EEE – 485

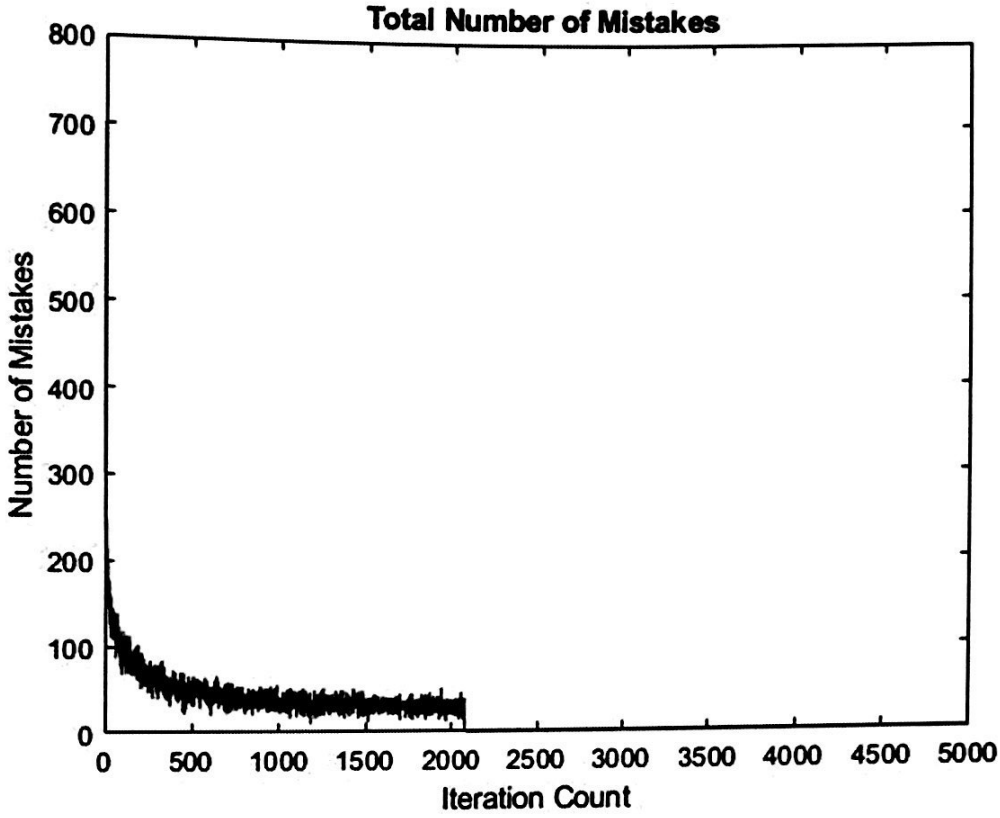
Statistical Learning
and
Data Analytics

Problem Set #3

Kerem Ayöz
21501569

Q1

b.)



Number of mistakes become 0 after ~2100 iterations, since perceptron finds the separating hyperplane.

c.) Yes, data is linearly seperable since perceptron gives 0 error after certain iterations.

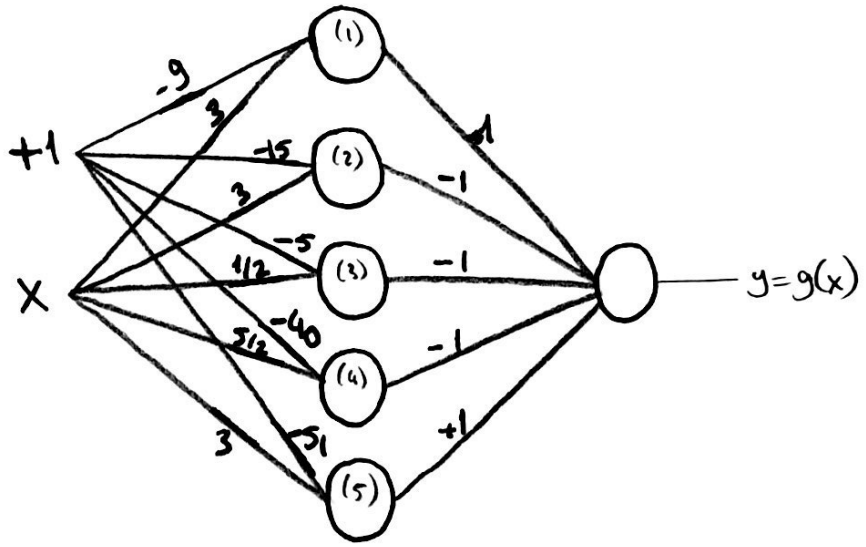
$$y = 57x_1 + 29.226x_2 + 231.6887x_3 - 201.7355x_4 - 115.8198x_5 - 176.0012x_6 - 57.7855x_7 + 231.6201x_8 + 87.1270x_9 - 115.821x_{10} - 86.59x_{11}$$

Yes, there might be different seperating hyperplanes which could be found with different learning rate and weight initializations.

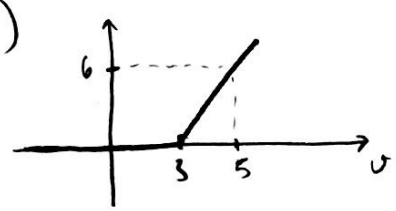
d.) We can use a method similar to SVM (Support vector machines). We can find the distances of closest points in each classes to each seperating hyperplane. We can select the hyperplane with the equal distance to closest points in each class.

QUESTION 1 Code

```
load("perceptrontest.mat");

x = D(:,1:11);
y = D(:,12);

[weights, mistakes] = train_perceptron(x,y,1,5000);
error = calculate_error(x,weights,y);
plot(mistakes);
title("Total Number of Mistakes");
ylabel("Number of Mistakes");
xlabel("Iteration Count");

function error = calculate_error(x,w,y)
    error = 0;
    for i = (1:length(x))
        est = perceptron(x(i,:),w);
        error = error + (y(i,:) - est);
    end
end

function [weig, mistakes] =
train_perceptron(x,y,learning_rate,iteration_count)
    w = zeros(1,11);
    mistakes = zeros(1,5000);
    for i = (1:iteration_count)
       [w, mistakes(i)] = single_iteration(x,y,learning_rate,w);
    end
    weig = w;
end

function [weigh, mistake] = single_iteration(x, y, learning_rate, w)
    mistake = 0;
    for i = (1:length(x))
        est = perceptron(x(i,:),w);
        w = w + learning_rate * (y(i,:) - est) * x(i,:);
        mistake = mistake + abs(y(i,:) - est);
    end
    weigh = w;
end


function estimate = perceptron(x_i,w)
    res = x_i*transpose(w);
    if res > 0
        estimate = 1;
    else
        estimate = 0;
    end
end
```

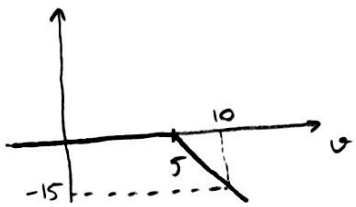


(1)



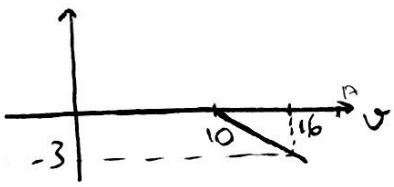
$\max(0, 3(v-3))$

(2)



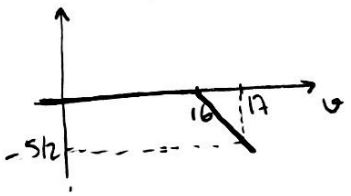
$-\max(0, 3(v-5))$

(3)



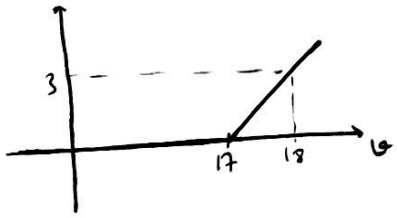
$-\max(0, \frac{1}{2}(v-10))$

(4)

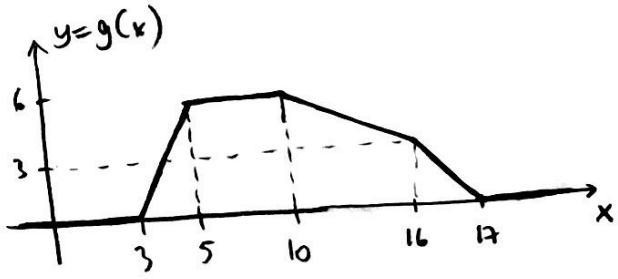


$-\max(0, \frac{5}{2}(v-16))$

(5)



$\max(0, 3(v-17))$



- Output of neural network -

Q3

a.) Because we did not standardize the samples. Third feature for each element in sample is much larger than other features. So not-standardizing the data causes one eigen value to be much bigger than others.

$$u_1 = \begin{bmatrix} 0.8278 \\ 0.5605 \\ -0.0227 \end{bmatrix} \quad u_2 = \begin{bmatrix} 0.5608 \\ -0.8279 \\ 0.015 \end{bmatrix} \quad u_3 = \begin{bmatrix} 0.0128 \\ 0.0214 \\ 0.9997 \end{bmatrix}$$

$e_1 = 0.2$ $\quad e_2 = 1.9$ $\quad e_3 = 1400.9$

b.) $$u_1' = \begin{bmatrix} 0.5245 \\ 0.5650 \\ -0.6370 \end{bmatrix} \quad u_2' = \begin{bmatrix} 0.7725 \\ -0.6303 \\ 0.0770 \end{bmatrix} \quad u_3' = \begin{bmatrix} 0.3580 \\ 0.5324 \\ 0.7670 \end{bmatrix}$$

$e_1' = 0.0742$ $\quad e_2' = 1.0276$ $\quad e_3' = 1.2981$

$$X' = \begin{matrix} z_1 & z_2 \\ \end{matrix} \begin{bmatrix} -0.6972 & -1.3093 \\ 2.1268 & -0.1135 \\ 0.1997 & 0.9804 \\ -1.0538 & 1.2970 \\ -0.5755 & -0.8646 \end{bmatrix} \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix}$$

Total Variance = 2.4

$PVM(z_1) = 1.2981 \sim \%54.08$

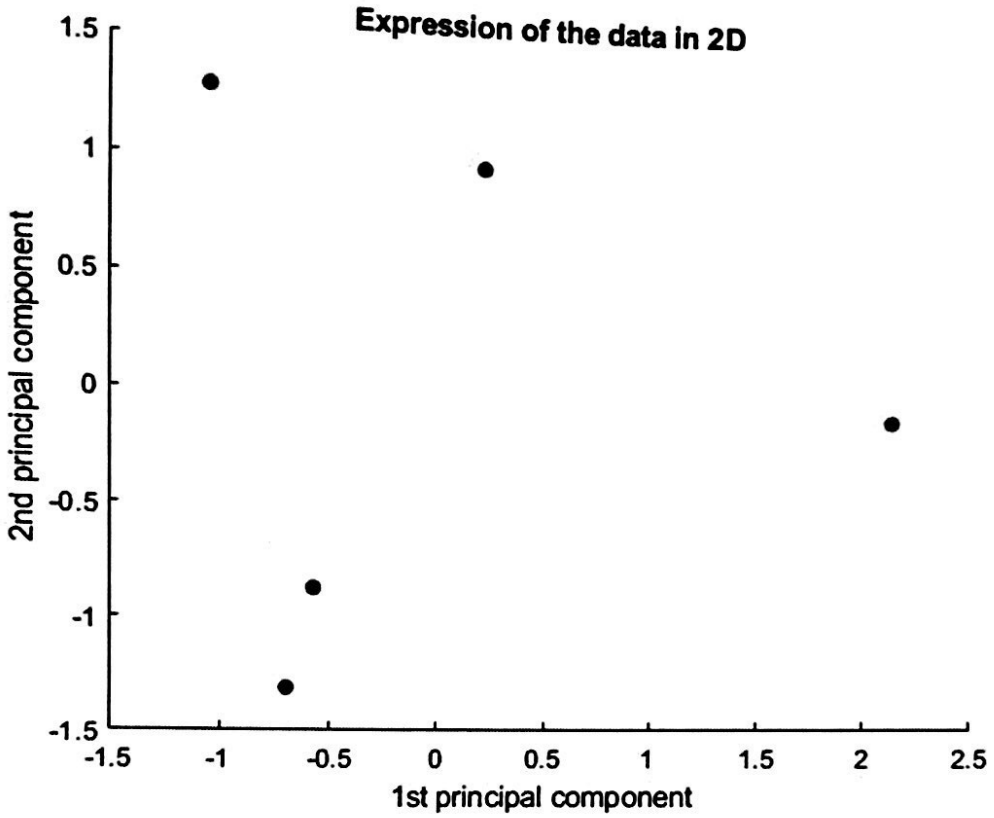
$PVM(z_2) = 1.0276 \sim \%42.82$

Variance explained by first two principal components

$\%96.9063$

Yes it is a good representation since it nearly explains all of the variance in data. ($\%96.9$)

c.) It means the non-zero dimensions are enough to represent all the variance in data. Zero eigen-value dimension does not have any variance, it is same for all the instances. We can remove that feature and represent the data with other dimensions.


Expression of the data in 2D
2nd principal component
1st principal component
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5

QUESTION 3 Code

```
x1 = [1; 4; 400];
x2 = [3; 5; 500];
x3 = [3; 2; 450];
x4 = [3; 1; 400];
x5 = [1; 3; 425];


x = [x1,x2,x3,x4,x5];
x_centered = zeros(size(x));
x_standardized = zeros(size(x));

% Centralize
for i = (1:3)
    x_centered(i,:) = x(i,:) - mean(x(i,:));
end

cov = x_centered * transpose(x_centered)./5;
[v1, e1] = eig(cov);

% Standardize
for i = (1:3)
    x_standardized(i,:) = x_centered(i,:) / std(x_centered(i,:));
end

cov = x_standardized * transpose(x_standardized)./5;
[v2, e2] = eig(cov);

x_standardized = transpose(x_standardized);
z1 = x_standardized * v2(:,3);
z2 = x_standardized * v2(:,2);

figure;
scatter(z1,z2, 'filled');
title("Expression of the data in 2D");
ylabel("2nd Principal Component");
xlabel("1st Principal Component");

totalV = sum(sum(x_standardized.*x_standardized))/5
pvm1 = sum(z1.^2) / 5
pvm2 = sum(z2.^2) / 5;
ratio = (pvm1+pvm2)/totalV*100
```

Q4

a.) Let's consider $X = A \cdot s$. If we multiply right side by $P^{-1}P = I$ where P is a permutation matrix, we get $X = A P^{-1} P s$.
If we call AP^{-1} as $\tilde{A}$ then $X = \tilde{A} P s$ which indicates that result may be a different permutation than the original s.

For sign ambiguity if we both multiply A and s with negative one we get $X = As = -A \cdot -s = \breve{A}\breve{s}$ where $\breve{s}$ has opposite sign of s, which means recovered signal might have opposite sign than the original.

b.) Let's take $s \sim \mathcal{N}(0,1)$ which is standart normal variable.
$X = As$ is the equation for recovering signal. Then x is also gaussian random variable and its variance can be calculated from covariance matrix.

$$E[xx^T] = E[Ass^TA^T] = E[AA^T] = AA^T$$
$$E[x] = E[As] = 0$$
$$\Rightarrow x \sim \mathcal{N}(0, AA^T)$$

Let's define $A' = AK$ where K is orthogonal which means $KK^T = I_p$
If we again apply recovery formula with A' we get $x' = A's$.
Calculate $E[x'x'^T]$ and $E[x']$;

$$E[x'x'^T] = E[A'ss^TA'^T] = E[A'A'^T] = E[AKK^TA^T] = E[AA^T] = AA^T$$
$$E[x'] = E[A's] = 0 \qquad \text{Then} \Rightarrow x' \sim \mathcal{N}(0, AA^T)$$

This means that with same distributed rv's x and x', we cannot decide mixing matrix since with 2 different mixing matrix we reach the same gaussian distribution for observations. Thus, sources need to be non-Gaussian.