important if running on Colab

```
#If the cv2 module is not found, run that:
#!pip install opencv-python
```

switch under runtime the "Change runtime type" to a GPU support. This will speed up you calculations immensely

```
import sys,os
import numpy as np                          #main package for scientific computing with Python.
import pandas as pd                         #main data handling package
from matplotlib import image                #plot graphs in Python.
import matplotlib.pyplot as plt             #lazy plotting
import cv2                                  #image opening
```

Building and training a neural network

In this notebook, we are going to see and build the basic components of neural networks. We will first build from scratch a two layer neural network that we will then train to separate images from birds and images from dogs. The we will use Torch with a similar network and finally use Keras to build a sequential neural network with some advanced corrections

There are six components to artificial neurons. From left (input) to right (output). If we only consider numerical values these are:

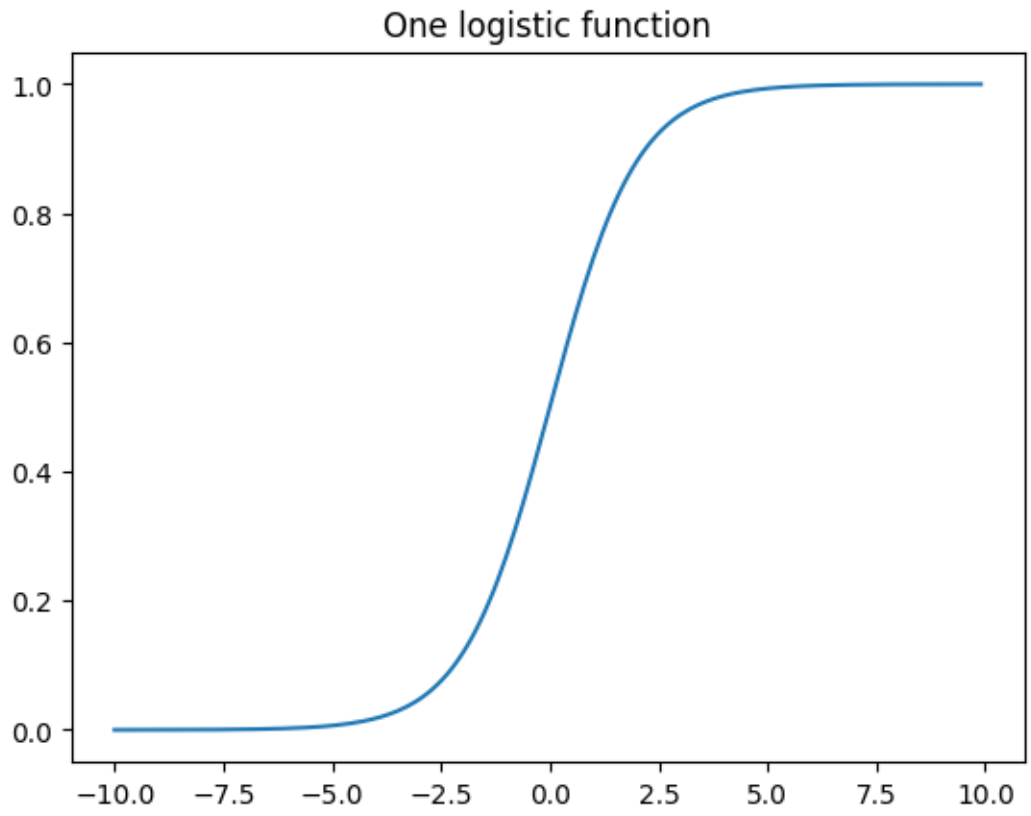
1. Input nodes. Each input node is a real number
2. Connections with weight. Each connection that departs from the input node has a weight, which is also a real number.
3. Calculate a weighted sum: $y = \sum_{i=1}^{D} w_i * x_i$
4. Feed this sum into a transfer or activation function. this can be idendity, but normally this is a treshold function. e.g. $\begin{cases} = 0 \text{ if } x \leq 0.5 \\ = 1 \text{ if } x > 0.5 \end{cases}$ Which however is not smooth, so alternatively a continous (sigmoid) function such as the logistic function is an alternative (see below)

5. output node of this chain
6. A perceptron is sometimes added to this, often this is called a bias, Which is an input node with a fixed value

With the percepton the treshold is easier defined, meaning that the transfer function can be switching at zero. (sometimes called ReLU Rectifying Linear Unit) There are other variants, but we will talk about them in the Keras section.

```
x=np.arange(-10,10,0.1);y=np.exp(x)/(1 + np.exp(x))
fig,ax=plt.subplots();plt.plot(x,y);ax.set_title('One logistic function')
```

```
Text(0.5, 1.0, 'One logistic function')
```



Data Loading

Now, we load the data that we will need for the NN