

```
In [1]: """NAME - BHAVANA SIDDINENI
SECTION CSE-D
ROLL NUMBER - AP18110010246
QUESTION - HEBB NETWORK FOR AND AND OR"""
```

```
Out[1]: 'NAME - BHAVANA SIDDINENI \nSECTION CSE-D\nROLL NUMBER - AP18110010246\nQUESTION - HEBB NETWORK FOR AND AND OR'
```

```
In [2]: #main function
import matplotlib.pyplot as plt
import numpy as np
def function(arr,d,w1,w2,b):
    error=[]
    a=[]
    def f(error):
        if(len(error)==4):
            if(all(e == 0 for e in error)):
                return 1
            return 0
        return 0
    epoch=1
    print("|      epoch      |no      |w1      |w2      |b      |D      |E      |w1f      |w2f      |bf      |")
    while(f(error)==0):
        for i in range(len(arr)):
            y=d[i]
            w1n=w1+y*arr[i][0]
            w2n=w2+y*arr[i][1]
            bn=b+y
            e=d[i]-y
            error.append(e)
            print("|\\t",epoch,"\\t",i+1,"\\t",w1,"\\t",w2,"\\t",b,"\\t",d[i],"\\t",e,"\\t",w1n,"\\t",w2n,"\\t",bn,"\\t|")
            w1=w1n
            w2=w2n
            b=bn
        mse=[x*x for x in error]
        x=sum(mse)
        a.append(x)
        if(f(error)):
            break
        print("converges")
    else:
        epoch+=1
```

```

        print("-----")
        error.clear()

x1=[]
x2=[]
for i in range(len(arr)):
    x1.append(arr[i][0])
    x2.append(arr[i][1])
def decisionboundary(l,x1,x2,d,w1,w2):
    plt.figure(figsize=(5,5))
    plt.title("Decision Boundary")
    for i in range(l):
        if d[i]==1:
            color="g"
        else:
            color="r"
        plt.scatter(x1[i],x2[i],c=color)
    x=np.linspace(0,2,4)
    y=-x+1
    plt.plot(x,y)
    plt.xlabel('x1')
    plt.ylabel('x2')
    plt.show()
decisionboundary(len(arr),x1,x2,d,w1,w2)
xarr=[k for k in range(1,epoch+1)]
yarr=a
plt.plot(xarr, yarr, color='green', linestyle='solid', linewidth = 2,
        marker='o', markerfacecolor='blue', markersize=6)
plt.ylim(-1,epoch+1)
plt.xlim(-1,epoch+1)
plt.xlabel("EPOCH")
plt.ylabel("ERROR")
plt.title("ERROR VS EPOCH CURVE")
plt.show()

```

In [3]:

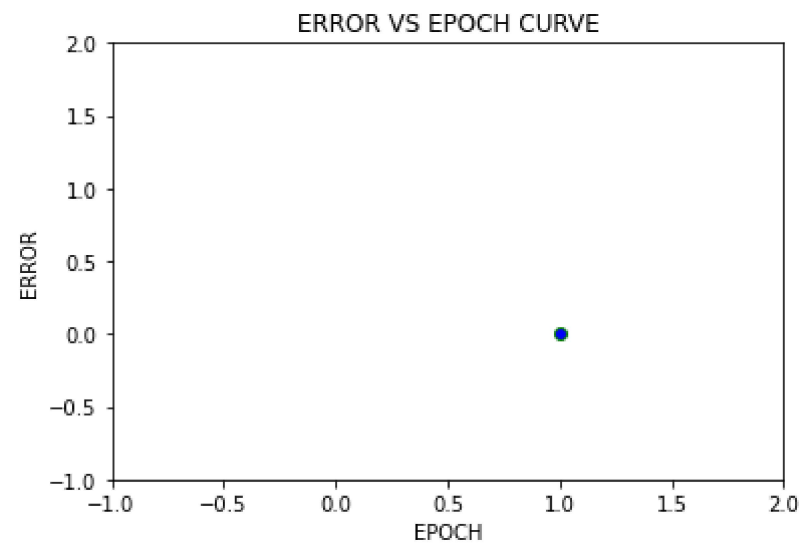
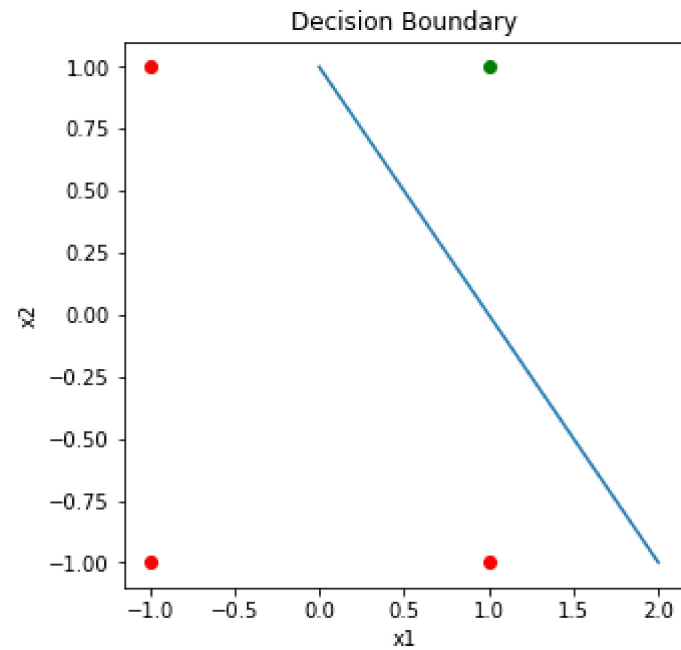
```

#and with bipolar inputs
arr=[[1,1],[1,-1],[-1,1],[-1,-1]]
d=[1,-1,-1,-1]
w1,w2,b=0,0,0
function(arr,d,w1,w2,b)

```

epoch	ino	w1	w2	b	D	E	w1f	w2f	bf
1	1	0	0	0	1	0	1	1	1

1	2	1	1	1	-1	0	0	2	0
1	3	0	2	0	-1	0	1	1	-1
1	4	1	1	-1	-1	0	2	2	-2



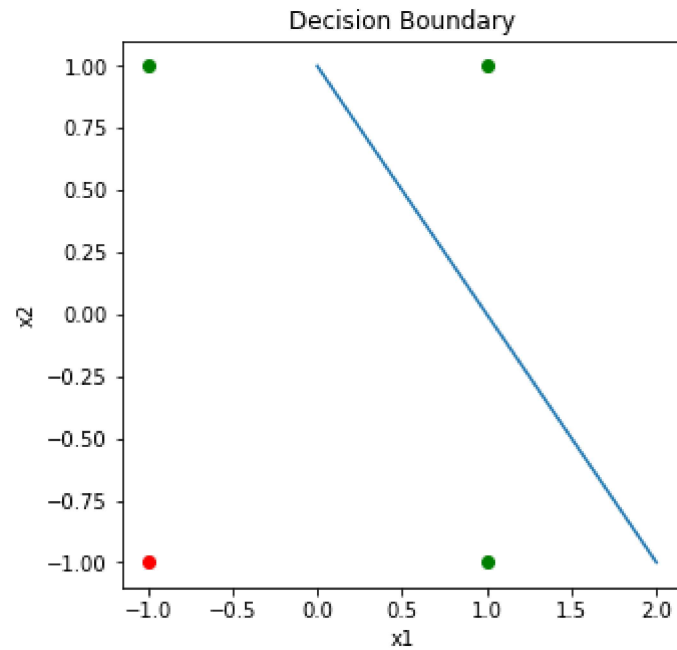
In [4]: *#or with bipolar inputs*

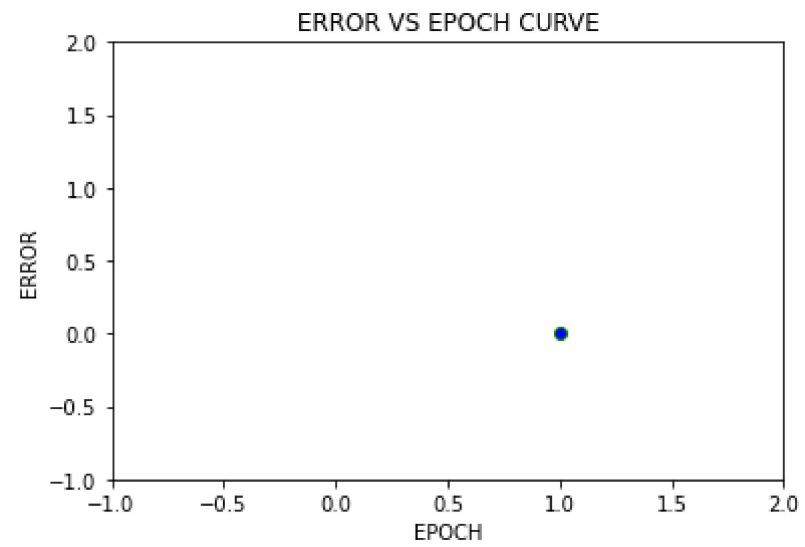
```

arr=[[1,1],[1,-1],[-1,1],[-1,-1]]
d=[1,1,1,-1]
w1,w2,b=0,0,0
function(arr,d,w1,w2,b)

```

epoch	ino	w1	w2	b	D	E	w1f	w2f	bf
1	1	0	0	0	1	0	1	1	1
1	2	1	1	1	1	0	2	0	2
1	3	2	0	2	1	0	1	1	3
1	4	1	1	3	-1	0	2	2	2





In []:

In []: