**Computer homework**

**3.5** the data is in the following table：



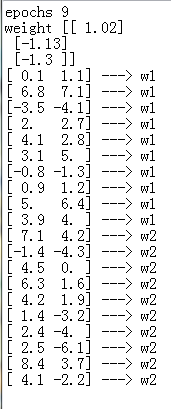
Write a program to implement the Perceptron algorithm.

(a) Starting with a = 0, apply our program to the training data fromand. Note the number of iterations required for convergence.

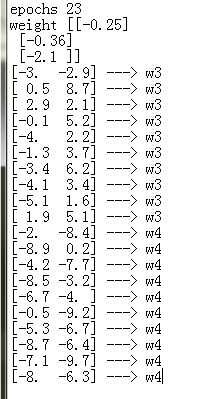
(b) Apply our program toand . Again, note the number of iterations required for convergence.

(c) Explain the difference between the iterations required in the two cases.

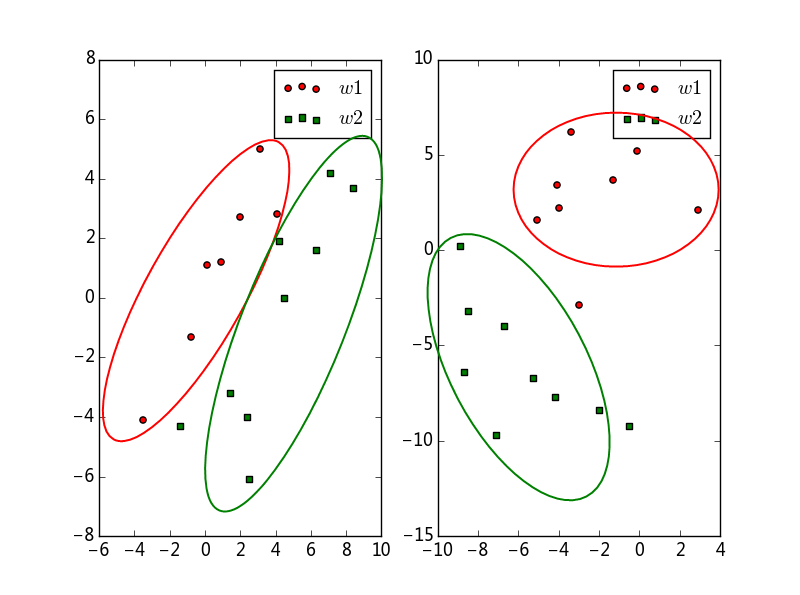
1. **When we start with a=0 forand, we finally can get a = [1.02,-1.03], b = -1.3. We also can find our algorithm would be convergent after about 9 iterations. The result as shown down.**

****

1. **When we start with a=0 forand, we finally can get a = [-0.25.-0.36], b = -2.1. We also can find our algorithm would be convergent after about 23 iterations. The result as shown down.**



1. **According to the scatter figure, we can find the distribution ofandis more tight, and the distribution of andis more scattering. So we need less iteration can achieve convergence for and, and 、 needs more.**



**Computer homework**

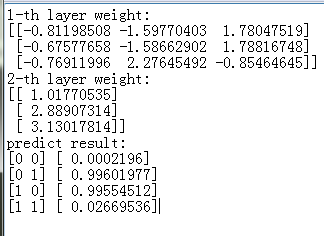
**4.6** Write a backpropagation program for a 2-2-1 network with bias to solve the XOR problem. Show the input-to-hidden weights and analyze the function of each hidden unit.

**According the table:**

|  |  |
| --- | --- |
| Input\_data | Output\_target |
| [0,0] | 0 |
| [0,1] | 1 |
| [1,0] | 1 |
| [1,1] | 0 |

**We design the 2-2-1 MLP, and we add a bias in input layer and hidden layer. And we set the learning rate is 0.2, 5000 iterations. Finally we output the weights and test the network. As the figure shown, the 1th-layer’s weight is  (*note: we add a bias in input layer and hidden layer, so the matrix is* 3x3), the 2th-layer’weight is .**

**In our test, we find the real output is  when our input is  , the result is very close expected output . So we think our backpropagation algorithm is effective.**



**Computer homework**

The inner-product kernel for a polynomial learning machine used to solve the XOR problem is defined by



What is the minimum value of power *p* for which the XOR problem is solved? Assume that *p* is a positive integer. What is the result of using a value for *p* larger than the minimum?

**We apply Platt SMO algorithm to achieve SVM and solve this problem. And we set *p* = 1,2,3,….**

**Finally, we find the minimum p is 3. According our result as shown down, we can find the error rate is 50%, the [0,0], [1,1] samples is falsely predicted when *p* = 1. And we also can find the error rate is 75% when *p* = 2, [0,0],[1,0],[0,1] have a wrong result. But the predict result will be all right when *p* >=3. For the reason. We think XOR is linear unseparable problem. And quadratic function or linear function only can solve a linear unseparable problem. So we need *p* >=3.**

