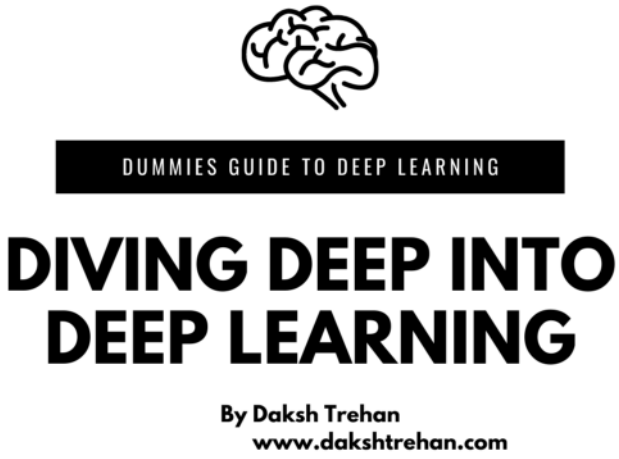
Deep Learning, Data Science

Diving Deep into Deep Learning

An Unconventional Guide to Deep Learning



We reside in a world, where we are constantly surrounded by deep learning algorithms be it for the good or worse cause. From the Netflix recommendation system to Tesla's autonomous cars, Deep Learning is leaving its stark appearance in our lives. This quaint & lucid technology is something that can impeach the whole thousand years old human civilization with just 4–5 years of training.

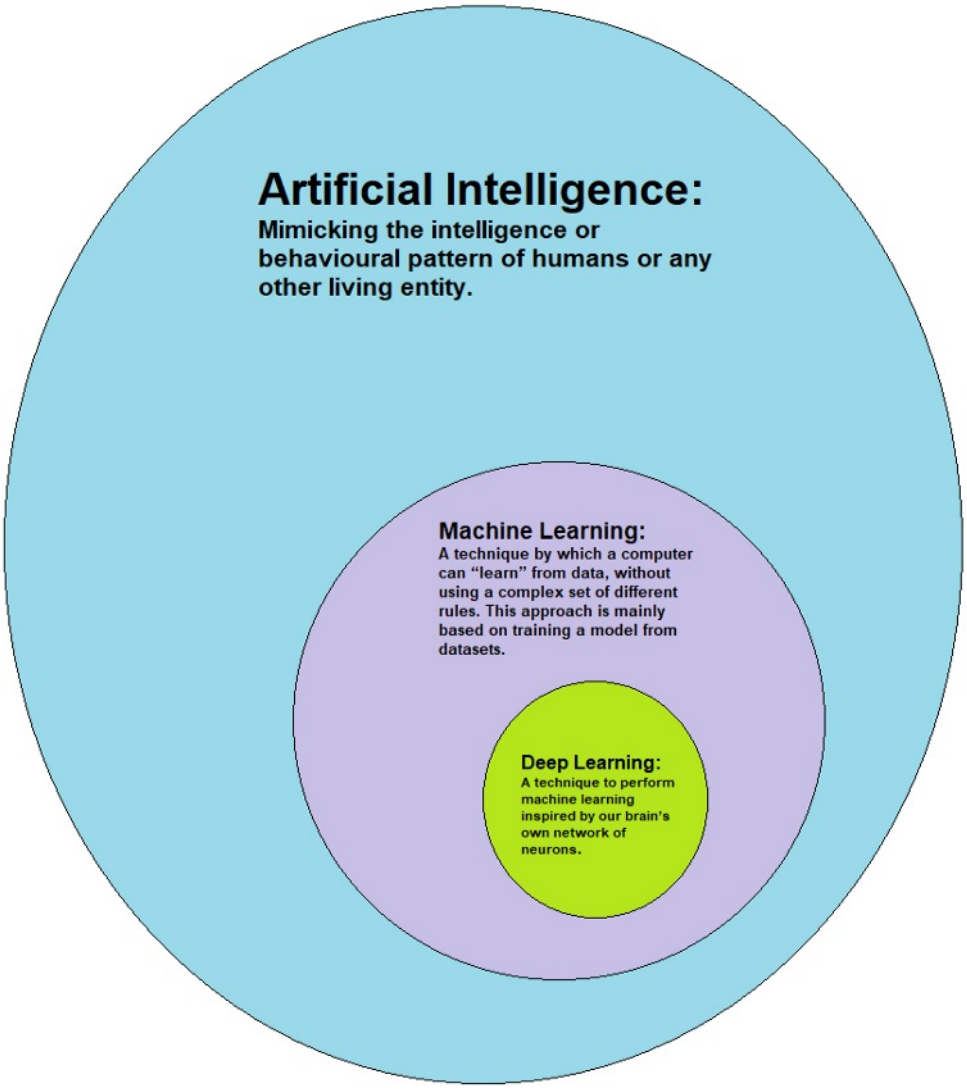
You've probably suggested this article because Deep Learning thinks you should see it.

Now let's jump right in!

Photo by Austin Neill on Unsplash

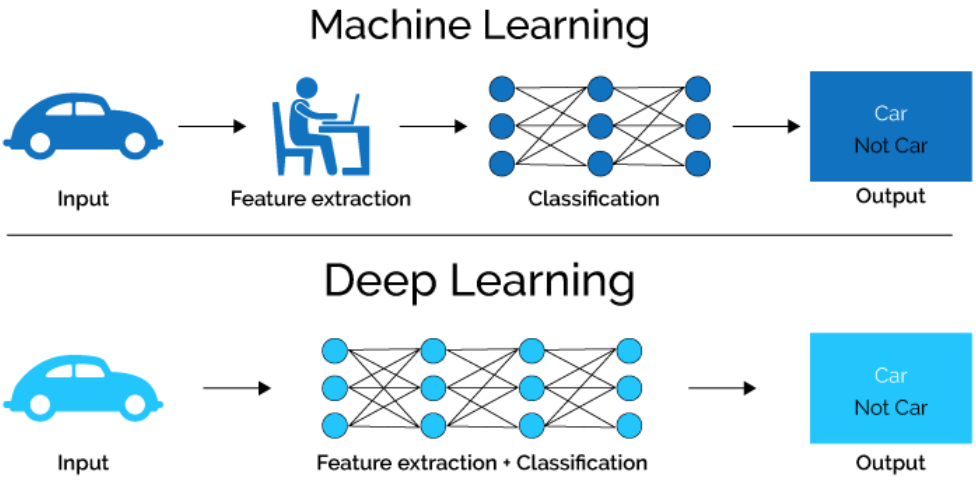
What is Deep Learning?

Deep learning is an extended arm of machine learning algorithms that teaches computers to do what is inherited naturally to humans i.e. learn by examples.



Artificial Intelligence vs Machine Learning vs Deep Learning, Source

How's it different from Machine Learning?



ML vs DL, Source

The major factor that distinguished Machine Learning & Deep Learning is the data representation and output. Machine Learning algorithms were developed for specific tasks whereas Deep Learning is more of a

data representation based upon different layers of a matrix, where each layer utilized the output provided by the previous layer.

The Intuition behind Deep Learning

The sole inspiration for deep learning is to imitate the human brain. It mimics the way the human brain filters out relevant information.

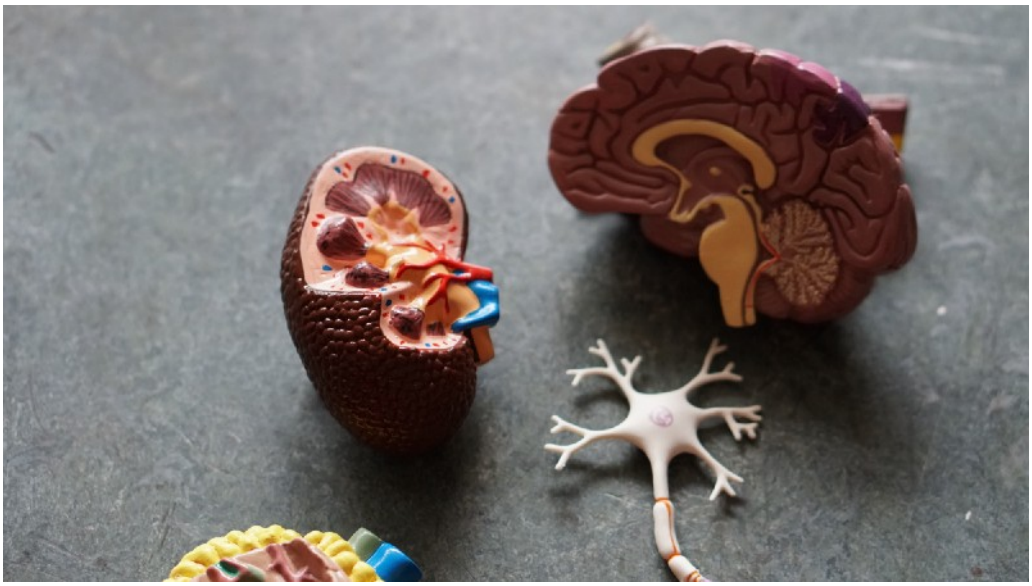


Photo by Robina Weermeijer on Unsplash

Our brain constitutes billions of biological neurons that are further connected to thousands of biological neurons in order to share and filter information. **Deep Learning is a way to recreate that arrangement in a way that works for our machines.**

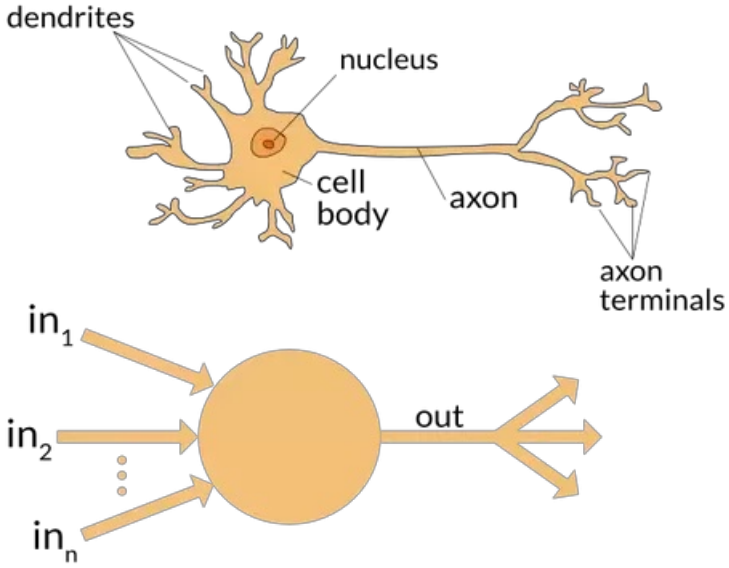
> **Using Deep learning, we try to develop an artificial neural network.**

In our brain, we have *Dendrites, Axon, cell bodies,* and *Synaptic gaps*. *T*he signal is passed from the axon(tail of a neuron) to dendrite(head of another neuron) with help of synapse.

Once dendrites get the signals, the cell body does some processing and then send the signal back to the axon, the modified signal is again sent to another neuron and this process repeats over and over again.

To recreate the magic in an artificial neural network, we introduced *Inputs, Weights, Outputs*, and *Activation*. A single neuron is useless for us, but when we get a bunch of them, you can recreate the magic. The artificial neural network purely impersonates the working of the natural brain. We feed weighted inputs to the neural layer, some processing happens at the same site that develops an output and which is further fed to the next layer and this happens recursively until we reach the last

layer.



Brain vs Artificial Neural Network, Source

The Neuron Magic

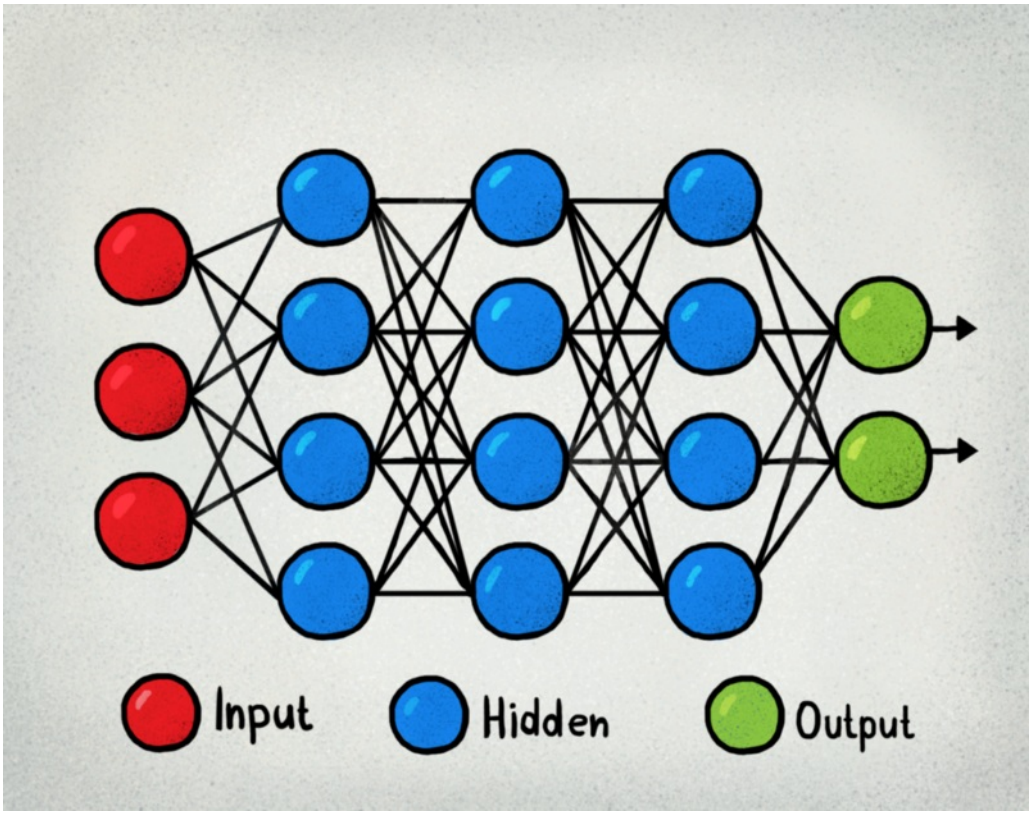
The input node is presented with numerical information. The higher the input, the greater its inclusion in the decision making factor.

Based on its weight, the activation value is calculated and passed o to the next node. Each node receives a weighted sum and modifies it based on the transfer function. The process is repeated over and over again until we reach the output layer.

How does Neural Network Work?

Neural Networks are multilayered network of neurons that are used to classify and predict.

It constitutes three types of layers wiz *Input Layer, Hidden Layer* & *Output Layer*.



Neural Network Structure, Source

The input layer accepts a numeral that lies between 0 to 1, that numeral act as activation for the input layer(layer1).

In the case of multiclass classification, the output layer consists of probabilities belonging to each class. The class with maximum probability is regarded as output for that particular input.

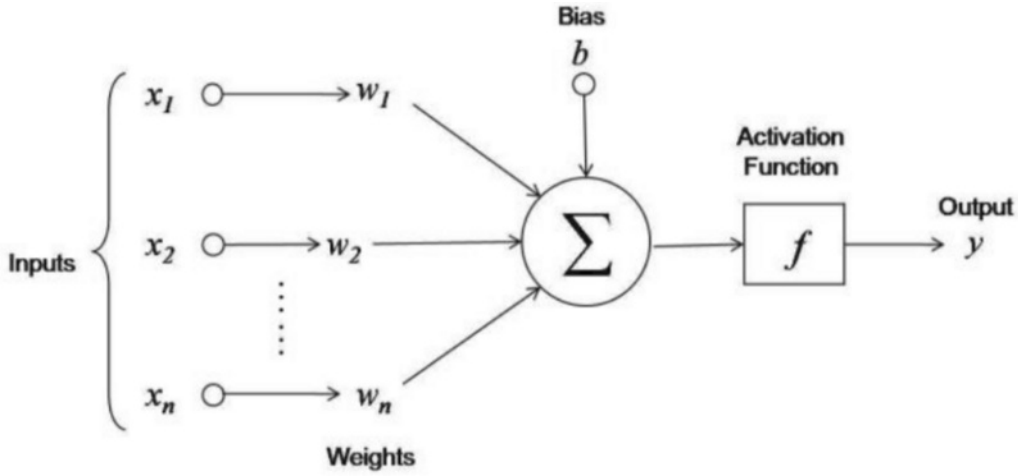
In the hidden layer, we compute activation using different activation functions such as TanH, Sigmoid, ReLu.

Working of Neural Network

A neuron's input is the sum of weighted outputs from all the neurons from the previous layer. Each input is a product of weight-associated and input to the current neuron. If there are 5 neurons in the previous layer, each neuron in the current layer will have 5 distinct weights.

A weighted sum is calculated for every neuron and is later added to bias before squishing to sigmoid function.

So in nutshell, activation for a neuron can be regarded as the sum of the product of weight from the previous layer and current input and a bias.



Activation function, Source

The activation function deciphers the input signals to output signals. It maps the output values on a range like 0 to 1. It is an approach to represent the rate of potential firing in the cell. The more activation function of any neuron, the more its firing capacity.

How do we choose Hidden Layers?

The number of hidden layers in your model is a hyperparameter i.e. it has to be chosen by you.

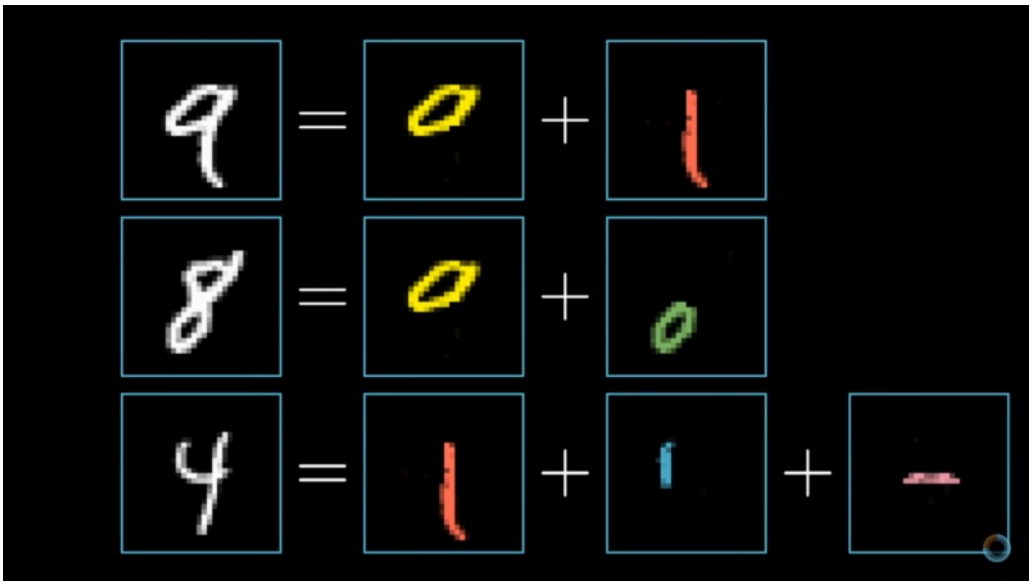
> The greater number of hidden layers ==> complex model ==> overfitting

Suppose, we want to build a handwritten digit recognition system. We feed in some images and expect the number present in images as our output.

When we see an orange, how does our brain recognize it? It identifies its shape like round, but we have several round fruits and objects, the next it analyzes its color and structure if it is something with an orange color and rough surface then we can categorize it as orange fruit but if it is orange and has smooth surface we can regard it as maybe an orange ball.

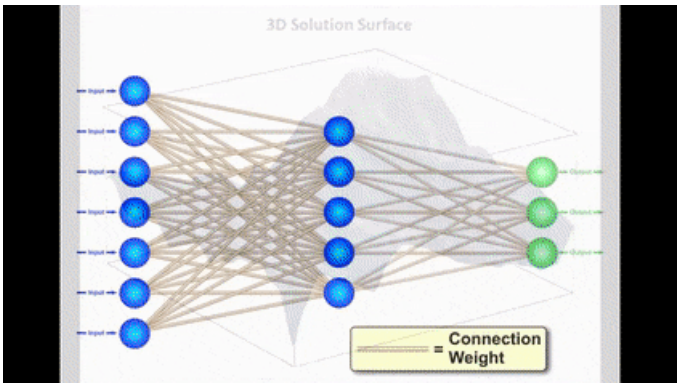
Photo by Anne Preble on Unsplash

Mimicking our natural neural network, an artificial neural network tries to follow a *Divide & Conquer approach* i.e. it will divide the problem into further sub-problems or being more specific it will divide digits into further shapes to classify the number.



Breaking digits into further sub-parts, Source: 3Blue1Brown

Each sub-part is expected to be recognized by each hidden layer. We can assume, first hidden layer to recognize (-) and next hidden layer to recognize (|), adding both sub-parts can be either 7 or 4. The third hidden layer will add outputs of both first and second hidden layer and will try to come up with probability of each class i.e. 10 classes(0 to 9) and class with maximum probability would be regarded as output.

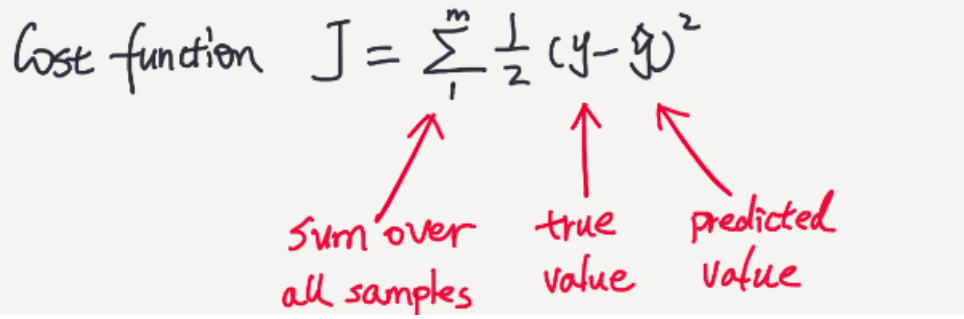


Working of Neural Network, Source

How does Neural Network Learn?

Once our model is trained, it is time to throw some unseen data to it and analyze its performance(cost function).

The way to calculate its performance is using Mean Squared Error(MSE), where we calculate the square of the difference between true and predicted value.

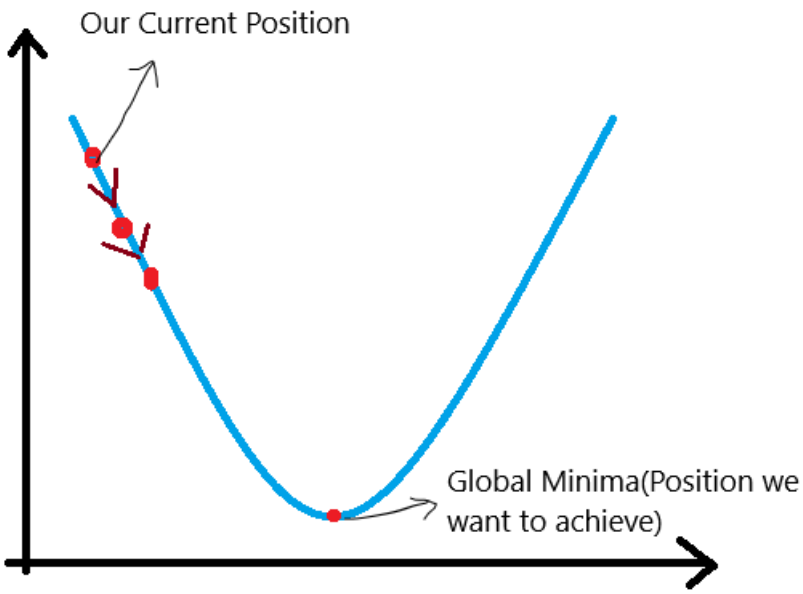


Cost function, Source

Once cost function is defined we can check how lousy our network is. Based on the cost function we can direct our model for some tweaks and improvisation.

By improvisation, we mean learning the right weights and biases.

To lower our loss, we use Gradient Descent. The idea behind Gradient Descent is to bring our loss to global minima.



Gradient Descent Algorithm, Source

Steps to implement Gradient Descent

1. Randomly initialize values.
2. Update values.

$$weight^{(new)} = weight^{(old)} - constant\frac{\partial J(\Theta)}{\partial weight}$$

3. Repeat until slope =0

Backpropagation

The backpropagation algorithm is probably a building block of a neural network. It utilizes a method called *chain rule* to effectively train our network.

Now at this point, we are well aware that our model is definitely doing something wrong and we need to inspect that, but, it is practically impossible to check for each and every neuron. But, also the only way possible for us to salvage our model is to retrograde.

If all major weights increase and irrelevant ones decrease, we will get our output with significantly great accuracies.

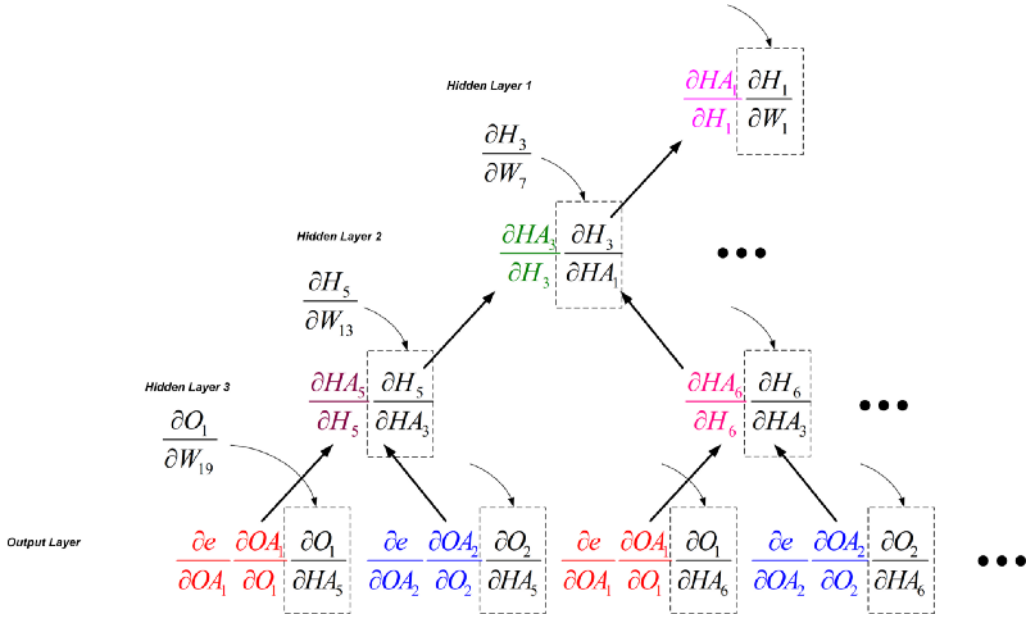
According to Hebbian Theory, “Neurons that fire together, wire together.”

The neural network particularly depends upon Weights, Activation, and biases. To develop a resilient and effective model, we have to change them.

Steps for Backpropagation

- We compute certain losses at the output and we will try to figure out who was responsible for that inefficiency.
- To do so, we will backtrack the whole network.
- Suppose, we found that the second layer(w3h2+b2) is responsible for our loss, and we will try to change it. But if we ponder upon our network, w3 and b2 are independent entities but h2 depends upon w2, b1 & h1 and h1 further depends upon our input i.e. x1, x2, x3...., xn. But since we don't have control over inputs we will try to amend w1 & b1.

To compute our changes we will use the chain rule.



Chain rule in BackPropagation, Source

Conclusion

Hopefully, this article will help you to understand about Deep Learning and Backpropagation in the best possible way and also assist you to its practical usage.

As always, thank you so much for reading, and please share this article if you found it useful!

Feel free to connect:

LinkedIn ~ https://www.linkedin.com/in/dakshtrehan/

Instagram ~ https://www.instagram.com/_daksh_trehan_/

Github ~ https://github.com/dakshtrehan

Follow for further Machine Learning/ Deep Learning blogs.

Medium ~ https://medium.com/@dakshtrehan

Want to learn more?

Detecting COVID-19 Using Deep Learning

The Inescapable AI Algorithm: TikTok

An insider's guide to Cartoonization using Machine Learning

Why are YOU responsible for George Floyd's Murder and Delhi Communal Riots?

Why Choose Random Forest and Not Decision Trees

Clustering: What it is? When to use it?

Start off your ML Journey with k-Nearest Neighbors

Naive Bayes Explained

Activation Functions Explained

Parameter Optimization Explained

Gradient Descent Explained

Logistic Regression Explained

Linear Regression Explained

Determining Perfect Fit for your ML Model

Cheers!