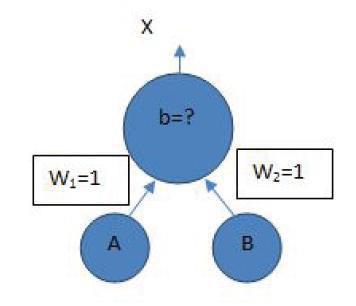
**Assignment No. 5**

**Titile**: Implement 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. To learn single layer perceptron.
2. To learn Boolean logic implementation using perceptron.

**Software Requirements**:

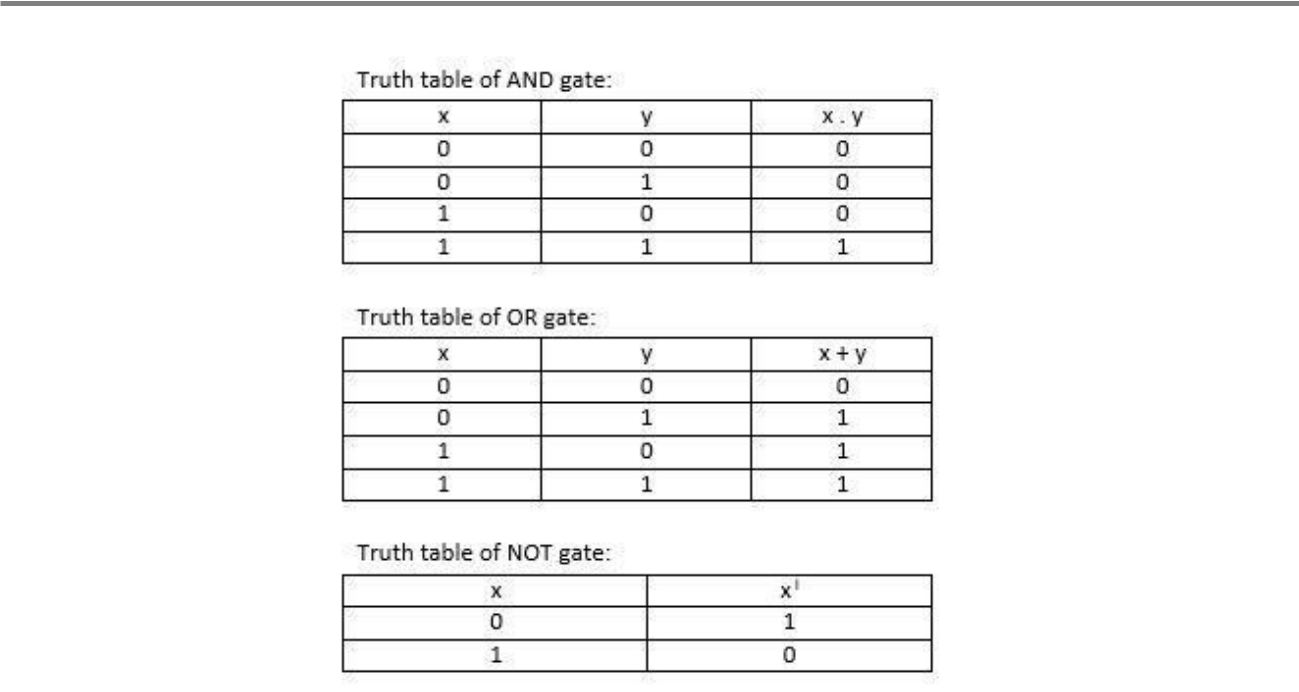
* Ubuntu 16.04

**Hardware Requirements**:

* Pentium IV system with latest configuration

**Theory:**

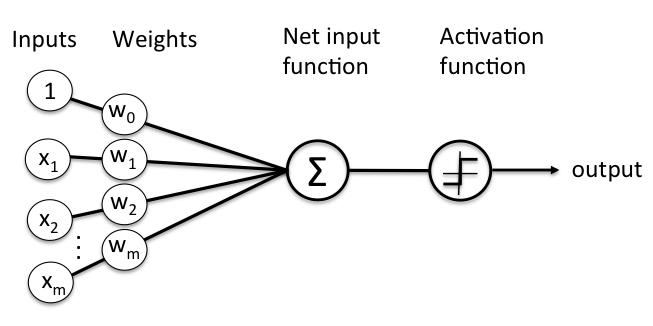
The most common Boolean functions are AND, OR, NOT. The Boolean logic AND only returns 1 if both inputs are 1 else 0, Boolean logic OR returns 1 for all inputs with 1, and will only return 0 if both input is 0 and lastly logic NOT returns the invert of the input, if the input is 0 it returns 1, if the input is 1 it returns 0. To make it clear the image below shows the truth table for the basic Boolean Function.

****

The columns X and Y are the inputs and last is the output. So, for the inputs X = 0, Y = 0 the output is 0.

**Perceptron**

A perceptron is the basic part of a neural network. A perceptron represents a single neuron on a human**’**s brain, it is composed of the dataset ( Xm ) , the weights ( Wm ) and an activation function, that will then produce an output and a bias.



The datasets (inputs) are converted into an ndarray which is then matrix multiplied to another ndarray that holds the weights. Summing up all matrix multiply and adding a bias will create the net input function, the output would then passed into an activation function that would determine if the neuron needs to fire an output or not.

Most common activation function used for classification used is a sigmoid function, which is a great function for classification (Although sigmoid is not the leading activation function for middle

layers of neural networks, it still is widely used for final classifications). The perceptron is simply separating the input into 2 categories, those that cause a fire, and those that don't. It does this by looking at (in the 2-dimensional case):

w1I1 + w2I2 < t

If the LHS is < t, it doesn't fire, otherwise it fires. That is, it is drawing the line:

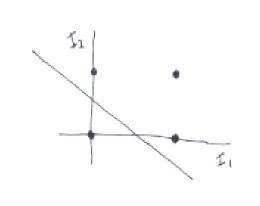
w1I1 + w2I2 = t

and looking at where the input point lies. Points on one side of the line fall into 1 category, points on the other side fall into the other category. And because the weights and thresholds can be anything, this is just *any line* across the 2 dimensional input space.

So what the perceptron is doing is simply drawing a line across the 2-d input space. Inputs to one side of the line are classified into one category, inputs on the other side are classified into another. e.g. the OR perceptron, w1=1, w2=1, t=0.5, draws the line:

I1+I2=0.5

across the input space, thus separating the points (0,1),(1,0),(1,1) from the point (0,0):



As you might imagine, not every set of points can be divided by a line like this. Those that can be, are called *linearly separable*.

In 2 input dimensions, we draw a 1 dimensional line. In n dimensions, we are drawing the (n-1)

dimensional *hyperplane*:

w1I1 + .. + wnIn = t

**Conclusion**

Thus we learned implementation of single layer perceptron for AND, OR and NOT Boolean function.