

Gradient Descent
Mohammad Shakirul Islam
ID: 151-15-311

```
In [105]: #Import Libraries  
import numpy as np  
import pandas as pd
```

```
In [106]: x1 = [0, 0, 1, 1]  
x2 = [0, 1, 0, 1]  
y = [0, 0, 0, 1]  
  
#Theta ,  $\theta_1$  ,  $\theta_2$   
  
t1 = 0.3  
t2 = -0.1  
  
n = 0.1 #Learning Rate  
Th = 0.2 #Threshold Value
```

```

In [110]: for i in range(5):
            print('-'*79)
            print(' '*35,'Epoch:', i+1)
            print('-'*79)
            error = []
            temp = []
            for j in range(len(x1)):
                h= np.dot(x[j],t1)+ np.dot(x2[j],t2) # Predict Y value according to hypothesis

                if h < Th: # h = Hypothesis
                    h = 0
                else:
                    h = 1
                cost=y[j]-h # Cost Function
                temp.append(h)
                error.append(cost)
                if temp == y:

                    print( 'Input:', x1[j], x2[j],'Output:', y[j],
                            'Old Weight:', t1_temp, t2_temp,'Output: ',h,
                            'Cost:', cost, 'New Weight:', t1, t2)
                    break
            else:
                t1_temp = t1
                # Updating the weights w1 and w2

                t1 = t1 + n * x1[j] * cost
                t1 = float("{0:.2f}".format(t1))

                t2_temp = t2
                t2 = t2 + n * x2[j] * cost
                t2 = float("{0:.2f}".format(t2))
            print( 'Input:', x1[j], x2[j],'Output:', y[j],
                    'Old Weight:', t1_temp, t2_temp,'Output: ', h,
                    'Cost:', cost, 'New Weight:', t1, t2)

```

```

-----
--
                                Epoch: 1
-----
--
Input: 0 0 Output: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1
0.1
Input: 0 1 Output: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1
0.1
Input: 1 0 Output: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1
0.1
Input: 1 1 Output: 1 Old Weight: 0.1 0.1 Output: 1 Cost: 0 New Weight: 0.1
0.1
-----
--
                                Epoch: 2
-----
--
Input: 0 0 Output: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1
0.1
Input: 0 1 Output: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1

```

```
0.1
Input: 1 0 Output: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1
0.1
Input: 1 1 Output: 1 Old Weight: 0.1 0.1 Output: 1 Cost: 0 New Weight: 0.1
0.1
-----
--
                                Epoch: 3
-----
--
Input: 0 0 Output: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1
0.1
Input: 0 1 Output: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1
0.1
Input: 1 0 Output: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1
0.1
Input: 1 1 Output: 1 Old Weight: 0.1 0.1 Output: 1 Cost: 0 New Weight: 0.1
0.1
-----
--
                                Epoch: 4
-----
--
Input: 0 0 Output: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1
0.1
Input: 0 1 Output: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1
0.1
Input: 1 0 Output: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1
0.1
Input: 1 1 Output: 1 Old Weight: 0.1 0.1 Output: 1 Cost: 0 New Weight: 0.1
0.1
-----
--
                                Epoch: 5
-----
--
Input: 0 0 Output: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1
0.1
Input: 0 1 Output: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1
0.1
Input: 1 0 Output: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1
0.1
Input: 1 1 Output: 1 Old Weight: 0.1 0.1 Output: 1 Cost: 0 New Weight: 0.1
0.1
```

In []:

16 feb 2018