Contents

Vincent Purcell - HW 4 - ECE487

```
clear; clc; close all;
```

Problem 4.1

```
% Classes given from prompt in text
A1 = [0.1 0.2; 1.2 0.9];
B1 = [1.1 1.2; 0.8 1.1];
C1 = [0.9 1.1 1.25; 0.1 -0.1 0.15];
D1 = [-0.15 0.1 0.2; 0.2 -0.2 0.1];

% Seed RNG
rng(1)

% Call multilayer perceptron function
[P1_1, P2_1, aa_1, span_1] = ML_Perceptron(A1, B1, C1, D1);

figure;
hold on
grid on
scatter([A1(1,:) C1(1,:)],[A1(2,:) C1(2,:)],60,'x','y')
scatter([B1(1,:) D1(1,:)],[B1(2,:) D1(2,:)],60,'o','r')
mesh(P1_1,P2_1,reshape(aa_1,length(span_1),length(span_1))-5);
xlabel('X Axis')
ylabel('Y Axis')
title('Multilayer Perceptron - Problem 4.1')
colormap parula
legend('Class 1','Class 2')
hold off
snapnow
```

Problem 4.2

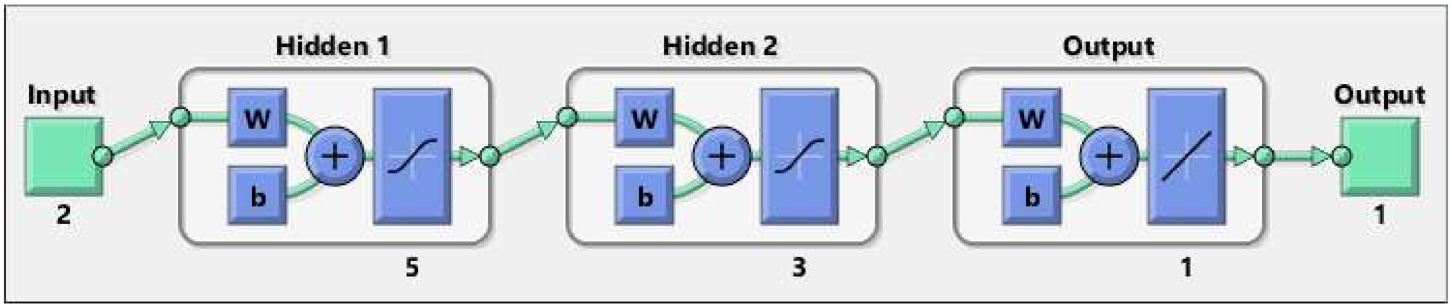
```
% Input variables from prompt in text
cov = [0.01, 0.0; 0.0, 0.01];
m1 = [0,1];
m2 = [1,1];
m3 = [1,0];
m4 = [0,0];

% Generate classes with gaussian distribution
A2 = mvnrnd(m1,cov,100)';
B2 = mvnrnd(m2,cov,100)';
C2 = mvnrnd(m3,cov,100)';
D2 = mvnrnd(m4,cov,100)';

% Seed RNG
rng(1);
```

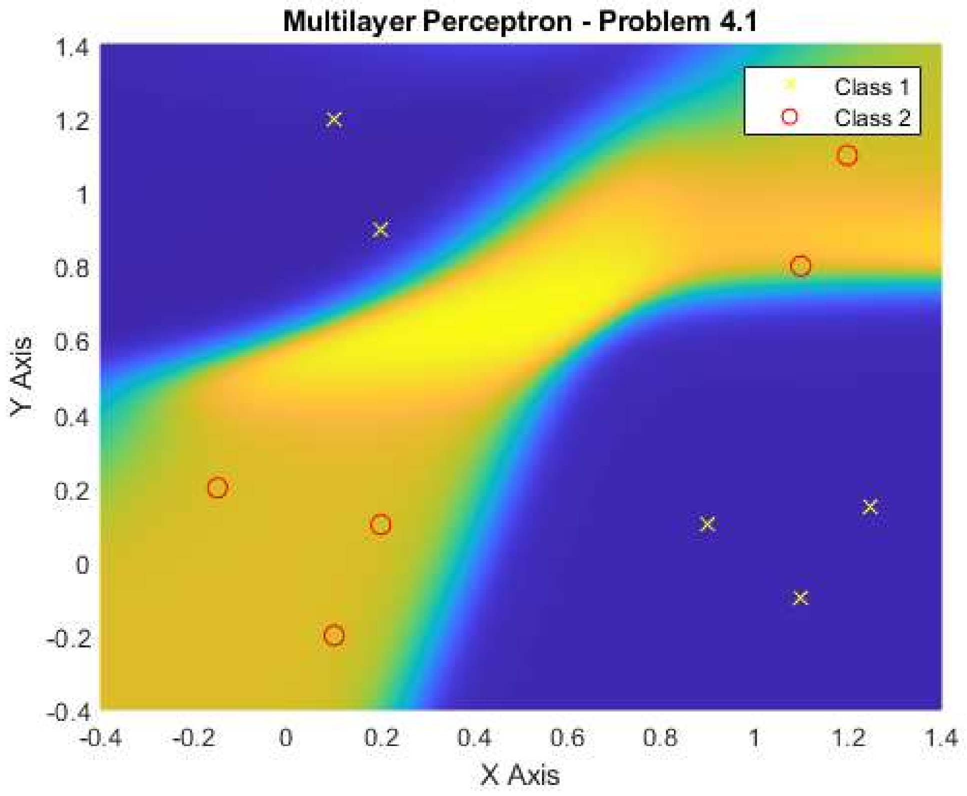
```
% Call multilayer perceptron function
[P1_2, P2_2, aa_2, span_2] = ML_Perceptron(A2, B2, C2, D2);

% Plot Perceptron and Data Points
figure;
hold on
grid on
scatter([A2(1,:) C2(1,:)],[A2(2,:) C2(2,:)],60,'x','y')
scatter([B2(1,:) D2(1,:)],[B2(2,:) D2(2,:)],60,'o','r')
mesh(P1_2,P2_2,reshape(aa_2,length(span_2),length(span_2))-5);
xlabel('X Axis')
ylabel('Y Axis')
title('Multilayer Perceptron - Problem 4.2')
colormap parula
legend('Class 1','Class 2')
hold off
snapnow
```

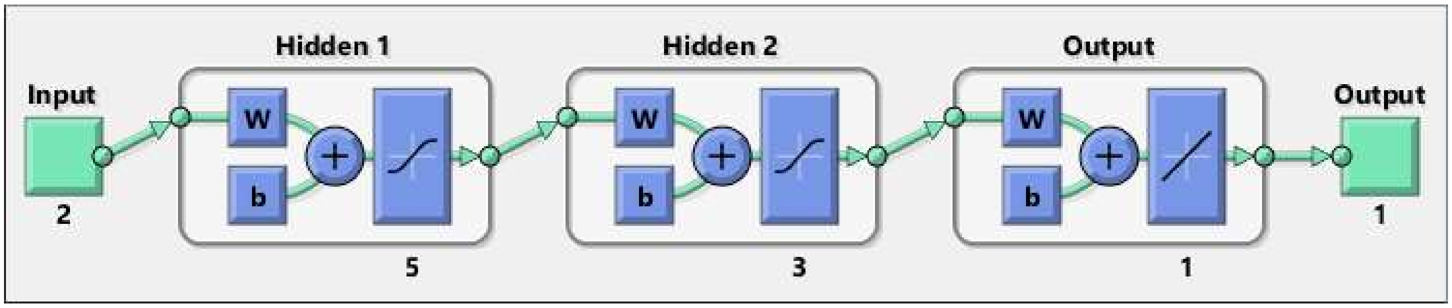
Back Propogation Multilayer Perceptron

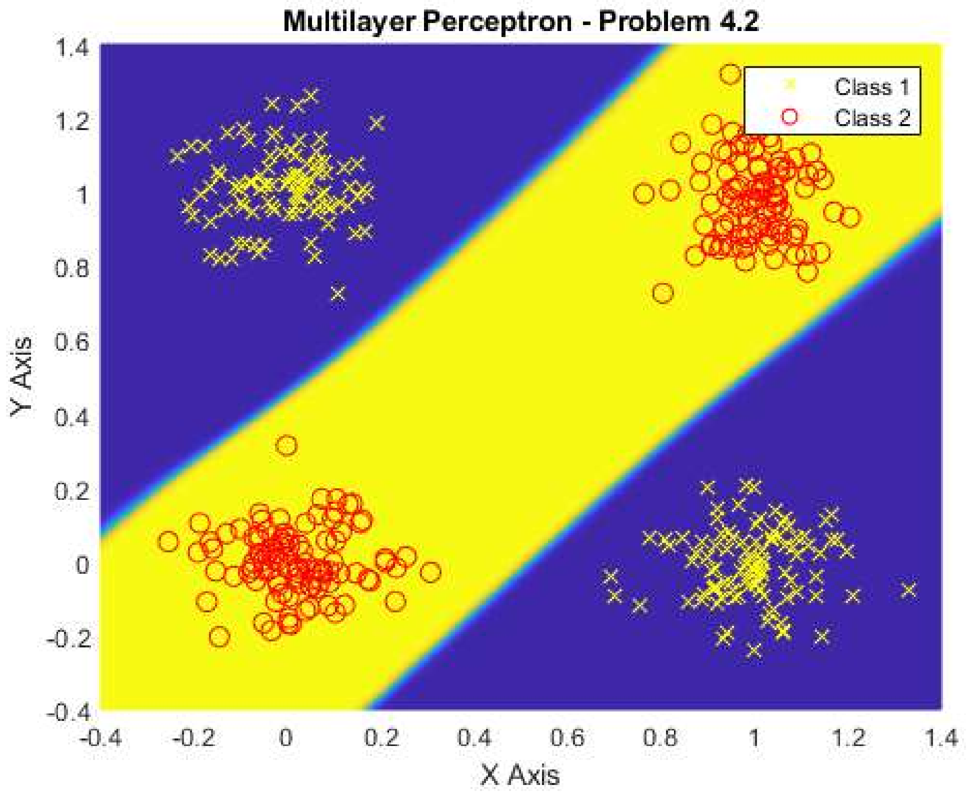
Multilayer Perceptron function that uses back propogation and a feed forward nueral network function from the MATLAB Deep Learning Toolbox

```
% Function is similar to code from a document that details examples of
% using the deep learning toolbox. Those examples are found at:
% http://lab.fs.uni-lj.si/lasin/wp/IMIT_files/neural/NN-examples.pdf
 function [P1, P2, aa, span] = ML_Perceptron(A, B, C, D)
    % Class descriptors for Input Features
    a = -1;
    c = -1;
    b =  1;
    d =  1;

    % Combine all data points
    P = [A B C D];

    % Targets for predicting classes
    T = [repmat(a,1,length(A)) repmat(b,1,length(B)) ...
         repmat(c,1,length(C)) repmat(d,1,length(D)) ];

    % Initialize Nueral Network
    net = feedforwardnet([5 3]);

    net.divideParam.trainRatio = 1; % training set [%]
    net.divideParam.valRatio = 0; % validation set [%]
    net.divideParam.testRatio = 0; % test set [%]

    % train a neural network with Target values
    [net,~,~,~] = train(net,P,T);
    % show network
    view(net)
    snapnow
    % generate a grid
    span = -0.4:.005:1.4;
    [P1,P2] = meshgrid(span,span);
    pp = [P1(:) P2(:)]';

    % simulate neural network on a grid
    aa = net(pp);
 end
```


Hidden 1
Hidden 2
Output
Input
Output
W
b
2
5
3
1
1


Multilayer Perceptron - Problem 4.1
Class 1
Class 2
X Axis
Y Axis
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
1.4





Published with MATLAB® R2019b