Demonstrate training and testing of two input AND gate Using perceptron

clc;

clear all;

close all;

p=[1 -1 1 -1 ; 1 1 -1 -1];

w=(rands(2))';

b=1;

t=[1 -1 -1 -1];

alpha=0.2;

theta=1;

flag=1;

epo=0;

hold on

axis([-2 2 -2 2]);

plot(p(1,1),p(2,1),'x');

plot(p(1,2),p(2,2),'o');

plot(p(1,3),p(2,3),'o');

plot(p(1,4),p(2,4),'o');

line=plotpc(w,b);

pause

while flag

    flag=0;

    for i=1:4

        yin=b+p(1,i)\*w(1)+p(2,i)\*w(2);

        if yin>theta

            y=1;

        end

        if yin>=-(theta)&&yin<=theta

            y=0;

        end

        if yin<-(theta)

            y=-1;

        end

        if y~=t(i)

            flag=1;

            for j=1:2

                w(j)=w(j)+alpha\*t(i)\*p(j,i);

            end

            b=b+alpha\*t(i);

        end

    end

    epo=epo+1;

    line=plotpc(w,b,line);

end

disp('Final Weights');

disp(w);

disp('Epochs');

disp(epo);

Demonstrate training and testing of Multilayer Perceptron for two input EX- OR gate

clc;

clear all;

close all;

p=[1 -1 1 -1 ; 1 1 -1 -1];

t=[-1 1 1 -1];

net=newff(minmax(p),[4,1]);

new=init(net);

net.trainParam.show=1;

net.trainParam.epoch=1000;

net.trainParam.goal=0.001;

net.trainParam.lr=0.4;

[net]=train(net,p,t);

a=sim(net,p);

a1=round(a)

view(net)

Demonstrate application of Multilayer perceptron for Dot Matrix numerical digit recognition

clc;

clear all;

close all;

[digit1,digit2,digit3,digit4,digit5,digit6,digit7,digit8,digit9,digit0] = bit\_maps();

p=[digit1(:),digit2(:),digit3(:),digit4(:),digit5(:),digit6(:),digit7(:),digit8(:),digit9(:),digit0(:)];

t=eye(10);

net=newff(minmax(p),[20 10]);

net1=train(net,p,t);

test=digit7(:);

output=round(sim(net,test))

view(net);

Perform the following classifications MLP network. The vectors (1 1 1 1); (-1 1 -1 -1) are members of one class (target =1). Vectors (1 1 1 -1) and (1 -1 -1 1) are not members of the class (target = -1).

clc;

clear all;

close all;

p=[1 1 1 1;-1 1 -1 -1;1 1 1 -1; 1 -1 -1 1];

t=[1 1 -1 -1];

net=newff(minmax(p),[8 1]);

net1=train(net,p,t);

y=round(sim(net1,p));

disp(y)

Demonstrate application of RBF for Dot Matrix numerical digit recognition.

clc; clear all;

close all;

[digit1,digit2,digit3,digit4,digit5,digit6,digit7,digit8,digit9,digit0] = bit\_maps();

p=[digit1(:),digit2(:),digit3(:),digit4(:),digit5(:),digit6(:),digit7(:),digit8(:),digit9(:),digit0(:)];

t=eye(10);

net=newrb(p,t);

test=digit7(:);

output=round(sim(net,test))

view(net)

Demonstrate application of Radial basis neural network for approximating the given function

clc;

clear all;

close all;

p=[0:0.25:4];

t=cos(pi\*p);

net=newrb(p,t);

plot(p,t,'\*b');

hold on;

test=[0:0.1:4];

y=sim(net,test);

plot(test,y,'or');

legend('training','testing');

xlabel('input')

ylabel('output')

Demonstrate training and testing of RBF for pattern classification application for given data pattern

clc;

clear all;

close all;

a=[rand(1,100);rand(1,100)];

a1=[a(:,1:80)];

plot(a(1,1:100),a(2,1:100),'or');

hold;

grid on;

b=[-1\*rand(1,100);rand(1,100)];

b1=[b(:,1:80)];

plot(b(1,1:100),b(2,1:100),'xb');

hold;

grid on;

c=[rand(1,100);-1\*rand(1,100)];

c1=[c(:,1:80)];

plot(c(1,1:100),c(2,1:100),'xb');

hold;

grid on;

d=(-1)\*[rand(1,100);rand(1,100)];

d1=[d(:,1:80)];

plot(d(1,1:100),d(2,1:100),'or');

hold;

grid on;

p=[a1 b1 c1 d1];

t=[ones(1,80) (-1)\*ones(1,80) ones(1,80) (-1)\*ones(1,80)];

net=newrb(p,t,0.01);

test=a(:,81:100);

out=sim(net,test);

y=round(out);

figure(2)

plot(test,out,'or');

Demonstrate how Kohenan Self Organising map can be used for image classification.

clc;

clear all;

close all;

a=imread('rice.png');

a1=double(a);

[m,n]=size(a1);

image1=reshape(a1,1,m\*n);

image2=image1(1:80);

net=newsom([0,255],[1 2]);

view(net);

net=train(net,image2);

a=sim(net,image1);

b=zeros(256,256);

for i=(1:m\*n)

    if a(1,i)==0

        b(i)=255;

    else

        b(i)=0;

    end

end

imshow(uint8(b));

Construct Kohenan SOM to cluster four vectors [0 0 1 1], [1 0 0 1]; [0 1 0 1]; [1 1 1 1]. The max number of clusters is 2

clc

close all

clear all

P=[0 0 1 1;1 0 0 1;0 1 0 1 ;1 1 1 1];

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

net1=train(net,P);

a=sim(net1,P);

Construct Kohenan SOM to cluster four vectors

(1 1 1 1); (-1 1 -1 -1) (1 1 1 -1) and (1 -1 -1 1). The max number of clusters is 2

clc

close all

clear all

P=[1 1 1 1;-1 1 -1 -1;1 1 1 -1 ;1 -1 -1 1];

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

net1=train(net,P);

a=sim(net1,P);

Demonstrate application of Sugeno Fuzzy system for Image enhancement.

clc

close all

clear all

img=imread('rice.png')

% img=rgb2gray(img)

img1=double(img);

fzzysys=readfis('enhance.fis')

getfis(fzzysys)

showrule(fzzysys);

for i= 1:1:256

    for j=1:1:256

   c(i,j) = evalfis(img1(i,j),fzzysys);

    end

end

imshow(uint8(c));

Demonstrate fuzzy operations AND, OR, complement on two fuzzy sets.

Demonstrate De Morgans laws on two fuzzy sets.

clc;

clear all;

close all;

a = input('Enter a fuzzy set A');

b = input('Enter a fuzzy set B');

c = input('Enter operation to be performed:1=Union 2=Intersection 3=Complement 4=De Morgans ');

d=ones(1,length(a));

abar=d-a;

bbar=d-b;

if(c==1)

    union = max(a,b)

end

if(c==2)

    intersection = min(a,b)

end

if(c==3)

    abar

    bbar

end

if(c==4)

    abar

    bbar

    lhs=max(abar,bbar)

    rhs=d-min(a,b)

end