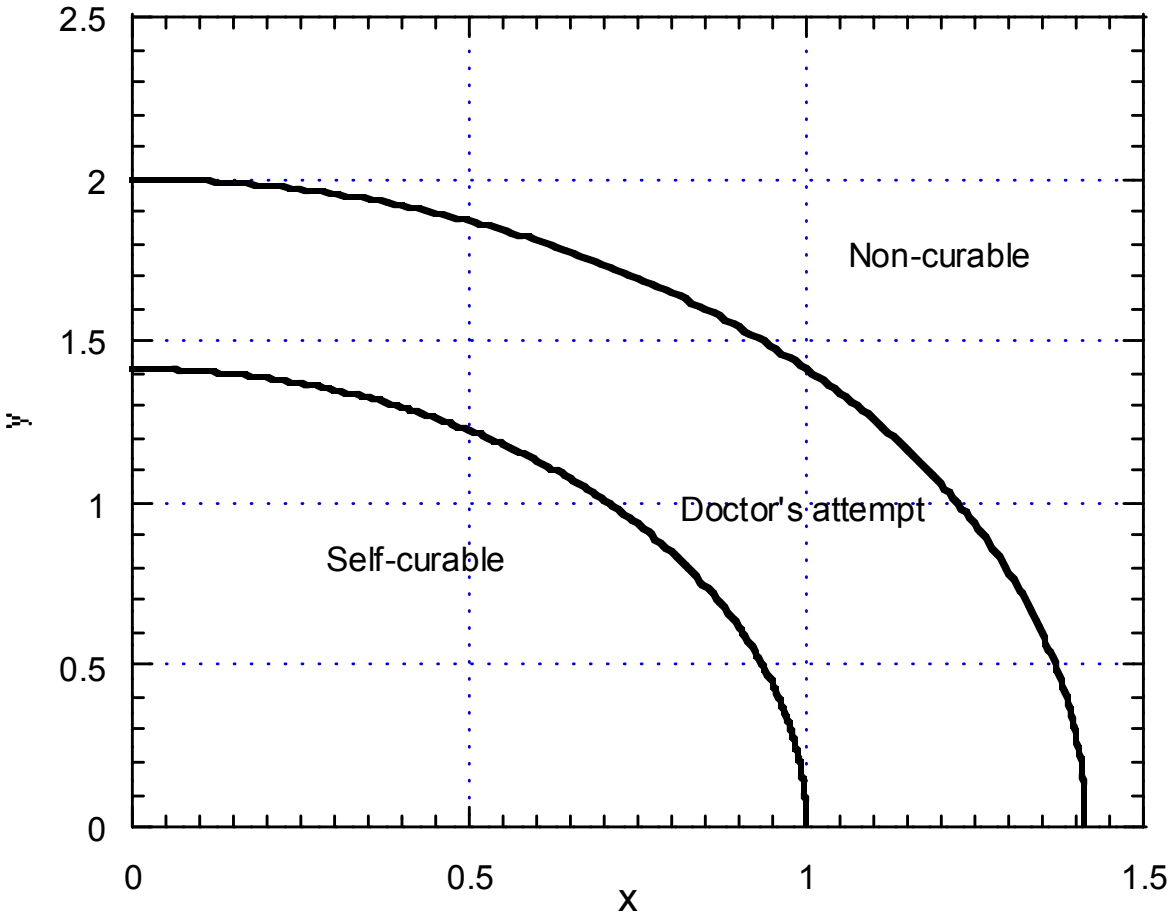
ENEE434 Spring 2004
Homework3 Solution: Y.J. 03/16/04
Problem 2:
a) Set up a feedforword neural network:
for x>0, and y>0:

$$x^2 + 0.5 \cdot y^2 = a^2$$

There are three cases:

- $a$<1, automatically curable:
- 1<$a$<sqrt(2), attempt cure needed by doctors.
- $a$>sqrt(2), non-curable.




Therefore, a three layer feedforward network is setup. The neural network netLesion has 3 neurons in the output layer that gives a vector output. The output is desired to be:

| Automatically curable: | Needs doctor attempt: | Non-curable |
|---|---|---|
| [1, 0, 0] | [0, 1, 0] | [0, 0, 1] |

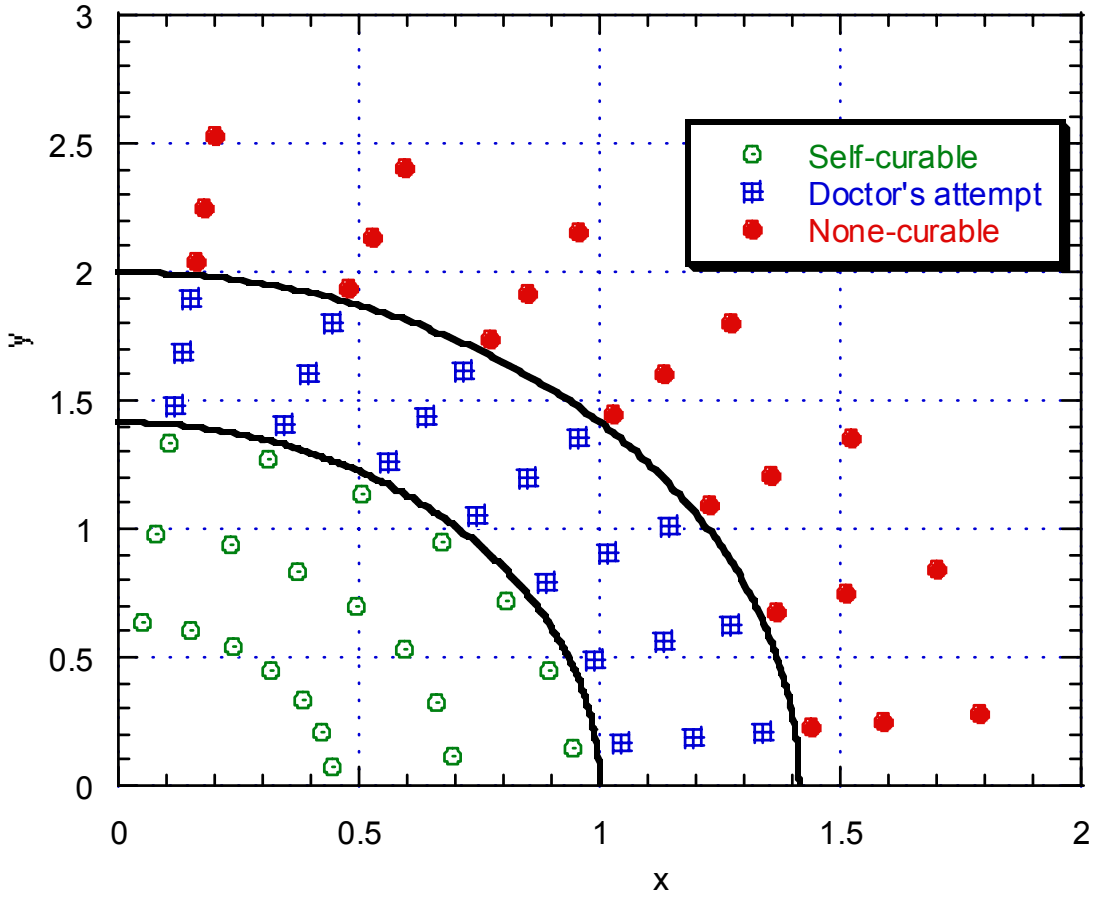
Matlab code for setting up the neural network:

```
%set up the neural network%
   netLesion = newff([0 2;0 2],[5 5 3],{'logsig' 'purelin' 'logsig'});
```

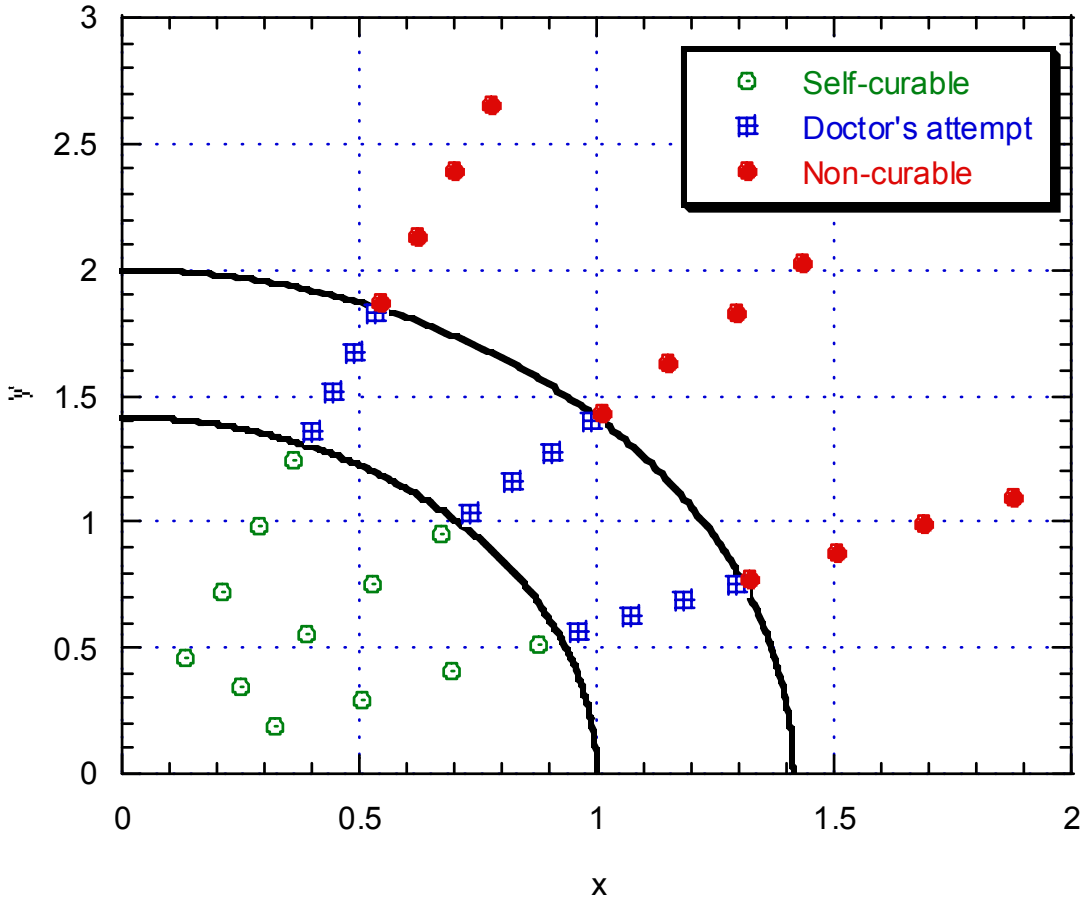
```
%network after 63 inputs training.%
   netLesion.trainParam.epochs = 50;
   netLesion = train(netLesion,P,T);

   %replace the transferfunction of the last layer to hardlim after training.
   netvec.layers{3}.transferFcn='hardlim';
```

b) 21 data points from each class for training:



c) 12 new data points from each class for testing:




Testing inputs:

| | |
|---|---|
| Testing Input: | Pt =<br>Columns 1 through 7<br>0.3234 0.5081 0.6929 0.8777 0.2475 0.3889 0.5303<br>0.1894 0.2977 0.4059 0.5141 0.3500 0.5500 0.7500<br>Columns 8 through 14<br>0.6718 0.1339 0.2105 0.2870 0.3635 0.9608 1.0717<br>0.9500 0.4573 0.7186 0.9799 1.2412 0.5628 0.6278<br>Columns 15 through 21<br>1.1826 1.2934 0.7354 0.8202 0.9051 0.9899 0.3980<br>0.6927 0.7577 1.0400 1.1600 1.2800 1.4000 1.3588<br>Columns 22 through 28<br>0.4439 0.4898 0.5358 1.3211 1.5059 1.6907 1.8755<br>1.5156 1.6724 1.8292 0.7739 0.8821 0.9904 1.0986<br>Columns 29 through 35<br>1.0112 1.1526 1.2940 1.4354 0.5472 0.6238 0.7003<br>1.4300 1.6300 1.8300 2.0300 1.8684 2.1297 2.3910<br>Column 36<br>0.7768<br>2.6523 |

| | |
|---|---|
| Target output: | Tt =<br>Columns 1 through 11<br>1 1 1 1 1 1 1 1 1 1 1<br>0 0 0 0 0 0 0 0 0 0 0<br>0 0 0 0 0 0 0 0 0 0 0<br>Columns 12 through 22<br>1 0 0 0 0 0 0 0 0 0 0<br>0 1 1 1 1 1 1 1 1 1 1<br>0 0 0 0 0 0 0 0 0 0 0<br>Columns 23 through 33<br>0 0 0 0 0 0 0 0 0 0 0<br>1 1 0 0 0 0 0 0 0 0 0<br>0 0 1 1 1 1 1 1 1 1 1<br>Columns 34 through 36<br>0 0 0<br>0 0 0<br>1 1 1 |
| Output: | Y =<br>Columns 1 through 7<br>1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000<br>0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000<br>0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000<br>Columns 8 through 14<br>1.0000 1.0000 1.0000 1.0000 1.0000 0.0000 0.0000<br>0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 1.0000<br>0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000<br>Columns 15 through 21<br>0.0000 **0.0000** 0.0000 0.0000 0.0000 0.0000 0.0000<br>1.0000 **0.9060** 1.0000 1.0000 1.0000 1.0000 1.0000<br>0.0000 **1.0000** 0.0000 0.0000 0.0000 1.0000 0.0000<br>Columns 22 through 28<br>0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000<br>1.0000 1.0000 0.0000 0.0000 0.0000 0.0000 0.0000<br>0.0000 0.0000 1.0000 1.0000 1.0000 1.0000 1.0000<br>Columns 29 through 35<br>**0.0000** 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000<br>**0.9983** 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000<br>**1.0000** 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000<br>Column 36<br>0.0000<br>0.0000<br>1.0000 |

Discussion of simulation result:

Except the 16th and 29th (red in the table above) testing exampler, the netLesion gives the correct results:

The 16th, x=1.2934, y=0.7577, a=1.40.

The 29th, x=1.0112, y=1.4300, a=1.43.
Both of them are on the border of needing doctor's attempt or non-curable.

d) x, y plot of the situation:
as shown in part a).

Appendix: Matlab code:

```
% enee434 hw3# problem2%

   % Three training sets:

   theta=1.5708/7*[0.5:1:6.5];

   for i=1:21;
      one(i)=1;
      zero(i)=0;
   end

   % Lesions are automaticlly cured.
   xself1=0.45*cos(theta);
   yself1=0.45*1.4142*sin(theta);

   xself2=0.7*cos(theta);
   yself2=0.7*1.4142*sin(theta);

   xself3=0.95*cos(theta);
   yself3=0.95*1.4142*sin(theta);

   xself=[xself1,xself2,xself3];
   yself=[yself1,yself2,yself3];

   % Training set Lesions are automaticlly cured.
   Pself=[xself;yself];
   Tself=[one;zero;zero];

   % Lesions have to be treated by doctors.
   xdoc1=1.05*cos(theta);
   ydoc1=1.05*1.4142*sin(theta);

   xdoc2=1.2*cos(theta);
   ydoc2=1.2*1.4142*sin(theta);

   xdoc3=1.35*cos(theta);
   ydoc3=1.35*1.4142*sin(theta);
```

```
xdoc=[xdoc1,xdoc2,xdoc3];
ydoc=[ydoc1,ydoc2,ydoc3];

% Training set Lesions have to be treated by doctors.
Pdoc=[xdoc;ydoc];
Tdoc=[zero;one;zero];


% Lesions can't be cured
xnot1=1.45*cos(theta);
ynot1=1.45*1.4142*sin(theta);

xnot2=1.6*cos(theta);
ynot2=1.6*1.4142*sin(theta);

xnot3=1.8*cos(theta);
ynot3=1.8*1.4142*sin(theta);

xnot=[xnot1,xnot2,xnot3];
ynot=[ynot1,ynot2,ynot3];

% Train set of Lesions can't be cured
Pnot=[xnot;ynot];
Tnot=[zero;zero;one];

% Complete training set for all three cases:
P=[Pself, Pdoc, Pnot];
T=[Tself, Tdoc, Tnot];

%Px=P(1,:);
%Py=P(2,:);
%a_sq=Px.^2+0.5*Py.^2;
%a=sqrt(a_sq);

%T1=T(1,:);
%T2=T(2,:);
%T3=T(3,:);

% Testing sets:
% Lesions are automaticlly cured.
a_min=0.35;
a_step=0.2; %a_max=0.95

for i=1:4;
   Pt(1, i)=(a_min+a_step*(i-1))*cos(0.3927);
   Pt(2, i)=(a_min+a_step*(i-1))*1.4142*sin(0.3927);
```

```
    Tt(1, i)=1;
    Tt(2, i)=0;
    Tt(3, i)=0;
end;
for i=5:8;
    Pt(1, i)=(a_min+a_step*(i-5))*cos(0.7854);
    Pt(2, i)=(a_min+a_step*(i-5))*1.4142*sin(0.7854);
    Tt(1, i)=1;
    Tt(2, i)=0;
    Tt(3, i)=0;
end;
for i=9:12;
    Pt(1, i)=(a_min+a_step*(i-9))*cos(1.1781);
    Pt(2, i)=(a_min+a_step*(i-9))*1.4142*sin(1.1781);
    Tt(1, i)=1;
    Tt(2, i)=0;
    Tt(3, i)=0;
end;

%Lesions have to be treated by doctors.
a_min=1.04;
a_step=0.12; %a_max=1.40

for i=13:16;
    Pt(1, i)=(a_min+a_step*(i-13))*cos(0.3927);
    Pt(2, i)=(a_min+a_step*(i-13))*1.4142*sin(0.3927);
    Tt(1, i)=0;
    Tt(2, i)=1;
    Tt(3, i)=0;
end;
for i=17:20;
    Pt(1, i)=(a_min+a_step*(i-17))*cos(0.7854);
    Pt(2, i)=(a_min+a_step*(i-17))*1.4142*sin(0.7854);
    Tt(1, i)=0;
    Tt(2, i)=1;
    Tt(3, i)=0;
end;
for i=21:24;
    Pt(1, i)=(a_min+a_step*(i-21))*cos(1.1781);
    Pt(2, i)=(a_min+a_step*(i-21))*1.4142*sin(1.1781);
    Tt(1, i)=0;
    Tt(2, i)=1;
    Tt(3, i)=0;
end;

% Lesions can't be cured
```

```
    a_min=1.43;
    a_step=0.2; %a_max=2.03

    for i=25:28;
        Pt(1, i)=(a_min+a_step*(i-25))*cos(0.3927);
        Pt(2, i)=(a_min+a_step*(i-25))*1.4142*sin(0.3927);
        Tt(1, i)=0;
        Tt(2, i)=0;
        Tt(3, i)=1;
    end;
    for i=29:32;
        Pt(1, i)=(a_min+a_step*(i-29))*cos(0.7854);
        Pt(2, i)=(a_min+a_step*(i-29))*1.4142*sin(0.7854);
        Tt(1, i)=0;
        Tt(2, i)=0;
        Tt(3, i)=1;
    end;
    for i=33:36;
        Pt(1, i)=(a_min+a_step*(i-33))*cos(1.1781);
        Pt(2, i)=(a_min+a_step*(i-33))*1.4142*sin(1.1781);
        Tt(1, i)=0;
        Tt(2, i)=0;
        Tt(3, i)=1;
    end;

    Tt1=Tt(1,:);
    Tt2=Tt(2,:);
    Tt3=Tt(3,:);
    Ptx=Pt(1,:);
    Pty=Pt(2,:);
    a_sq=Ptx.^2+0.5*Pty.^2;
    a=sqrt(a_sq);

%set up the neural network%
    netLesion = newff([0 2;0 2],[5 5 3],{'logsig' 'purelin' 'logsig'});

    %net = newff creates a new network with a dialog box.
    %newff(PR,[S1 S2...SNl],{TF1 TF2...TFNl},BTF,BLF,PF) takes,
    %PR - R x 2 matrix of min and max values for R input elements.
    %Si - Size of ith layer, for Nl layers.
    %TFi - Transfer function of ith layer, default = 'tansig'.
    %BTF - Backpropagation network training function, default = 'traingdx'.
    %BLF - Backpropagation weight/bias learning function, default = 'learngdm'.
    %PF - Performance function, default = 'mse'.
    %and returns an N layer feed-forward backprop network.
```

```
%network after 63 inputs training.%
   netLesion.trainParam.epochs = 50;
   netLesion = train(netLesion,P,T);

   %replace the transferfunction of the last layer to hardlim after training.
   netvec.layers{3}.transferFcn='hardlim';

%output after training with the test 24 differenet inputs Pt:
   Y = sim(netLesion,Pt);
```