In [15]:
```
import numpy as np
```

In [16]:
```
def train(inputs,b,t,rho,learning_rate,n):
    print("Initially the Bottom-up weights b=",b,"\n")
    print("Initially the Top-down weights t = ",t,"\n")
    for s in inputs:
        print("S = ",s)
        norm_s=np.sum(s)
        print("||s|| = ",norm_s,"\n")
        x=s
        y=np.dot(x,b)
        print("y = bij*xi ",y)
        J=winner(y)
        print("Winner J = ",J,"\n")

        xi=s*t[J]
        print("xi = Si*tJi = ",xi)
        norm_x=np.sum(xi)
        print("||x||",norm_x)
        test_reset=norm_x/norm_s
        print("||x||/||s|| = ",test_reset)
        print("rho = ",rho)
        if(test_reset>=rho):
            print("Reset is False \n")
            for i in range(n):
                b[i][J]=learning_rate*xi[i]/(learning_rate-1+norm_x)
                t[J][i]=xi[i]
            print("bij(new) = ",b,"\n")
            print("tji(new) = ",t,"\n")
```

In [17]:
```
def winner(y):
    j=0
    for i in range(len(y)):
        if(y[i]>y[j]):
            j=i
    return j
```

In [18]:
```
import numpy as np

inputs = np.array([[0, 0, 0, 1], [0, 1, 0, 1], [0, 0, 1, 1], [1, 0, 0, 0]])
rho = 0.4 #vigilance parameter
learning_rate = 2
n = 4 #no of components in vector
m = 3 #no of clusters
norm_s = 1 #
norm_x = 1 #
bottom = 1/(1+n)  #bij(0)
```

In [19]:
```
b =np.full((n, m), bottom)
b
```

Out[19]:
```
array([[0.2, 0.2, 0.2],
       [0.2, 0.2, 0.2],
       [0.2, 0.2, 0.2],
       [0.2, 0.2, 0.2]])
```

In [20]:
```
t = np.full((m,n),1)
t
```

Out[20]:
```
array([[1, 1, 1, 1],
       [1, 1, 1, 1],
       [1, 1, 1, 1]])
```

In [21]:
```
train(inputs,b,t,rho,learning_rate,n)
```

```
Initially the Bottom-up weights b= [[0.2 0.2 0.2]
 [0.2 0.2 0.2]
 [0.2 0.2 0.2]
 [0.2 0.2 0.2]]

Initially the Top-down weights t =  [[1 1 1 1]
 [1 1 1 1]
 [1 1 1 1]]

S =  [0 0 0 1]
||s|| =  1

y = bij*xi  [0.2 0.2 0.2]
Winner J =  0

xi = Si*tJi =  [0 0 0 1]
||x|| 1
||x||/||s|| =  1.0
rho =  0.4
Reset is False

bij(new) =  [[0.  0.2 0.2]
 [0.  0.2 0.2]
 [0.  0.2 0.2]
 [1.  0.2 0.2]]

tji(new) =  [[0 0 0 1]
 [1 1 1 1]
 [1 1 1 1]]

S =  [0 1 0 1]
||s|| =  2

y = bij*xi  [1.  0.4 0.4]
Winner J =  0

xi = Si*tJi =  [0 0 0 1]
||x|| 1
||x||/||s|| =  0.5
rho =  0.4
Reset is False

bij(new) =  [[0.  0.2 0.2]
 [0.  0.2 0.2]
 [0.  0.2 0.2]
 [1.  0.2 0.2]]

tji(new) =  [[0 0 0 1]
 [1 1 1 1]
 [1 1 1 1]]

S =  [0 0 1 1]
||s|| =  2

y = bij*xi  [1.  0.4 0.4]
Winner J =  0

xi = Si*tJi =  [0 0 0 1]
||x|| 1
||x||/||s|| =  0.5
rho =  0.4
Reset is False

bij(new) =  [[0.  0.2 0.2]
 [0.  0.2 0.2]
 [0.  0.2 0.2]
 [1.  0.2 0.2]]

tji(new) =  [[0 0 0 1]
 [1 1 1 1]
 [1 1 1 1]]

S =  [1 0 0 0]
||s|| =  1

y = bij*xi  [0.  0.2 0.2]
Winner J =  1

xi = Si*tJi =  [1 0 0 0]
||x|| 1
||x||/||s|| =  1.0
rho =  0.4
Reset is False

bij(new) =  [[0.  1.  0.2]
 [0.  0.  0.2]
 [0.  0.  0.2]
 [1.  0.  0.2]]

tji(new) =  [[0 0 0 1]
```

```
[1 0 0 0]
[1 1 1 1]]
```

In [ ]: