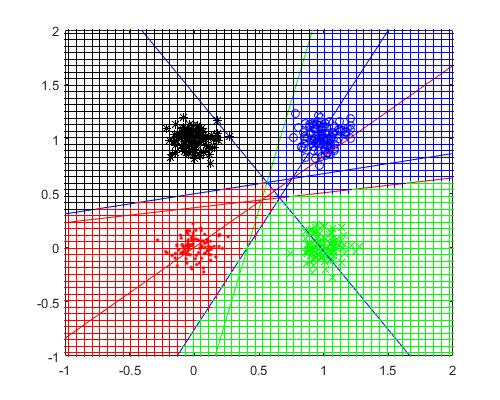
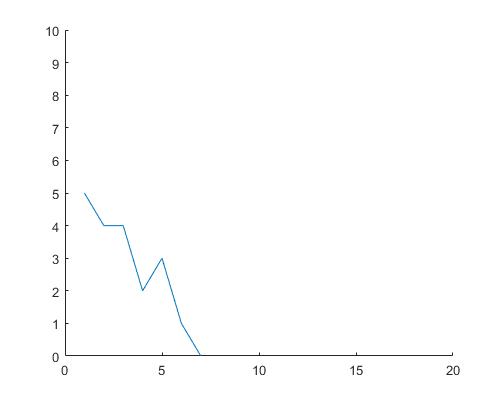
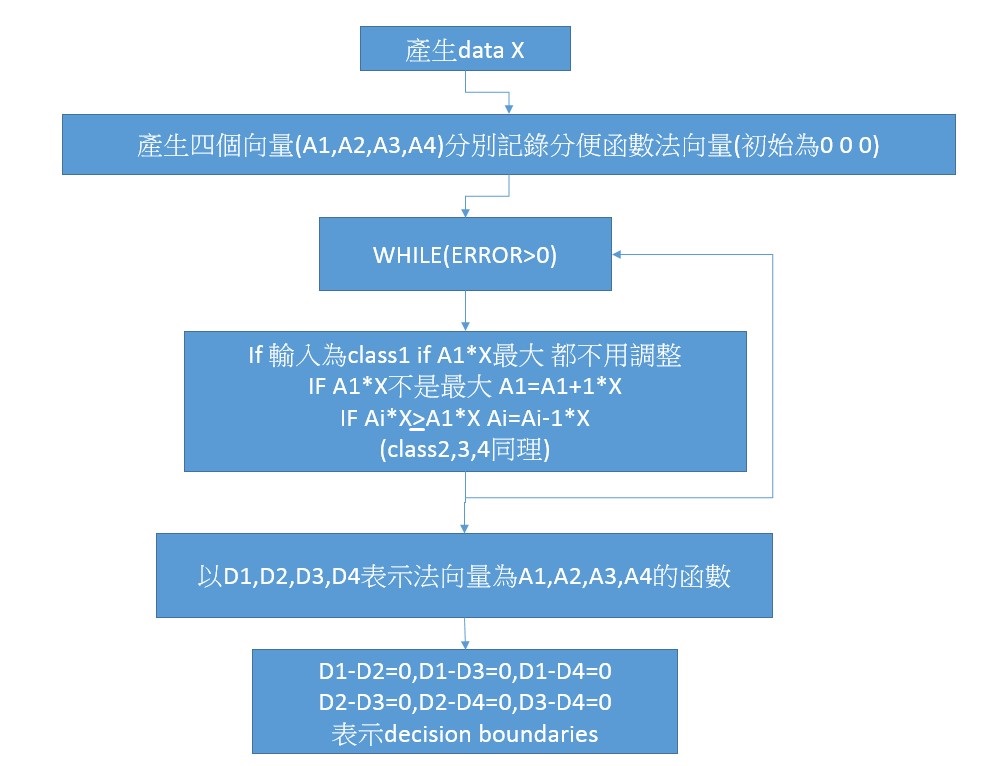
1. multi-class linear discriminant functions to find the discriminant functions, the linear decision boundaries, and regions。



Flowchart



我採取不同class分開調整的方式，1~100筆為class1，101~200為class2，201~300為class2，301~400為class2，1~100筆若A1\*X不是最大，則A1=A1+X，再判斷其他A2\*X，A3\*X，A4\*X取大於A1\*X的Ai=Ai-X。一次迴圈跑400筆data，error在每次調整向量的地方加一，當error=0結束。

此處很明顯分割線與分割區域沒有交於一點，根據課本分析當分割區塊>3後就不一定交於一點。

上表\*表示內積(忘了要用內積表示)

MATLAB CODE

hFig = figure(1);

set(hFig,'name','multi-class linear discriminant functions to find the discriminant functions, the linear decision boundaries, and regions¡C','Position', [300 0 500 400]);

Mean1 = [0;0];

Mean2=[1;0];

Mean3=[1;1];

Mean4=[0;1];

MySigma = [0.01 0; 0 0.01];

NormalPoint1 = mvnrnd(Mean1, MySigma, 100);

NormalPoint2 = mvnrnd(Mean2, MySigma, 100);

NormalPoint3 = mvnrnd(Mean3, MySigma, 100);

NormalPoint4 = mvnrnd(Mean4, MySigma, 100);

L=plot(NormalPoint1(:,1),NormalPoint1(:,2),'.','MarkerEdgeColor','r');

hold on;

M=plot(NormalPoint2(:,1),NormalPoint2(:,2),'x','MarkerEdgeColor','g');

N=plot(NormalPoint3(:,1),NormalPoint3(:,2),'o','MarkerEdgeColor','b');

O=plot(NormalPoint4(:,1),NormalPoint4(:,2),'\*','MarkerEdgeColor','k');

axis([-1 2 -1 2]);

alpha=1;

A1=[0 0 0]';

A2=[0 0 0]';

A3=[0 0 0]';

A4=[0 0 0]';

error=1;

it=0;

while(error>0)

it=it+1;

error=0;

k=0;

for i=1:100

X=[NormalPoint1(i,1) NormalPoint1(i,2) 1]';

D1=A1'\*X;

D2=A2'\*X;

D3=A3'\*X;

D4=A4'\*X;

if (D1>D2 && D1>D3) && D1>D4

continue ;

end

error=error+1;

A1=A1+X;

if D2>=D1

A2=A2-alpha\*X;

end

if D3>=D1

A3=A3-alpha\*X;

end

if D4>=D1

A4=A4-alpha\*X;

end

end

for i=1:100

X=[NormalPoint2(i,1) NormalPoint2(i,2) 1]';

D1=A1'\*X;

D2=A2'\*X;

D3=A3'\*X;

D4=A4'\*X;

if (D2>D1 && D2>D3) && D2>D4

continue ;

end

error=error+1;

A2=A2+X;

if D1>=D2

A1=A1-alpha\*X;

end

if D3>=D2

A3=A3-alpha\*X;

end

if D4>=D2

A4=A4-alpha\*X;

end

end

for i=1:100

X=[NormalPoint3(i,1) NormalPoint3(i,2) 1]';

D1=A1'\*X;

D2=A2'\*X;

D3=A3'\*X;

D4=A4'\*X;

if (D3>D1 && D3>D2) && D3>D4

continue ;

end

error=error+1;

A3=A3+X;

if D1>=D3

A1=A1-alpha\*X;

end

if D2>=D3

A2=A2-alpha\*X;

end

if D4>=D3

A4=A4-alpha\*X;

end

end

for i=1:100

X=[NormalPoint4(i,1) NormalPoint4(i,2) 1]';

D1=A1'\*X;

D2=A2'\*X;

D3=A3'\*X;

D4=A4'\*X;

if (D4>D1 && D4>D2) && D4>D3

continue ;

end

error=error+1;

A4=A4+X;

if D1>=D4

A1=A1-alpha\*X;

end

if D2>=D4

A2=A2-alpha\*X;

end

if D3>=D4

A3=A3-alpha\*X;

end

end

EIT(1,it)=it;

EIT(2,it)=error;

end

E1=[(A1(1)-A2(1)) (A1(2)-A2(2)) (A1(3)-A2(3))];

x=linspace(-1,2);

y=(-E1(1)\*x-E1(3))/E1(2);

L=plot(x,y);

set(L,'Color','blue');

E1=[(A1(1)-A3(1)) (A1(2)-A3(2)) (A1(3)-A3(3))];

x=linspace(-1,2);

y=(-E1(1)\*x-E1(3))/E1(2);

L=plot(x,y);

set(L,'Color','blue');

E1=[(A1(1)-A4(1)) (A1(2)-A4(2)) (A1(3)-A4(3))];

x=linspace(-1,2);

y=(-E1(1)\*x-E1(3))/E1(2);

L=plot(x,y);

set(L,'Color','blue');

E1=[(A2(1)-A3(1)) (A2(2)-A3(2)) (A2(3)-A3(3))];

x=linspace(-1,2);

y=(-E1(1)\*x-E1(3))/E1(2);

L=plot(x,y);

set(L,'Color','red');

E1=[(A2(1)-A4(1)) (A2(2)-A4(2)) (A2(3)-A4(3))];

x=linspace(-1,2);

y=(-E1(1)\*x-E1(3))/E1(2);

L=plot(x,y);

set(L,'Color','red');

E1=[(A3(1)-A4(1)) (A3(2)-A4(2)) (A3(3)-A4(3))];

x=linspace(-1,2);

y=(-E1(1)\*x-E1(3))/E1(2);

L=plot(x,y);

set(L,'Color','green');

for i=1:50

for j=1:50

x=-1+3\*i/50-0.03;

y=-1+3\*j/50-0.03;

D1=x\*A1(1)+y\*A1(2)+A1(3);

D2=x\*A2(1)+y\*A2(2)+A2(3);

D3=x\*A3(1)+y\*A3(2)+A3(3);

D4=x\*A4(1)+y\*A4(2)+A4(3);

if D1>D2&&D1>D3&&D1>D4

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

end

if D2>D1&&D2>D3&&D2>D4

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

end

if D3>D1&&D3>D2&&D3>D4

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

end

if D4>D1&&D4>D2&&D4>D3

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

end

end

end

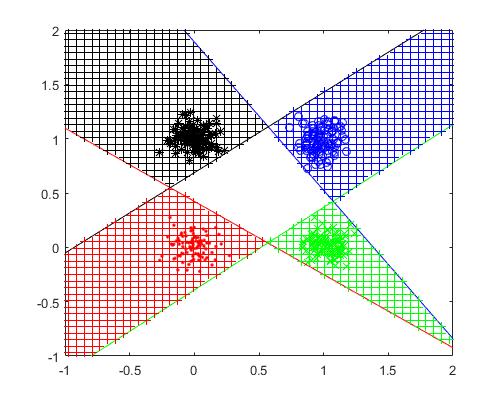
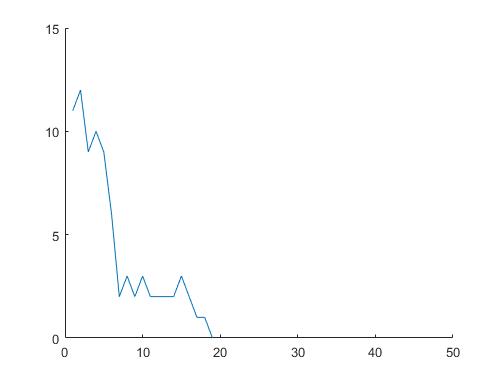
hFig = figure(2);

set(hFig,'name','error','Position', [300 0 500 400]);

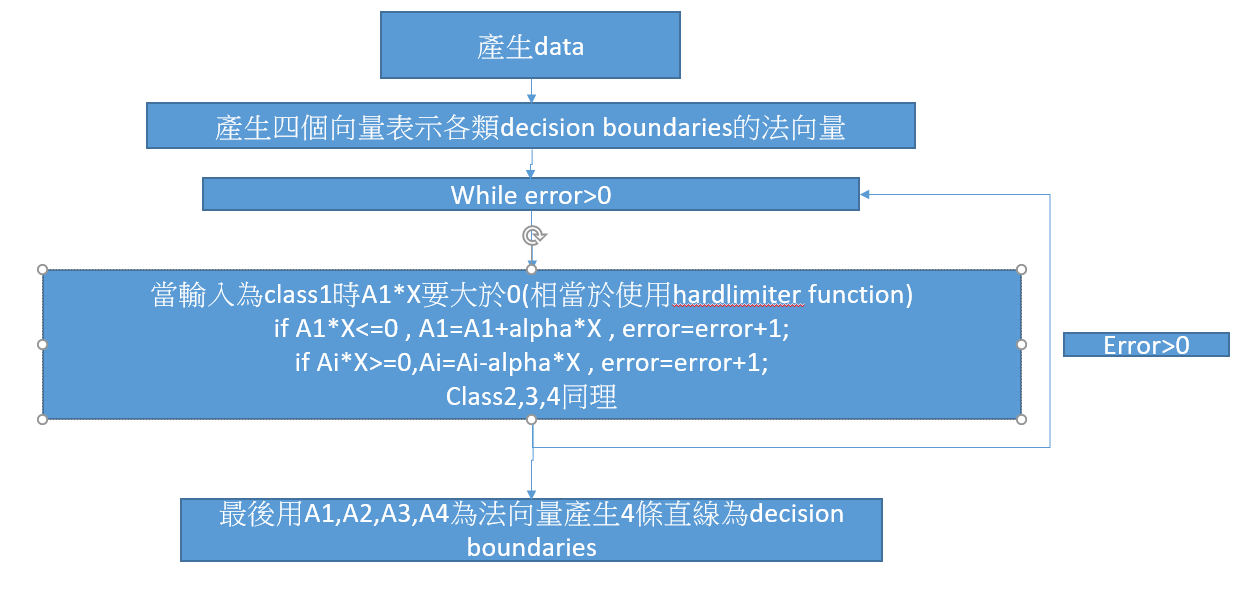
line(EIT(1,:),EIT(2,:));

axis([0 20 0 10]);

1. multi-class linear perceptrons to find the linear decision boundaries and regions。Activation functions 為 hardlimiter function。



Flowchart



當輸入為class1時除了A1\*X要為正數(+1)其他皆要為負數(-1)，其中\*為內積表示，當不符合時需要調整，調整方式如上表。

在這次的範例中4筆data可以直接被劃分出來(error可以到0)，若無法分割(error無法降為0)，則可設定當error小於一個值時即停止，或是當iteration達到一個定量時停止。

MATLAB COED

hFig = figure(1);

set(hFig,'name','multi-class linear perceptrons to find the linear decision boundaries and regions¡CActivation functions ¬° hardlimiter function¡C','Position', [300 0 500 400]);

Mean1 = [0;0];

Mean2=[1;0];

Mean3=[1;1];

Mean4=[0;1];

MySigma = [0.01 0; 0 0.01];

NormalPoint1 = mvnrnd(Mean1, MySigma, 100);

NormalPoint2 = mvnrnd(Mean2, MySigma, 100);

NormalPoint3 = mvnrnd(Mean3, MySigma, 100);

NormalPoint4 = mvnrnd(Mean4, MySigma, 100);

L=plot(NormalPoint1(:,1),NormalPoint1(:,2),'.','MarkerEdgeColor','r');

hold on;

M=plot(NormalPoint2(:,1),NormalPoint2(:,2),'x','MarkerEdgeColor','g');

N=plot(NormalPoint3(:,1),NormalPoint3(:,2),'o','MarkerEdgeColor','b');

O=plot(NormalPoint4(:,1),NormalPoint4(:,2),'\*','MarkerEdgeColor','k');

axis([-1 2 -1 2]);

alpha=1;

A1=[0 0 0]';

A2=[0 0 0]';

A3=[0 0 0]';

A4=[0 0 0]';

it=0;

error=1000;

while(error>0)

it=it+1;

error=0;

for i=1:100

X=[NormalPoint1(i,1) NormalPoint1(i,2) 1]';

D1=A1'\*X;

D2=A2'\*X;

D3=A3'\*X;

D4=A4'\*X;

%ª½±µ¥Î¥¿­t¸¹§PÂ\_¬Û·í©ó¨Ï¥Îhardlimiter function

if D1<=0

error=error+1;

A1=A1+alpha\*X;

end

if D2>=0

error=error+1;

A2=A2-alpha\*X;

end

if D3>=0

error=error+1;

A3=A3-alpha\*X;

end

if D4>=0

error=error+1;

A4=A4-alpha\*X;

end

end

for i=1:100

X=[NormalPoint2(i,1) NormalPoint2(i,2) 1]';

D1=A1'\*X;

D2=A2'\*X;

D3=A3'\*X;

D4=A4'\*X;

if D2<=0

error=error+1;

A2=A2+alpha\*X;

end

if D1>=0

error=error+1;

A1=A1-alpha\*X;

end

if D3>=0

error=error+1;

A3=A3-alpha\*X;

end

if D4>=0

error=error+1;

A4=A4-alpha\*X;

end

end

for i=1:100

X=[NormalPoint3(i,1) NormalPoint3(i,2) 1]';

D1=A1'\*X;

D2=A2'\*X;

D3=A3'\*X;

D4=A4'\*X;

if D3<=0

error=error+1;

A3=A3+alpha\*X;

end

if D2>=0

error=error+1;

A2=A2-alpha\*X;

end

if D1>=0

error=error+1;

A1=A1-alpha\*X;

end

if D4>=0

error=error+1;

A4=A4-alpha\*X;

end

end

for i=1:100

X=[NormalPoint4(i,1) NormalPoint4(i,2) 1]';

D1=A1'\*X;

D2=A2'\*X;

D3=A3'\*X;

D4=A4'\*X;

if D4<=0

error=error+1;

A4=A4+alpha\*X;

end

if D2>=0

error=error+1;

A2=A2-alpha\*X;

end

if D3>=0

error=error+1;

A3=A3-alpha\*X;

end

if D1>=0

error=error+1;

A1=A1-alpha\*X;

end

end

EIT(1,it)=it;

EIT(2,it)=error;

end

x=linspace(-1,2);

y=(-A1(1)\*x-A1(3))/A1(2);

L=plot(x,y);

set(L,'Color','red');

x=linspace(-1,2);

y=(-A2(1)\*x-A2(3))/A2(2);

L=plot(x,y);

set(L,'Color','green');

x=linspace(-1,2);

y=(-A3(1)\*x-A3(3))/A3(2);

L=plot(x,y);

set(L,'Color','blue');

x=linspace(-1,2);

y=(-A4(1)\*x-A4(3))/A4(2);

L=plot(x,y);

set(L,'Color','black');

for i=1:50

for j=1:50

x=-1+3\*i/50-0.03;

y=-1+3\*j/50-0.03;

D1=x\*A1(1)+y\*A1(2)+A1(3);

D2=x\*A2(1)+y\*A2(2)+A2(3);

D3=x\*A3(1)+y\*A3(2)+A3(3);

D4=x\*A4(1)+y\*A4(2)+A4(3);

if D1>0 && D2<0 && D3<0 && D4<0

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

elseif D2>0 && D1<0 && D3<0 && D4<0

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

elseif D3>0 && D2<0 && D1<0 && D4<0

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

elseif D4>0 && D2<0 && D3<0 && D1<0

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

end

end

end

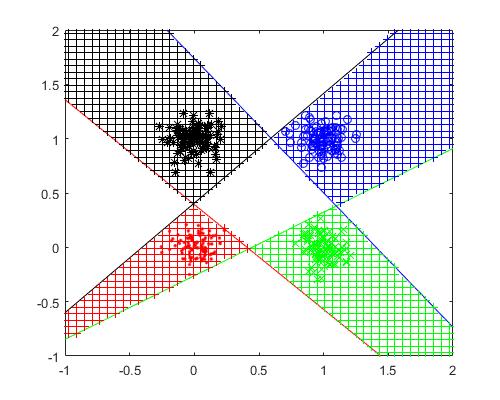
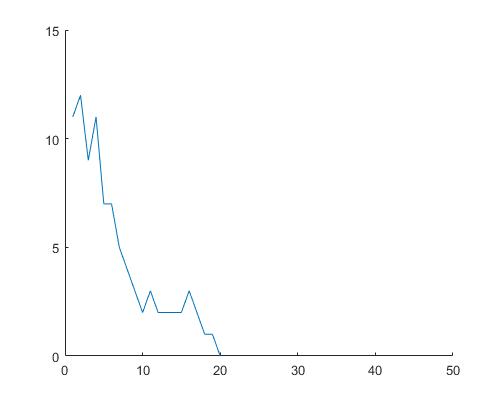
hFig = figure(2);

set(hFig,'name','error','Position', [300 0 500 400]);

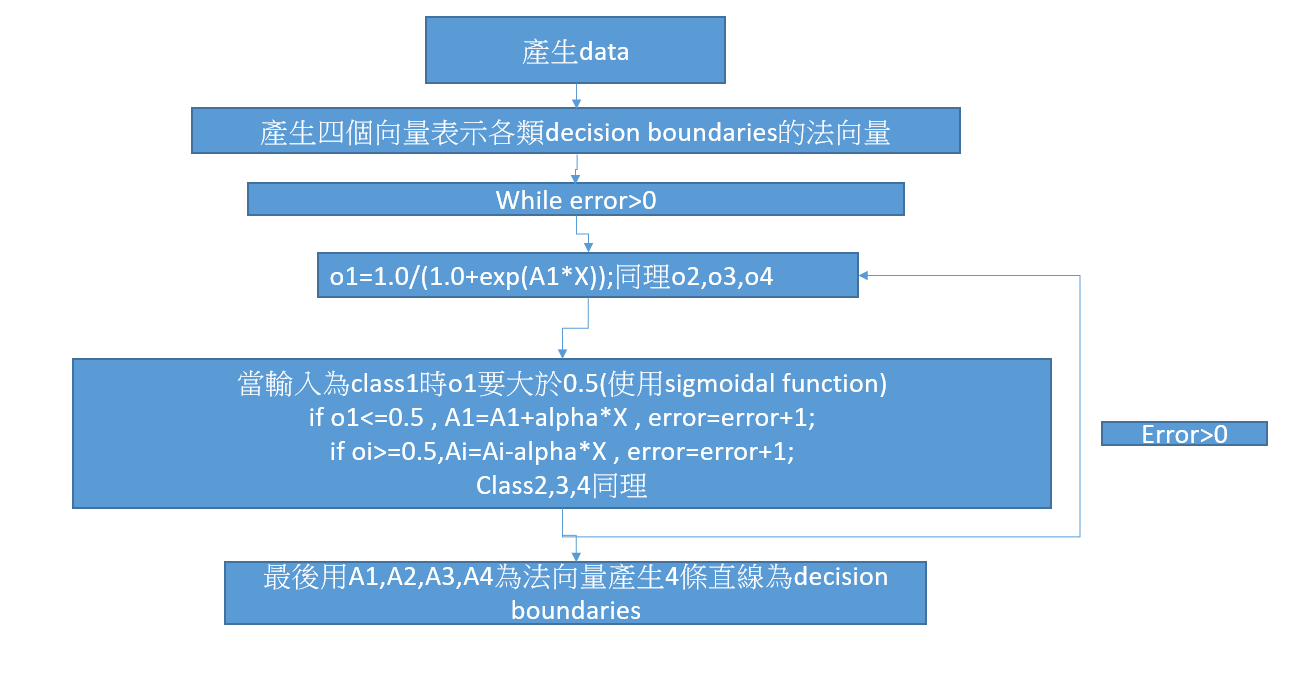
line(EIT(1,:),EIT(2,:));

axis([0 50 0 15]);

1. multi-class linear perceptrons to find the “decision regions”。Activation functions 為 sigmoidal function。



Flowchart



當輸入為class1時除了o1(sigmoid後的結果)要>0.5其他皆要<0.5，其他class同理。其中\*為內積表示，當不符合時需要調整，調整方式如上表。

在這次的範例中4筆data可以直接被劃分出來(error可以到0)，若無法分割(error無法降為0)，則可設定當error小於一個值時即停止，或是當iteration達到一個定量時停止。

MATLAB CODE

hFig = figure(3);

set(hFig,'name',' multi-class linear perceptrons to find the ¡§decision regions¡¨¡CActivation functions ¬° sigmoidal function¡C','Position', [300 0 500 400]);

Mean1 = [0;0];

Mean2=[1;0];

Mean3=[1;1];

Mean4=[0;1];

MySigma = [0.01 0; 0 0.01];

NormalPoint1 = mvnrnd(Mean1, MySigma, 100);

NormalPoint2 = mvnrnd(Mean2, MySigma, 100);

NormalPoint3 = mvnrnd(Mean3, MySigma, 100);

NormalPoint4 = mvnrnd(Mean4, MySigma, 100);

L=plot(NormalPoint1(:,1),NormalPoint1(:,2),'.','MarkerEdgeColor','r');

hold on;

M=plot(NormalPoint2(:,1),NormalPoint2(:,2),'x','MarkerEdgeColor','g');

N=plot(NormalPoint3(:,1),NormalPoint3(:,2),'o','MarkerEdgeColor','b');

O=plot(NormalPoint4(:,1),NormalPoint4(:,2),'\*','MarkerEdgeColor','k');

axis([-1 2 -1 2]);

alpha=1;

A1=[0 0 0]';

A2=[0 0 0]';

A3=[0 0 0]';

A4=[0 0 0]';

it=0;

error=1000;

while(error>0)

it=it+1;

error=0;

for i=1:100

X=[NormalPoint1(i,1) NormalPoint1(i,2) 1]';

D1=A1'\*X;

D2=A2'\*X;

D3=A3'\*X;

D4=A4'\*X;

%sigmoid function

o1=1.0/(1.0+exp(-D1));

o2=1.0/(1.0+exp(-D2));

o3=1.0/(1.0+exp(-D3));

o4=1.0/(1.0+exp(-D4));

if o1<=0.5

error=error+1;

A1=A1+alpha\*X;

end

if o2>=0.5

error=error+1;

A2=A2-alpha\*X;

end

if o3>=0.5

error=error+1;

A3=A3-alpha\*X;

end

if o4>=0.5

error=error+1;

A4=A4-alpha\*X;

end

end

for i=1:100

X=[NormalPoint2(i,1) NormalPoint2(i,2) 1]';

D1=A1'\*X;

D2=A2'\*X;

D3=A3'\*X;

D4=A4'\*X;

%sigmoid function

o1=1.0/(1.0+exp(-D1));

o2=1.0/(1.0+exp(-D2));

o3=1.0/(1.0+exp(-D3));

o4=1.0/(1.0+exp(-D4));

if o2<=0.5

error=error+1;

A2=A2+alpha\*X;

end

if o1>=0.5

error=error+1;

A1=A1-alpha\*X;

end

if o3>=0.5

error=error+1;

A3=A3-alpha\*X;

end

if o4>=0.5

error=error+1;

A4=A4-alpha\*X;

end

end

for i=1:100

X=[NormalPoint3(i,1) NormalPoint3(i,2) 1]';

D1=A1'\*X;

D2=A2'\*X;

D3=A3'\*X;

D4=A4'\*X;

%sigmoid function

o1=1.0/(1.0+exp(-D1));

o2=1.0/(1.0+exp(-D2));

o3=1.0/(1.0+exp(-D3));

o4=1.0/(1.0+exp(-D4));

if o3<=0.5

error=error+1;

A3=A3+alpha\*X;

end

if o2>=0.5

error=error+1;

A2=A2-alpha\*X;

end

if o1>=0.5

error=error+1;

A1=A1-alpha\*X;

end

if o4>=0.5

error=error+1;

A4=A4-alpha\*X;

end

end

for i=1:100

X=[NormalPoint4(i,1) NormalPoint4(i,2) 1]';

D1=A1'\*X;

D2=A2'\*X;

D3=A3'\*X;

D4=A4'\*X;

%sigmoid function

o1=1.0/(1.0+exp(-D1));

o2=1.0/(1.0+exp(-D2));

o3=1.0/(1.0+exp(-D3));

o4=1.0/(1.0+exp(-D4));

if o4<=0.5

error=error+1;

A4=A4+alpha\*X;

end

if o2>=0.5

error=error+1;

A2=A2-alpha\*X;

end

if o3>=0.5

error=error+1;

A3=A3-alpha\*X;

end

if o1>=0.5

error=error+1;

A1=A1-alpha\*X;

end

end

EIT(1,it)=it;

EIT(2,it)=error;

end

x=linspace(-1,2);

y=(-A1(1)\*x-A1(3))/A1(2);

L=plot(x,y);

set(L,'Color','red');

x=linspace(-1,2);

y=(-A2(1)\*x-A2(3))/A2(2);

L=plot(x,y);

set(L,'Color','green');

x=linspace(-1,2);

y=(-A3(1)\*x-A3(3))/A3(2);

L=plot(x,y);

set(L,'Color','blue');

x=linspace(-1,2);

y=(-A4(1)\*x-A4(3))/A4(2);

L=plot(x,y);

set(L,'Color','black');

for i=1:50

for j=1:50

x=-1+3\*i/50-0.03;

y=-1+3\*j/50-0.03;

D1=x\*A1(1)+y\*A1(2)+A1(3);

D2=x\*A2(1)+y\*A2(2)+A2(3);

D3=x\*A3(1)+y\*A3(2)+A3(3);

D4=x\*A4(1)+y\*A4(2)+A4(3);

if D1>0 && D2<0 && D3<0 && D4<0

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

elseif D2>0 && D1<0 && D3<0 && D4<0

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

elseif D3>0 && D2<0 && D1<0 && D4<0

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

elseif D4>0 && D2<0 && D3<0 && D1<0

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

end

end

end

hFig = figure(4);

set(hFig,'name','error','Position', [300 0 500 400]);

line(EIT(1,:),EIT(2,:));

axis([0 50 0 15]);

Extra analysis and Discussion

使用分辨函數的方式，在求4類的decision boundaries總共會有6條線C(4,2)，每兩類中間就會有一條decision boundary，所以會較為凌亂。

若要將區域用線劃分開來，更方便觀看的話，可以把一些不必要的線截掉。

以CLASS1為例

利用分辨函數求出三條decision boundary D1-D2=0,D1-D3=0,D1-D4=0。求出來後將三條線落在彼此負半空間的線段截去，只顯示落在彼此正半空間的線段，即可將CLASS1獨立劃分出來。

在這次的範例中使用hardlimiter function與sigmoid function差異並不大，大部分調整方式都差不多。

參考資料

教授的課本 類神經網路 黃國源 著