

ESERCITAZIONE

if **A** is TRUE and **B** is TRUE, then **A OR B** is TRUE

if **A** is TRUE and **B** is FALSE, then **A OR B** is TRUE

if **A** is FALSE and **B** is TRUE, then **A OR B** is TRUE

if **A** is FALSE and **B** is FALSE, then **A OR B** is FALSE

TRUE -> 1

FALSE -> 0

A	B	A OR B
1	1	1
1	0	1
0	1	1
0	0	0

ESERCITAZIONE

```
import random
lr = 0.1 #learning rate
bias = -1
weights = [random.random(),random.random(),random.random()]
```

#HEBB RULE

```
def Perceptron(input1, input2, output) :
    netinput = input1*weights[0]+input2*weights[1]+bias*weights[2]
    if netinput > 0 : #activation function (here Heaviside)
        outputP = 1
    else :
        outputP = 0
```

ESERCITAZIONE

Regola di HEBB

```
if outputP != output:  
    weights[0] += input1 * output * lr  
    weights[1] += input2 * output * lr  
    weights[2] += bias * output * lr
```

ESERCITAZIONE

Attivazione della rete neurale presentando tutti gli esempi

```
for i in range(100) :  
    Perceptron(1,1,1) #True or true  
    Perceptron(1,0,1) #True or false  
    Perceptron(0,1,1) #False or true  
    Perceptron(0,0,0) #False or false
```

A	B	A OR B
1	1	1
1	0	1
0	1	1
0	0	0

Testiamo il comportamento della rete neurale

```
x = int(input())  
y = int(input())  
netinput = x*weights[0] + y*weights[1] + bias*weights[2]  
if netinput > 0: #activation function  
    outputP = 1  
else :  
    outputP = 0  
print(x, "or", y, "is : ", outputP)
```