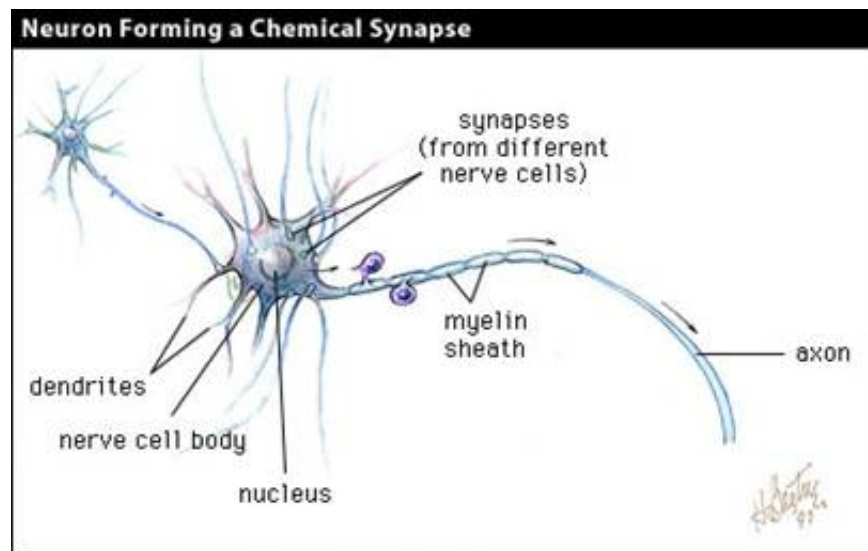


ai - junkie

So, what exactly is a Neural Network?

A neural network is mans crude way of trying to simulate the brain electronically. So to understand how a neural net works we first must have a look at how the old grey matter does its business...

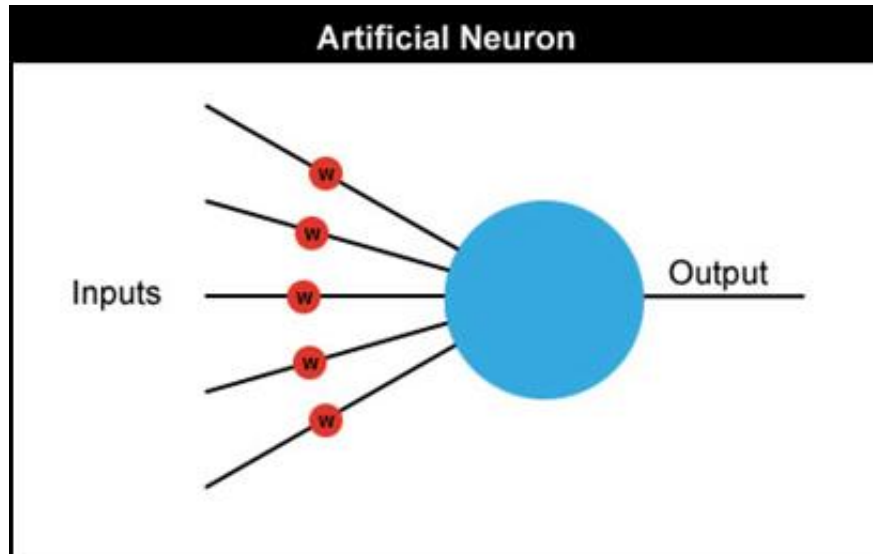
Our brains are made up of about 100 billion tiny units called *neurons*. Each neuron is connected to thousands of other neurons and communicates with them via electrochemical signals. Signals coming into the neuron are received via junctions called *synapses*, these in turn are located at the end of branches of the neuron cell called *dendrites*. The neuron continuously receives signals from these inputs and then performs a little bit of magic. What the neuron does (this is over simplified I might add) is sum up the inputs to itself in some way and then, if the end result is greater than some threshold value, the neuron fires. It generates a voltage and outputs a signal along something called an *axon*. Don't worry too much about remembering all these new words as we won't be using many of them from this moment onwards, just have a good look at the illustration and try to picture what is happening within this simple little cell.



Thanks to w hoever drew this picture. I hope I haven't offended you by using it

Neural networks are made up of many artificial neurons. An artificial neuron is simply an electronically modelled biological neuron. How many neurons are used depends on the task at hand. It could be as few as three or as many as several thousand. One optimistic researcher has even hard wired 2 million neurons together in the hope he can come up with something as intelligent as a cat although most people in the AI community doubt he will be successful (*Update: he wasn't!*). There are many different ways of connecting artificial neurons together to create a neural network but I shall be concentrating on the most common which is called a *feedforward*

network. So, I guess you are asking yourself, *what does an artificial neuron look like?* Well here you go:



Each input into the neuron has its own weight associated with it illustrated by the red circle. A weight is simply a floating point number and it's these we adjust when we eventually come to train the network. The weights in most neural nets can be both negative and positive, therefore providing excitatory or inhibitory influences to each input. As each input enters the nucleus (blue circle) it's multiplied by its weight. The nucleus then sums all these new input values which gives us the *activation* (again a floating point number which can be negative or positive). If the activation is greater than a threshold value - let's use the number 1 as an example - the neuron outputs a signal. If the activation is less than 1 the neuron outputs zero. This is typically called a *step* function (take a peek at the following diagram and have a guess why).

