淮阴工业大学

实验报告

课程名称： 神经网络与深度学习

实验项目： Rosenblatt 感知器模型的实际应用

专业班级： 智能科学与技术 1801

姓　　名： 陈智深　　**学　　号：** 180407109

实验室号：　　**实验组号：**

实验时间： 2020.9.22　**批阅时间：**

指导教师：　　**成　　绩：**

沈阳工业大学实验报告

（适用计算机程序设计类）

专业班级：智能 1801　　学号：180407109　　姓名：陈智深

实验名称：Rosenblatt 感知器模型的实际应用

1.实验目的：

1、 利用 Rosenblatt 感知器实现模式分类（线性可分）；

2、 说明 Rosenblatt 感知器算法对线性可分模式正确分类的能力，并说明当线性可分性不满足时 Rosenblatt 感知器会崩溃。

要求复习 Rosenblatt 感知器及其学习算法等内容。

2.实验原理：

感知器示意图如下图：

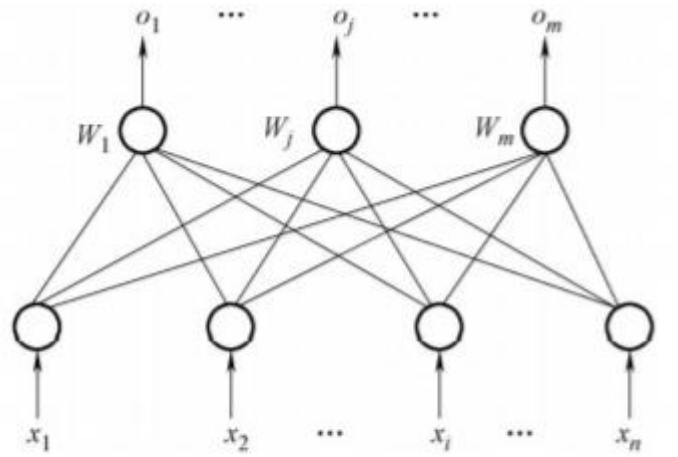


图 1.1　感知器示意图

模型输入、输出、权值参数如下：

$$\boldsymbol{X}=\left(x_1,x_2,\cdots x_i,\cdots,x_n\right)^T$$

$$\boldsymbol{O}=\left(o_1,o_2,\cdots o_j,\cdots,o_m\right)^T$$

$$\boldsymbol{W}_j=\left(w_{1j},w_{2j},\cdots w_{ij},\cdots,w_{nj}\right)^T \quad j=1,2,\cdots,m$$

$$o_j=sgn(net_j^{'}-T_j)=sgn(\sum_{i=0}^{n}w_{ij}x_i)=sgn(W_j^TX)$$

感知器的学习算法步骤：

（1）对各权值 $w_{0j}(0)$，$w_{1j}(0)$，…，$w_{nj}(0)$，$j=1,2,\cdots,m$（m 为计算层的节点数）赋予较小的非零随机数。

（2）输入样本对 $\{\boldsymbol{X}^p, \boldsymbol{d}^p\}$，其中 $\boldsymbol{X}^p=(-1, x_1^p, x_2^p, \cdots, x_n^p)$，$\boldsymbol{d}^p=(d_1^p, d_2^p, \cdots, d_m^p)$ 为期望的输出向量（教师信号），上标 p 代表样本对的序号，设样本集中的样本总数为 P，则 $p=1,2,\cdots,P$。

（3）计算各节点的实际输出 $o_j^p(t)=\mathrm{sgn}[\boldsymbol{W}_j^{\mathrm{T}}(t)\boldsymbol{X}^p]$，$j=1,2,\cdots,m$。

（4）调整各节点对应的权值，$\boldsymbol{W}_j(t+1)=\boldsymbol{W}_j(t)+\eta[d_j^p-o_j^p(t)]\boldsymbol{X}^p$，$j=1,2,\cdots,m$，其中 η 为学习率，用于控制调整速度，η 值太大会影响训练的稳定性，太小则使训练的收敛速度变慢，一般取 $0<\eta\leqslant 1$。

（5）返回到步骤（2）输入下一对样本。

以上步骤周而复始，直到感知器对所有的样本的实际输出与期望输出相等。

应用上述学习算法，当被分开的模式是线性可分时，即能用一个超平面将两类输入模式分隔开时，感知器就可以通过有限次的学习，学会正确分开两类模式，知识感知器的收敛定理。

3.实验内容

1、二维平面上的两类模式，如图 1.2 及表 1.1 所示。

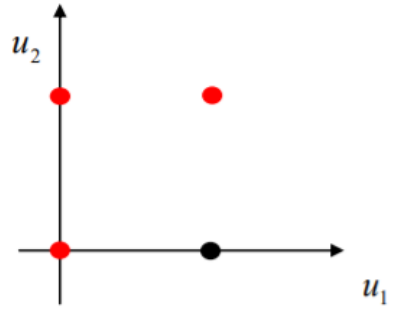


图 1.2　二维平面的两类模式示意图

表 1.1　二维平面的两类模式

u_1	u_2	y
0	0	0
0	1	0
1	1	0
1	0	1

根据权重系数的迭代方法，用你自己熟悉的编程语言（C、Matlab、C++、Python 等）实现其分类直线的求取。

2、二维平面上的两类模式，如图 1.3 及表 1.2 所示。

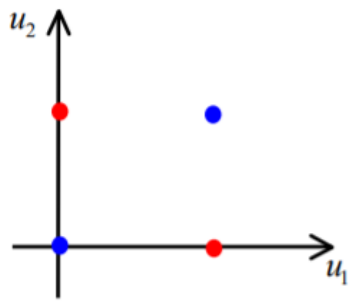


图 1.3　二维平面的两类模式示意图

表 1.2　二维平面的两类模式

u_1	u_2	y
0	0	0
0	1	1
1	0	1
1	1	0

沈阳工业大学实验报告

（适用计算机程序设计类）

专业班级：智能 1801　　学号：180407109　　姓名：陈智深

根据权重系数的迭代方法，用你自己熟悉的编程语言（C、Matlab、C++、Python 等）实现其分类直线的求取。

3、通过上述上机，验证感知机的收敛定理。

4.实验步骤或程序（经调试后正确的源程序）

```
clc,clear,close;

repeat = 10;%最大计算量
count = 0;%迭代次数

%绘制数据点
sample_X = [-1 0 0;
            -1 0 1;
            -1 1 1;
            -1 1 0];
lable_d = [1 1 1 -1]';                  %代价
%lable_d = [-1 1 -1 1]';                %线性不可分代价
date2 = lable_d';                       %转置矩阵后用于之后比较
figure(1);
draw(sample_X(:,2:3) ,date2);           %画目标点函数
%初始化 w，b，alpha
w = [0.42,0.38,0.44];
b = 0.06;
alpha = 0.15;                           %学习率

x1 = -1:0.01:10;
%缓存替代、初始化退出标志
buff = zeros(1,4);
flag = 0;

% while 1
while flag == 0
    for i = 1:4
        net = w * sample_X(i,:)';       %净输入
        %符号函数
        if net > 0
            o=1;
        elseif net == 0
            o = 0;
        else
            o=-1;
        end
```

```
            buff(i) = o;                                %记录一轮的实际输出
            %数据更迭
            w = w' + alpha * (lable_d(i,:) - o) * sample_X(i,:)';
            w = w';
            w(1) = w(1) + b * (lable_d(i,:) - o);
            %实时划线测试用
%               b = b + b*(lable_d(i) - o);
%               x2 = (-w(2)*x1+w(1))/w(3);
%               hold on;
%               plot(x1,x2);
%               pause(1);
        end
         %退出判定
        if date2 == buff                            %感知器收敛要求
            flag = 1;
        end

        count = count + 1;
end
%最后结果画线
x2 = (-w(2)*x1+w(1))/w(3);
hold on;
axis([0 2 -1 2]);                                   %坐标轴确定
plot(x1,x2);
%输出显示
text = ['经过',num2str(count-1),'次迭代后收敛'];
disp(text);
fprintf('函数为：y=(%.4f*x-%d)/%.4f\n',w(2),b,w(3));
fprintf('权重为：%.4f    %.4f\n',w(2),w(3));
```

函数 draw 详细：

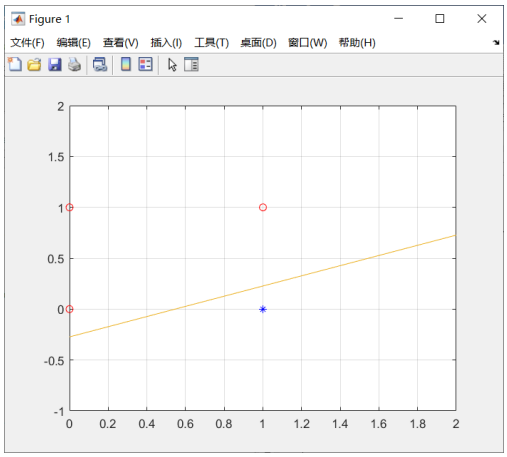
```
function draw(sample,label)                         %输入参数为坐标点和期望输出
    %区分两种不同的输出的点
    idx_pos = find(label==1);
   idx_neg = find(label~=1);
   %使用两种不同的方式画出要求的点
    plot(sample(idx_pos, 1), sample(idx_pos, 2),'ro')
    hold on
    plot(sample(idx_neg, 1), sample(idx_neg, 2),'b*')
    axis([0 2 0 2])
    grid on
end
```

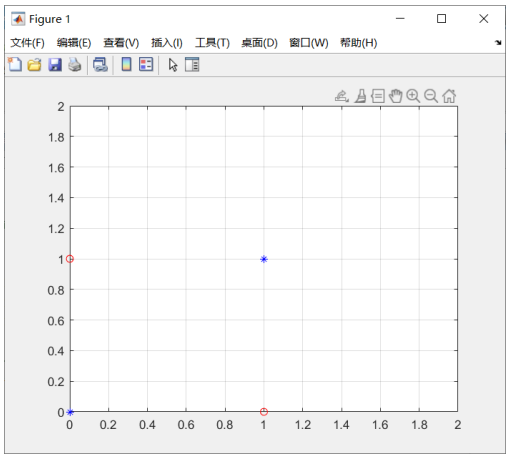
沈阳工业大学实验报告

（适用计算机程序设计类）

专业班级：智能 1801　　学号：180407109　　姓名：陈智深

5.程序运行结果





经过5次迭代后收敛
函数为：y=(-0.2200*x-6.000000e-02)/0.4400
权重为：-0.2200 0.4400

图 1.4 线性可分结果图　　图 1.5 线性可分性不满足结果图

6.结果分析和解释

在线性可分情况下，由图 1.4 可得程序在第 5 次迭代后达到期望输出和实际输出相等，达到分类的效果。函数为$y=\frac{-0.22x-0.06}{0.44}$，权重为-0.22，0.44.由此可得实验成功，感知器模型设计成功。

在线性可分性不满足的时候，实际输出和期望输出无法达到相等的输出，程序在更迭权重中，在更新权重、计算实际输出时，始终无法达到收敛的条件，导致无法跳出，即无法收敛，使得感知器崩溃。

中途强制停止，可得如图 1.5 的输出。

在进行线性可分性条件满足时，在修改不同的学习率，参数阈值等，得到的分类线也时不同的，但是相同的是，都可以达到分类的最终目的。

7.实验结论

Rosenblatt 感知器算法对线性可分模式有着正确分类的能力，但当线性可分性不满足时 Rosenblatt 感知器会崩溃，无法进行正常的分类。

所以 Rosenblatt 感知器可以应用于线性可分模式，且正确率良好。不能够处理线性可分条件不满足的情况，线性可分条件不满足时需要使用其他的分类方法。