Table of Contents

Tai Duc Nguyen - ECEC 487 - 10/23/2019

```
clear; close all;
```

Disclaimer

```
% Both problems 4.1 and 4.2 (below) uses the Multilayer Perceptron
 Neural
% Network Framework from Marcelo Augusto Costa Fernandes
 (mfernandes@dca.ufrn.br) available here:
% https://www.mathworks.com/matlabcentral/fileexchange/36253-the-
matrix-implementation-of-the-two-layer-multilayer-perceptron-mlp-
neural-networks
% with modifications for easy exploration of different parameters used
 in
% side the algorithm.

% The architecture of the neural network written by Fernandes is a
 simple 1
% input layer size p, 1 hidden layer with number of neurons H, and 1
 output
% layer size m. The error calculation uses Mean Square Error. This
% algoirthm only back-propagates after finishing feed-forwarding on a
 batch
% of inputs.
```

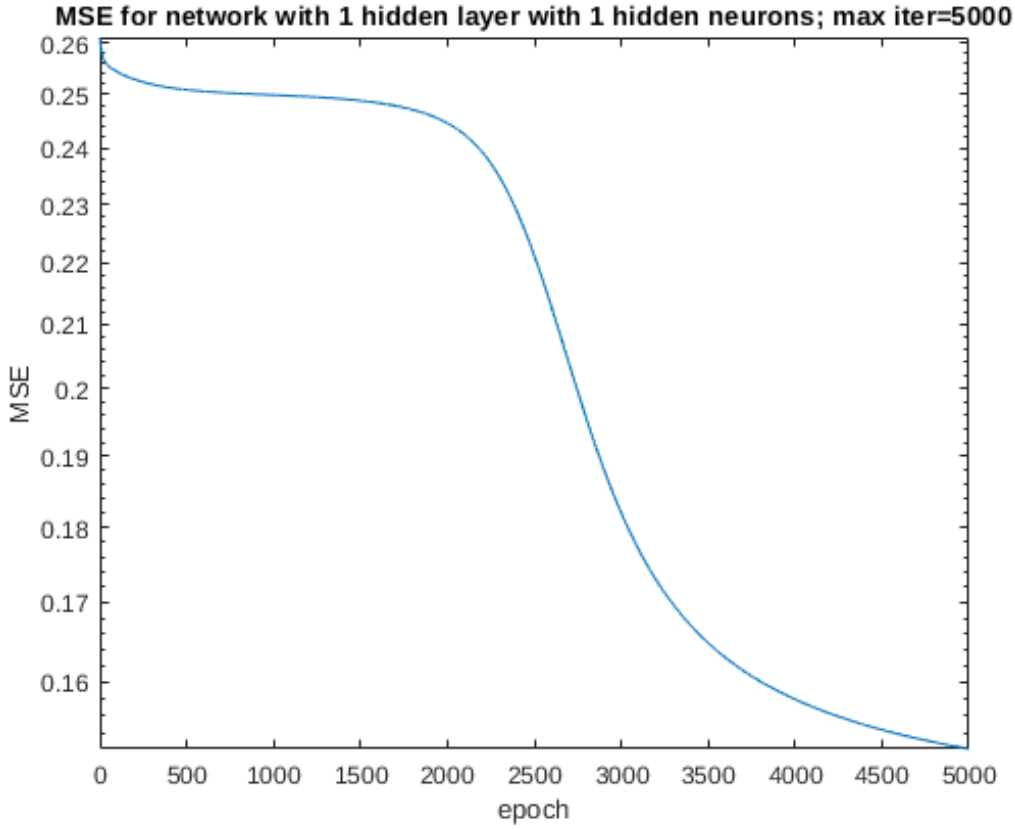
Problem 4.1 Chapter 4 Page 240

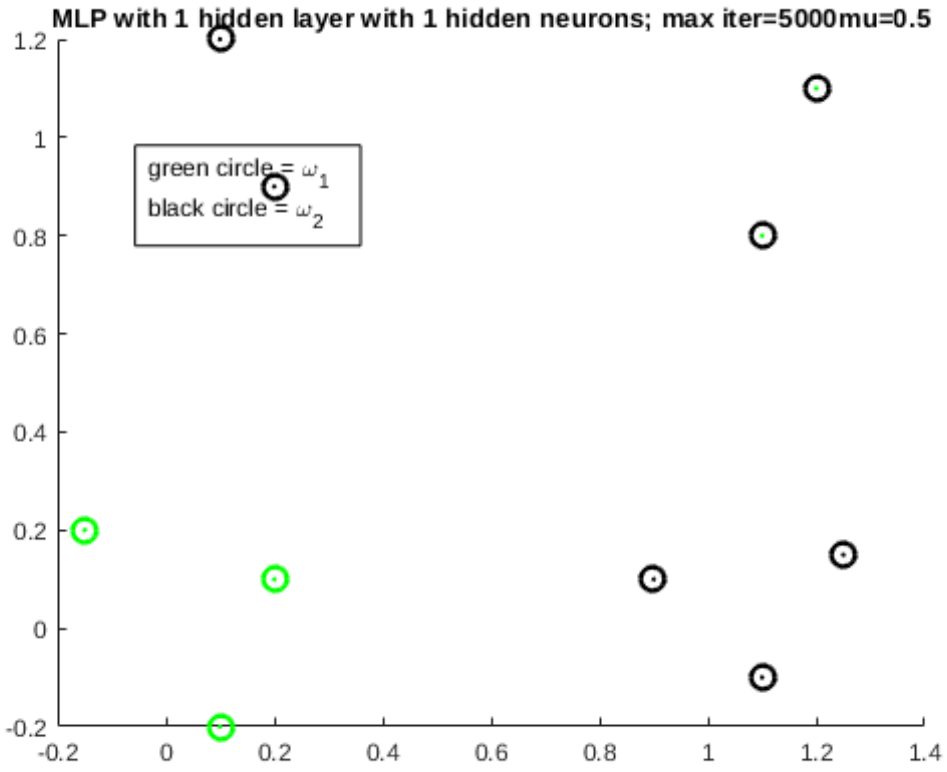
```
x1 = [0.1 0.2 -0.15 1.1 1.2; -0.2 0.1 0.2 0.8 1.1];
x2 = [1.1 1.25 0.9 0.1 0.2; -0.1 0.15 0.1 1.2 0.9];
x = [x1 x2];

y = [ones(1,size(x1,2))*1 ones(1,size(x2,2))*0];
```

H = 1; maxiter = 5000;

```
p = 2;
```

```
H = 1;
m = 1;

maxiter = 5000;
mu = .5;
alpha = 0;
MSEmin = 1e-20;
actfn1 = @(x) (1./(1+exp(-x)));
actfn2 = @(x) (1./(1+exp(-x)));

[Wx,Wy,MSE]=trainMLP(p,H,m,mu,alpha,x,y,actfn1,maxiter,MSEmin);

figure();
semilogy(MSE);
title(['MSE for network with 1 hidden layer with ', num2str(H), '
 hidden neurons; max iter=', num2str(maxiter)]);
xlabel('epoch'); ylabel('MSE');

disp(['D = [' num2str(y) ']']);

t = runMLP(x,Wx,Wy,actfn2);

disp(['Y = [' num2str(t) ']']);

figure();
hold on
scatter(x1(1,:),x1(2,:),'g.', 'LineWidth', 5);
scatter(x2(1,:),x2(2,:),'k.', 'LineWidth', 5);

for i=1:size(x,2)
    if t(i) > 0.5
        plot(x(1,i), x(2,i), 'go','MarkerSize',10, 'LineWidth', 2)
    else
        plot(x(1,i), x(2,i), 'ko','MarkerSize',10, 'LineWidth', 2)
    end
end
hold off

title(['MLP with 1 hidden layer with ', num2str(H), ' hidden neurons;
 max iter=', num2str(maxiter), 'mu=', num2str(mu)]);
dim = [.2 .5 .3 .3];
annotation('textbox',dim,'String', {'green circle = \omega_1', 'black
 circle = \omega_2'},'FitBoxToText','on');

D = [1  1  1  1  1  0  0  0  0  0]
Y = [0.96812     0.82346     0.95647     0.28313     0.28272
 0.30541     0.28628     0.30966     0.29757     0.31478]
```


MSE for network with 1 hidden layer with 1 hidden neurons; max iter=5000
MSE
epoch
0.26
0.25
0.24
0.23
0.22
0.21
0.2
0.19
0.18
0.17
0.16
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000

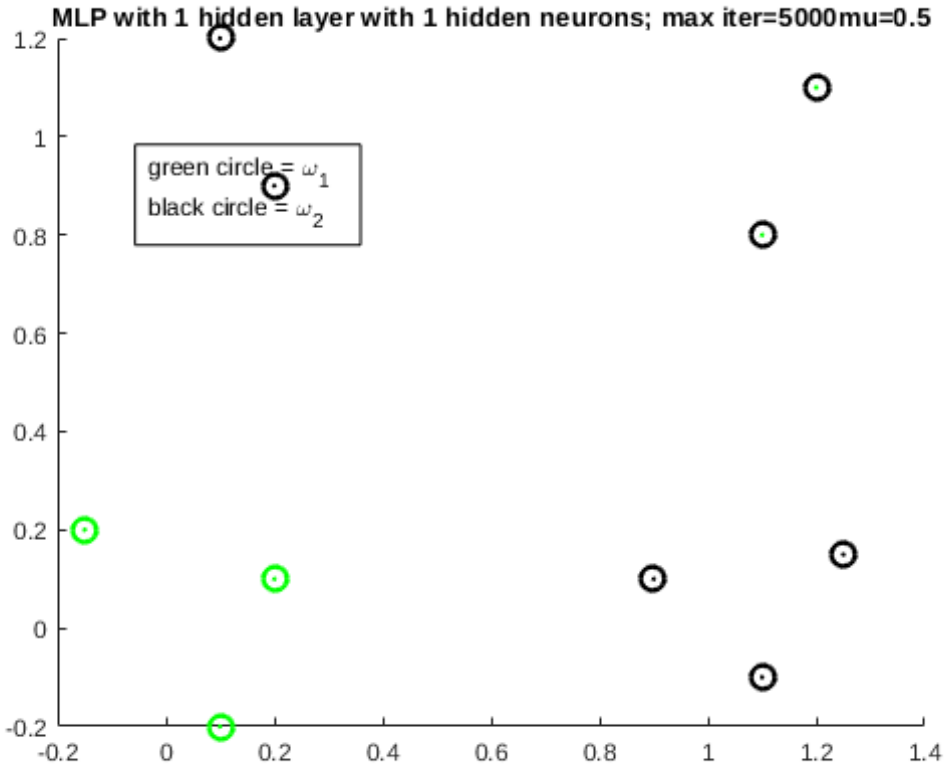


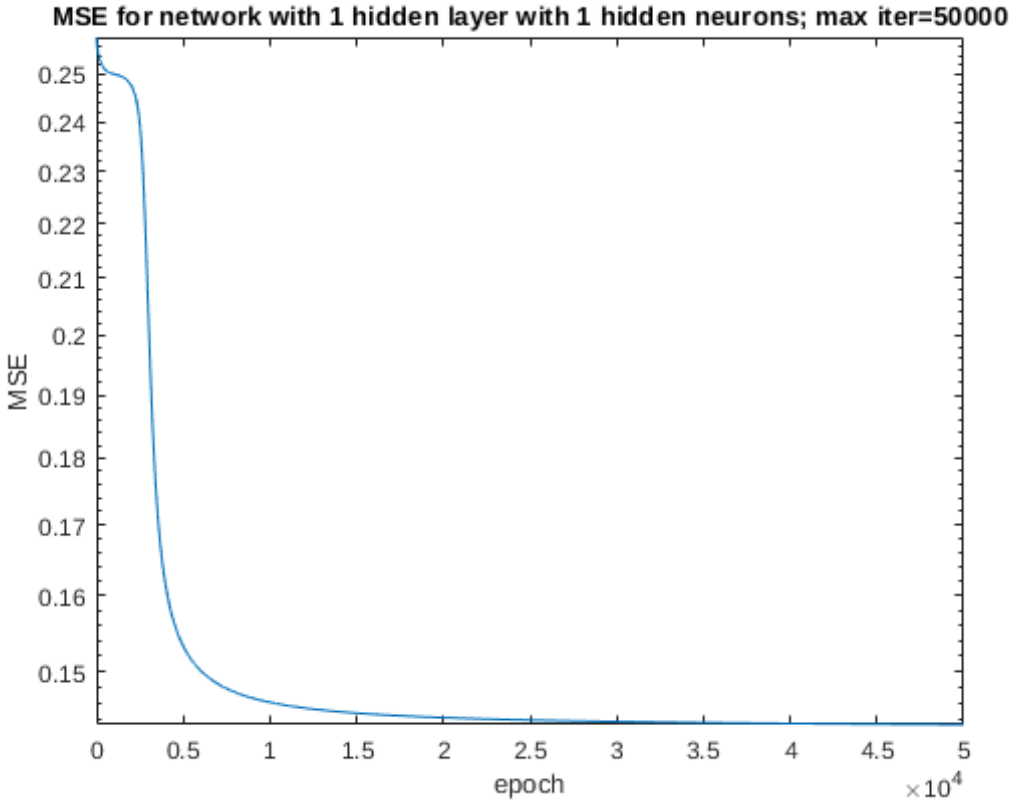
H = 1; maxiter = 50000;

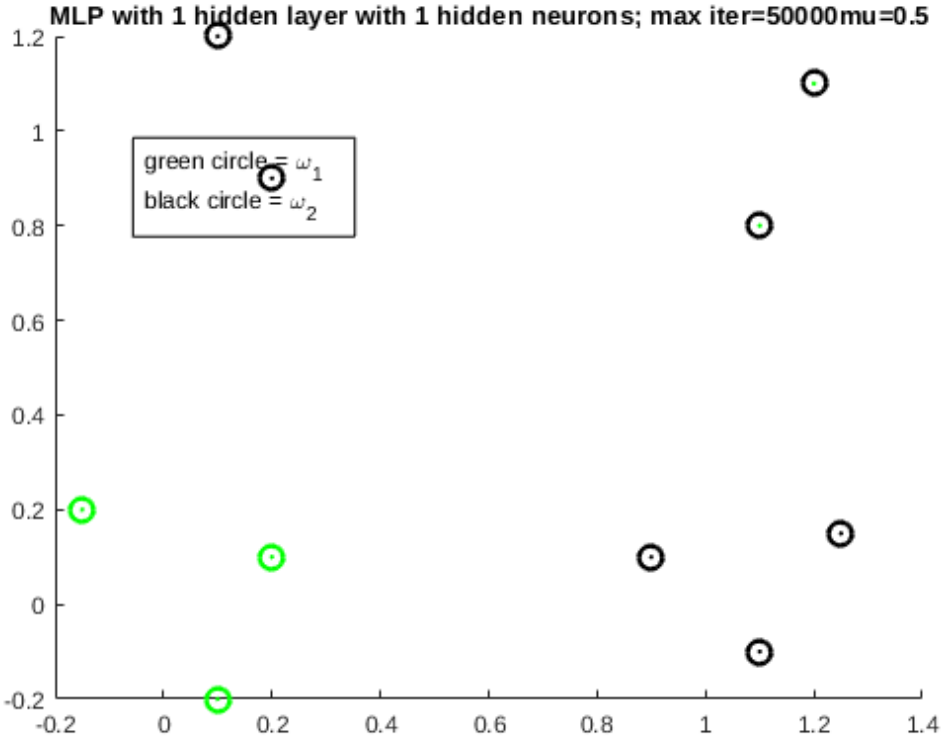
```
p = 2;
H = 1;
m = 1;

maxiter = 50000;
mu = .5;
alpha = 0;
MSEmin = 1e-20;
actfn1 = @(x) (1./(1+exp(-x)));
actfn2 = @(x) (1./(1+exp(-x)));

[Wx,Wy,MSE]=trainMLP(p,H,m,mu,alpha,x,y,actfn1,maxiter,MSEmin);

figure();
semilogy(MSE);
title(['MSE for network with 1 hidden layer with ', num2str(H), '
 hidden neurons; max iter=', num2str(maxiter)]);
xlabel('epoch'); ylabel('MSE');

disp(['D = [' num2str(y) ']']);

t = runMLP(x,Wx,Wy,actfn2);
```

```
disp(['Y = [' num2str(t) ']']);

figure();
hold on
scatter(x1(1,:),x1(2,:),'g.', 'LineWidth', 5);
scatter(x2(1,:),x2(2,:),'k.', 'LineWidth', 5);

for i=1:size(x,2)
    if t(i) > 0.5
        plot(x(1,i), x(2,i), 'go','MarkerSize',10, 'LineWidth', 2)
    else
        plot(x(1,i), x(2,i), 'ko','MarkerSize',10, 'LineWidth', 2)
    end
end
hold off

title(['MLP with 1 hidden layer with ', num2str(H), ' hidden neurons;
 max iter=', num2str(maxiter), 'mu=', num2str(mu)]);
dim = [.2 .5 .3 .3];
annotation('textbox',dim,'String', {'green circle = \omega_1', 'black
 circle = \omega_2'},'FitBoxToText','on');
```

```
D = [1  1  1  1  1  0  0  0  0  0]
Y = [0.99879     0.96727     0.99844     0.28502     0.28502
 0.28643     0.28506     0.28693     0.28553     0.28737]
```



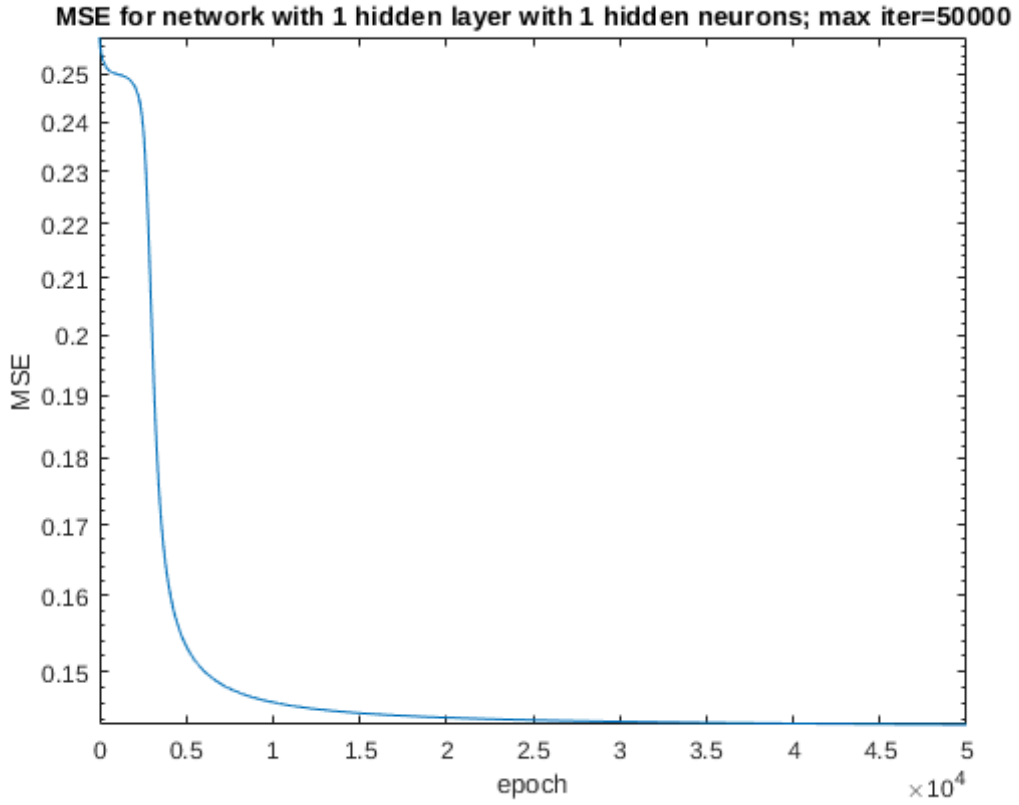

MSE for network with 1 hidden layer with 1 hidden neurons; max iter=50000
MSE
epoch
×10^4
0.25
0.24
0.23
0.22
0.21
0.2
0.19
0.18
0.17
0.16
0.15
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

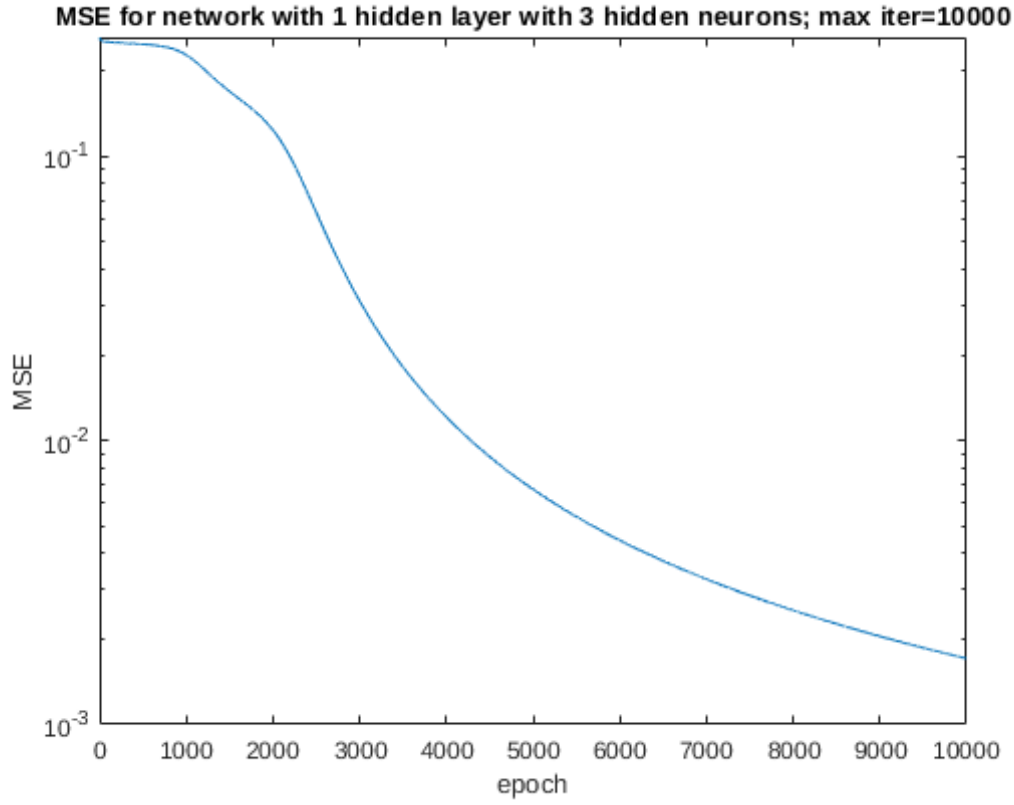

MLP with 1 hidden layer with 1 hidden neurons; max iter=50000mu=0.5
green circle = ω1
black circle = ω2
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
1.4

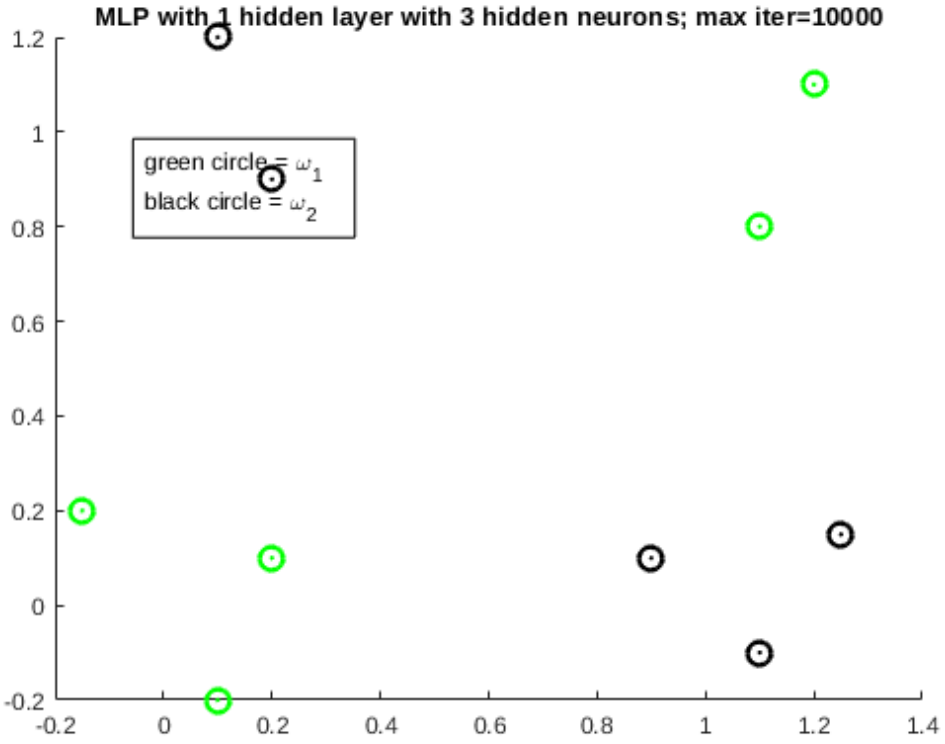
H = 3; maxiter = 10000;

```
p = 2;
H = 3;
m = 1;

maxiter = 10000;
mu = .75;
alpha = 0;
MSEmin = 1e-20;
actfn1 = @(x) (1./(1+exp(-x)));
actfn2 = @(x) (1./(1+exp(-x)));

[Wx,Wy,MSE]=trainMLP(p,H,m,mu,alpha,x,y,actfn1,maxiter,MSEmin);

figure();
semilogy(MSE);
title(['MSE for network with 1 hidden layer with ', num2str(H), '
 hidden neurons; max iter=', num2str(maxiter)]);
xlabel('epoch'); ylabel('MSE');

disp(['D = [' num2str(y) ']']);

t = runMLP(x,Wx,Wy,actfn2);

disp(['Y = [' num2str(t) ']']);

figure();
hold on
scatter(x1(1,:),x1(2,:),'g.', 'LineWidth', 5);
scatter(x2(1,:),x2(2,:),'k.', 'LineWidth', 5);

for i=1:size(x,2)
    if t(i) > 0.5
        plot(x(1,i), x(2,i), 'go','MarkerSize',10, 'LineWidth', 2)
    else
        plot(x(1,i), x(2,i), 'ko','MarkerSize',10, 'LineWidth', 2)
    end
end
hold off

title(['MLP with 1 hidden layer with ', num2str(H), ' hidden neurons;
 max iter=', num2str(maxiter)]);
dim = [.2 .5 .3 .3];
annotation('textbox',dim,'String', {'green circle = \omega_1', 'black
 circle = \omega_2'},'FitBoxToText','on');
```

```
D = [1  1  1  1  1  0  0  0  0  0]
Y = [0.97578     0.95095     0.97402     0.92652     0.99353
 0.016309    0.064865    0.017591    0.051845    0.022489]
```


MSE for network with 1 hidden layer with 1 hidden neurons; max iter=50000
MSE
0.25
0.24
0.23
0.22
0.21
0.2
0.19
0.18
0.17
0.16
0.15
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
epoch
×10^4


MSE for network with 1 hidden layer with 3 hidden neurons; max iter=10000
MSE
10^-1
10^-2
10^-3
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
epoch



Problem 4.2

```
M1 = [0 0];
M2 = [1 1];
M3 = [0 1];
M4 = [1 0];

S = [0.01 0; 0 0.01];
N = 100;

a1 = mvnrnd(M1, S, N);
a2 = mvnrnd(M2, S, N);
a3 = mvnrnd(M3, S, N);
a4 = mvnrnd(M4, S, N);

x1 = [a1' a2'];
x2 = [a3' a4'];

x = [x1 x2];

y = [ones(1,N*2) ones(1,N*2)*0];
```

maxiter = 10000; mu (learning param) = 0.2

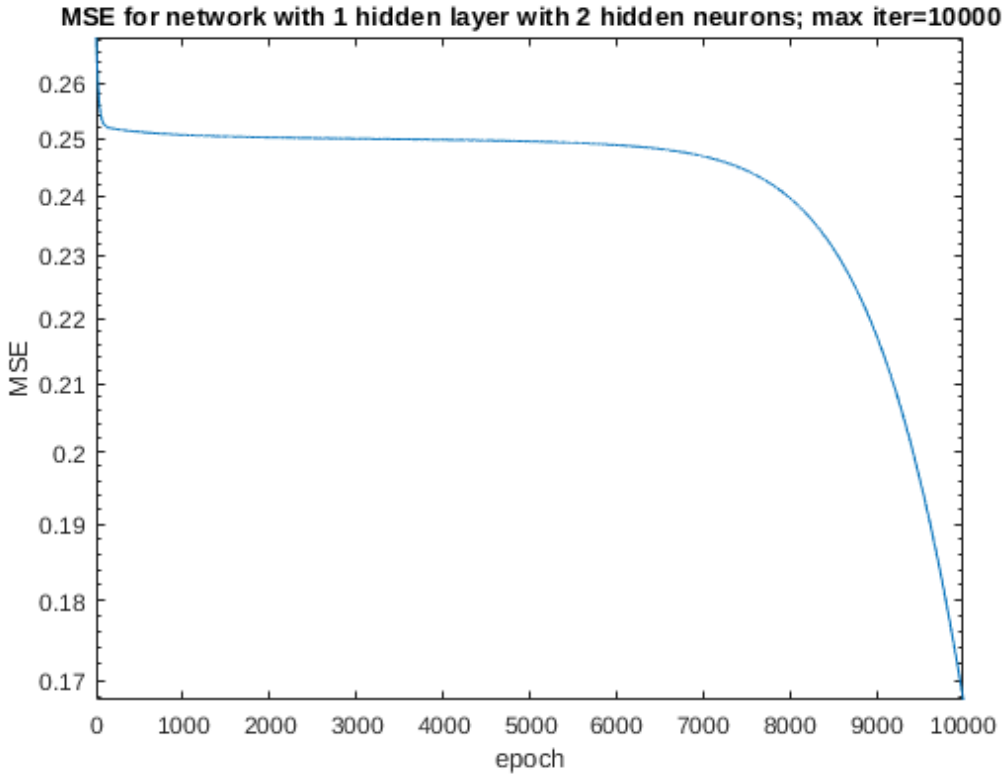
```
p = 2;
H = 2;
```

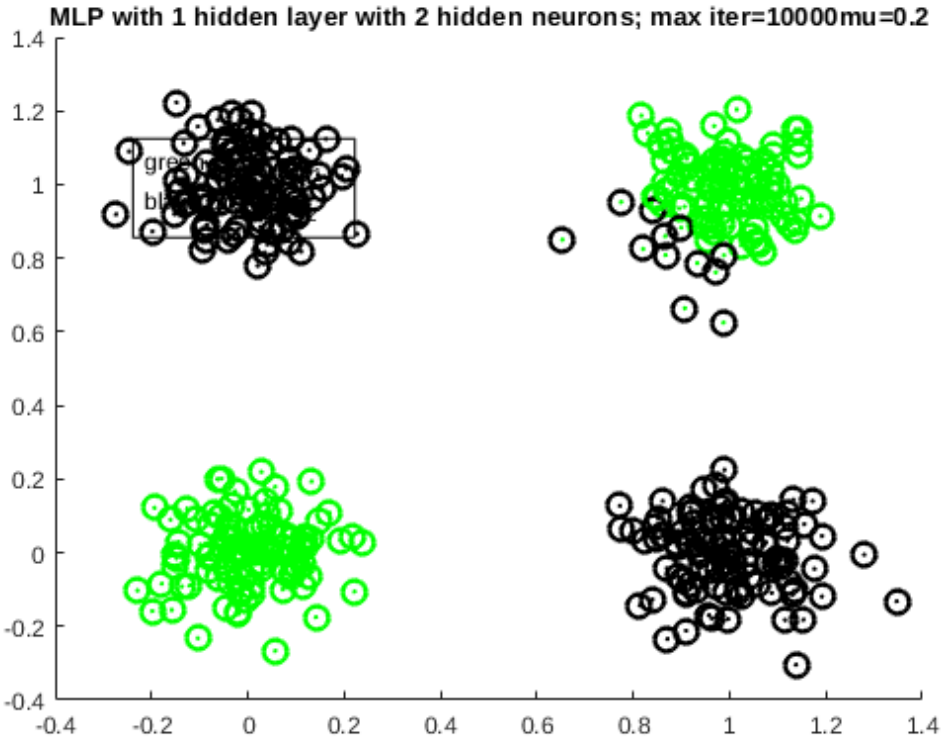
```
m = 1;

maxiter = 10000;
mu = .2;
alpha = 0;
MSEmin = 1e-20;
actfn1 = @(x) (1./(1+exp(-x)));
actfn2 = @(x) (1./(1+exp(-x)));

[Wx,Wy,MSE]=trainMLP(p,H,m,mu,alpha,x,y,actfn1,maxiter,MSEmin);

figure();
semilogy(MSE);
title(['MSE for network with 1 hidden layer with ', num2str(H), '
 hidden neurons; max iter=', num2str(maxiter)]);
xlabel('epoch'); ylabel('MSE');

disp(['D = [' num2str(y) ']']);

t = runMLP(x,Wx,Wy,actfn2);

disp(['Y = [' num2str(t) ']']);

figure();
hold on
scatter(x1(1,:),x1(2,:),'g.', 'LineWidth', 5);
scatter(x2(1,:),x2(2,:),'k.', 'LineWidth', 5);

for i=1:size(x,2)
    if t(i) > 0.5
        plot(x(1,i), x(2,i), 'go','MarkerSize',10, 'LineWidth', 2)
    else
        plot(x(1,i), x(2,i), 'ko','MarkerSize',10, 'LineWidth', 2)
    end
end
hold off

title(['MLP with 1 hidden layer with ', num2str(H), ' hidden neurons;
 max iter=', num2str(maxiter), 'mu=', num2str(mu)]);
dim = [.2 .5 .3 .3];
annotation('textbox',dim,'String', {'green circle = \omega_1', 'black
 circle = \omega_2'},'FitBoxToText','on');
```

```
D = [1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

```
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0]
Y = [0.57364      0.5983     0.63826     0.62952     0.68154
 0.67937     0.59839     0.69402     0.61983     0.62612     0.68634
     0.68342     0.55511     0.73876     0.64879     0.63538
 0.58034     0.67344     0.68631     0.62731     0.72036      0.6499
     0.66993     0.60957      0.6426     0.67457     0.65676
 0.65644     0.59094     0.63453     0.59696     0.73517     0.70389
     0.59565     0.61129     0.51458     0.70022     0.62164
 0.69581     0.59302      0.5885     0.62978     0.57709     0.66833
     0.67353     0.59828     0.60289     0.63273     0.54574
 0.64046     0.62968     0.58686     0.63431     0.62444     0.60311
     0.60597     0.70776     0.61897     0.73429     0.59183
 0.63848     0.67173     0.63071     0.68758     0.59951     0.57335
     0.64387     0.66801     0.55215      0.6876     0.65769
 0.58938      0.6558     0.65336     0.64882     0.59059     0.62909
     0.70238     0.65546     0.62575     0.55052     0.65256
 0.6444     0.65072     0.64559     0.62051     0.54131     0.73087
   0.70809     0.53998     0.71239      0.6718     0.62107     0.64636
     0.67539     0.64576     0.58427     0.53349     0.62362
 0.7019     0.50134      0.5428     0.57107     0.53177     0.50805
   0.51119     0.49505     0.51637     0.52284     0.51965      0.5546
     0.45851     0.53635     0.53955     0.48702     0.54862
 0.55742      0.5262     0.55497     0.58525     0.54249     0.54278
   0.52887     0.53253     0.54143     0.54167     0.50993     0.55661
     0.54332      0.5245     0.54601     0.53486     0.53883
 0.50148     0.50669     0.44693      0.5404      0.5865      0.5115
   0.53644     0.48805     0.52601     0.55266     0.51995     0.52835
     0.55445     0.57447     0.53966      0.5198     0.52779
 0.53423      0.5064     0.52935     0.55162     0.51089     0.51449
   0.51659     0.54673     0.57539     0.53806     0.56671     0.46595
     0.53519     0.53612     0.50927      0.5415     0.53437
 0.4998     0.50268     0.50272     0.58033     0.53996     0.54018
   0.54467     0.56054      0.5315     0.48683     0.53397     0.53332
     0.49723     0.55941     0.55531     0.53079     0.53117
 0.51653     0.54825     0.55911     0.51984     0.55097     0.52053
     0.50755     0.48555     0.55116     0.55312     0.51351
 0.5188      0.5318     0.52672     0.47192     0.47748     0.41387
   0.39125     0.39717     0.40458     0.39873      0.3917     0.40006
     0.39366      0.3914     0.40065     0.39652      0.4019
 0.39706     0.39189     0.39411     0.39172     0.39161     0.39224
     0.39126     0.40097     0.39603      0.3922       0.397
 0.4192      0.3924     0.39509     0.40128     0.39574     0.39469
    0.3936     0.39514     0.39146     0.39139     0.39128     0.39364
     0.39339     0.39156     0.39767     0.39508     0.39278
 0.39141     0.39135     0.40015     0.40238     0.39499      0.4056
   0.40675     0.39887     0.39227     0.39975     0.41441     0.39357
     0.39145     0.39133     0.39653     0.39224     0.39137
 0.39469     0.39259     0.39479      0.3993     0.39197     0.39333
```

```
  0.39124     0.40065     0.39381     0.39362     0.39942     0.39167
    0.39782     0.39586     0.40642     0.39221     0.39298
0.39145     0.39371     0.39392     0.39907      0.3943     0.39389
  0.40962     0.40588     0.39595     0.40104     0.39126     0.39211
    0.40152     0.39732     0.39142     0.39455     0.39126
0.39414     0.39126     0.39388     0.39453     0.39167     0.39131
  0.39367     0.39128     0.39155     0.39247     0.39631     0.39439
    0.40015      0.3964     0.39255     0.39142     0.39513
0.3913     0.39427     0.39243     0.39167     0.39232     0.39125
    0.39394     0.39753      0.3913     0.39122     0.39582
0.3936     0.39122     0.39118     0.39667     0.39449     0.39287
  0.40754     0.39168     0.39123     0.39129     0.39118     0.40048
    0.39238     0.41774     0.39177     0.39303     0.39167
0.40604     0.39128     0.39479     0.40795     0.39915     0.39157
  0.39189     0.39666     0.39164     0.39555     0.39902     0.40295
    0.39121     0.41303     0.39212     0.39264     0.39173
0.40088     0.39446     0.39891     0.39485     0.39207     0.39122
  0.39356     0.39295     0.40848     0.39845     0.39198     0.39153
    0.40343     0.39188     0.39573     0.39152     0.41017
0.39157     0.39131     0.39816     0.39134     0.40158     0.39149
  0.39168      0.3955     0.39522     0.39118     0.39545     0.39119
    0.39339     0.42093     0.39361       0.406     0.39765
0.3959     0.39192     0.39179     0.39805     0.41464     0.39753
 0.40789     0.39544      0.4128     0.39499     0.39524     0.39634
   0.39998]
```





maxiter = 10000; mu (learning param) = 0.75

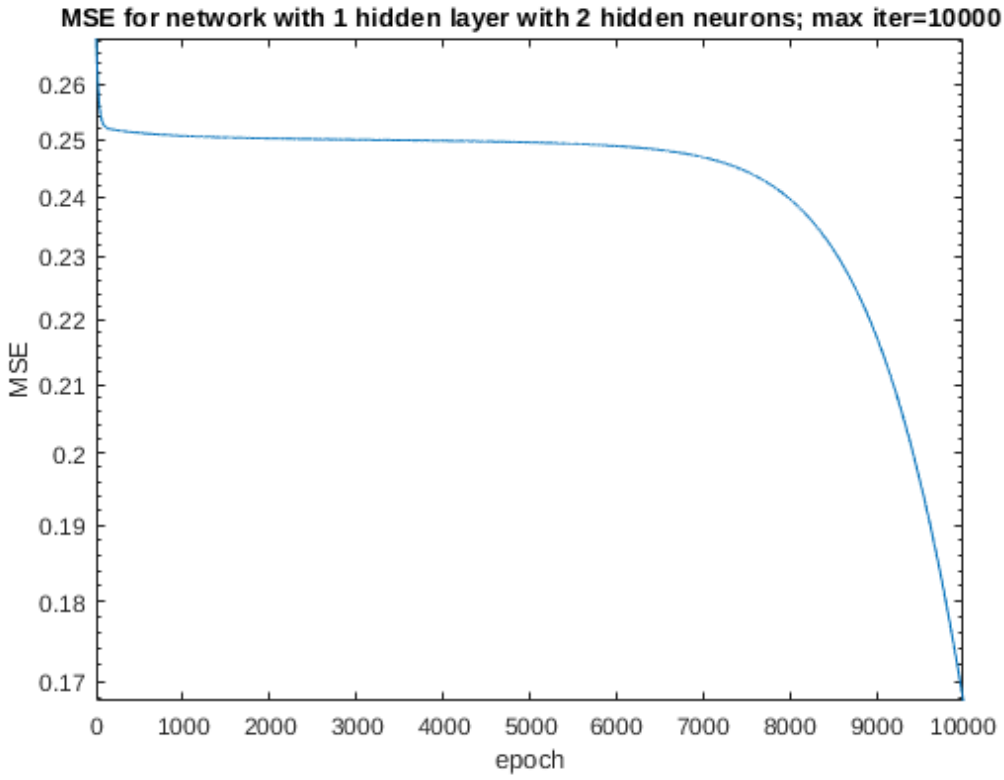
```
p = 2;
H = 2;
m = 1;

maxiter = 10000;
mu = .75;
alpha = 0;
MSEmin = 1e-20;
actfn1 = @(x) (1./(1+exp(-x)));
actfn2 = @(x) (1./(1+exp(-x)));

[Wx,Wy,MSE]=trainMLP(p,H,m,mu,alpha,x,y,actfn1,maxiter,MSEmin);

figure();
semilogy(MSE);
title(['MSE for network with 1 hidden layer with ', num2str(H), '
 hidden neurons; max iter=', num2str(maxiter)]);
xlabel('epoch'); ylabel('MSE');

disp(['D = [' num2str(y) ']']);

t = runMLP(x,Wx,Wy,actfn2);

disp(['Y = [' num2str(t) ']']);
```

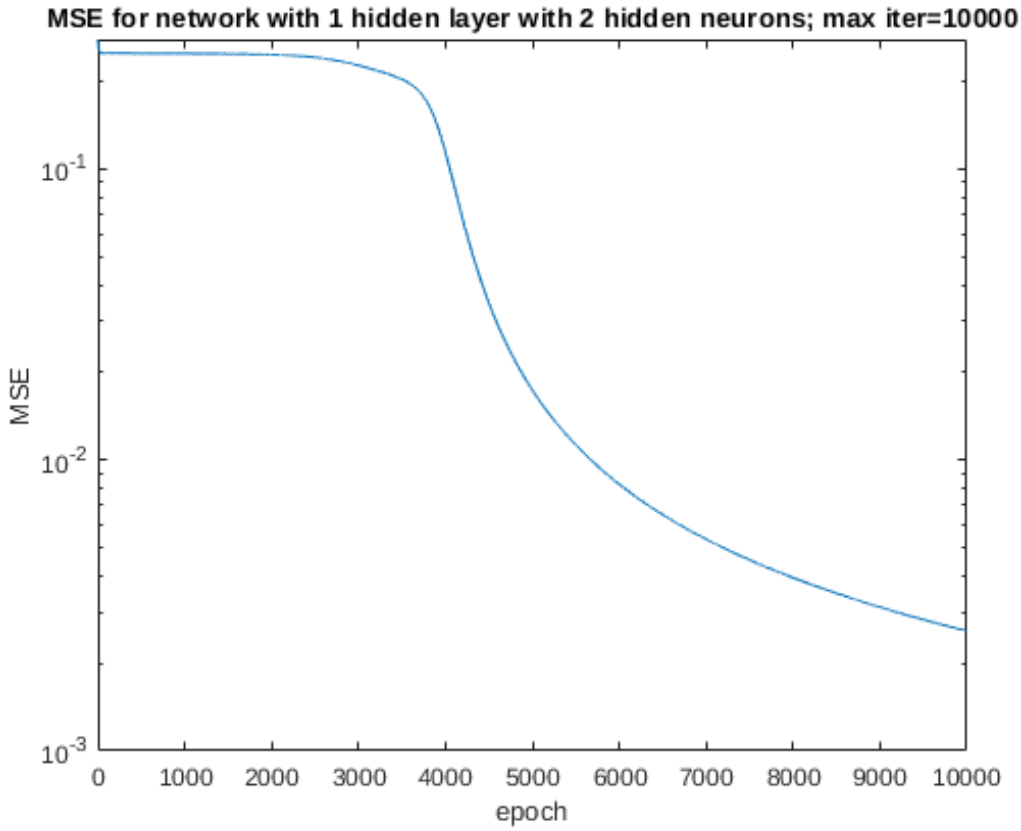
```
figure();
hold on
scatter(x1(1,:),x1(2,:),'g.', 'LineWidth', 5);
scatter(x2(1,:),x2(2,:),'k.', 'LineWidth', 5);

for i=1:size(x,2)
    if t(i) > 0.5
        plot(x(1,i), x(2,i), 'go','MarkerSize',10, 'LineWidth', 2)
    else
        plot(x(1,i), x(2,i), 'ko','MarkerSize',10, 'LineWidth', 2)
    end
end
hold off

title(['MLP with 1 hidden layer with ', num2str(H), ' hidden neurons;
 max iter=', num2str(maxiter), 'mu=', num2str(mu)]);
dim = [.2 .5 .3 .3];
annotation('textbox',dim,'String', {'green circle = \omega_1', 'black
 circle = \omega_2'},'FitBoxToText','on');
```

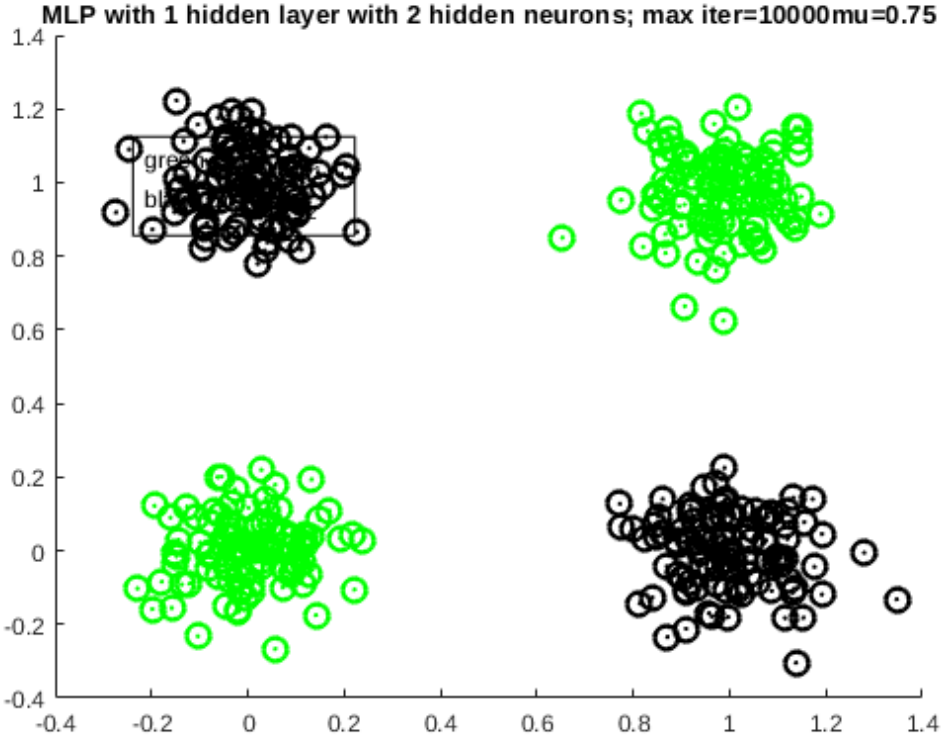
```
D = [1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0]
Y = [0.93217     0.95015     0.96557      0.9631      0.97351
 0.97323     0.95019     0.97492     0.95988     0.96204     0.97409
     0.97374     0.91049     0.97811     0.96807     0.96481
 0.93805     0.97243     0.97408     0.96242     0.97709      0.9683
     0.97191     0.95575     0.96666     0.97258     0.96968
 0.96962     0.94571     0.96458     0.94941     0.97794     0.97585
     0.94864     0.95651     0.81003     0.97552     0.96053
 0.9751      0.94706     0.94411     0.96319     0.93533     0.97166
   0.97244     0.95015     0.95258     0.96406     0.89519     0.96613
     0.96317     0.94295     0.96452     0.96149      0.9527
 0.95409     0.97617     0.95957      0.9779     0.94632     0.96562
   0.97218     0.96346     0.97423     0.95085     0.93191     0.96696
     0.97161     0.90604     0.97423     0.96985     0.94469
 0.9695      0.96903     0.96808     0.94552     0.96297     0.97571
     0.96944     0.96192      0.9034     0.96886     0.96708
```

0.96848 0.96736 0.96013 0.88685 0.97772 0.9762
0.88413 0.97653 0.97219 0.96034 0.96753
0.9727 0.96739 0.94111 0.86938 0.96121 0.97567
0.95508 0.97704 0.98094 0.97426 0.96129 0.96383
0.94691 0.96717 0.97076 0.96892 0.97911
0.78007 0.97544 0.9763 0.93248 0.9781 0.97952
0.97214 0.97917 0.98188 0.97698 0.97696
0.97311 0.97444 0.97662 0.9767 0.96284 0.97934
0.97703 0.97153 0.97769 0.97511 0.97607
0.95483 0.9602 0.64406 0.97637 0.98195 0.96386
0.97558 0.93313 0.97187 0.97879 0.96921
0.97293 0.97906 0.98123 0.97618 0.96897 0.97271
0.9748 0.96008 0.97327 0.97863 0.96384
0.96604 0.96714 0.97783 0.98127 0.97606 0.98054
0.84087 0.97506 0.97542 0.96248 0.97674
0.97496 0.95269 0.95661 0.95664 0.9816 0.97632
0.97653 0.97741 0.97988 0.97414 0.93127
0.97484 0.97457 0.94975 0.97979 0.97914 0.97391
0.97397 0.96752 0.97807 0.9797 0.9692
0.97858 0.96955 0.96091 0.92761 0.97858 0.97888
0.96569 0.96845 0.97404 0.97226 0.88016
0.90358 0.088021 0.03132 0.047473 0.078663 0.043839
0.032511 0.056695 0.037518 0.031048 0.060028
0.045026 0.063969 0.040465 0.032821 0.038205 0.031543
0.032073 0.033507 0.031018 0.061512 0.043637
0.033381 0.046192 0.11092 0.033852 0.036865 0.062301
0.041747 0.040419 0.034515 0.040701 0.031999
0.031166 0.031248 0.037019 0.036604 0.032084 0.048244
0.036965 0.033026 0.032023 0.031037 0.057807
0.066935 0.040589 0.083407 0.089807 0.04412 0.033539
0.055928 0.15064 0.036998 0.031317 0.031702
0.03954 0.03361 0.031679 0.036282 0.034511 0.040281
0.054478 0.033076 0.036469 0.031196 0.059188 0.034775
0.034397 0.053939 0.032517 0.041885 0.038309
0.08681 0.03232 0.03349 0.031781 0.036969 0.037715
0.05302 0.03897 0.037925 0.10809 0.061256
0.042759 0.048917 0.03126 0.032276 0.049977 0.041005
0.031157 0.036171 0.031338 0.035193 0.031122
0.0349 0.039502 0.03234 0.031569 0.036833 0.031358
0.032128 0.031465 0.040035 0.035199 0.051033 0.039855
0.031629 0.029981 0.035868 0.029484 0.035156
0.031856 0.030306 0.031502 0.029495 0.034316 0.043128
0.029667 0.029614 0.038304 0.033508 0.02964
0.029624 0.040633 0.035145 0.032672 0.081061 0.030313
0.029316 0.029753 0.029451 0.051489 0.031494
0.15696 0.030409 0.032995 0.030183 0.070958 0.029695
0.035579 0.082847 0.043923 0.030243 0.030624 0.040771
0.03032 0.036846 0.047182 0.06002 0.02972
0.11637 0.030822 0.031722 0.030698 0.052922 0.035587
0.043423 0.036604 0.03105 0.02955 0.033426 0.032331
0.085284 0.04472 0.030949 0.030124 0.062162
0.0306 0.038549 0.029959 0.074161 0.030209 0.029548
0.044869 0.029674 0.055189 0.030001 0.030356 0.03675
0.036221 0.029466 0.036643 0.0296 0.033268

```
0.11931    0.034024    0.074685    0.040838    0.038645    0.030906
 0.030362    0.041655    0.090826    0.043321    0.066746    0.036607
    0.11176    0.036505    0.037023    0.038215    0.049539]
```

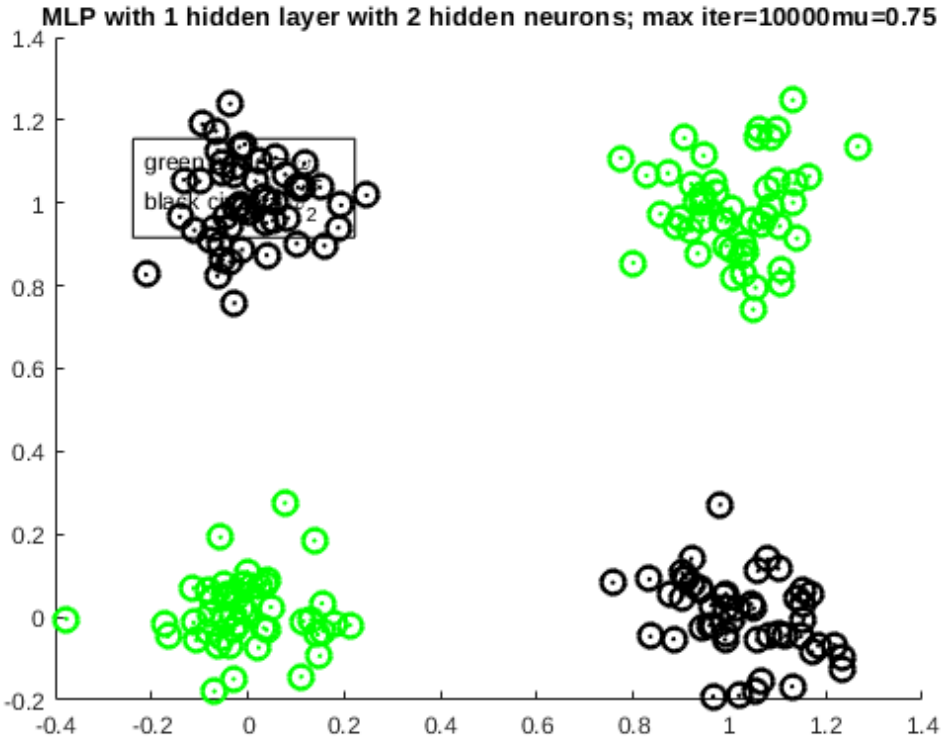



MSE for network with 1 hidden layer with 2 hidden neurons; max iter=10000
MSE
epoch
10^{-1}
10^{-2}
10^{-3}
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000


MLP with 1 hidden layer with 2 hidden neurons; max iter=10000mu=0.75
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
1.4

Produce 50 more vectors, use calculated weights to classify.

```
M1 = [0 0];
M2 = [1 1];
M3 = [0 1];
M4 = [1 0];

S = [0.01 0; 0 0.01];
N = 50;

a1 = mvnrnd(M1, S, N);
a2 = mvnrnd(M2, S, N);
a3 = mvnrnd(M3, S, N);
a4 = mvnrnd(M4, S, N);

x1 = [a1' a2'];
x2 = [a3' a4'];

x = [x1 x2];

y = [ones(1,N*2) ones(1,N*2)*0];

t = runMLP(x,Wx,Wy,actfn2);

figure();
hold on
scatter(x1(1,:),x1(2,:),'g.', 'LineWidth', 5);
scatter(x2(1,:),x2(2,:),'k.', 'LineWidth', 5);

for i=1:size(x,2)
    if t(i) > 0.5
        plot(x(1,i), x(2,i), 'go','MarkerSize',10, 'LineWidth', 2)
    else
        plot(x(1,i), x(2,i), 'ko','MarkerSize',10, 'LineWidth', 2)
    end
end
hold off

title(['MLP with 1 hidden layer with ', num2str(H), ' hidden neurons;
 max iter=', num2str(maxiter), 'mu=', num2str(mu)]);
dim = [.2 .5 .3 .3];
annotation('textbox',dim,'String', {'green circle = \omega_1', 'black
 circle = \omega_2'},'FitBoxToText','on');
```



Published with MATLAB® R2018b