IE406: Machine Learning Lab

Assignment 2

(Date: 05/09/2021)

1. Implement polynomial regression as a special case of linear regression. First generate some data as follows.

```
import numpy as np

x = np.arange(0, 20.1, 0.1)

np.random.seed(0)

y = 1*x**5 + 3*x**4 - 100*x**3 + 8*x**2 -300*x - 1e5 + np.random.randn(len(x))*1e5
```

   Now, we want to learn a polynomial function of degree p on this dataset,

   i.e. $y = q_0 + q_1 \times x_1 + q_2 \times x_2 + ...q_p \times x_p$.

   We can use the linear regression implementations for doing so, by transforming the dataset and creating the matrix X containing columns corresponding to $x_0, x_1, x_2, ..., x_p$. Using any of your implementations above learn the regression coefficients for p = 5 and p = 4. How close are your coefficients for p = 5 to the ones used to generate the data?

2. Find minima of following functions using Gradient Descent method with learning rate 0.01 and 0.1 and different number of iterations. Try choosing a large value of learning rate and test the convergence. For $L_5(\theta)$, use the data file.
   a. $L_1(\theta) = \theta^2$
   b. $L_2(\theta) = \theta_1^2 + \theta_2^2$
   c. $L_3(\theta) = (\theta-1)^2$
   d. $L_4(\theta) = 2(\theta_1-1)^2 + 2(\theta_2-1)^2$
   e. $L_5(\theta) = \sum_{i=1}^{m} \left(y^{(i)} - \left(\theta_0 + \theta_1 . x^{(i)}\right)\right)^2$

3. Find minimum of the function $L(\theta) = \sum_{i=1}^{m} \left(y^{(i)} - \left(\theta_0 + \theta_1 . x^{(i)}\right)\right)^2$ using the Stochastic Gradient Descent method (Take the data from the data file). Choose different learning rates and number of iterations.

4. Find the minima of following functions using the Steepest Descent method.
   a. $L_1(\theta) = \theta^2$
   b. $L_2(\theta) = \theta_1^2 + \theta_2^2$
   c. $L_3(\theta) = (\theta-1)^2$
   d. $L_4(\theta) = 2(\theta_1-1)^2 + 2(\theta_2-1)^2$

e. $L_5(\theta) = \sum_{i=1}^{m} \left(y^{(i)} - \left(\theta_0 + \theta_1 . x^{(i)} \right) \right)^2$

5. The following question is to aid your understanding of gradient descent with the help of some visualization. Consider the data X and Y as given below.

$$X = \begin{bmatrix} 1 \\ 3 \\ 6 \end{bmatrix} \qquad Y = \begin{bmatrix} 6 \\ 10 \\ 16 \end{bmatrix}$$

(a) Create a contour plot in the θ_0 and θ_1 space of residual sum of squares.

(b) Create a Matplotlib animation where the plot contains two columns: the first one being the contour plot and the second one being the linear regression fit on the data. The different frames in the animation correspond to different iterations of gradient descent applied on the dataset to learn θ_0 and θ_1. For each iteration, draw the current value of q_0 and q_1 on the contour plot and also an arrow to the next q_0 and q_1 as learnt by gradient update rule. Correspondingly draw the $y = \theta_0 + \theta_1 \times x$ line on the other subplot showing the scatter plot. The overall title of the plot shows the iteration number and the residual sum of squares. You are free to use any gradient descent implementation i.e. your own or using libraries like scikit learn.

Note:

Submission Deadline: 21:00 hrs, Saturday, 11th September 2021

Strictly follow the submission guidelines given in the classroom.