劉波洲 0540153

**Question 1：**

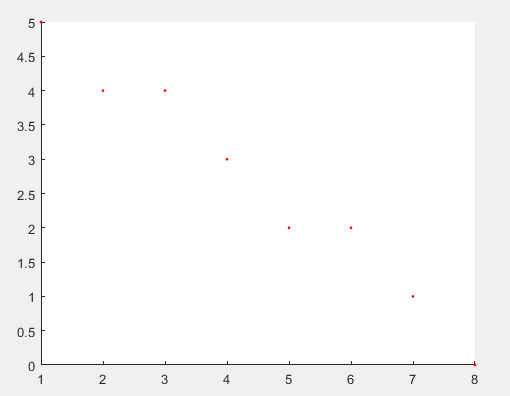
1. use multi-class linear discriminant functions to find the discriminant functions, the linear decision boundaries, and regions。 在(-1, -1) (2, -1) (2, 2) (-1, 2) 四點所形成的四方形的區域內，產生 50x50 (或 100x100) 的格子點，將每一點代入 model 做分類，得區域的分類圖。
2. 共分4類，4個linear discriminant functions對應4個weight向量，均初始化為0。每次learning，用maximum selector選出最大的，與期望的輸出比較警醒調整。

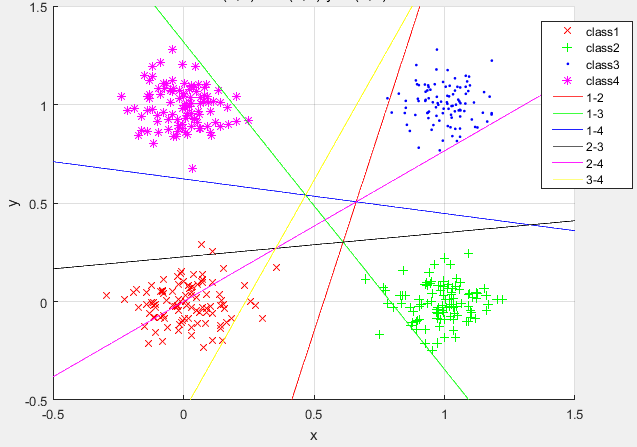
Learning rule：自己的輸出值不是最大，自己的weight值調整(weight=weight+ay)；別人的輸出值比自己大，別人的weight要調整(weight=weight-ay;別人的輸出值自己小，不用調整。計算error，檔error等於0或者iteration次數大於5000次時停止學習。最後用格子點帶入model做分類得到分類圖。

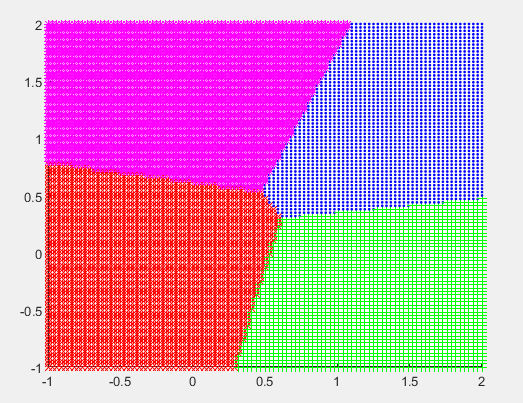
Flowchart：



1. Result：經過8次iteration到達error等於0，後面的decision boundarie也能區分4類樣本點。效果還行。







1. 分析與討論

分割現及分割區域沒有交與一點，如果是分三類的話，最後分割綫會交與一點。

1. 參考文獻

1.張智星 MATLAB程式設計：入門篇

MATLAB程式設計：進階篇

2.Matlab幫助文檔

（6） matlab program：

%(a)

%use multi-class linear discriminant functions to find the discriminant functions, the

%linear decision boundaries, and regions¡£

clear

%--------generate training data-----------%

mu=[0 0]';

sigma=[0.01 0;0 0.01];

data(:,:,1)=mvnrnd(mu,sigma,100);

mu=[1 0]';

sigma=[0.01 0;0 0.01];

data(:,:,2)=mvnrnd(mu,sigma,100);

mu=[1 1]';

sigma=[0.01 0;0 0.01];

data(:,:,3)=mvnrnd(mu,sigma,100);

mu=[0 1]';

sigma=[0.01 0;0 0.01];

data(:,:,4)=mvnrnd(mu,sigma,100);

%---------training------------------%

weight=[0 0 0 0;0 0 0 0;0 0 0 0];%3x4

iteration=0;%max=5000

error=1;%max=100,min=0.0

alpha=1.0;

while error>0 && iteration<5000

error=0;

iteration=iteration+1;

%class1

for c=1:4

for n=1:100

y=[data(n,1,c) data(n,2,c) 1];%1x3

out=y\*weight;%1x4

flag=0;

for t=1:4

if c==t

continue;

elseif out(1,c)<=out(1,t)

flag=1;

weight(:,t)=weight(:,t)-alpha.\*y';

end

end

if(flag==1)

weight(:,c)=weight(:,c)+alpha.\*y';

error=error+1;%

end

end

end

hold on

plot(iteration,error,'r.');

end

% hold off

% figure(2)

% hold on

% plot(data(:,1,1),data(:,2,1),'rx')%class1

% plot(data(:,1,2),data(:,2,2),'g+')%class2

% plot(data(:,1,3),data(:,2,3),'b.')%class3

% plot(data(:,1,4),data(:,2,4),'m\*')%class4

% for m=1:4

% fh=@(x,y) weight(1,m)\*x+weight(2,m)\*y+weight(3,m);

% ezplot(fh,[-2,3])

% hold on

% end

% hold off

figure(3)

hold on

plot(data(:,1,1),data(:,2,1),'rx')%class1

plot(data(:,1,2),data(:,2,2),'g+')%class2

plot(data(:,1,3),data(:,2,3),'b.')%class3

plot(data(:,1,4),data(:,2,4),'m\*')%class4

%1-2

w=weight(:,1)-weight(:,2);

fh=@(x,y) w(1,1).\*x+w(2,1).\*y+w(3,1)';

set(ezplot(fh,[-0.5,1.5]),'Color','r');

hold on

%1-3

w=weight(:,1)-weight(:,3);

fh=@(x,y) w(1,1).\*x+w(2,1).\*y+w(3,1)';

set(ezplot(fh,[-0.5,1.5]),'Color','g');

hold on

%1-4

w=weight(:,1)-weight(:,4);

fh=@(x,y) w(1,1).\*x+w(2,1).\*y+w(3,1)';

set(ezplot(fh,[-0.5,1.5]),'Color','b');

hold on

%2-3

w=weight(:,2)-weight(:,3);

fh=@(x,y) w(1,1).\*x+w(2,1).\*y+w(3,1)';

set(ezplot(fh,[-0.5,1.5]),'Color','k');

hold on

%2-4

w=weight(:,2)-weight(:,4);

fh=@(x,y) w(1,1).\*x+w(2,1).\*y+w(3,1)';

set(ezplot(fh,[-0.5,1.5]),'Color','m');

hold on

%3-4

w=weight(:,3)-weight(:,4);

fh=@(x,y) w(1,1).\*x+w(2,1).\*y+w(3,1)';

set(ezplot(fh,[-0.5,1.5]),'Color','y');

hold off

grid on

legend('class1','class2','class3','class4','1-2','1-3','1-4','2-3','2-4','3-4')

%----------test-----------%

figure(4)

for x=-1:0.03:2

for y=-1:0.03:2

out=[x y 1]\*weight;

class=1;

temp=out(1,1);

for c=2:4

if out(1,c)>temp

class=c;

temp=out(1,c);

end

end

hold on

if class==1

plot(x,y,'rx');

elseif class==2

plot(x,y,'g+');

elseif class==3

plot(x,y,'b.');

else

plot(x,y,'m\*');

end

end

end

axis on

**Question 2：**

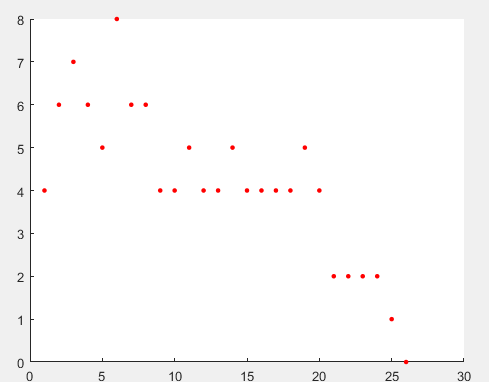
multi-class linear perceptrons to find the linear decision boundaries and regions。

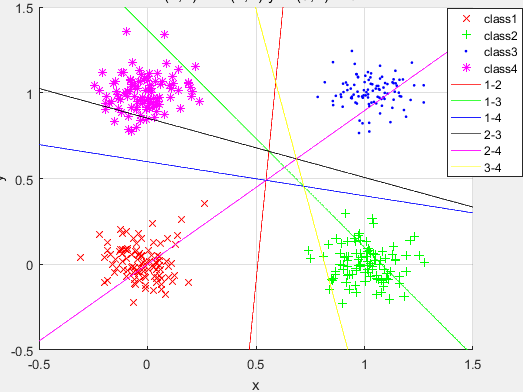
Activation functions 為 hardlimiter function。

（2）共4個perceptron.每個樣本輸入后立即調整，根據error和iteration的情況決定是否繼續調整。Flowchart

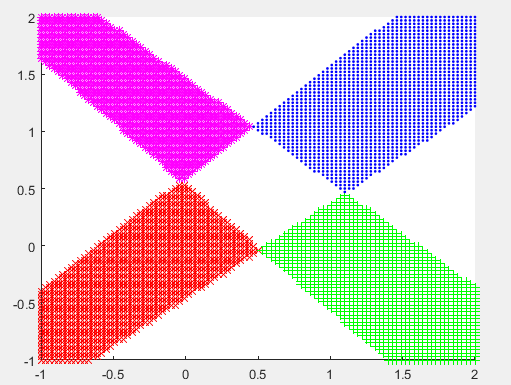


1. result：





1. 分析討論：明顯，分割綫沒有交與一點，按照第一問的方法得到分類圖，做出分類的點畫出，不能做分類的點不畫出，如下圖所示，説明還存在不能做分類的區域。改變這種情況，可增大training樣本點的範圍，如果保持樣本不變，這種情況無法改變。



（5） 參考文獻

1.張智星 MATLAB程式設計：入門篇

MATLAB程式設計：進階篇

2.Matlab幫助文檔

1. matlab program

%multi-class linear perceptrons to find the linear decision boundaries and

%regions¡£Activation functions žé hardlimiter function¡£

clear

%--------generate training data-----------%

mu=[0 0]';

sigma=[0.01 0;0 0.01];

data(:,:,1)=mvnrnd(mu,sigma,100);

mu=[1 0]';

sigma=[0.01 0;0 0.01];

data(:,:,2)=mvnrnd(mu,sigma,100);

mu=[1 1]';

sigma=[0.01 0;0 0.01];

data(:,:,3)=mvnrnd(mu,sigma,100);

mu=[0 1]';

sigma=[0.01 0;0 0.01];

data(:,:,4)=mvnrnd(mu,sigma,100);

%---------training------------------%

weight=[0 0 0 0;0 0 0 0;0 0 0 0];%3x4

iteration=0;%max=5000

error=1;%max=100,min=0.0

alpha=1.0;

while error>0 && iteration<5000

error=0;

iteration=iteration+1;

%class1

for c=1:4

for n=1:100

y=[data(n,1,c) data(n,2,c) 1];%1x3

out=y\*weight;%1x4

for m=1:4 %hardlimiter

if out(1,m)>0

out(1,m)=1;

elseif out(1,m)<0

out(1,m)=-1;

else

out(1,m)=0;

end

end

flag=0;

for m=1:4 % adjust

if m==c

if out(1,m) ~= 1

weight(:,m)=weight(:,m)+alpha.\*y';

flag=1;

end

else

if out(1,m) ~= -1

weight(:,m)=weight(:,m)-alpha.\*y';

flag=1;

end

end

end

if flag

error=error+1;

end

end

end

hold on

plot(iteration,error,'r.','MarkerSize',10);

end

hold off

figure(2)

hold on

plot(data(:,1,1),data(:,2,1),'rx')%class1

plot(data(:,1,2),data(:,2,2),'g+')%class2

plot(data(:,1,3),data(:,2,3),'b.')%class3

plot(data(:,1,4),data(:,2,4),'m\*')%class4

%1-2

w=weight(:,1)-weight(:,2);

fh=@(x,y) w(1,1).\*x+w(2,1).\*y+w(3,1)';

set(ezplot(fh,[-0.5,1.5]),'Color','r');

hold on

%1-3

w=weight(:,1)-weight(:,3);

fh=@(x,y) w(1,1).\*x+w(2,1).\*y+w(3,1)';

set(ezplot(fh,[-0.5,1.5]),'Color','g');

hold on

%1-4

w=weight(:,1)-weight(:,4);

fh=@(x,y) w(1,1).\*x+w(2,1).\*y+w(3,1)';

set(ezplot(fh,[-0.5,1.5]),'Color','b');

hold on

%2-3

w=weight(:,2)-weight(:,3);

fh=@(x,y) w(1,1).\*x+w(2,1).\*y+w(3,1)';

set(ezplot(fh,[-0.5,1.5]),'Color','k');

hold on

%2-4

w=weight(:,2)-weight(:,4);

fh=@(x,y) w(1,1).\*x+w(2,1).\*y+w(3,1)';

set(ezplot(fh,[-0.5,1.5]),'Color','m');

hold on

%3-4

w=weight(:,3)-weight(:,4);

fh=@(x,y) w(1,1).\*x+w(2,1).\*y+w(3,1)';

set(ezplot(fh,[-0.5,1.5]),'Color','y');

hold off

grid on

axis on

legend('class1','class2','class3','class4','1-2','1-3','1-4','2-3','2-4','3-4')

**Question 3：**

multi-class linear perceptrons to find the “decision regions”。

Activation functions 為 sigmoidal function。

（2）在第二問的基礎上講hardlimiter換成可微分的sigmoidal function。

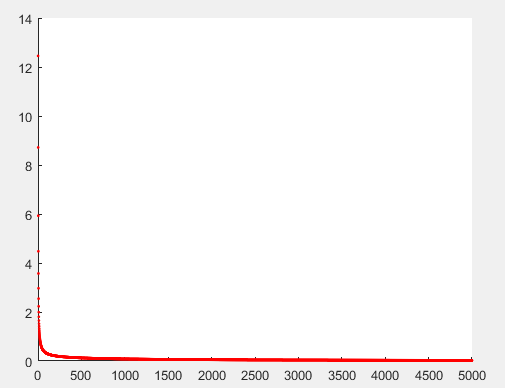
f(s)=1/(1+e^(-s))，f'(s)=f(s)(1-f(s))=1/(1+e^(-s))\*(1-1/(1+e^(-s)))

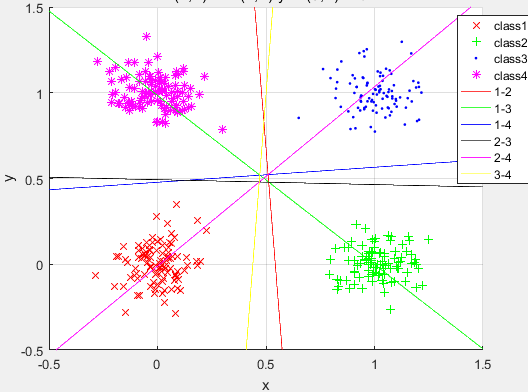
同時設置error的最低值為0.02，檔error小於0.02時，停止學習。

Flowchart



（3）result：在5000次iteration中，error并不能小於0.02，error的值為0.0257，最終是可以小於0.02的。





（4） 分析討論：通過和第二問比較，分割綫明顯的交點集中了一些，在某種程度上來説，可認爲分割綫交與一點。對比發現，sigmoidal function的效果要好，不能分辨的區域要少。

（5） 參考文獻

1.張智星 MATLAB程式設計：入門篇

MATLAB程式設計：進階篇

2.Matlab幫助文檔

（6）matlab program

%multi-class linear perceptrons to find the ¡°decision regions¡±¡£

%Activation functions žé sigmoidal function¡£

clear

%--------generate training data-----------%

mu=[0 0]';

sigma=[0.01 0;0 0.01];

data(:,:,1)=mvnrnd(mu,sigma,100);

mu=[1 0]';

sigma=[0.01 0;0 0.01];

data(:,:,2)=mvnrnd(mu,sigma,100);

mu=[1 1]';

sigma=[0.01 0;0 0.01];

data(:,:,3)=mvnrnd(mu,sigma,100);

mu=[0 1]';

sigma=[0.01 0;0 0.01];

data(:,:,4)=mvnrnd(mu,sigma,100);

%---------training------------------%

weight=[0 0 0 0;0 0 0 0;0 0 0 0];%3x4

iteration=0;

error=1;

alpha=0.5;

%f(s)=1/(1+e^(-s)),f'(s)=f(s)(1-f(s))=1/(1+e^(-s))\*(1-1/(1+e^(-s)))

while error>0.02 && iteration<5000

error=0;

iteration=iteration+1;

%class1

for c=1:4

d=[0 0 0 0];

d(c)=1;%desired output

for n=1:100

y=[data(n,1,c) data(n,2,c) 1];%1x3

s=y\*weight;%1x4

o=1./(1+exp(-s));

error=error+sum((d-o).\*(d-o)/2);

% error=error+sum(abs(d-o));

% weight(:,c)=weight(:,c)+alpha .\* (d(c)-o(c)).\*(1/(1+exp(-s(c))) ) .\*(1-1/(1+exp(-s(c)))).\*y';

for w=1:4

weight(:,w)=weight(:,w)+alpha .\* (d(w)-o(w)).\*(1/(1+exp(-s(w))) ) .\*(1-1/(1+exp(-s(w)))).\*y';

end

end

end

error=error/4;

hold on

plot(iteration,error,'r.');

end

hold off

figure(2)

hold on

plot(data(:,1,1),data(:,2,1),'rx')%class1

plot(data(:,1,2),data(:,2,2),'g+')%class2

plot(data(:,1,3),data(:,2,3),'b.')%class3

plot(data(:,1,4),data(:,2,4),'m\*')%class4

%1-2

w=weight(:,1)-weight(:,2);

fh=@(x,y) w(1,1).\*x+w(2,1).\*y+w(3,1)';

set(ezplot(fh,[-0.5,1.5]),'Color','r');

hold on

%1-3

w=weight(:,1)-weight(:,3);

fh=@(x,y) w(1,1).\*x+w(2,1).\*y+w(3,1)';

set(ezplot(fh,[-0.5,1.5]),'Color','g');

hold on

%1-4

w=weight(:,1)-weight(:,4);

fh=@(x,y) w(1,1).\*x+w(2,1).\*y+w(3,1)';

set(ezplot(fh,[-0.5,1.5]),'Color','b');

hold on

%2-3

w=weight(:,2)-weight(:,3);

fh=@(x,y) w(1,1).\*x+w(2,1).\*y+w(3,1)';

set(ezplot(fh,[-0.5,1.5]),'Color','k');

hold on

%2-4

w=weight(:,2)-weight(:,4);

fh=@(x,y) w(1,1).\*x+w(2,1).\*y+w(3,1)';

set(ezplot(fh,[-0.5,1.5]),'Color','m');

hold on

%3-4

w=weight(:,3)-weight(:,4);

fh=@(x,y) w(1,1).\*x+w(2,1).\*y+w(3,1)';

set(ezplot(fh,[-0.5,1.5]),'Color','y');

hold off

grid on

axis on

legend('class1','class2','class3','class4','1-2','1-3','1-4','2-3','2-4','3-4')