

In [1]: *#and*

```
In [2]: import numpy as np
x=np.array([[1,1],[1,0],[0,1],[0,0]])
t=np.array([[1],[0],[0],[0]])
w=np.array([[0],[0]])
theta=1
yin=np.zeros(shape=(4,1))
y=np.zeros(shape=(4,1))
yin=np.dot(x,w)
i=0
found=0
while(found==0):
    i=0
    yin=np.dot(x,w)
    print("Y is initiallised",yin)
    while(i<4):
        if yin[i]>=theta:
            y[i]=1
            i=i+1
        else:
            y[i]=0
            i=i+1
    print("Calculated y",y)
    print("Expected Target t",t)
    if (y==t).all():
        print("MODEL IS TRAINED ")
        print("\nOutput : \n",y)
        print("\nweights : ",w,"\n")
        print("theta : ",theta)
        found=1
    else:
        print("MODEL IS NOT TRAINED")
        w=np.zeros(shape=(0,0))
        theta=int(input("Enter New Theta : "))
        for k in range(int(2)):
            w1=int(input("Enter Weight : "))
            w=np.append(w,w1)
```

```
Y is initialised [[0]
[0]
[0]
[0]]
Calculated y [[0.]
[0.]
[0.]]
Expected Target t [[1]
[0]
[0]
[0]]
MODEL IS NOT TRAINED
Enter New Theta : 2
Enter Weight : 1
Enter Weight : 1
Y is initialised [2. 1. 1. 0.]
Calculated y [[1.]
[0.]
[0.]]
Expected Target t [[1]
[0]
[0]
[0]]
MODEL IS TRAINED

Output :
[[1.]
[0.]
[0.]
[0.]]

weights : [1. 1.]

theta : 2
```

In [3]: *# or*

```
In [4]: import numpy as np
x=np.array([[1,1],[1,0],[0,1],[0,0]])
t=np.array([[1],[1],[1],[0]])
w=np.array([[0],[0]])
theta=1
yin=np.zeros(shape=(4,1))
y=np.zeros(shape=(4,1))
yin=np.dot(x,w)
i=0
found=0
while(found==0):
    i=0
    yin=np.dot(x,w)
    print("Y is initiallised",yin)
    while(i<4):
        if yin[i]>=theta:
            y[i]=1
            i=i+1
        else:
            y[i]=0
            i=i+1
    print("Calculated y",y)
    print("Expected Target t",t)
    if (y==t).all():
        print("MODEL IS TRAINED ")
        print("\nOutput : \n",y)
        print("\nweights : ",w,"\n")
        print("theta : ",theta)
        found=1
    else:
        print("MODEL IS NOT TRAINED")
        w=np.zeros(shape=(0,0))
        theta=int(input("Enter New Theta : "))
        for k in range(int(2)):
            w1=int(input("Enter Weight : "))
            w=np.append(w,w1)
```

```
Y is initialised [[0]
[0]
[0]
[0]]
Calculated y [[0.]
[0.]
[0.]]
Expected Target t [[1]
[1]
[1]
[0]]
MODEL IS NOT TRAINED
Enter New Theta : 1
Enter Weight : 11
Enter Weight : 1
Y is initialised [12. 11.  1.  0.]
Calculated y [[1.]
[1.]
[1.]
[0.]]
Expected Target t [[1]
[1]
[1]
[0]]
MODEL IS TRAINED

Output :
[[1.]
[1.]
[1.]
[0.]]

weights :  [11.  1.]

theta :  1
```

```
In [5]: #NOR
import numpy as np
x=np.array([[1,1],[1,0],[0,1],[0,0]])
t=np.array([[0],[0],[0],[1]])
w=np.array([[0],[0]])
theta=1
yin=np.zeros(shape=(4,1))
y=np.zeros(shape=(4,1))
yin=np.dot(x,w)
i=0
found=0
while(found==0):
    i=0
    yin=np.dot(x,w)
    print("Y is initiallised",yin)
    while(i<4):
        if yin[i]>=theta:
            y[i]=1
            i=i+1
        else:
            y[i]=0
            i=i+1
    print("Calculated y",y)
    print("Expected Target t",t)
    if (y==t).all():
        print("MODEL IS TRAINED ")
        print("\nOutput : \n",y)
        print("\nweights : ",w,"\n")
        print("theta : ",theta)
        found=1
    else:
        print("MODEL IS NOT TRAINED")
        w=np.zeros(shape=(0,0))
        theta=int(input("Enter New Theta : "))
        for k in range(int(2)):
            w1=int(input("Enter Weight : "))
            w=np.append(w,w1)
```

```
Y is initialised [[0]
[0]
[0]
[0]]
Calculated y [[0.]
[0.]
[0.]
[0.]]
Expected Target t [[0]
[0]
[0]
[1]]
MODEL IS NOT TRAINED
Enter New Theta : 0
Enter Weight : -1
Enter Weight : -1
Y is initialised [-2. -1. -1.  0.]
Calculated y [[0.]
[0.]
[1.]
[1.]]
Expected Target t [[0]
[0]
[0]
[1]]
MODEL IS TRAINED

Output :
[[0.]
[0.]
[0.]
[1.]]

weights :  [-1. -1.]

theta :  0
```

In [6]: *# and not*

```
In [7]: import numpy as np
x=np.array([[1,1],[1,0],[0,1],[0,0]])
t=np.array([[0],[1],[0],[0]])
w=np.array([[0],[0]])
theta=1
yin=np.zeros(shape=(4,1))
y=np.zeros(shape=(4,1))
yin=np.dot(x,w)
i=0
found=0
while(found==0):
    i=0
    yin=np.dot(x,w)
    print("Y is initiallised",yin)
    while(i<4):
        if yin[i]>=theta:
            y[i]=1
            i=i+1
        else:
            y[i]=0
            i=i+1
    print("Calculated y",y)
    print("Expected Target t",t)
    if (y==t).all():
        print("MODEL IS TRAINED ")
        print("\nOutput : \n",y)
        print("\nweights : ",w,"\n")
        print("theta : ",theta)
        found=1
    else:
        print("MODEL IS NOT TRAINED")
        w=np.zeros(shape=(0,0))
        theta=int(input("Enter New Theta : "))
        for k in range(int(2)):
            w1=int(input("Enter Weight : "))
            w=np.append(w,w1)
```



```
Y is initialised [[0]
[0]
[0]
[0]]
Calculated y [[0.]
[0.]
[0.]]
Expected Target t [[0]
[1]
[0]
[0]]
MODEL IS NOT TRAINED
Enter New Theta : 1
Enter Weight : 1
Enter Weight : -1
Y is initialised [ 0.  1. -1.  0.]
Calculated y [[0.]
[1.]
[0.]
[0.]]
Expected Target t [[0]
[1]
[0]
[0]]
MODEL IS TRAINED

Output :
[[0.]
[1.]
[0.]
[0.]]

weights : [ 1. -1.]

theta : 1
```

In [8]: *#not*

```
In [9]: import numpy as np
x=np.array([[0],[1]])
t=np.array([[1],[0]])
w=np.array([0])
theta=1
yin=np.zeros(shape=(2,1))
y=np.zeros(shape=(2,1))
yin=np.dot(x,w)
i=0
found=0
while(found==0):
    i=0
    yin=np.dot(x,w)
    print(yin)
    while(i<2):
        if yin[i]>=theta:
            y[i]=1
            i=i+1

        #if(i==4):
        #break
        else:
            y[i]=0
            i=i+1
    print("y",y)
    print("t",t)
    if (y==t).all():
        print("MODEL IS TRAINED ")
        print("\nOutput : \n",y)
        print("\nweights : ",w,"\n")
        print("theta : ",theta)
        found=1
    else:
        print("MODEL IS NOT TRAINED")
        w=np.zeros(shape=(0,0))
        theta=int(input("Enter New Theta : "))
    for k in range(int(1)):
        w=int(input("Enter Weight : "))
```

```
[0 0]
y [[0.]
   [0.]]
t [[1]
   [0]]
MODEL IS NOT TRAINED
Enter New Theta : 0
Enter Weight : -1
[[ 0]
 [-1]]
y [[1.]
   [0.]]
t [[1]
   [0]]
MODEL IS TRAINED

Output :
[[1.]
 [0.]]

weights :  -1

theta :  0
Enter Weight : -1
```

In [ ]: