General rules for all exercises

- *One member of the group submits the compiled Jupyter notebook via moodle.*

- *The name of the notebook should be*
  group25XX_exerciseY.ipynb
  where XX is the code of the group and Y is the number of the exercise.

- *The list of students (Name, Family Name, matriculation number) must appear at the beginning of the notebook.*

- *The notebook should be structured to be easily understood, with some commented figures but without long and unnecessary output.*

LCPB 24-25 exercise 1 (Gradient descent & Deep Neural Networks, DNNs)

Analyze the labeled data introduced in the lesson on DNNs. Consider samples with L=8 values. Use a network with 3 layers, each with 20 units.

1. Random search of best hyperparameters

Implement a "random search," with the **keras_tuner** package; see
https://keras.io/guides/keras_tuner/getting_started/
to improve the performace of the DNN by a random search in the hyperparameters' space of the values that give the best accuracy of the validation data set.
Hyperparameters that could be considered for tuning include:

- Minimizer: ADAM, RMSprop, Nesterov, ...
- Learning rate: 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}
- Nonlinear activations: sigmoid, ReLU, ELU, ...
- dropout values: 0, 0.1, 0.2.

Which conclusions can you draw by comparing the accuracy in the various cases? To answer this question, consider performing a cross validation for results with similar accuracy.

2. Data augmentation

Take a DNN with good parameters from point 1 and study how its performance changes when the number N of samples is

a) reduced
b) *augmented*

Point 2.b means taking the original N samples, splitting them in training and validation, and "augmenting" the training samples by generating artificial ones similar to the real ones. For example, one could add some random, small fluctuations to a training sample to generate a new one.