NN's Overview

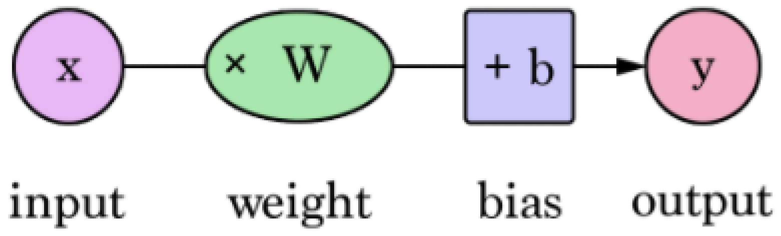
Start with why NN's are powerful

In traditional ML, we spend a lot of time doing Feature Engineering(FE) i.e, extracting features from the data and then passing them to the model so that it can perform better. The more important features that we are able to extract and pass to the model, the better it performs.

But when it comes to the DL which is a part of ML but we use Neural Networks (NN's) to train the model, we don't do FE. The NN's handles it for us which makes them very powerful and at the same time a bit complicated.

> Rough Idea of a NN

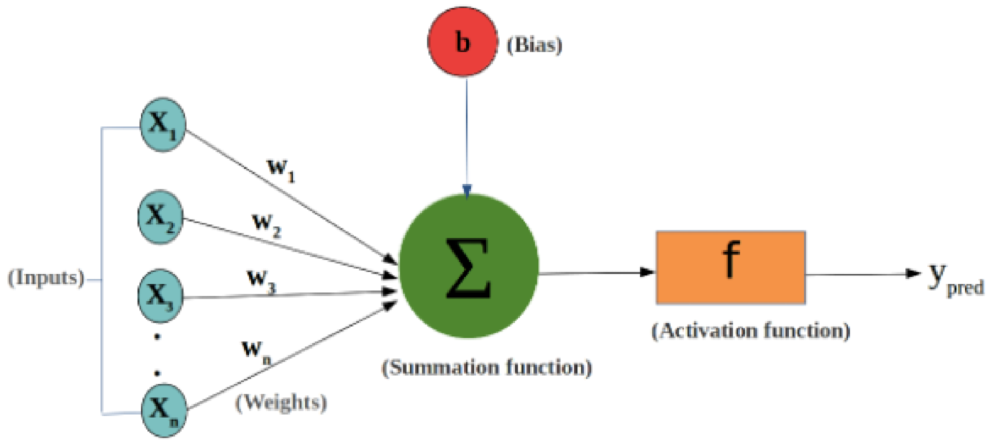
> A neural network is a series of nodes or neurons. Within each node is a set of inputs, weight, and a bias value. As an input enters the node, it gets multiplied by a weight value and the resulting output is either observed or passed to the next layer in the neural network.



Start with learning about a simple perceptron, which is nothing but a neural network with a single neuron in it. We call these architectures, other architectures we have are

- CNN's - Convolutional Neural Networks (Good with images)
- RNN's - Recurrent Neural Networks (Good with text-based)
- GAN's - Generative Adversarial Networks and etc. (Good with generating new data)

A simple perceptron will look like this,



Notice that we don't have any layers inside it because that's the whole point of perceptron. It is the building block of everything else. Once you understand this everything else is just an extension of this.

Read this to get a general idea: https://medium.com/technologymadeeasy/for-dummies-the-introduction-to-neural-networks-we-all-need-c50f6012d5eb

Now, once you have the general idea of the perceptron, I will try to explain as much as I can about the internal technical terms like weights, biases, and activation function.

Weights and bias:

To put it simply, Weight affects the amount of influence a change in the input will have upon the output. A low weight value will have no change on the input, and alternatively, a larger weight value will more significantly change the output.

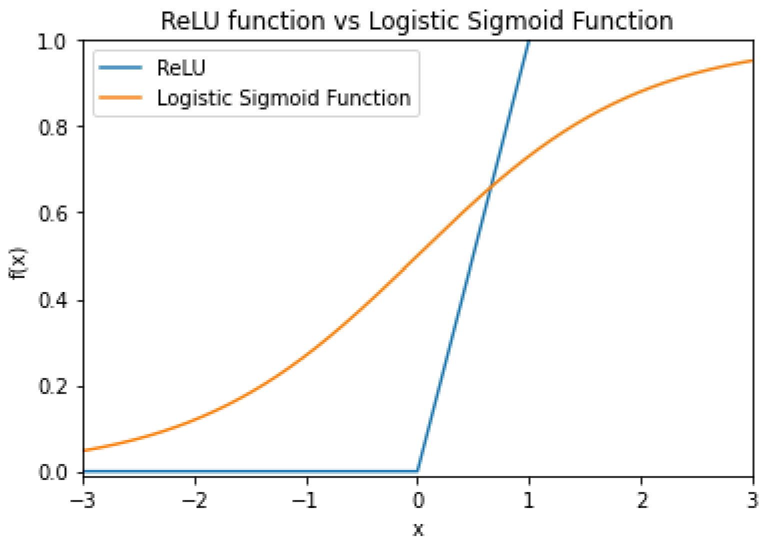
Bias represents how far off the predictions are from their intended value. Biases make up the difference between the function's output and its intended output. A low bias suggests that the network is making more assumptions about the form of the output, whereas a high bias value makes fewer assumptions about the form of the output.

Activation function:

Activation functions in computer science are inspired by the action potential in neuroscience. If the electrical potential between a neuron's interior and exterior exceeds a value called the action potential, the neuron in our brain undergoes a chain reaction which allows it to 'fire' and transmits a signal to neighboring neurons. The resultant sequence of activations, called a 'spike train', enables sensory neurons to transmit feeling from the fingers to the brain, and allows motor neurons to transmit instructions from the brain to the limbs.

An activation function in Neural Network is a function used in artificial neural networks which output a small value for small inputs, and a larger value if its inputs exceed a threshold. If the inputs are large enough, the activation function "fires", otherwise it does nothing. In other words, an activation function is like a gate that checks that an incoming value is greater than a critical number.

Two commonly used activation functions: the rectified linear unit (ReLU) and the logistic sigmoid function. The ReLU has a hard cutoff at 0 where its behavior changes, while the sigmoid exhibits a gradual change. Both tend to 0 for small x, and the sigmoid tends to 1 for large x.



Now for the big question. How do all of these things come together so that the NN's actually work?

Working:

Take any simple example of either tabular data or an image, all we need to know is there is an input that we want to pass into the NN for training and later do inference (predict).

So how does this training works?

The input is flattened as an array and passed into the network