CSC 603: Machine learning

Assignment 1

Q1- Consider the problem of predicting the weather temperature. We apply a polynomial regression, and our hypothesis is defined as follows:

$$h(x)_\theta = 4\theta_1 x_1^2 x_2^5 + \theta_2 x_2^3 - \theta_2\theta_3 x_2^4 + \theta_0$$

The cost function is given by:

$$J(\theta_0, \theta_1, \theta_2, \theta_3) = \frac{1}{2}(y^{(i)} - h_\theta(x^{(i)})^2$$

a. Find the following partial derivatives (show/explain your work):
   1. $\frac{\partial J}{\partial \theta_0}$=
   2. $\frac{\partial J}{\partial \theta_1}$=
   3. $\frac{\partial J}{\partial \theta_2}$=
   4. $\frac{\partial J}{\partial \theta_3}$=

b. Write the formula to calculate θ_3^{new} to minimize the cost function?

<u>Programming tasks</u>

Q2) Find one dataset that is suitable for linear regression with **multiple input variables**. (provide the source of the dataset (link)). Your code must follow the structure of code given in HW1.ipynb.

a. Split the dataset into 60% train data and 40% test data.

b. Write a program to implement linear regression using gradient decent based on **matrix design (vectorization).** The formula used for updating θ is defined as follows:

$$\theta_j := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} \qquad \text{(simultaneously update } \theta_j \text{ for all } j).$$

c. Plot the cost function against the iterations (epochs)

d. Experiment different values of learning rate $\propto$ and print the mean squared error

Q3) Find **a dataset** that is suitable for linear regression with **one** input variable.

a. Split the dataset into 70% train data and 30% test data.
b. Create a scatter plot to visualize the training data.
c. Use **sklearn** to implement the following:
   - Linear regression
   - Lasso regression
   - Ridge regression
   - Polynomial linear regression

d. Draw bar chart that compares the mean squared error between the correct outputs and the predicted outputs on testing set for linear models.
e. Create a scatter plot of testing data along with the best fit line for linear regressions models in Q3-c.

4) Normal equation (using dataset in Q2)

You learned normal equation (closed-form solution) that can be used for finding the optimal parameter values θ for linear regression given the matrix of training examples X and the corresponding response variables y. Write a Python code to implement normal equation**.** *The formula is defined as follows:*

$$\theta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

جامعة تبوك
University of Tabuk

In [2]:

```
import numpy as np
```

In [9]:

```
## your code must be implmented based on matrices and vectors.
class linear_regression:
    # X is the X_dataset
    # Y is the Y_dataset
    # n is the number of paramaters (theta)
    def __init__(self, X,Y, n):
        self.X=X
        self.Y=Y
        self.X_train=None
        self.Y_train
        self.X_train=None
        self.X_test=None
        self.y_train=None
        self.y_test=None

        # create a vector of theta with length n+1 (theta_0, theta_1, ......, theta_n)
        self.theta=np.zeros(n+1)
        ## write your code there


    # Add column of ones to X to represent x0
    def concatenate_X0_with_X(self):
        pass

    def split_dataset(self, test_percentage=0.3):
        ## used the train_test_split in sklearn
        pass


    # y_hat=theta' X
    def predict(self, X):
        return np.dot(self.theta.T,X)

    def fit(self, lr, epochs):

        pass


    ######## You can add any functions

```