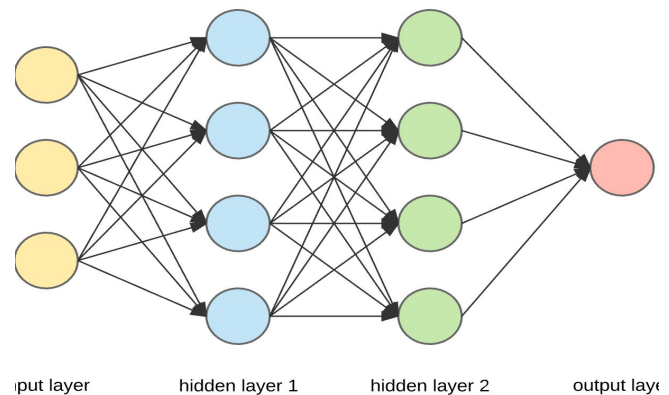


Understanding the Backpropagation Algorithm in Neural Networks

By: Adi Gudi

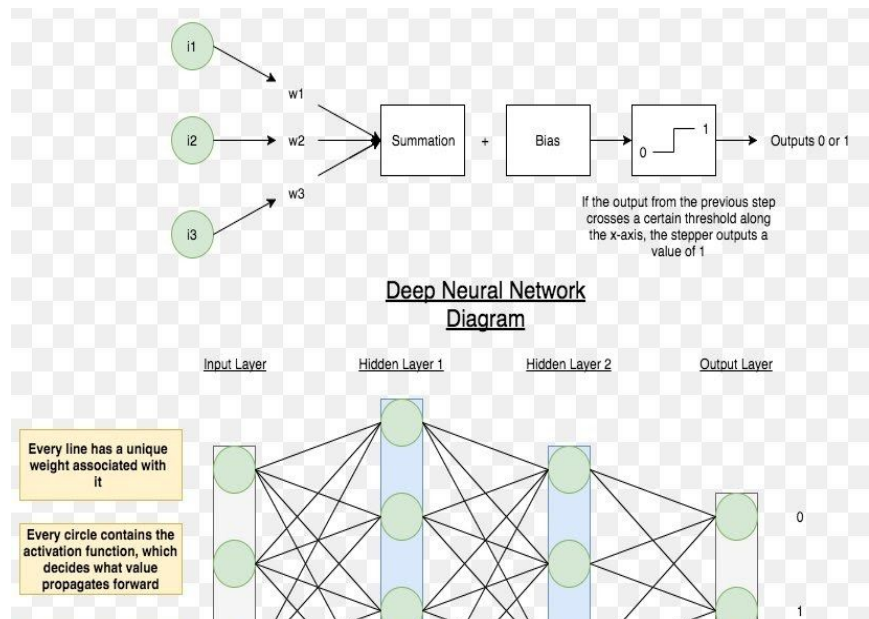
Neural Networks (and AI especially!) have been mentioned and/or talked about in numerous Computer Science classes and textbooks, but many do not know about how they work at a deeper level. You may know that in a Neural Network, there are input nodes, hidden layers, and output nodes, and that these are used to learn how to perform certain tasks without being explicitly programmed. However, the neural network almost never gives you the desired output after its first iteration; it can take hundreds, if not, thousands of iterations for the network to give you the desired output. The large number of iterations before finally achieving the desired output is due to this algorithm and fancy word, Backpropagation.

Backpropagation is the central process by which neural networks learn in order to produce the desired output. It is essentially a messenger communicating whether or not the network made a mistake when it made a prediction within the system. The network propagates the signal of the input layer through the network's hidden layer towards the output layer, and then backpropagates information about the error in reverse through the network so that it can alter the network's parameters (the hidden layer's colored nodes in the picture above).



A good analogy to model this process would be a large piece of artillery attempting to strike a distant object with a shell. When the neural network makes a prediction about an instance of data, the piece of artillery fires and the gunner attempts to discern the location of the shell, and how far it was from the target; that distance from the target is the measure of error and the bullet is the signal. The measure of error is then applied to the angle and direction of the gun (parameters), before it takes another shot. Backpropagation takes the error associated with an inaccurate guess by a neural network, and uses that error to adjust the network's parameters in order to reduce the error. How does it know the direction of less error?

In order to know the direction of less error, we must use differential calculus: we use the tangent of a changing slope to express the relationship of the parameter to the neural network's error. As the parameter changes, the error changes, and we want to move both variables in the direction of less error. A neural network has many parameters, so what one essentially measures are the partial derivatives of each parameter's contribution to the total change in error.



Additionally, neural networks have parameters that process the input data sequentially. Therefore, backpropagation establishes the relationship between the neural network's error and the parameters of the network's last hidden layer; then it establishes the relationship between the parameters of the network's last hidden layer to those of the network's second to last hidden layer, and so forth.

In conclusion, the backpropagation algorithm in neural networks is responsible for minimizing the error while the network is producing an output. If this algorithm was not implemented in neural networks, then everything that uses artificial intelligence would be extremely faulty and unsafe.

Citations:

<https://pathmind.com/wiki/backpropagation>

<https://towardsdatascience.com/how-does-back-propagation-in-artificial-neural-networks-work-c7cad873ea7>

<https://www.youtube.com/watch?v=aircAruvnKk>

<http://neuralnetworksanddeeplearning.com/chap2.html>

<https://medium.com/datathings/neural-networks-and-backpropagation-explained-in-a-simple-way-f540a3611f5e>

Images:

[1*livHOtvW8PSptrSb7OXpyA.jpeg](#)

[artificial-neural-network-deep-learning-activation-function-machine-learning-backpropagation-png-favpng-QaQRY1xgkcPU1Bpq0MrAtZnua.jpg](#)