In [41]:
```
import numpy as np
```

In [42]:
```
# Step-1: Initialize Input and Output Variables
features=np.array([
    [0,0]
    ,[0,1]
    ,[1,0]
    ,[1,1]
])
```

In [43]:
```
labels= np.array([0, 0, 0, 1])#AND Gate
```

In [44]:
```
# step-2: Intialize the netwrok parameters
# epoch( Training Iterations)
# Bias=1, Learning Rate=(0to1)
# input Weights (w1,w2)
epoch = int(input('Enter the Epochs:'))
threshold= float(input('Enter the Threshold:'))
learning_rate = float(input('Enter the Learning Rate'))
a=[]
a.append(float(input('Enter the weights for x0:')))
a.append(float(input('Enter the weights for x1:')))
w=np.array(a)
#n=x1.shape[0]
```

```
Enter the Epochs:30
Enter the Threshold:0.3
Enter the Learning Rate0.4
Enter the weights for x0:0.4
Enter the weights for x1:0.6
```

In [45]:
```
for j in range(0,epoch):
    print("epoch:", j)
    global_delta = 0
    for i in range(0,features.shape[0]):
        #print(features[i])
        actual = labels[i]
        instance = features[i]

        x0 = instance[0]
        x1 = instance[1]
        print("X0:",x0)
        print("X1:",x1)

        sum_unit = x0 * w[0] + x1 * w[1]
        print("Sum Value:",sum_unit)

        if sum_unit > threshold:
            fire =1
        else:
            fire =0

        delta = actual - fire
        print("Delta:",delta)
        global_delta = global_delta + abs(delta)
        #print("global_delta:",global_delta)

        print("prediction:",fire,"where as actual was", actual, " (error:",delta,")")

        w[0] = w[0] + delta * learning_rate
        w[1] = w[1] + delta * learning_rate
    print("--------------------")

    if global_delta == 0:
        break
```

```
X1: 0
Sum Value: 0.0
Delta: 0
prediction: 0 where as actual was 0  (error: 0 )
X0: 0
X1: 1
Sum Value: 0.6
Delta: -1
prediction: 1 where as actual was 0  (error: -1 )
X0: 1
X1: 0
Sum Value: 0.0
Delta: 0
prediction: 0 where as actual was 0  (error: 0 )
X0: 1
X1: 1
Sum Value: 0.19999999999999996
Delta: 1
prediction: 0 where as actual was 1  (error: 1 )
--------------------
```

In [46]:
```
w
```

Out[46]:
```
array([0.4, 0.6])
```

In [ ]:
```

```