Assignment 2

Due to 1.12.2019

1.Submission Instructions

1.1 Theoretical Part

a) The objective of this part is to practice the terms and concepts in the task of regression.

b) If needed, make sure to explain or present an example to any given answer to assure full understanding of your idea.

c) Submission: please submit a PDF version named "ex2-theoretical-part.pdf".

1.2 Practical Part

a) The main objective of this part is to assure you know and understand the necessary methods in *Python* to implement regression.

b) You are allowed to use standard Python packages, *Numpy* and *Matplotlib* only.

c) Use *Python 3.7*

d) Name your main code as regression.ipyn

Good Luck!

2. Questions

2.1 Theoretical Part

In the class we present two different methods to solve the regression problem. The first method is called *normal equation.* Recall the normal equation is an analytical approach to Regression,the way we can directly find out the value of our parameter w. The second method is based on *Gradient Descent.* In this method we update our parameter w, using the gradient, until we converge to the optimal w.

1. Describes the regression problem. How the training set looks like? What is the loss function that we are trying to minimized?
2. Explain what the GD algorithm does? Which type of optimal it finds?
3. Explain why solving the regression problem using SG is guarantied to converge to the global minimal?
4. Explain what the epsilon and lambda represent in the GD algorithm.
5. Suppose we plot out training set and we found that our training set is not linear. In this situation we would like to use some polynomial fitting. What should we change/add in this situation? Explain.
6. Write the bias and variance trade off the Mean-Square-error(MSE). Explain what each term represents and what is the trade-off between them.
7. Suppose we decided to solve regression problem and our training set's size is very large. Which method we would prefer to use in this scenario? (Hint: what is the complexity of a matrix inverse vs the complexity of using the gradient)

2.2 Practical Part:

In this part you will implement the two methods for *Regression* task and understand practically how different parameters effect the result.

A template code named *regression.ipyn* is attended to the exercises. In the template you will find a variable called *max_poly_deg.* This variable will affect maximum polynomial fitting to the data. You will proved with three different data sets make sure that your code support them.

1. Plot the loaded data (X vs Y).
2. Set the right *max_poly_deg* parameter for the data.
3. Using the *max_poly_deg* variable implement the regression with normal equation method.
4. Using the *max_poly_deg* variable implement the regression with GD method.

5. Implement a function named *plot_prediction* that takes the following parameters: the training set(X,Y) and a vector w. This function should plot the curve created by w and the training set on the same plot(choose different colors for curve and the training set).

6. Apply *plot_prediction* to plot the results of the two methods.