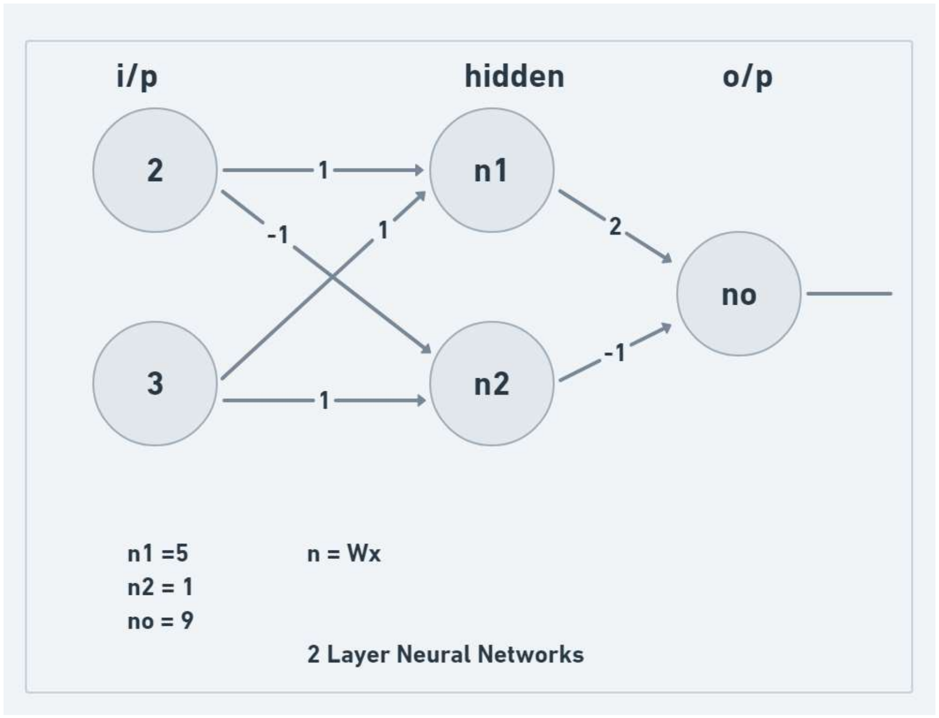
IMPLEMENTATION OF NEURAL NETWORKS

Q1. Implementation of Artificial Neural Network for OR logic gate with 2-bit Binary Input.

```
import numpy as np

def perceptron(x,w,b):
  v=np.dot(w,x)+b
  return v

def OR(x):
  w = np.array([1,1])
  b = 0
  return perceptron(x,w,b)

p1 = np.array([0,0])
p2 = np.array([0,1])
p3 = np.array([1,0])
p4 = np.array([1,1])

print("OR({},{}) = {}".format(0,0, OR(p1)))
print("OR({},{}) = {}".format(0,1, OR(p2)))
print("OR({},{}) = {}".format(1,0, OR(p3)))
print("OR({},{}) = {}".format(1,1, OR(p4)))
```

```
OR(0,0) = 0
OR(0,1) = 1
OR(1,0) = 1
OR(1,1) = 2
```

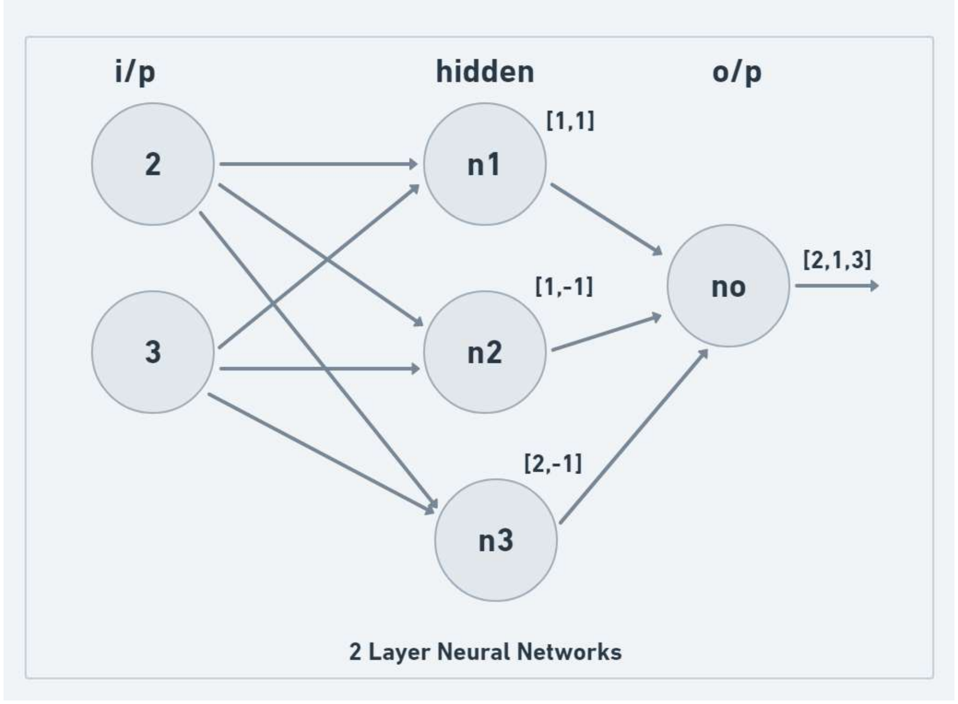
Q2. Implement the network as a python program.



```
x = [2,3]
w1 = [1,1]
n1 = np.dot(x,w1)
w2 = [-1,1]
n2 = np.dot(x,w2)
w0 =[2,-1]
y = [n1,n2]
n = np.dot(y,w0)
print(n1)
print(n2)
print(n)
```

```
5
1
9
```

Q3 Implement the network as a python program



2 Layer Neural Networks

```
import numpy as np

def percept(n,ip,wtHidden,wtOutput):
  lst = list()
  ip = np.transpose(ip)
  for i in range(n):
    lst.append(np.dot(wtHidden[i],ip))
  lst = np.transpose(lst)
  return np.dot(lst,wtOutput)

print("Enter the inputs: ")
ip = list(map(int, input().split()))
n = int(input("Enter the number of neurons in the hidden layer: \n"))
wtHidden = []
for i in range(n):
  wt = []
  print("Enter the weight W", i+1, ": ")
  wt = list(map(int,input().split()))
  wtHidden.append(wt)
print("Enter the weight of output layer WtOutput: ")
wtOutput = list(map(int, input().split()))
print("The output is: ", percept(n,ip,wtHidden, wtOutput))
```

```
Enter the inputs: 
2 3
Enter the number of neurons in the hidden layer: 
3
Enter the weight W 1 : 
1 2
Enter the weight W 2 : 
1 -1
Enter the weight W 3 : 
2 -1
Enter the weight of output layer WtOutput: 
2 1 3
The output is:  18
```