% Solve a Pattern Recognition Problem with a Neural Network

% Script generated by Neural Pattern Recognition app

% Created 21-May-2023 23:04:22

%

% This script assumes these variables are defined:

%

% x - input data.

% cluster - target data.

clc;

clear;

close;

x = xlsread('x');

cluster = xlsread('clustertarget');

x = x';

t = cluster';

% Choose a Training Function

% For a list of all training functions type: help nntrain

% 'trainlm' is usually fastest.

% 'trainbr' takes longer but may be better for challenging problems.

% 'trainscg' uses less memory. Suitable in low memory situations.

trainFcn = 'trainscg'; % Scaled conjugate gradient backpropagation.

% Create a Pattern Recognition Network

hiddenLayerSize =10;

net = patternnet(hiddenLayerSize, trainFcn);

% Setup Division of Data for Training, Validation, Testing

net.divideParam.trainRatio = 70/100;

net.divideParam.valRatio = 10/100;

net.divideParam.testRatio = 20/100;

% Train the Network

[net,tr] = train(net,x,t);

%fprintf('x = %i \n',x)

% Test the Network

y = net(x);

e = gsubtract(t,y);

performance = perform(net,t,y)

tind = vec2ind(t);

yind = vec2ind(y);

percentErrors = sum(tind ~= yind)/numel(tind);

% View the Network

view(net)

% Plots

% Uncomment these lines to enable various plots.

%figure, plotperform(tr)

%figure, plottrainstate(tr)

%figure, ploterrhist(e)

%figure, plotconfusion(t,y)

figure, plotroc(t,y)

sample1=[1;3;2;2;3;5;2;2;3;1;1;2;2;1;1;1];

y1=net(sample1);

z1=round(y1);

classes1=vec2ind(y1);

sample2=[1;2;2;2;1;2;1;1;3;2;2;2;1;3;1;1];

y2=net(sample2);

z2=round(y2);

classes2=vec2ind(y2);

sample3=[1;4;4;3;1;4;3;4;3;3;5;5;3;3;2;2];

y3=net(sample3);

z3=round(y3);

classes3=vec2ind(y3);