1 * (-\,0.08041)$$
$$= 0.108041$$
$$= 0.108$$

After updating W_{10} the neural network:



For update W_{12}: $W_{12'} = W_{12} - \eta * \frac{\partial E(w)}{\partial w_{12}}$ ---------------- -(*i*)

First Calculate:

$$\frac{\partial E}{\partial w_{12}} = \frac{\partial E}{\partial f(y_2)} \times \frac{\partial f(y_2)}{\partial y_2} \times \frac{\partial y2}{\partial w_{12}} \quad \text{-------------------- -(}\textit{ii}\text{)}$$

$$\frac{\partial E}{\partial f(y_2)} = \frac{\partial}{\partial f(y_2)} \times E$$
$$= \frac{\partial}{\partial f(y_2)} \times \tfrac{1}{2} [(f(y1) - T_1)^2 + (f(y2) - T_2)^2]$$
$$= \tfrac{1}{2} \times \frac{\partial}{\partial f(y_2)} \times [(f(y1) - T_1)^2 + (f(y2) - T_2)^2]$$
$$= \tfrac{1}{2} \times [0 + 2 \times (f(y2) - T_2)^{2-1}]$$
$$= \tfrac{1}{2} \times [2 \times (f(y2) - T_2)]$$
$$= \tfrac{1}{2} \times 2 \times (f(y2) - T_2)$$
$$= f(y2) - T_2$$
$$= 0.579 - 1$$
$$\frac{\partial E}{\partial f(y_2)} = -0.421$$

$$\frac{\partial f(y_2)}{\partial y_2} = \frac{\partial}{\partial y_2} \times f(y_2)$$
$$= f(y_2) \times [1 - f(y_2)]$$
$$= 0.579 \times [1 - 0.579]$$
$$= 0.243$$

$$\frac{\partial y2}{\partial w_{12}} = \frac{\partial}{\partial w_{12}} \times y_2$$
$$= \frac{\partial}{\partial w_{12}} \times [f(h21)*w10 + f(h22)*w12]$$
$$= 0 + f(h22) = 0.808$$

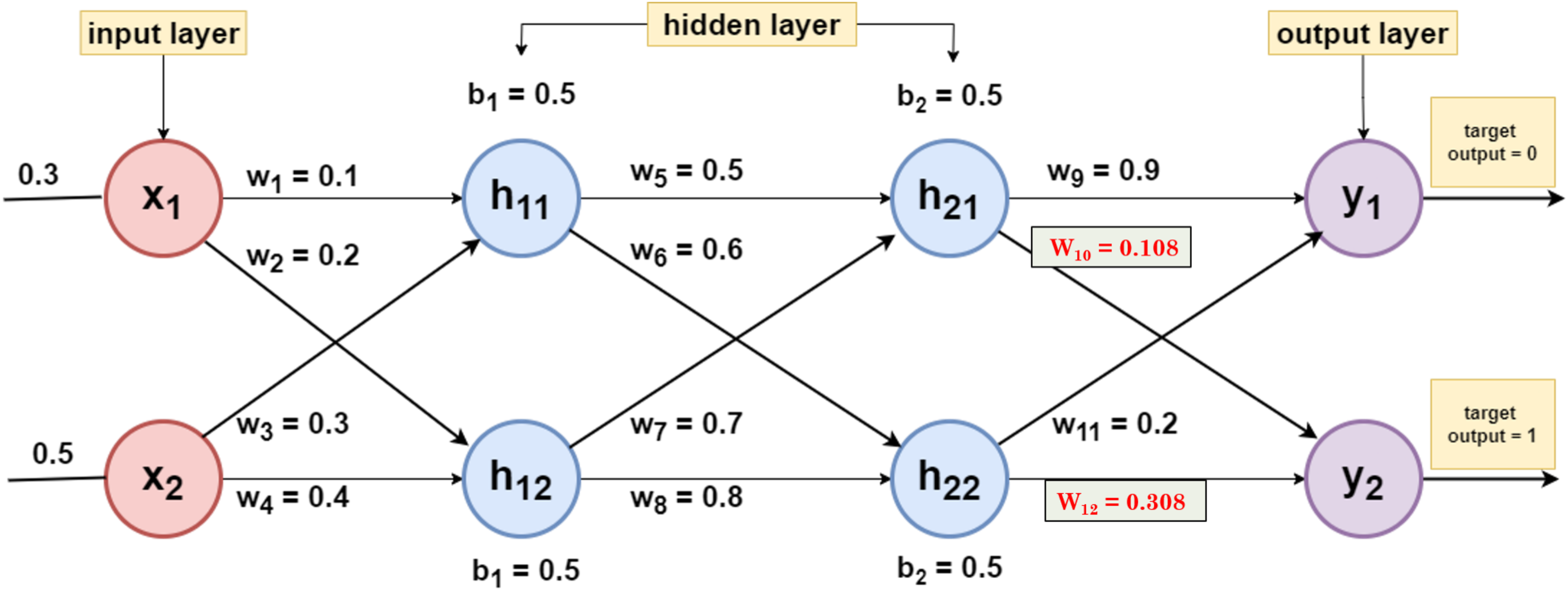
Putting all the values in Equation (ii) :

$$\frac{\partial E}{\partial w_{12}} = \frac{\partial E}{\partial f(y_2)} \times \frac{\partial f(y_2)}{\partial y_2} \times \frac{\partial y2}{\partial w_{12}}$$
$$= (-0.421 * 0.243 * 0.808) = -0.08266$$

Putting all the values in Equation (i) :

$$W_{12'} = W_{12} - \eta * \frac{\partial E(w)}{\partial w_{12}} = 0.3 - 0.1 * (-0.08266)$$
$$= 0.308266$$
$$= 0.308$$

After updating W_{12} the neural network:



<u>For update W_9:</u>

$$W_{9'} = W_9 - \eta * \frac{\partial E(w)}{\partial w_9} \text{ ——————————————— } (i)$$

<u>First Calculate:</u>

$$\frac{\partial E}{\partial w_9} = \frac{\partial E}{\partial f(y_1)} \times \frac{\partial f(y_1)}{\partial y_1} \times \frac{\partial y1}{\partial w_9} \text{ ——————————————————— } (ii)$$

$$\frac{\partial E}{\partial f(y_1)} = \frac{\partial}{\partial f(y_1)} \times E$$
$$= \frac{\partial}{\partial f(y_1)} \times ½ [(f(y1) – T_1)^2 + (f(y2) – T_2)^2]$$
$$= ½ \times \frac{\partial}{\partial f(y_1)} \times [(f(y1) – T_1)^2 + (f(y2) – T_2)^2]$$
$$= ½ \times [2 \times (f(y1) – T_1)^{2-1} + 0]$$
$$= ½ \times [2 \times (f(y1) – T_1)^1]$$
$$= ½ \times 2 \times (f(y1) – T_1)$$
$$= f(y1) – T_1$$
$$= 0.704 – 0$$
$$\frac{\partial E}{\partial f(y_2)} = 0.704$$

$$\frac{\partial f(y_1)}{\partial y_1} = \frac{\partial}{\partial y_1} \times f(y_1)$$
$$= f(y_1) \times [1 – f(y_1)]$$
$$= 0.704 \times [1 – 0.704]$$
$$= 0.208$$

$$\frac{\partial y1}{\partial w_9} = \frac{\partial}{\partial w_9} \times y_1$$
$$= \frac{\partial}{\partial w_9} \times [f(h21)*w9 + f(h22)*w11]$$
$$= f(h21) + 0 = 0.786$$

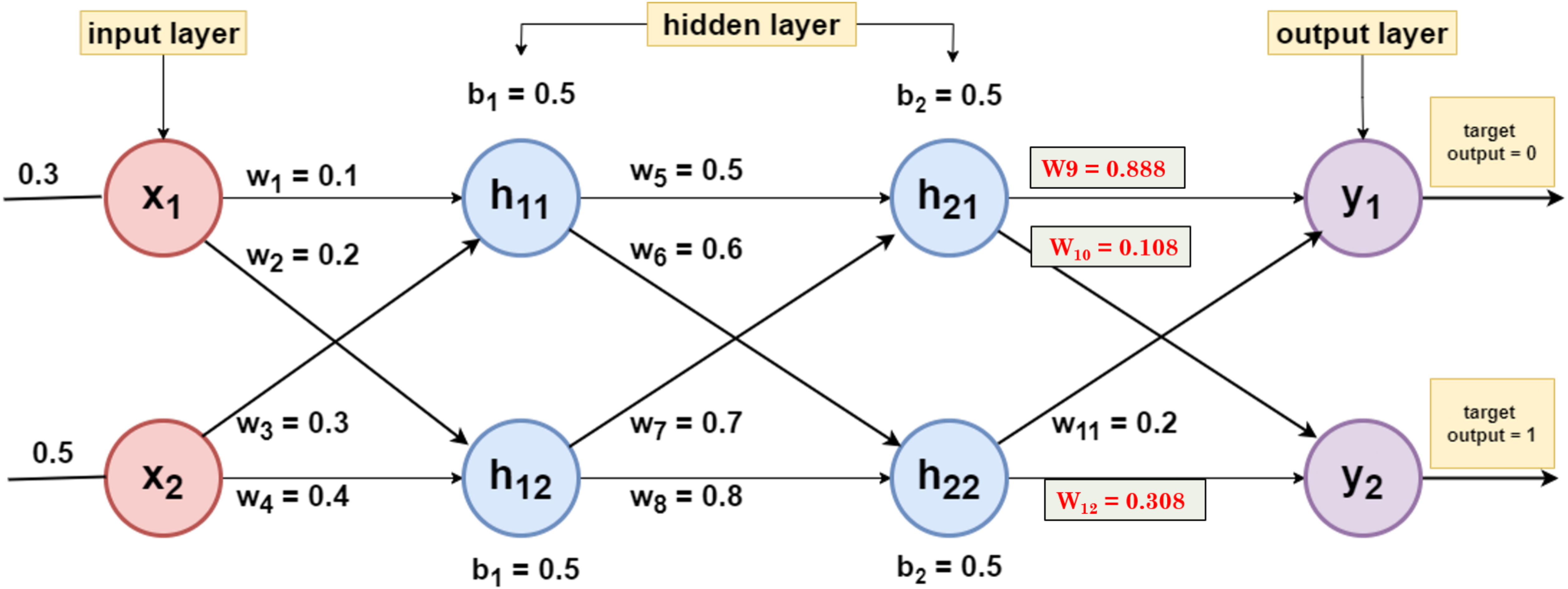
<u>Putting all the values in Equation (ii) :</u>

$$\frac{\partial E}{\partial w_9} = \frac{\partial E}{\partial f(y_1)} \times \frac{\partial f(y_1)}{\partial y_1} \times \frac{\partial y_1}{\partial w_9}$$
$$= (0.704 * 0.208 * 0.786) = 0.115095$$

<u>Putting all the values in Equation (i) :</u>

$$W_{9'} = W_9 - \eta * \frac{\partial E(w)}{\partial w_9} = 0.9 – 0.1 * 0.115095$$
$$= 0.88849$$
$$= 0.888$$

After updating W_9 the neural network:



For update W_{11}:

$$W_{11'} = W_{11} - \eta * \frac{\partial E(w)}{\partial w_{11}} \quad ---------------(i)$$

First Calculate:

$$\frac{\partial E}{\partial w_{11}} = \frac{\partial E}{\partial f(y_1)} \times \frac{\partial f(y_1)}{\partial y_1} \times \frac{\partial y1}{\partial w_{11}} \quad ---------------------(ii)$$

$$\frac{\partial E}{\partial f(y_1)} = \frac{\partial}{\partial f(y_1)} \times E$$

$$= \frac{\partial}{\partial f(y_1)} \times \tfrac{1}{2} [(f(y1) - T_1)^2 + (f(y2) - T_2)^2]$$

$$= \tfrac{1}{2} \times \frac{\partial}{\partial f(y_1)} \times [(f(y1) - T_1)^2 + (f(y2) - T_2)^2]$$

$$= \tfrac{1}{2} \times [2 \times (f(y1) - T_1)^{2-1} + 0]$$

$$= \tfrac{1}{2} \times [2 \times (f(y1) - T_1)^1]$$

$$= \tfrac{1}{2} \times 2 \times (f(y1) - T_1)$$

$$= f(y1) - T_1$$

$$= 0.704 - 0$$

$$\frac{\partial E}{\partial f(y_2)} = 0.704$$

$$\frac{\partial f(y_1)}{\partial y_1} = \frac{\partial}{\partial y_1} \times f(y_1)$$

$$= f(y_1) \times [1 - f(y_1)]$$

$$= 0.704 \times [1 - 0.704]$$

$$= 0.208$$

$$\frac{\partial y1}{\partial w_{11}} = \frac{\partial}{\partial w_{11}} \times y_1$$

$$= \frac{\partial}{\partial w_{11}} \times [f(h21)*w9 + f(h22)*w11]$$

$$= 0 + f(h22) = 0.808$$

Putting all the values in Equation (ii) :

$$\frac{\partial E}{\partial w_{11}} = \frac{\partial E}{\partial f(y_1)} \times \frac{\partial f(y_1)}{\partial y_1} \times \frac{\partial y_1}{\partial w_{11}}$$

$$= (0.704 * 0.208 * 0.808) = 0.118317$$

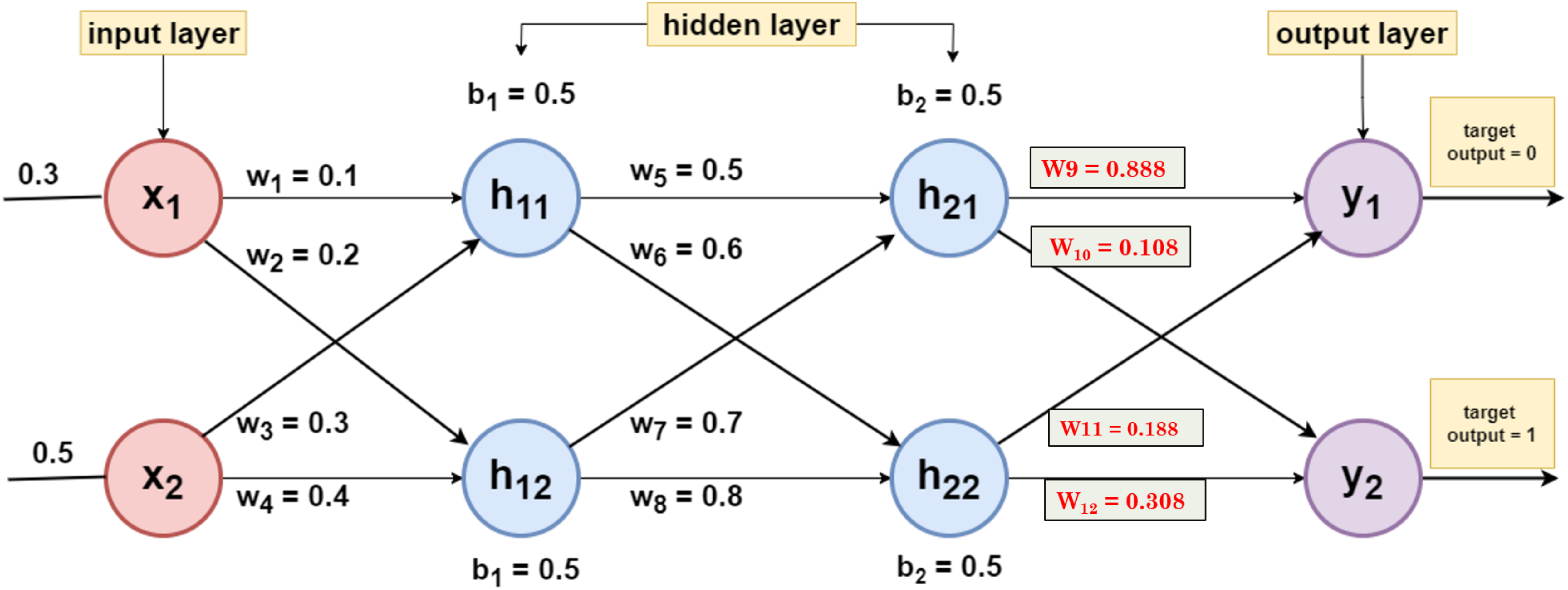
Putting all the values in Equation (i) :

$$W_{11'} = W_{11} - \eta * \frac{\partial E(w)}{\partial w_{11}} = 0.2 - 0.1 * 0.118317$$

$$= 0.188168$$

$$= 0.188$$

After updating W_{11} the neural network:



For update W_5: $$W_{5'} = W_5 - \eta * \frac{\partial E(w)}{\partial w_5} \quad ------------------(i)$$

Rough:

$$\begin{tikzcd} & y1 \arrow[r] & f(y1) \arrow[dr] & \\ \frac{\partial E}{\partial w_5} = W_5 \rightarrow h21 \rightarrow f(h21) \arrow[ur] \arrow[dr] & & & E \\ & y2 \arrow[r] & f(y2) \arrow[ur] & \end{tikzcd}$$

<u>First Calculate:</u>

$$\frac{\partial E}{\partial w_5} = \frac{\partial E}{\partial f(h21)} \times \frac{\partial f(h21)}{\partial h21} \times \frac{\partial h21}{\partial w_5} \quad ---------------------(ii)$$

Where,

$$\frac{\partial E}{\partial f(h21)} = \left[\frac{\partial E}{\partial f(y1)} \times \frac{\partial f(y1)}{\partial y1} \times \frac{\partial y1}{\partial f(h21)}\right] + \left[\frac{\partial E}{\partial f(y2)} \times \frac{\partial f(y2)}{\partial y2} \times \frac{\partial y2}{\partial f(h21)}\right]$$

Calculate this Equation:

$$\frac{\partial E}{\partial f(h21)} = \left[\frac{\partial E}{\partial f(y1)} \times \frac{\partial f(y1)}{\partial y1} \times \frac{\partial y1}{\partial f(h21)}\right] + \left[\frac{\partial E}{\partial f(y2)} \times \frac{\partial f(y2)}{\partial y2} \times \frac{\partial y2}{\partial f(h21)}\right] \text{------------------(iii)}$$

The values from previous calculation is found :

$$\frac{\partial E}{\partial f(y1)} = 0.704 \qquad \frac{\partial E}{\partial f(y2)} = -0.421$$

$$\frac{\partial f(y1)}{\partial y1} = 0.208 \qquad \frac{\partial f(y2)}{\partial y2} = 0.243$$

$$\frac{\partial y1}{\partial f(h21)} = \frac{\partial}{\partial f(h21)} . y1 = \frac{\partial}{\partial f(h21)} . [f(h21)*w9 + f(h22)*w11] = [w9 + 0] = 0.9$$

$$\frac{\partial y2}{\partial f(h21)} = \frac{\partial}{\partial f(h21)} . y2 = \frac{\partial}{\partial f(h21)} . [f(h21)*w10 + f(h22)*w12] = [w10 + 0] = 0.1$$

Putting these values in Equation (iii) we get:

$$\frac{\partial E}{\partial\, f(h21)} = [\frac{\partial E}{\partial\, f(y1)} \text{ x } \frac{\partial\, f(y1)}{\partial\, y1} \text{ x } \frac{\partial y1}{\partial\, f(h21)}] + [\frac{\partial E}{\partial\, f(y2)} \text{ x } \frac{\partial\, f(y2)}{\partial\, y2} \text{ x } \frac{\partial y2}{\partial\, f(h21)}]$$

$$= [0.704 \text{ x } 0.208 \text{ x } 0.9] + [\,-0.421 \text{ x } 0.243 \text{ x } 0.1]$$

$$= 0.1317888 - 0.0102303$$

$$= 0.1215585 = 0.121$$

$$\frac{\partial\, f(h21)}{\partial\, h21} = \frac{\partial}{\partial\, h21}\,.\, f(h21) = f(h21) * [1 - f(h21)] = 0.786 * [1 - 0.786] = 0.208$$

$$\frac{\partial\, h21}{\partial\, w_5} = \frac{\partial}{\partial\, w_5}\,.\, h21 = \frac{\partial}{\partial\, w_5} * [\, f(h11) * w5 + f(h12)*w7 + b2]$$

$$= [\, f(h11) + 0\,]$$

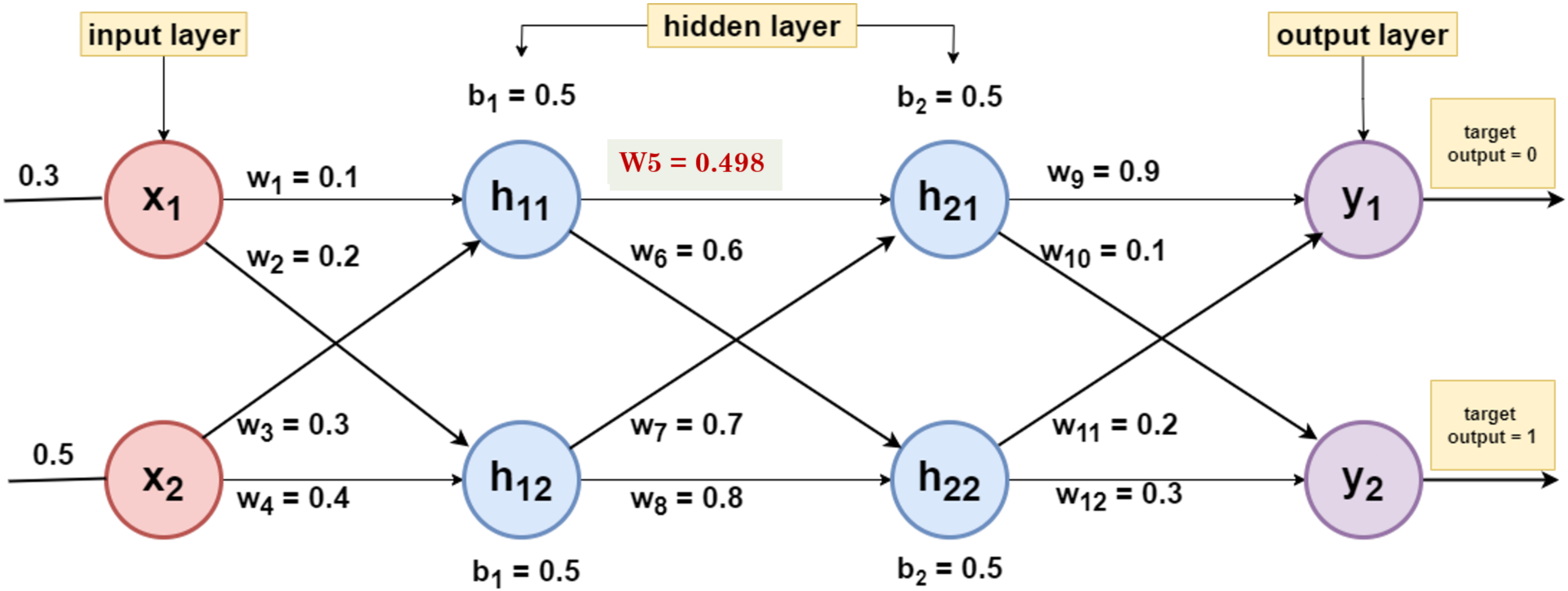
$$= 0.66$$

Putting all the values in Equation(ii) :

$$\frac{\partial E}{\partial w_5} = \frac{\partial E}{\partial\, f(h21)} \text{ x } \frac{\partial\, f(h21)}{\partial\, h21} \text{ x } \frac{\partial\, h21}{\partial\, w_5} = 0.121 \text{ x } 0.208 \text{ x } 0.66 = 0.01661 = 0.016$$

After updating W_5 the neural network:

$$W_{5'} = W_5 - \eta * \frac{\partial E(w)}{\partial w_5} = 0.5 - 0.1 * 0.01661 = 0.498339 = 0.498$$



For update W_7: $$W_{7'} = W_7 - \eta * \frac{\partial E(w)}{\partial w_7} \quad ----------------(i)$$

Rough:

$$\begin{tikzcd}
 & & & y1 \arrow[r] & f(y1) \arrow[dr] & \\
\frac{\partial E}{\partial w_7} = W_7 \arrow[r] & h21 \arrow[r] & f(h21) \arrow[ur] \arrow[dr] & & & E \\
 & & & y2 \arrow[r] & f(y2) \arrow[ur] &
\end{tikzcd}$$

<u>First Calculate:</u>

$$\frac{\partial E}{\partial w_7} = \frac{\partial E}{\partial f(h21)} \times \frac{\partial f(h21)}{\partial h21} \times \frac{\partial h21}{\partial w_7} \quad ------------------(ii)$$

Where,

$$\frac{\partial E}{\partial f(h21)} = \left[\frac{\partial E}{\partial f(y1)} \times \frac{\partial f(y1)}{\partial y1} \times \frac{\partial y1}{\partial f(h21)}\right] + \left[\frac{\partial E}{\partial f(y2)} \times \frac{\partial f(y2)}{\partial y2} \times \frac{\partial y2}{\partial f(h21)}\right]$$

Calculate this Equation:

$$\frac{\partial E}{\partial f(h21)} = [\frac{\partial E}{\partial f(y1)} \text{ x } \frac{\partial f(y1)}{\partial y1} \text{ x } \frac{\partial y1}{\partial f(h21)}] + [\frac{\partial E}{\partial f(y2)} \text{ x } \frac{\partial f(y2)}{\partial y2} \text{ x } \frac{\partial y2}{\partial f(h21)}] \text{ ----------------- (iii)}$$

The values from previous calculation is found :

$$\frac{\partial E}{\partial f(y1)} = 0.704 \qquad \frac{\partial E}{\partial f(y2)} = -0.421$$

$$\frac{\partial f(y1)}{\partial y1} = 0.208 \qquad \frac{\partial f(y2)}{\partial y2} = 0.243$$

$$\frac{\partial y1}{\partial f(h21)} = \frac{\partial}{\partial f(h21)} . y1 = \frac{\partial}{\partial f(h21)} . [f(h21)*w9 + f(h22)*w11] = [w9 + 0] = 0.9$$

$$\frac{\partial y2}{\partial f(h21)} = \frac{\partial}{\partial f(h21)} . y2 = \frac{\partial}{\partial f(h21)} . [f(h21)*w10 + f(h22)*w12] = [w10 + 0] = 0.1$$

Putting these values in Equation (iii) we get:

$$\frac{\partial E}{\partial\, f(h21)} = [\frac{\partial E}{\partial\, f(y1)} \text{ x } \frac{\partial\, f(y1)}{\partial\, y1} \text{ x } \frac{\partial y1}{\partial\, f(h21)}] + [\frac{\partial E}{\partial\, f(y2)} \text{ x } \frac{\partial\, f(y2)}{\partial\, y2} \text{ x } \frac{\partial y2}{\partial\, f(h21)}]$$

$$= [0.704 \text{ x } 0.208 \text{ x } 0.9] + [\,-0.421 \text{ x } 0.243 \text{ x } 0.1]$$

$$= 0.1317888 - 0.0102303$$

$$= 0.1215585 = 0.121$$

$$\frac{\partial\, f(h21)}{\partial\, h21} = \frac{\partial}{\partial\, h21}\,.\, f(h21) = f(h21) * [1 - f(h21)] = 0.786 * [1 - 0.786] = 0.208$$

$$\frac{\partial\, h21}{\partial\, w_7} = \frac{\partial}{\partial\, w_7}\,.\, h21 = \frac{\partial}{\partial\, w_7} * [\, f(h11) * w5 + f(h12)*w7 + b2]$$

$$= [\, 0 + f(h12)\,]$$

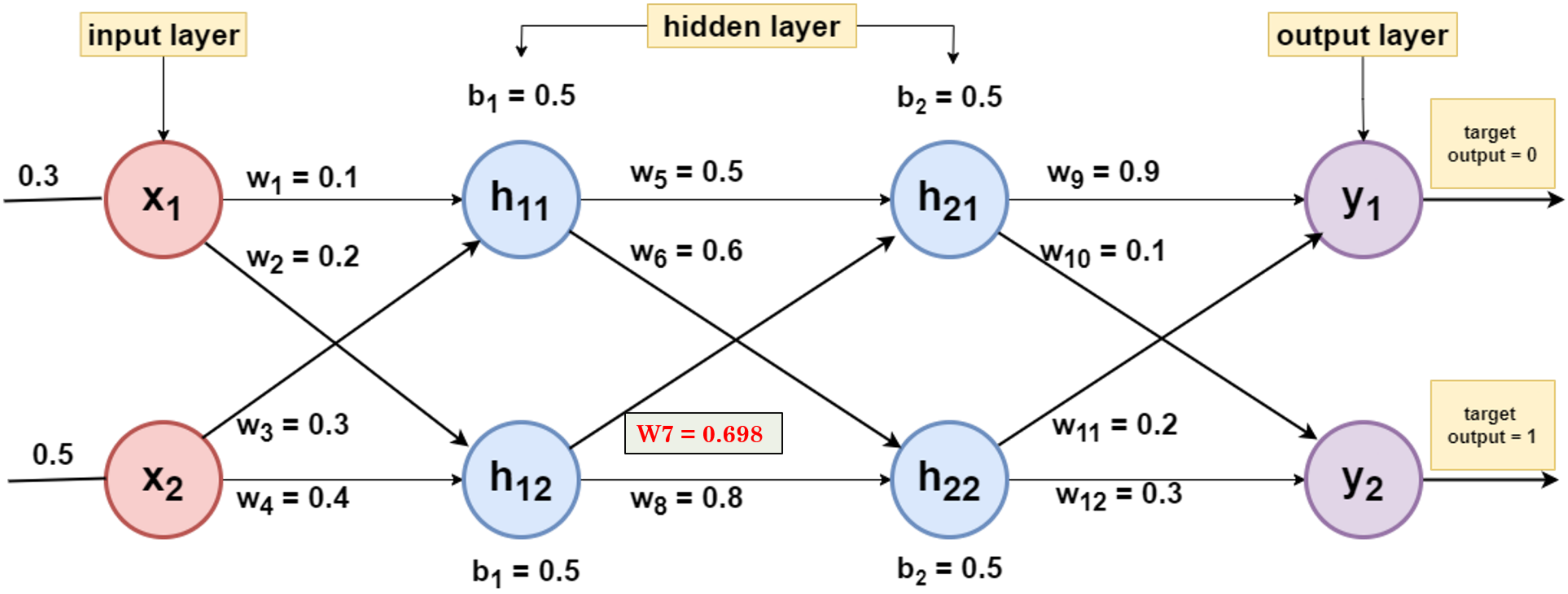
$$= 0.68$$

Putting all the values in Equation(ii) :

$$\frac{\partial E}{\partial w_7} = \frac{\partial E}{\partial\, f(h21)} \text{ x } \frac{\partial\, f(h21)}{\partial\, h21} \text{ x } \frac{\partial\, h21}{\partial\, w_7} = 0.121 \text{ x } 0.208 \text{ x } 0.68 = 0.017114 = 0.017$$

After updating W_7 the neural network:

$$W_{7'} = W_7 - \eta * \frac{\partial E(w)}{\partial w_7} = 0.7 - 0.1 * 0.017114 = 0.698288 = 0.698$$



<u>For update W_6:</u> $$W_{6'} = W_6 - \eta * \frac{\partial E(w)}{\partial w_6} \quad ----------------(i)$$

Rough:

$$\frac{\partial E}{\partial w_6} = W_6 \rightarrow h22 \rightarrow f(h22) \begin{matrix} \nearrow y1 \rightarrow f(y1) \searrow \\ \searrow y2 \rightarrow f(y2) \nearrow \end{matrix} E$$

<u>First Calculate:</u>

$$\frac{\partial E}{\partial w_6} = \frac{\partial E}{\partial f(h22)} \times \frac{\partial f(h22)}{\partial h22} \times \frac{\partial h22}{\partial w_6} \quad ---------------------(ii)$$

Where,

$$\frac{\partial E}{\partial f(h22)} = [\frac{\partial E}{\partial f(y1)} \times \frac{\partial f(y1)}{\partial y1} \times \frac{\partial y1}{\partial f(h22)}] + [\frac{\partial E}{\partial f(y2)} \times \frac{\partial f(y2)}{\partial y2} \times \frac{\partial y2}{\partial f(h22)}]$$

Calculate this Equation:

$$\frac{\partial E}{\partial f(h22)} = [\frac{\partial E}{\partial f(y1)} \times \frac{\partial f(y1)}{\partial y1} \times \frac{\partial y1}{\partial f(h22)}] + [\frac{\partial E}{\partial f(y2)} \times \frac{\partial f(y2)}{\partial y2} \times \frac{\partial y2}{\partial f(h22)}] \text{-----------------}(iii)$$

The values from previous calculation is found :

$$\frac{\partial E}{\partial f(y1)} = 0.704 \qquad \frac{\partial E}{\partial f(y2)} = -0.421$$

$$\frac{\partial f(y1)}{\partial y1} = 0.208 \qquad \frac{\partial f(y2)}{\partial y2} = 0.243$$

$$\frac{\partial y1}{\partial f(h22)} = \frac{\partial}{\partial f(h22)} . y1 = \frac{\partial}{\partial f(h22)} . [f(h21)*w9 + f(h22)*w11] =[w11 + 0] = 0.2$$

$$\frac{\partial y2}{\partial f(h22)} = \frac{\partial}{\partial f(h22)} . y2 = \frac{\partial}{\partial f(h22)} . [f(h21)*w10 + f(h22)*w12] =[w12 + 0] = 0.3$$

Putting these values in Equation (iii) we get:

$$\frac{\partial E}{\partial f(h22)} = [\frac{\partial E}{\partial f(y1)} \text{ x } \frac{\partial f(y1)}{\partial y1} \text{ x } \frac{\partial y1}{\partial f(h22)}] + [\frac{\partial E}{\partial f(y2)} \text{ x } \frac{\partial f(y2)}{\partial y2} \text{ x } \frac{\partial y2}{\partial f(h22)}]$$

$$= [0.704 \text{ x } 0.208 \text{ x } 0.2] + [-0.421 \text{ x } 0.243 \text{ x } 0.3]$$

$$= 0.029286 – 0.03069$$

$$= - 0.00137$$

$$\frac{\partial f(h22)}{\partial h22} = \frac{\partial}{\partial h22} . f(h22) = f(h22) * [1 – f(h22)] = 0.808 * [1 – 0.808] = 0.155136$$

$$\frac{\partial h22}{\partial w_6} = \frac{\partial}{\partial w_6} . h22 = \frac{\partial}{\partial w_6} * [f(h11) * w6 + f(h12)*w8+b2]$$

$$= [f(h11) + 0]$$

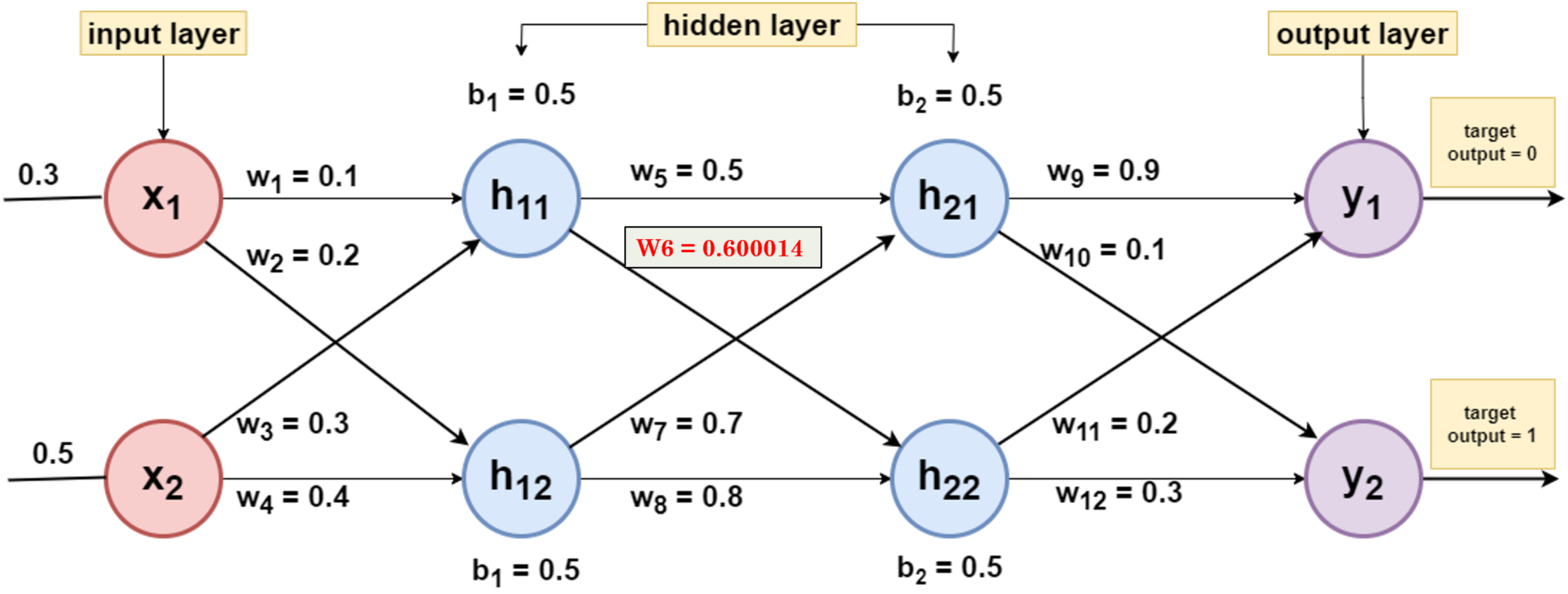
$$= 0.66$$

Putting all the values in Equation(ii) :

$$\frac{\partial E}{\partial w_7} = \frac{\partial E}{\partial f(h22)} \text{ x } \frac{\partial f(h22)}{\partial h22} \text{ x } \frac{\partial h22}{\partial w_7} = - 0.00137 \text{ x } 0.155 \text{ x } 0.66 = -0.00014 = -0.00014$$

After updating W_6 the neural network:

$$W_{6'} = W_6 - \eta * \frac{\partial E(w)}{\partial w_6} = 0.6 - 0.1 * (-0.00014) = 0.600014$$



<u>For update W_8:</u> $$W_{8'} = W_8 - \eta * \frac{\partial E(w)}{\partial w_8} \text{ ----------------- } (i)$$

Rough:

$$\frac{\partial E}{\partial w_8} = W_8 \rightarrow h22 \rightarrow$$
$$\begin{tikzcd}[column sep=small] & & y1 \arrow[r] & f(y1) \arrow[dr] & \\ f(h22) \arrow[urr] \arrow[drr] & & & & E \\ & & y2 \arrow[r] & f(y2) \arrow[ur] & \end{tikzcd}$$

<u>First Calculate:</u>

$$\frac{\partial E}{\partial w_8} = \frac{\partial E}{\partial\, f(h22)} \times \frac{\partial\, f(h22)}{\partial\, h22} \times \frac{\partial\, h22}{\partial\, w_8} \text{ -------------------- } (ii)$$

Where,

$$\frac{\partial E}{\partial\, f(h22)} = \left[\frac{\partial E}{\partial\, f(y1)} \times \frac{\partial\, f(y1)}{\partial\, y1} \times \frac{\partial y1}{\partial\, f(h22)}\right] + \left[\frac{\partial E}{\partial\, f(y2)} \times \frac{\partial\, f(y2)}{\partial\, y2} \times \frac{\partial y2}{\partial\, f(h22)}\right]$$

Calculate this Equation:

$$\frac{\partial E}{\partial f(h22)} = \left[\frac{\partial E}{\partial f(y1)} \times \frac{\partial f(y1)}{\partial y1} \times \frac{\partial y1}{\partial f(h22)}\right] + \left[\frac{\partial E}{\partial f(y2)} \times \frac{\partial f(y2)}{\partial y2} \times \frac{\partial y2}{\partial f(h22)}\right] \text{-----------------} (iii)$$

The values from previous calculation is found :

$$\frac{\partial E}{\partial f(y1)} = 0.704 \qquad \frac{\partial E}{\partial f(y2)} = -0.421$$

$$\frac{\partial f(y1)}{\partial y1} = 0.208 \qquad \frac{\partial f(y2)}{\partial y2} = 0.243$$

$$\frac{\partial y1}{\partial f(h22)} = \frac{\partial}{\partial f(h22)} . y1 = \frac{\partial}{\partial f(h22)} . [f(h21)*w9 + f(h22)*w11] = [\ w11 + 0\ ] = 0.2$$

$$\frac{\partial y2}{\partial f(h22)} = \frac{\partial}{\partial f(h22)} . y2 = \frac{\partial}{\partial f(h22)} . [f(h21)*w10 + f(h22)*w12] = [\ w12 + 0\ ] = 0.3$$

Putting these values in Equation (iii) we get:

$$\frac{\partial E}{\partial\, f(h22)} = [\frac{\partial E}{\partial\, f(y1)} \text{ x } \frac{\partial\, f(y1)}{\partial\, y1} \text{ x } \frac{\partial y1}{\partial\, f(h22)}] + [\frac{\partial E}{\partial\, f(y2)} \text{ x } \frac{\partial\, f(y2)}{\partial\, y2} \text{ x } \frac{\partial y2}{\partial\, f(h22)}]$$

$$= [0.704 \text{ x } 0.208 \text{ x } 0.2] + [\,-0.421 \text{ x } 0.243 \text{ x } 0.3]$$

$$= 0.029286 - 0.03069$$

$$= -\ 0.00137$$

$$\frac{\partial\, f(h22)}{\partial\, h22} = \frac{\partial}{\partial\, h22}\,.\, f(h22) = f(h22) * [1 - f(h22)] = 0.808 * [1 - 0.808] = 0.155136$$

$$\frac{\partial\, h22}{\partial\, w_8} = \frac{\partial}{\partial\, w_8}\,.\, h22 = \frac{\partial}{\partial\, w_8} * [\, f(h11) * w6 + f(h12)*w8+b2]$$

$$= [\, 0 + f(h12)\,]$$

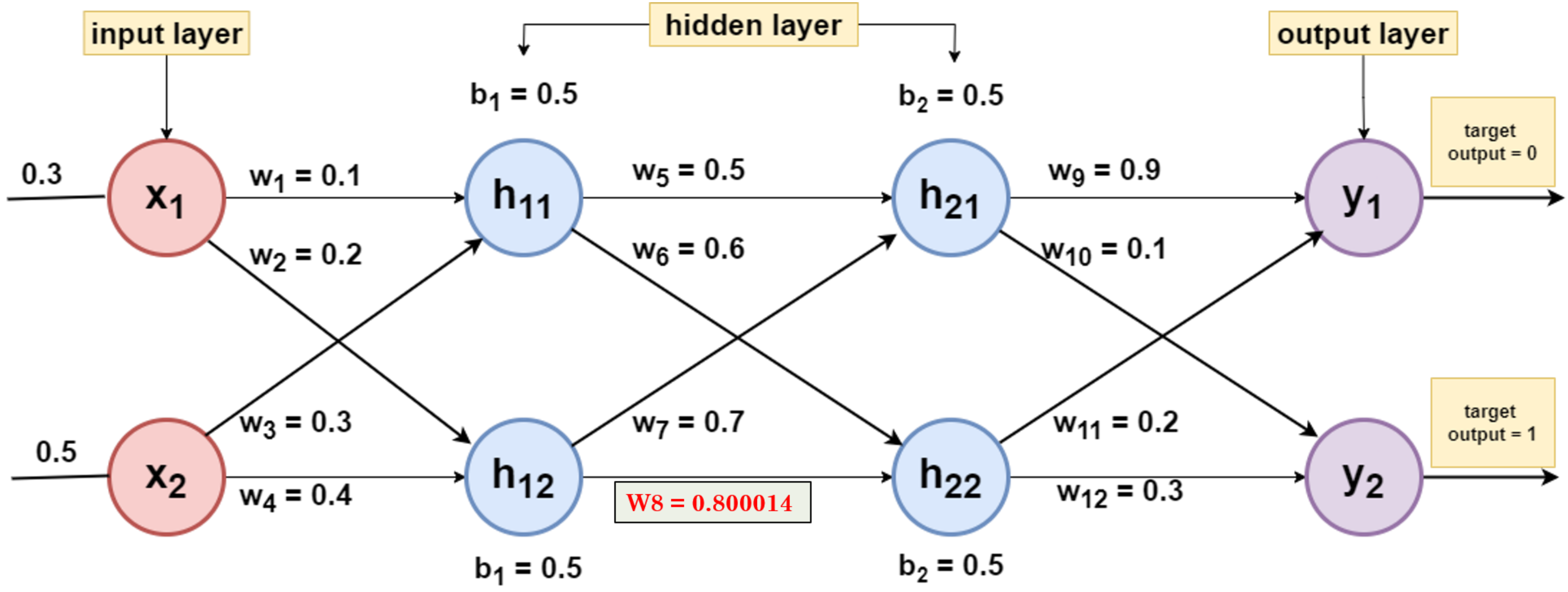
$$= 0.68$$

Putting all the values in Equation(ii) :

$$\frac{\partial E}{\partial w_7} = \frac{\partial E}{\partial\, f(h22)} \text{ x } \frac{\partial\, f(h22)}{\partial\, h22} \text{ x } \frac{\partial\, h22}{\partial\, w_7} = -\ 0.00137 \text{ x } 0.155 \text{ x } 0.68 = -0.00014 = -0.00014$$

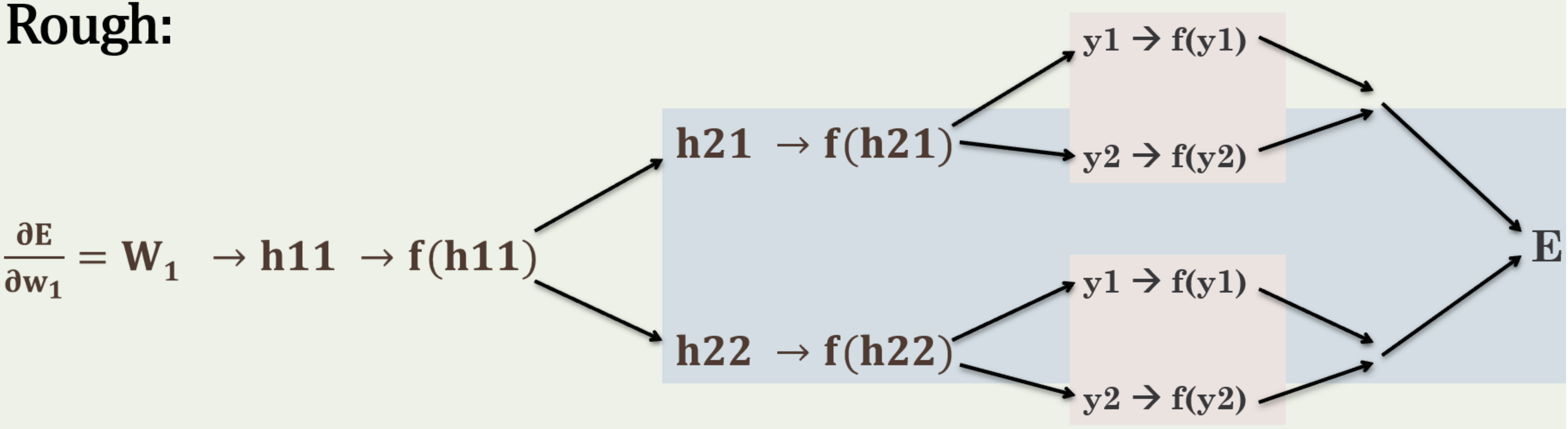
After updating W_8 the neural network:

$$W_{8'} = W_8 - \eta * \frac{\partial E(w)}{\partial w_8} = 0.8 - 0.1 * (-0.00014) = 0.800014$$



<u>For update</u> $\mathbf{W_1}$**:** $$W_{1'} = W_1 - \eta * \frac{\partial E(w)}{\partial w_1} \quad ----------------(i)$$

Rough:



First Calculate:

$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial f(h11)} \text{ x } \frac{\partial f(h11)}{\partial h11} \text{ x } \frac{\partial h11}{\partial w_1} \text{ ------------------- --(ii)}$$

Where,

$$\frac{\partial E}{\partial f(h11)} = \left[\frac{\partial E}{\partial f(h21)} \text{ x } \frac{\partial f(h21)}{\partial h21} \text{ x } \frac{\partial h21}{\partial f(h11)} \right] + \left[\frac{\partial E}{\partial f(h22)} \text{ x } \frac{\partial f(h22)}{\partial h22} \text{ x } \frac{\partial h22}{\partial f(h11)} \right] \text{ -----------------(iii)}$$

Where,

$$\frac{\partial E}{\partial f(h21)} = \left[\frac{\partial E}{\partial f(y1)} \text{ x } \frac{\partial f(y1)}{\partial y1} \text{ x } \frac{\partial y1}{\partial f(h22)}\right] + \left[\frac{\partial E}{\partial f(y2)} \text{ x } \frac{\partial f(y2)}{\partial y2} \text{ x } \frac{\partial y2}{\partial f(h22)}\right] \text{ -----------(iv)}$$

And,

$$\frac{\partial E}{\partial f(h22)} = \left[\frac{\partial E}{\partial f(y1)} \text{ x } \frac{\partial f(y1)}{\partial y1} \text{ x } \frac{\partial y1}{\partial f(h22)}\right] + \left[\frac{\partial E}{\partial f(y2)} \text{ x } \frac{\partial f(y2)}{\partial y2} \text{ x } \frac{\partial y2}{\partial f(h22)}\right] \text{ -------------(v)}$$

Calculate Eq (iv): $\frac{\partial E}{\partial f(h21)} = 0.1215585$ Calculate Eq (v): $\frac{\partial E}{\partial f(h22)} = -0.00137$

<u>Now for Eq (iii) :</u>

$$\frac{\partial h21}{\partial f(h11)} = \frac{\partial}{\partial f(h11)} . h21 = \frac{\partial}{\partial f(h11)} . [f(h11)*w5 + f(h21)*w7 + b2]$$
$$= [w5 + 0 + 0] = 0.5$$

And,

$$\frac{\partial h22}{\partial f(h11)} = \frac{\partial}{\partial f(h11)} . h22 = \frac{\partial}{\partial f(h11)} . [f(h11)*w6 + f(h21)*w8 + b2]$$
$$= [w6 + 0 + 0] = 0.6$$

Calculate Eq (iii):

$$\frac{\partial E}{\partial f(h11)} = \left[\frac{\partial E}{\partial f(h21)} \times \frac{\partial f(h21)}{\partial h21} \times \frac{\partial h21}{\partial f(h11)} \right] + \left[\frac{\partial E}{\partial f(h22)} \times \frac{\partial f(h22)}{\partial h22} \times \frac{\partial h22}{\partial f(h11)} \right]$$
$$= [0.121 \times 0.208 \times 0.5] + [-0.0013 \times 0.155 \times 0.6] = 0.0124$$

Now for Eq (ii) :

$$\frac{\partial\, f(h11)}{\partial\, h11} = f(h11)\ .\ [1 - f(h11)] = 0.66 * [1 - 0.66] = 0.2244$$

$$\frac{\partial\, h11}{\partial\, w_1} = \frac{\partial}{\partial\, w_1}\ .\ [x1 * w1 + x2 * w3 + b1] = x1 + 0 + 0 = 0.3$$

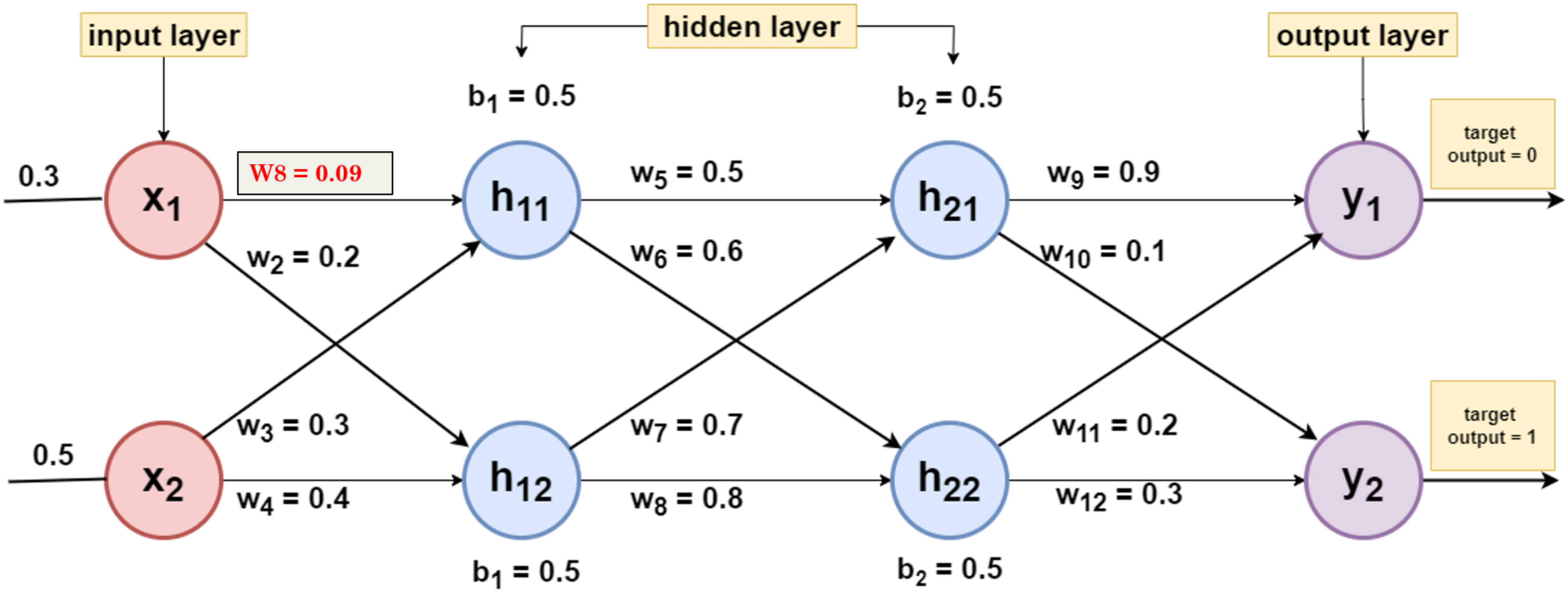
Now calculate Eq(ii):

$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial\, f(h11)}\ \text{x}\ \frac{\partial\, f(h11)}{\partial\, h11}\ \text{x}\ \frac{\partial\, h11}{\partial\, w_1} = 0.0124\ \text{x}\ 0.2244\ \text{x}\ 0.3 = 0.000834$$

Finally the update values for W_1:

$$W_{1'} = W_1 - \eta * \frac{\partial E(w)}{\partial w_1} = 0.1 - 0.1 * (0.000834) = 0.0999166 = 0.09$$

After updating W_1 the neural network:

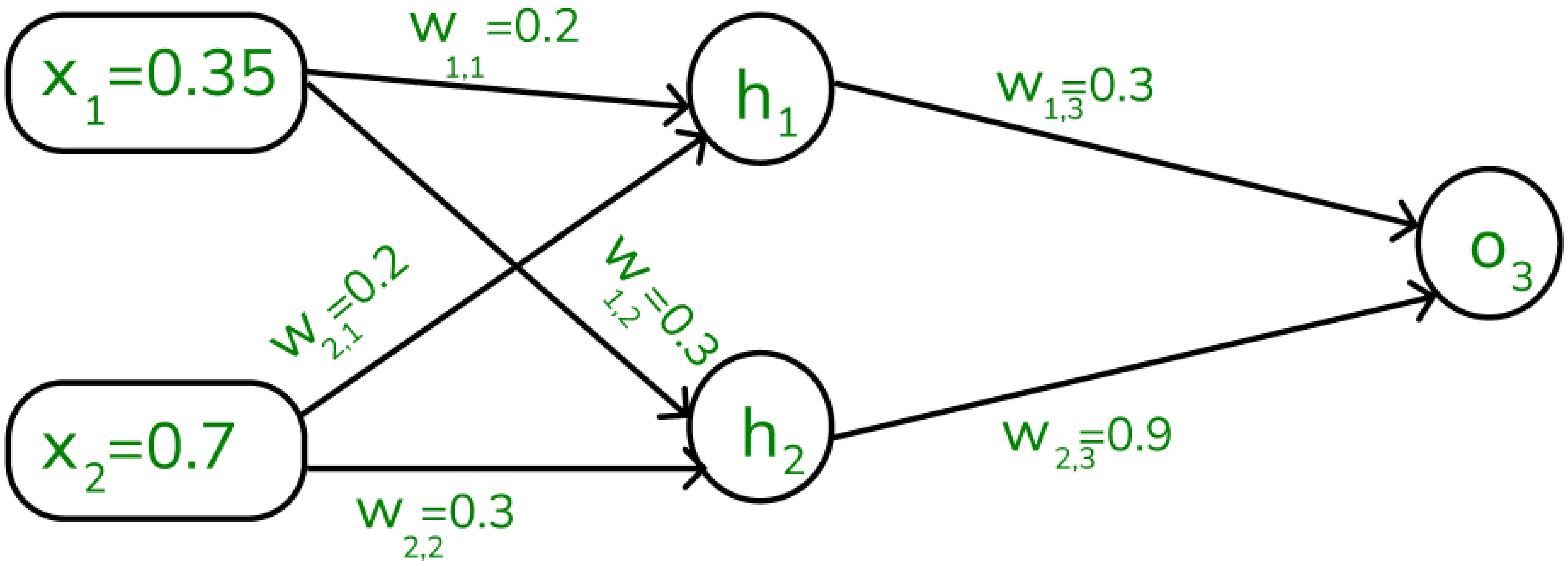


Solve w2, w3, w4 in this similar way.

Example-02: **Backpropagation in Machine Learning**

Let us now take an example to explain backpropagation in Machine Learning,

Assume that the neurons have the sigmoid activation function to perform forward and backward pass on the network. And also assume that the actual output of y is 0.5 and the learning rate is 1. Now perform the backpropagation using backpropagation algorithm.



Implementing forward propagation:

Step1: Before proceeding to calculating forward propagation, we need to know the two formulae:

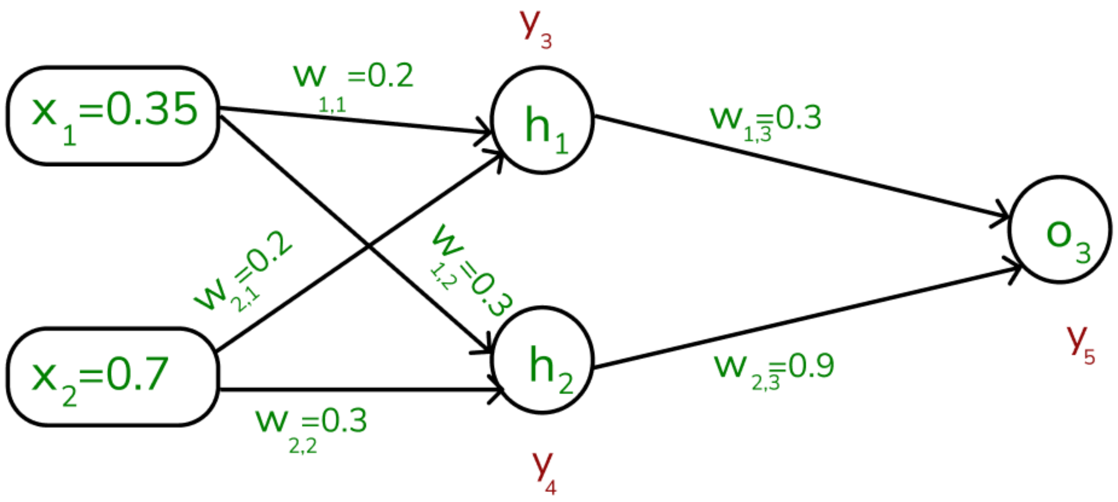
$a_j = \sum(w_i, j * x_i)$

Where,

- a_j is the weighted sum of all the inputs and weights at each node,
- wi,j – represents the weights associated with the j^{th} input to the i^{th} neuron,
- x_i – represents the value of the j^{th} input,

$y_j = F(a_j) = \frac{1}{1+e^{-a_j}}$, y_i – is the output value, F denotes the activation function [sigmoid activation function is used here), which transforms the weighted sum into the output value.

Step 2: To compute the forward pass, we need to compute the output for y_3 , y_4 , and y_5.



We start by calculating the weights and inputs by using the formula:

$a_j = \sum(w_{i,j} * x_i)$ To find y_3 , we need to consider its incoming edges along with its weight and input. Here the incoming edges are from X_1 and X_2.

At h1 node,

$$a_1 = (w_{1,1}x_1) + (w_{2,1}x_2)$$
$$= (0.2 * 0.35) + (0.2 * 0.7)$$
$$= 0.21$$

Once, we calculated the a_1 value, we can now proceed to find the y_3 value:

$$y_j = F(a_j) = \frac{1}{1+e^{-aj}}$$

$$y_3 = F(0.21) = \frac{1}{1+e^{-0.21}}$$

$$y_3 = 0.56$$

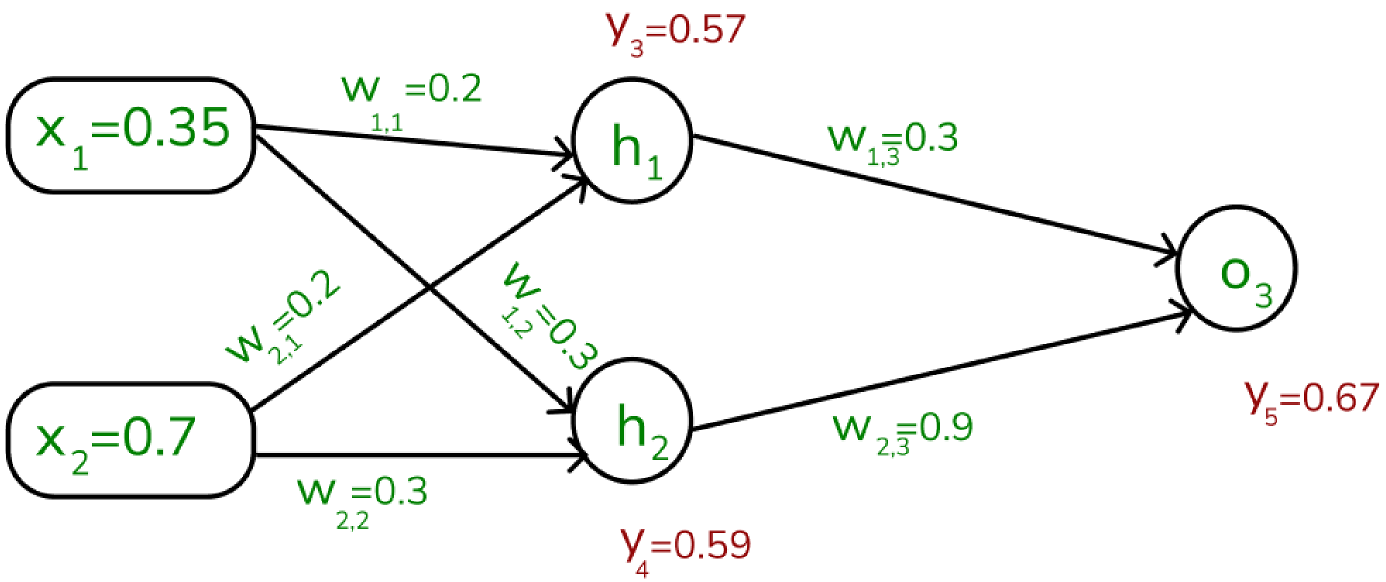
Similarly find the values of y_4 at $\mathbf{h_2}$ and y_5 at O_3 ,

$$a2 = (w_{1,2} * x_1) + (w_{2,2} * x_2) = (0.3 * 0.35) + (0.3 * 0.7) = 0.315$$

$$y_4 = F(0.315) = \frac{1}{1+e^{-0.315}}$$

$$a3 = (w_{1,3} * y_3) + (w_{2,3} * y_4) = (0.3 * 0.57) + (0.9 * 0.59) = 0.702$$

$$y_5 = F(0.702) = \frac{1}{1+e^{-0.702}} = 0.67$$



Note that, our actual output is 0.5 but we obtained 0.67. To calculate the error, we can use the below formula:

$$Error_j = y_{target} - y_5$$

Error = 0.5 – 0.67

= -0.17

Using this error value, we will be backpropagating.

Implementing Backward Propagation

Step 3: To calculate the backpropagation, we need to start from the output unit:

To compute the $?_5$, we need to use the output of forward pass,

$?_5 = y_5(1-y_5)\ (y_{target} - y_5)$

$= 0.67(1-0.67)\ (-0.17)$

$= -0.0376$

For hidden unit,

To compute the hidden unit, we will take the value of $?_5$

$?_3 = y_3(1-y_3)\ (w_{1,3} * ?_5)$

$=0.56(1-0.56)\ (0.3*-0.0376)$

$=-0.0027$

$?_4 = y_4\ (1-y_5)\ (w_{2,3} * ?_5)$

$=0.59(1-0.59)\ (0.9*-0.0376)$

$=-0.0819$

Step 4: We need to update the weights, from output unit to hidden unit,

Note- Here our learning rate is 1

$\nabla\ w_{2,3} = \eta\ ?_5\ O_4$

$= 1 * (-0.376) * 0.59$

$= -0.22184$

We will be updating the weights based on the old weight of the network,

$w_{2,3}(new) = \nabla\ w_{4,5} + w_{4,5}\ (old)$

$= -0.22184 + 0.9$

$= 0.67816$

From hidden unit to input unit,

For an hidden to input node, we need to do calculations by the following;

$\nabla\ w_{1,1} = \eta\ ?_3\ O_4$

$= 1 * (-0.0027) * 0.35$

$= 0.000945$

Similarly, we need to calculate the new weight value using the old one:

$w_{1,1}(new) = \nabla w_{1,1} + w_{1,1}\ (old)$

$= 0.000945 + 0.2$

$= 0.200945$

Similarly, we update the weights of the other neurons: The new weights are mentioned below

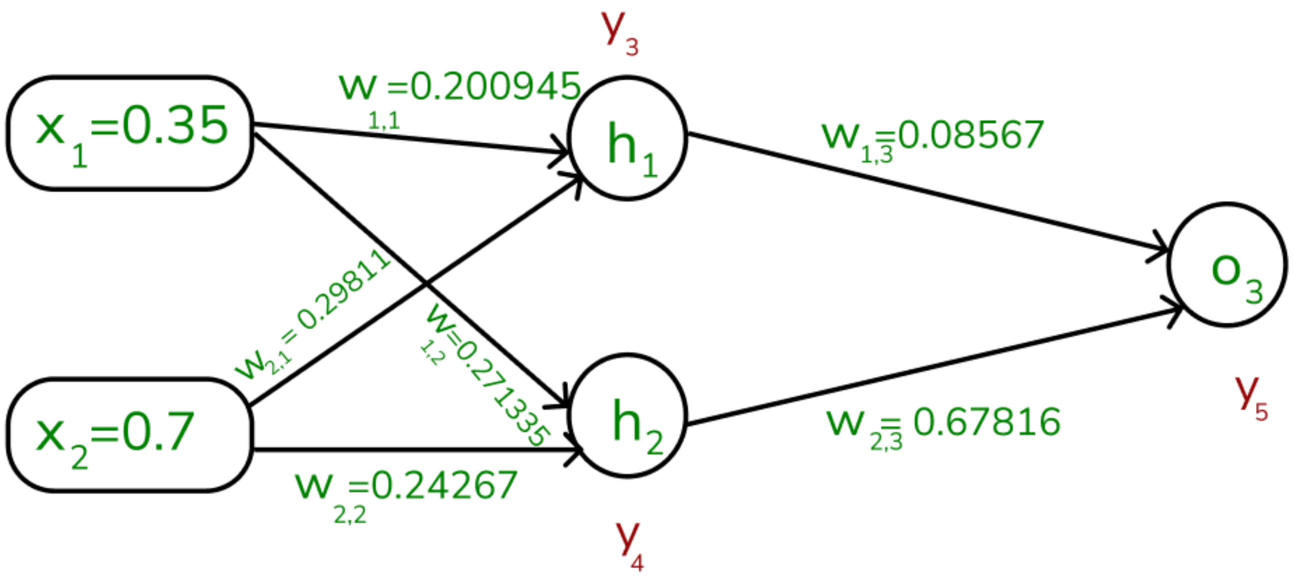
$w_{1,2}\ (new) = 0.271335$

$w_{1,3}\ (new) = 0.08567$

$w_{2,1}\ (new) = 0.29811$

$w_{2,2}\ (new) = 0.24267$

The updated weights are illustrated below,



Once, the above process is done, we again perform the forward pass to find if we obtain the actual output as 0.5.

While performing the forward pass again, we obtain the following values:

$y_3 = 0.57$

$y_4 = 0.56$

$y_5 = 0.61$

We can clearly see that our y_5 value is 0.61 which is not an expected actual output, So again we need to find the error and backpropagate through the network by updating the weights until the actual output is obtained.

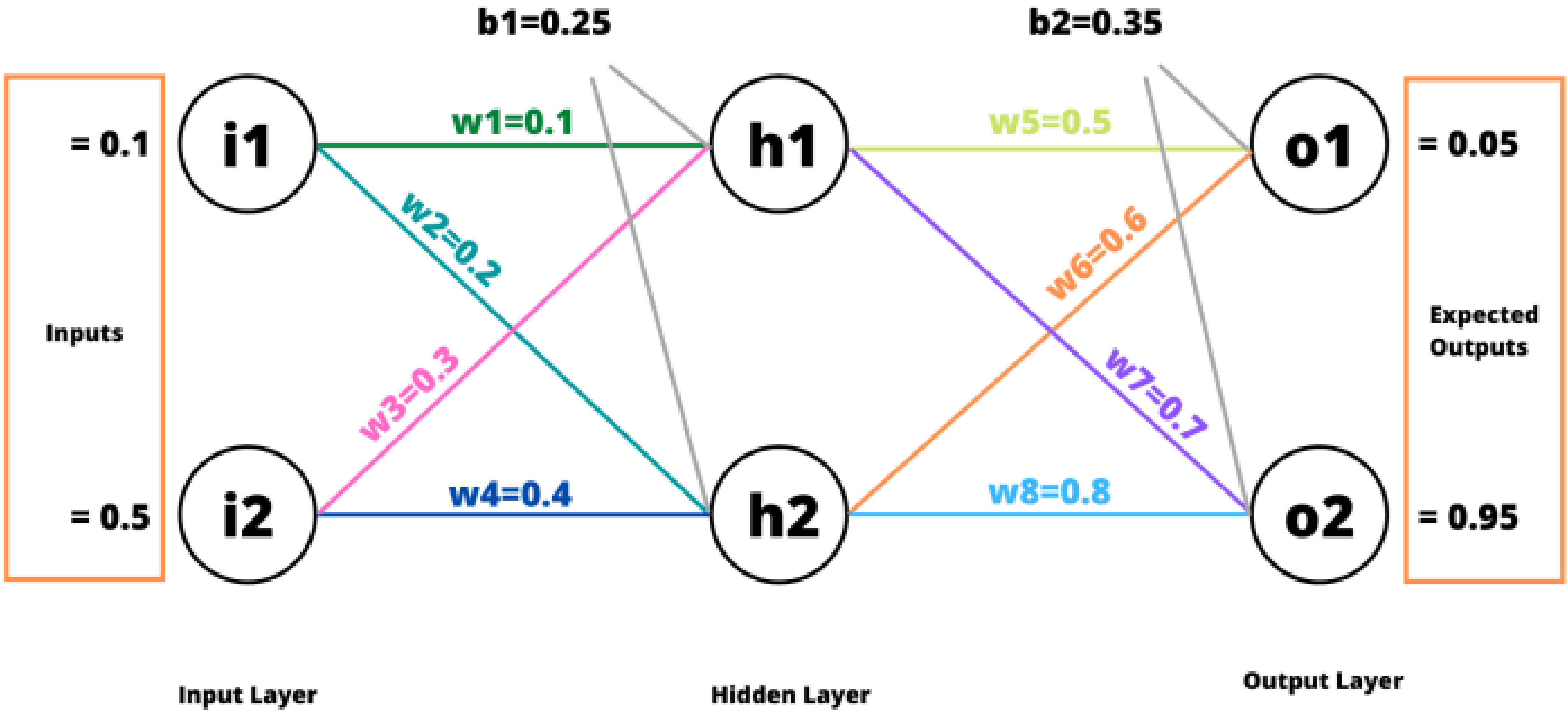
$Error = ytarget - y5$Error=ytarget−y5

$= 0.5 - 0.61$

$= -0.11$

This is how the backpropagate works, it will be performing the forward pass first to see if we obtain the actual output, if not we will be finding the error rate and then backpropagating backwards through the layers in the network by adjusting the weights according to the error rate. This process is said to be continued until the actual output is gained by the neural network.

Example-03: simple neural network



The Forward Pass

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'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,

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

For weights in the output layer (w5, w6, w7, w8)

Let's compute how much contribution w5 has on E_1. If we become clear on how w5 is updated, then it would be really easy for us to generalize the same to the rest of the weights. If we look closely at the example neural network, we can see that E_1 is affected by $output_{o1}$, $output_{o1}$ is affected by sum_{o1}, and sum_{o1} is affected by $w5$. It's time to recall the Chain Rule.

$$\frac{\partial E_{total}}{\partial w5} = \frac{\partial E_{total}}{\partial output_{o1}} * \frac{\partial output_{o1}}{\partial sum_{o1}} * \frac{\partial sum_{o1}}{\partial w5}$$

Let's deal with each component of the above chain separately.

Component 1: partial derivative of Error w.r.t. Output

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

$$E_{total} = \frac{1}{2}(target_1 - output_{o1})^2 + \frac{1}{2}(target_2 - output_{o2})^2$$

Therefore,

$$\frac{\partial E_{total}}{\partial output_{o1}} = 2 * \frac{1}{2} * (target_1 - output_{o1}) * -1$$

$$= output_{o1} - target_1$$

Component 2: partial derivative of Output w.r.t. Sum

The output section of a unit of a neural network uses non-linear activation functions. The activation function used in this example is Logistic Function. When we compute the derivative of the Logistic Function, we get:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

$$\frac{\mathrm{d}}{\mathrm{dx}}\sigma(x) = \sigma(x)(1-\sigma(x))$$

Therefore, the derivative of the Logistic function is equal to output multiplied by (1 – output).

$$\frac{\partial output_{o1}}{\partial sum_{o1}} = output_{o1}(1 - output_{o1})$$

Component 3: partial derivative of Sum w.r.t. Weight

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

Therefore,

$$\frac{\partial sum_{o1}}{\partial w5} = output_{h1}$$

Putting them together,

$$\frac{\partial E_{total}}{\partial w5} = \frac{\partial E_{total}}{\partial output_{o1}} * \frac{\partial output_{o1}}{\partial sum_{o1}} * \frac{\partial sum_{o1}}{\partial w5}$$

$$\frac{\partial E_{total}}{\partial w5} = [output_{o1} - target_1] * [output_{o1}(1 - output_{o1})] * [output_{h1}]$$

$$\frac{\partial E_{total}}{\partial w5} = 0.68492 * 0.19480 * 0.60108$$

$$\frac{\partial E_{total}}{\partial w5} = 0.08020$$

The new_w_5 is,

$new\_w_5 = w5 - n * \frac{\partial E_{total}}{\partial w5}$, where n is learning rate.

$$new\_w_5 = 0.5 - 0.6 * 0.08020$$

$$new\_w_5 = 0.45187$$

We can proceed similarly for w6, w7 and w8.

For w6,

$$\frac{\partial E_{total}}{\partial w6} = \frac{\partial E_{total}}{\partial output_{o1}} * \frac{\partial output_{o1}}{\partial sum_{o1}} * \frac{\partial sum_{o1}}{\partial w6}$$

The first two components of this chain have already been calculated. The last component $\frac{\partial sum_{o1}}{\partial w6} = output_{h2}$.

$$\frac{\partial E_{total}}{\partial w6} = 0.68492 * 0.19480 * 0.61538 = 0.08211$$

The new_w_6 is,

$$new\_w_6 = w6 - n * \frac{\partial E_{total}}{\partial w6}$$

$$new\_w_6 = 0.6 - 0.6 * 0.08211$$

$$new\_w_6 = 0.55073$$

For w7,

$$\frac{\partial E_{total}}{\partial w7} = \frac{\partial E_{total}}{\partial output_{o2}} * \frac{\partial output_{o2}}{\partial sum_{o2}} * \frac{\partial sum_{o2}}{\partial w7}$$

For the first component of the above chain, Let's recall how the partial derivative of Error is computed w.r.t. Output.

$$\frac{\partial E_{total}}{\partial output_{o2}} = output_{o2} - target_2$$

For the second component,

$$\frac{\partial output_{o2}}{\partial sum_{o2}} = output_{o2}(1 - output_{o2})$$

For the third component,

$$\frac{\partial sum_{o2}}{\partial w7} = output_{h1}$$

Putting them together,

$$\frac{\partial E_{total}}{\partial w7} = [output_{o2} - target_2] * [output_{o2}(1 - output_{o2})] * [output_{h1}]$$

$$\frac{\partial E_{total}}{\partial w7} = -0.17044 * 0.17184 * 0.60108$$

$$\frac{\partial E_{total}}{\partial w7} = -0.01760$$

The new_w_7 is,

$$new\_w_7 = w7 - n * \frac{\partial E_{total}}{\partial w7}$$

$$new\_w_7 = 0.7 - 0.6 * -0.01760$$

$$new\_w_7 = 0.71056$$

Proceeding similarly, we get $new\_w_8 = 0.81081$ (with $\frac{\partial E_{total}}{\partial w8} = -0.01802$).

For weights in the hidden layer (w1, w2, w3, w4)

For w1 (with respect to E1),

For simplicity let us compute $\frac{\partial E_1}{\partial w1}$ and $\frac{\partial E_2}{\partial w1}$ separately, and later we can add them to compute $\frac{\partial E_{total}}{\partial w1}$.

$$\frac{\partial E_1}{\partial w1} = \frac{\partial E_1}{\partial output_{o1}} * \frac{\partial output_{o1}}{\partial sum_{o1}} * \frac{\partial sum_{o1}}{\partial output_{h1}} * \frac{\partial output_{h1}}{\partial sum_{h1}} * \frac{\partial sum_{h1}}{\partial w1}$$

Let's quickly go through the above chain. We know that E_1 is affected by $output_{o1}$, $output_{o1}$ is affected by sum_{o1}, sum_{o1} is affected by $output_{h1}$, $output_{h1}$ is affected by sum_{h1}, and finally sum_{h1} is affected by $w1$. It is quite easy to comprehend, isn't it?

For the first component of the above chain,

$$\frac{\partial E_1}{\partial output_{o1}} = output_{o1} - target_1$$

For the third component,

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

$$\frac{\partial sum_{o1}}{\partial output_{h1}} = w5$$

For the fourth component,

$$\frac{\partial output_{h1}}{\partial sum_{h1}} = output_{h1} * (1 - output_{h1})$$

For the fifth component,

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

$$\frac{\partial sum_{h1}}{\partial w1} = i_1$$

Putting them all together,

$$\frac{\partial E_1}{\partial w1} = \frac{\partial E_1}{\partial output_{o1}} * \frac{\partial output_{o1}}{\partial sum_{o1}} * \frac{\partial sum_{o1}}{\partial output_{h1}} * \frac{\partial output_{h1}}{\partial sum_{h1}} * \frac{\partial sum_{h1}}{\partial w1}$$

$$\frac{\partial E_1}{\partial w1} = 0.68492 * 0.19480 * 0.5 * 0.23978 * 0.1 = 0.00159$$

Similarly, for w1 (with respect to E2),

$$\frac{\partial E_2}{\partial w1} = \frac{\partial E_2}{\partial output_{o2}} * \frac{\partial output_{o2}}{\partial sum_{o2}} * \frac{\partial sum_{o2}}{\partial output_{h1}} * \frac{\partial output_{h1}}{\partial sum_{h1}} * \frac{\partial sum_{h1}}{\partial w1}$$

For the first component of the above chain,

$$\frac{\partial E_2}{\partial output_{o2}} = output_{o2} - target_2$$

The second component is already computed.

For the third component,

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

$$\frac{\partial sum_{o2}}{\partial output_{h1}} = w7$$

The fourth and fifth components have also been already computed while computing $\frac{\partial E_1}{\partial w1}$.

Putting them all together,

$$\frac{\partial E_2}{\partial w1} = \frac{\partial E_2}{\partial output_{o2}} * \frac{\partial output_{o2}}{\partial sum_{o2}} * \frac{\partial sum_{o2}}{\partial output_{h1}} * \frac{\partial output_{h1}}{\partial sum_{h1}} * \frac{\partial sum_{h1}}{\partial w1}$$

$$\frac{\partial E_2}{\partial w1} = -0.17044 * 0.17184 * 0.7 * 0.23978 * 0.1 = -0.00049$$

Now we can compute $\frac{\partial E_{total}}{\partial w1} = \frac{\partial E_1}{\partial w1} + \frac{\partial E_2}{\partial w1}$.

$\frac{\partial E_{total}}{\partial w1} = 0.00159 + (-0.00049) = 0.00110.$

The new_w_1 is,

$$new\_w_1 = w1 - n * \frac{\partial E_{total}}{\partial w1}$$

$$new\_w_1 = 0.1 - 0.6 * 0.00110$$

$$new\_w_1 = 0.09933$$

Proceeding similarly, we can easily update the other weights (w2, w3 and w4).

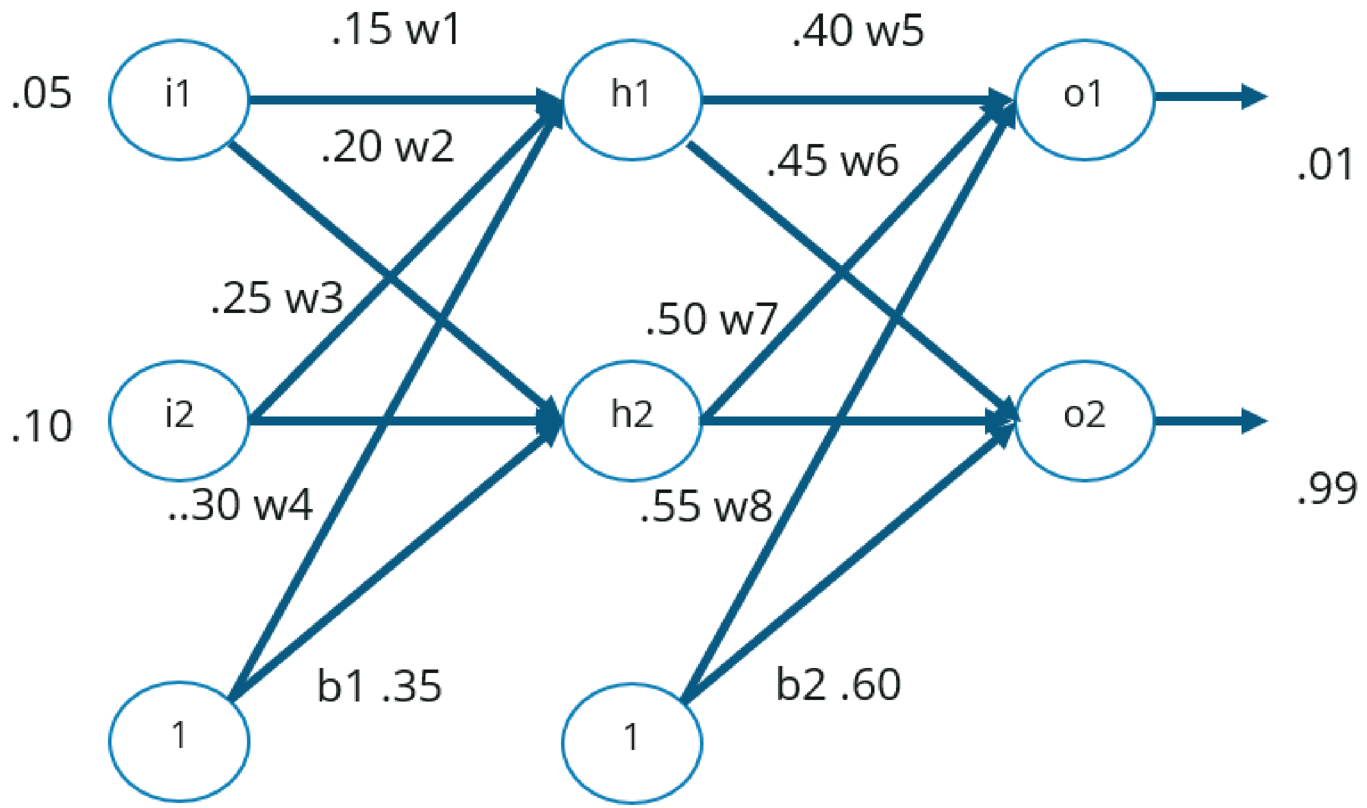
$$new\_w_2 = 0.19919$$

$$new\_w_3 = 0.29667$$

$$new\_w_4 = 0.39597$$

Once we've computed all the new weights, we need to update all the old weights with these new weights. Once the weights are updated, one backpropagation cycle is finished. Now the forward pass is done and the total new error is computed. And based on this newly computed total error the weights are again updated. This goes on until the loss value converges to minima. This way a neural network starts with random values for its weights and finally converges to optimum values.

Example-04: Consider the below Neural Network:



The above network contains the following:

- two inputs
- two hidden neurons
- two output neurons
- two biases

Below are the steps involved in Backpropagation:

- Step – 1: Forward Propagation
- Step – 2: Backward Propagation
- Step – 3: Putting all the values together and calculating the updated weight value

Step – 1: Forward Propagation

We will start by propagating forward.

Net Input For h1:

$net\ h1 = w1*i1 + w2*i2 + b1*1 \rightarrow net\ h1 = 0.15*0.05 + 0.2*0.1 + 0.35*1 = 0.3775$

Output Of h1:

$out\ h1 = 1/1+e^{-net\ h1} \rightarrow 1/1+e^{.3775} = 0.593269992$

Output Of h2:

$out\ h2 = 0.596884378$

Output For o1:

$net\ o1 = w5*out\ h1 + w6*out\ h2 + b2*1 \rightarrow 0.4*0.593269992 + 0.45*0.596884378 + 0.6*1 = 1.105905967$

$Out\ o1 = 1/1+e^{-net\ o1} \rightarrow 1/1+e^{-1.105905967} = 0.75136507$

Output For o2:

$Out\ o2 = 0.772928465$

Error For o1:

$E_{o1} = \Sigma 1/2(target - output)^2$ → $\frac{1}{2}(0.01 - 0.75136507)^2 = 0.274811083$

Error For o2:

$E_{o2} = 0.023560026$

Total Error:

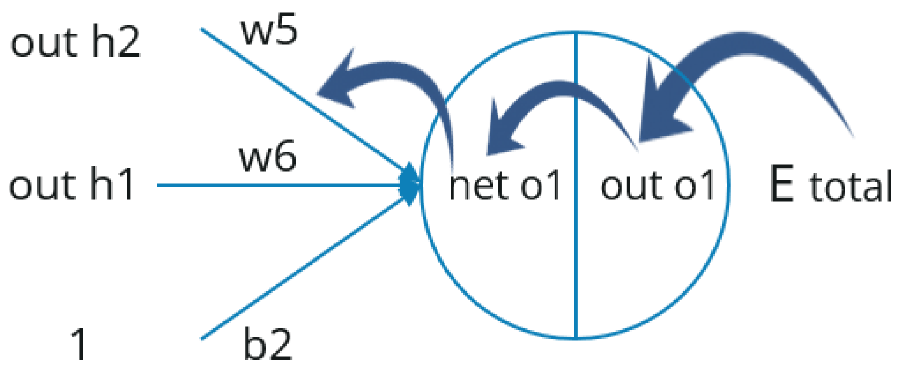
$E_{total} = E_{o1} + E_{o2}$ → $0.274811083 + 0.023560026 = 0.298371109$

Step – 2: Backward Propagation

Now, we will propagate backwards. This way we will try to reduce the error by changing the values of weights and biases.

Consider W5, we will calculate the rate of change of error w.r.t change in weight W5.

$$\frac{\delta Etotal}{\delta w5} = \frac{\delta Etotal}{\delta out\ o1} * \frac{\delta out\ o1}{\delta net\ o1} * \frac{\delta net\ o1}{\delta w5}$$



$$E_{total} = 1/2(\text{target o1} - \text{out o1})^2 + 1/2(\text{target o2} - \text{out o2})^2$$

$$\frac{\delta Etotal}{\delta out\ o1} = -(\text{target o1} - \text{out o1}) = -(0.01 - 0.75136507) = 0.74136507$$

$$\text{out o1} = 1/1+e^{-neto1}$$

$$\frac{\delta out\ o1}{\delta net\ o1} = \text{out o1 (1 - out o1)} = 0.75136507\ (1 - 0.75136507) = 0.186815602$$

$$\text{net o1} = \text{w5 * out h1 + w6 * out h2 + b2 * 1}$$

$$\frac{\delta net\ o1}{\delta w5} = 1 * \text{out h1}\ w5^{(1-1)} + 0 + 0 = 0.593269992$$

Step – 3: Putting all the values together and calculating the updated weight value

Now, let's put all the values together:

$$\frac{\delta Etotal}{\delta w5} = \frac{\delta Etotal}{\delta out\ o1} * \frac{\delta out\ o1}{\delta net\ o1} * \frac{\delta net\ o1}{\delta w5} \rightarrow 0.082167041$$

$$w5^{+} = w5 - n\frac{\delta Etotal}{\delta w5} \rightarrow w5^{+} = 0.4 - 0.5 * 0.082167041$$

Updated w5 → 0.35891648

We can repeat this process to get the new weights w_6, w_7, and w_8:

$$w_6^{+} = 0.408666186$$

$$w_7^{+} = 0.511301270$$

$$w_8^{+} = 0.561370121$$

Hidden Layer

Next, we'll continue the backwards pass by calculating new values for w_1, w_2, w_3, and w_4.

Big picture, here's what we need to figure out:

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

We're going to use a similar process as we did for the output layer, but slightly different to account for the fact that the output of each hidden layer neuron contributes to the output (and therefore error) of multiple output neurons. We know that out_{h1} affects both out_{o1} and out_{o2} therefore the $\frac{\partial E_{total}}{\partial out_{h1}}$ needs to take into consideration its effect on the both output neurons:

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$

Starting with $\frac{\partial E_{o1}}{\partial out_{h1}}$:

$$\frac{\partial E_{o1}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial out_{h1}}$$

We can calculate $\frac{\partial E_{o1}}{\partial net_{o1}}$ using values we calculated earlier:

$$\frac{\partial E_{o1}}{\partial net_{o1}} = \frac{\partial E_{o1}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} = 0.74136507 * 0.186815602 = 0.138498562$$

And $\frac{\partial net_{o1}}{\partial out_{h1}}$ is equal to w_5:

$$net_{o1} = w_5 * out_{h1} + w_6 * out_{h2} + b_2 * 1$$

$$\frac{\partial net_{o1}}{\partial out_{h1}} = w_5 = 0.40$$

Plugging them in:

$$\frac{\partial E_{o1}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial out_{h1}} = 0.138498562 * 0.40 = 0.055399425$$

Following the same process for $\frac{\partial E_{o2}}{\partial out_{h1}}$, we get:

$$\frac{\partial E_{o2}}{\partial out_{h1}} = -0.019049119$$

Therefore:

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}} = 0.055399425 + -0.019049119 = 0.036350306$$

Now that we have $\frac{\partial E_{total}}{\partial out_{h1}}$, we need to figure out $\frac{\partial out_{h1}}{\partial net_{h1}}$ and then $\frac{\partial net_{h1}}{\partial w}$ for each weight:

$$out_{h1} = \frac{1}{1+e^{-net_{h1}}}$$

$$\frac{\partial out_{h1}}{\partial net_{h1}} = out_{h1}(1 - out_{h1}) = 0.59326999(1 - 0.59326999) = 0.241300709$$

We calculate the partial derivative of the total net input to h_1 with respect to w_1 the same as we did for the output neuron:

$$net_{h1} = w_1 * i_1 + w_3 * i_2 + b_1 * 1$$

$$\frac{\partial net_{h1}}{\partial w_1} = i_1 = 0.05$$

Putting it all together:

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial w_1} = 0.036350306 * 0.241300709 * 0.05 = 0.000438568$$

We can now update w_1:

$$w_1^+ = w_1 - \eta * \frac{\partial E_{total}}{\partial w_1} = 0.15 - 0.5 * 0.000438568 = 0.149780716$$

Repeating this for w_2, w_3, and w_4

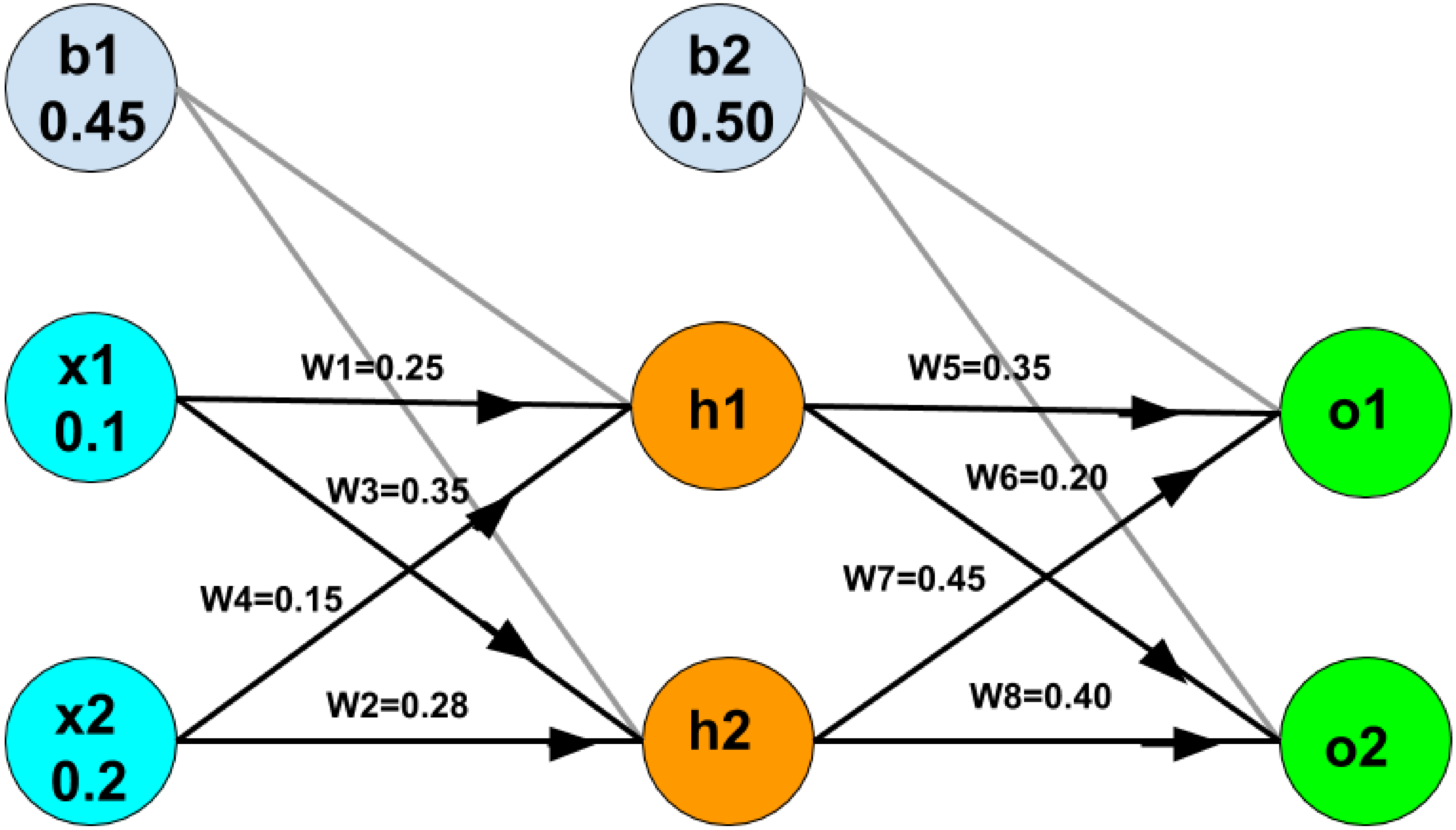
$$w_2^+ = 0.19956143$$

$$w_3^+ = 0.24975114$$

$$w_4^+ = 0.29950229$$

Finally, we've updated all of our weights! When we fed forward the 0.05 and 0.1 inputs originally, the error on the network was 0.298371109. After this first round of backpropagation, the total error is now down to 0.291027924. It might not seem like much, but after repeating this process 10,000 times, for example, the error plummets to 0.0000351085. At this point, when we feed forward 0.05 and 0.1, the two outputs neurons generate 0.015912196 (vs 0.01 target) and 0.984065734 (vs 0.99 target).

Example-5: Consider the below Neural Network:



The correct output from output node *o1* and *o2* be *y1* and *y2* respectively. Let's assume the value of *y1 = 0.05* and the value of *y2 = 0.95* which are the correct outputs labeled for the given inputs.
Here, the values of *h1* and *h2* can be calculated as below:
*h1 = w1 * x1 + w4 * x2 + b1*
*h1 = 0.25 * 0.1 + 0.15 * 0.2 + 0.45 = 0.505*
*h2 = w3 * x1 + w2 * x2 + b1*
*h2 = 0.35 * 0.1 + 0.28 * 0.2 + 0.45 = 0.541*
If we use the **sigmoid** function as the **activation** function, then the value of the output of *h1* is : $H(h1) = \frac{1}{1+e^{-h1}}$

H(h1) is the final output of node *h1* which is equal to *0.623633*
Similarly, the value of *H(h2)* can be calculated as *0.632045*
The output of the node can be calculated as below:
*o1 = w5 * H(h1) + w6 * H(h2) + b2*
*o1 = 0.35 * 0.623633 + 0.20 * 0.632045 + 0.5 = 0.794609*
*o2 = w7 * H(h1) + w8 * H(h2) + b2*
*o2 = 0.45 * 0.623633 + 0.40 * 0.632045 + 0.5 = 1.0334528*
Then, we apply the activation function and store it in the output node. $out_{o1} = \frac{1}{1+e^{-o_1}}, out_{o2} = \frac{1}{1+e^{-o_2}}$

After the final activation function, the values of *out* (o1) is *0.6888201* and the *out* (o2) is *0.7375445*. Till now, we have done Forward Propagation only, now let's jump into the Backpropagation.

Backpropagation Algorithm with Derivation

The total squared error of the neural network can be written like below:

$$E_{total} = E_{out_{o1}} + E_{out_{o2}} = \frac{1}{2}(y_1 - out_{o1})^2 + \frac{1}{2}(y_2 - out_{o2})^2$$

Here, *y1* and *y2* are target outputs that are expected from the node *o1* and *o2* .
If we put all the values, we get the total error as *0.22661423*.

Here, for example, for the weight *w5* ,

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial w_5} * \frac{\partial out_{o1}}{\partial out_{o1}} * \frac{\partial net_{o1}}{\partial net_{o1}} = \frac{\partial E_{total}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial w_5}$$

Here *net(o1)* means the net input that is being received by the *o1* output node. After receiving the net input, *net(o1)* , the output node applies activation function on it and *out(o1)* is stored as output.
Let's see each component of the chain product and find its value. The change in total error with respect to output *o1* is:

$$\frac{\partial E_{total}}{\partial out_{o1}} = \frac{\partial(\frac{1}{2} * (y_1 - out_{o1})^2 + \frac{1}{2} * (y_2 - out_{o2})^2)}{\partial out_{o1}} = -(y_1 - out_{o1})$$

$$\frac{\partial E_{total}}{\partial out_{o1}} = -(0.05 - 0.688201) = 0.6388201$$

Since *out(o1)* is a sigmoid function that depends on *net(o1)* . Now, we will find out the change in output in *o1* which is *out(o1)* with respect to net-input in *o1* node:

$$\frac{\partial out_{o1}}{\partial net_{o1}} = \frac{\partial(\frac{1}{1+e^{net_{o1}}})}{\partial net_{o1}} = -(out_{o1}(1-out_{o1}))$$

$$\frac{\partial out_{o1}}{\partial net_{o1}} = -0.6888201 * (1-0.6888201) = -0.214342$$

The only partial differential left is the change in net-input in *o1* with respect to *w5* :

$$\frac{\partial net_{o1}}{\partial w_5} = \frac{\partial(w5 * H(h1) + w6 * H(h2) + b2 * 1)}{\partial net_{o1}} = H(h1) = out_{h1}$$

$$\frac{\partial net_{o1}}{\partial w_5} = 0.6888201$$

Putting all the value of the above three equation in chain rule, we get:

$$\frac{\partial E_{total}}{\partial w_5} = -(y1 - out_{o1}) * -out_{01}(1-out_{o1}) * out_{h1}$$

$$\frac{\partial E_{total}}{\partial w_5} = 0.6388201 * -0.214342 * 0.6888201 = -0.0943173658$$

To change the value of weight *w5* , we add it by the partial differential of total error by *w5* multiplied by learning rate of *0.5* :

$$w_5^+ = w5 + \eta * \frac{\partial E_{total}}{\partial w_5}$$

$$w_5^+ = 0.35 + 0.5 * -0.09431736 = 0.3026242$$

So, we need to calculate the partial derivative of the total error with respect to *w1* , as we want to see the change in total error with respect to change in *w1* . Applying the chain rule, we get:

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial w_1} * \frac{\partial out_{h1}}{\partial out_{h1}} * \frac{\partial net_{h1}}{\partial net_{h1}} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

Let's calculate the change in total error with respect to *out(h1)*. For easier calculation, lets break *E(total)* into *E(o1)* and *E(o2)* . We know the *E(o1)* is an error in *o1* node and *E(o2)* is an error in the *o2* node.

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$

At first, we will calculate error in *o1* with respect to *out(h1)* : $\frac{\partial E_{o1}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial out_{h1}}$

As we know the values of some part of the differential equation while solving *w5* . So, we simply put the value and calculate the equation.

$$\frac{\partial E_{o1}}{\partial net_{o1}} = \frac{\partial E_{o1}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} = 0.6388201 * -0.214342 = -0.136925$$

We know the equation of *net(o1)* which is net-input in *h1* while solving the Forward Propagation, so we can differentiate it with respect to *out(h1)* .

$$\frac{\partial net_{o1}}{\partial out_{h1}} = \frac{\partial w_5 * out_{h1} + w_6 * out_{h2} + b_2}{\partial out_{h1}} = w_5 = 0.35$$

Putting the value of the above two equations to get the change in the error of *o1* with respect to *out(h1)* , we get:

$$\frac{\partial E_{o1}}{\partial out_{h1}} = -0.136925 * 0.35 = -0.047923$$

Similarly, we calculate the change in the error of *o2* with respect to *out(h1)*. $\frac{\partial E_{o2}}{\partial out_{h1}} = 0.0411254$

Using the value of the above two equations to get the value of the change in the total error with respect to *out(h1)*.

$$\frac{\partial E_{total}}{\partial out_{h1}} = -0.047923 + 0.0411254 = -0.0067976$$

Our first differential equation of the RHS side of the chain rule is calculated. Now we will calculate differential of *out(h1)* w.r.t net-input of *h1* . We know *out(h1)* is a sigmoid function, so its equation is:

$$out_{h1} = \frac{1}{1 + e^{-net_{h1}}}$$

Differentiating it w.r.t *net(h1)* , as, we already know *out(h1),* we can calculate the value like below:

$$\frac{\partial out_{h1}}{\partial net_{h1}} = out_{h1}(1 - out_{h1}) = 0.623633 * 0.376361 = 0.234714$$

We know the formula of *net(h1)* from the Forward Propagation which is: $net_{h1} = w1 * x1 + w4 * x4 + b1$

Differentiating it w.r.t *w1* , we get: $\frac{\partial net_{h1}}{\partial w_1} = x_1 = 0.1$

Putting all these equations to get change in total error w.r.t change in weight of *w1* , we get:

$$\frac{\partial E_{total}}{\partial w_1} = -0.0067976 * 0.234714 * 0.1 = -0.0001595$$

The negative change in total error w.r.t w1 indicates that we need to reduce the value of *w1* to minimize the error. Now, we subtract *w1* with the change in total error with respect to *w1* at a specific learning rate of *0.5* .

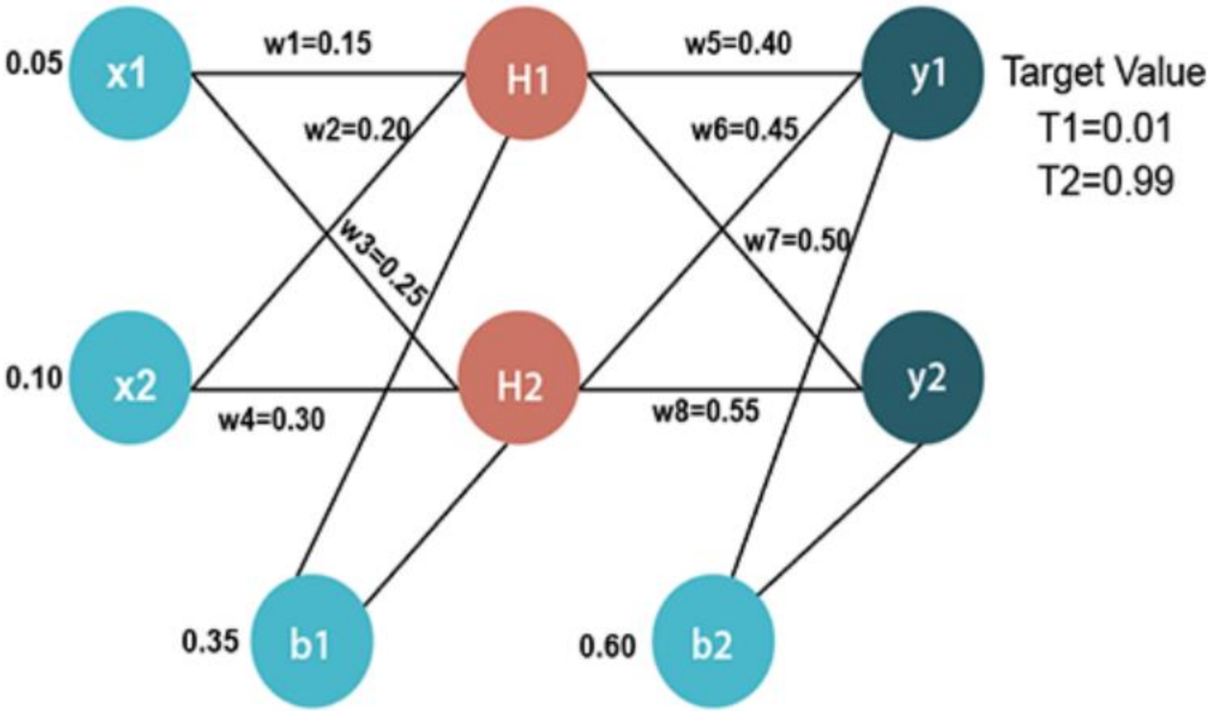
$$w_{+}^{1} = w_1 + \eta * \frac{\partial E_{total}}{\partial w_1}$$

The final value of *w1* will be *0.249921*
Similarly, we will calculate *w1* , *w2* , *w3* , and *w4* . And, we again calculate the total error. We can calculate the total error and check that previously it was *0.22661423* and now it is decreased to *0.224991003* . Though this change might look not so effective, after thousands of iterations, the error will be less than *0.1* .

Problem-06: Apply backpropagation in this neural network where

→ AF [f(y)] = sigmoid &

→ Learning rate (η) = 0.5



Solution:

Forward Pass

To find the value of H1 we first multiply the input value from the weights as

$$H1 = x1 \times w_1 + x2 \times w_2 + b1$$

$$H1 = 0.05 \times 0.15 + 0.10 \times 0.20 + 0.35$$

$$\mathbf{H1 = 0.3775}$$

To calculate the final result of H1, we performed the sigmoid function as

$$H1_{final} = \frac{1}{1 + \frac{1}{e^{H1}}}$$

$$H1_{final} = \frac{1}{1 + \frac{1}{e^{0.3775}}}$$

$$\mathbf{H1_{final} = 0.593269992}$$

We will calculate the value of H2 in the same way as H1

$$H2=x1\times w_3+x2\times w_4+b1$$

$$H2=0.05\times0.25+0.10\times0.30+0.35$$

$$\mathbf{H2=0.3925}$$

To calculate the final result of H1, we performed the sigmoid function as

$$H2_{final} = \frac{1}{1+\frac{1}{e^{H2}}}$$

$$H2_{final} = \frac{1}{1+\frac{1}{e^{0.3925}}}$$

$$\mathbf{H2_{final} = 0.596884378}$$

Now, we calculate the values of y1 and y2 in the same way as we calculate the H1 and H2.

To find the value of y1, we first multiply the input value i.e., the outcome of H1 and H2 from the weights as

$$y1=H1\times w_5+H2\times w_6+b2$$

$$y1=0.593269992\times0.40+0.596884378\times0.45+0.60$$

$$\mathbf{y1=1.10590597}$$

To calculate the final result of y1 we performed the sigmoid function as

$$y1_{final} = \frac{1}{1+\frac{1}{e^{y1}}}$$

$$y1_{final} = \frac{1}{1+\frac{1}{e^{1.10590597}}}$$

$$\mathbf{y1_{final} = 0.75136507}$$

We will calculate the value of y2 in the same way as y1

$$y2=H1\times w_7+H2\times w_8+b2$$

$$y2=0.593269992\times0.50+0.596884378\times0.55+0.60$$

$$\mathbf{y2=1.2249214}$$

To calculate the final result of H1, we performed the sigmoid function as

$$y2_{final} = \frac{1}{1 + \frac{1}{e^{y2}}}$$

$$y2_{final} = \frac{1}{1 + \frac{1}{e^{1.2249214}}}$$

$$\mathbf{y2_{final} = 0.772928465}$$

Our target values are 0.01 and 0.99. Our y1 and y2 value is not matched with our target values T1 and T2.

Now, we will find the **total error**, which is simply the difference between the outputs from the target outputs. The total error is calculated as

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

So, the total error is

$$= \frac{1}{2}(t1 - y1_{final})^2 + \frac{1}{2}(T2 - y2_{final})^2$$

$$= \frac{1}{2}(0.01 - 0.75136507)^2 + \frac{1}{2}(0.99 - 0.772928465)^2$$

$$= 0.274811084 + 0.0235600257$$

$$\mathbf{E_{total} = 0.29837111}$$

Backward pass at the output layer

To update the weight, we calculate the error correspond to each weight with the help of a total error. The error on weight w is calculated by differentiating total error with respect to w.

$$\mathrm{Error}_w = \frac{\partial E_{total}}{\partial w}$$

We perform backward process so first consider the last weight w5 as

$$\mathrm{Error}_{w5} = \frac{\partial E_{total}}{\partial w5} \dots \dots \dots \dots (1)$$

$$E_{total} = \frac{1}{2}(T1 - y1_{final})^2 + \frac{1}{2}(T2 - y2_{final})^2 \dots \dots \dots \dots (2)$$

From equation two, it is clear that we cannot partially differentiate it with respect to w5 because there is no any w5. We split equation one into multiple terms so that we can easily differentiate it with respect to w5 as

$$\frac{\partial E_{total}}{\partial w5} = \frac{\partial E_{total}}{\partial y1_{final}} \times \frac{\partial y1\_final}{\partial y1} \times \frac{\partial y1}{\partial w5} \dots \dots \dots \dots (3)$$

Now, we calculate each term one by one to differentiate E_{total} with respect to w5 as

$$\frac{\partial E_{total}}{\partial y1_{final}} = \frac{\partial(\frac{1}{2}(T1 - y1_{final})^2 + \frac{1}{2}(T2 - y2_{final})^2)}{\partial y1_{final}}$$

$$= 2 \times \frac{1}{2} \times (T1 - y1_{final})^{2-1} \times (-1) + 0$$

$$= -(T1 - y1_{final})$$

$$= -(0.01 - 0.75136507)$$

$$\boldsymbol{\frac{\partial E_{total}}{\partial y1_{final}} = 0.74136507 \ldots\ldots\ldots (4)}$$

$$y1_{final} = \frac{1}{1 + e^{-y1}} \ldots\ldots\ldots\ldots\ldots\ldots (5)$$

$$\frac{\partial y1_{final}}{\partial y1} = \frac{\partial(\frac{1}{1 + e^{-y1}})}{\partial y1}$$

$$= \frac{e^{-y1}}{(1 + e^{-y1})^2}$$

$$= e^{-y1} \times (y1_{final})^2 \ldots\ldots\ldots\ldots\ldots (6)$$

$$y1_{final} = \frac{1}{1 + e^{-y1}}$$

$$e^{-y1} = \frac{1 - y1_{final}}{y1_{final}} \ldots\ldots\ldots\ldots\ldots (7)$$

Putting the value of e^{-y} in equation (5)

$$= \frac{1 - y1_{final}}{y1_{final}} \times (y1_{final})^2$$

$$= y1_{final} \times (1 - y1_{final})$$

$$= 0.75136507 \times (1 - 0.75136507)$$

$$\boldsymbol{\frac{\partial y1_{final}}{\partial y1} = 0.186815602 \ldots\ldots\ldots (8)}$$

$$y1 = H1_{final} \times w5 + H2_{final} \times w6 + b2 \ldots\ldots\ldots\ldots\ldots\ldots (9)$$

$$\frac{\partial y1}{\partial w5} = \frac{\partial(H1_{final} \times w5 + H2_{final} \times w6 + b2)}{\partial w5}$$

$$= H1_{final}$$

$$\boldsymbol{\frac{\partial y1}{\partial w5} = 0.596884378 \ldots\ldots\ldots (10)}$$

So, we put the values of $\frac{\partial E_{total}}{\partial y1_{final}}$, $\frac{\partial y1_{final}}{\partial y1}$, and $\frac{\partial y1}{\partial w5}$ in equation no (3) to find the final result.

$$\frac{\partial E_{total}}{\partial w5} = \frac{\partial E_{total}}{\partial y1_{final}} \times \frac{\partial y1_{final}}{\partial y1} \times \frac{\partial y1}{\partial w5}$$

$$= 0.74136507 \times 0.186815602 \times 0.593269992$$

$$\mathbf{Error_{w5} = \frac{\partial E_{total}}{\partial w5} = 0.0821670407 \ldots\ldots\ldots (11)}$$

Now, we will calculate the updated weight $w5_{new}$ with the help of the following formula

$$w5_{new} = w5 - \eta \times \frac{\partial E_{total}}{\partial w5} \text{ Here, } \eta = \text{learning rate} = 0.5$$

$$= 0.4 - 0.5 \times 0.0821670407$$

$$\mathbf{w5_{new} = 0.35891648 \ldots\ldots\ldots (12)}$$

In the same way, we calculate $w6_{new}$, $w7_{new}$, and $w8_{new}$ and this will give us the following values

$w5_{new}$=0.35891648

$w6_{new}$=408666186

$w7_{new}$=0.511301270

$w8_{new}$=0.561370121

Backward pass at Hidden layer

Now, we will backpropagate to our hidden layer and update the weight w1, w2, w3, and w4 as we have done with w5, w6, w7, and w8 weights.

We will calculate the error at w1 as

$$\mathrm{Error}_{w1} = \frac{\partial E_{total}}{\partial w1}$$

$$E_{total} = \frac{1}{2}(T1 - y1_{final})^2 + \frac{1}{2}(T2 - y2_{final})^2$$

From equation (2), it is clear that we cannot partially differentiate it with respect to w1 because there is no any w1. We split equation (1) into multiple terms so that we can easily differentiate it with respect to w1 as

$$\frac{\partial E_{total}}{\partial w1} = \frac{\partial E_{total}}{\partial H1_{final}} \times \frac{\partial H1\_final}{\partial H1} \times \frac{\partial H1}{\partial w1} \dots\dots\dots\dots (13)$$

Now, we calculate each term one by one to differentiate E_{total} with respect to w1 as

$$\frac{\partial E_{total}}{\partial H1_{final}} = \frac{\partial(\frac{1}{2}(T1 - y1_{final})^2 + \frac{1}{2}(T2 - y2_{final})^2)}{\partial H1} \dots\dots\dots\dots (14)$$

We again split this because there is no any H1^{final} term in E^{toatal} as

$$\frac{\partial E_{total}}{\partial H1_{final}} = \frac{\partial E_1}{\partial H1_{final}} + \frac{\partial E_2}{\partial H1_{final}} \dots\dots\dots (15)$$

$\frac{\partial E_1}{\partial H1_{final}}$ and $\frac{\partial E_2}{\partial H1_{final}}$ will again split because in E1 and E2 there is no H1 term. Splitting is done as

$$\frac{\partial E_1}{\partial H1_{final}} = \frac{\partial E_1}{\partial y1} \times \frac{\partial y1}{\partial H1_{final}} \dots\dots\dots (16)$$

$$\frac{\partial E_2}{\partial H1_{final}} = \frac{\partial E_2}{\partial y2} \times \frac{\partial y2}{\partial H1\_final} \dots\dots\dots (17)$$

We again Split both $\frac{\partial E_1}{\partial y1}$ and $\frac{\partial E_2}{\partial y2}$ because there is no any y1 and y2 term in E1 and E2. We split it as

$$\frac{\partial E_1}{\partial y1} = \frac{\partial E_1}{\partial y1_{fianl}} \times \frac{\partial y1_{final}}{\partial y1} \dots\dots\dots (18)$$

$$\frac{\partial E_2}{\partial y2} = \frac{\partial E_2}{\partial y2_{fianl}} \times \frac{\partial y2_{final}}{\partial y2} \dots\dots\dots (19)$$

Now, we find the value of $\frac{\partial E_1}{\partial y1}$ and $\frac{\partial E_2}{\partial y2}$ by putting values in equation (18) and (19) as

From equation (18)

$$\frac{\partial E_1}{\partial y1} = \frac{\partial E_1}{\partial y1_{fianl}} \times \frac{\partial y1_{final}}{\partial y1}$$

$$= \frac{\partial(\frac{1}{2}(T1 - y1_{final})^2)}{\partial y1_{final}} \times \frac{\partial y1_{final}}{\partial y1}$$

$$= 2 \times \frac{1}{2}(T1 - y1_{final}) \times (-1) \times \frac{\partial y1_{final}}{\partial y1}$$

From equation (8)

$$= 2 \times \frac{1}{2}(0.01 - 0.75136507) \times (-1) \times 0.186815602$$

$$\boldsymbol{\frac{\partial E_1}{\partial y1} = 0.138498562} \dots\dots\dots (20)$$

From equation (19)

$$\frac{\partial E_2}{\partial y2} = \frac{\partial E_2}{\partial y2_{fianl}} \times \frac{\partial y2_{final}}{\partial y2}$$

$$= \frac{\partial(\frac{1}{2}(T2 - y2_{final})^2)}{\partial y2_{final}} \times \frac{\partial y2_{final}}{\partial y2}$$

$$= 2 \times \frac{1}{2}(T2 - y2_{final}) \times (-1) \times \frac{\partial y2_{final}}{\partial y2} \dots\dots\dots (21)$$

$$y2_{final} = \frac{1}{1 + e^{-y2}} \dots\dots\dots\dots\dots\dots (22)$$

$$\frac{\partial y2_{fianl}}{\partial y2} = \frac{\partial(\frac{1}{1 + e^{-y2}})}{\partial y2}$$

$$= \frac{e^{-y2}}{(1 + e^{-y2})^2}$$

$$= e^{-y2} \times (y2_{final})^2 \dots\dots\dots\dots\dots (23)$$

$$y2_{final} = \frac{1}{1 + e^{-y2}}$$

$$e^{-y2} = \frac{1 - y2_{final}}{y2_{final}} \dots\dots\dots\dots\dots (24)$$

Putting the value of e^{-y2} in equation (23)

$$= \frac{1 - y2_{final}}{y2_{final}} \times (y2_{final})^2$$

$$= y2_{final} \times (1 - y2_{final})$$

$$= 0.772928465 \times (1 - 0.772928465)$$

$$\frac{\partial \mathbf{y2_{fianl}}}{\partial \mathbf{y2}} = \mathbf{0.175510053} \ldots\ldots\ldots.\mathbf{(25)}$$

From equation (21)

$$= 2 \times \frac{1}{2}(0.99 - 0.772928465) \times (-1) \times 0.175510053$$

$$\frac{\partial \mathbf{E_1}}{\partial \mathbf{y1}} = \mathbf{-0.0380982366126414} \ldots\ldots\ldots(\mathbf{26})$$

Now from equation (16) and (17)

$$\frac{\partial E_1}{\partial H1_{final}} = \frac{\partial E_1}{\partial y1} \times \frac{\partial y1}{\partial H1_{final}}$$

$$= 0.138498562 \times \frac{\partial(H1_{final} \times w_5 + H2_{final} \times w_6 + b2)}{\partial H1_{final}}$$

$$= 0.138498562 \times \frac{\partial(H1_{final} \times w_5 + H2_{final} \times w_6 + b2)}{\partial H1_{final}}$$

$$= 0.138498562 \times w5$$

$$= 0.138498562 \times 0.40$$

$$\frac{\partial \mathbf{E_1}}{\partial \mathbf{H1_{final}}} = \mathbf{0.0553994248} \ldots\ldots\ldots(\mathbf{27})$$

$$\frac{\partial E_2}{\partial H1_{final}} = \frac{\partial E_2}{\partial y2} \times \frac{\partial y2}{\partial H1\_final}$$

$$= -0.0380982366126414 \times \frac{\partial(H1_{final} \times w_7 + H2_{final} \times w_8 + b2)}{\partial H1_{final}}$$

$$= -0.0380982366126414 \times w7$$

$$= -0.0380982366126414 \times 0.50$$

$$\frac{\partial \mathbf{E_2}}{\partial \mathbf{H1_{final}}} = \mathbf{-0.0190491183063207} \ldots\ldots\ldots(\mathbf{28})$$

Put the value of $\frac{\partial E_1}{\partial H1_{final}}$ and $\frac{\partial E_2}{\partial H1_{final}}$ in equation (15) as

$$\frac{\partial E_{total}}{\partial H1_{final}} = \frac{\partial E_1}{\partial H1_{final}} + \frac{\partial E_2}{\partial H1_{final}}$$

$$= 0.0553994248 + (-0.0190491183063207)$$

$$\boldsymbol{\frac{\partial E_{total}}{\partial H1_{final}} = 0.0364908241736793} \dots\dots\dots \mathbf{(29)}$$

We have $\frac{\partial E_{total}}{\partial H1_{final}}$; we need to figure out $\frac{\partial H1\_final}{\partial H1}$, $\frac{\partial H1}{\partial w1}$ as

$$\frac{\partial H1_{final}}{\partial H1} = \frac{\partial\left(\frac{1}{1+e^{-H1}}\right)}{\partial H1}$$

$$= \frac{e^{-H1}}{(1+e^{-H1})^2}$$

$$e^{-H1} \times (H1_{final})^2 \dots\dots\dots (30)$$

$$H1_{final} = \frac{1}{1+e^{-H1}}$$

$$e^{-H1} = \frac{1-H1_{final}}{H1_{final}} \dots\dots\dots\dots\dots (31)$$

Putting the value of e^{-H1} in equation (30)

$$= \frac{1-H1_{final}}{H1_{final}} \times (H1_{final})^2$$

$$= H1_{final} \times (1-H1_{final})$$

$$= 0.593269992 \times (1-0.593269992)$$

$$\boldsymbol{\frac{\partial H1_{final}}{\partial H1} = 0.2413007085923199}$$

We calculate the partial derivative of the total net input to H1 with respect to w1 the same as we did for the output neuron:

$$\mathrm{H1} = \mathrm{H1}_{\mathrm{final}} \times \mathrm{w5} + \mathrm{H2}_{\mathrm{final}} \times \mathrm{w6} + \mathrm{b2} \dots\dots\dots\dots\dots\dots (32)$$

$$\frac{\partial \mathrm{y1}}{\partial \mathrm{w1}} = \frac{\partial(\mathrm{x1} \times \mathrm{w1} + \mathrm{x2} \times \mathrm{w3} + \mathrm{b1} \times 1)}{\partial \mathrm{w1}}$$

$$= \mathrm{x1}$$

$$\frac{\partial \mathbf{H1}}{\partial \mathbf{w1}} = \mathbf{0.05} \dots\dots\dots (\mathbf{33})$$

So, we put the values of $\frac{\partial \mathrm{E}_{\mathrm{total}}}{\partial \mathrm{H1}_{\mathrm{final}}}$, $\frac{\partial \mathrm{H1}_{\mathrm{final}}}{\partial \mathrm{H1}}$, and $\frac{\partial \mathrm{H1}}{\partial \mathrm{w1}}$ in equation (13) to find the final result.

$$\frac{\partial \mathrm{E}_{\mathrm{total}}}{\partial \mathrm{w1}} = \frac{\partial \mathrm{E}_{\mathrm{total}}}{\partial \mathrm{H1}_{\mathrm{final}}} \times \frac{\partial \mathrm{H1}_{\mathrm{final}}}{\partial \mathrm{H1}} \times \frac{\partial \mathrm{H1}}{\partial \mathrm{w1}}$$

$$= 0.0364908241736793 \times 0.2413007085923199 \times 0.05$$

$$\mathbf{Error_{w1}} = \frac{\partial \mathbf{E_{total}}}{\partial \mathbf{w1}} = \mathbf{0.000438568} \dots\dots\dots (\mathbf{34})$$

Now, we will calculate the updated weight $w1_{new}$ with the help of the following formula

$$w1_{new} = w1 - \eta \times \frac{\partial E_{total}}{\partial w1} \text{ Here } \eta = \text{learning rate} = 0.5$$

$$= 0.15 - 0.5 \times 0.000438568$$

$$\mathbf{w1_{new} = 0.149780716 \ldots\ldots\ldots (35)}$$

In the same way, we calculate $w2_{new}$, $w3_{new}$, and w4 and this will give us the following values

$w1_{new}$=0.149780716

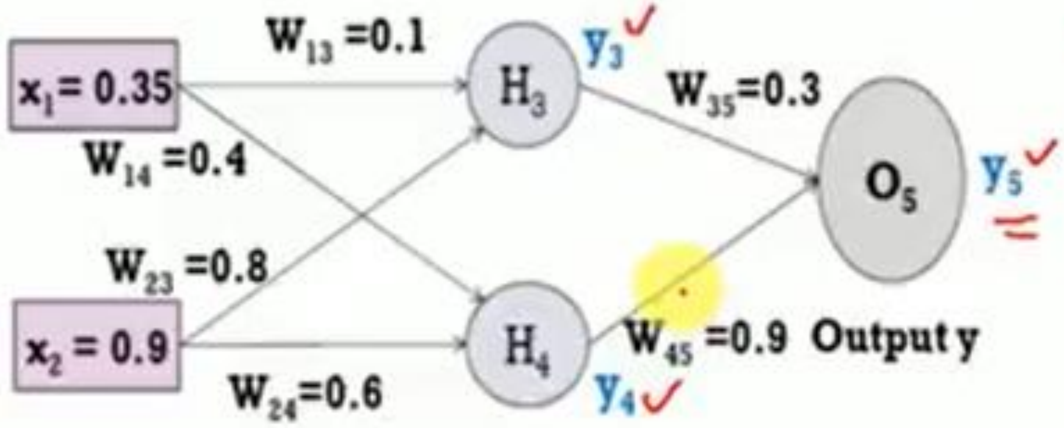
$w2_{new}$=0.19956143

$w3_{new}$=0.24975114

$w4_{new}$=0.29950229

We have updated all the weights. We found the error 0.298371109 on the network when we fed forward the 0.05 and 0.1 inputs. In the first round of Backpropagation, the total error is down to 0.291027924. After repeating this process 10,000, the total error is down to 0.0000351085. At this point, the outputs neurons generate 0.159121960 and 0.984065734 i.e., nearby our target value when we feed forward the 0.05 and 0.1.

Example-07: assume that the neurons have a sigmoid activation function, perform a forward pass and a backward pass on the network. assume that the target output is t= 0.5 and learning rate(η) = 1



- Forward Pass: Compute output for y3, y4 and y5.

$$a_j = \sum_j (w_{i,j} * x_i) \qquad yj = F(aj) = \frac{1}{1+e^{-a_j}}$$

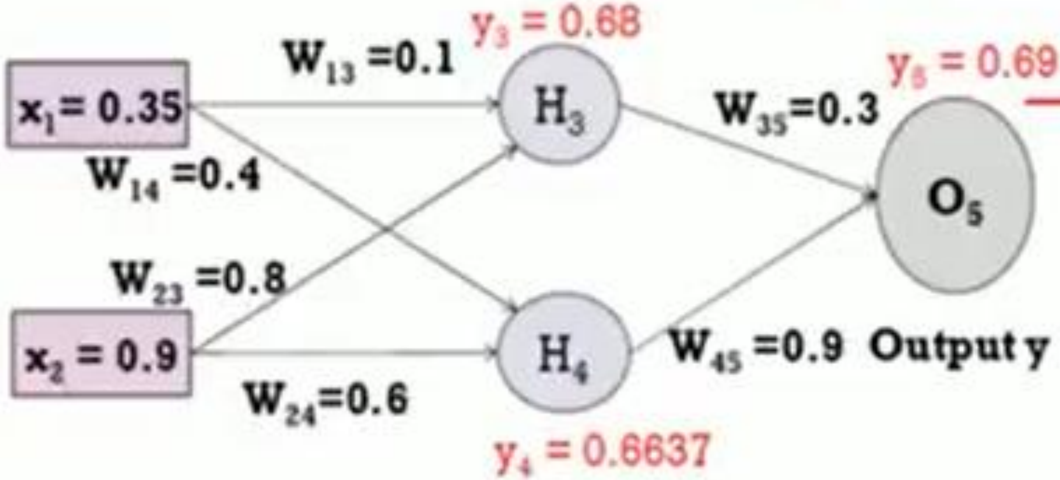
$$a_1 = (w_{13} * x_1) + (w_{23} * x_2)$$
$$= (0.1 * 0.35) + (0.8 * 0.9) = 0.755$$
$$y_3 = f(a_1) = 1/\,(1 + e^{-0.755}) = 0.68$$

$$a_2 = (w_{14} * x_1) + (w_{24} * x_2)$$
$$= (0.4 * 0.35) + (0.6 * 0.9) = 0.68$$
$$y_4 = f(a_2) = 1/\,(1 + e^{-0.68}) = 0.6637$$

$$a_3 = (w_{35} * y_3) + (w_{45} * y_4)$$
$$= (0.3 * 0.68) + (0.9 * 0.6637) = 0.801$$
$$y_5 = f(a_3) = 1/\,(1 + e^{-0.801}) = 0.69 \text{ (Network Output)}$$

$$\text{Error} = y_{target} - y_5 = -0.19$$

$0.5 - 0.69$



- Backward Pass: Compute δ3, δ4 and δ5.

For output unit:

$\delta_5 = y(1-y)\,(y_{target} - y)$

$= 0.69*(1-0.69)*(0.5-0.69) = -0.0406$

For hidden unit:

$\delta_3 = y_3(1-y_3)\, w_{35} * \delta_5$

$= 0.68*(1 - 0.68)*(0.3 * -0.0406) = -0.00265$

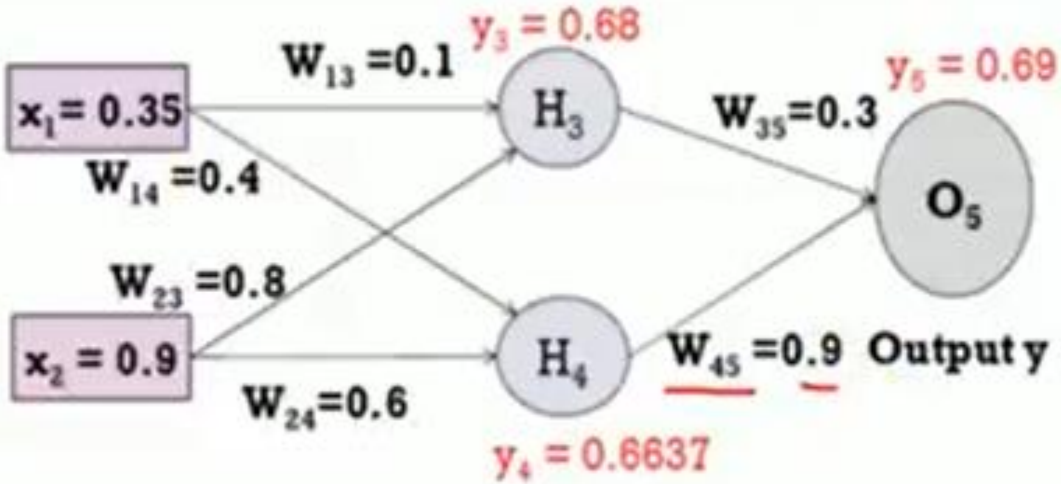
$\delta_4 = y_4(1-y_4) w_{45} * \delta_5$

$= 0.6637*(1 - 0.6637)*(0.9 * -0.0406) = -0.0082$

$$\Delta w_{ji} = \eta \delta_j o_i$$

$$\delta_j = o_j(1-o_j)(t_j - o_j) \qquad \text{if } j \text{ is an output unit}$$

$$\delta_j = o_j(1-o_j)\sum_k \delta_k w_{kj} \qquad \text{if } j \text{ is a hidden unit}$$



- Backward Pass: Compute δ3, δ4 and δ5.

For output unit:

$$\delta_5 = y(1-y)\,(y_{target} - y)$$
$$= 0.69*(1-0.69)*(0.5-0.69) = -0.0406$$

For hidden unit:

$$\delta_3 = y_3(1-y_3)\,w_{35} * \delta_5$$
$$= 0.68*(1 - 0.68)*(0.3 * -0.0406) = -0.00265$$

$$\delta_4 = y_4(1-y_4) w_{45} * \delta_5$$
$$= 0.6637*(1 - 0.6637)* (0.9 * -0.0406) = -0.0082$$

Compute new weights

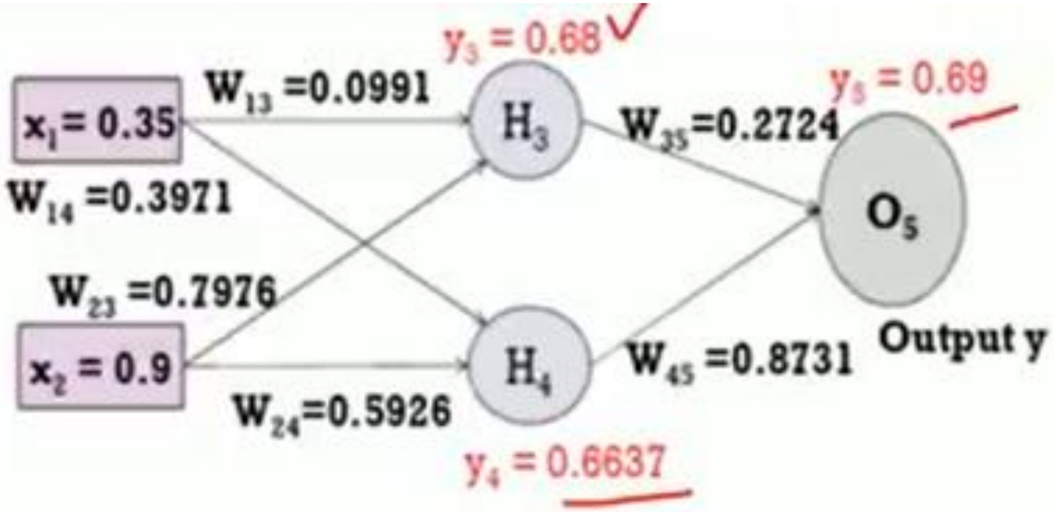
$$\Delta w_{ji} = \eta \delta_j o_i$$

$$\Delta w_{45} = \eta\, \delta_5\, y_4 = 1 * -0.0406 * 0.6637 = -0.0269$$
$$w_{45}\,(\text{new}) = \Delta w_{45} + w_{45}(\text{old}) = -0.0269 + (0.9) = 0.8731$$

$$\Delta w_{14} = \eta\, \delta_4\, x_1 = 1 * -0.0082 * 0.35 = -0.00287$$
$$w_{14}\,(\text{new}) = \Delta w_{14} + w_{14}(\text{old}) = -0.00287 + 0.4 = 0.3971$$

- Similarly, update all other weights

i	j	w_{ij}	δ_j	x_i	η	Updated w_{ij}
1	3	0.1	-0.00265	0.35	1	0.0991
2	3	0.8	-0.00265	0.9	1	0.7976
1	4	0.4	-0.0082	0.35	1	0.3971
2	4	0.6	-0.0082	0.9	1	0.5926
3	5	0.3	-0.0406	0.68	1	0.2724
4	5	0.9	-0.0406	0.6637	1	0.8731



- Forward Pass: Compute output for y3, y4 and y5.

$$a_j = \sum_j (w_{i,j} * x_i) \qquad yj = F(aj) = \frac{1}{1 + e^{-a_j}}$$

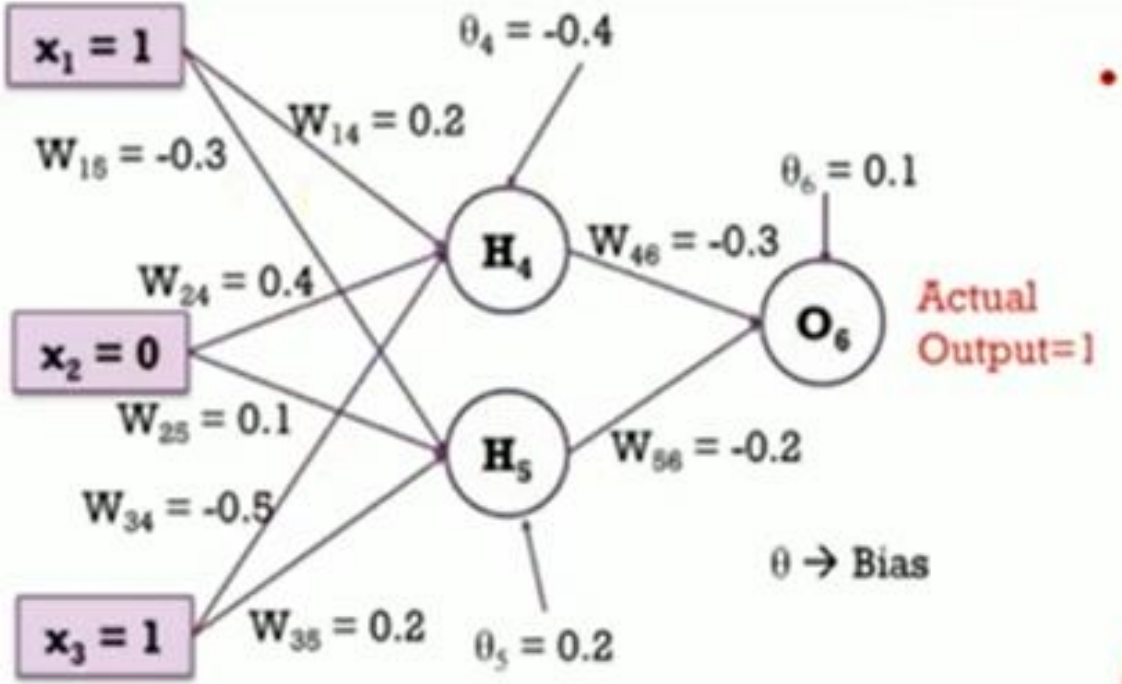
$a_1 = (w_{13} * x_1) + (w_{23} * x_2)$
$= (0.0991 * 0.35) + (0.7976 * 0.9) = 0.7525$
$y_3 = f(a_1) = 1/(1 + e^{-0.7525}) = 0.6797$

$a_2 = (w_{14} * x_1) + (w_{24} * x_2)$
$= (0.3971 * 0.35) + (0.5926 * 0.9) = 0.6723$
$y_4 = f(a_2) = 1/(1 + e^{-0.6723}) = 0.6620$

$a_3 = (w_{35} * y_3) + (w_{45} * y_4)$
$= (0.2724 * 0.6797) + (0.8731 * 0.6620) = 0.7631$
$y_5 = f(a_3) = 1/(1 + e^{-0.7631}) = \mathbf{0.6820}$ (Network Output)

Error $= y_{target} - y_5 = -0.182$

Example-08: assume that the neurons have a sigmoid activation function, perform a forward pass and a backward pass on the network. assume that the target output is $t = 1$ and learning rate(η) = 0.9



$$\text{Error} = y_{target} - y_6 = 0.526$$

- Forward Pass: Compute output for y4, y5 and y6.

$$a_j = \sum_j (w_{i,j} * x_i) \qquad yj = F(aj) = \frac{1}{1 + e^{-a_j}}$$

$$a_4 = (w_{14} * x_1) + (w_{24} * x_2) + (w_{34} * x_3) + \theta_4$$
$$= (0.2 * 1) + (0.4 * 0) + (-0.5 * 1) + (-0.4) = -0.7$$
$$O(H_4) = y_4 = f(a_4) = 1/(1 + e^{0.7}) = 0.332$$

$$a_5 = (w_{15} * x_1) + (w_{25} * x_2) + (w_{35} * x_3) + \theta_5$$
$$= (-0.3 * 1) + (0.1 * 0) + (0.2 * 1) + (0.2) = 0.1$$
$$O(H_5) = y_5 = f(a_5) = 1/(1 + e^{-0.1}) = 0.525$$

$$a_6 = (w_{46} * H_4) + (w_{56} * H_5) + \theta_6$$
$$= (-0.3 * 0.332) + (-0.2 * 0.525) + 0.1 = -0.105$$
$$O(O_6) = y_6 = f(a_6) = 1/(1 + e^{0.105}) = \mathbf{0.474}$$

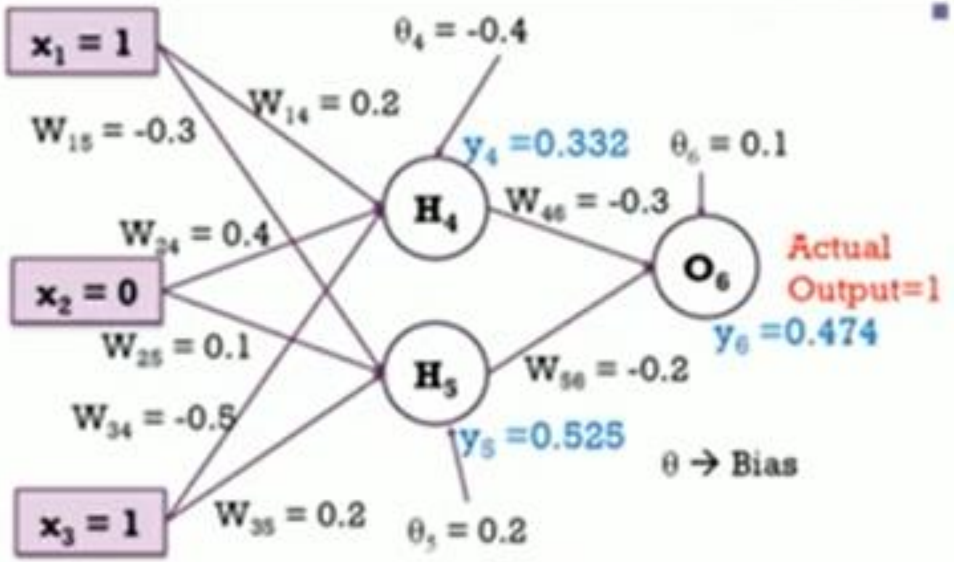
- Each weight changed by:

$$\Delta w_{ji} = \eta \delta_j o_i$$

$$\delta_j = o_j(1-o_j)(t_j - o_j) \qquad \text{if } j \text{ is an output unit}$$

$$\delta_j = o_j(1-o_j)\sum_k \delta_k w_{kj} \qquad \text{if } j \text{ is a hidden unit}$$

- where η is a constant called the learning rate
- tj is the correct teacher output for unit j
- δj is the error measure for unit j



- **Backward Pass: Compute δ4, δ5 and δ6.**

For output unit:

$$\delta_6 = y_6(1-y_6)\,(y_{target} - y_6)$$
$$= 0.474*(1-0.474)*(1-0.474) = 0.1311$$

For hidden unit:

$$\delta_5 = y_5(1-y_5)\, w_{56} * \delta_6$$
$$= 0.525*(1 - 0.525)*(-0.2 * 0.1311) = -0.0065$$

$$= y_4(1-y_4)\, w_{46} * \delta_6$$
$$= 0.332*(1 - 0.332)* (-0.3 * 0.1331) = -0.0087$$

Compute new weights

$$\Delta w_{ji} = \eta \delta_j o_i$$

$\Delta w_{46} = \eta\, \delta_6\, y_4 = 0.9 * 0.1311 * 0.332 = 0.03917$

w_{46} (new) $= \Delta w_{46} + w_{46}$(old) $= 0.03917 + (-0.3) = -0.261$

$\Delta w_{14} = \eta\, \delta_4\, x_1 = 0.9 * -0.0087 * 1 = -0.0078$

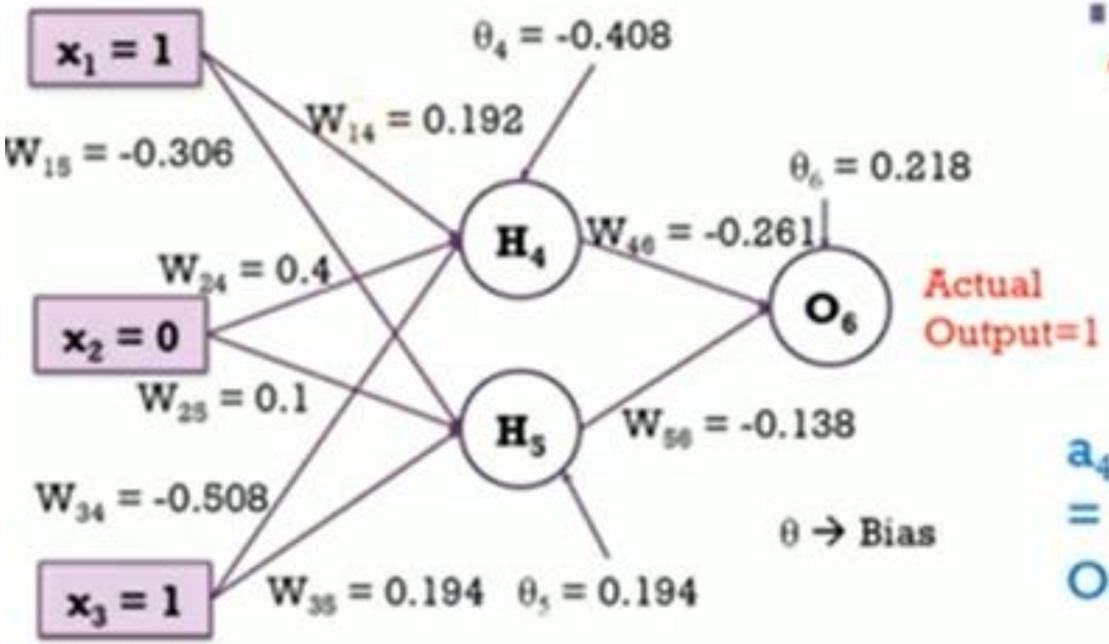
w_{14} (new) $= \Delta w_{14} + w_{14}$(old) $= -0.0078 + 0.2 = 0.192$

Similarly, update all other weights

i	j	w_{ij}	δ_j	x_i	η	Updated w_{ij}
4	6	-0.3	0.1311	0.332	0.9	-0.261
5	6	-0.2	0.1311	0.525	0.9	-0.138
1	4	0.2	-0.0087	1	0.9	0.192
1	5	-0.3	-0.0065	1	0.9	-0.306
2	4	0.4	-0.0087	0	0.9	0.4
2	5	0.1	-0.0065	0	0.9	0.1
3	4	-0.5	-0.0087	1	0.9	-0.508
3	5	0.2	-0.0065	1	0.9	0.194

- Similarly, update bais weights

θ_j	Previous θ_j	δ_j	η	Updated θ_j
Θ_6	0.1	0.1311	0.9	0.218
Θ_5	0.2	-0.0065	0.9	0.194
Θ_4	-0.4	-0.0087	0.9	-0.408



Error $= y_{target} - y_6 = 0.485$

- Forward Pass: Compute output for y4, y5 and y6.

$$a_j = \sum_j (w_{i,j} * x_i) \qquad yj = F(aj) = \frac{1}{1 + e^{-aj}}$$

$a_4 = (w_{14} * x_1) + (w_{24} * x_2) + (w_{34} * x_3) + \theta_4$
$= (0.192 * 1) + (0.4 * 0) + (-0.508 * 1) + (-0.408) = -0.724$
$O(H_4) = y_4 = f(a_4) = 1/\ (1 + e^{0.724}) = 0.327$

$a_5 = (w_{15} * x_1) + (w_{25} * x_2) + (w_{35} * x_3) + \theta_5$
$= (-0.306 * 1) + (0.1 * 0) + (0.194 * 1) + (0.194) = 0.082$
$O(H_5) = y_5 = f(a_5) = 1/\ (1 + e^{-0.082}) = 0.520$

$a_6 = (w_{46} * H_4) + (w_{56} * H_5) + \theta_6$
$= (-0.261 * 0.327) + (-0.138 * 0.520) + 0.218 = 0.061$
$O(O_6) = y_6 = f(a_6) = 1/\ (1 + e^{-0.061}) =$ **0.515** (Network Output)