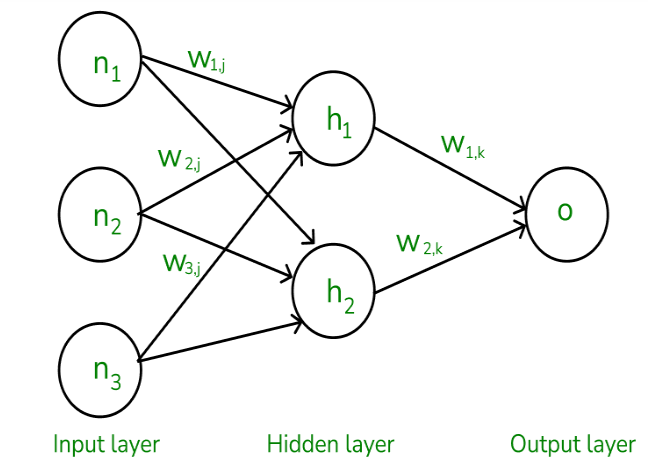
**How does Forward pass work?**

* In forward pass, initially the input is fed into the input layer. Since the inputs are raw data, they can be used for training our neural network.
* The inputs and their corresponding weights are passed to the hidden layer. The hidden layer performs the computation on the data it receives. If there are two hidden layers in the neural network, for instance, consider the illustration fig(a), h1 and h2 are the two hidden layers, and the output of h1 can be used as an input of h2. Before applying it to the activation function, the bias is added.
* To the weighted sum of inputs, the activation function is applied in the hidden layer to each of its neurons. One such activation function that is commonly used is ReLU can also be used, which is responsible for returning the input if it is positive otherwise it returns zero. By doing this so, it introduces the non-linearity to our model, which enables the network to learn the complex relationships in the data. And finally, the weighted outputs from the last hidden layer are fed into the output to compute the final prediction, this layer can also use the activation function called the softmax function which is responsible for converting the weighted outputs into probabilities for each class.



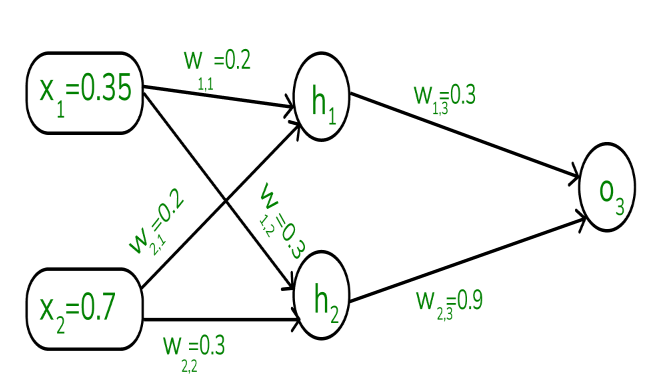
### How does backward pass work?

* In the backward pass process shows, the error is transmitted back to the network which helps the network, to improve its performance by learning and adjusting the internal weights.
* To find the error generated through the process of forward pass, we can use one of the most commonly used methods called mean squared error which calculates the difference between the predicted output and desired output. The formula for mean squared error is:  2*Meansquarederror*=(*predictedoutput*–*actualoutput*)2
* Once we have done the calculation at the output layer, we then propagate the error backward through the network, layer by layer.
* The key calculation during the backward pass is determining the gradients for each weight and bias in the network. This gradient is responsible for telling us how much each weight/bias should be adjusted to minimize the error in the next forward pass. The chain rule is used iteratively to calculate this gradient efficiently.
* In addition to gradient calculation, the activation function also plays a crucial role in backpropagation, it works by calculating the gradients with the help of the derivative of the activation function.

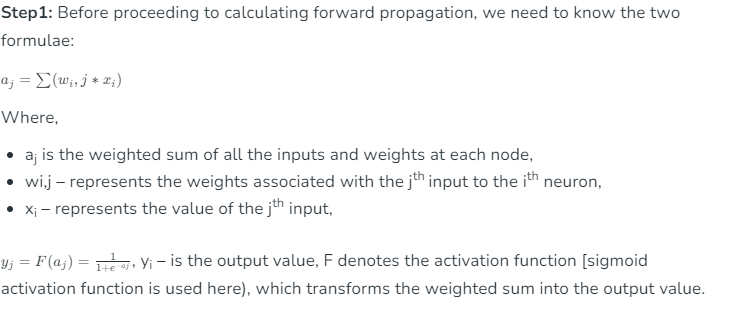
## Example of Backpropagation in Machine Learning

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

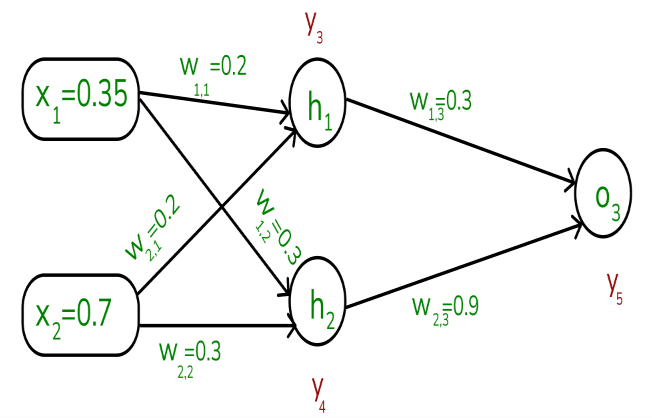
**Assume that the neurons have the sigmoid activation function to perform forward and backward pass on the network. And also assume that the actual output of y is 0.5 and the learning rate is 1. Now perform the backpropagation using backpropagation algorithm.**

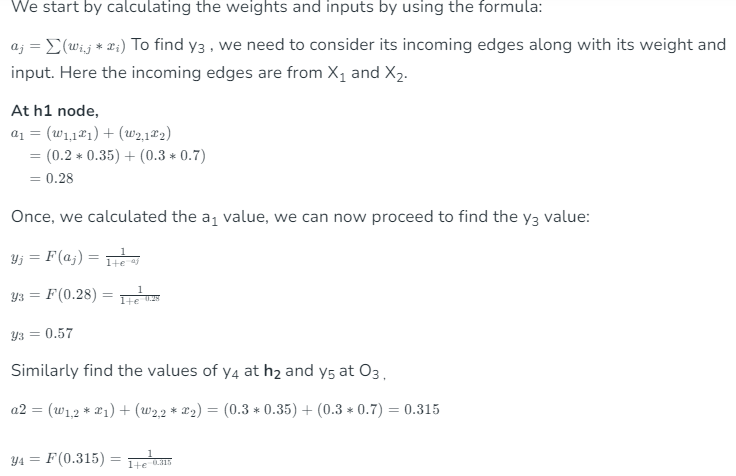


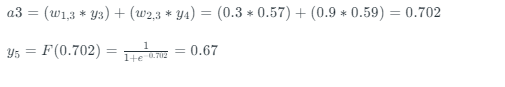
### Implementing forward propagation:

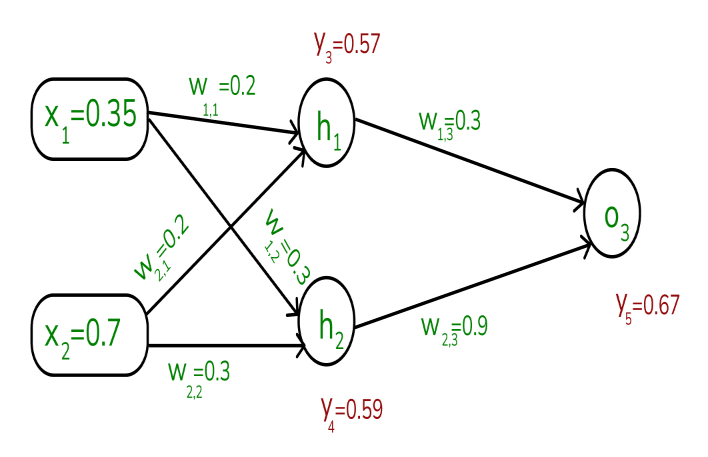


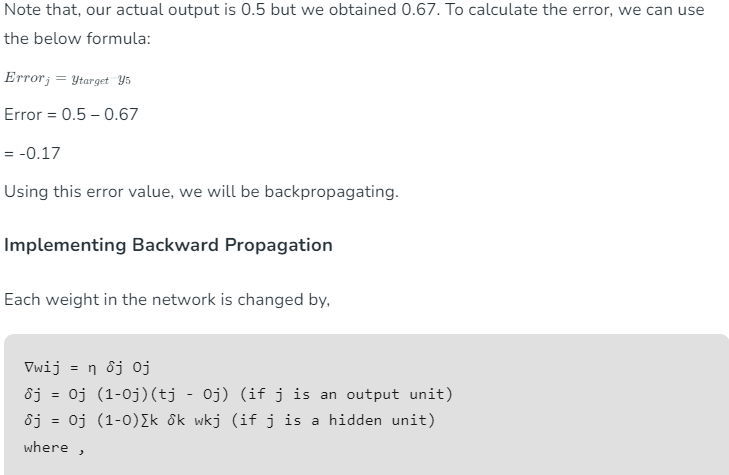
**Step 2: To compute the forward pass, we need to compute the output for y3 , y4 , and y5.**

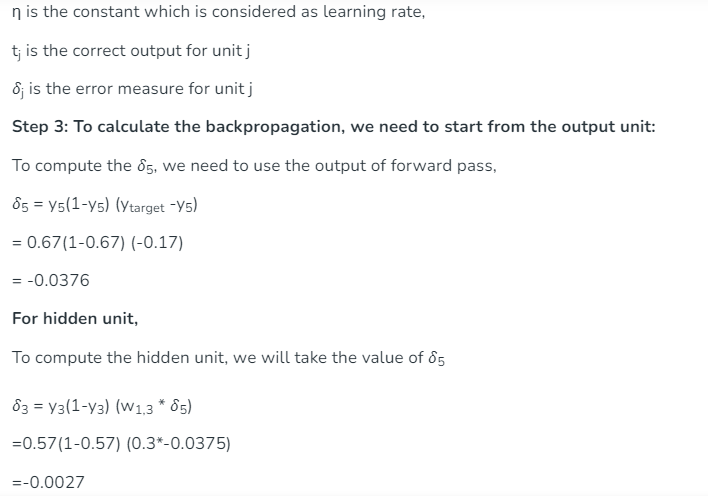


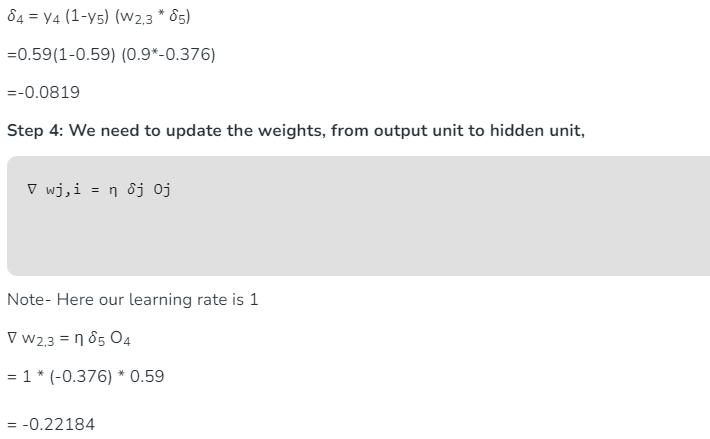


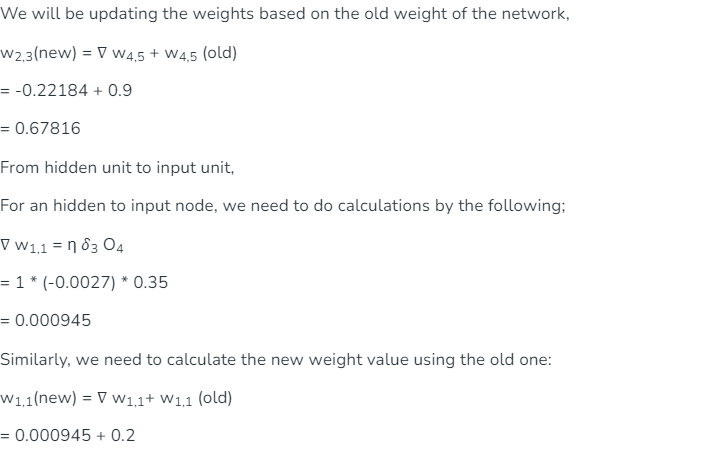


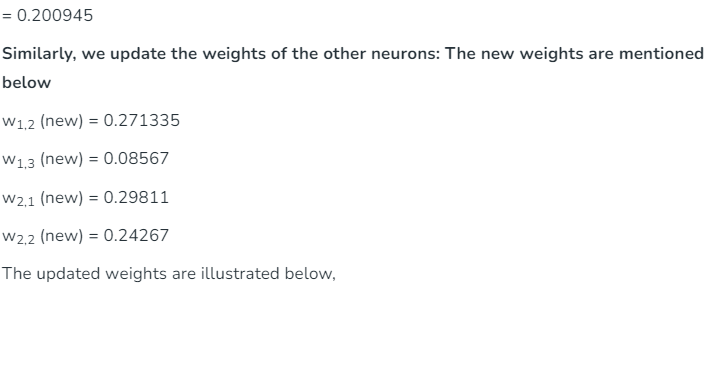


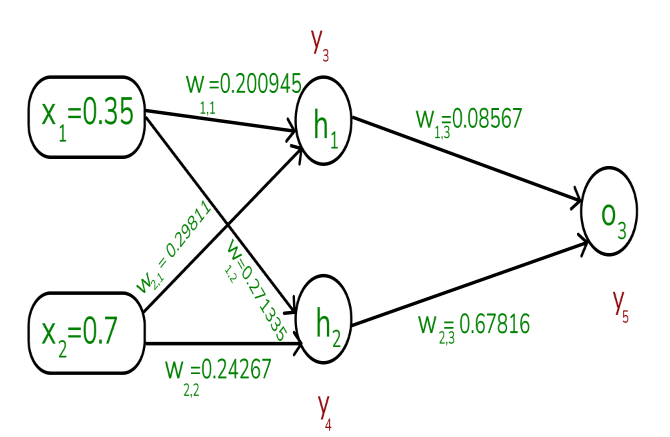


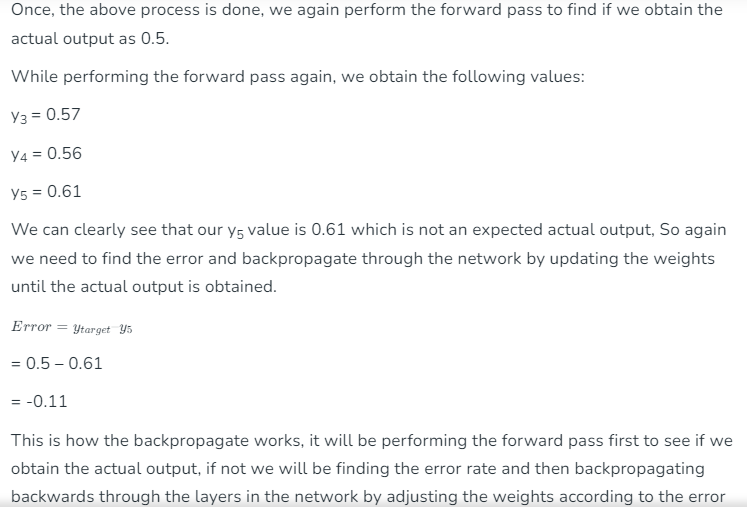












rate. This process is said to be continued until the actual output is gained by the neural network.