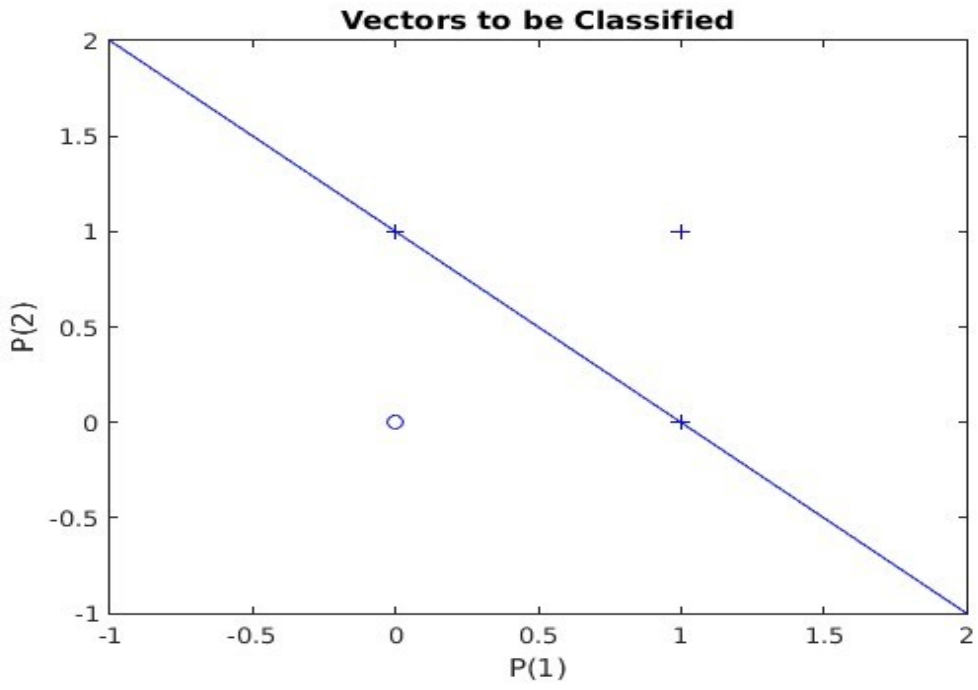
SAGNIK BASU 113EC0199

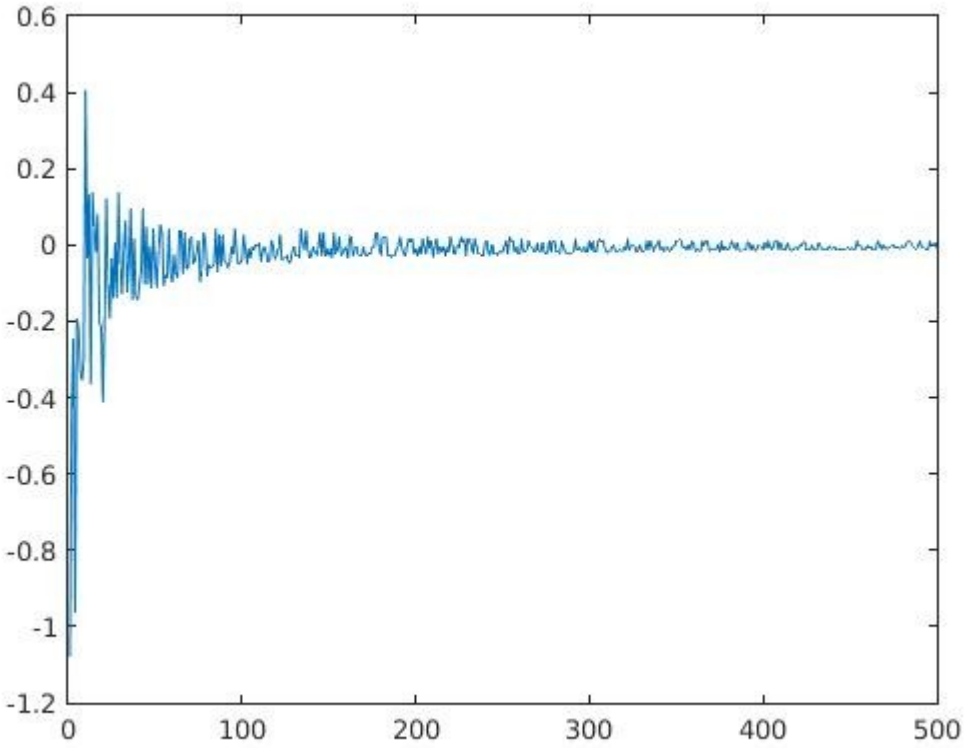
LAB 1

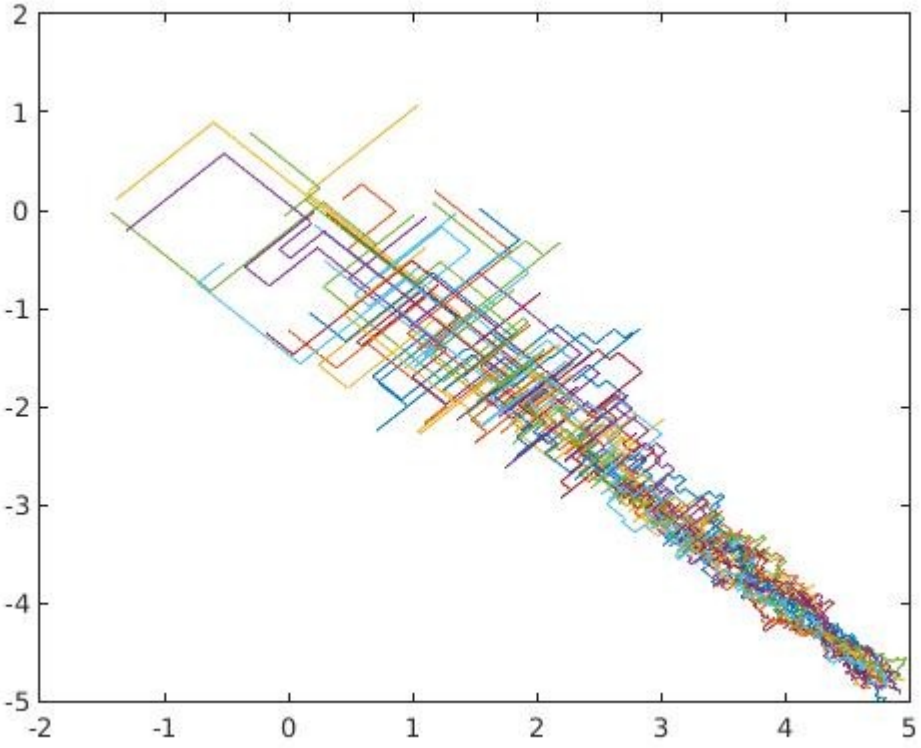
Logic gates using perceptron

1) Two input or gates

```
        % Run_or_Perceptron.m
P = [0 0 1 1; 0 1 0 1]; % or Function
T = [0 1 1 1];
plotpv(P,T,[-1, 2, -1, 2]); % plot data
% initial weight vector and bias
W = [1 1]; b = -1;
plotpc(W,b); % plot line
epoc = 1; % number of epoc
for j=1:epoc
for i=1:size(P,2)
p = P(:,i);
t = T(i);
net=perceptron;
[W b] = train(p',t,W,b);
plotpc(W,b);
end
end
% test data test = [0 0]'
test = [0 0];
output = hardlim(W*test'+b)
```

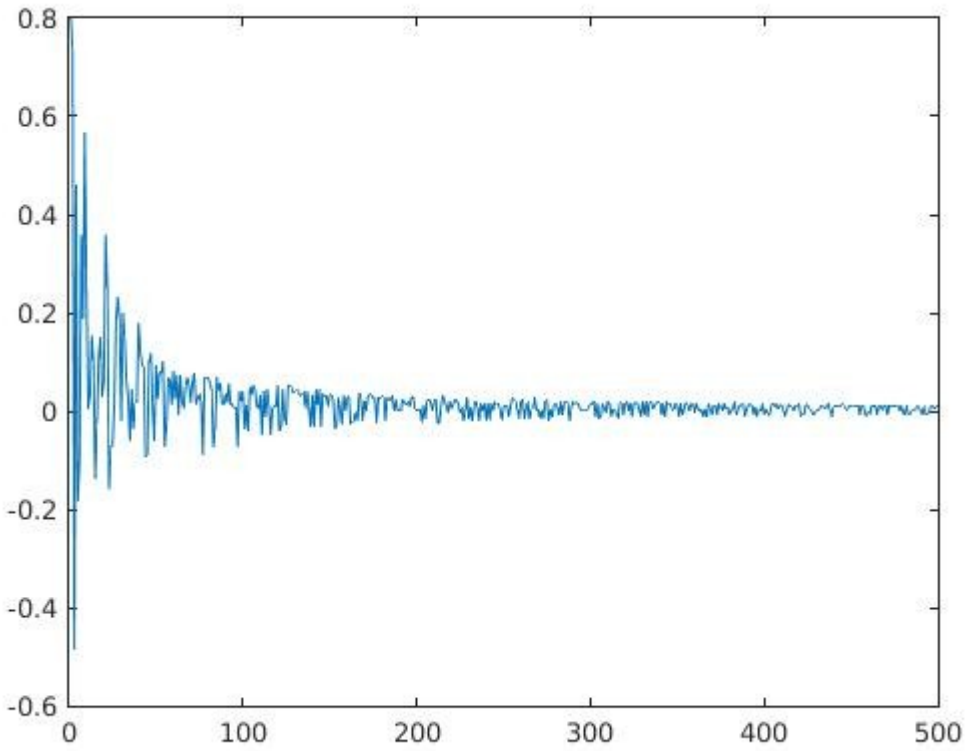



0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
0
100
200
300
400
500

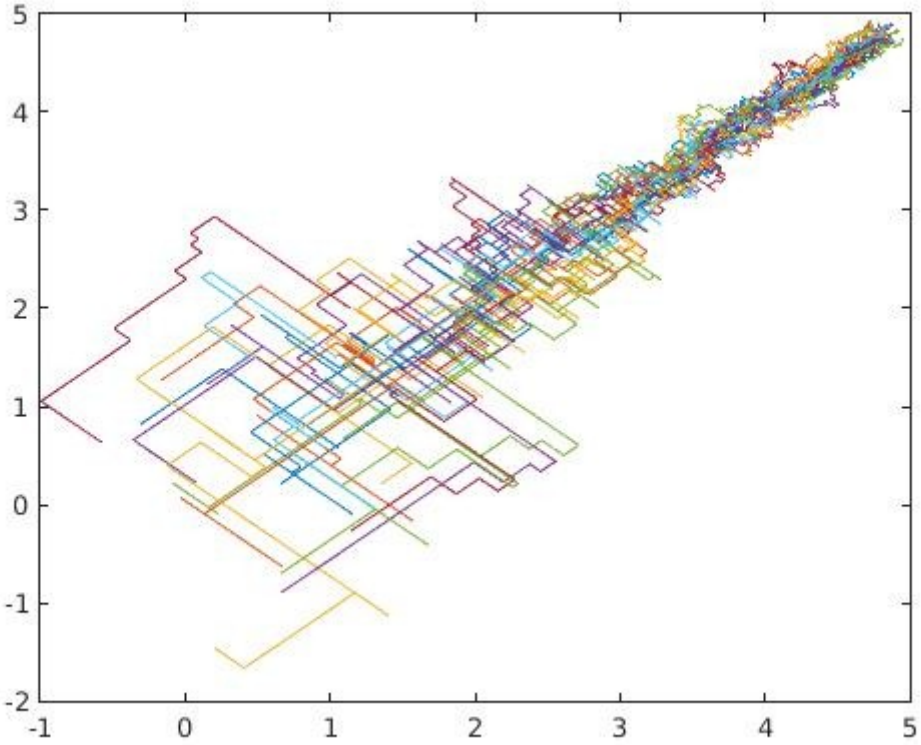

2
1
0
-1
-2
-3
-4
-5
-2
-1
0
1
2
3
4
5

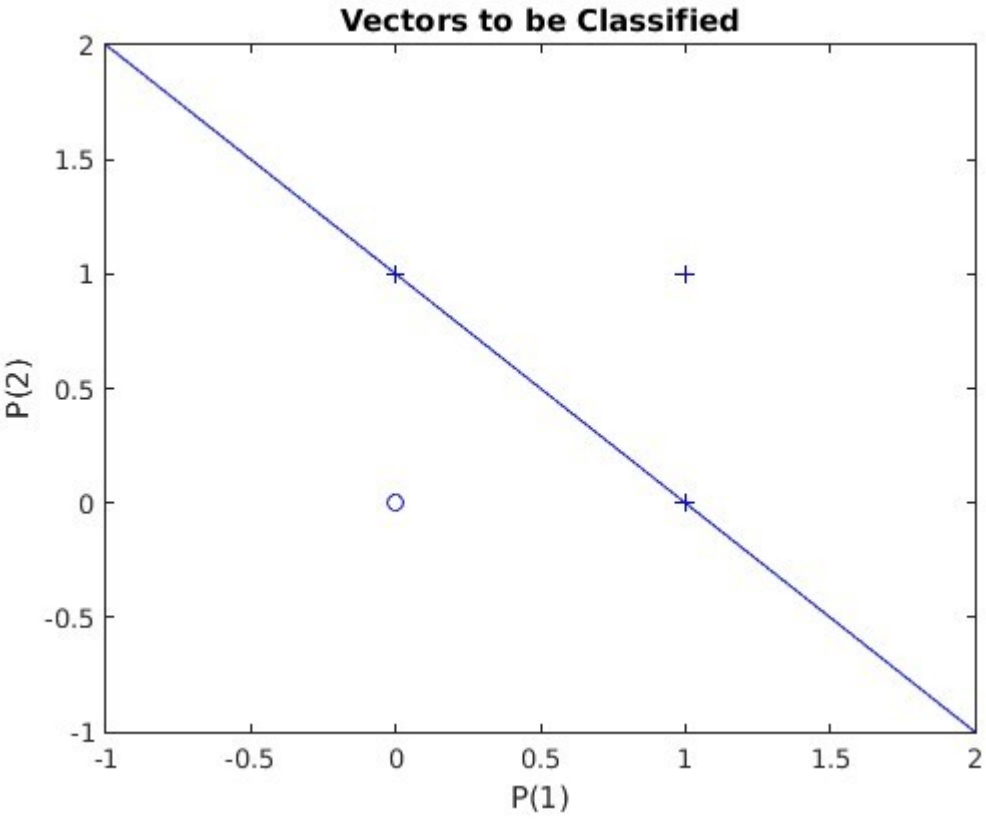
two input and gates

```
% Run_AND_Perceptron.m
P = [0 0 1 1; 0 1 0 1]; % AND Function
T = [0 0 0 1];
plotpv(P,T,[-1, 2, -1, 2]); % plot data
% initial weight vector and bias
W = [1 1]; b = -1;
plotpc(W,b); % plot line
epoc = 1; % number of epoc
for j=1:epoc
for i=1:size(P,2)
p = P(:,i);
t = T(i);
net=perceptron;
[W b] = train(p',t,W,b);
plotpc(W,b);
end
end
% test data test = [0 0]'
test = [0 0];
output = hardlim(W*test'+b)
```



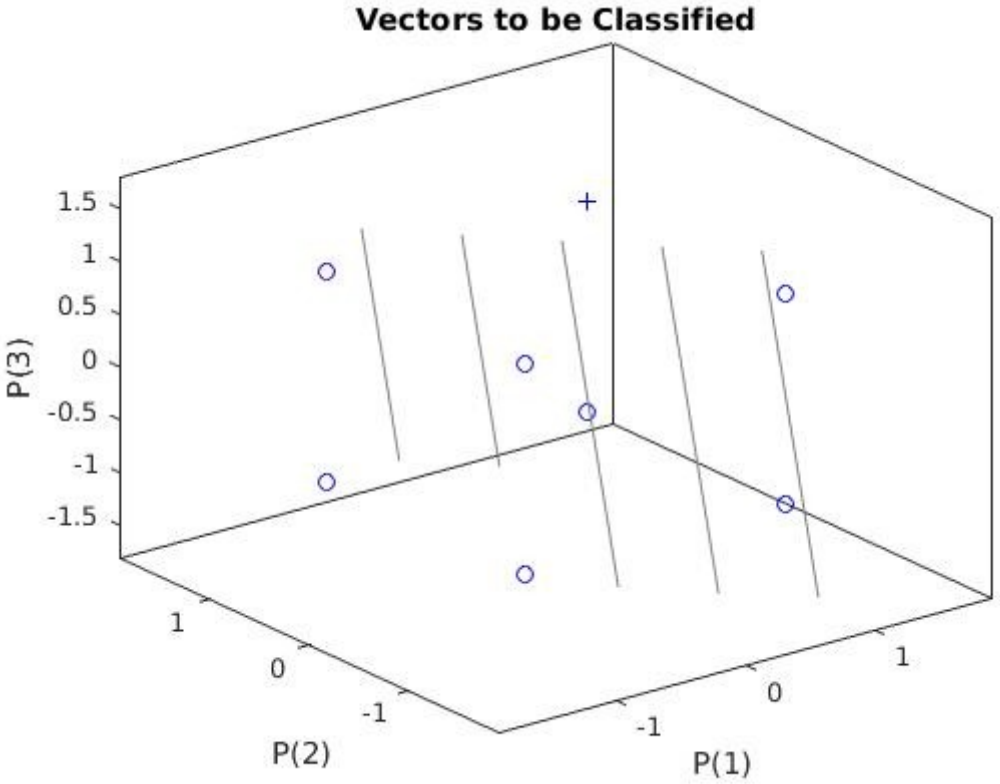
3)t
hr
ee

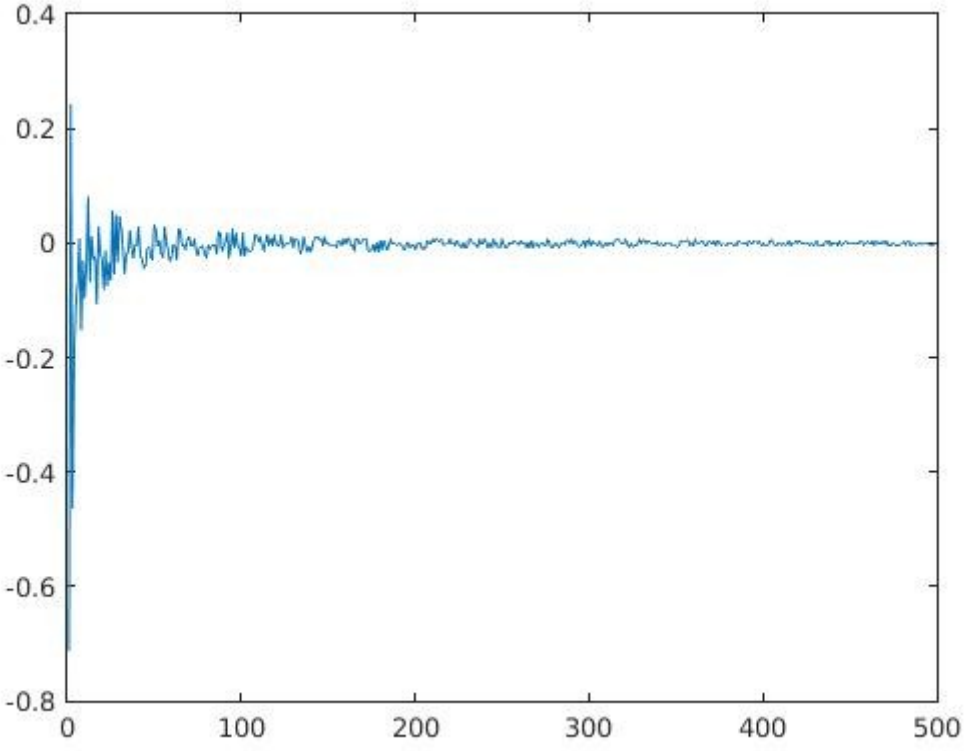

5
4
3
2
1
0
-1
-2
-1
0
1
2
3
4
5

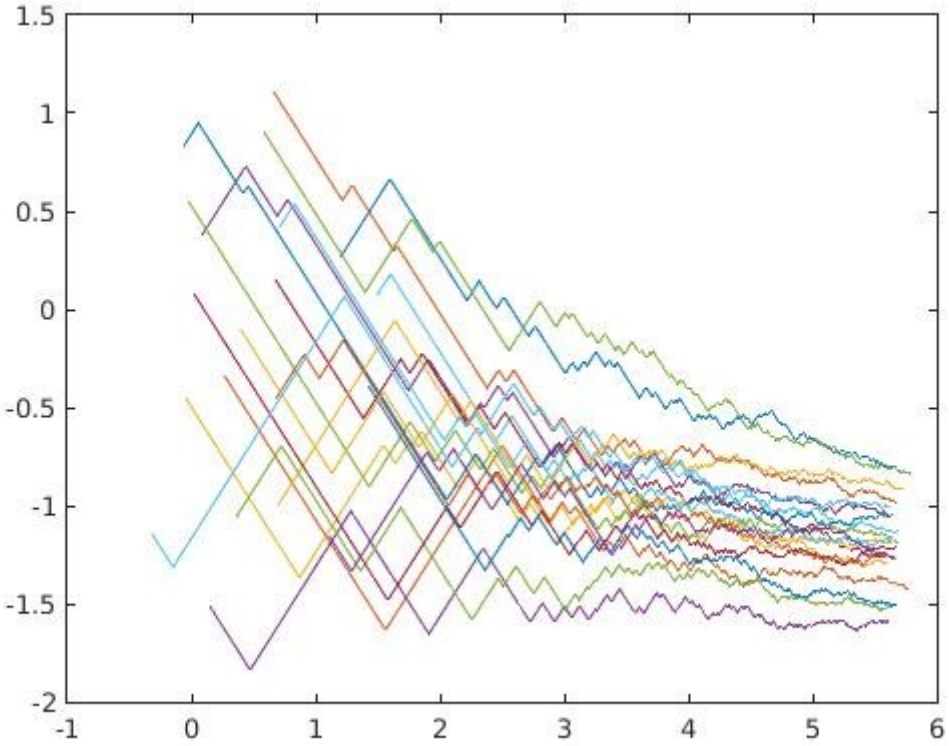

Vectors to be Classified
2
1.5
1
0.5
0
-0.5
-1
P(2)
-1
-0.5
0
0.5
1
1.5
2
P(1)

4)three input and gates

```
clc;
clear all;
close all;
input=[1 -1 -1 -1 1 1 1 -1; 1 -1 1 -1 1 -1 -1 1;1 -1 1 1 -1 -1 1 -1];
%b=1;
output=[1 0 0 0 0 0 0 0];
weight=zeros(500,2,20);
bias = zeros(500,1,20);
err = zeros(500,1,20);
x=input;
t=output;
%[row col]= size(x);
%out = zeros(1,4);
for k=1:20
    weight=rand([1,3])*2-1;
    bias=rand([1,1])*2-1;
    weight1(1,:,1)=weight;
    bias1(1,:,1)=bias;
    for j=1:500

        r = randi(4);
        x1(:,j)=x(:,r);
        y(1,j)=(weight*x(:,r)+bias);
        out(1,j) = (1/(1+exp(-y(1,j))));
        e=t(r)-out(j);
        bias=bias+e;
        weight=weight+e.*transpose(x(:,r));
        weight1(j,:,k)= weight;
        bias1(j,1,k) = bias;
        err(j,1,k) = e;

    end
    plot(weight1(:,1,k),bias1(:,1,k)); hold on;
end

for j= 1:500
    ave = err(j,1,1);
    for k =1:20
        ave = ave + (err(j,1,k)/20);
    end
    mse(j,1)=ave;
end
[ x1' out']
figure;
 plotpv(x,t);
 plotpc(weight,bias);
figure;
plot(mse)
```

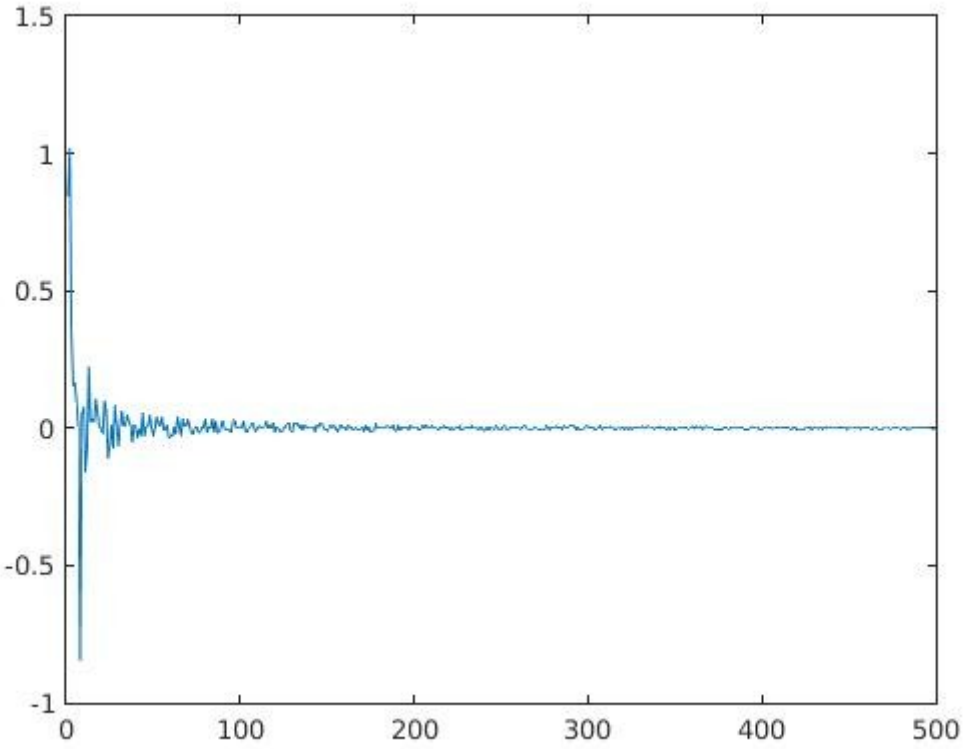

Vectors to be Classified
P(3)
1.5
1
0.5
0
-0.5
-1
-1.5
P(2)
1
0
-1
P(1)
-1
0
1

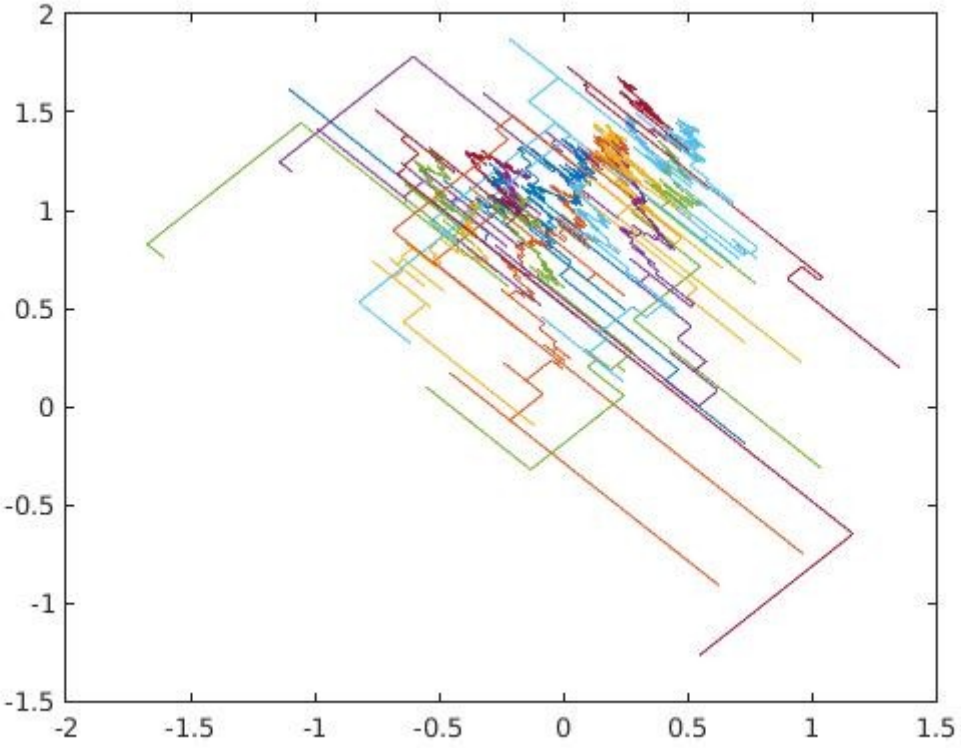

0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
100
200
300
400
500

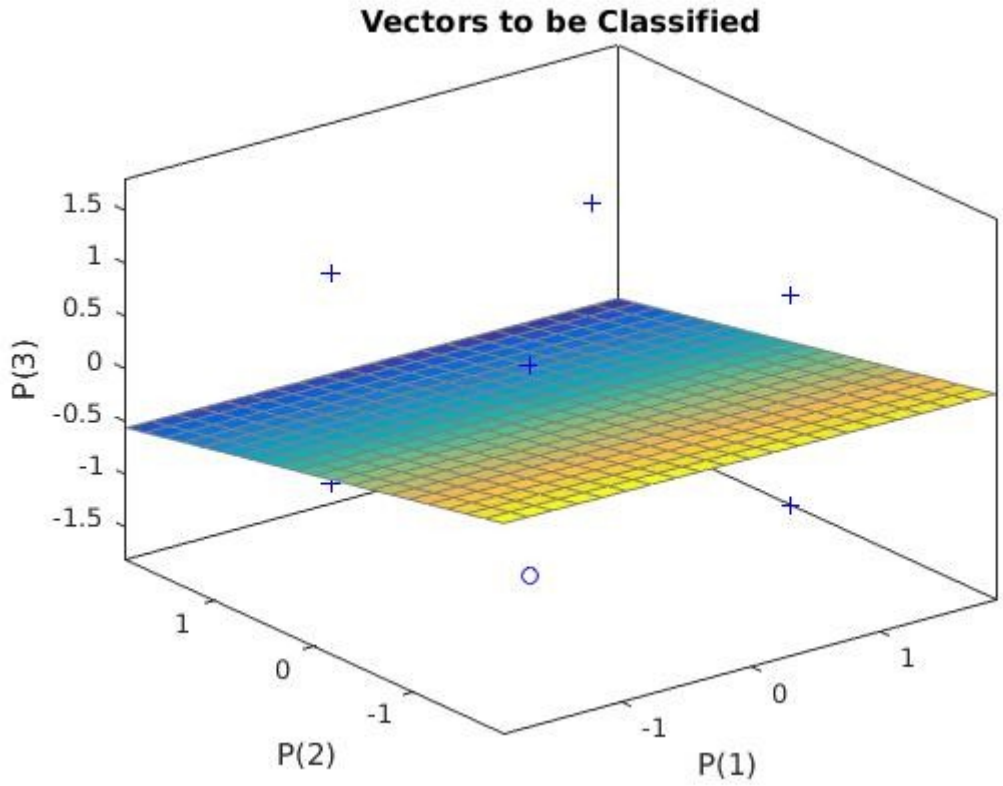

1.5
1
0.5
0
-0.5
-1
-1.5
-2
-1
0
1
2
3
4
5
6

4)three input or gates

```
  clc;
clear all;
close all;
input=[1 -1 -1 -1 1 1 1 -1; 1 -1 1 -1 1 -1 -1 1;1 -1 1 1 -1 -1 1 -1];
%b=1;
output=[1 0 1 1 1 1 1 1];
weight=zeros(500,2,20);
bias = zeros(500,1,20);
err = zeros(500,1,20);
x=input;
t=output;
%[row col]= size(x);
%out = zeros(1,4);
for k=1:20
    weight=rand([1,3])*2-1;
    bias=rand([1,1])*2-1;
    weight1(1,:,1)=weight;
    bias1(1,:,1)=bias;
    for j=1:500

        r = randi(4);
        x1(:,j)=x(:,r);
        y(1,j)=(weight*x(:,r)+bias);
        out(1,j) = (1/(1+exp(-y(1,j))));
        e=t(r)-out(j);
        bias=bias+e;
        weight=weight+e.*transpose(x(:,r));
        weight1(j,:,k)= weight;
        bias1(j,1,k) = bias;
        err(j,1,k) = e;

    end
    plot(weight1(:,1,k),bias1(:,1,k)); hold on;
end

for j= 1:500
    ave = err(j,1,1);
    for k =1:20
        ave = ave + (err(j,1,k)/20);
    end
    mse(j,1)=ave;
end
[ x1' out']
figure;
 plotpv(x,t);
 plotpc(weight,bias);
figure;
plot(mse)
```

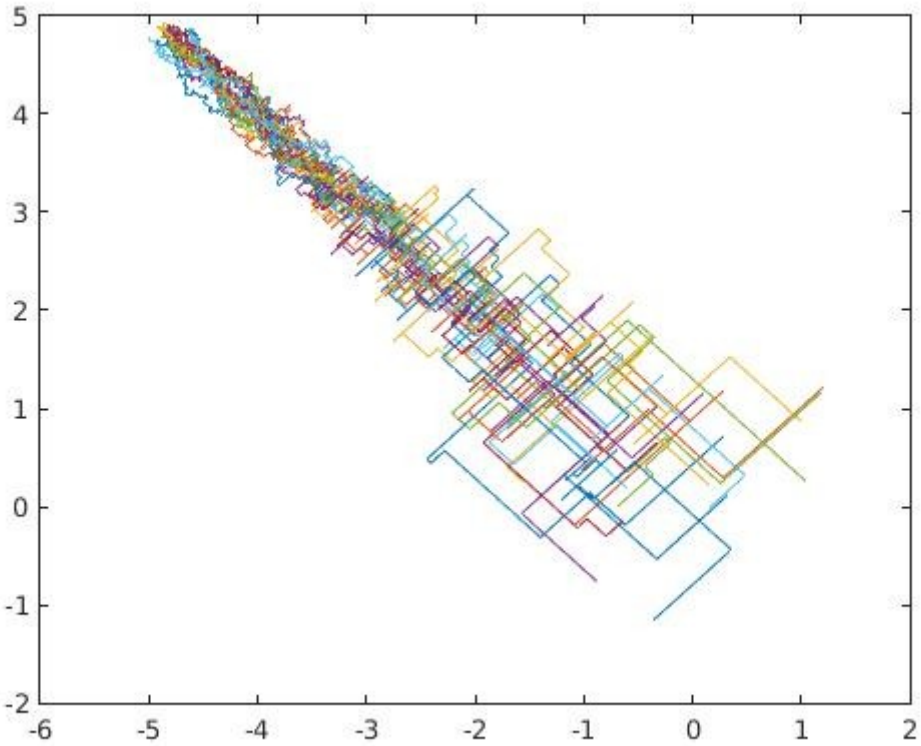

1.5
1
0.5
0
-0.5
-1
0
100
200
300
400
500

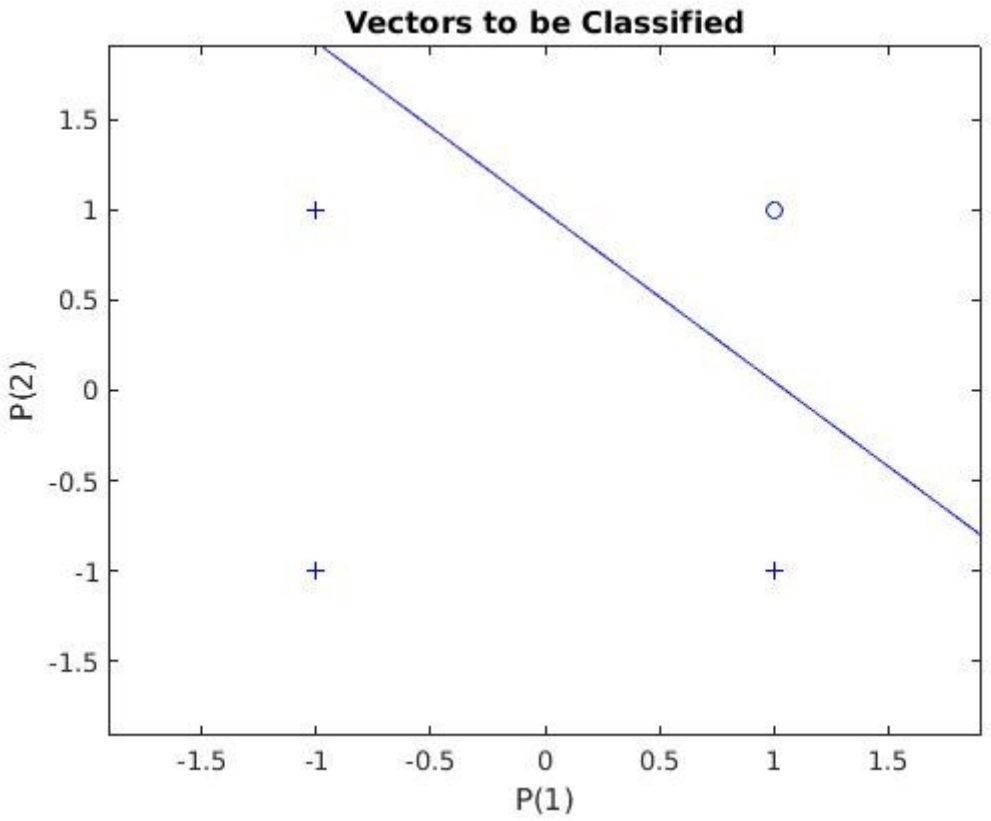

2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5


Vectors to be Classified
1.5
1
0.5
0
-0.5
-1
-1.5
P(3)
1
0
-1
P(2)
-1
0
1
P(1)

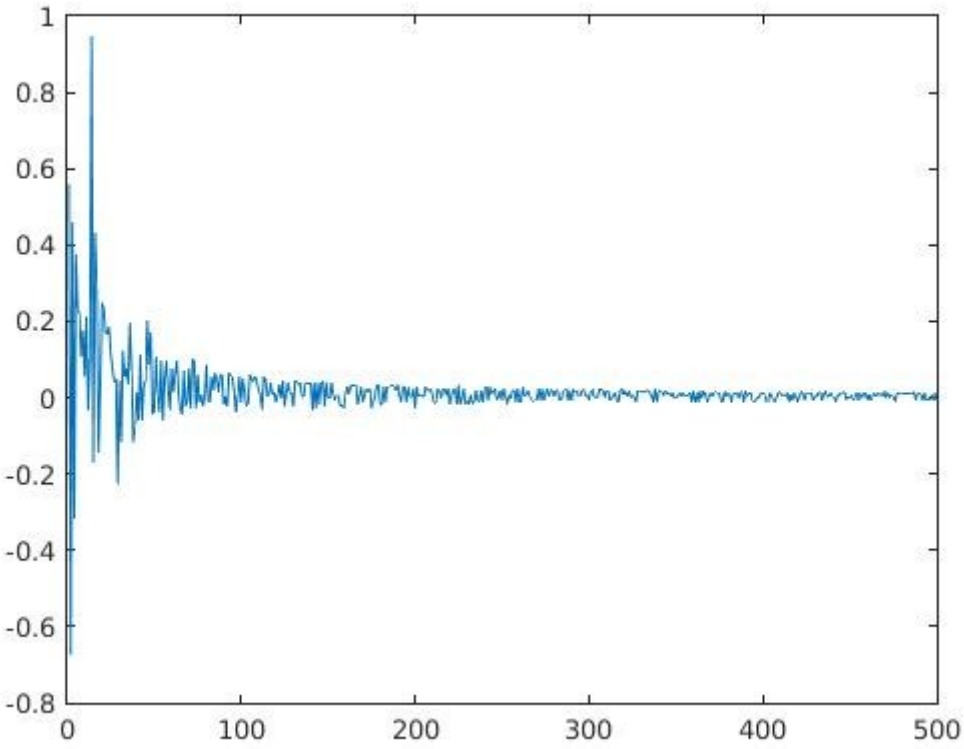
2 input nand gate

```
clc;
clear all;
close all;
%input=[1 1 1 1 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 ; 1 1 1 1 -1 -1 -1 -1 1 1 1 1 -1 -1 -1 -1; 1 1 -1
-1 1 1 -1 -1 1 1 -1 -1 1 1 -1 -1;1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 ];
%b=1;
input=[1 1 -1 -1; 1 -1 -1 1];
output=[0 1 1 1];
weight=zeros(500,2,20);
bias = zeros(500,1,20);
err = zeros(500,1,20);
x=input;
t=output;
%[row col]= size(x);
%out = zeros(1,4);
for k=1:20
    weight=rand([1,2])*2-1;
    bias=rand([1,1])*2-1;
    weight1(1,:,1)=weight;
    bias1(1,:,1)=bias;
    for j=1:500

        r = randi(4);
        x1(:,j)=x(:,r);
        y(1,j)=(weight*x(:,r)+bias);
        out(1,j) = (1/(1+exp(-y(1,j))));
        e=t(r)-out(j);
        bias=bias+e;
        weight=weight+e.*transpose(x(:,r));
        weight1(j,:,k)= weight;
        bias1(j,1,k) = bias;
        err(j,1,k) = e;

    end
    plot(weight1(:,1,k),bias1(:,1,k)); hold on;
end

for j= 1:500
    ave = err(j,1,1);
    for k =1:20
        ave = ave + (err(j,1,k)/20);
    end
    mse(j,1)=ave;
end
[ x1' out']
figure;
 plotpv(x,t);
 plotpc(weight,bias);
figure;
plot(mse)
```


5
4
3
2
1
0
-1
-2
-6
-5
-4
-3
-2
-1
0
1
2

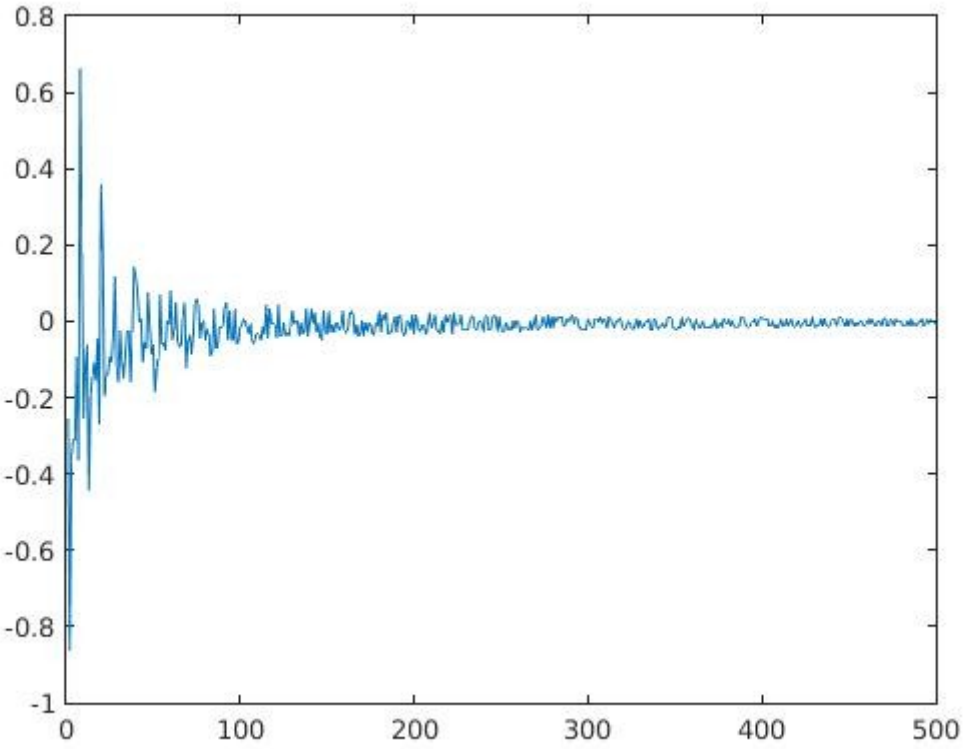

Vectors to be Classified
1.5
1
0.5
0
-0.5
-1
-1.5
P(2)
-1.5
-1
-0.5
0
0.5
1
1.5
P(1)

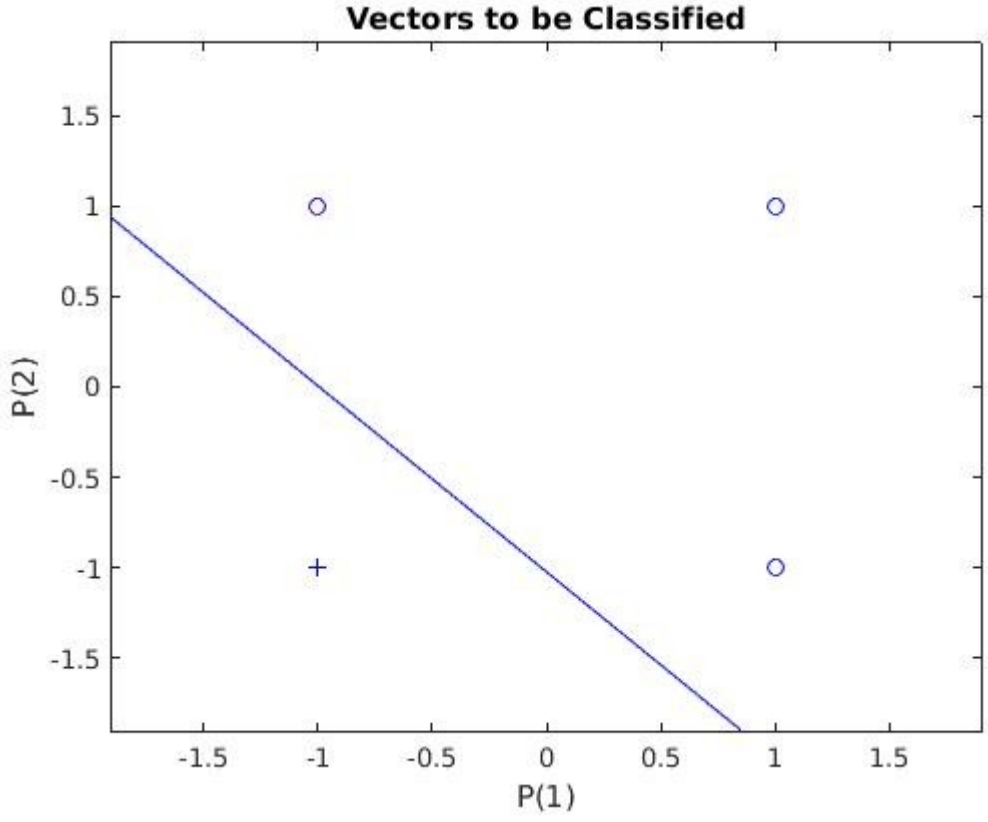
'

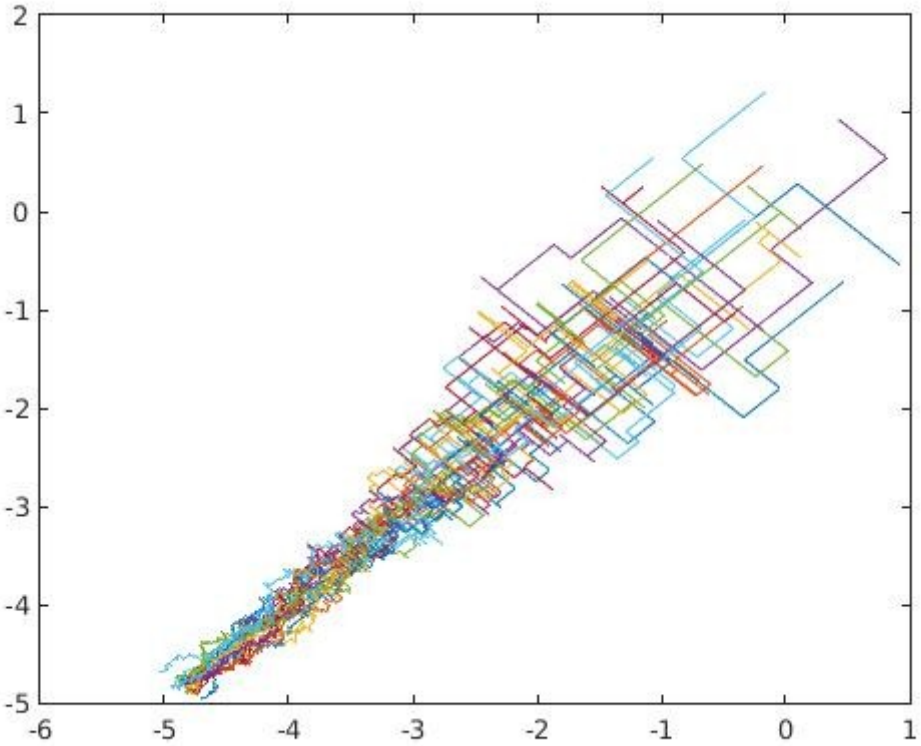

1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
100
200
300
400
500

2 input nor gate

```
clc;
clear all;
close all;
%input=[1 1 1 1 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 ; 1 1 1 1 -1 -1 -1 -1 1 1 1 1 -1 -1 -1 -1; 1 1 -1
-1 1 1 -1 -1 1 1 -1 -1 1 1 -1 -1;1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 ];
%b=1;
input=[1 1 -1 -1; 1 -1 -1 1];
output=[0 0 1 0];
weight=zeros(500,2,20);
bias = zeros(500,1,20);
err = zeros(500,1,20);
x=input;
t=output;
%[row col]= size(x);
%out = zeros(1,4);
for k=1:20
    weight=rand([1,2])*2-1;
    bias=rand([1,1])*2-1;
    weight1(1,:,1)=weight;
    bias1(1,:,1)=bias;
    for j=1:500

        r = randi(4);
        x1(:,j)=x(:,r);
        y(1,j)=(weight*x(:,r)+bias);
        out(1,j) = (1/(1+exp(-y(1,j))));
        e=t(r)-out(j);
        bias=bias+e;
        weight=weight+e.*transpose(x(:,r));
        weight1(j,:,k)= weight;
        bias1(j,1,k) = bias;
        err(j,1,k) = e;

    end
    plot(weight1(:,1,k),bias1(:,1,k)); hold on;
end

for j= 1:500
    ave = err(j,1,1);
    for k =1:20
        ave = ave + (err(j,1,k)/20);
    end
    mse(j,1)=ave;
end
[ x1' out']
figure;
 plotpv(x,t);
 plotpc(weight,bias);
figure;
plot(mse)
```

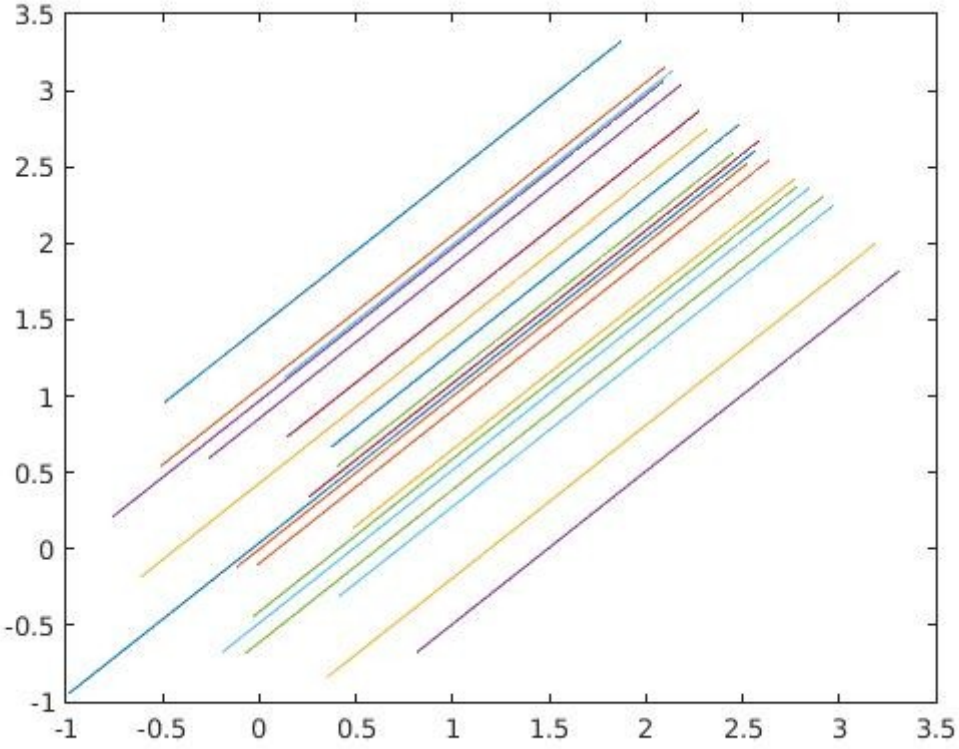

0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
100
200
300
400
500

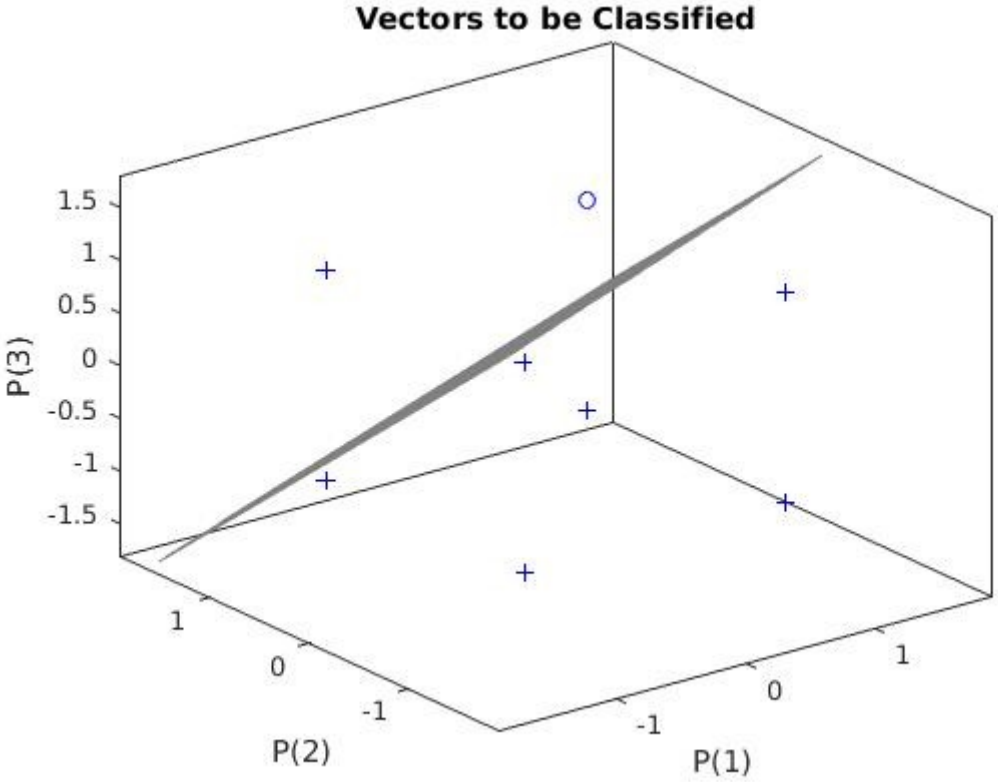

Vectors to be Classified
1.5
1
0.5
0
-0.5
-1
-1.5
P(2)
-1.5
-1
-0.5
0
0.5
1
1.5
P(1)

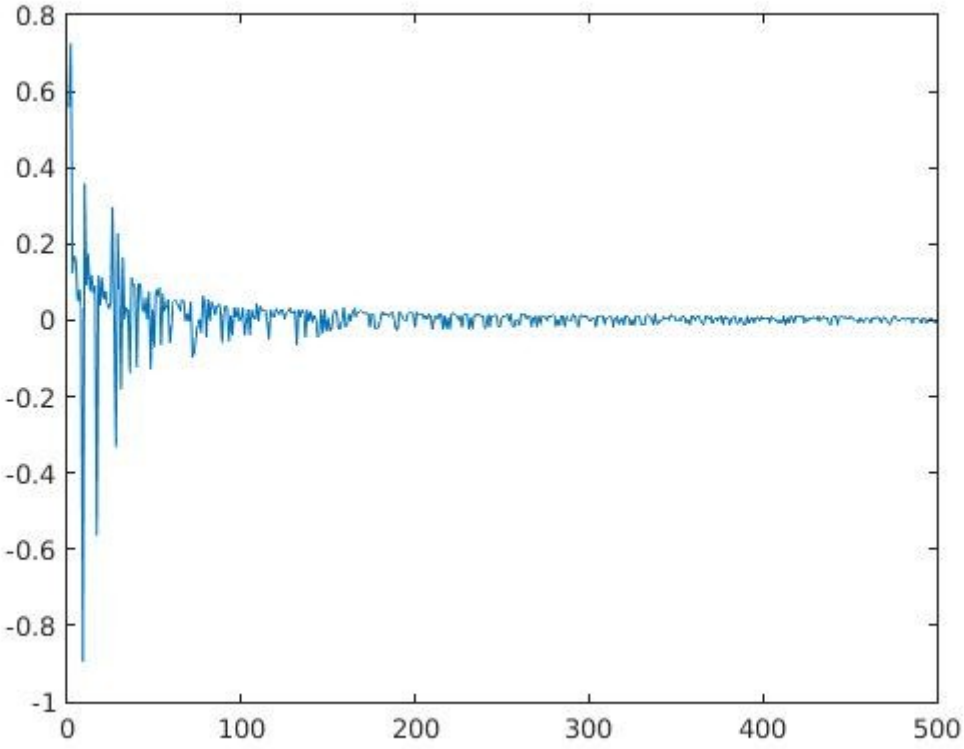

2
1
0
-1
-2
-3
-4
-5
-6
-5
-4
-3
-2
-1
0
1

3 input nand gate

```
clc;
clear all;
close all;
%input=[1 1 1 1 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 ; 1 1 1 1 -1 -1 -1 -1 1 1 1 1 -1 -1 -1 -1; 1 1 -1
-1 1 1 -1 -1 1 1 -1 -1 1 1 -1 -1;1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 ];
%b=1;
input=[1 1 1 1 -1 -1 -1 -1; 1 1 -1 -1 1 1 -1 -1;1 -1 1 -1 1 -1 1 -1];
output=[0 1 1 1 1 1 1 1];
weight=zeros(500,2,20);
bias = zeros(500,1,20);
err = zeros(500,1,20);
x=input;
t=output;
%[row col]= size(x);
%out = zeros(1,4);
for k=1:20
    weight=rand([1,3])*2-1;
    bias=rand([1,1])*2-1;
    weight1(1,:,1)=weight;
    bias1(1,:,1)=bias;
    for j=1:500

        r = randi(4);
        x1(:,j)=x(:,r);
        y(1,j)=(weight*x(:,r)+bias);
        out(1,j) = (1/(1+exp(-y(1,j))));
        e=t(r)-out(j);
        bias=bias+e;
        weight=weight+e.*transpose(x(:,r));
        weight1(j,:,k)= weight;
        bias1(j,1,k) = bias;
        err(j,1,k) = e;

    end
    plot(weight1(:,1,k),bias1(:,1,k)); hold on;
end

for j= 1:500
    ave = err(j,1,1);
    for k =1:20
        ave = ave + (err(j,1,k)/20);
    end
    mse(j,1)=ave;
end
[ x1' out']
figure;
 plotpv(x,t);
 plotpc(weight,bias);
figure;
plot(mse)
```


3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
1.5
2
2.5
3
3.5


Vectors to be Classified
P(3)
1.5
1
0.5
0
-0.5
-1
-1.5
P(2)
1
0
-1
P(1)
-1
0
1


0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
100
200
300
400
500