RYERSON UNIVERSITY

Faculty of Engineering, Architecture, and Science

Department of Electrical and Computer Engineering

Course Number	EE8603
Course Title	Neural Networks and Deep Learning
Semester/Year	Summer/2018

Instructor	Dr. Kandasamy Illanko

Assignment No.	**2**

Assignment Title	Analytical and MATLAB Perceptron Training

Submission Date	29^{th} May 2018
Due Date	29^{th} May 2018

Student Name	Muhammad Obaidullah
Student ID.	500671408
Signature*	Obaidullah

*By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at: www.ryerson.ca/senate/current/pol60.pdf.

1 EXERSICE 3.5.2

1.1 MATLAB CODE

```
%%% PART A %%%
CP = [5 10; 10 5]; % Cluster center points
N = 25; % Number of points per cluster
C = size(CP,1); % Number of clusers
a = 2; % Standard deviation of the clusters
mean = 0; % Mean of the clusters
m = mean*ones(C*N, 1); % Array of means (zeros)
s = a*ones(C*N, 1); % Array of standard deviation (all a)
r = normrnd(m, s); % Random number array (Normally distributed)
theta = pi*rand(C*N, 1); % Random angles
x = zeros(C*N,2); % 2D array containing x_1 and x_2
col = ['b' 'r' 'g' 'y' 'm' 'c' 'k'];
figure
for c_i = 1:C
    %disp(['c_i = ',num2str(c_i)]);
    for i = 1+(c_i-1)*N:N+(c_i-1)*N
        %disp(['i = ',num2str(i)]);
        x(i,1) = CP(c_i, 1) + r(i)*sin(theta(i)); % x_1
        x(i,2) = CP(c_i, 2) + r(i)*cos(theta(i)); % x_2
    end
    scatter(x(1+(c_i-1)*N:N+(c_i-1)*N,1), x(1+(c_i-1)*N:N+(c_i-1)*N,2), col(c_i), 'filled'); hold on
end
hold off

%%% PART B %%%
points = 1:1:50; % Array of 50 integers
n_choose = 20; % Choose amount
sample = randsample(points,n_choose); % Choose 20 integers at random from 50 integers
chosen_points = zeros(n_choose, 2); % 2D array containing x_1 and x_2 of chosen points
for i = 1:n_choose
    % Copy over into chosen array
    chosen_points(i,1) = x(sample(i),1); % x_1
    chosen_points(i,2) = x(sample(i),2); % x_2
end

%%% PART C %%%
n_other = (C*N)-n_choose;
other_points = zeros(n_other, 2); % 2D array containing x_1 and x_2 of other points
cursor = 1;
for i = 1:C*N
    % If not chosen, copy over into other array
    if(~ismember(sample,i))
        other_points(cursor,1) = x(i,1); % x_1
        other_points(cursor,2) = x(i,2); % x_2
        cursor = cursor+1;
    end
end
figure
scatter(chosen_points(:,1), chosen_points(:,2), 'r', 'o'); hold on
scatter(other_points(:,1), other_points(:,2), 'b','s'); hold off

%%% PART D %%%
d_p = sqrt((CP(1, 1)-chosen_points(:,1)).^2 + (CP(1, 1)-chosen_points(:,1)).^2); % Distance from
d_q = sqrt((CP(2, 1)-chosen_points(:,1)).^2 + (CP(2, 1)-chosen_points(:,1)).^2); % Distance from Q
count_closer_to_p = 0;
for i = 1:n_choose`
    if(d_p(i) < d_q(i))
        count_closer_to_p = count_closer_to_p + 1;
    end
end
```

```
disp(['Out of ',num2str(n_choose),' chosen points ',num2str(count_closer_to_p),' points are closer
    to P than Q.']);
```

1.2 Figures

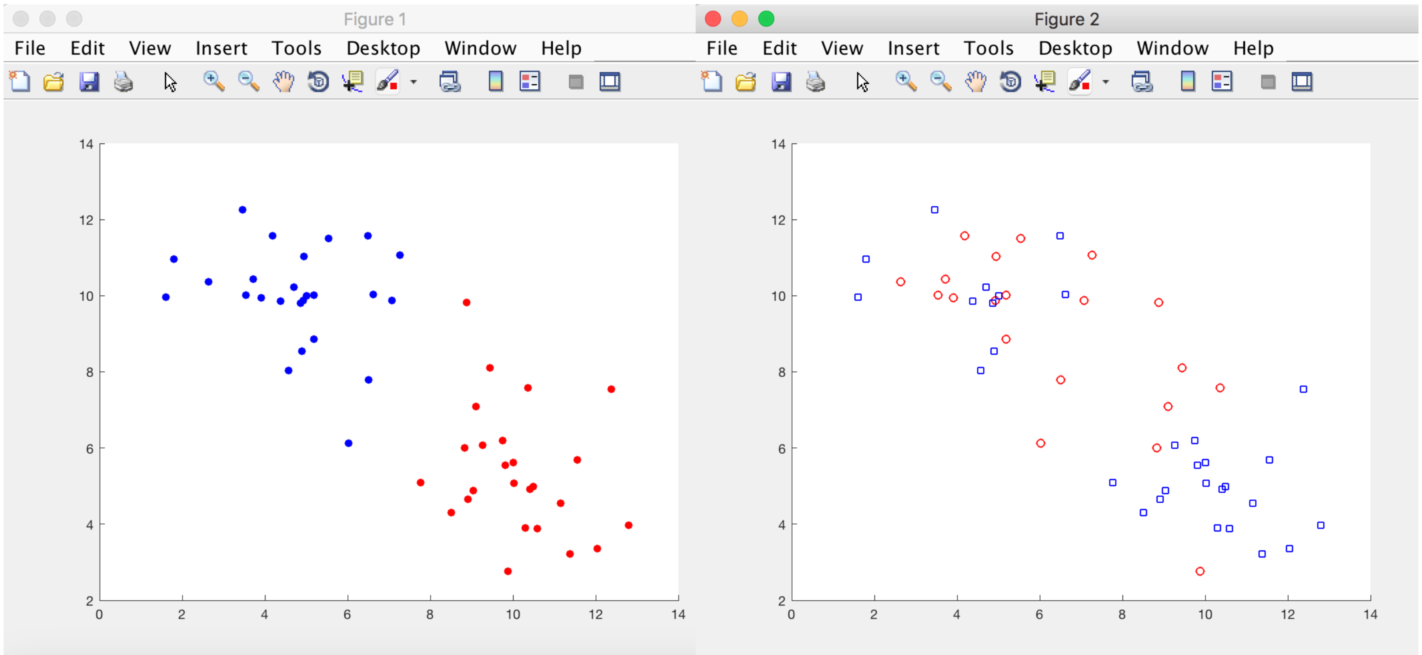


Figure 1.1: **Left:** cluster points marked with different colors.
Right: Chosen points are marked with red circles and other points are marked with blue squares.

1.3 Output

```
>> exersice_3_5_2
Out of 20 chosen points 11 points are closer to P than Q.
```

1.4 Numerical Answer

Distance between two points in a R^2 space is given by:-

$$d = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}$$

For points nearer to P than Q:

$$nearPthanQ = \begin{cases} 0 & \sqrt{(10-y)^2 + (5-x)^2} - \sqrt{(5-y)^2 + (10-x)^2} < 0 \\ 1 & \sqrt{(10-y)^2 + (5-x)^2} - \sqrt{(5-y)^2 + (10-x)^2} > 0 \end{cases}$$

Where y and x are coordinates of a point to test

2 EXERSICE 4.1.2

2.1 MATLAB CODE

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Data Creation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
CB = [14 20; 0 6; 0 6; 0 6; 14 20; 14 20]; % Cluster boundaries
D = 3; % Space dimensions
N = 2000; % Number of points per cluster
C = size(CB,1)/D; % Number of clusers
t = zeros(C*N,1); % Preallocate for the input and the target vector
x = zeros(C*N,D); % Dimension array containing x_1, x_2, ... x_D
col = ['b' 'r' 'g' 'y' 'm' 'c' 'k'];
figure
for c_i = 1:C
    %disp(['c_i = ',num2str(c_i)]);
    for p_i = 1+(c_i-1)*N:N+(c_i-1)*N
        %disp(['i = ',num2str(i)]);
        for axis = 1:D
            pos = ((c_i-1)*D)+axis;
            range = CB(pos,2) - CB(pos,1);
            x(p_i,axis) = range*rand + CB(pos, 1);
        end
        t(p_i) = c_i - 1;
    end
    start = 1+(c_i-1)*N;
    finish = N+(c_i-1)*N;
    scatter3(x(start:finish,1), x(start:finish,2), x(start:finish,3), 3, col(c_i), 'filled');
    hold on
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Training and Testing
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Parameter intialization
w1 = 0.5;
w2 = 0.5;
w3 = 0.5;
b = 0.5;
error_rate = 1;
M = 0;
while error_rate > 0.1
    M = M + 1;  % Gradually increasing number of points used for training
    % Indices for random selection of data for training and testing.
    sp = randsample(C*N,M+1000);
    for i = 1:M
        y = b + w1*x(sp(i),1) + w2*x(sp(i),2) + w3*x(sp(i),3);
        if(y < 0)
            y = 0;
        else
            y = 1;
        end
        e = t(sp(i)) - y;
        w1 = w1+e*x(sp(i),1);
        w2 = w2+e*x(sp(i),2);
        w3 = w3+e*x(sp(i),3);
        b = b + e;
    end
    er = 0;
    for i=M+1:M+1000
        y = b + w1*x(sp(i),1) + w2*x(sp(i),2) + w3*x(sp(i),3);
        if (y < 0)
            y = 0;
        else
            y = 1;
```

```
        end
        e = abs(t(sp(i))-y);
        er = er + e;
    end
    error_rate = er/1000
end
M
% Drawing a seperating plane
%Decide on a suitable showing range
x1Lim = [0 20];
x2Lim = [0 20];
[X1,X2] = meshgrid(x1Lim,x2Lim);
X3 = (w1*X1 + w2*X2 + b)/ (-w3);
reOrder = [1 2 4 3];
patch(X1(reOrder),X2(reOrder),X3(reOrder),'g');
grid on;
alpha(0.3);
```

2.2 FIGURES

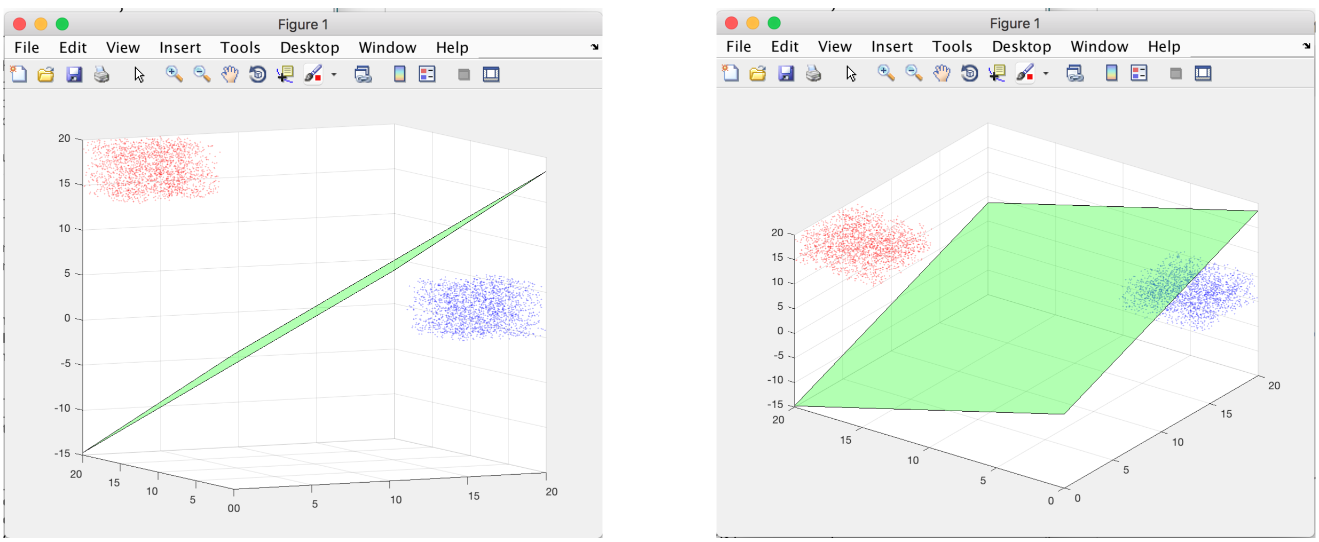
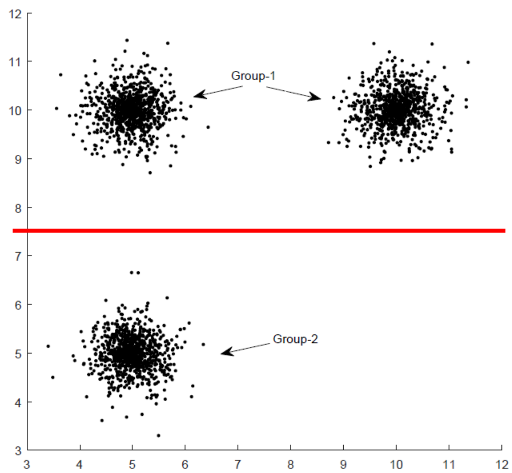


Figure 2.1: Separating plane obtained by training the basic perceptron. Seperating plane is shown in green while two clusters are shown in red and blue.

```
>> exersice_4_1_2

error_rate =              error_rate =

    0.5030                        0


error_rate =              M =

    0.5070                        3
```

Figure 2.2: The ouput of the program is shown. The error rate drops to 0 within 3 iterations.

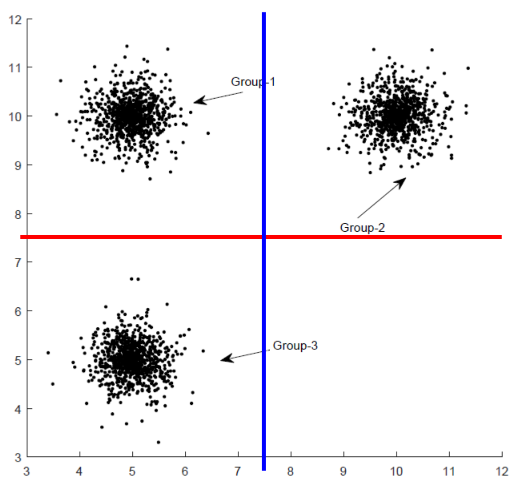
3 EXERSICE 4.1.3



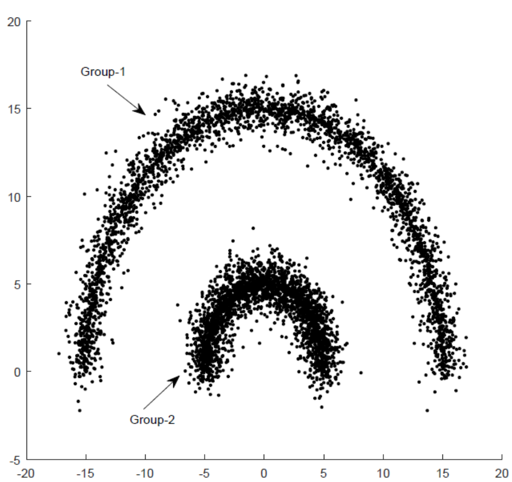
FOR PLOT 1: Since there are just 2 groups and both are linearly seperable, a basic perceptron can seperate the two groups. Line $y = 7.5$ **CAN** perfectly seperate the two cluster groups.

$$y = \begin{cases} 1 & x_2 - 7.5 < 0 \\ 0 & x_2 - 7.5 > 0 \end{cases}$$

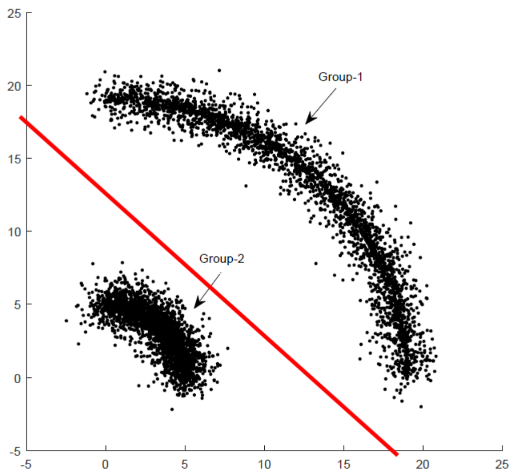
Where $b = -7.5$, $w_1 = 0$, and $w_2 = 1$.



FOR PLOT 2: Since there are 3 groups, a basic perceptron (a straight line seperator) **CANNOT** seperate these groups. They are however, seperable using more than 1 perceptron.



FOR PLOT 3: There cannot be a straight line which can divide or seperate the two groups. The groups are not linearly seperable. Therefore, a basic perceptron **CANNOT** be used to seperate the groups.



FOR PLOT 4: It can be seen that a straight line passing through points $(17.5, 0)$ and $(0, 17.5)$ **CAN** seperate the two groups. Find the gradient (m) of seperating line:

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{17.5 - 0}{0 - 17.5} = -1$$

Find the equation of seperating line:

$$\begin{aligned} &y - y_1 = m(x - x_1) \\ &y - 0 = -1(x + 17.5) \\ &y = -x - 17.5 \\ &y + x + 17.5 = 0 \end{aligned}$$

Replacing the variables:

$$x_1 + x_2 + 17.5 = 0$$

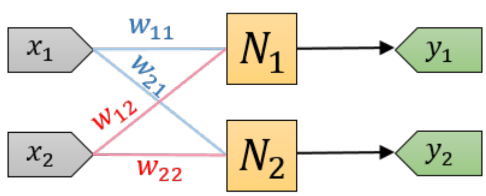
$$y = \begin{cases} 1 & x_1 + x_2 + 17.5 > 0 \\ 0 & x_1 + x_2 + 17.5 < 0 \end{cases}$$

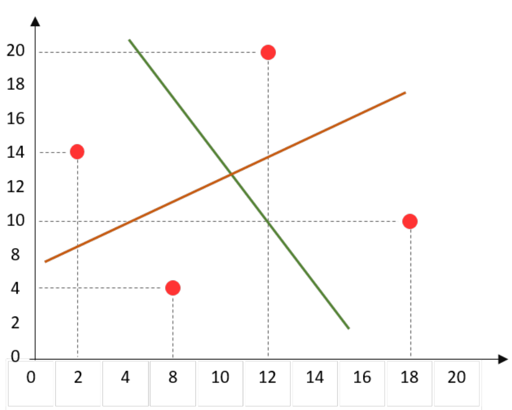
Where $w_1 = 1$, $w_2 = 1$, and $b = 17.5$.

4 EXERSICE 4.3.1

4.1 SOLUTION PART A

Since the search space is R^2, the number of inputs to the neural network will be 2. Two lines can seperate 4 linearly seperable clusters. Therefore 2 perceptrons are required. Midpoints for x_1 seperator:





$$MP_1 = \left(\frac{8 + 18}{2}, \frac{4 + 10}{2}\right) = (13, 7)$$

$$MP_2 = \left(\frac{18 + 12}{2}, \frac{10 + 20}{2}\right) = (15, 15)$$

$$MP_3 = \left(\frac{12 + 2}{2}, \frac{20 + 14}{2}\right) = (7, 17)$$

$$MP_4 = \left(\frac{2 + 8}{2}, \frac{14 + 4}{2}\right) = (5, 9)$$

Find the gradients (m_1 and m_2) of seperating lines:

$$m_1 = \frac{y_2 - y_1}{x_2 - x_1} = \frac{15 - 9}{15 - 5} = \frac{3}{5} = 0.6$$

$$m_2 = \frac{y_2 - y_1}{x_2 - x_1} = \frac{7 - 17}{13 - 7} = -\frac{5}{3}$$

Equation of line 1 (Perceptron N_1):

$$y - y_1 = m(x - x_1)$$

$$y - 15 = \frac{3}{5}(x - 15)$$

$$y - 15 = \frac{3}{5}x - 9$$

$$y - 6 - \frac{3}{5}x = 0$$

Renaming variables:

$$-\frac{3}{5}x_1 + x_2 - 6 = 0$$

Therefore:

$$w_{11} = -\tfrac{3}{5},\ w_{12} = 1, \text{ and } b_1 = -6$$

Equation of line 2 (Perceptron N_2):

$$y - y_1 = m(x - x_1)$$

$$y - 17 = -\frac{5}{3}(x - 7)$$

$$y - 17 = -\frac{5}{3}x + \frac{35}{3}$$

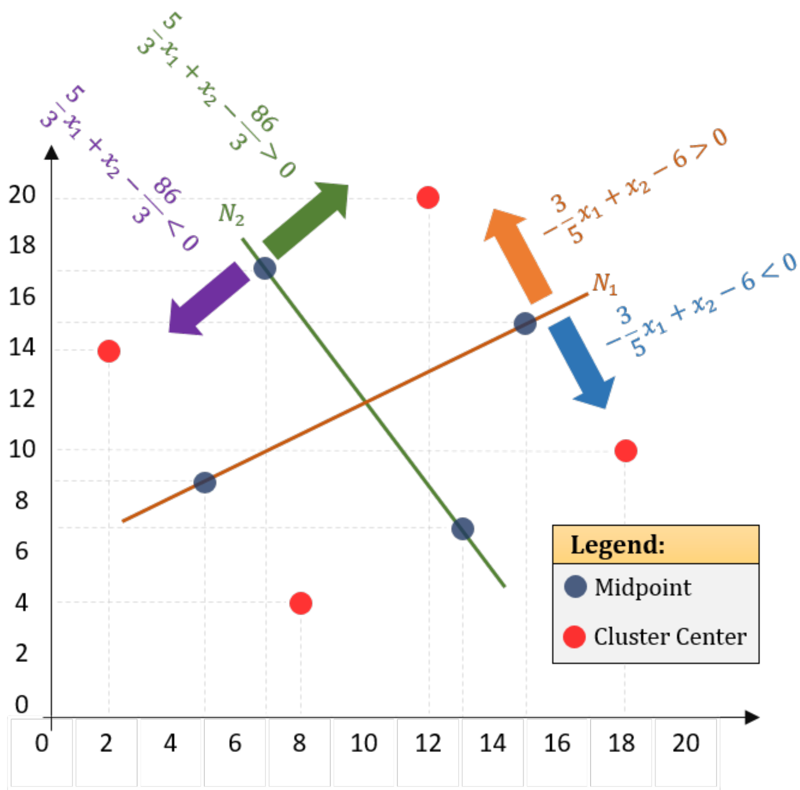
$$y - \frac{86}{3} + \frac{5}{3}x = 0$$

Renaming variables:

$$\frac{5}{3}x_1 + x_2 - \frac{86}{3} = 0$$

Therefore:

$$w_{21} = \tfrac{5}{3},\ w_{22} = 1, \text{ and } b_1 = -\tfrac{86}{3}$$



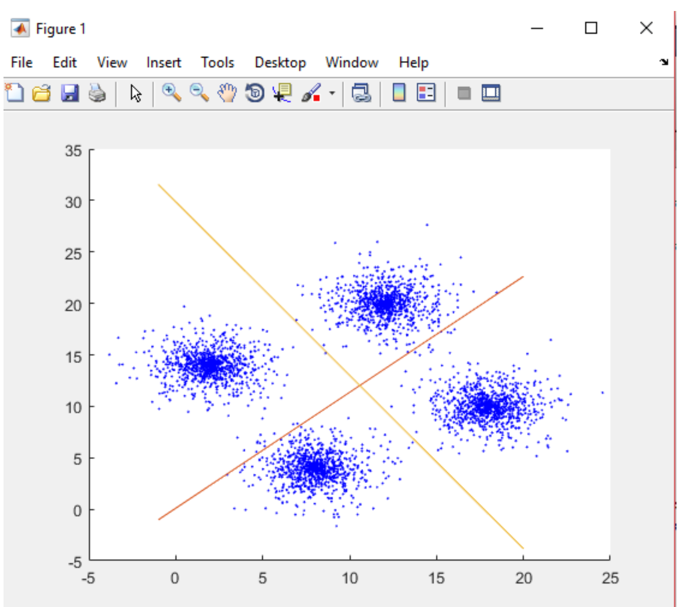
4.2 Solution Part B

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%                          DATA  CREATION                        %%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
p = [2 14 8 4 12 20 18 10];
a = 2;
N = 1000;
M = 1000;
m = zeros(4*N,1);
s = a*ones(4*N,1);
r = normrnd(m,s);
```

```
the = pi*rand(4*N,1);
x = zeros(4*N,2);
t = zeros(4*N,2);
for i = 1:N
    x(i,1) = p(1) + r(i)*cos(the(i));
    x(i,2) = p(2) + r(i)*sin(the(i));
    t(i,1) = 0;
    t(i,2) = 1;
end
for i=N+1:2*N
    x(i,1) = p(3) + r(i)*cos(the(i));
    x(i,2) = p(4) + r(i)*sin(the(i));
    t(i,1) = 1;
    t(i,2) = 1;
end
for i=2*N+1:3*N
    x(i,1) = p(5) + r(i)*cos(the(i));
    x(i,2) = p(6) + r(i)*sin(the(i));
    t(i,1) = 0;
    t(i,2) = 0;
end
for i=3*N+1:4*N
    x(i,1) = p(7) + r(i)*cos(the(i));
    x(i,2) = p(8) + r(i)*sin(the(i));
    t(i,1) = 1;
    t(i,2) = 0;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%                              TRAINING                              %%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
w11 = 0.5 - rand;
w12 = 0.5 - rand;
b1 = 0.5 - rand;
sp = randsample(4*N,M+1000);
for i = 1:M
    y1 = b1 + w11*x(sp(i),1) + w12*x(sp(i),2);
    if(y1 < 0)
        y1 = 0;
    else
        y1 = 1;
    end
    e1(i) = t(sp(i),1) - y1;
    w11 = w11 + e1(i)*x(sp(i),1);
    w12 = w12 + e1(i)*x(sp(i),2);
    b1 = b1 + e1(i);
end
w21 = 0.5 - rand;
w22 = 0.5 - rand;
b2 = 0.5 - rand;
for i=1:M
    y2 = b2 + w21*x(sp(i),1) + w22*x(sp(i),2);
    if(y2 < 0)
        y2 = 0;
    else
        y2 = 1;
    end
    e2(i) = t(sp(i),2) - y2;
    w21 = w21 + e2(i)*x(sp(i),1);
    w22 = w22 + e2(i)*x(sp(i),2);
    b2 = b2 + e2(i);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%                              TESTING                               %%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
er=0;
for i = M+1:M+1000
    y1 = b1 + w11*x(sp(i),1) + w12*x(sp(i),2);
    if(y1<0)
        y1=0;
    else
        y1=1;
    end
    e1 = abs(t(sp(i),1) - y1);
    y2 = b2 + w21*x(sp(i),1) + w22*x(sp(i),2);
    if(y2<0)
        y2 = 0;
    else
        y2 = 1;
    end
    e2 = abs(t(sp(i),2) - y2);
    if(e1 == 1||e2 == 1)
        er = er + 1;
    end
end
er = er/1000;
fprintf('Iterations: %i, Error Rate: %.2f\n', M, er);
fprintf('Perceptron 1: %.2f x_1 + x_2 + %.2f = 0\n', w11/w12, b1);
fprintf('Perceptron 2: %.2f x_1 + x_2 + %.2f = 0\n', w21/w22, b2);
if(er < 0.1)
    figure
    scatter(x(:,1),x(:,2),3,'b','filled');
    hold on
    i = -1:20;
    plot(i,(-b1-w11*i)/w12);
    hold on
    plot(i,(-b2-w21*i)/w22);
    hold off
end
```

4.3 Figures




```
>> exersice_4_3_1_another_sol
Iterations: 1000, Error Rate: 0.01
Perceptron 1: -1.13 x_1 + x_2 + 3.84 = 0
Perceptron 2: 1.69 x_1 + x_2 + 103.53 = 0
```

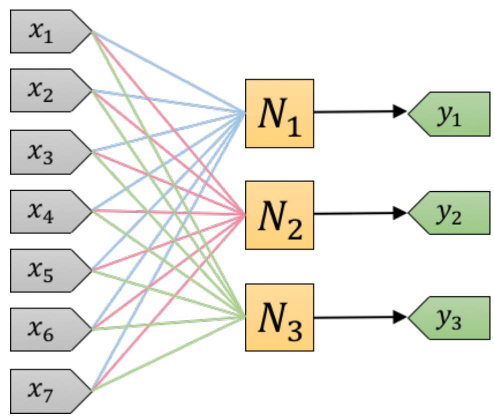
Figure 4.1: **Left:** Separating lines obtained by training.
Right: Output of program showing weights and biases.

5 EXERSICE 4.3.3

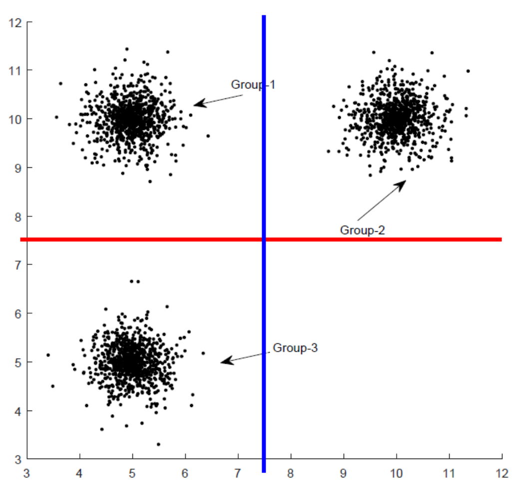
Since the search space is R^7, the number of inputs to the neural network will be 7. The number of outputs and number of neurons depend on the number of clusters to classify. In classification neuron the output is usually kept binary. Therefore, number of minimal neurons is given by:

$$N_{neurons} = \lceil log_2(N_{clusters}) \rceil = \lceil log_2(5) \rceil = \lceil 2.32 \rceil = 3$$

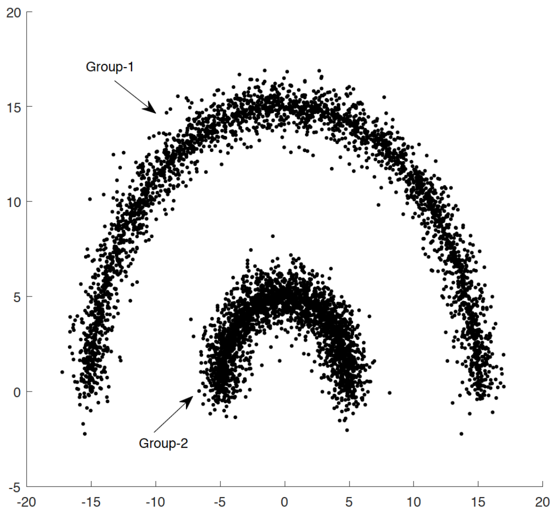
Another analogy to think of while determining number of minimal neurons required is stack and cut cake. If a person takes a whole cake and cuts it once, the cake is divided into two slices. Then if the cake's two peices are stacked on top of each other then cut again, we end up with four slices. If this is repeated, we end up with eight slices. So on an so forth. In this analogy the knife is a separating hyper-plane and the slices are clusters.



6 EXERSICE 4.3.4



FOR PLOT 1: Since the 3 groups are linearly sperable from each other, a single layer multi-perceptron **CAN** seperate these groups. A single layer with 2 neurons (each neuron having two inputs) can seperate these groups.



FOR PLOT 2: Since the groups are not linearly seperable, a single layer multi-perceptron **CANNOT** seperate them.

7 EXERSICE 5.1.1

$$z = f(x, y) = \frac{1}{2}x^2 + y^2$$

7.1 SOLUTION PART A

$$\frac{\partial z}{\partial x} = x$$

$$\frac{\partial z}{\partial y} = 2y$$

$$\therefore \nabla f(x, y) = (x, 2y)$$

$$\nabla f(1, 1) = (1, 2)$$

Therefore, required direction is $(1, 2)$ and maximum rate of change in this direction or greatest rate of change is $||\nabla f|| = \sqrt{1^2 + 2^2} = \sqrt{5}$

7.2 SOLUTION PART B

Given vector:

$$u = (3, 4)$$

$$||u|| = \sqrt{3^2 + 4^2} = \sqrt{9 + 16} = 5$$

Unit vector of u:

$$\hat{u} = \frac{u}{||u||} = \frac{(3, 4)}{5} = \Big(\frac{3}{5}, \frac{4}{5}\Big)$$

The directional derivative of f in the direction given by the vector u is given by:

$$D_u f = \nabla f . \frac{u}{||u||}$$

$$= (x, 2y).\Big(\frac{3}{5}, \frac{4}{5}\Big)$$

$$= \frac{3}{5}x + \frac{8}{5}y$$

$$D_u f(1, 1) = \frac{3}{5}x + \frac{8}{5}y = \frac{3}{5}(1) + \frac{8}{5}(1) = \frac{11}{5}$$

7.3 Solution Part c

Steepest Decent at point (1,1):

$$\nabla f(x,y) = (x, 2y) = (1, 2)$$

Direction unit vector of most rapid increase/decrease:

$$\hat{d} = \frac{\nabla f(x,y)}{||\nabla f(x,y)||} = \frac{\nabla f(1,1)}{||\nabla f(1,1)||} = \frac{(1,2)}{\sqrt{1^2+2^2}} = \frac{(1,2)}{\sqrt{5}}$$

$$\therefore \hat{d}_{increase} = \Big(\frac{1}{\sqrt{5}}, \frac{2}{\sqrt{5}}\Big)$$

To get direction of most rapid decrease at point (1,1), we multiply by negative:

$$\therefore \hat{d}_{decrease} = -\Big(\frac{1}{\sqrt{5}}, \frac{2}{\sqrt{5}}\Big) = \Big(-\frac{1}{\sqrt{5}}, -\frac{2}{\sqrt{5}}\Big)$$

8 Exersice 5.1.2

$$z = f(x,y) = xy^2$$

$$x = r\ cos(\theta)$$

$$y = r\ sin(\theta)$$

To find $\frac{\partial z}{\partial r}$, we have to find dependencies. One dependency of r comes through x and another through y:

$$\frac{\partial z}{\partial r} = \frac{\partial z}{\partial x}\frac{\partial x}{\partial r} + \frac{\partial z}{\partial y}\frac{\partial y}{\partial r}$$

$$= y^2\ cos(\theta) + 2xy\ sin(\theta)$$

Expanding x and y:-

$$= (r\ sin(\theta))^2\ cos(\theta) + 2(r\ cos(\theta))(r\ sin(\theta))sin(\theta)$$

$$= r^2\ sin^2(\theta)cos(\theta) + 2r^2\ sin^2(\theta)cos(\theta) = 3r^2\ sin^2(\theta)cos(\theta)$$

$$\therefore \frac{\partial z}{\partial r} = 3r^2\ sin^2(\theta)cos(\theta)$$

To find $\frac{\partial z}{\partial \theta}$, we have to find dependencies. One dependency of r comes through x and another through y:

$$\frac{\partial z}{\partial \theta} = \frac{\partial z}{\partial x}\frac{\partial x}{\partial \theta} + \frac{\partial z}{\partial y}\frac{\partial y}{\partial \theta}$$

$$= 2rxy\ cos(\theta) - ry^2\ sin(\theta)$$

Expanding x and y:-

$$= 2r(r\ cos(\theta))(r\ sin(\theta))\ cos(\theta) - r(r\ sin(\theta))^2\ sin(\theta)$$

$$= 2r^3\ cos^2(\theta)sin(\theta) - r^3\ sin^3(\theta)$$

$$\therefore \frac{\partial z}{\partial \theta} = 2r^3\ cos^2(\theta)sin(\theta) - r^3\ sin^3(\theta)$$

9 EXERSICE 5.1.3

$$y_1 = x_1 - 2x_2 + 3x_3$$
$$y_2 = 3x_1 + 4x_2 - 2x_3$$
$$yy = 7y_1 + 11y_2$$

To find $\frac{\partial yy}{\partial x_1}$, we have to find dependencies. One dependency of x_1 comes through y_1 and another through y_2:

$$\frac{\partial yy}{\partial x_1} = \frac{\partial yy}{\partial y_1}\frac{\partial y_1}{\partial x_1} + \frac{\partial yy}{\partial y_2}\frac{\partial y_2}{\partial x_1}$$
$$= 7(1) + 11(3) = 40$$
$$\therefore \frac{\partial yy}{\partial x_1} = 40$$

To find $\frac{\partial yy}{\partial x_3}$, we have to find dependencies. One dependency of x_3 comes through y_1 and another through y_2:

$$\frac{\partial yy}{\partial x_3} = \frac{\partial yy}{\partial y_1}\frac{\partial y_1}{\partial x_3} + \frac{\partial yy}{\partial y_2}\frac{\partial y_2}{\partial x_3}$$
$$= 7(3) + 11(-2) = -1$$
$$\therefore \frac{\partial yy}{\partial x_3} = -1$$

10 EXERSICE 5.1.4

$$y_1 = ax_1 + bx_2 + cx_3$$
$$y_2 = dx_1 + ex_2 + fx_3$$
$$yy_1 = py_1 + qy_2$$
$$yy_2 = ry_1 + sy_2$$
$$yyy = kyy_1 + lyy_2$$

To find $\frac{\partial yyy}{\partial y_1}$, we have to find dependencies. One dependency of y_1 comes through yy_1 and another through yy_2:

$$\frac{\partial yyy}{\partial y_1} = \frac{\partial yyy}{\partial yy_1}\frac{\partial yy_1}{\partial y_1} + \frac{\partial yyy}{\partial yy_2}\frac{\partial yy_2}{\partial y_1}$$
$$\therefore \frac{\partial yyy}{\partial y_1} = kp + lr$$

To find $\frac{\partial yyy}{\partial x_3}$, we have to find dependencies. One dependency of x_1 comes through yy_1, yy_2 where both have a path to x_1 through y_1 and y_2:

$$\frac{\partial yyy}{\partial x_3} = \frac{\partial yyy}{\partial yy_1}\frac{\partial yy_1}{\partial x_3} + \frac{\partial yyy}{\partial yy_2}\frac{\partial yy_2}{\partial x_3}$$
$$\frac{\partial yy_1}{\partial x_3} = \frac{\partial yy_1}{\partial y_1}\frac{\partial y_1}{\partial x_3} + \frac{\partial yy_1}{\partial y_2}\frac{\partial y_2}{\partial x_3}$$
$$\frac{\partial yy_2}{\partial x_3} = \frac{\partial yy_2}{\partial y_1}\frac{\partial y_1}{\partial x_3} + \frac{\partial yy_2}{\partial y_2}\frac{\partial y_2}{\partial x_3}$$
$$\frac{\partial yyy}{\partial x_3} = \frac{\partial yyy}{\partial yy_1}\left(\frac{\partial yy_1}{\partial y_1}\frac{\partial y_1}{\partial x_3} + \frac{\partial yy_1}{\partial y_2}\frac{\partial y_2}{\partial x_3}\right) + \frac{\partial yyy}{\partial yy_2}\left(\frac{\partial yy_2}{\partial y_1}\frac{\partial y_1}{\partial x_3} + \frac{\partial yy_2}{\partial y_2}\frac{\partial y_2}{\partial x_3}\right)$$
$$= k(pc + qf) + l(rc + sf)$$
$$\therefore \frac{\partial yyy}{\partial x_3} = kpc + kqf + lrc + lsf$$

11 EXERSICE 5.1.5

$$z = T(y)$$
$$y = S(x)$$
$$T = \begin{bmatrix} ww_{11} & ww_{12} & ww_{13} \\ ww_{21} & ww_{22} & ww_{23} \\ ww_{31} & ww_{32} & ww_{33} \end{bmatrix}, \quad S = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix}$$
$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} ww_{11} & ww_{12} & ww_{13} \\ ww_{21} & ww_{22} & ww_{23} \\ ww_{31} & ww_{32} & ww_{33} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$
$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Find $\frac{\partial z_2}{\partial x_3}$:

$$\frac{\partial z_2}{\partial x_3} = \frac{\partial z_2}{\partial y_1}\frac{\partial y_1}{\partial x_3} + \frac{\partial z_2}{\partial y_2}\frac{\partial y_2}{\partial x_3} + \frac{\partial z_2}{\partial y_3}\frac{\partial y_3}{\partial x_3} = ww_{21}.w_{13} + ww_{22}.w_{23} + ww_{23}.w_{33}$$

Find $\frac{\partial z_3}{\partial x_1}$:

$$\frac{\partial z_3}{\partial x_1} = \frac{\partial z_3}{\partial y_1}\frac{\partial y_1}{\partial x_1} + \frac{\partial z_3}{\partial y_2}\frac{\partial y_2}{\partial x_1} + \frac{\partial z_3}{\partial y_3}\frac{\partial y_3}{\partial x_1} = ww_{31}.w_{w11} + ww_{32}.w_{21} + ww_{33}.w_{31}$$

Find $\frac{\partial z_1}{\partial ww_{12}}$:

$$\frac{\partial z_1}{\partial ww_{12}} = y_2$$

Find $\frac{\partial z_1}{\partial w_{22}}$:

$$\frac{\partial z_1}{\partial w_{22}} = \frac{\partial z_1}{\partial y_1}\frac{\partial y_1}{\partial w_{22}} + \frac{\partial z_1}{\partial y_2}\frac{\partial y_2}{\partial w_{22}} + \frac{\partial z_1}{\partial y_3}\frac{\partial y_3}{\partial w_{22}}$$
$$= (ww_{11})(0) + (ww_{12})(x_2) + (ww_{13})(0)$$
$$\therefore \frac{\partial z_1}{\partial w_{22}} = ww_{12}x_2$$

Find $\frac{\partial z_2}{\partial w_{13}}$:

$$\frac{\partial z_2}{\partial w_{13}} = \frac{\partial z_2}{\partial y_1}\frac{\partial y_1}{\partial w_{13}} + \frac{\partial z_2}{\partial y_2}\frac{\partial y_2}{\partial w_{13}} + \frac{\partial z_2}{\partial y_3}\frac{\partial y_3}{\partial w_{13}}$$
$$= (ww_{21})(x_3) + (ww_{22})(0) + (ww_{23})(0)$$
$$\therefore \frac{\partial z_2}{\partial w_{13}} = ww_{21}x_3$$

12 EXERSICE 5.2.1

Determining if the following matrix is positive definitive:

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 2 & 3 & 3 \\ 1 & -1 & 1 \end{bmatrix}$$

1 > 0, det $\begin{bmatrix} 1 & 1 \\ 2 & 3 \end{bmatrix} = 3 - 2 > 0$, and $det(A) = 1(3+3) - 1(2-3) + 0(-2-3) = 6 + 1 = 7 > 0$,
Hence A is positive definite.
Determining if the following matrix is positive definitive:

$$B = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & 3 \\ 1 & -1 & 1 \end{bmatrix}$$

$1 > 0$, det $\begin{bmatrix} 1 & 1 \\ 2 & 3 \end{bmatrix} = 3 - 2 > 0$, and $det(A) = 1(3+3) - 1(2-3) + 1(-2-3) = 6 + 1 - 5 = 2 > 0$,
Hence B is positive definite.

13 EXERSICE 5.2.2

13.1 SOLUTION FOR PART A

Find Hessian matrix for following function:

$$f = x^2y + y^3x$$

$$H = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{bmatrix}$$

$$f_x = \frac{\partial f}{\partial x} = 2xy + y^3$$

$$f_{xx} = \frac{\partial f_x}{\partial x} = 2y$$

$$f_{xy} = \frac{\partial f_x}{\partial y} = 2x + 3y^2$$

$$f_y = \frac{\partial f}{\partial y} = x^2 + 3xy^2$$

$$f_{yy} = \frac{\partial f_y}{\partial y} = 6xy$$

$$f_{yx} = \frac{\partial f_y}{\partial x} = 2x + 3y^2$$

$$\therefore H = \begin{bmatrix} 2y & 2x + 3y^2 \\ 2x + 3y^2 & 6xy \end{bmatrix}$$

13.2 SOLUTION FOR PART B

Find Hessian matrix for following function:

$$g = x^2 + 2y^2 + 3z^2 + xy + yz + zx + xyz$$

$$H = \begin{bmatrix} g_{xx} & g_{xy} & g_{xz} \\ g_{yx} & g_{yy} & g_{yz} \\ g_{zx} & g_{zy} & g_{zz} \end{bmatrix}$$

$$g_x = \frac{\partial g}{\partial x} = 2x + y + z + yz$$

$$g_y = \frac{\partial g}{\partial y} = 4y + x + z + xz$$

$$g_z = \frac{\partial g}{\partial z} = 6z + y + x + xy$$

$$\therefore H = \begin{bmatrix} 2 & 1+z & 1+y \\ 1+z & 4 & 1+x \\ 1+y & 1+x & 6 \end{bmatrix}$$

14 EXERSICE 5.2.3

Prove that the following function is convex in the first quadrant of its domain:

$$f = x^3 + 3y^2 + 6xy \tag{14.1}$$

$$H = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{bmatrix}$$

$$f_x = \frac{\partial f}{\partial x} = 3x^2 + 6y$$

$$f_y = \frac{\partial f}{\partial y} = 6y + 6x$$

$$H = \begin{bmatrix} 6x & 6 \\ 6 & 6 \end{bmatrix}$$

In first quadrant, $x > 0$ and $y > 0$ which makes the first value of matrix grater than 0 (ie. $6x > 0$) and $det(A) = 36x - 6 > 0$ which makes the hessian matrix positive definitive. When hessian matrix of a function is positive definitive in domain D, the function is convex in domain D.