

| | | |
|--|--|--|
| | | |
|--|--|--|

Name: Nachiket Erlekar

Roll: 41434

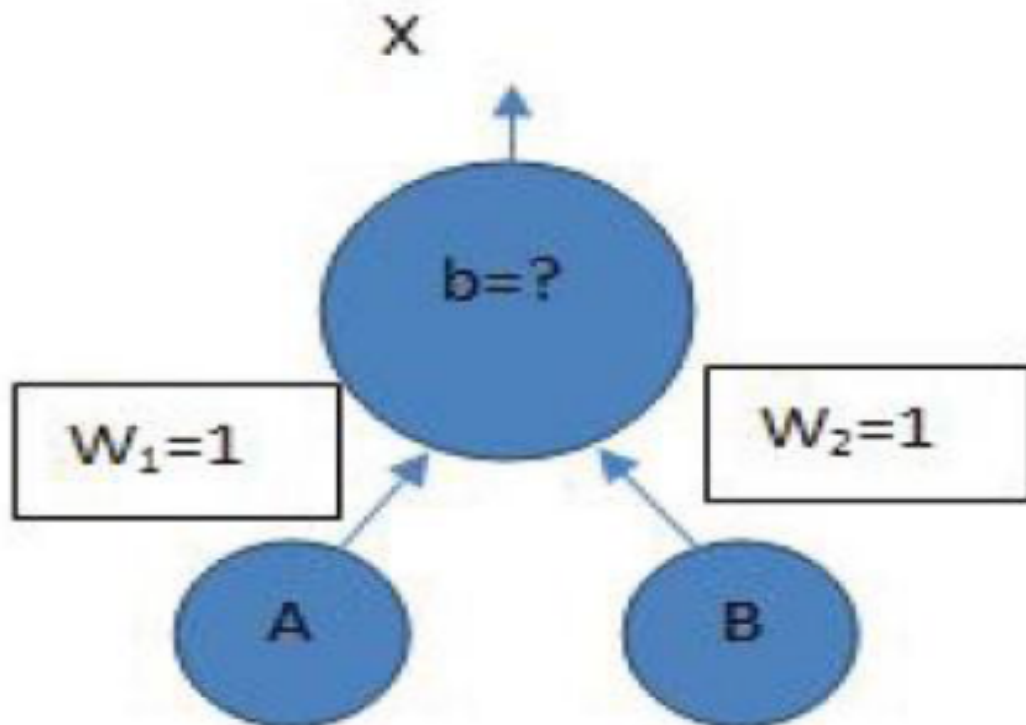
Batch: Q4

Assignment 5

Title: Single Layer Perceptron

Problem Statement:

Write a program to find the Boolean function to implement following single layer perceptron. Assume all activation functions to be the threshold function which is 1 for all input values greater than zero and 0, otherwise.



Objectives:

1. Understand concept and basics of single layer perceptron.

| | | |
|--|--|--|
| | | |
|--|--|--|

| | | |
|--|--|--|
| | | |
|--|--|--|

2.Understand Boolean logic implementation using perceptron.

Outcomes:

- 1.Understand and implement concept and basics of single layer perceptron.
- 2.Understand and implement Boolean logic implementation using perceptron.

Software Requirements:

1. Jupyter Notebook
2. Python 3.8.5

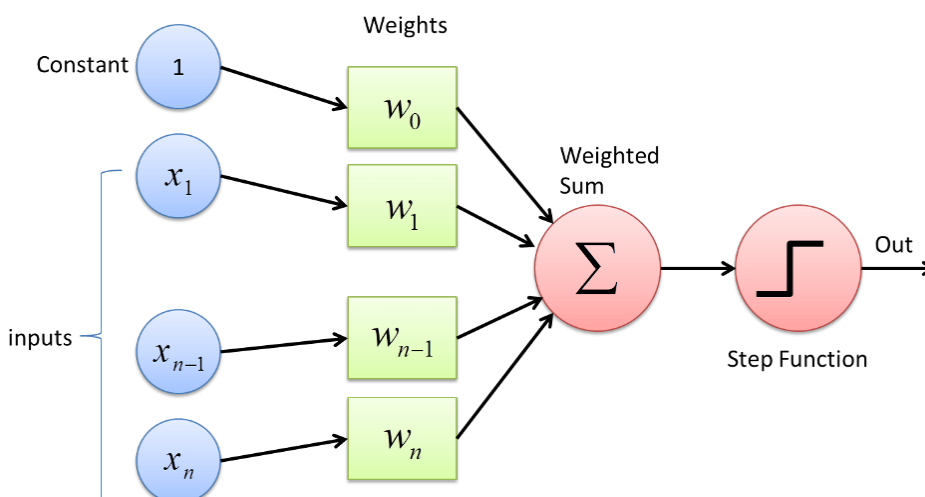
Hardware Requirements:

4GB RAM, processor i3 and above, 500GB HDD, OS Windows/Ubuntu/Fedora.

Theory:

Perceptron:

A Perceptron is an algorithm used for supervised learning of binary classifiers. Binary classifiers decide whether an input, usually represented by a series of vectors, belongs to a specific class. In short, a perceptron is a single-layer neural network. They consist of four main parts including input values, weights and bias, net sum, and an activation function.



| | | |
|--|--|--|
| | | |
|--|--|--|

| | | |
|--|--|--|
| | | |
|--|--|--|

Perceptron is as a model that implements the following function:

$$\hat{y} = \Theta(w_1x_1 + w_2x_2 + \dots + w_nx_n + b)$$

$$= \Theta(\mathbf{w} \cdot \mathbf{x} + b)$$

$$\text{where } \Theta(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Test Cases:

| Operation | Input | Expected O/P | Actual O/P | Result |
|-----------|-------|--------------|------------|---------|
| And | 0 0 | 0 | 0 | Success |
| | 0 1 | 0 | 0 | |
| | 1 0 | 0 | 0 | |
| | 1 1 | 1 | 1 | |
| OR | 0 0 | 0 | 0 | Success |
| | 0 1 | 1 | 1 | |
| | 1 0 | 1 | 1 | |
| | 1 1 | 1 | 1 | |
| NOT | 0 | 1 | 0 | Success |
| | 1 | 0 | 1 | |
| XOR | 0 0 | 0 | 0 | Success |
| | 0 1 | 1 | 1 | |
| | 1 0 | 1 | 1 | |
| | 1 1 | 0 | 0 | |

Conclusion:

Thus, we implemented single layer perceptron for Boolean functions.

| | | |
|--|--|--|
| | | |
|--|--|--|

| | | |
|--|--|--|
| | | |
|--|--|--|

| | | |
|--|--|--|
| | | |
|--|--|--|

Source code and Output

Code

```
import numpy as np
w=np.array([1,1])
A=int(input('Enter the value of A:'))
B=int(input('Enter the value of B:'))
b=1
x=np.array([A,B])
y=np.dot(w,x)+b
output=0
if y>1:
    output=1
elif y==0 or y<0 or y==1:
    output=0
print('The ouput of perceptron: ',y)
print('The bitwise OR of the inputs is:',output)
```

Output—

Enter the value of A:1

Enter the value of B:0

The ouput of perceptron: 2

The bitwise OR of the inputs is: 1

Enter the value of A:0

Enter the value of B:0

The ouput of perceptron: 1

The bitwise OR of the inputs is: 0