****

模式识别

感知器算法实验报告

|  |  |
| --- | --- |
| 姓名： | 黄勉 |
| 学号： | 57118304 |

东南大学网络空间安全学院

School of Cyber Science & Engineering

Southeast University

2020年12月

### 一．问题描述

编写感知器算法程序，求下列模式分类的解向量：

ω1: {(0 0 0) T,(1 0 0)T,(1 0 1 ) T,(1 1 0) T}

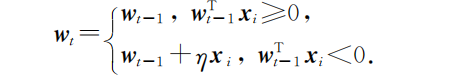
ω2: {(0 0 1)T, (0 1 1) T,(0 1 0) T, (1 1 1) T}

设w(1)=(-1 -2 -2 0)T

### 算法描述

1. 选择n个分属于ω1、ω2 两类的模式样本构成训练样本集，对ω2类样本全部乘以(-l),

任取权重向量初始值w(1),开始迭代t=1  
  
2)用全部训练样本进行一轮迭代,计算ωTx;的值，并修正权重向量;



3)只要有一个错误分类,回到步骤2),直至对所有样本正确分类

### 计算过程

1. 取增量校正参数为1。
2. 通过Augmented\_vector函数将训练样本写成增广向量形式
3. 通过Iteration函数迭代，设置flag为标志位，判断是否还需要继续迭代

### 代码实现

#include<iostream>

using namespace std;

//初始化训练样本

int w1[4][3]={{0,0,0},{1,0,0},{1,0,1},{1,1,0}};

int w2[4][3]={{0,0,1},{0,1,1},{0,1,0},{1,1,1}};

int l=0;//记录W迭代次数

int t=1;//记录迭代轮数

int W[100][4]={{-1,-2,-2,0}};

int c=1;//校正参量系数

int X[8][4];//增广向量

//将训练样本写成增广向量形式

void Augmented\_vector(int a[4][3],int b[4][3])

{

for(int i=0;i<4;i++){

for(int j=0;j<3;j++){

X[i][j]=a[i][j];

X[i][3]=1;

}

}

for(int i=4;i<8;i++){

for(int j=0;j<3;j++){

X[i][j]=-b[i-4][j];

X[i][3]=-1;

}

}

/\* for(int i=0;i<8;i++){

for(int j=0;j<4;j++){

cout<<X[i][j];

}

cout<<endl;

} \*/

}

//判别迭代

void Iteration()

{

cout<<"第"<<t<<"轮迭代："<<endl;

int flag=0;

for(int k=0;k<8;k++){

int s=0;

for(int j=0;j<4;j++){

s+=W[l][j]\*X[k][j];

}

cout<<"W["<<l+1<<"]TX"<<k+1<<"="<<s;

if(s>0){

cout<<">0,故"<<endl;

cout<<"W["<<l+2<<"]=W["<<l+1<<"]=";

for(int j=0;j<4;j++){

W[l+1][j]=W[l][j];

cout<<W[l+1][j]<<" ";

}

cout<<endl;

}

else{

cout<<"<=0,故"<<endl;

cout<<"W["<<l+2<<"]=W["<<l+1<<"]+c\*X["<<k+1<<"]=";

for(int j=0;j<4;j++){

W[l+1][j]=W[l][j]+c\*X[k][j];

cout<<W[l+1][j]<<" ";

}

cout<<endl;

flag++;

}

l++;

}

t++;

if(flag>0)

Iteration();

else{

cout<<"迭代次数："<<t-1<<endl;

cout<<"解向量为"<<endl;

for(int j=0;j<4;j++){

cout<<" "<<W[l][j];

}

}

}

int main()

{

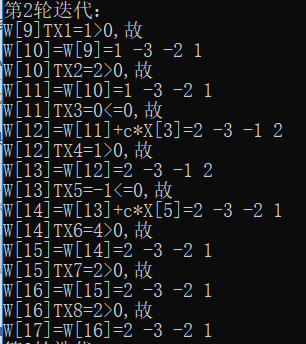
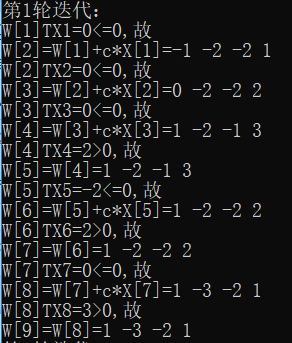
Augmented\_vector(w1,w2);

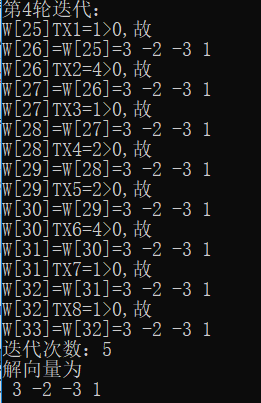
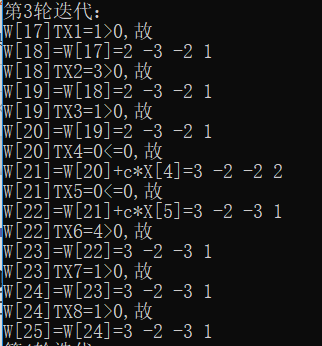
Iteration();

return 0;

}

### 结果分析







感知器算法是一种赏罚过程：当分类器发生分类错误时，对分类器进行“罚”——修改权向量；分类正确时，对分类器进行“奖”——即权向量不变。

感知器算法是收敛的。只要模式类别是线性可分的，就可以在有限的迭代步数中求出权向量的解。