a program in Python to implement Adaline Neural Network.

In [2]:
```
import numpy as np
x1=np.array([[1,1,-1,-1]])
x2=np.array([[1,-1,1,-1]])
t=np.array([[1],[1],[1],[-1]])
w11=0.1
w21=0.1
w01=0.1
alpha=0.1

i=0
bias=1

w1=np.zeros((4,1))
w2=np.zeros((4,1))
w0=np.zeros((4,1))
Yin=np.zeros((4,1))
y=np.zeros((4,1))
error=np.zeros((4,1))

count=0
while(count!=3):
    i=0
    if(count!=0):
        w11=w1[3]
        w21=w2[3]
        w01=w0[3]
    while(i!=4):
        if(i==0):
            Yin[i]= (x1[0][i]*w11)+(x2[0][i]*w21)+(bias*w01)
            y[i]=t[i][0]-Yin[i]
            w1[i]=w11+(alpha*y[i]*x1[0][i])
            w2[i]=w21+(alpha*y[i]*x2[0][i])
            w0[i]=w01+(alpha*y[i]*bias)
        else:
            if(i>0 & i<=4):
                Yin[i]= (x1[0][i]*w1[i-1])+(x2[0][i]*w2[i-1])+(bias*w0[i-1])
                y[i]=t[i][0]-Yin[i]
                w1[i]=w1[i-1]+(alpha*y[i]*x1[0][i])
                w2[i]=w2[i-1]+(alpha*y[i]*x2[0][i])
                w0[i]=w0[i-1]+(alpha*y[i]*bias)

                error[i]=(y[i])**2
        i=i+1
    print('EPOCH',(count+1),':')
    print('\n')
    print('w1:',w1)
    print('\n')
    print('w2:',w2)
    print('\n')
    print('w0:',w0)
    print('\n')
    print('error',error)
    print('\n\n')
    count=count+1
```

```
EPOCH 1 :


w1: [[0.17   ]
 [0.253  ]
 [0.1617 ]
 [0.26213]]


w2: [[0.17   ]
 [0.087  ]
 [0.1783 ]
 [0.27873]]


w0: [[0.17   ]
 [0.253  ]
 [0.3443 ]
 [0.24387]]


error [[0.        ]
 [0.6889    ]
 [0.833569  ]
 [1.00861849]]



EPOCH 2 :


w1: [[0.283657  ]
 [0.3587773 ]
 [0.27946497]
 [0.36305653]]


w2: [[0.300257  ]
 [0.2251367 ]
 [0.30444903]
 [0.38804059]]


w0: [[0.265397  ]
 [0.3405173 ]
 [0.41982963]
 [0.33623807]]


error [[0.        ]
 [0.56430595]
 [0.62904457]
 [0.69875494]]



EPOCH 3 :
```

```
w1: [[0.35432301]
 [0.42407096]
 [0.35234503]
 [0.42587987]]


w2: [[0.37930707]
 [0.30955912]
 [0.38128506]
 [0.45481989]]


w0: [[0.32750455]
 [0.3972525 ]
 [0.46897843]
 [0.3954436 ]]


error [[0.        ]
 [0.48647767]
 [0.51446097]
 [0.54073719]]
```

In [ ]: