**Assignment**

**Gradient Descent**

**Course: CSE 837**

Submitted by

|  |  |  |
| --- | --- | --- |
| *Md. Shayakh Shihab Uddin* | *Roll: 813* | *2015-16* |

Submitted to

Dr. B M Mainul Hossain

Associate Professor



**Institute of Information Technology**

**University of Dhaka**

[25-08-2019]

**Code**

import numpy as np

def gradient\_descent (x,y):

m = c = 0

iterations = 10000

n = len(x)

learning\_rate = 0.01

threshold = 0.0000000000000000001

cost = 0

for i in range(iterations):

previous\_cost = cost

y\_predicted = m \* x + c

cost = (1/n) \* sum([val\*\*2 for val in (y-y\_predicted)])

partial\_diff\_m = -(2/n)\*sum(x\*(y-y\_predicted))

partial\_diff\_c = -(2/n)\*sum(y-y\_predicted)

m = m - learning\_rate \* partial\_diff\_m

c = c - learning\_rate \* partial\_diff\_c

if abs(previous\_cost-cost) < threshold or i ==iterations:

return m, c, cost, i

x = np.array([1, 2, 3, 4, 5,7,9])

y = np.array([5, 7, 9, 11, 13, 15, 17])

m, c, cost, iteration=gradient\_descent(x, y)