# 5. Implement and test RBFN

clear all;

close all;

clc;

no\_of\_classes=input('Enter the number of classes');

training\_patterns=2^no\_of\_classes

center\_class=[0 0;1 1];

x=[0 0;0 1;1 0;1 1]

out=[0 1 0 1];

[rows,col]=size(x);

d\_max=sqrt(((center\_class(2,2)-center\_class(1,2))^2)+((center\_class(2,1)-center\_class(1,1))^2))

a=(-1\*(no\_of\_classes/(d\_max\*d\_max)))

for j=1:no\_of\_classes

for i=1:4

if j==1

aa=((x(i,1)-center\_class(1,1))^2)+((x(i,2)-center\_class(1,2))^2);

if aa<0

aa=aa\*-1;

end

Q1(j,i)=exp(a\*aa);

else

aa=((x(i,1)-center\_class(2,1))^2)+((x(i,2)-center\_class(2,2))^2);

if aa<0

aa=aa\*-1;

end

Q1(j,i)=exp(a\*aa);

end

end

end

for i=1:4

Q1(3,i)=1;

end

G=Q1'

d=out'

G1=(G'\*G);

G1=1./G1

G2=G'\*d

W=G1\*G2

**Output:**

Enter the number of classes 2

training\_patterns =

4

x =

0 0

0 1

1 0

1 1

d\_max =

1.4142

a =

-1.0000

G =

1.0000 0.1353 1.0000

0.3679 0.3679 1.0000

0.3679 0.3679 1.0000

0.1353 1.0000 1.0000

d =

0

1

0

1

G1 =

0.7758 1.8473 0.5344

1.8473 0.7758 0.5344

0.5344 0.5344 0.2500

G2 =

0.5032

1.3679

2.0000

W =

3.9861

3.0597

1.5000