Exercise 1 – April 28th
Experimenting with a simple Neural Network

We come back to the playground exercise already presented this morning:

https://playground.tensorflow.org

Unless otherwise indicated, we work with the following hyper-parameters:
Learning rate =0.03, Batch Size=10, Noise=0, Training Data=70%.
Since there is a total of 500 data points, there are 350 (or 70%) training data and 150 test data.
We examine the behaviour of the networks until about 3500 epochs.

It is important to note that there will be differences between different runs obtained with the same hyperparameters, because the initial values of the neural network parameters (weights and bias terms) are sampled randomly. The dataset itself may also change if you click the *Regenerate* Button at the bottom left of the screen.

We work on a classification problem on the Spiral dataset. The yellow dots have a value of -1 and the blue dots have a value of +1.

1. Use the sigmoid activation function and no hidden layer. What do you observe? How do you interpret it?

2. Now put just one hidden layer, a sigmoid activation function, and make the number of neurons in the hidden layer equal to 4. Then make it equal to the maximum offered by the application, that is 8. What is the number of trained parameters and what do you observe in each case?

3. Now do the same as in question 3, but using the ReLU activation function. What do you observe?

4. Now use two hidden layers each with four neurons and the ReLU activation function. How many trained parameters do you have? What do you observe?

6 Now try two hidden layers with 8 neurons each and a ReLU activation function. How many trained parameters do you have? What do you observe?

7. Now try three hidden layers with 8 neurons each. What happens if you compare sigmoid and ReLU?

8. Now keep the three layers with 8 neurons and change the batch size to 30, and then 1. What happens if you use ReLU?

9. Now that we have obtained a good model see what happens if you introduce all the 7 input variables : $X_1, X_2, X_1^2, X_2^2, X_1X_2, \sin X_1, \sin X_2$.

10. Now assume that the data are affected by noise. Just use the maximum noise value of 50. The application is not very clear about the mechanism by which the noise affects the initial data classes. This does not matter, just assume that, after introducing this noise value of 50, you are now dealing with new binary class values at each point. What do you observe?