<https://7-hiddenlayers.com/wp-content/uploads/2020/06/NEURONS-IN-NUERAL-NETWORK.gif>

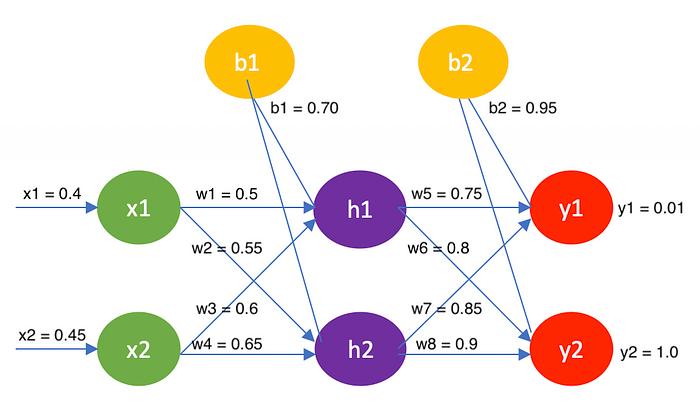
**Forward Propagation** is the way to move from the Input layer (left) to the Output layer (right) in the neural network. The process of moving from the right to left i.e backward from the Output to the Input layer is called the **Backward Propagation.**

Backward Propagation is the preferable method of adjusting or correcting the weights to reach the minimized loss function.

# How does the mathematics behind Back-propagation Algorithm work? Math Time!!!

we will be discussing the step by step approach for a forward pass and backward propagation. I will be using overleaf for mathematics, as it is discussed in depth.

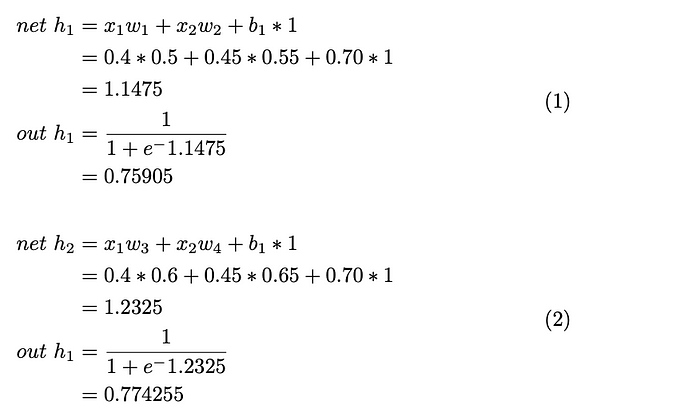
A 3 layer neural network will help us understand the propagation in a simplified manner.



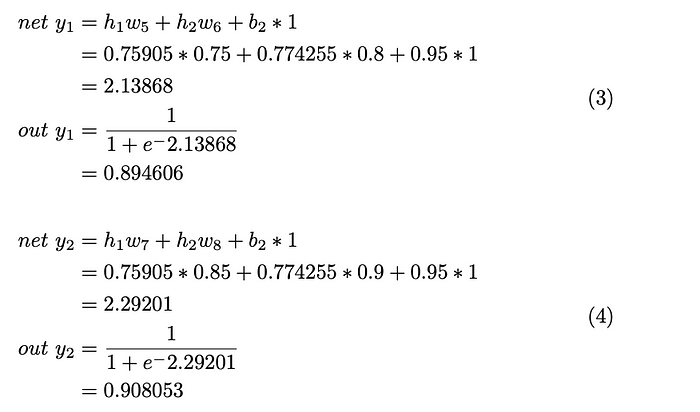
3-layer Neural Network

Let’s start by calculating the values of the hidden layer and the output layer.

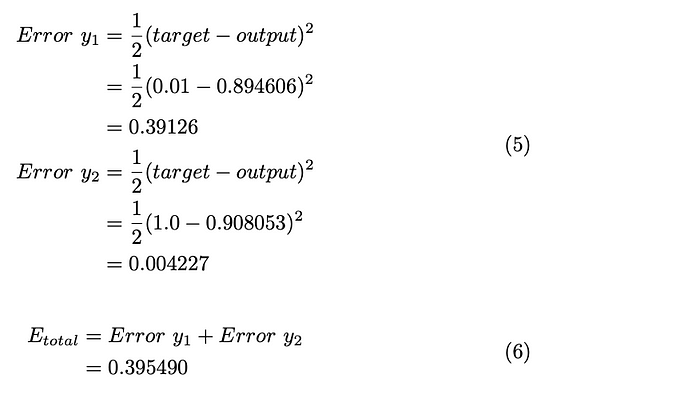
**Hidden layer: h1 and h2**



**Output layer: y1 and y2**



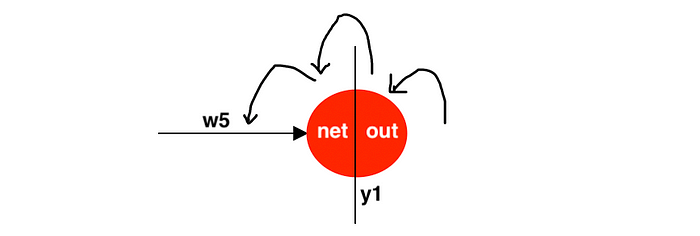
**Error in the node y1 and y2, add up to get the total error.**

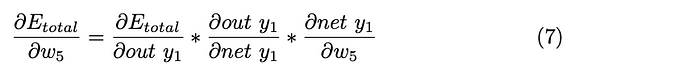


# Backward propagation

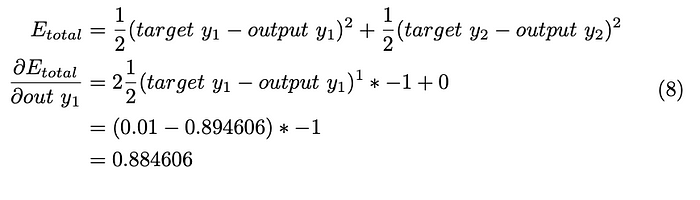
There is a difference between the values of the output layer (y1=0.01 and y2=1.0) and the calculated values (y1 = 0.894606 and y2 = 0.908053). To reduce the error, we have to adjust the weights accordingly.

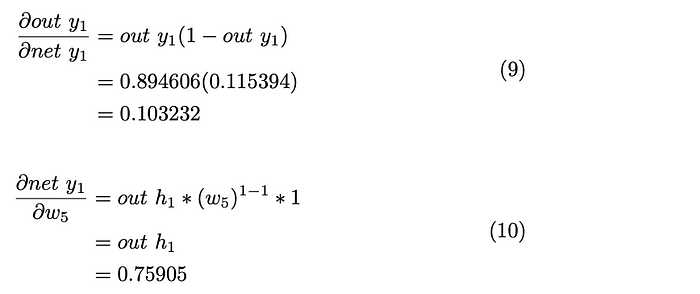
*Consider the weight w5 and apply the chain rule.*



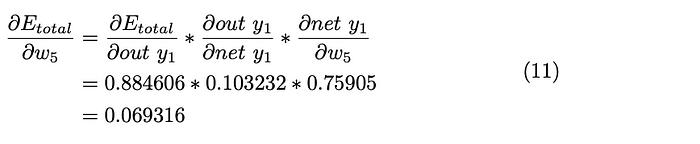


It gets complicated if we tackle all the partial differentiation at the same time. So, we will solve them one by one.





Substituting the equations 8,9,10 in eq.7

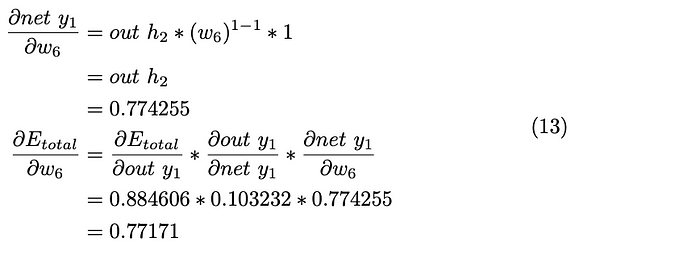


Finally, the weight w5 is updated. Here the learning rate is assumed to be η=0.4(It can vary from 0 to 1).



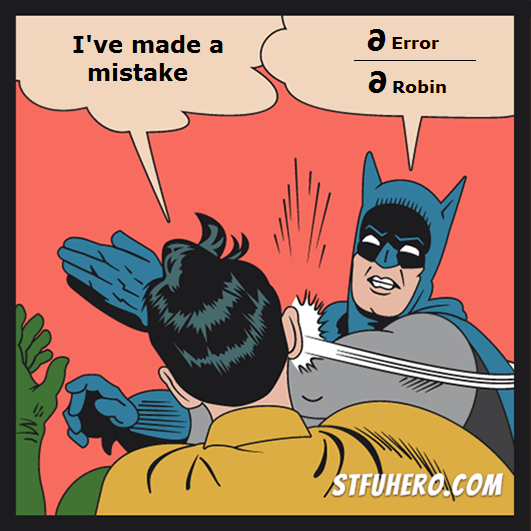
*Now, consider wight w6.*

We already have the values for the first two parts of the equation in equations 8 and 9. Solving for the last part of the equation and substitute back into equation 7.



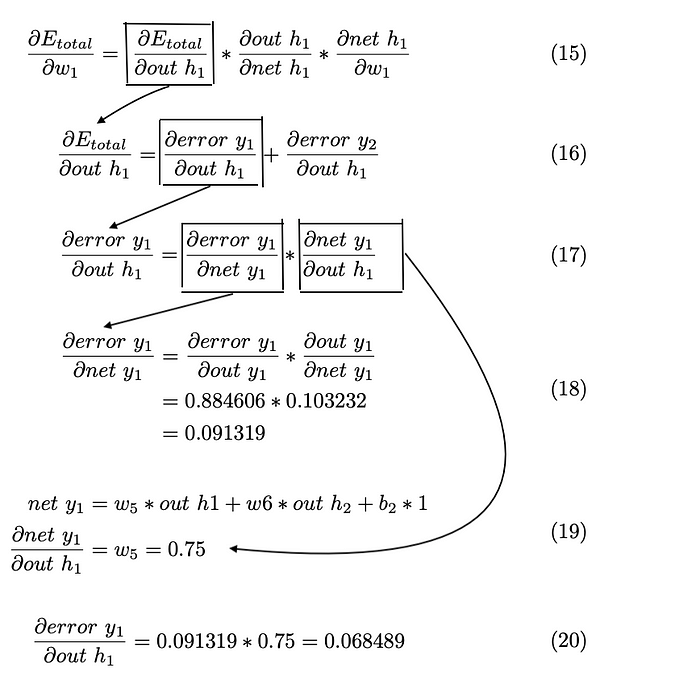
In the similar manner, we update the weights w7 and w8.



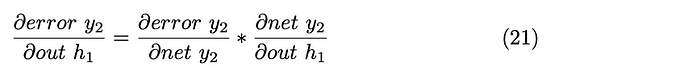


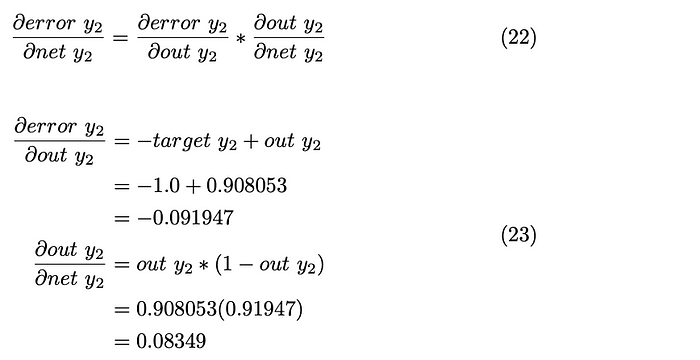
After solving the layer between the hidden and output layers, we will be updating the weights between the hidden and input layers.

Weight w1 can be updated through a series of equations. As it can be complicated, I have used arrows to indicate the continue chains of the equations.

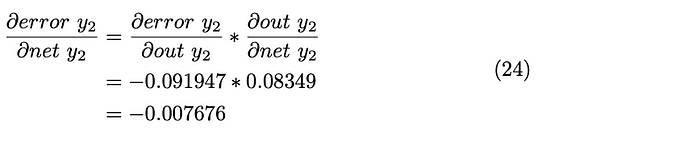


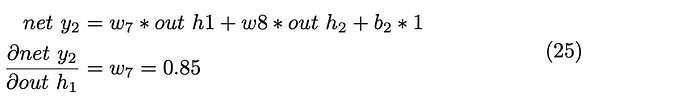
In the same way, the error in the out node y2 because of the hidden node h1 in eq.16 is calculated.



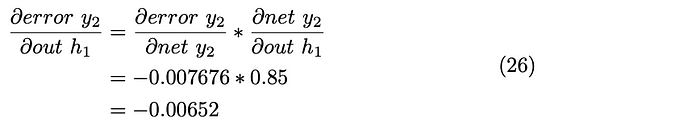


Substitute eq.23 in eq.22

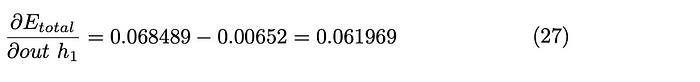


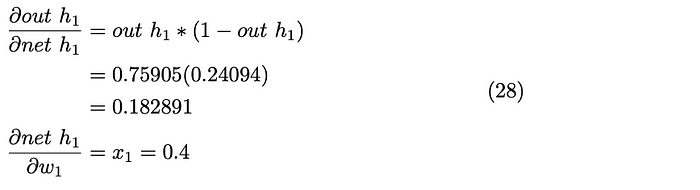


Now substitute eq.24 and 25 in eq.21

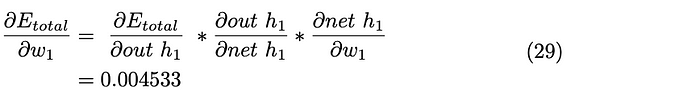


We now have the first part of the equation. By substituting the equations 26 and 20 into eq.16





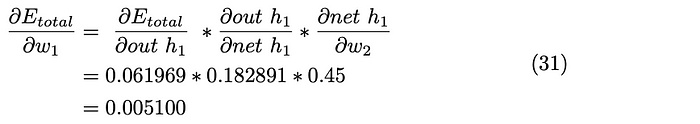
We get the final result by substituting the eq.27 and 28 into eq.15



Weight w1 is updated by multiplying the error with the learning rate.



Now, calculate the updated weight w2 as follows. But it is quicker as we already have the values from eq.27 and 28. The only change is the last part of the equation. i.e the weight.





Follow the same methods to calculate the updated weights w3 and w4.



This completes the first iteration. And be iterated until the error is minimized .