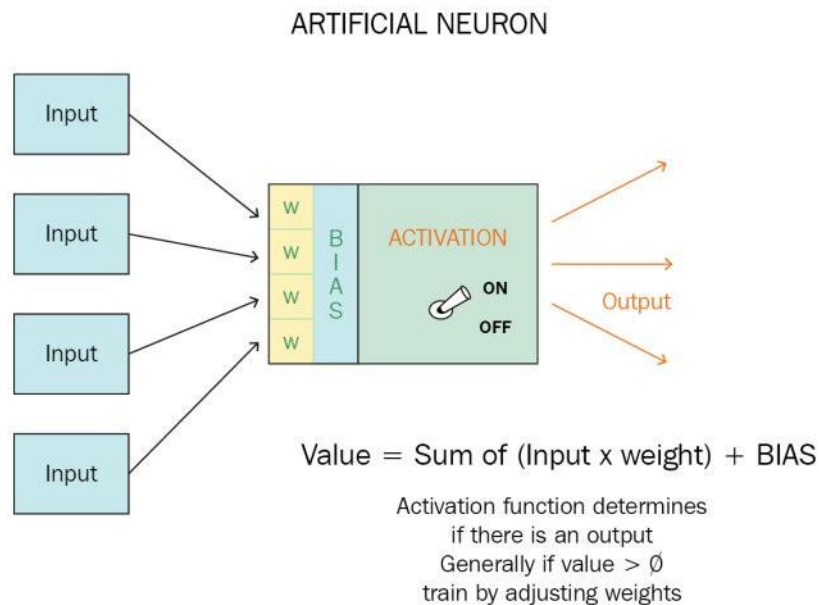## Artificial neurons

Now, let's get into the guts of a neural network. What is a neuron? And how do we make a network out of them?
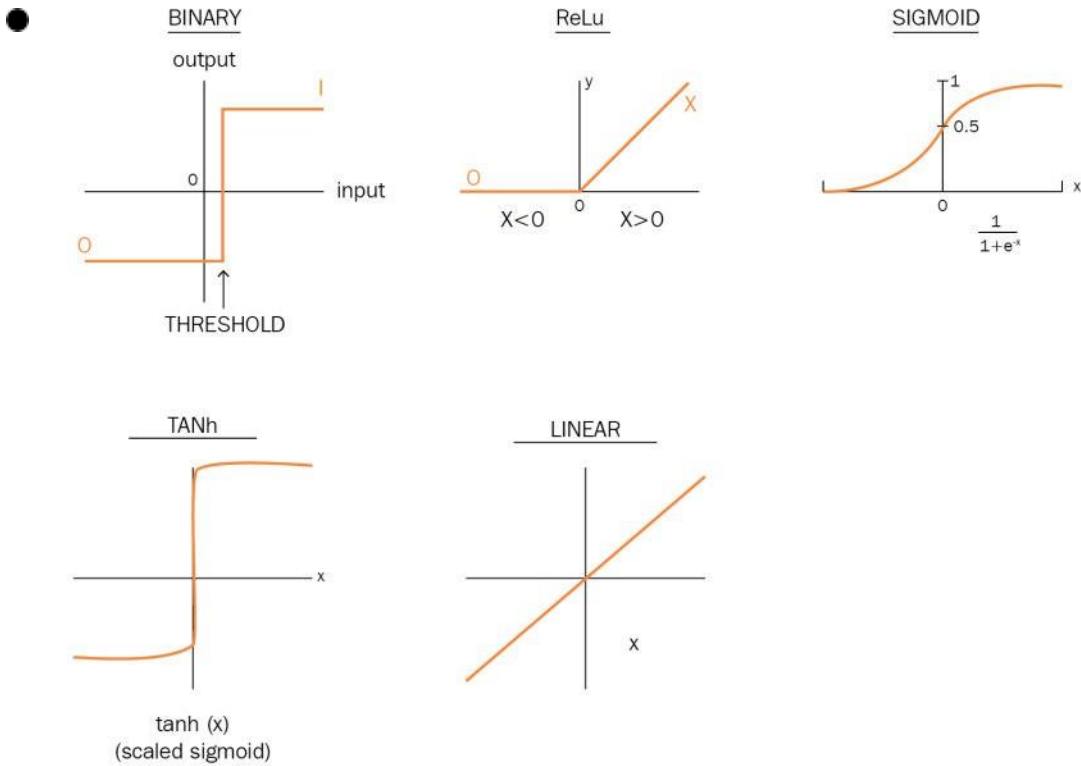
An artificial neuron is a software construction that approximates the workings of the neurons in your brain. If you can remember what you learned in biology, a biological or natural neuron has inputs, or dendrites, that connect it to other neurons or to sensor inputs. All of the inputs come to a central body, and then leave via the axion, or connection, to other neurons via other dendrites. The connection between neurons is called a synapse, which is a tiny gap that the signal from the nerve has to jump. A neuron takes inputs, processes them, and activates, or sends an output, after some level or threshold is reached:

ARTIFICIAL NEURON

Value = Sum of (Input x weight) + BIAS

Activation function determines
if there is an output
Generally if value > 0
train by adjusting weights

An artificial neuron is a very, very simplified version of this natural neuron. It has a number of inputs, a set of weights, a bias, an activation function, and then some number of outputs to other neurons or as a result of the network. The following section describes each component in detail:

- **Inputs**: A number or value received from other neurons or as an input to the network. In our image processing example, these are pixels. This number can be a float or an integer – but it must be just one number.

- **Weight**: To determine the value of a neuron, we sum up all of the inputs. As the neural network is trained, the weights are adjusted on each input that favors some inputs over others. We multiply the input by the weight and then total the inputs.

- **Bias**: A number added to the sum of the weights. Bias prevents the neuron from getting "stuck" on zero and improves training. This is usually a small number. **Activation function**: This

- determines the output of the neuron based on the

- inputs. The most common types are the **Rectifier Linear Unit**, or **ReLU** (if the value of the neuron is less than zero, the output is zero, otherwise, the output is the value) or the **SIGMOID** function (Sigmoid means "s-shaped"), which is a log function. The activation function propagates information across the network:

BINARY

output

I

0

0

input

O

↑
THRESHOLD

ReLu

y

X

O

0

X<0   X>0

SIGMOID

1

0.5

0

x

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

TANh

x

tanh (x)
(scaled sigmoid)

LINEAR

x

- **Outputs**: Each layer in the sequential neural network is connected to the next layer. Some layers are fully connected – with each neuron in the first layer connected to each neuron in the second layer. Others are sparsely connected. There is a common process in neural network training called **dropout**, where we randomly remove connections. This forces the network to have multiple paths for each bit of information it learns, which strengthens the network.

- **Max Pooling of outputs:** We will use a special type of output connection called **Max Pooling**, where groups of neurons corresponding to regions in our image – say a 5 x 5 block of pixels – go to one neuron in the next level. The maxpool neuron only takes the largest value from each of the 25 input neurons. This has the effect of downsampling the image (making it smaller). This allows the network to associate small features (such as the wheels in a Hot Wheels™ car) with larger features, such as the hood, or windshield, to identify a toy car:

Max Pooling

80x80

| 25 | 15 |
| 12 | 14 |

40x40

| 25 |

take maximum
Value