Tensorflow + Numpy = Neural network

use different data type

eg. dataset

temperature (Celsius)	duration (minutes)	Good coffee? (1/0)
200.0	17.0	1
425.0	18.5	0
...	...	...

syntax :-

```
x = np.array([[200, 17]])
```

why are double brackets being used?

single brackets

↳ `x = np.array([200, 17])`

↓

stores it as a list (1D structure)
no row or column

For tensorflow, we use 2-D structure - like a matrix, thus the double brackets.

→ For row matrix : `x = np.array([[200], [17]])`

$$\begin{bmatrix} 200 \\ 17 \end{bmatrix}$$

→ For column matrix : `x = np.array([[200, 17]])`

↳ `[200, 17]`

eg.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \Rightarrow x = np.array([[1,2,3],[4,5,6]])$$

2×3 (2D i.e. row (1) and column (1) which makes it 2(1+1))
row column

Q. What is the data type tensor?

```
x = np.array([[200.0, 17.0]])
layer_1 = Dense(units=3, activation='sigmoid')
a1 = layer_1(x)
```

[[0.2 0.7 0.8]] → 1x3 matrix

if we print a_1 (print(a_1)) we get this:-

tf.Tensor([[0.2, 0.7, 0.3]], shape(1,3), dtype=float32)

We can convert a tensor to numpy as well:-

a_1.numpy() ⟶ converts tensor data type to numpy.

Tensorflow converts all objects to tensor in order to carry out processes more efficiently.

It is generally advised to use numpy data structure for loading and manipulating data.