MACHINE LEARNING

Assignment-1

Deadline: April 12th

Submission site: PRADO

12 Points

For this work, as for the others, it is mandatory to present a written report 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.

Rules to follow:

- The code of each exercise/section of the practice must be structured, including the functions that have been defined.
- All the numerical or graphic 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 path used to read any auxiliary data file must always be "data/filename." The code is expected to read from a directory called "data," located within the directory where the code is run.
- 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.
- Submission: Upload the zip to PRADO

UPLOAD ONLY THE SOURCE CODE, NEVER THE DATA OR IMAGES THAT HAVE BEEN PROVIDED.

EXERCISES

Gradient Descent (GD) (6.5 puntos).

1. (1 point) Code the GD algorithm.

2. (2 points) Consider the function $E(u,v) = (u^3e^{(v-2)} - 2v^2e^{-u})^2$. Use GD to find the minimun of this function. Set as initial point $(u,v) = (1,1)$ and use a learnig rate $\eta = 0.1$.

A) Compute analytically the gradient of $E(u,v)$, and show the expression.

B) How many iterations does it take for GD to obtain for the first time a value of $E(u,v)$ less than 10^{-14}. (Use 64-bit floats)

C) In what (u,v) coordinates a value equal to or less than 10^{-14} was reached for the first time in the previous point?

3.(2 points) Consider the function $f(x,y) = (x+2)^2+2(y-2)^2+2\sin(2\pi x)\sin(2\pi y)$

A) Use GD to minimize this function. Use as starting point ($x_0 = -1$, $y_0 = 1$), (learning rate $\eta = 0.01$ and a maximum of 50 iterations). Generate a figure showing how the value of the function (y-axis) decreases with the iterations (x-axis). Repeat the experiment using $\eta = 0.1$, and comment on the differences and their dependence on η

B) Obtain the minimum, and the values of (x, y) where this minimum is reached, when the starting point is set at: (−0.5, −0.5), (1,1), (2,1, −2.1), (- 3, 3), (- 2, 2). Generate a table with the obtained values. Discuss the dependence on the starting point.

4.(1.5 points) What would be your conclusion about the true difficulty of finding the global minimum of an arbitrary function using GD?

2. Linear Regression (5.5 points)

This exercise fits regression models to feature vectors extracted from handwritten digit images. In particular, two specific features are extracted: a) the average value of the gray level; and b) the symmetry of the number with respect to its vertical axis. Only the images of numbers 1 and 5 will be selected for this exercise.

1. (2.5 points) Estimate a linear regression model from the data provided by the feature vectors (Average intensity, Symmetry) using both the pseudo-inverse algorithm and the Stochastic Gradient Descent (SGD). The labels will be {−1, 1}, one for each feature vector of each number. Draw the solutions obtained together with the data used in the fitting. Assess the goodness of the result using E_{in} and E_{out} (for E_{out} calculate the predictions using the data from the test file). (use Regress_Lin(data, label) as call for function (optional)).

2. (3 points) In this section we explore how the E_{in} and E_{out} errors are transformed when we increase the complexity of the linear model used. Now, we use the function simula_unif(N, 2, size) that returns N 2D coordinates of uniformly sampled points within the square defined by [−size, size] × [−size, size]

EXPERIMENT:

a) Generate a training sample of N = 1000 points (x_1,x_2) in the square X = [−1,1] × [−1,1]. Draw the 2D map displaying the points. (see help function)

b) Let's consider the function $f(x_1,x_2) = \text{sign}((x_1-0{,}2)^2 + x_2^2-0{,}6)$ that we will use to assign a label to each point of the previous sample. We introduce noise on the labels, randomly changing the sign of 10% of them. Draw the obtained labels map.

c) Using $(1, x_1, x_2)$ as feature vector, fit a linear regression model to the generated dataset and estimate the weights w. Estimate the fitting error of E_{in} using Stochastic Gradient Descent (SGD).

d) Run the entire experiment defined by (a) - (c) 1000 times (1000 different samples have to be generated) and:

- Compute the mean value of E_{in} errors in all 1000 samples.
- Generate 1000 new points on each iteration, and compute with them the value of E_{out}. Compute the mean value of E_{out} in all iterations.

e) Assess how good you consider the fit with this linear model is according to the mean values obtained for E_{in} and E_{out}. Repeat the same previous experiment but using non-linear characteristics. Now, we will use the following feature vector: $\Phi 2\ (x) = (1, x_1, x_2, x_1x_2, x_1^2, x_2^2)$. Fit the new linear regression model and calculate the new vector of weights w. Calculate the average errors of E_{in} and E_{out}. Which model do you consider the most appropriate according to the average errors for E_{in} and E_{out}?

BONUS:

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

1. (2 points) **Newton's method**: Implement Newton's minimization algorithm and apply it to the function f(x, y) given in exercise.1.3. Carry out the same experiments using the same starting points.
    a. Generate a graph of how the value of the function decreases with the iterations.
    b. Draw conclusions about the behaviour of the algorithms by comparing the decrease curve of the function calculated in the previous point and the corresponding one obtained with GD.