Meta Pitts Neuron (AND & OR Gates)

```python
import numpy as np
def mp_neuron(x,w,theta):
    return 1 if np.dot(x, w) >= theta else 0

print("--------AND GATE--------")

# AND Gate
w_and = np.array ([1,1])
theta_and = 2
for x in [(0,0), (0,1), (1,0), (1,1)]:
    print(x, mp_neuron(np.array(x), w_and, theta_and))

print("--------OR GATE--------")

# OR Gate
w_or = np.array ([1,1])
theta_or = 1
for x in [(0,0), (0,1), (1,0), (1,1)]:
    print(x, mp_neuron(np.array(x), w_or, theta_or))


print("+-------------------------------+")
print("Shivam Sharma, 1/23/SET/BCS/424")
print("22 January 2026")
```

```
--------AND GATE--------
(0, 0) 0
(0, 1) 0
(1, 0) 0
(1, 1) 1
--------OR GATE--------
(0, 0) 0
(0, 1) 1
(1, 0) 1
(1, 1) 1
+-------------------------------+
Shivam Sharma, 1/23/SET/BCS/424
22 January 2026
```

AND & OR GATE(3-inputs)

```python
import numpy as np
def mp_neuron(x,w,theta):
    return 1 if np.dot(x, w) >= theta else 0

print("--------AND GATE(3-inputs)--------")

w_and = np.array ([1,1,1])
theta_and = 3
for x in [(0,0,0), (0,0,1), (0,1,0), (0,1,1), (1,0,0), (1,0,1), (1,1,0), (1,1,1)]:
    print(x, mp_neuron(np.array(x), w_and, theta_and))


print("--------OR GATE(3-inputs)--------")

w_or = np.array ([1,1,1])
theta_or = 1
for x in [(0,0,0), (0,0,1), (0,1,0), (0,1,1), (1,0,0), (1,0,1), (1,1,0), (1,1,1)]:
    print(x, mp_neuron(np.array(x), w_or, theta_or))

print("+-----------------------------+")
print("Shivam Sharma, 1/23/SET/BCS/424")
print("22 January 2026")
```

```
--------AND GATE(3-inputs)--------
(0, 0, 0) 0
(0, 0, 1) 0
(0, 1, 0) 0
(0, 1, 1) 0
(1, 0, 0) 0
(1, 0, 1) 0
(1, 1, 0) 0
(1, 1, 1) 1
--------OR GATE(3-inputs)-------
(0, 0, 0) 0
(0, 0, 1) 1
(0, 1, 0) 1
(0, 1, 1) 1
(1, 0, 0) 1
(1, 0, 1) 1
(1, 1, 0) 1
(1, 1, 1) 1
+-----------------------------+
Shivam Sharma, 1/23/SET/BCS/424
22 January 2026
```

Artificial Neuron

```python
import numpy as np

def art_neuron(x, w, b, activation='step'):
  z = np.dot(x, w) + b

  if activation == 'step':
    return 1 if z >= 0 else 0
  elif activation == 'sigmoid':
    return 1 / (1 + np.exp(-z))
  elif activation == 'relu':
    return max(0, z)
  else:
    return "Invalid Activation Function"

x = np.array([1,2])
w = np.array([0.5,1.5])
b = -1

print("---------------------")
print("Step Output      :", art_neuron(x, w, b, activation='step'))
print("---------------------")
print("Sigmoid Output      :", art_neuron(x, w, b, activation='sigmoid'))
print("---------------------")
print("ReLu Output      :", art_neuron(x, w, b, activation='relu'))


print("+------------------------------+")
print("Shivam Sharma, 1/23/SET/BCS/424")
print("22 January 2026")
```

```
---------------------
Step Output     : 1
---------------------
Sigmoid Output     : 0.9241418199787566
---------------------
ReLu Output     : 2.5
+------------------------------+
Shivam Sharma, 1/23/SET/BCS/424
22 January 2026
```