MACHINE LEARNING

Assignment 2

Submission deadline: 1st May

12 points

DEVELOPMENT AND SUBMISSION RULES

For this work, as for the others, it is mandatory to present a written report (in PDF) with your evaluations and decisions taken to develop each of the sections. Include the generated graphics in the report. You should also include an assessment of the quality of the achieved results. **Without this report it is considered that the work has NOT been submitted.**

Rules to follow (each non-compliance will imply the loss of two points):

- The code of each exercise/section of the practice must be structured, including the functions that have been defined.
- All the numerical or graphical results will be shown on the screen, stopping the execution after each section. The code MUST NOT write anything to disk.
- The path used to read any auxiliary data file must always be “data / filename”. That is, the code is expected to read from a directory called "data", located within the directory where the practice is developed and executed.
- The code is accepted if it can be executed from start to finish without errors.
- The use of options at the entrances is NOT ACCEPTABLE. Set at the beginning the default parameters that you consider to be optimal.
- The code must be compulsorily commented explaining what the different sections and / or blocks do.
- Include stopping points to display images or data by console.
- All files (* .py, * .pdf) are delivered together in a single zip file, without any directory containing them.
- GIVE ONLY THE SOURCE CODE, NEVER SUBMIT THE DATA.
- Submission: Upload the zip to PRADO

1. Exercise on the complexity of H and noise (5 points)

In this exercise we must learn the difficulty that the appearance of noise in the labels introduces when choosing the most suitable class of functions. We will use three functions included in the file *template_trabajo2.py*:

- $simula\_unif(N, dim, range)$, which computes a list of N vectors of dimension dim. Each vector contains dim uniform random numbers in the $range$ interval.
- $simula\_gaus(N, dim, sigma)$, which computes a list of length N of vectors of dimension dim, where each position of the vector contains a random number drawn from a Gaussian distribution of mean 0 and variance given, for each dimension, by the position of the $sigma$ vector.
- *simula_linea(interval)*, which randomly simulates the parameters, $v = (a, b)$ of a line, $y = ax + b$, which crosses the square $[-50, 50] \times [-50,50]$.

1. (1 point) Draw graphs with the simulated point clouds under the following conditions:
   a) Consider $N = 50$, $dim = 2$, $range = [-50, +50]$ with $simula\_unif(N, dim, range)$.
   b) Consider $N = 50$, $dim = 2$ and $sigma = [5, 7]$ with $simula\_gaus(N, dim, sigma)$.
2. We are going to assess the influence of noise in the selection of the complexity of the class of functions. With the help of the function $simula\_unif(100, 2, [-50, 50])$ we generate a sample of 2D points to which we are going to add a label using the sign of the function $f(x, y) = y - ax - b$, that is, the sign of the distance from each point to the line simulated with *simula_recta()*.
   a) (1 point) Draw a 2D graph where the points show (use colors) the result of their label. Also draw the line used for labeling. (Note that all the points are well classified with respect to the line)
   b) (0.5 points) Randomly modify 10 % of the positive labels and another 10 % of the negative labels and save the points with your new labels. Redraw the above graph. (Now there will be misclassified points with respect to the line)
   c) (2.5 points) Suppose now that the following functions define the classification boundary of the sample points instead of a line
      - $f(x, y) = (x - 10)^2 + (y - 20)^2 - 400$
      - $f(x, y) = 0{,}5(x + 10)^2 + (y - 20)^2 - 400$
      - $f(x, y) = 0{,}5(x - 10)^2 - (y + 20)^2 - 400$
      - $f(x, y) = y - 20x^2 - 5x + 3$

      Visualize the annotations/labels generated in $2b$ together with each of the graphs for each function. Compare the positive and negative regions of these new functions with those obtained in the case of the line. Discuss whether these more complex functions are better classifiers than the linear function. Look at the graphs and discuss the consequences on the influence of the label modification process on the learning process. Explain the rationale behind.

2. Linear Models (7 points)

a) **(3 points) Perceptron learning algorithm:** Implement the function

$$ajusta\_PLA(data, label, max\_iter, vini)$$

that calculates the hyperplane solution to a binary classification problem using the PLA algorithm. The input *data* is an array where each item with its label is represented by a row of the matrix, *label* is the vector of labels (each label is a value +1 or −1), *max_iter* is the maximum number of iterations allowed and *vini* the initial value of the vector. The function returns the coefficients of the hyperplane.

1) Execute the PLA algorithm with the simulated data in the $2a$ part of section. 1. Initialize the algorithm with: a) the zero vector and, b) with vectors of random numbers at $[0, 1]$ (10 times). Record the average number of iterations required in both to converge. Evaluate the result by relating the starting point to the number of iterations.
2) Do the same as before using now the data from block $2b$ of section.1. Do you observe any different behavior? If so, state which one and the reasons for this to happen.

b) **(4 points) Logistic Regression**: In this exercise we will create our own objective function f (a probability in this case) and our data set $\mathcal{D}$ to see how logistic regression works. We will assume for simplicity that f is a probability with values 0/1 and therefore that the label y is a deterministic function of $\mathbf{x}$.

Let consider $d = 2$ for the data to be displayable, and let $\mathcal{X} = [0, 2] \times [0, 2]$ with uniform probability of choosing each $\mathbf{x} \in \mathcal{X}$. Choose a line in the plane that passes through $\mathcal{X}$ as the border between $f(\mathbf{x}) = 1$ (where y takes value +1) and $f(\mathbf{x}) = 0$ (where y takes value −1). To do so, select two random points of $\mathcal{X}$ and calculate the line that passes through both.

EXPERIMENT: Select $N = 100$ random points $\{\mathbf{x}_n\}$ from $\mathcal{X}$ and evaluate the responses $\{y_n\}$ of all of them against the chosen boundary. Run Logistic Regression (see conditions below) to find the solution function g and evaluate the error E_{out} using a new large data sample (> 999). Repeat the experiment 100 times, and

- Calculate the value of E_{out} for size N = 100.
- Calculate how many times it takes to converge on average RL for N = 100 under the conditions set for its implementation.

Implement Logistic Regression (RL) with Stochastic Gradient Descent (SGD) under the following conditions:

- Initialize the vector of weights to 0.
- Stop the algorithm when $||\mathbf{w}^{(t-1)} - \mathbf{w}^{(t)}|| < 0{,}01$, where $\mathbf{w}^{(t)}$ denotes the vector of weights at the end of epoch t. An epoch is a complete pass through the N data.
- Apply a random permutation of $\{1, 2, ..N\}$ to the indices of the data, before using them in each epoch of the algorithm.
- Use a learning rate $\eta = 0{,}01$.

3. BONUS

The BONUS will only be taken into account if at least 75 % of the points of the mandatory part have been obtained.

(1.5 points) Digit Classification. Consider the handwritten digits dataset, and select the samples corresponding to digits 4 and 8. Use the training and test files provided. Extract the characteristics of **average intensity** and **symmetry** in the manner indicated in assignment 1.

1. Set up a binary classification problem that considers the training set as input data to learn the g function.
2. Use a Linear Regression model and apply PLA-Pocket as an improvement. Answer the following questions.
   a) Generate separate graphs (in color) of the training and test data together with the estimated function.
   b) Calculate E_{in} and E_{test} (error over test data).
   c) Get bounds on the true value of E_{out}. Two bounds can be calculated, one based on E_{in} and the other based on E_{test}. Use a tolerance $\delta = 0{,}05$. What bound is better?