Aprendizagem Automática Avançada

Project Nº 2

Muli-Layer Perceptron Neural Networks

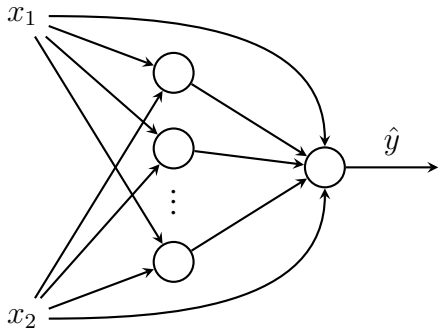
2º Semester, 2021/2022

Work Objectives:

Implement, train and test Multi-Layer Perceptron (MLP) neural networks. The report and code implemented must be in a Jupyter Notebook with the name `A#####A#####_T2.ipynb` (`A#####` corresponds to the student number).

.

1. Without resorting to the TensorFlow/Keras library, implement the following three-layer MLP network, for the task of binary classification of two-dimensional data (XOR problem):



The following points should be taken in account:

- In the diagram above, each circle represents a perceptron, and as such performs a sum of the inputs multiplied by the respective weights plus a polarization term, followed by an activation function (see slide 4/21 of "An Introduction to Artificial Neural Networks").
- Use the hyperbolic tangent for the activation functions of this network.
- The number of hidden units must be a parameter that can be changed in the code.
- Use data from file *pickle* `xorData.p`. This file contains a dictionary with 1000 two-dimensional points related to the XOR problem (key `data`) and the respective classes (key `target` - array of 0s and 1s).

(a) Implement a network training algorithm. It should be possible to choose the value of the adaptation step, η, the moment term, α, as well as the number of adaptation iterations.

(b) Train a network with one hidden unit. Run the training several times in order to choose the values that you deem appropriate for η, α and for the number of iterations. Make a graph of the evolution of the error throughout the training with the chosen

parameters. Visualize the training set errors. Plot the network output function, using the command `plot_wireframe`, for a grid of 50×50 in the range $[-1, 2]$.

(c) Repeat point 1.b) for a network with 10 hidden units.

(d) Repeat point 1.b) for a network with 50 hidden units.

2. The objective is to implement, train and test several MLP networks with the databaseCIFAR-10.

   (a) Train and evaluate an MLP network with 10 hidden layers and 100 units per layer. Use the function and ReLU for the activation functions of the hidden layers, the function *softmax* for the output layer, and the Nadam optimization.

   (b) Train an MLP network of your choice with the objective of reaching a probability of success above 50% in the test set. Bear in mind the following points:

   - Check if pre-processing the data is beneficial (eg: *standard scaler*, PCA).
   - Use a validation set to visualize the evolution of the error during the training process.
   - Try using different parameters/methods in your training, like *dropout* layers, different activation functions, etc.

Report Preparation: You should take into account several aspects inherent to a supervised learning project such as the training/test methodologies used, evaluation metrics used, etc. Compare the results of item 2 with those obtained in the 1st project.
The Jupyter Notebook should be properly commented in order to clearly perceive the various stages of the work developed and the results obtained.