

```
In [1]: x1 = [0, 0, 1, 1]
        x2 = [0, 1, 0, 1]
        y = [0, 0, 0, 1]
        w1 = 0.3
        w2 = -0.1
        n = 0.1
        th = 0.2
```

```

In [16]: for i in range(5):
          print('Round:', i+1, )
          print()
          error = []
          temp = []
          for j in range(len(x1)):
              # Calculating the Y prediction
              y_pred =x1[j] * w1 + x2[j] * w2
              #  $h(x) = \theta_1 * x_1 + \theta_2 * x_2$ 
              if y_pred < th:
                  # If the predicted value is lesser than the threshold it will assign
                  y_pred = 0
              else:
                  y_pred = 1
              # Cost Function
              cost=y[j]-y_pred
              temp.append(y_pred)
              error.append(cost)
              if temp == y:
                  # print(w1, w2)
                  print()
                  print('Result: ')
                  print()
                  print( 'Inputs:', x1[j], x2[j], 'Outputs:', y[j], 'Old Weight:', w1_te
                        'Output: ', y_pred, 'Cost:', cost, 'New Weight:', w1, w2)
                  break
              else:
                  w1_temp = w1
                  # Updating the weights w1 and w2
                  # Rule: Old weight + learning rate * input * cost
                  w1 = w1 + n * x1[j] * cost
                  w1 = float("{0:.2f}".format(w1))
                  w2_temp = w2
                  w2 = w2 + n * x2[j] * cost
                  w2 = float("{0:.2f}".format(w2))
                  print( 'Inputs:', x1[j], x2[j], 'Outputs:', y[j], 'Old Weight:', w1_te
                        'Output: ', y_pred, 'Cost:', cost, 'New Weight:', w1, w2)

```

Round: 1

Inputs: 0 0 Outputs: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1 0.1

Inputs: 0 1 Outputs: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1 0.1

Inputs: 1 0 Outputs: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1 0.1

Result:

Inputs: 1 1 Outputs: 1 Old Weight: 0.1 0.1 Output: 1 Cost: 0 New Weight: 0.1 0.1

Round: 2

Inputs: 0 0 Outputs: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1 0.1

Inputs: 0 1 Outputs: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1 0.1

Inputs: 1 0 Outputs: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1 0.1

Result:

Inputs: 1 1 Outputs: 1 Old Weight: 0.1 0.1 Output: 1 Cost: 0 New Weight: 0.1 0.1

Round: 3

Inputs: 0 0 Outputs: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1 0.1

Inputs: 0 1 Outputs: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1 0.1

Inputs: 1 0 Outputs: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1 0.1

Result:

Inputs: 1 1 Outputs: 1 Old Weight: 0.1 0.1 Output: 1 Cost: 0 New Weight: 0.1 0.1

Round: 4

Inputs: 0 0 Outputs: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1 0.1

Inputs: 0 1 Outputs: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1 0.1

Inputs: 1 0 Outputs: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1 0.1

Result:

Inputs: 1 1 Outputs: 1 Old Weight: 0.1 0.1 Output: 1 Cost: 0 New Weight: 0.1 0.1

Round: 5

Inputs: 0 0 Outputs: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1 0.1

Inputs: 0 1 Outputs: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1 0.1

Inputs: 1 0 Outputs: 0 Old Weight: 0.1 0.1 Output: 0 Cost: 0 New Weight: 0.1 0.1

Result:

Inputs: 1 1 Outputs: 1 Old Weight: 0.1 0.1 Output: 1 Cost: 0 New Weight: 0.1 0.1