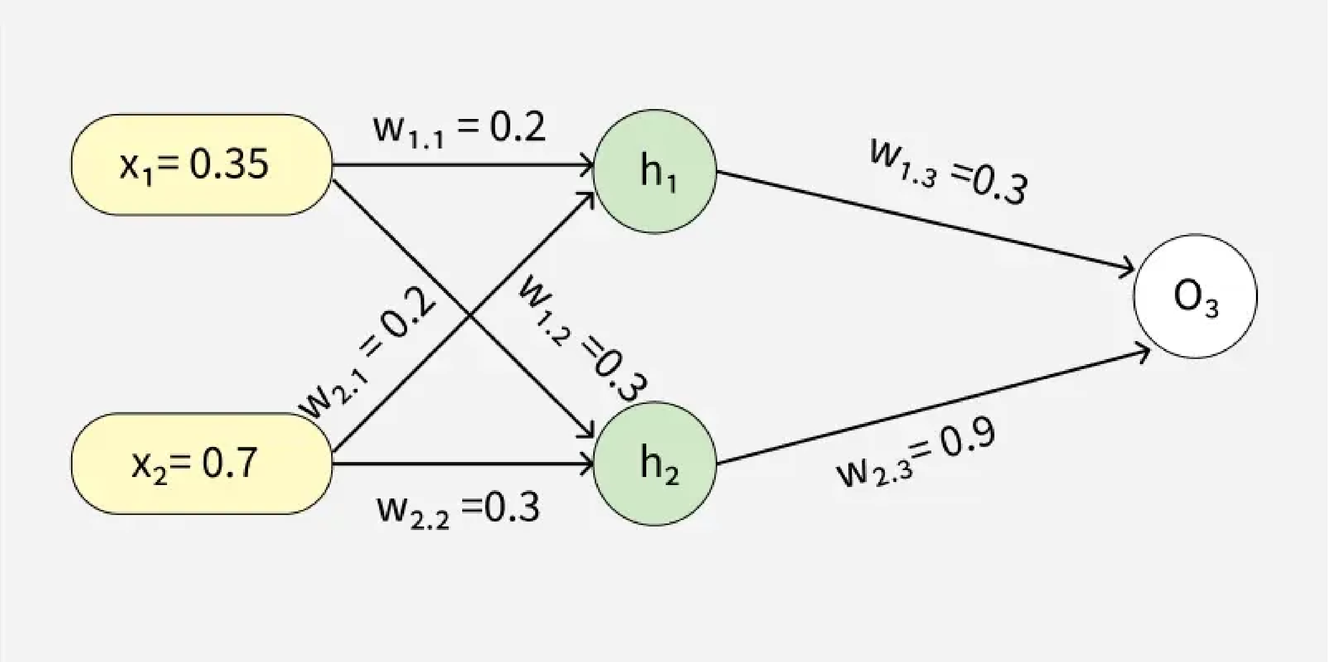
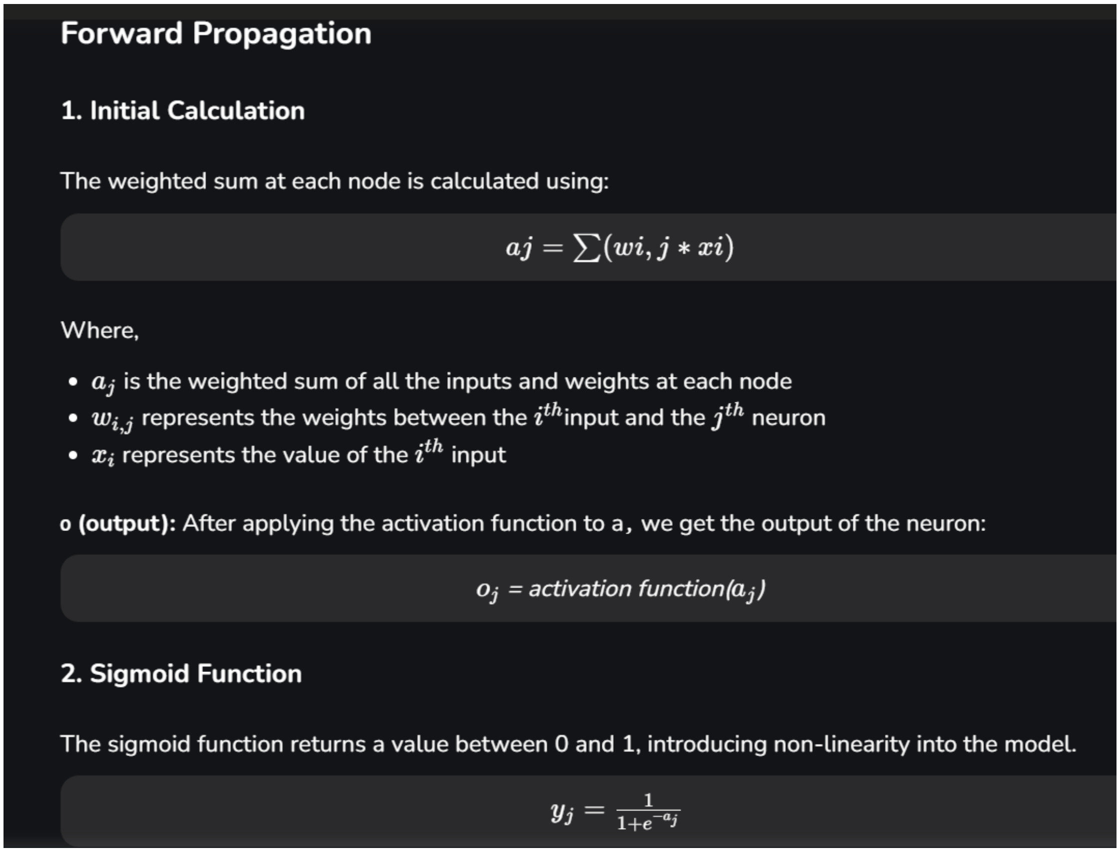
Example of Back Propagation in Machine Learning

Let's walk through an example of Back Propagation in machine learning. Assume the neurons use the sigmoid activation function for the forward and backward pass. The target output is 0.5 and the learning rate is 1.





Forward Propagation

1. Initial Calculation

The weighted sum at each node is calculated using:

$$aj = \sum(wi, j * xi)$$

Where,

- a_j is the weighted sum of all the inputs and weights at each node
- $w_{i,j}$ represents the weights between the i^{th}input and the j^{th} neuron
- x_i represents the value of the i^{th} input

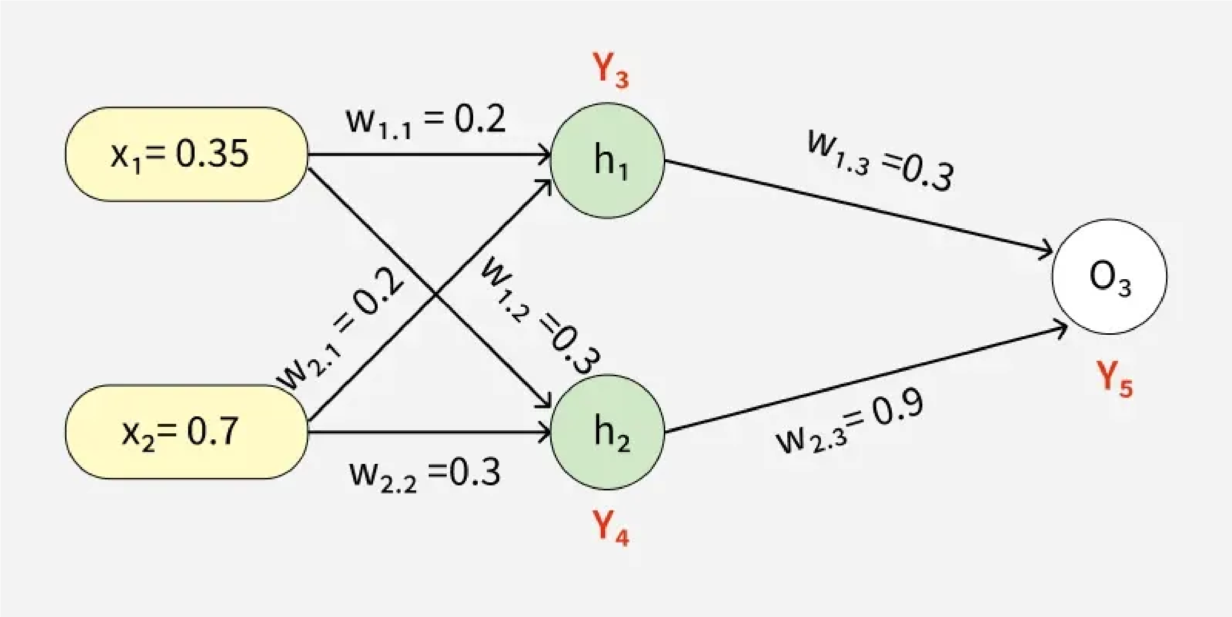
o (output): After applying the activation function to a, we get the output of the neuron:

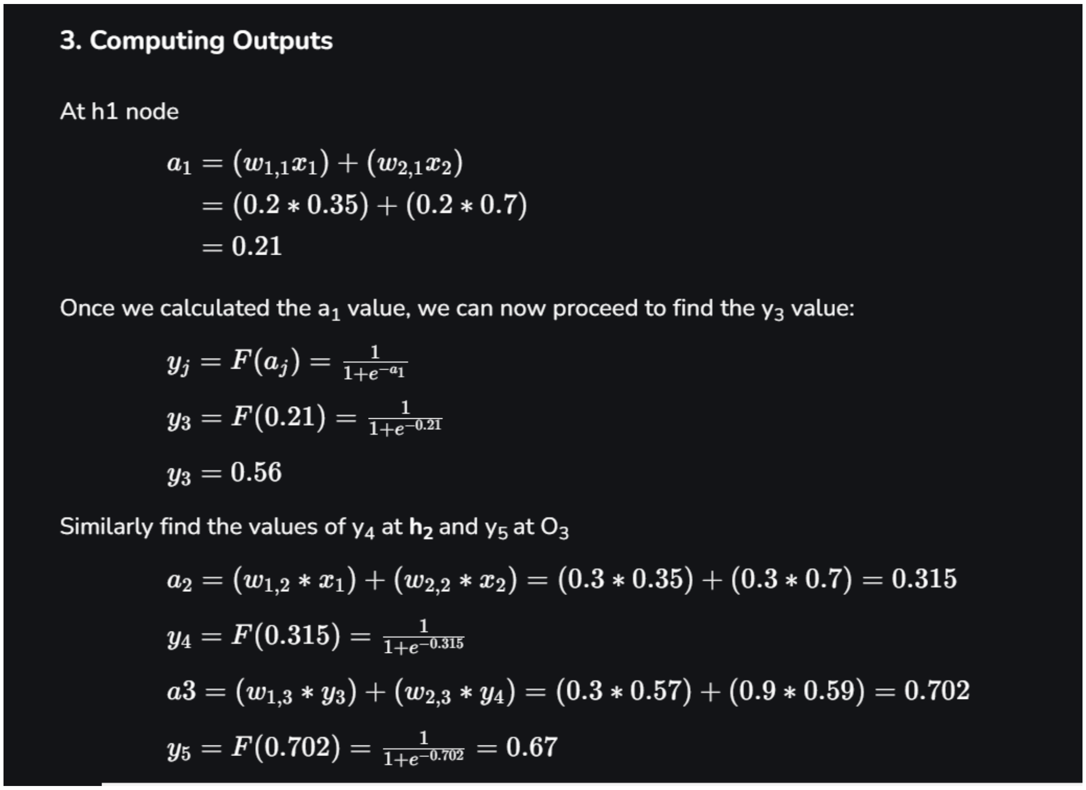
$$o_j = activation\ function(a_j)$$

2. Sigmoid Function

The sigmoid function returns a value between 0 and 1, introducing non-linearity into the model.

$$y_j = \frac{1}{1+e^{-a_j}}$$





3. Computing Outputs

At h1 node

$$\begin{aligned} a_1 &= (w_{1,1}x_1) + (w_{2,1}x_2) \\ &= (0.2 * 0.35) + (0.2 * 0.7) \\ &= 0.21 \end{aligned}$$

Once we calculated the a$_1$ value, we can now proceed to find the y$_3$ value:

$$y_j = F(a_j) = \frac{1}{1+e^{-a_1}}$$

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

$$y_3 = 0.56$$

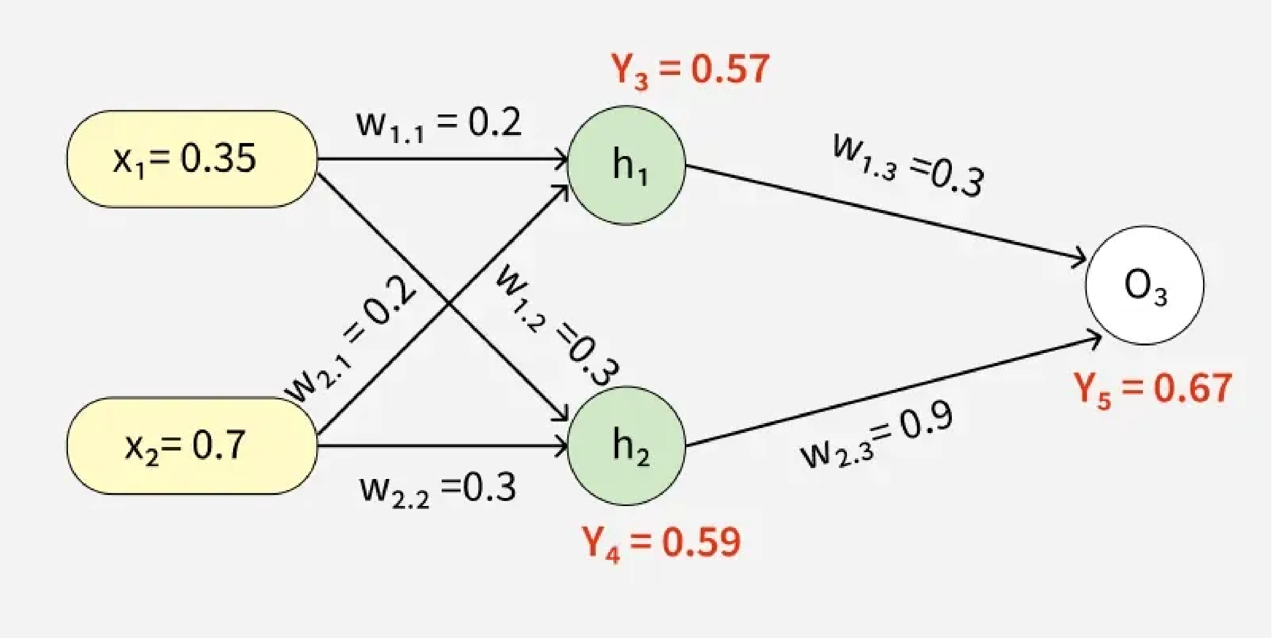
Similarly find the values of y$_4$ at **h$_2$** and y$_5$ at O$_3$

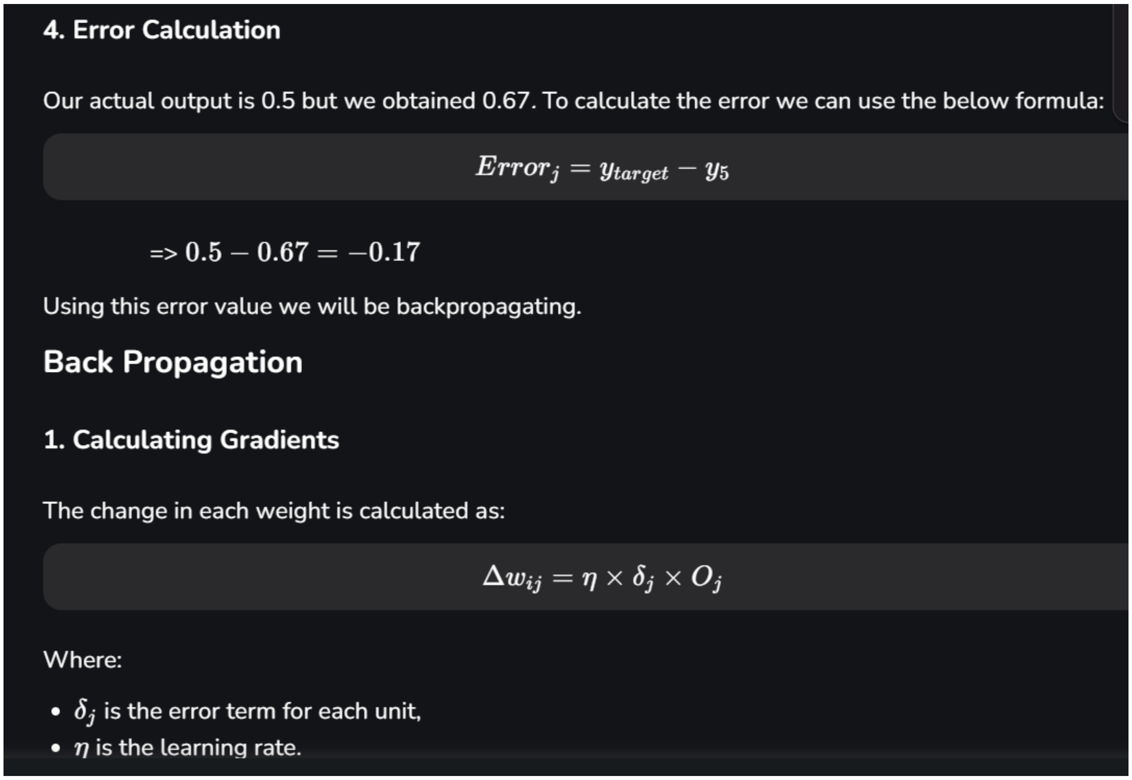
$$a_2 = (w_{1,2} * x_1) + (w_{2,2} * x_2) = (0.3 * 0.35) + (0.3 * 0.7) = 0.315$$

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

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

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





4. Error Calculation

Our actual output is 0.5 but we obtained 0.67. To calculate the error we can use the below formula:

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

=> $0.5 - 0.67 = -0.17$

Using this error value we will be backpropagating.

Back Propagation

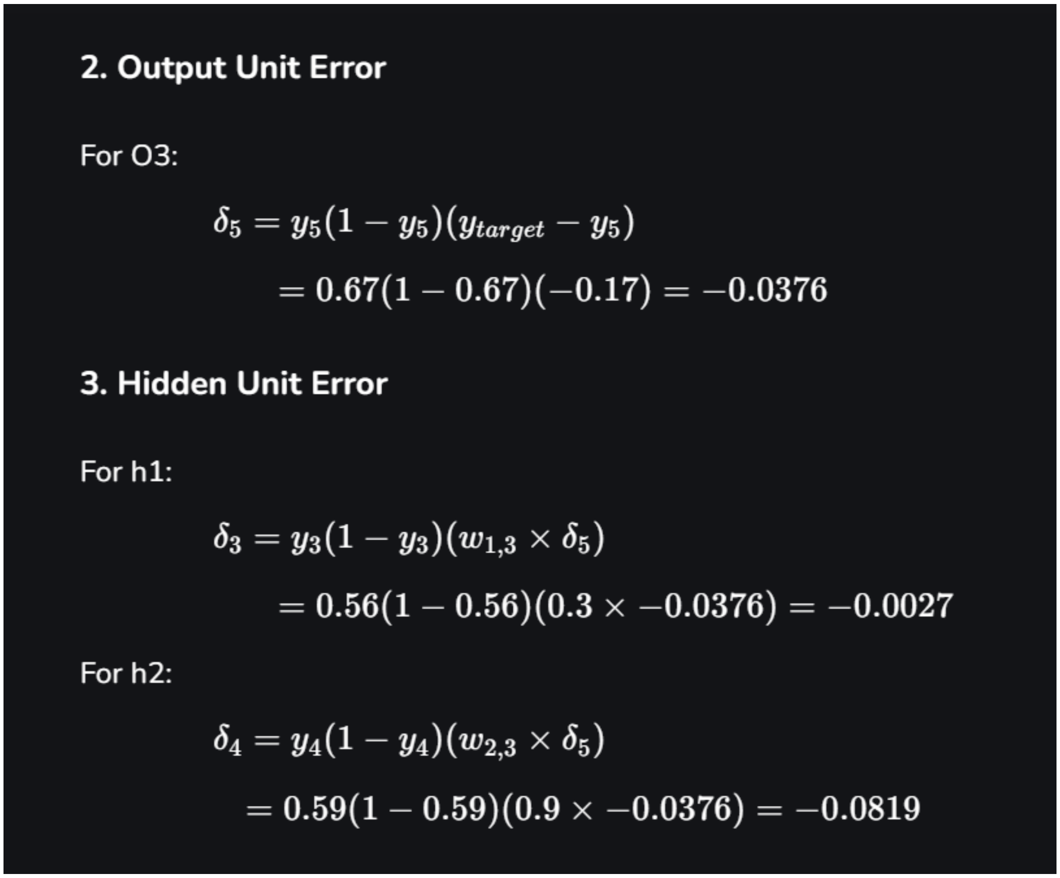
1. Calculating Gradients

The change in each weight is calculated as:

$$\Delta w_{ij} = \eta \times \delta_j \times O_j$$

Where:

- δ_j is the error term for each unit,
- η is the learning rate.



2. Output Unit Error

For O3:

$$\begin{aligned}\delta_5 &= y_5(1 - y_5)(y_{target} - y_5) \\ &= 0.67(1 - 0.67)(-0.17) = -0.0376\end{aligned}$$

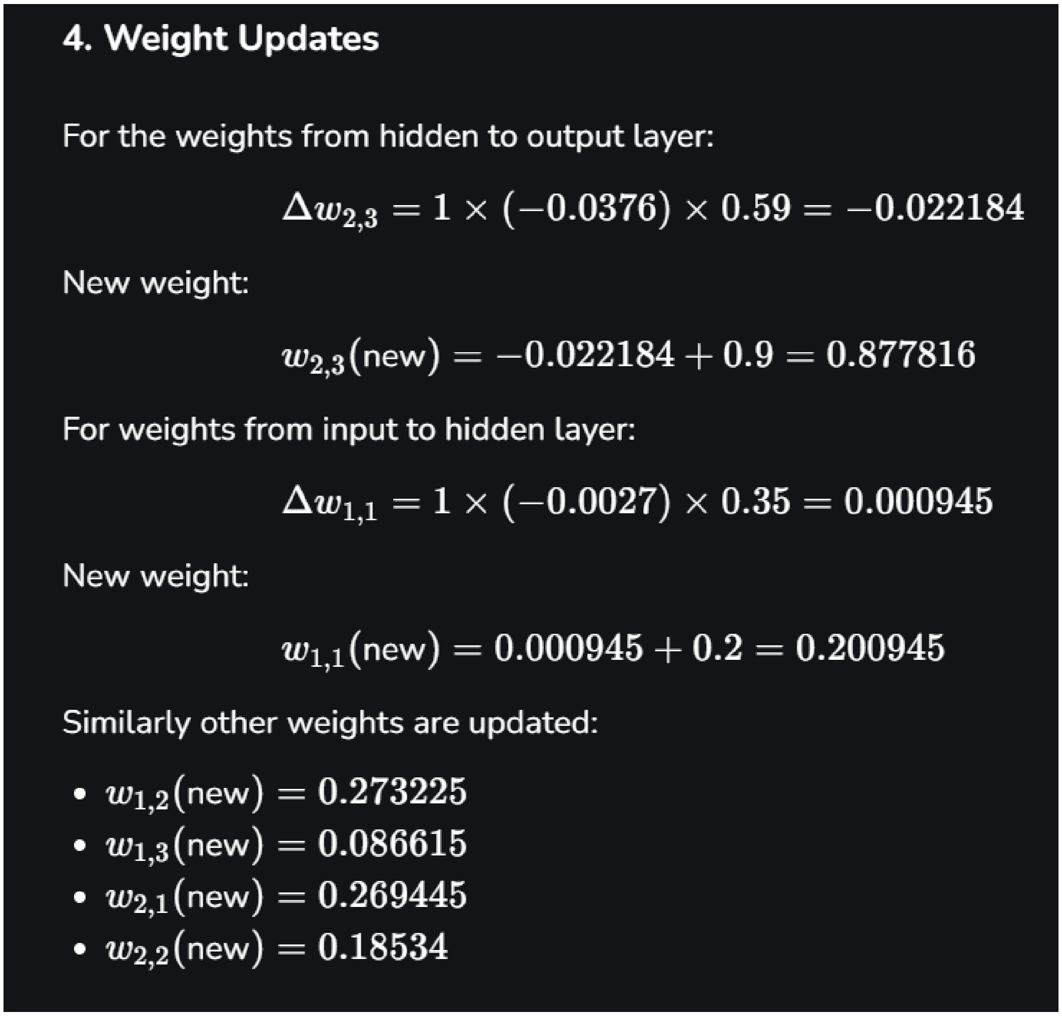
3. Hidden Unit Error

For h1:

$$\begin{aligned}\delta_3 &= y_3(1 - y_3)(w_{1,3} \times \delta_5) \\ &= 0.56(1 - 0.56)(0.3 \times -0.0376) = -0.0027\end{aligned}$$

For h2:

$$\begin{aligned}\delta_4 &= y_4(1 - y_4)(w_{2,3} \times \delta_5) \\ &= 0.59(1 - 0.59)(0.9 \times -0.0376) = -0.0819\end{aligned}$$



4. Weight Updates

For the weights from hidden to output layer:

$$\Delta w_{2,3} = 1 \times (-0.0376) \times 0.59 = -0.022184$$

New weight:

$$w_{2,3}(\text{new}) = -0.022184 + 0.9 = 0.877816$$

For weights from input to hidden layer:

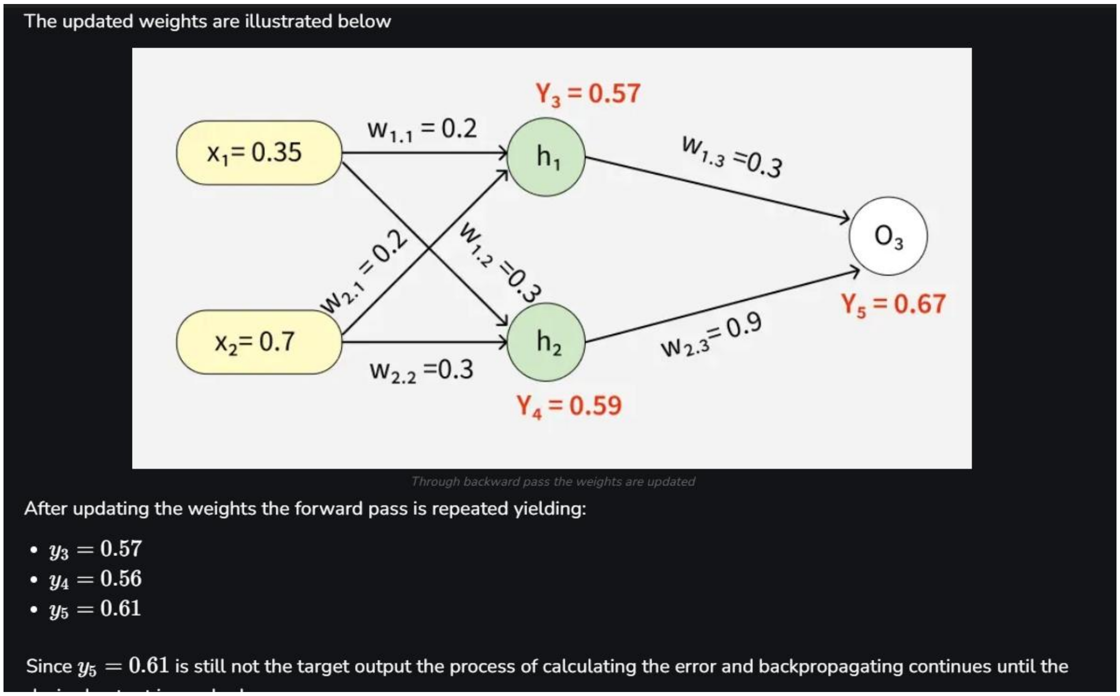
$$\Delta w_{1,1} = 1 \times (-0.0027) \times 0.35 = 0.000945$$

New weight:

$$w_{1,1}(\text{new}) = 0.000945 + 0.2 = 0.200945$$

Similarly other weights are updated:

- $w_{1,2}(\text{new}) = 0.273225$
- $w_{1,3}(\text{new}) = 0.086615$
- $w_{2,1}(\text{new}) = 0.269445$
- $w_{2,2}(\text{new}) = 0.18534$



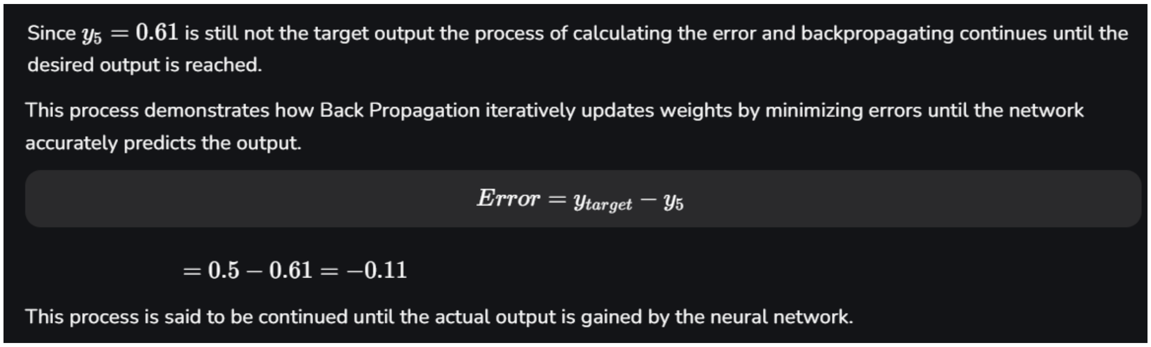
The updated weights are illustrated below

Through backward pass the weights are updated

After updating the weights the forward pass is repeated yielding:

- $y_3 = 0.57$
- $y_4 = 0.56$
- $y_5 = 0.61$

Since $y_5 = 0.61$ is still not the target output the process of calculating the error and backpropagating continues until the



Since $y_5 = 0.61$ is still not the target output the process of calculating the error and backpropagating continues until the desired output is reached.

This process demonstrates how Back Propagation iteratively updates weights by minimizing errors until the network accurately predicts the output.

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

$$= 0.5 - 0.61 = -0.11$$

This process is said to be continued until the actual output is gained by the neural network.