Assignment Set 5 - Machine Learning

Instructor: Dr Enrique Hortal

December 1, 2023

Individual assignment

These assignments should be handed in and defended (if required) individually.

Handing in

Upload your documented code for Linear RegresCanva the Canvas assignment in a single zip archive. In addition, upload a single report in the form of a PDF. We really mean PDF - no other format. The report should **NOT** be part of the zip archive so that we could provide you with feedback on that report directly (if needed) and automatically checked for plagiarism. Submissions by email or other types of archives are not accepted. Thank you for your understanding.

1 Graded Assignment: Linear Regression

In this lab, you will implement the cost function and the gradient descent algorithms for Linear Regression. The implementation must be in Python. To do so, a start code is provided. See the file “LinearRegression.py” for a basis to start with. The pieces of code that you are expected to update are marked with ”HERE YOU…”.

After the implementation of the required modifications, **perform and document experiments** to answer the following questions as part of your report:

1. What happens if the learning rate is too low?
   Hint: check different values for the learning grade and analyse their impact
2. What happens if the learning rate is too high?
   Hint: check different values for the learning grade and analyse their impact
3. Can Linear Regression really find the absolute global minimum? Explain why/why not.
   Hint: you can explain the theoretical approach and prove/disprove it empirically
4. What effect does it have if you change the initial guess for thet0 and thet1 for the gradient descent to something completely off?
   Hint: check different values for the theta values and analyse their impact
5. What happens if you are not updating thet0 and thet1 “simultaneously” as you should but you are updating both parameters in separate for-loops (see code)?
   Hint: evaluate it empirically by modifying the code accordingly
6. How many iterations of the gradient descent algorithm do you have to perform to reach the correct exact values of theta0 and theta1?
   Hint: explain how you calculated it and demonstrate it with experimental results

We really mean perform **and document** experiments. E.g. modify the learning rate, run your code, and see what happens if this parameter is very small. Just providing an answer will only give you very few points. Also, provide a plot (screenshots are valid) to document the result of your experiments. Do not forget to provide a short explanation in form of one or two sentences so we know what is the cause for what you show in the plot. You can also try to generate linear data with some noise and check how it affects the training process and the results.

2 Clarification dot product

If you face problems when implementing the Gradient Descent algorithm, please revise how the dot product is implemented in NumPy here:

https://numpy.org/doc/stable/reference/generated/numpy.dot.html

Additionally, the following equations may be helpful:

$$X = \begin{bmatrix} 1 & x_1^{(1)} \\ 1 & x_1^{(2)} \end{bmatrix}$$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$$

$$\text{numpy.dot}(X, \theta) = \begin{bmatrix} \theta_0 * 1 + \theta_1 * x_1^{(1)} \\ \theta_0 * 1 + \theta_1 * x_1^{(2)} \end{bmatrix}$$

3 [NOT GRADED] Extra implementation

If you manage to implement the linear regression solution satisfactory, we highly encourage you to modify your code in order to perform classification using the diabetes dataset used in Lab 1 by applying **Logistic regression**.