Exercise 2 – April 28th
Step by step back-propagation for a simple neural network

We come back to the Half-Moons dataset and show how to do back-propagation step-by-step. The associated Python code is HalfMoonsBackProp.py.

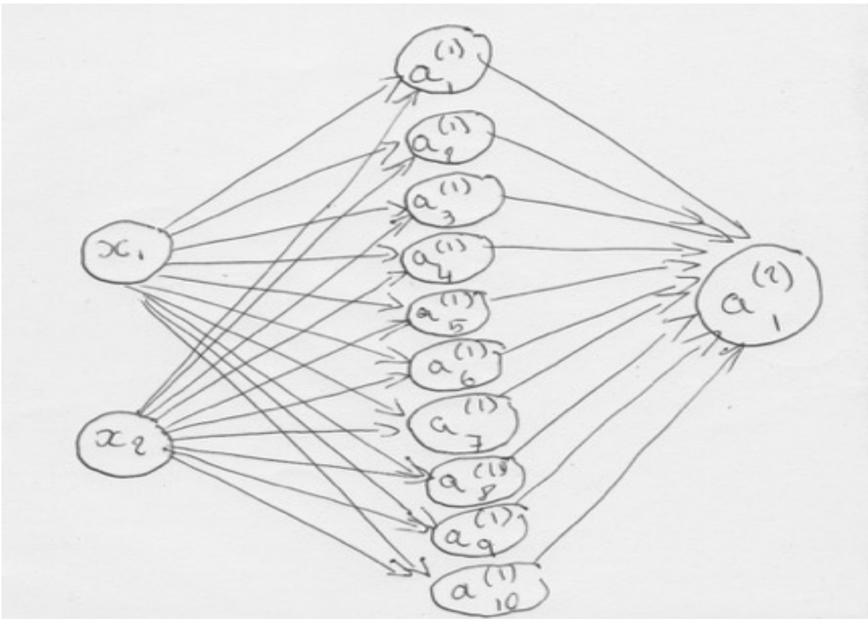
The input to the network is the two coordinates of a point $X = (x_1, x_2)$ and the output is its predicted class. We use a simple neural network with ten neurons in the hidden layer.

We have, in vectorial form:

$$z^{(1)} = W_1 X + b_1$$
$$a^{(1)} = \sigma\left(z^{(1)}\right)$$

$$z^{(2)} = W_2 a^{(1)} + b_2$$
$$a^{(2)} = \sigma\left(z^{(2)}\right)$$

Where σ Is the logistic function: $\sigma(z) = \frac{1}{1+e^{-z}}$.



1. How many weights and bias terms have to be trained?

2. If y is the class (0 or 1) associated with a data point, the cross-entropy L for this point is:

$$L = -\left(y \log a_1^{(2)} + (1-y)\log\left(1 - a_1^{(2)}\right)\right)$$

   Show that the derivative $\frac{dL}{da_1^{(2)}}$ of this loss function is:

$$\frac{dL}{da_1^{(2)}} = \frac{a_1^{(2)} - y}{a_1^{(2)}\left(1 - a_1^{(2)}\right)}$$

3. Show that:

$$\frac{da_1^{(2)}}{dz_1^{(2)}} = a_1^{(2)}\left(1 - a_1^{(2)}\right)$$

4. Show that

$$\frac{\partial L}{\partial W_2} = \left(a_1^{(2)} - y\right)a^{(1)}$$

5. How can we calculate $\frac{\partial L}{\partial X}$ and what does the plot of $\frac{\partial L}{\partial X}$ for each training point illustrate?