Machine Learning per il Calcolo Scientifico

Second laboratory exercises

General guide

For each laboratory, an incomplete Python notebook will be provided with exercises (steps) that must be completed in the given order (some of the exercises will be needed in future laboratories). In step zero, all the Python packages that are needed in order to complete the notebook are listed. This PDF includes the text for the exercises and the expected outcomes for each step. While following the notebook is recommended, you are also welcome to attempt the exercises without using it.

Step Zero

Here are the required Python (`https://www.python.org/`) packages for this laboratory:

- PyTorch (`https://pytorch.org/`)
- Numpy (`https://numpy.org/`)
- Matplotlib (`https://matplotlib.org/`)
- plotly (`https://plotly.com/`)
- ipywidgets (`https://ipywidgets.readthedocs.io/en/stable/`)

First Step: Define the gradient

a. Using torch.autograd.grad complete the function grad in lab2.py (Will be used in other labs)
b. Use the function grad to evaluate the gradient of the function $x^2 + y^2$ (Example1.ipynb).
c. Confront the gradient obtained with torch.autograd with the exact gradient on 1000 points.

Second step: Implement the Gradient descent method

Algorith:

$$x_{k+1} = x_k - \eta \nabla f(x_k)$$

a. Complete the function GD(lab2.py).
b. In each Example run the GD method.
c. Visualize the results using plot_3D_GD.
d. Explore different intial points, learning rate and number of epochs using the the code provided (interact).

Third step: Implement the Momentum method

Algorith:

$$\begin{cases} v_{k+1} = \beta v_k - \alpha \nabla f(x_k), \\ x_{k+1} = x_k + v_{k+1}. \end{cases}$$

a. Complete the function momentum (lab2.py).
b. In each Example run the momentum method.
c. Visualize the results using plot_3D_GD.
d. Explore different intial points, learning rate and number of epochs using the the code provided (interact).

Fourth step: Implement the Neserov Momentum method

Algorith:

$$\begin{cases} v_{k+1} = \beta v_k - \alpha \nabla f(x_k + \beta v_k), \\ x_{k+1} = x_k + v_{k+1}. \end{cases}$$

a. Complete the function Nesterov (lab2.py).
b. In each Example run the Nesterov method.
c. Visualize the results using plot_3D_GD.
d. Explore different intial points, learning rate and number of epochs using the the code provided (interact).

Fifth step: Confront the results obtained

a. Using the function err, confront the convergence of the methods for each example.

Sixth step: Scheduler of the learning rate

a. In Example4.ipynb using the interactive 2D plot change the number of epochs what do you observe?
b. Implement the Step learning rate function.
c. Use the step_LR function with step = 5 and gamma = 0.55.
d. Confront the result using the err function with and without the scheduler.