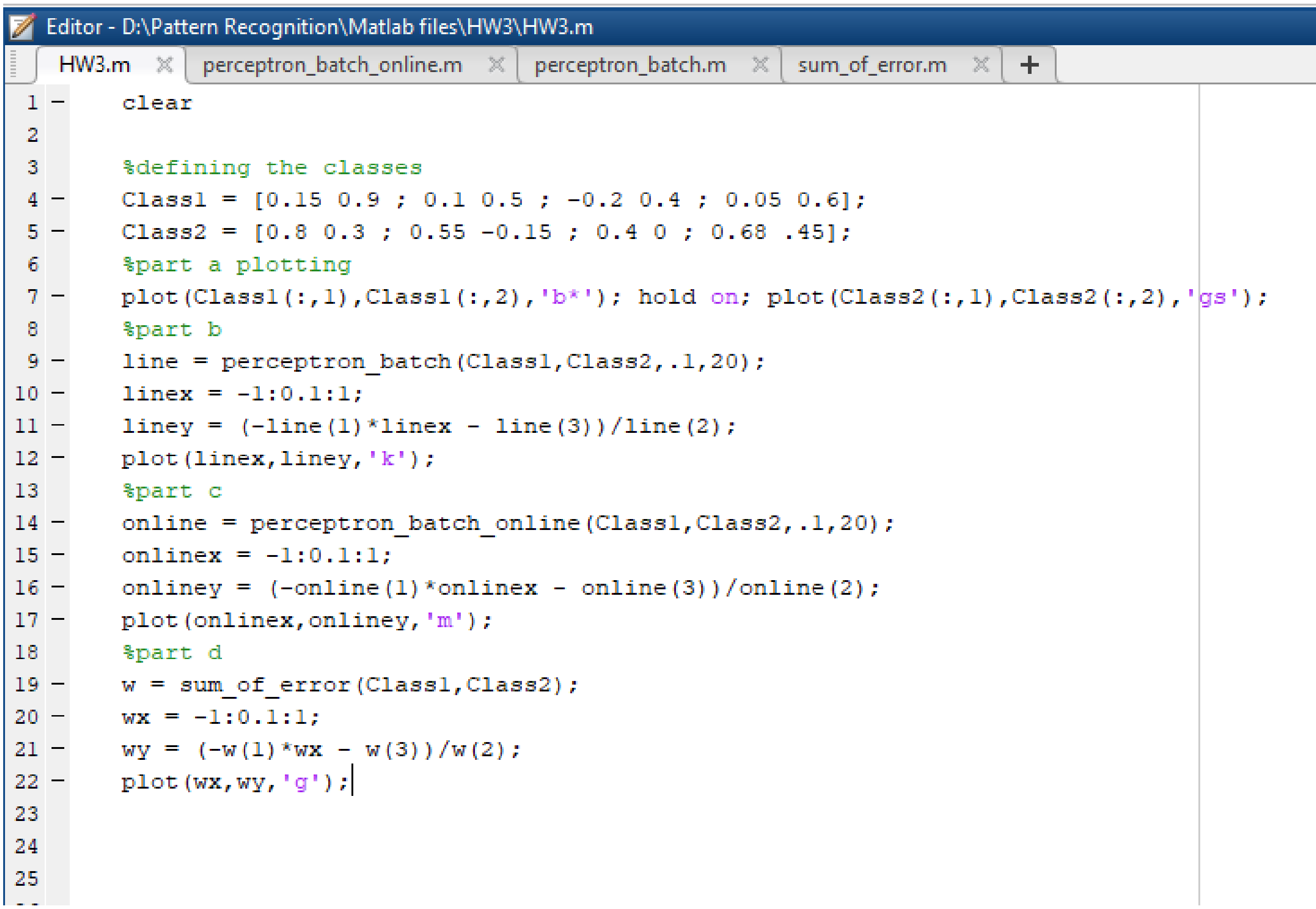
Shane Bolding 6/18/2019

Homework 2

Main Code


Editor - D:\Pattern Recognition\Matlab files\HW3\HW3.m
HW3.m
perceptron_batch_online.m
perceptron_batch.m
sum_of_error.m
+

```
clear

%defining the classes
Class1 = [0.15 0.9 ; 0.1 0.5 ; -0.2 0.4 ; 0.05 0.6];
Class2 = [0.8 0.3 ; 0.55 -0.15 ; 0.4 0 ; 0.68 .45];
%part a plotting
plot(Class1(:,1),Class1(:,2),'b*'); hold on; plot(Class2(:,1),Class2(:,2),'gs');
%part b
line = perceptron_batch(Class1,Class2,.1,20);
linex = -1:0.1:1;
liney = (-line(1)*linex - line(3))/line(2);
plot(linex,liney,'k');
%part c
online = perceptron_batch_online(Class1,Class2,.1,20);
onlinex = -1:0.1:1;
onliney = (-online(1)*onlinex - online(3))/online(2);
plot(onlinex,onliney,'m');
%part d
w = sum_of_error(Class1,Class2);
wx = -1:0.1:1;
wy = (-w(1)*wx - w(3))/w(2);
plot(wx,wy,'g');
```

Figure 1 Main code

Part a) plotting Class1 and Class2

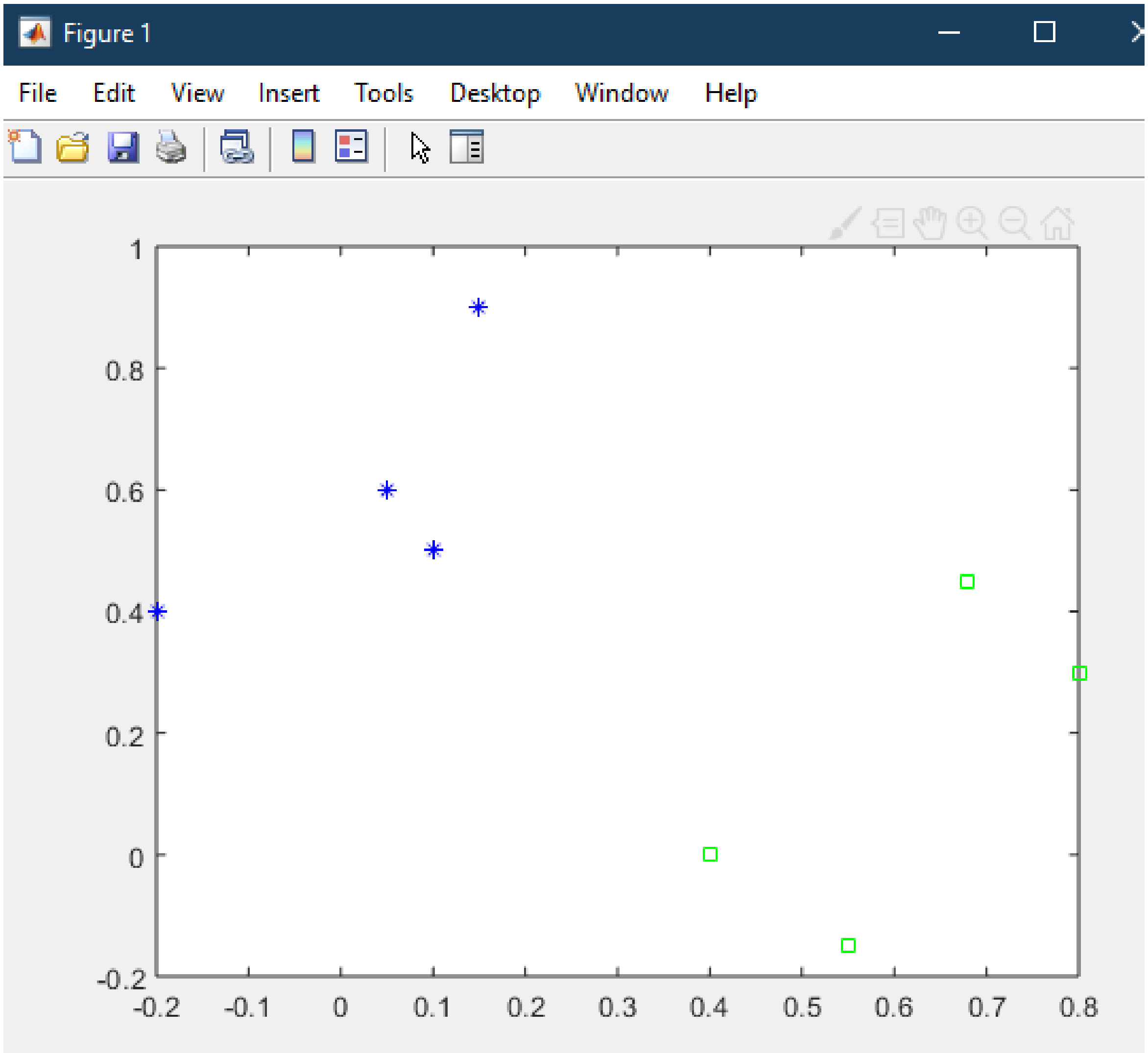


Figure 2 part a

Part b) using the batch version given (black line)

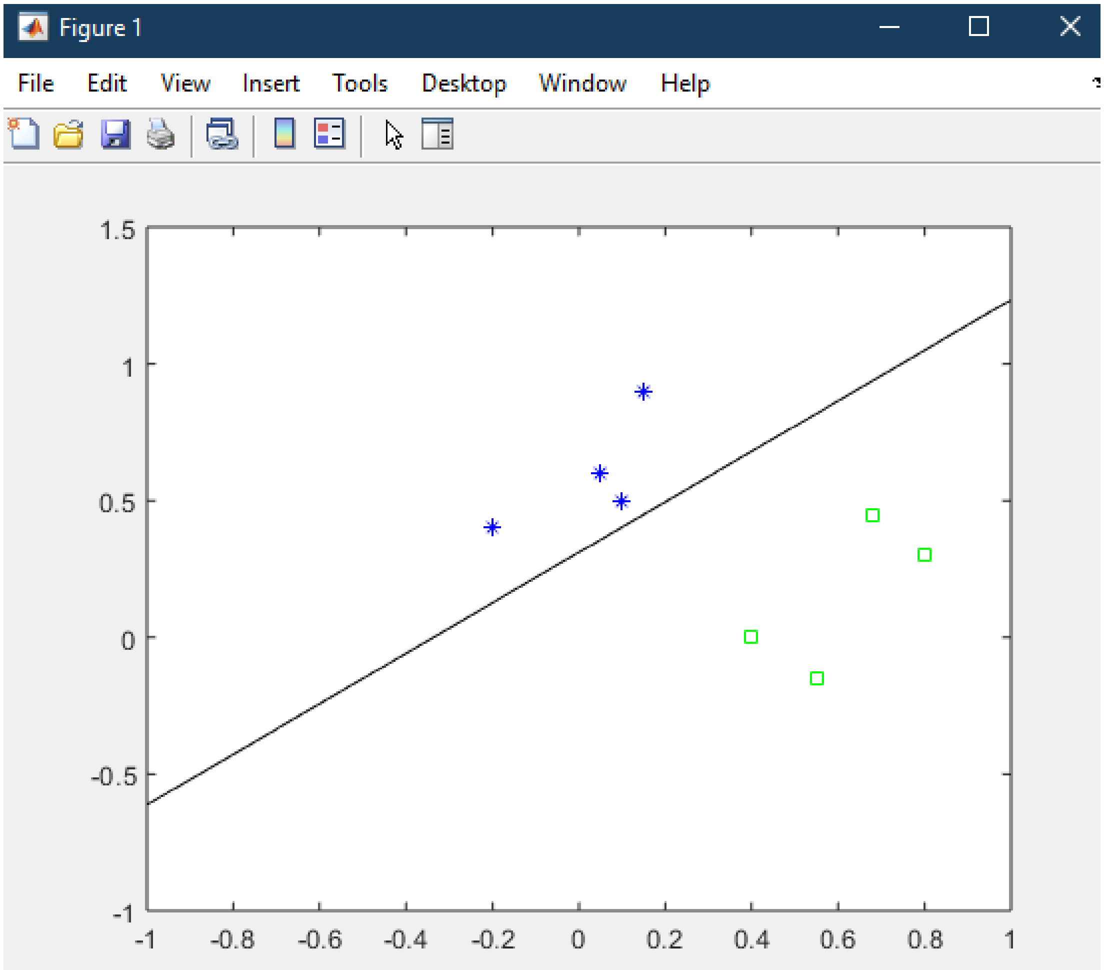
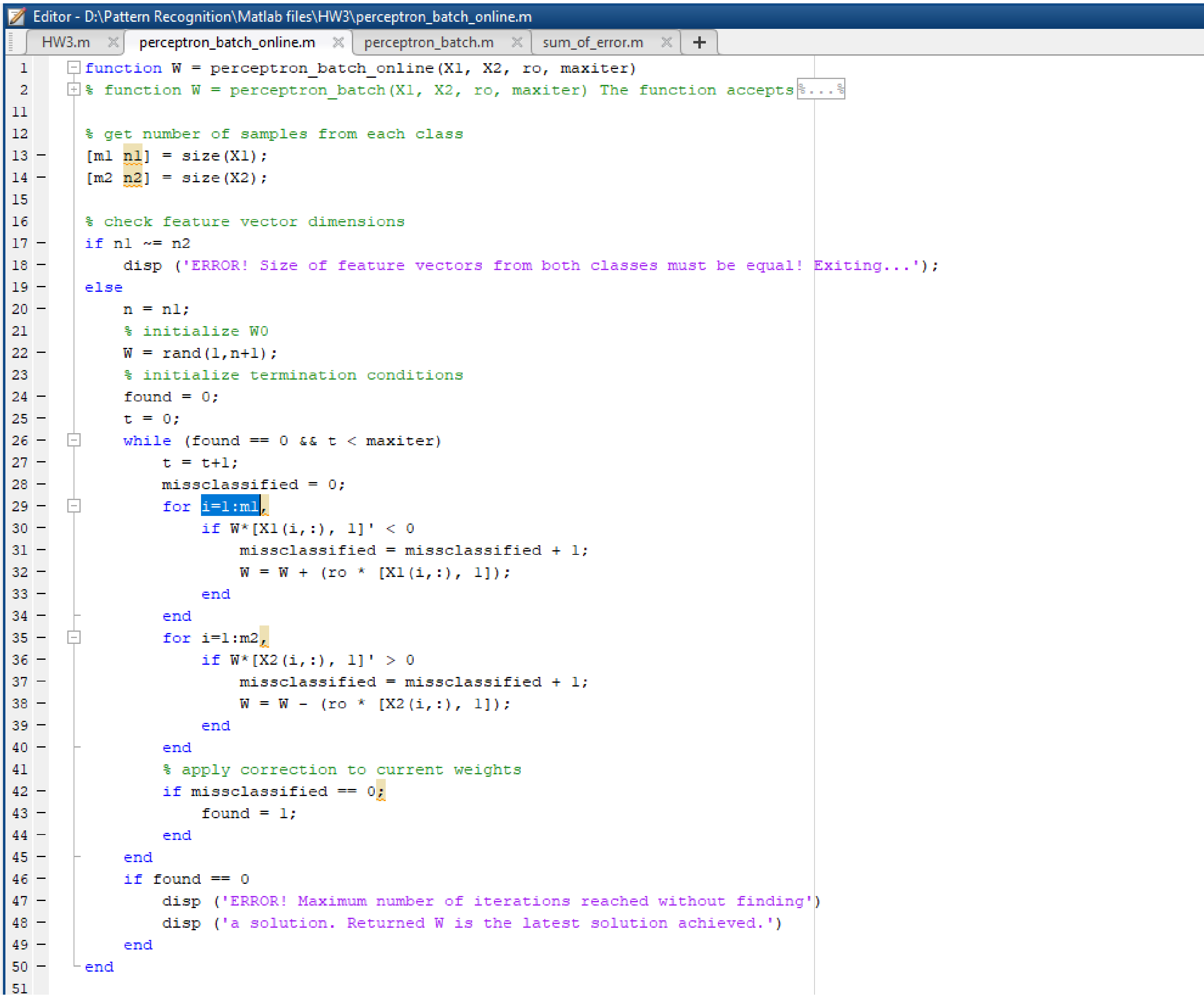


Figure 3 part b

Part c) using the online perceptron algorithm (magenta line)


Editor - D:\Pattern Recognition\Matlab files\HW3\perceptron_batch_online.m
HW3.m
perceptron_batch_online.m
perceptron_batch.m
sum_of_error.m
+

```
function W = perceptron_batch_online(X1, X2, ro, maxiter)
% function W = perceptron_batch(X1, X2, ro, maxiter) The function accepts %...%

% get number of samples from each class
[m1 n1] = size(X1);
[m2 n2] = size(X2);

% check feature vector dimensions
if n1 ~= n2
    disp ('ERROR! Size of feature vectors from both classes must be equal! Exiting...');
else
    n = n1;
    % initialize W0
    W = rand(1,n+1);
    % initialize termination conditions
    found = 0;
    t = 0;
    while (found == 0 && t < maxiter)
        t = t+1;
        missclassified = 0;
        for i=1:m1,
            if W*[X1(i,:), 1]' < 0
                missclassified = missclassified + 1;
                W = W + (ro * [X1(i,:), 1]);
            end
        end
        for i=1:m2,
            if W*[X2(i,:), 1]' > 0
                missclassified = missclassified + 1;
                W = W - (ro * [X2(i,:), 1]);
            end
        end
        % apply correction to current weights
        if missclassified == 0;
            found = 1;
        end
    end
    if found == 0
        disp ('ERROR! Maximum number of iterations reached without finding')
        disp ('a solution. Returned W is the latest solution achieved.')
    end
end

```

Figure 4 part c full code

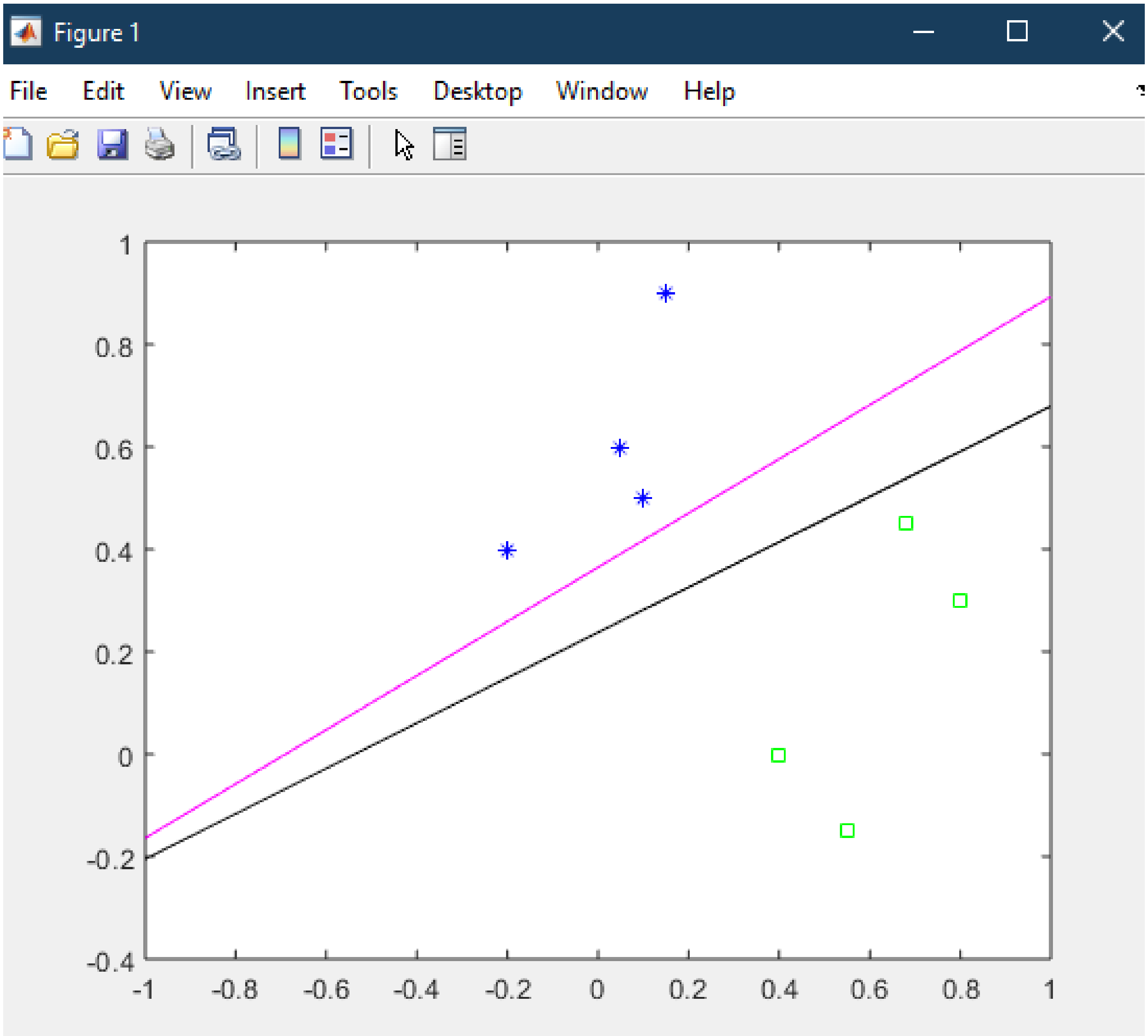
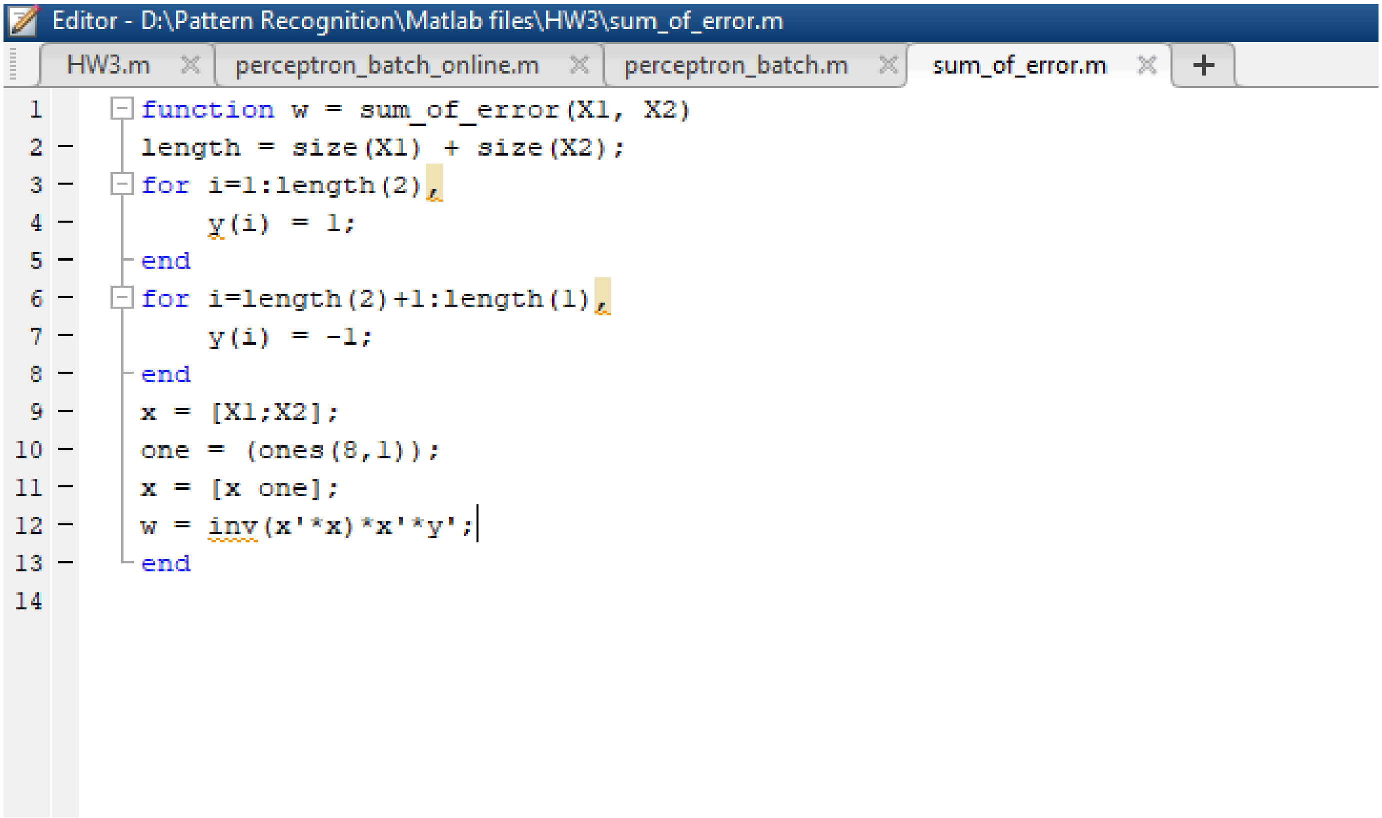


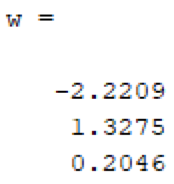
Figure 5 part c plot

Part d) using the sum of errors algorithm (green line)


Editor - D:\Pattern Recognition\Matlab files\HW3\sum_of_error.m
HW3.m
perceptron_batch_online.m
perceptron_batch.m
sum_of_error.m
+

```
function w = sum_of_error(X1, X2)
length = size(X1) + size(X2);
for i=1:length(2),
    y(i) = 1;
end
for i=length(2)+1:length(1),
    y(i) = -1;
end
x = [X1;X2];
one = (ones(8,1));
x = [x one];
w = inv(x'*x)*x'*y';
end
```

Figure 6 part d full code



```
w =

   -2.2209
    1.3275
    0.2046
```

Figure 7 equation of the line that separates the two batches

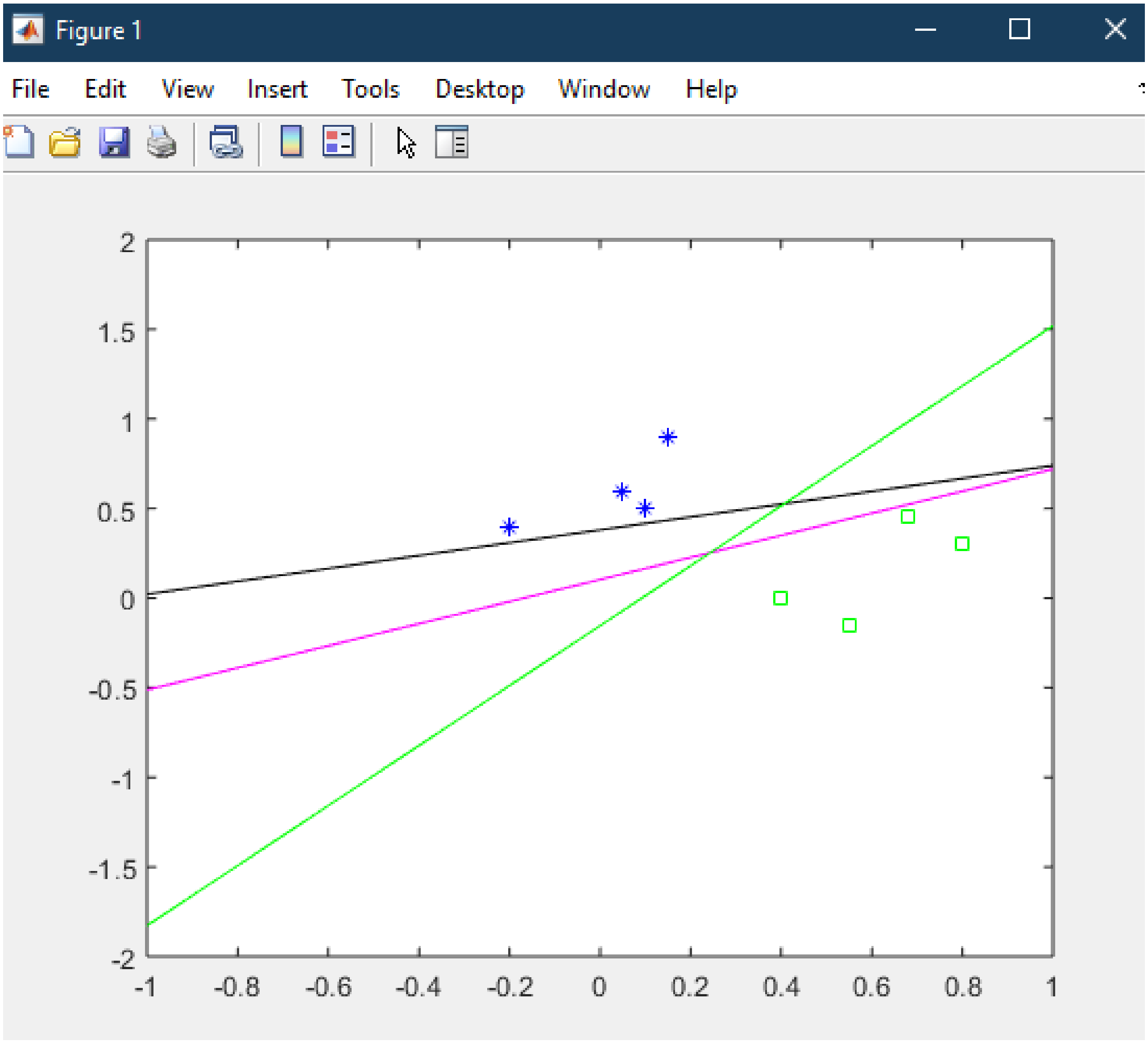


Figure 8 plot of all three different lines to separate batches

All the lines did separate the classes like they were programmed to do; however, the one best solution is the sum of errors line. This line is the best one because it created a line that shot straight down the middle of both batches giving our future features a greater chance to be correct.