**LINEAR REGRESSION USING GRADIENT DESCENT**

**Preamble**

In this tutorial, First, we will learn about the linear regression and loss function. Second, working of gradient descent algorithm, how it optimizes loss function for the given linear regression problem and implement it from scratch in scilab/python.

**Linear Regression:**

It’s a supervised machine learning algorithm which learns from given x dependent variable and Y as quantifiable variable , it predicts New Y from given new X. With simple linear regression we want to model our data as follows:

**H = aX + b**

where **‘a’** is the slope of the line and **‘b’** is the y intercept. The goal of linear regression is to fit a line to a set of points.

**Exercise I** Consider X = [1, 2, 3, 4, 5] and Y = [10, 40, 50, 78, 83] , where X is the object size and P is the corresponding price. Determine the value of **a** and **b**, such that the line corresponding to those values is the best fitting line. Plot the hypothesis for the predicted model.

**Loss Function:**

The loss is the error in our predicted value of a and b. Our goal is to minimize this error to obtain the most accurate value of a and b. Most commonly, Mean Squared Error function is used to calculate the loss function for linear regression problem.

**Exercise II** Evaluate the minimum value of cost function manually for the linear model Hi = aXi+ 0 ( i.e. with different values of **a** and **b**= 0 ) . Plot the cost function.

In Exercise II, we find minimum cost function, manually. However, we need an optimization algorithm to do it automatically.

**Gradient Descent Algorithm:**

Gradient descent is an iterative optimization algorithm to find the minimum of a loss function. The equation of gradient descent is as given below:

**Exercise III** Evaluate the minimum value of cost function for the linear model Hi = aXi+ b using Gradient Descent Algorithm (and iterative update). Plot the cost function with respect to slope.