NNFL

EXPT – 1

```
clc
close all

a=input('Enter Fuzzy set A in []: ');
b=input('Enter Fuzzy set B in []: ');
n=length(a);

choice=input('\nSelect a operation: \n1. Compliment \n2. Union \n3. Intersection \n4.De Morgans Law\n');

switch choice
  case 1
    i=1:n;
    ca(i)=1-a(i);
    cb(i)=1-b(i);
    disp('Compliment of set A : ');
    disp(ca)
    disp('Compliment of set B : ');
    disp(cb)

  case 2
    i=1:n;
    aub(i)=max(a(i),b(i));
    disp('Union of set A and set B : ');
    disp(aub)

  case 3
    i=1:n;
    anb(i)=min(a(i),b(i));
    disp('Intersection of set A and set B : ');
    disp(anb)

  case 4
    i=1:n;
    ca(i)=1-a(i);
    cb(i)=1-b(i);
    aub(i)=max(a(i),b(i));
    anb(i)=min(a(i),b(i));

    disp('De Morgans Law : Compliment of Union of A and B = Intersection of Compliment of A and Compliment of B');
    i=1:n;
    caub(i)=1-aub(i);
    cancb(i)=min(ca(i),cb(i));
    disp('Compliment of Union of A and B')
    disp(caub)
    disp('Intersection of Compliment of A and Compliment of B')
    disp(cancb)
    if (caub==cancb)
      disp('De Morgans Law is proved...')
    end
end
```

EXPT – 4 (Edge Detection)

```
clc
clear all
close all

img=im2double(rgb2gray(imread('peppers.png')));
img = imresize(img,[256 256]);
a=max(max(img));
b=min(min(img));
img_norm=(img-b)/(a-b);
gx=[-1 1];
gy=gx';
Ix=conv2(img_norm,gx);
Iy=conv2(img_norm,gy);

fis1=readfis('expt4');
getfis(fis1);
showrule(fis1);

for i=1:256
  for j=1:256
    Ieval(i,j)=evalfis([Ix(i,j); Iy(i,j)]',fis1);
  end
end

figure(1)
subplot(2,2,1)
imshow(img)
title('Original Image')

subplot(2,2,2)
imshow(Ix)
title('Gradient-X Image')

subplot(2,2,3)
imshow(Iy)
```

```
title('Gradient-Y Image')

subplot(2,2,4)
imshow(Ieval)
title('Edge-Detected Image')
```

EXPT – 5 (Perceptron Learning Rule for AND operation)

```
clc
clear all
close all
p= [1 1 -1 -1;1 -1 1 -1];
t= [1 -1 -1 -1];
alpha=input('Enter the value of alpha: ');
theta=input('Enter the value of theta: ');
w1=rand;
w2=rand;
w=[w1;w2]'
b=0
axis([-2 2 -2 2])
hold on
plot(p(1,1),p(2,1),'*')
plot(p(1,2),p(2,2),'o')
plot(p(1,3),p(2,3),'o')
plot(p(1,4),p(2,4),'o')
linehandle=plotpc(w,b)
% pause
flag=1;
while(flag==1)
for i=1:4
    yin=(p(1,i)*w1)+(p(2,i)*w2)+b;
    if yin>theta
        y=1;
    end
    if yin<-theta
        y=-1;
    end
    if -theta<=yin && yin<=theta
        y=0;
    end
    if y~=t(i)
        w1=w1+(alpha*t(i)*p(1,i));
        w2=w2+(alpha*t(i)*p(2,i));
        b=b+(alpha*t(i));
        disp(w1)
        disp(w2)
        disp(b)
%          axis([-2 2 -2 2])
%          hold on
%          plot(p(1,1),p(2,1),'*')
%         plot(p(1,2),p(2,2),'o')
%         plot(p(1,3),p(2,3),'o')
%         plot(p(1,4),p(2,4),'o')


    else
        flag=0;
    end
    linehandle=plotpc(w,b,linehandle)
    pause
end
end
```

EXPT – 6 (MLP for EX-OR)

```
clc
clear all
close all
p=[1 -1 1 -1; 1 1 -1 -1];
t=[-1 1 1 -1];
s1=4;
s2=1;
net=newff(minmax(p),[s1 s2]);
net.trainParam.lr=0.1;
net.trainParam.goal=0.0001;
net.trainParam.epochs=1000;
net.trainParam.show=1;
net1=train(net,p,t);
y=sim(net1,p)
view(net)
```

EXPT- 07 (Pattern Recognition)

```
clc
clear all
close all
[digit1 digit2  digit3 digit4 digit5 digit6 digit7 digit8 digit9
digit0]=bit_maps;
p=[digit0(:) digit1(:) digit2(:) digit3(:) digit4(:) digit5(:)
digit6(:) digit7(:) digit8(:) digit9(:)];
t=eye(10);
net=newff(minmax(p),[20 10]);
net1=train(net,p,t);
a=digit6(:);
y=sim(net1,a);
view(net1)
% digit6(:)=digit6(:)+rand(45,1)*0.1;
% p1=[p p+rand(45,10)*0.1 p+rand(45,10)*0.2 p+rand(45,10)*0.3];
% t1=[t t t t];
% net=newff(minmax(p1),[20 10]);
% net2=train(net,p1,t1);
% a1=digit6(:);
% y1=sim(net2,a1);
% view(net2)
```

EXPT – 08 (RBF classifier)

```
clc;
clear all;
close all;
A=[rand(1,100);rand(1,100)];
B=[rand(1,100);-rand(1,100)];
C=[-rand(1,100);rand(1,100)];
D=[-rand(1,100);-rand(1,100)];
P=[A(:,1:80) B(:,1:80) C(:,1:80) D(:,1:80)];
Aout=[1;0;0;0];
Bout=[0;1;0;0];
Cout=[0;0;1;0];
Dout=[0;0;0;1];
axis([-1 1 -1 1])
hold on
plot(A(1,:),A(2,:),'*')
plot(B(1,:),B(2,:),'^')
plot(C(1,:),C(2,:),'O')
plot(D(1,:),D(2,:),'.')
text(0.5,0.5,'Class A')
text(0.5,-0.5,'Class B')
text(-0.5,0.5,'Class C')
text(-0.5,-0.5,'Class D')
for i=1:316
    if i<=79
        Aout=[Aout,[1;0;0;0]];
    elseif i<=158
        Bout=[Bout,[0;1;0;0]];
    elseif i<=237
        Cout=[Cout,[0;0;1;0]];
    else
        Dout=[Dout,[0;0;0;1]];
        end
end
T=[Aout Bout Cout Dout];
net=newrbe(P,T);
A1=A(:,81:100);
y=sim(net,A1);
view(net);
```

EXPT – 09 (RBF non separable classifier)

```
clc
clear all
close all
pi=3.14159
x=0:0.25:4
y=sin(pi*x)+cos(pi*x);
net=newrb(x,y);
view(net)
x1=0:0.2:4
y1=sim(net,x1)
figure
hold on
plot(x,y,'*')
plot(x1,y1,'o')
legend('Training','Testing')
xlabel('Inputs')
ylabel('Targets')
```

EXPT – 10 (SOM image classification)

```
clc
clear all
close all

img=imread('rice.png');
figure(1)
imshow(img)
img=double(img(:))';


p=img(1,1:80);

net=newsom(minmax(p),[1 2]);
net1=train(net,p);

y=sim(net1,img);

final=reshape(y(1,:)*256,[256 256]);
figure(2)
imshow(final)
```