Criterion A: Planning

Background:

As I learned logic gates in computer science, which means people give two inputs(most of the time, except for NOT gate, which only takes one argument) , 1 or 0, to the computer and it gives out 1 or 0, meaning True or False, and as what I learned, the computer uses voltage level to represent 1 and 0. What I want to do is to create a program to perform the function of AND gate and OR gate.

Rationale:

This product does not simply use conditional sentence, which might be the first thought to solve this problem. Instead, it uses perceptron, the algorithm in machine learning. The advantage of this is that it can be extend to all Binary problem and it allows slight error for the input. Besides, it has the graphic interface which allows users to tell the dividing line visually.

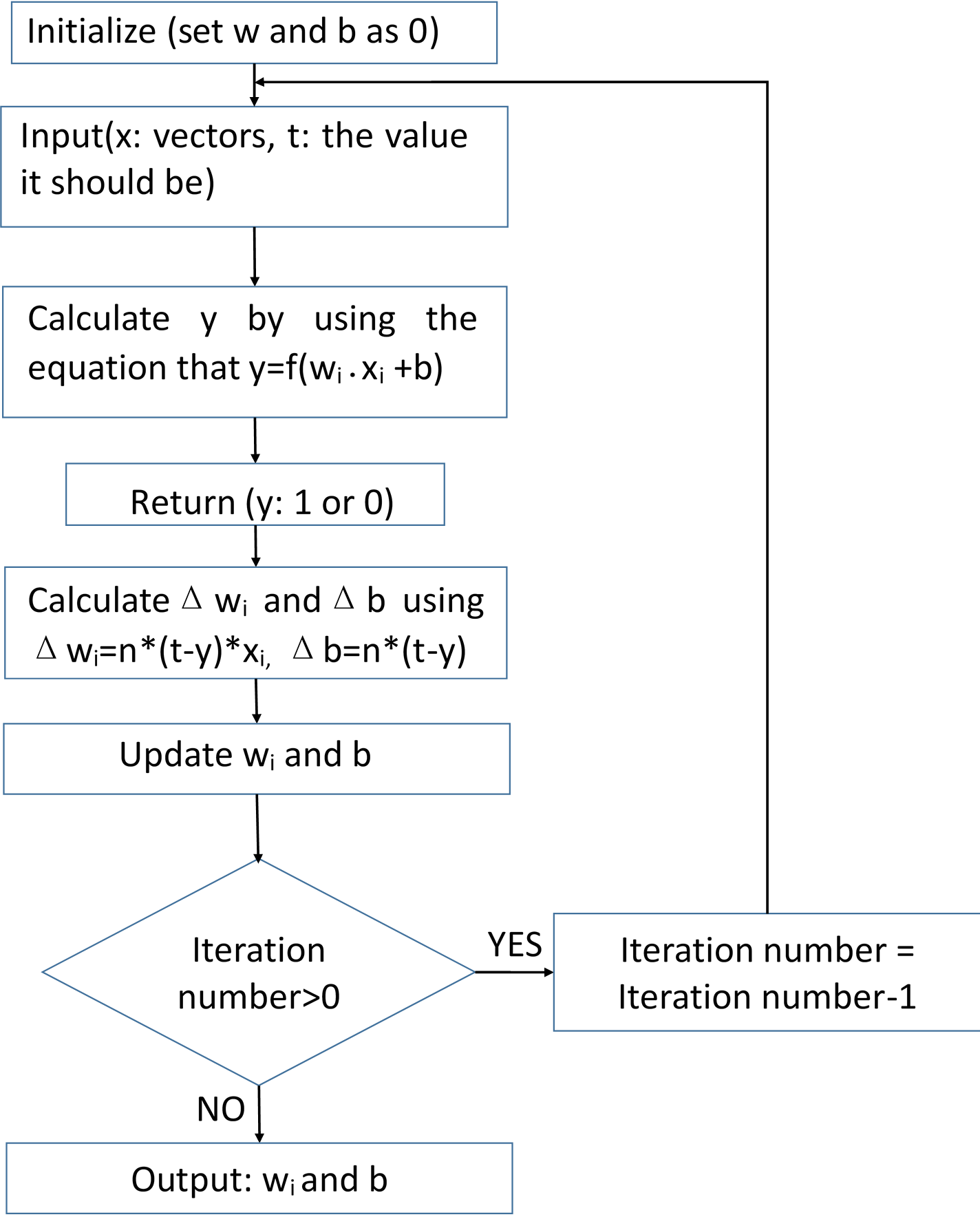
Success Criteria:

1. It can correctly show the answer for the AND gate and OR gate
2. It can be extended to other kinds of binary questions
3. It creates a graph for user to find the diving line

Criterion B: Design

Basically, it contains two part, the first part is the core perception, which calculate the weight for each item in the vector and the value of the bias which determine the value overall by training from data containing vectors and labels. The training part is to update the value of weight and bias by iterating the same set of vectors for several times and using the formula of gradient descent that Δwi=n\*(t-y)\*xi, Δb=n\*(t-y) to update. In this formula, n stands for the studying rate, t stands for the labelled value, which means the value it should be and y stands for the value we get through the calculation using the weight and bias we get so far. The flow chart below shows the process of training weight and bias. After this training, we could predict the value of a vector by the weight and bias we already know, and this would give us a value of one or zero.

The second part of this product is application. First of all, I draw the graph of some random points in the plane and find the dividing line of it which can be used to check the validity of the calculation. Plus, I wrote two functions to perform AND gate and OR gate, which will give us the truth table.



Criterion C: Development

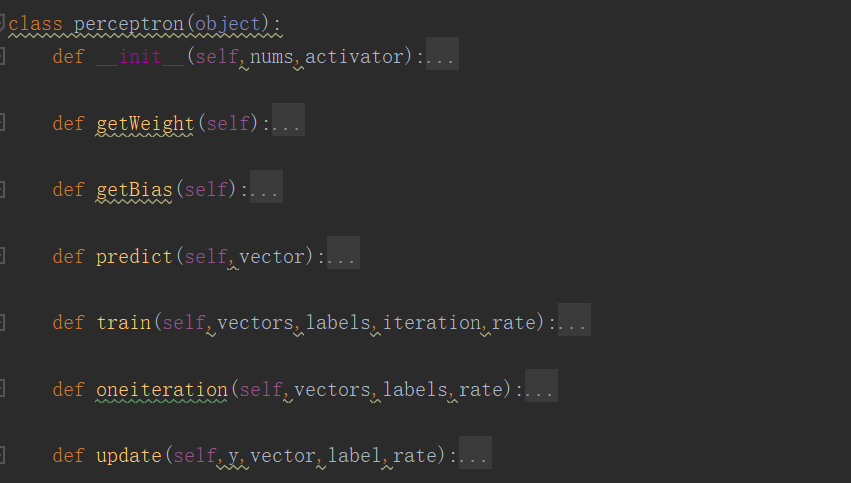
The structure of this product is basically a class and two functions, in the middle there are some sentences for drawing the graph.

For the class, perceptron, it needs two parameters, which are the dimensionality of the vector, which is an integer and the activator function, for example, it could be f(x), when x>0, it returns 1 and else, it returns 0.

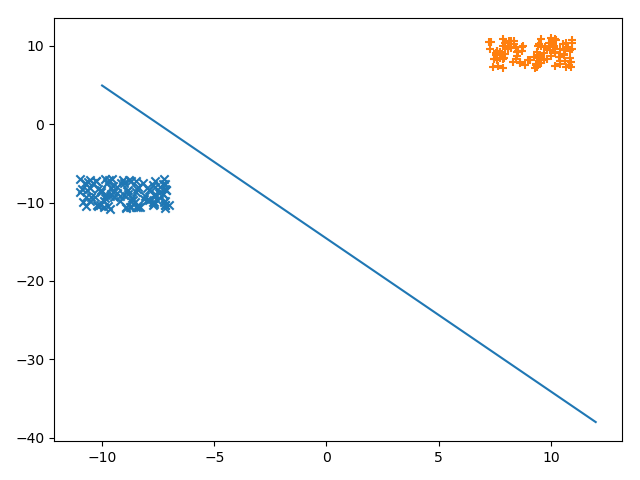
Then it has two functions which allow users to get the value of weight and bias.

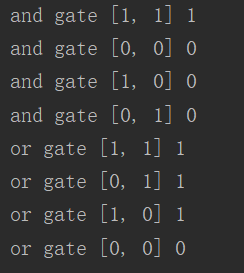
After this, there is a predict function which will calculate the value for the vector inputs using the weight and bias from what we trained.

Moreover, it is the training part. It could be divided into 3 parts. The first one is to perform iteration, the second one is to work as one iteration to update weight and bias and the last one is how to update those two values which uses gradient descent.



For the graph part, it takes some random points and labels to train the perceptron and draw a graph showing the points and the dividing points for it. I use Matplotlib and Numpy in this part. The graph I get is as shown below. The orange part of points means one and the blue part means zero. The blue line is the dividing line.





For the two function, they are basically in the same structure.

Taking the AND gate as an example. First of all, define the activator function. Then, I train the perceptron by putting all the possibilities in the truth table and iterate it for 10 times. Then, I use this trained perceptron to predict the value of [1,0],[1,1],[0,1],[0,0].

The output is as shown and the same process for the OR gate.

Criterion E: Evaluation

The final product met most of the success criteria. This program solve the problem I proposed at the beginning that it can provide a perceptron to be trained using data with labels and iteration. As putting the truth table of AND gate and OR gate, it can help to illustrate the AND gate and OR gate. Besides, it provides a graphic interface to demonstrate the answer. In addition, it can solve all binary problems if the user provide training data to train the perceptron.

However, this program does not allow user to interact with it, needing to improve. Plus, the outcome of the graph is not stable that the dividing line sometimes doesn’t divide two parts of the points correctly that it may passes one part of the points.