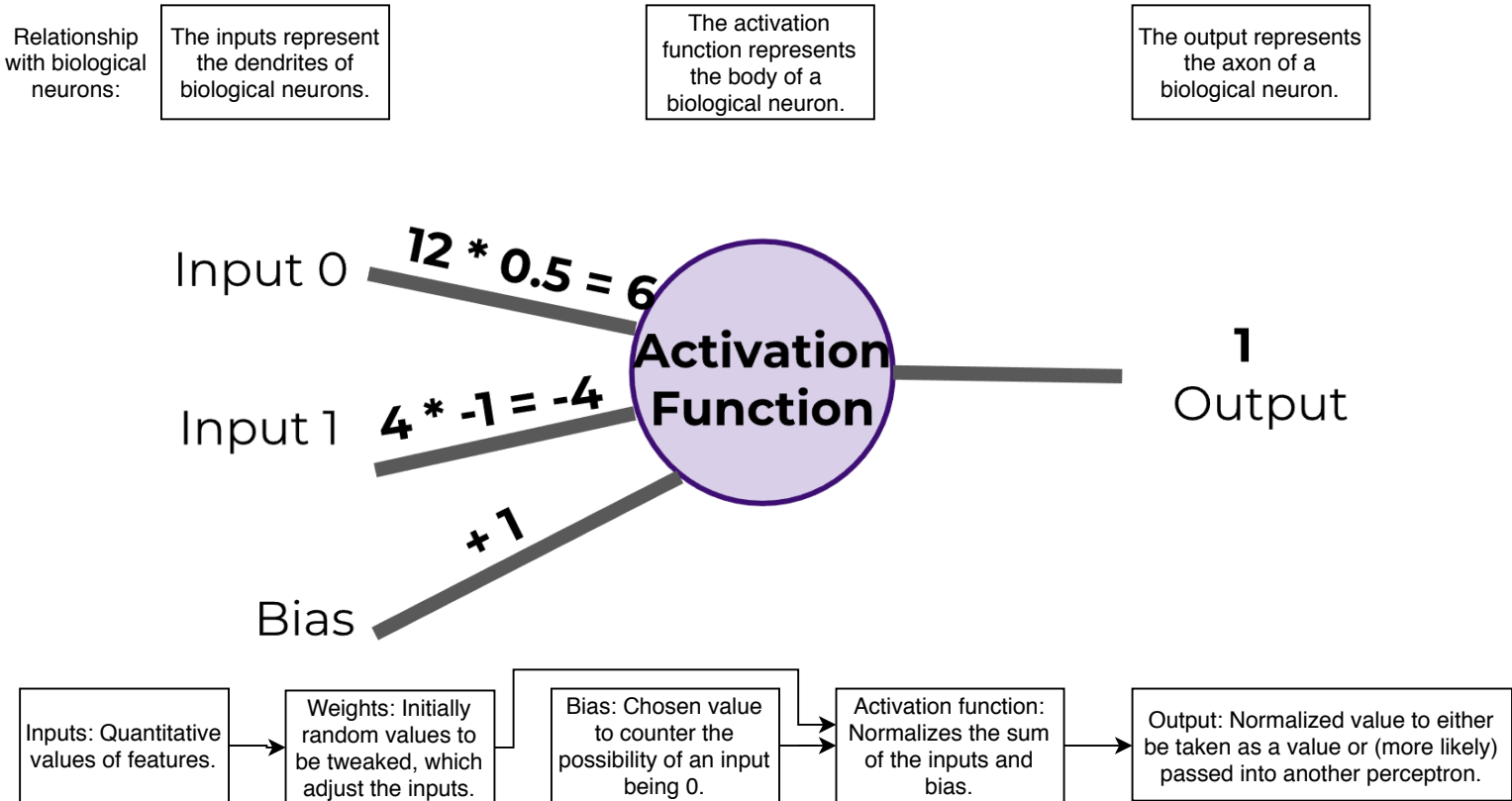


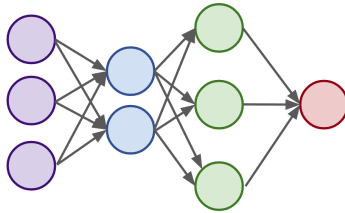
# Perceptrons - Artificial model of biological neurons!



Mathematically, this is represented as:

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

Perceptrons are the building blocks of neural networks:



Neural networks have three types of layer (columns in the diagram):

1. Input layer: Real values from the data.
2. Hidden layers: Layers in between input and output layers (3 or more is called a "deep network").
3. Output layer: Final prediction of the output.

## Types of activation functions

There are many different activation functions. Depending on the model and the task, one activation function may be better than the other.

The four main activation functions are: ReLu, Tanh, Sigmoid, Linear. ReLu and Tanh tend to have the best performance out of these four.

Many others exist, including more state-of-the-art ones.

## Cost Functions

Cost functions are used to measure how far off a model is from its expected output.

Different cost functions include: Quadratic Cost (slower learning speed), Cross Entropy (faster learning speed).

Cross Entropy has faster learning speed as it generates bigger distances between the predicted and true values. Think of cost functions as distance metrics between vectors.

## Gradient Descent and Backpropagation

Gradient Descent is an optimization algorithm for finding the minimum of a function (think of a ball rolling into a trough). It works by taking steps proportional to the negative of the gradient.

Of course, there are many other optimization algorithms, but Gradient Descent is the most popular.

Optimization algorithms enable us to figure out the best parameters (ie. weights) for minimizing our cost function.

Backpropagation is used to calculate the error contribution of each neuron after a batch of data has been processed. It works by calculating the error at the output of the final perceptron, then distributing the error back to the inputs of the perceptron, and repeating this process throughout the network. It relies heavily on the chain rule to do this.

Backpropagation requires labelled data (ie. supervised learning), otherwise it wouldn't be able to calculate errors.

See 3Blue1Brown's video series for further details.