% Solve an Input-Output Fitting problem with a Neural Network

% Script generated by Neural Fitting app

% Created 03-May-2024 21:30:21

%

% This script assumes these variables are defined:

%

% x - input data.

% y - target data.

x = x';

t = y';

% 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 = 'trainbr'; % Bayesian Regularization backpropagation.

% Create a Fitting Network

hiddenLayerSize = 10;

net = fitnet(hiddenLayerSize,trainFcn);

% Setup Division of Data for Training, Validation, Testing

net.divideParam.trainRatio = 80/100;

net.divideParam.valRatio = 5/100;

net.divideParam.testRatio = 15/100;

% Train the Network

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

% Test the Network

y = net(x);

e = gsubtract(t,y);

performance = perform(net,t,y)

% View the Network

view(net)

% Plots

% Uncomment these lines to enable various plots.

%figure, plotperform(tr)

%figure, plottrainstate(tr)

%figure, ploterrhist(e)

%figure, plotregression(t,y)

%figure, plotfit(net,x,t)