Today we will build a simple neural network from scratch in python using only the numpy library. We will follow the instructions from the following link:
https://iamtrask.github.io/2015/07/12/basic-python-network/
(You should try on your own first. If you find difficulty with the implementation – follow the link. Be aware – the code in the website is written in python 2, and not python 3!)

We will first build a neural network with only one input layer (consisting of three neurons) and one output layer.

1. Construct a function returning a sigmoid function. (Reminder: $s(x) = \frac{1}{1+e^{-x}}$)
2. Construct a function returning the derivative of a sigmoid function.
   (Reminder: $\frac{ds(x)}{dx} = s(x)(1 - s(x))$)
3. Build an array of three weights (3x1 array – think why these dimensions!) and initialize their value randomly. (It is good practice to use weights with mean = 0)
4. Create a loop, iterating 1000 times (equal to the desired number of learning steps).
   For each iteration, calculate the error between the network prediction and the real value of y.
   Multiply that error with the sigmoid derivative - use the dot product of this number with the input layer in order to update your weights for the next iteration.
5. Run this network, using the following input and output data sets:

Input dataset (4x3 matrix):

```
[  [0,0,1],
[0,1,1],
[1,0,1],
[1,1,1] ]
```

Output dataset (4x1 matrix):

```
[[0],
[0],
[1],
[1]]
```

6. For a slightly harder problem: We will add another layer to the network - a hidden layer. The hidden layer will also have three neurons in it. The output layer will still have only one neuron. When using a “for loop” in order to update the weights of the network, make sure to update the weights of both layers.

7. Run the new network on these new input and output data sets:

Input dataset (4x3 matrix):

```
[  [0,0,1],
[0,1,1],
[1,0,1],
[1,1,1]  ]
```

Output dataset(4x1 matrtix):

```
[[0],
[1],
[1],
[0]]
```

Good Luck!