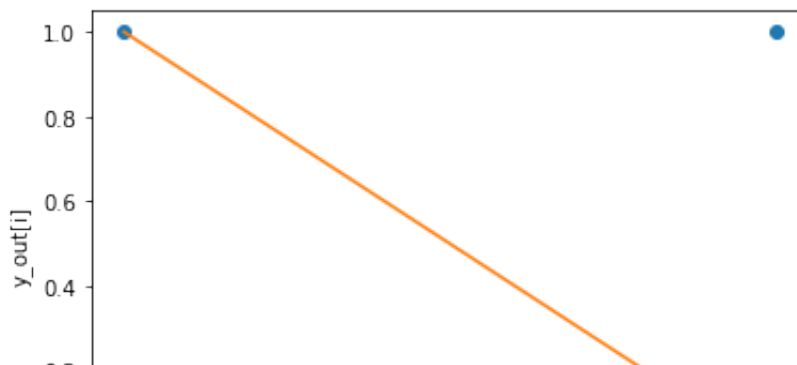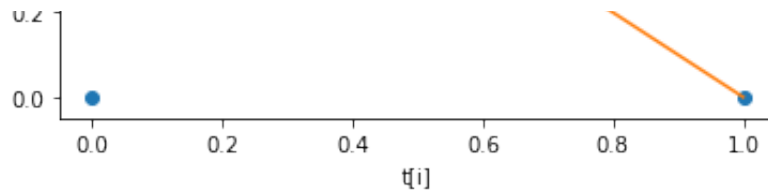## implement perceptron using OR

In [6]:
```python
import matplotlib.pyplot as plt
x0=[1,1,1,1]
x1=[0,0,1,1]
x2=[0,1,0,1]
t=[0,1,1,1]
w0=0
w1=0
w2=0
y_out=[0,0,0,0]
for j in range (4):
    for i in range(len(x1)):
        y_in=w0+(x1[i]*w1)+(x2[i]*w2)
        if(y_in>=0):
            y_out[i]=1
        else:
            y_out[i]=0
        if (y_out[i]!=t[i]):
            eta=0.2
            cw0=(eta*(t[i]-y_out[i])*x0[i])
            cw1=(eta*(t[i]-y_out[i])*x1[i])
            cw2=(eta*(t[i]-y_out[i])*x2[i])
            w0=w0+cw0
            w1=w1+cw1
            w2=w2+cw2
        i=i+1
        j=j+1
for i in range(4):
    print(t[i],y_out[i])
plt.xlabel('t[i]')
plt.ylabel('y_out[i]')
plt.plot(x1,x2,"o")
c=(-w0/w2,0.0)
d=(0.0,-w0/w1)
plt.plot(d,c)
```
```
1 1
1 1
```
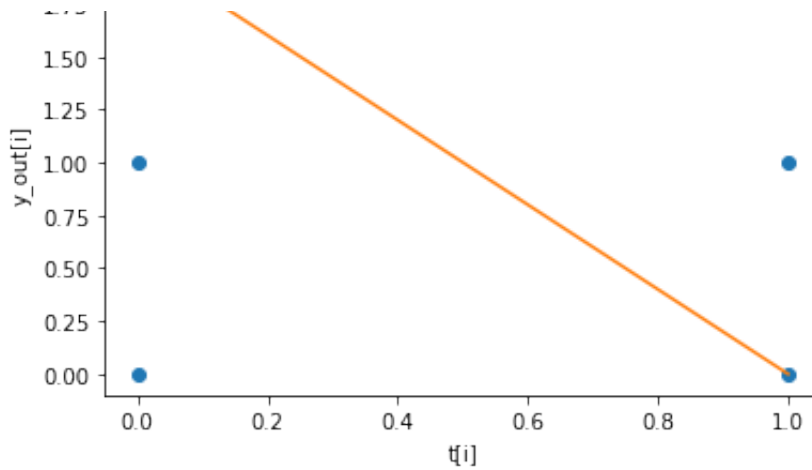
Out[6]: [<matplotlib.lines.Line2D at 0x11eb0d550>]

implement perceptron using AND

In [7]:
```python
import matplotlib.pyplot as plt
x0=[1,1,1,1]
x1=[0,0,1,1]
x2=[0,1,0,1]
t=[0,0,0,1]
w0=0
w1=0
w2=0
y_out=[0,0,0,0]
for j in range (4):
    for i in range(len(x1)):
        y_in=w0+(x1[i]*w1)+(x2[i]*w2)
        if(y_in>=0):
            y_out[i]=1
        else:
            y_out[i]=0
        if (y_out[i]!=t[i]):
            eta=0.2
            cw0=(eta*(t[i]-y_out[i])*x0[i])
            cw1=(eta*(t[i]-y_out[i])*x1[i])
            cw2=(eta*(t[i]-y_out[i])*x2[i])
            w0=w0+cw0
            w1=w1+cw1
            w2=w2+cw2
        i=i+1
        j=j+1
for i in range(4):
    print(t[i],y_out[i])
plt.xlabel('t[i]')
plt.ylabel('y_out[i]')
plt.plot(x1,x2,"o")
c=(-w0/w2,0.0)
d=(0.0,-w0/w1)
plt.plot(d,c)
```

```
0 0
0 0
0 0
1 1
```
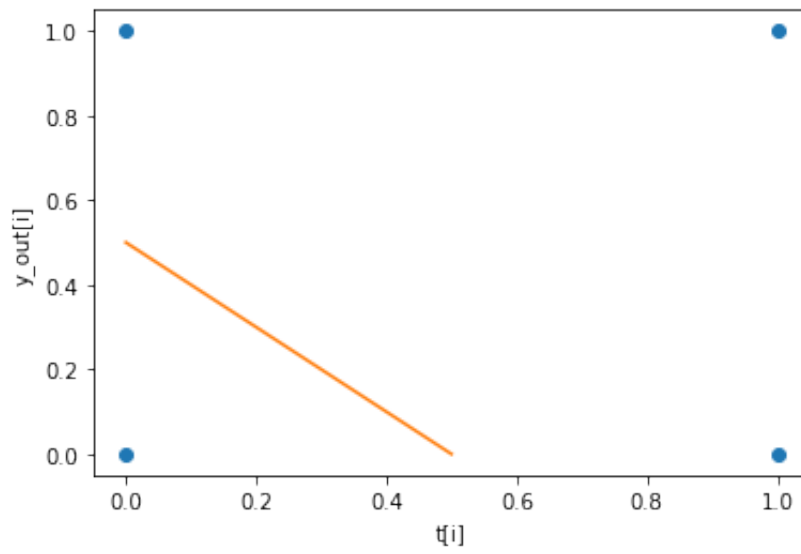
Out[7]: [<matplotlib.lines.Line2D at 0x11ebeb710>]

implement perceprton using NAND

```
In [8]: import matplotlib.pyplot as plt
        x0=[1,1,1,1]
        x1=[0,0,1,1]
        x2=[0,1,0,1]
        t=[1,1,1,0]
        w0=0
        w1=0
        w2=0
        y_out=[0,0,0,0]
        for j in range (4):
            for i in range(len(x1)):
                y_in=w0+(x1[i]*w1)+(x2[i]*w2)
                if(y_in>=0):
                    y_out[i]=1
                else:
                    y_out[i]=0
                if (y_out[i]!=t[i]):
                    eta=0.2
                    cw0=(eta*(t[i]-y_out[i])*x0[i])
                    cw1=(eta*(t[i]-y_out[i])*x1[i])
                    cw2=(eta*(t[i]-y_out[i])*x2[i])
                    w0=w0+cw0
                    w1=w1+cw1
                    w2=w2+cw2
                i=i+1
                j=j+1
        for i in range(4):
            print(t[i],y_out[i])
        plt.xlabel('t[i]')
        plt.ylabel('y_out[i]')
        plt.plot(x1,x2,"o")
        c=(-w0/w2,0.0)
        d=(0.0,-w0/w1)
        plt.plot(d,c)
```

1 1

```
1 1
1 0
0 1
```

Out[8]: [<matplotlib.lines.Line2D at 0x11ece6588>]

In [ ]: