A70 Pratik Jade Pratical No- 4

Implimenting logic gate using mc culloch pitts model AND gate using mc culloch-pits network

In [1]:
```
#Step 1: generate a vector of inputs and a vector of weights##Implimenting logic
import numpy as np
```

In [2]:
```
# matrix of inputs
input_table = np.array([
    [0,0], # both no
    [0,1], # one no, one yes
    [1,0], # one yes, one no
    [1,1]  # bot yes
])

print(f'input table:\n{input_table}')
```

```
input table:
[[0 0]
 [0 1]
 [1 0]
 [1 1]]
```

In [3]:
```
# array of weights
weights = np.array([1,1])
print(f'weights: {weights}')
```

```
weights: [1 1]
```

In [4]:
```
#Step 2: compute the dot product between the matrix of inputs and weights
# dot product matrix of inputs and weights
dot_products = input_table @ weights
print(f'Dot products: {dot_products}')
```

```
Dot products: [0 1 1 2]
```

In [5]:
```
#Step 3: define the threshold activation function
def linear_threshold_gate(dot: int, T: float) -> int:
    '''Returns the binary threshold output'''
    if dot >= T:
        return 1
    else:
        return 0
```

In [6]:
```
#Step 4: compute the output based on the threshold value
T = 2
for i in range(0,4):
    activation = linear_threshold_gate(dot_products[i], T)
    print(f'Activation: {activation}')
```

```
Activation: 0
Activation: 0
Activation: 0
Activation: 1
```

OR Function using mc culloch-pits network

In [7]:
```
#Now, let’s repeat the same four steps.

#Step 1: generate a vector of inputs and a vector of weights
        #Neither the matrix of inputs nor the array of weights changes, so we can

#Step 2: compute the dot product between the matrix of inputs and weights
        #Since neither the matrix of inputs nor the vector of weights changes, th

#Step 3: define the threshold activation function
         #We can use the linear_threshold_gate function again.
```

In [8]:
```
T = 1
for i in range(0,4):
    activation = linear_threshold_gate(dot_products[i], T)
    print(f'Activation: {activation}')
```

```
Activation: 0
Activation: 1
Activation: 1
Activation: 1
```

NOT gate using mc culloch-pits network

In [9]:
```
for i in range(0,2):
  if i<1:
    print('output= 1 ')
  elif i >=1:
    print('output = 0')
```

```
output= 1
output = 0
```