Make Your Own NN p35-p63

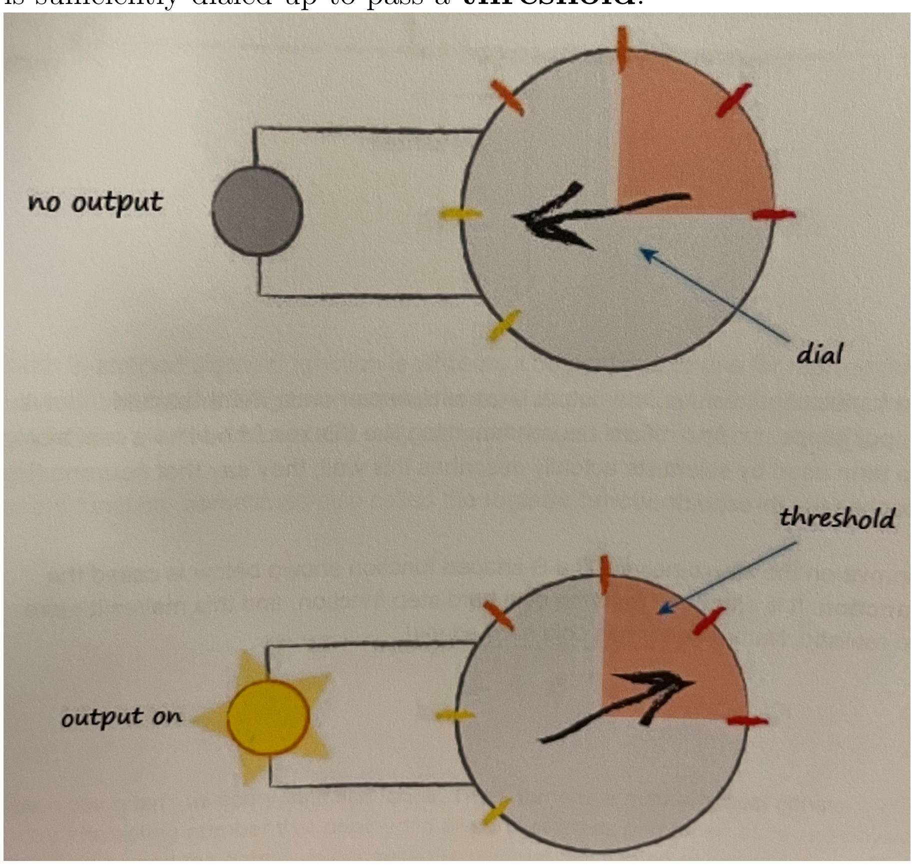
2021.1.27

Key Points

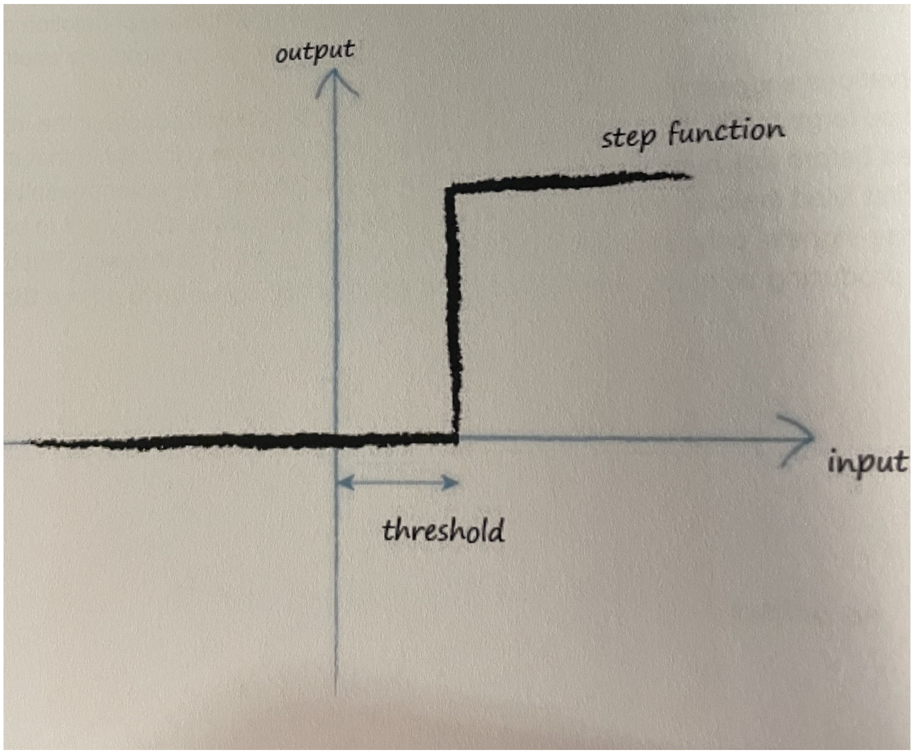
⋆ A simple linear classifier can not separate data where that data itself is not governed by a single linear process. For example, data governed by the logical XOR operator.

⋆ However the solution is easy, you just use multiple linear classifiers to divide up data that can not be separated by a single straight dividing line.

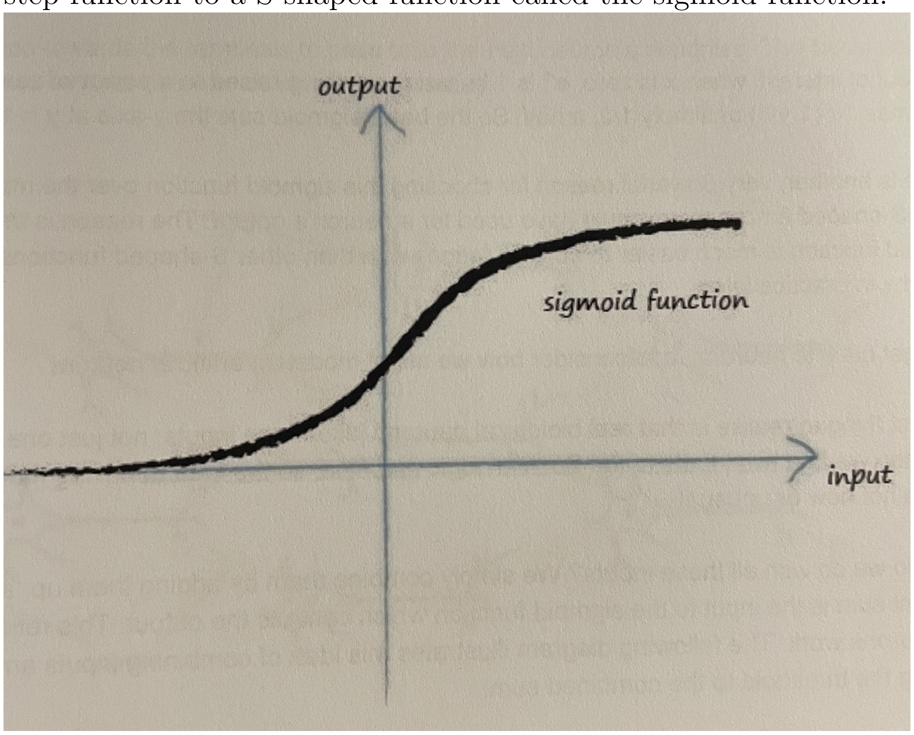
The neurons do not want to be passing on tiny noise signal, only emphatically strong intentional signals and the idea of only producing an output signal if the input is sufficiently dialed up to pass a **threshold**.



A function that takes the input signal and generates an output signal, but takes into account some kind of threshold is called an **activation function**. Mathematically, there are many such activation functions that could achieve this effect. A simple **step function** could do this.

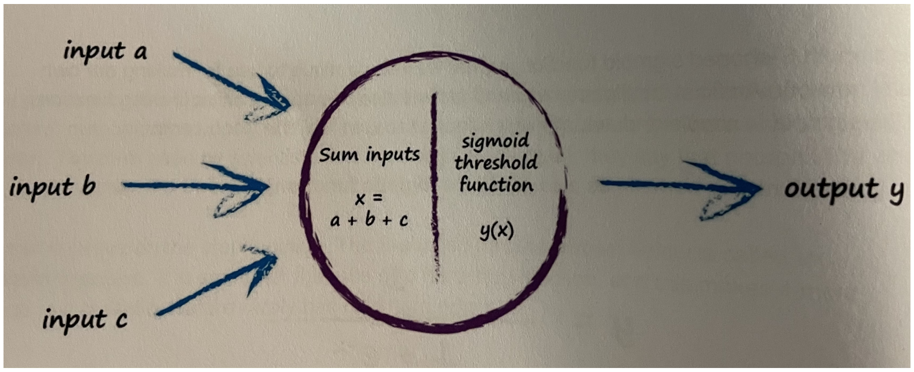


In the figure above, for low input values, the output is zero. However once the threshold input is reached, output jumps up. An artificial neuron behaving like this is called neurons fire when the input reaches the threshold. We can improve on the step function to a S-shaped function called the sigmoid function.

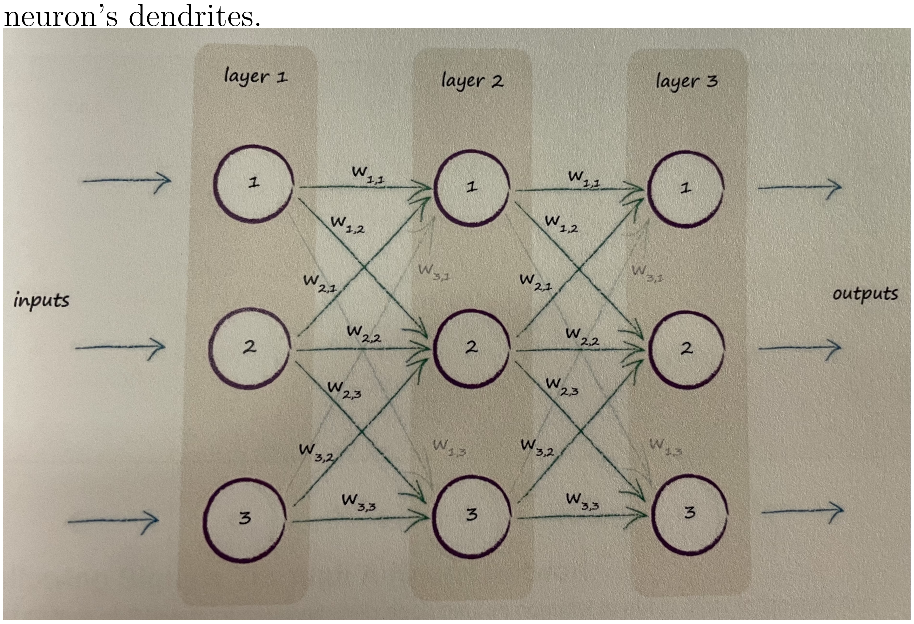


The sigmoid function sometimes also called the logistic function is

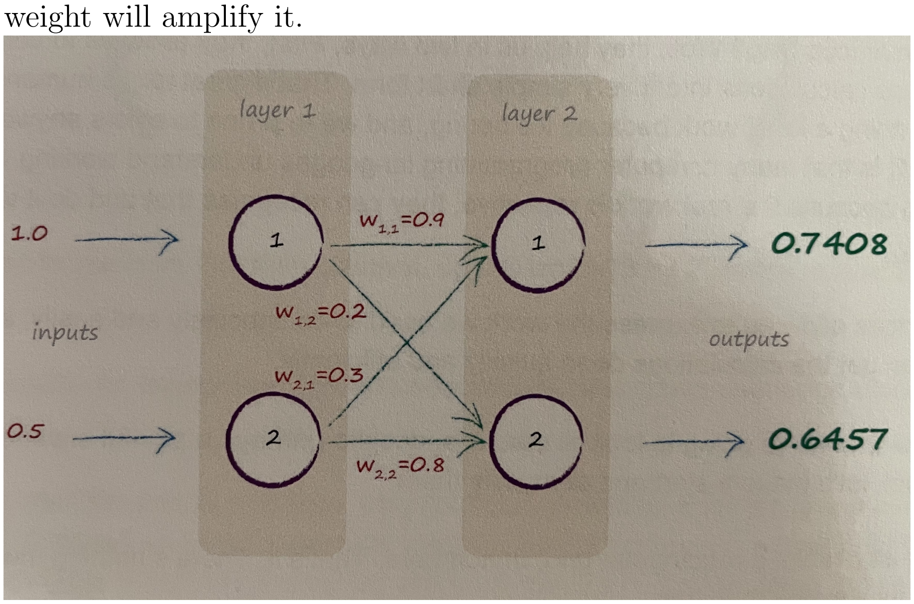
$$y = \frac{1}{1 + e^{-x}}$$

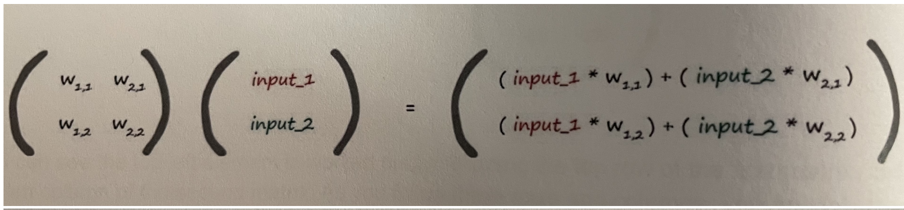


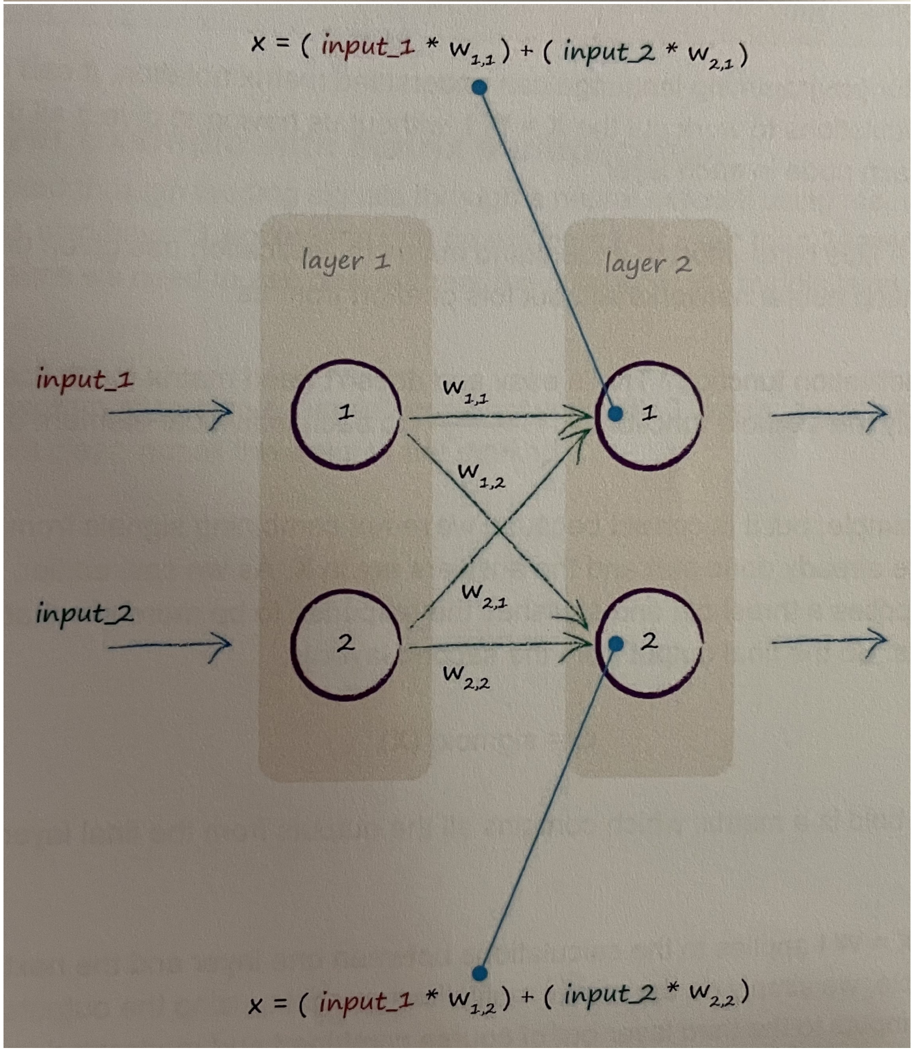
The electrical signals(input a,b,c) are collected by the dendrites and these combine to form a stronger electrical signal. If it is strong enough to pass the threshold, the neuron fires a signal down the axon toward the terminals to pass onto the next neuron's dendrites.



In figure above, you can see three layers, each with three artificial neurons or **nodes**. You can also see many connect among these nodes and this time a **weight** is shown associated with each connection. A low weight will de-emphasise a signal, and a high weight will amplify it.







The figures above can be expressed by matrix

$$\mathbf{X} = \mathbf{W} \cdot \mathbf{I}$$

$\mathbf{W}$ is the matrix of weights, $\mathbf{I}$ is the matrix of inputs, and $\mathbf{X}$ is the resultant matrix of combined moderated signals into layer 2.

$$\mathbf{O} = sigmoid(\mathbf{X})$$

$\mathbf{O}$ contains all the outputs from the final layer of the neural network.