EXPERIMENT NO. 4

AIM: Implement OR and AND logic function using Perceptron Neural Network.

SOFTWARE: Python

LABORATORY OUTCOMES:

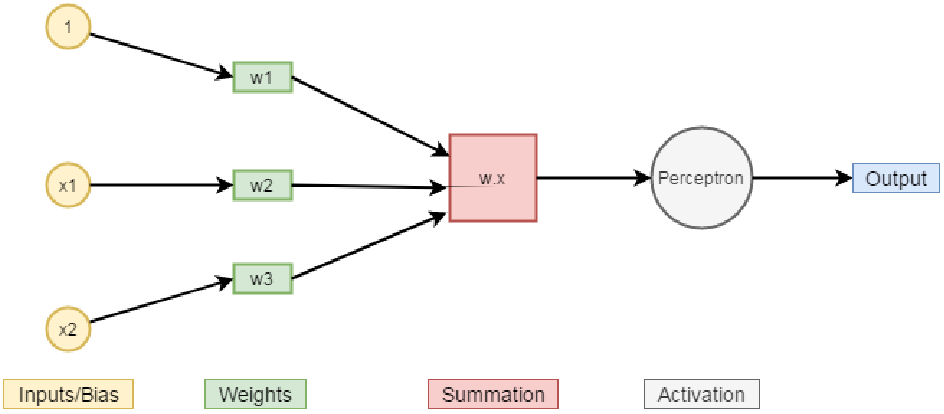
- Students will be able to implement OR and AND logic functions using Perceptron Network.
- Students will be able to plot error to visualize learning process of perceptron network.

THEORY:

The perceptron is made up of the following parts:

- Input values/One input layer
- Weights and Bias
- Net Sum
- Activation Function

These are shown in the figure given below:



The perceptron takes in a vector *x* as the input, multiplies it by the corresponding weight vector, *w*, then adds it to the bias, *b*. The result is then passed through an activation function. The weights are used to show the strength of a particular node. The value of the bias will allow you to shift the curve of the activation function either up or down. The activation function will help you to map input between the values that are required, for example, (-1, 1) or (0, 1).

Note that a perceptron can have any number of inputs but it produces a binary output. But how do you take many inputs and produce a binary output? First, each input is assigned a weight, which is the amount of influence that the input has over the output. To determine the activation for the perceptron, we check whether the weighted sum of each input is below or above a particular threshold, or bias, *b*. If the weighted sum is equal to or less than the threshold, or bias, b, the outcome becomes 0. If the weighted sum is greater than the threshold, or bias, b, the output becomes 1. This formula is referred to as *Heaviside step function* and it can be written as follows:

$$\mathbf{f(x) = \{x<=b: 0, x>b: 1\}}$$

Where *x* is the weighted sum and *b* is the bias.

The Perceptron Algorithm is used to solve problems in which data is to be classified into two parts. Because of this, it is also known as the *Linear Binary Classifier*.

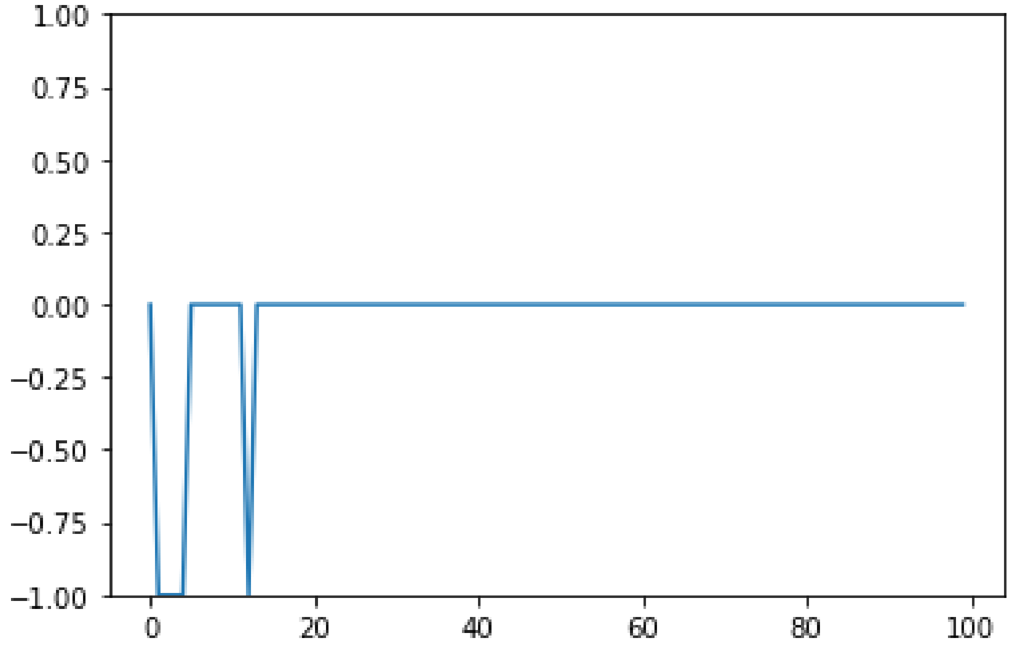
The Perceptron algorithm output values can only take two possible values, 0 or 1. The Perceptron can only be used to classify linear separable vector sets. According to the perceptron convergence theorem, the perceptron learning rule guarantees to find a solution within a finite number of steps if the provided data set is linearly separable. Input vectors are said to be linearly separable if they can be separated into their correct categories using a straight line/plane. If the input vectors aren’t linearly separable, they will never be classified properly.

PROGRAM:

```
from numpy import array, random, dot
from random import choice
from pylab import ylim, plot
from matplotlib import pyplot as plt
step_function = lambda x: 0 if x < 0 else 1
training_dataset = [
    (array([0,0,1]), 0),
    (array([0,1,1]), 1),
    (array([1,0,1]), 1),
    (array([1,1,1]), 1),
]
weights = random.rand(3)
error = []
learning_rate = 0.2
n = 100
for j in range(n):
    x, expected = choice(training_dataset)
    result = dot(weights, x)
    err = expected - step_function(result)
    error.append(err)
    weights += learning_rate *err *x
for x, _ in training_dataset:
    result = dot(x, weights)
    print('{}: {} -> {}'.format(x[:2], result, step_function(result)))
ylim([-1,1])
plot(error)
plt.show()
```

OUTPUTS:

```
[0 0]: -0.01196744749958073 -> 0
[0 1]: 0.9721261952652205 -> 1
[1 0]: 0.9823208215068193 -> 1
[1 1]: 1.9664144642716206 -> 1
```



STUDENT INSTRUCTIONS:

1.Execute above program for OR logic function.

2. Then make changes in the program and implement AND logic function and plot error for that function. Attach output for assessment with experiment write-up.

3. Solve exercises mentioned at the end of the experiment.

CONCLUSION:

The perceptron algorithm is implemented using python, it is the simplest form of artificial neural networks. It can be used to create a single Neuron model to solve binary classification problems. The perceptron takes in a vector *x* as the input, multiplies it by the corresponding weight vector, *w*, then adds it to the bias, *b*. The output is then passed through an activation function to map the input between the required values.

TEXT/REFERENCE BOOKS:

- "Neural Network a Comprehensive Foundation" By Simon Haykin
- "Introduction to Soft Computing" By Dr. S. N. Shivnandam, Mrs. S. N. Deepa
- "Neural Network: A classroom Approach" By Satish Kumar
- "Neural Network, Fuzzy Logic and Genetic Algorithms" By Rajshekharan S, Vijayalakshmi Pai
- "Neural Network Design" by Hagan Demuth, Beale"Neural Network for Pattern Recognition", Christopher M. Bishop

WEB ADDRESS (URLS):

- https://sdsclub.com/the-complete-guide-to-perceptron-algorithm-in-python
- nptel.ac.in/courses/117101002/downloads/Lec23.pdf
- https://towardsdatascience.com/mcculloch-pitts-model-5fdf65ac5dd1
- https://en.wikipedia.org/wiki/Artificial_neuron

EXERCISES:

1. Implement OR function using MP neuron (take binary data).

2. Implement AND function using MP neuron (take binary data).

--------------------------------x------------------------------x---------------------------------