Name :-Kunal R Agarwal

SRN No:-201700110

Roll No:-18

Question:-**Implement Neural network model using logical operations and ,or ,xor.**

**Input:-**

1 1

1 0

0 1

0 0

For AND:-

w1 = 1, w2 = 1, b = -1.5

For OR

w1 = 1, w2 = 1, b = 0.5

**Code:-**

import numpy as np

def unit\_step(v):

if v >= 0:

return 1

else:

return 0

def perceptron(x, w, b):

v = np.dot(w, x) + b

y = unit\_step(v)

return y

def NOT\_percep(x):

return perceptron(x, w=-1, b=0.5)

def AND\_percep(x):

w = np.array([1, 1])

b = -1.5

return perceptron(x, w, b)

def OR\_percep(x):

w = np.array([1, 1])

b = -0.5

return perceptron(x, w, b)

def XOR\_net(x):

gate\_1 = AND\_percep(x)

gate\_2 = NOT\_percep(gate\_1)

gate\_3 = OR\_percep(x)

new\_x = np.array([gate\_2, gate\_3])

output = AND\_percep(new\_x)

return output

# Test

example1 = np.array([1, 1])

example2 = np.array([1, 0])

example3 = np.array([0, 1])

example4 = np.array([0, 0])

print("The output of AND with weight w1=1 ,w2=1 and b=-1.5 is:-")

print("AND({}, {}) = {}".format(1, 1, AND\_percep(example1)))

print("AND({}, {}) = {}".format(1, 0, AND\_percep(example2)))

print("AND({}, {}) = {}".format(0, 1, AND\_percep(example3)))

print("AND({}, {}) = {}".format(0, 0, AND\_percep(example4)))

print("The output of OR with weight w1=1 ,w2=1 and b=-0.5 is:-")

print("OR({}, {}) = {}".format(1, 1, OR\_percep(example1)))

print("OR({}, {}) = {}".format(1, 0, OR\_percep(example2)))

print("OR({}, {}) = {}".format(0, 1, OR\_percep(example3)))

print("OR({}, {}) = {}".format(0, 0, OR\_percep(example4)))

print("The output of XOR is:-")

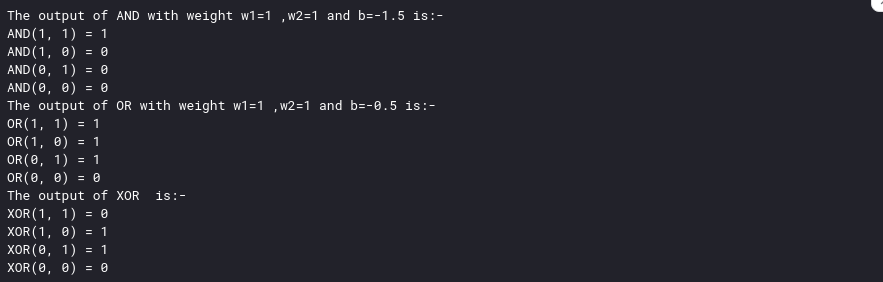
print("XOR({}, {}) = {}".format(1, 1, XOR\_net(example1)))

print("XOR({}, {}) = {}".format(1, 0, XOR\_net(example2)))

print("XOR({}, {}) = {}".format(0, 1, XOR\_net(example3)))

print("XOR({}, {}) = {}".format(0, 0, XOR\_net(example4)))

**Output:-**

****