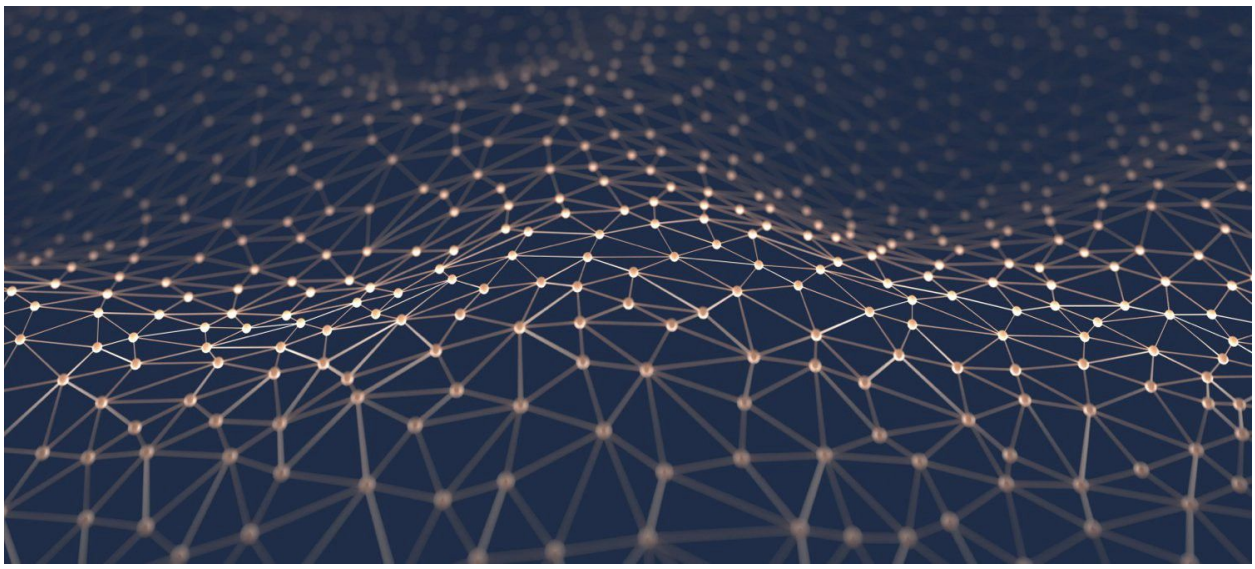
Machine Learning

Neural Networks & Deep Learning

Definition, layers, types of classifiers, LSTM...etc



Introduction

Neural Networks is a programming paradigm inspired by the biology of the human's brain and nervous system and how neurons behave to signals. *Neural networks* recognises patterns. They interpret the input data through machine perception and then label, or cluster, the raw input. You can think of *Neural networks* as a clustering and classification layer on top of the data you store and manage.

Digging deep...

As mentioned above, *Neural networks* help us cluster and classify data. They group unlabeled data according to similarities among the inputs and then they classify the data according to some other labeled dataset that they have trained on.

But what is *Deep Learning*?!

Deep learning is the name used for "stacked neural networks", or in other words, networks composed of several layers. *Deep learning* does not require data to be labeled in order to detect similarities. In fact, learning without labels is known as *unsupervised learning*. And since most of the data in the world is unlabeled, *Deep learning* algorithms must be trained in order to achieve higher accuracy. In fact, the more data an algorithm ican train on, the more accurate it will be.

With classification, *Deep learning* also finds and constructs correlations between data, exposing *Static prediction*. *Deep learning* can establish correlations between current events and future events, if and only if, it has been exposed to enough "correct" data. So in some sense, we can say that, given a time series, *Deep learning* may feed on some input data and predict the event most likely to occur next. Therefore, we, thanks to *Neural networks,* are moving towards a world of fewer surprises.

What are they made of ...?!

Layers are basically a set of *Nodes.* And a *Neural network,* is composed of several layers. *Nodes* are simply a place where some computation happens, patterned on a neuron in the human brain, which fires when some stimuli arrives. A *Node* combines input data with some weights for each input and either amplifies or dampens that input, resulting in a probabilistic function for the significance of each input. Those input-weight products are then summed up and then the sum is passed to an *Active function node* to determine whether and to what extent the signal should progress further through the network to affect the outcome.