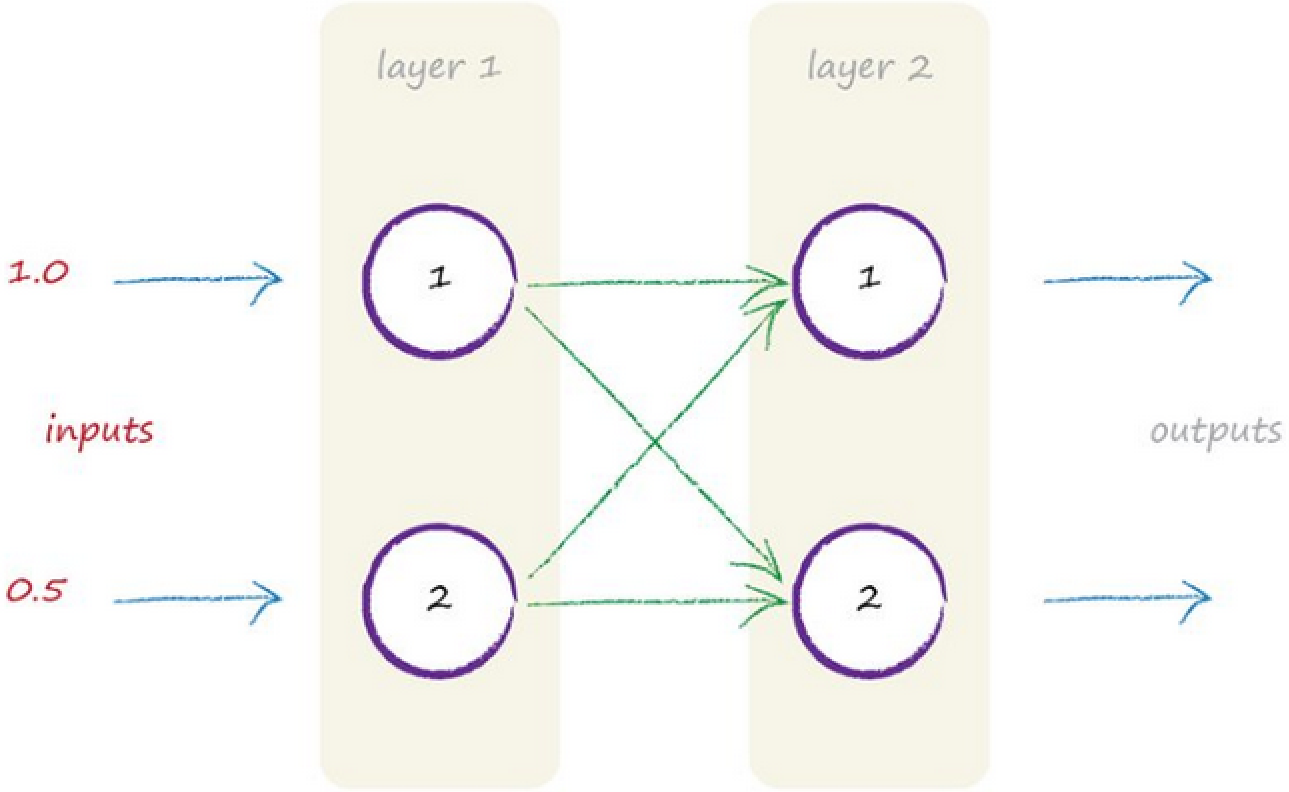
Understanding Neural Networks

Let imagine a neural network with 2 layers and 2 inputs that are 1 and 0.5.



Each node turns the sum of the inputs into an output using an activation function. We'll also use the sigmoid function y = - 1 / (1 + e*) that we saw before, where x is the sum of incoming signals to a neuron, and y is the output of that neuron.

What about the weights? That's a very good question - what value should they start with? Let's go with some random weights:

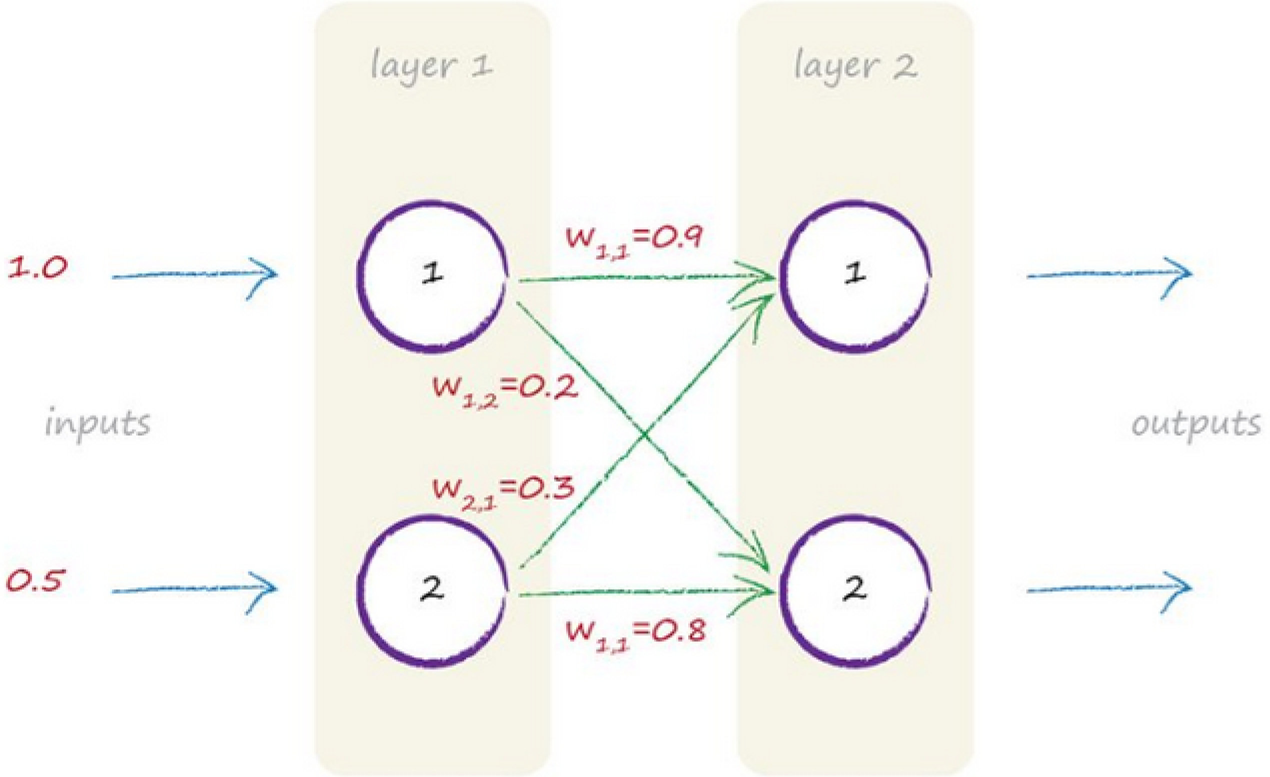
- w1,1 = 0.9

- w1,2 = 0.2

- w2,1 = 0.3

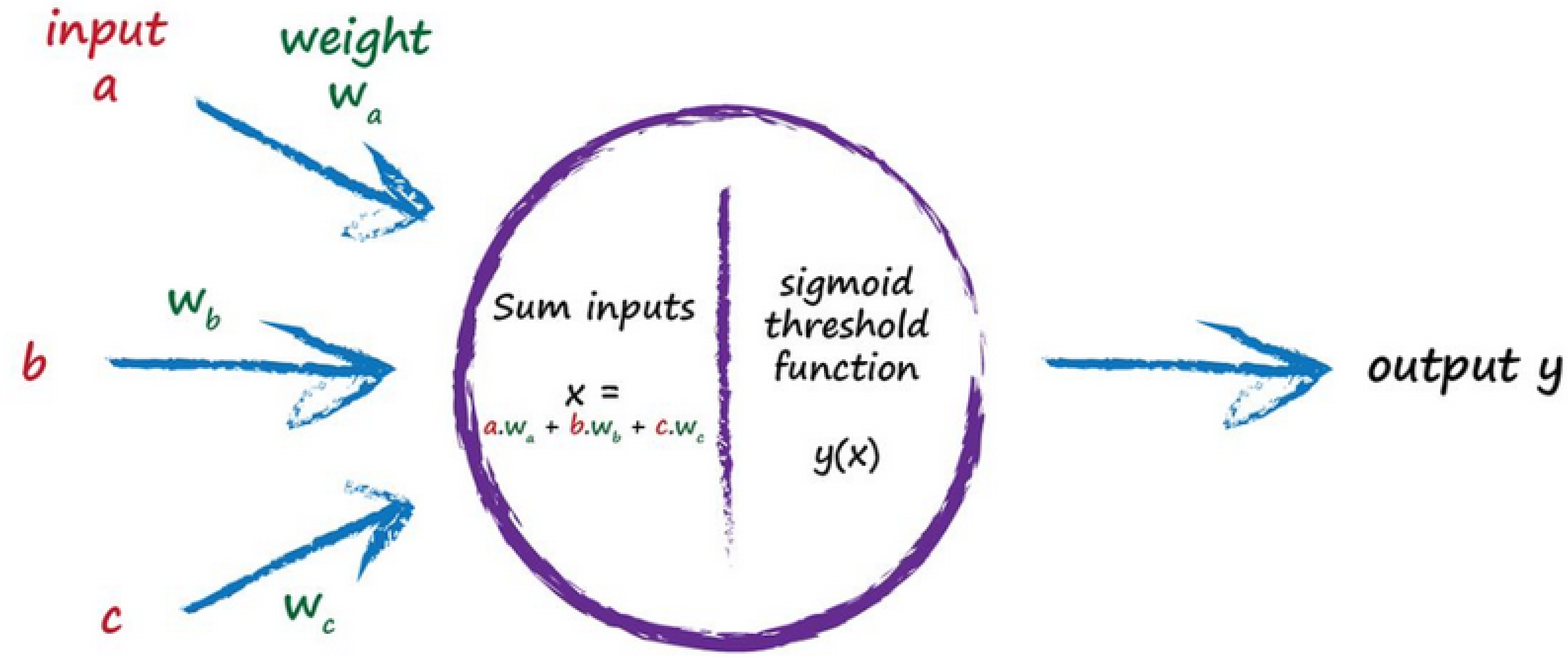
- w2,2 = 0.8

Going with random is not such a bad idea. just like previousely we chose slope of the linear classifier to be random.



The first layer of nodes is the input layer and does not do any calculations, it just passes it to the next layer.

Well, Next is the second layer where we do need to do some calculations. For each node in this layer we need to work out the combined input. Remember that sigmoid function, y =(1/1+e^-x) the x in that function is the combined input into a node. That combination was the raw outputs from the connected nodes in the previous layer, but moderated by the link weights. The following diagram is like the one we saw previously but now includes the need to moderate the incoming signals with the link weights.n.



x = (output from first node * link weight) + (output from second node * link weight)

x = (1.0 * 0.9) + (0.5 * 0.3)

x = 0.9 + 0.15

x = 1.05

If we didn't moderate the signal, we'd have a very simple addition of the signals 1.0 + 0.5, but we don't want that. It is the weights that do the learning in a neural networks as they are iteratively refined to give better and better results.

So, we've now got x = 1.05 for the combined moderated input into the first node of the second layer. We can now, finally, calculate that node's output using the activation function y = 1/(1 +e^-x) . Feel free to use a calculator to do this. The answer is y = 1/(1 + 0.3499) = 1/ 1.3499. So y = 0.7408.

That's great work! We now have an actual output from one of the network's two output nodes. Let's do the calculation again with the remaining node which is node 2 in the second layer. The combined moderated input x is:

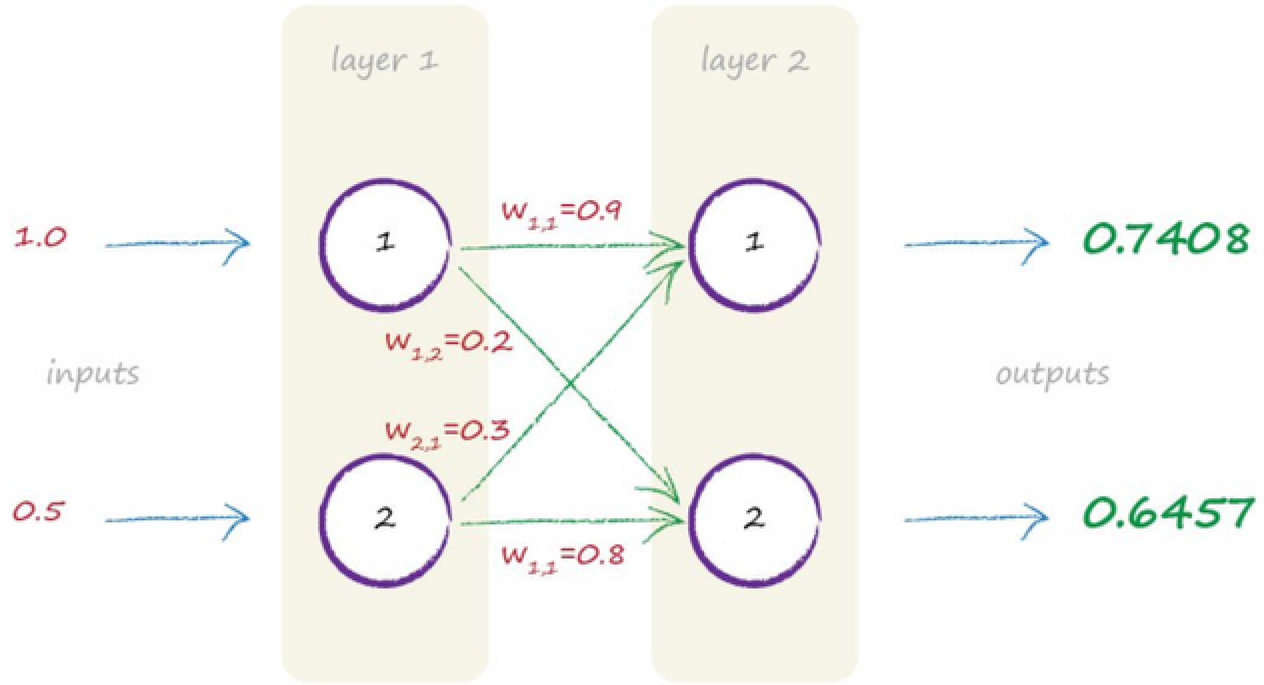
x = (output from first node * link weight) + (output from second node * link weight)

x = (1.0 * 0.2) + (0.5 * 0.8)

x = 0.2 + 0.4

x = 0.6

So now we have x, we can calculate the node output using the sigmoid function. 1/(1+0.5488)=1/1.5488=0.6457



That was a fair bit of work just to get two outputs from a very simplified network. I wouldn't want to do the calculations for a larger network by hand at all! Luckily computers are perfect for doing lots of calculations quickly and without getting bored.

The concise approach would be to use matrices, we would look into it some other time.