ai-junkie

Now for some maths

I now have to introduce you to some equations. I'm going to try to keep the maths down to an absolute minimum but it will be useful for you to learn some notation. I'll feed you the maths little by little and introduce new concepts when we get to the relevant sections. This way I hope your mind can absorb all the ideas a little more comfortably and you'll be able to see how the maths are put to work at each stage in the development of a neural net.

A neuron can have any number of inputs from one to n, where n is the total number of inputs. The inputs may be represented therefore as x_1, x_2, x_3… x_n. And the corresponding weights for the inputs as w_1, w_2, w_3… $w_{n.}$ Now, the summation of the weights multiplied by the inputs we talked about above can be written as $x_1w_1 + x_2w_2 + x_3w_3$ …. + $x_nw_{n,}$ which I hope you remember is the activation value. So…

$$a = x_1w_1+x_2w_2+x_3w_3... +x_nw_n$$

Fortunately there is a quick way of writing this down which uses the Greek capital letter sigma Σ, which is the symbol used by mathematicians to represent summation.

$$a = \sum_{i=0}^{i=n} w_i x_i$$

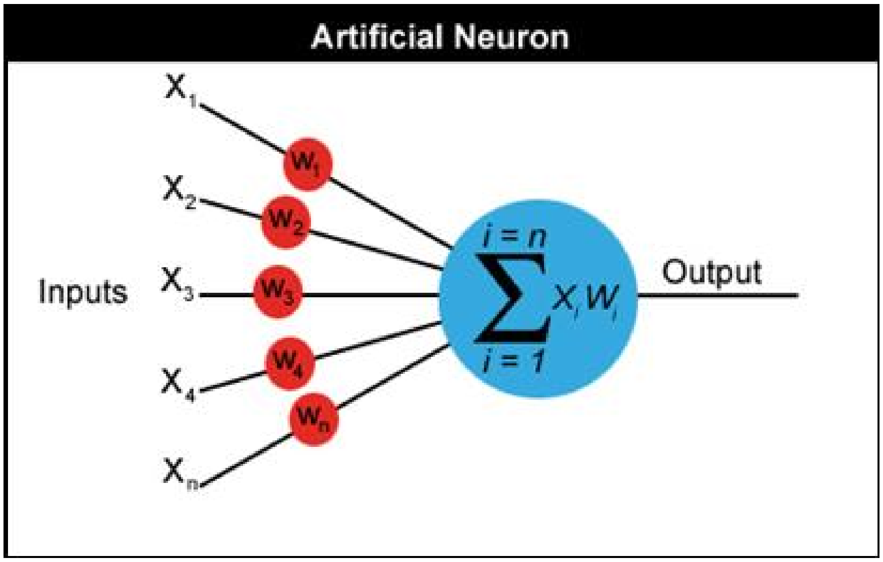
Maybe just to clarify what this means I should write it out in code. Assuming an array of inputs and weights are already initialized as x[n] and w[n] then:

```
double activation = 0;

for (int i=0; i<n; i++)
{
   activation += x[i] * w[i];
}
```

Got it? Now remember that if the activation > threshold we output a 1 and if activation < threshold we output a 0.

Let me illustrate everything I've shown you so far with a diagram.



Please ensure that you understand exactly how to calculate the activation value before you move on.

---

1 2 4 5 6 7 8 Next Home