## Experiment No.: 05

**Title:** Write a program for implementation of simple linear regression.

**Objectives:** 1. To learn simple linear regression.

**Theory:**

Regression is a method of modeling a target value based on independent predictors. This method is mostly used for forecasting and finding out cause and effect relationship between variables. There are two main types: **Simple Regression** and **Multivariable Regression**

- Simple linear regression uses traditional slope-intercept form where m and c are the variables our algorithm will try to "learn" to produce the most accurate predictions. x represents our input data and y represents our prediction. y=mx+c.
- Multivariable regression have more complex form

### Hypothesis Function

Hypothesis function has the general form:

$$\hat{y} = h\theta\ (x) = \theta_0 + \theta_1 x$$

For historical reasons, this function h is called a hypothesis. The job of the hypothesis is a function that takes as input for e.g. the size of a house like maybe the size of the new house your friend's trying to sell so it takes in the value of x and it tries to output the estimated value of y for the corresponding house. So h is a function that maps from x's to y's. When the target variable that we're trying to predict is continuous, such as in our housing example, we call the learning problem a **regression problem**.

Note that this is like the equation of a straight line. We give to $h_\theta(x)$ values for $\theta_0$ and $\theta_1$ to get our estimated output $\hat{y}$. In other words, we are trying to create a function called $h_\theta$ that is trying to map our input data (the x's) to our output data (the y's).

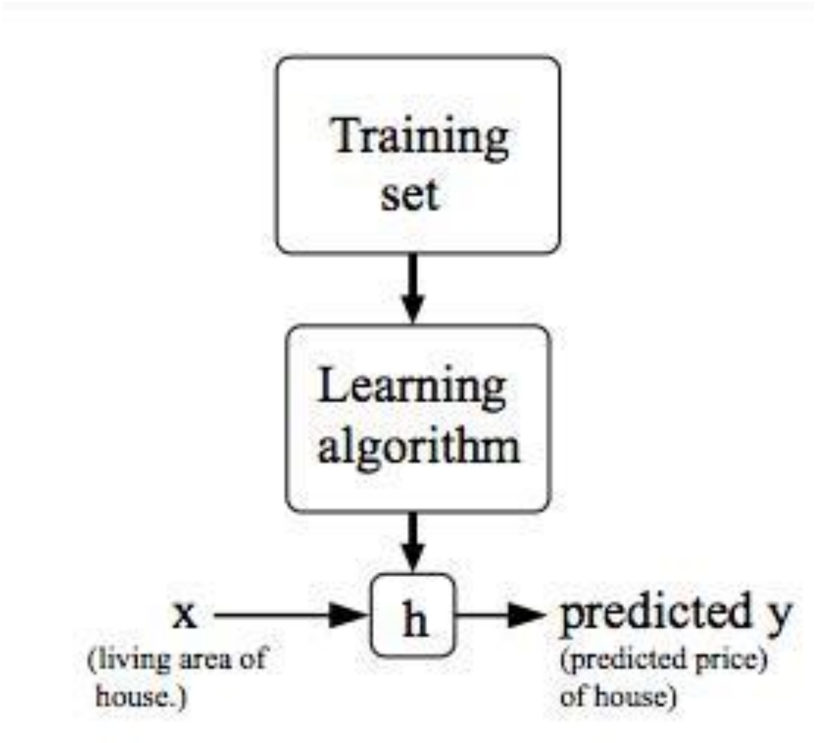
**Example:**

Suppose we have the following set of training data:

| input x | output y |
|---|---|
| 0 | 4 |
| 1 | 7 |
| 2 | 7 |
| 3 | 8 |

Now we can make a random guess about our **h***θ* function: *θ₀* =2 and *θ₁* =2. The hypothesis function becomes **h***θ***(x)** = ***2 + 2x***.

So for input of 1 to our hypothesis, **y** will be 4. This is off by 3. Note that we will be trying out various values of $\theta_0$ and $\theta_1$ to try to find values which provide the best possible "fit" or the most representative "straight line" through the data points mapped on the x-y plane.



**Cost Function:**

We can measure the accuracy of our hypothesis function by using a cost function. This takes an average (actually a fancier version of an average) of all the results of the hypothesis with inputs from x's compared to the actual output y's.

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(\hat{y}_i - y_i\right)^2 = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x_i) - y_i\right)^2$$

To break it apart, it is **(½)***x*□ where *x*□ is the mean of the squares of $h_\theta(x_i) - y_i$ , or the difference between the predicted value and the actual value.

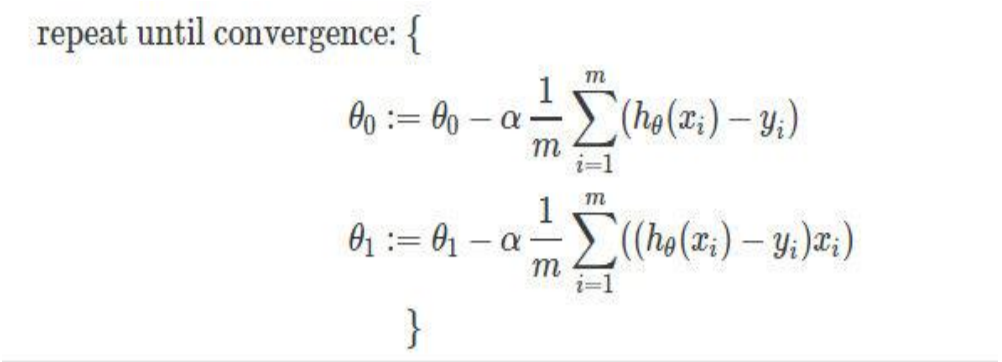
This function is otherwise called the "Squared error function", or "Mean squared error". The mean is halved **(½m)** as a convenience for the computation of the gradient descent, as the derivative term of the square function will cancel out the ½ term.

Now we are able to concretely measure the accuracy of our predictor function against the correct results we have so that we can predict new results we don't have.

If we try to think of it in visual terms, our training data set is scattered on the x-y plane. We are trying to make straight line (defined by $h_\theta(x)$) which passes through this scattered set of data. Our objective is to get the best possible line. The best possible line will be such so that the average squared vertical distances of the scattered points from the line will be the least. In the best case, the line should pass through all the points of our training data set. In such a case the value of **J(*$\theta_0$*, *$\theta_1$*)** will be **0**.

**Gradient Descent for Linear Regression**

When specifically applied to the case of linear regression, a new form of the gradient descent equation can be derived. We can substitute our actual cost function and our actual hypothesis function and modify the equation to:



repeat until convergence: {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} ((h_\theta(x_i) - y_i) x_i)$$

}

where **$\alpha$**, which is called the learning rate , **m** is the size of the training set, $\theta_0$ is a constant that will be changing simultaneously with $\theta_1$ and $x_i$ , $y_i$ are values of the given training set (data).

**Algorithm:**

1. Take housing price sample data containing 5 entries for x & y columns.
   (x=size of house in sq.ft., y=price of house)

| **input x**<br>**House Size(sq. ft.)** | **output y**<br>**House Price** |
|---|---|
| 1000 | 1180000 |
| 1120 | 1250000 |
| 1250 | 1300000 |
| 1400 | 1500000 |
| 1500 | 1650000 |

2. Calculate Gradient Descent.
3. Compute Cost Function.
4. Predict house price for the size of house=1300 sq.ft.

**Key Concepts:** linear regression, cost function, gradient descent.