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

% Script generated by Neural Fitting app

% Created Mon Mar 27 13:30:27 MST 2017

%

% This script assumes these variables are defined:

%

% Input - input data.

% Target - target data.

x = Input';

t = Target';

% 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. NFTOOL falls back to this in low memory situations.

trainFcn = 'trainlm'; % Levenberg-Marquardt

% Create a Fitting Network

hiddenLayerSize = 10;

net = fitnet(hiddenLayerSize,trainFcn);

% Choose Input and Output Pre/Post-Processing Functions

% For a list of all processing functions type: help nnprocess

net.input.processFcns = {'removeconstantrows','mapminmax'};

net.output.processFcns = {'removeconstantrows','mapminmax'};

% Setup Division of Data for Training, Validation, Testing

% For a list of all data division functions type: help nndivide

net.divideFcn = 'dividerand'; % Divide data randomly

net.divideMode = 'sample'; % Divide up every sample

net.divideParam.trainRatio = 70/100;

net.divideParam.valRatio = 15/100;

net.divideParam.testRatio = 15/100;

% Choose a Performance Function

% For a list of all performance functions type: help nnperformance

net.performFcn = 'mse'; % Mean squared error

% Choose Plot Functions

% For a list of all plot functions type: help nnplot

net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...

'plotregression', 'plotfit'};

% Train the Network

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

% Test the Network

y = net(x);

e = gsubtract(t,y);

performance = perform(net,t,y)

% Recalculate Training, Validation and Test Performance

trainTargets = t .\* tr.trainMask{1};

valTargets = t .\* tr.valMask{1};

testTargets = t .\* tr.testMask{1};

trainPerformance = perform(net,trainTargets,y)

valPerformance = perform(net,valTargets,y)

testPerformance = perform(net,testTargets,y)

% View the Network

view(net)

% Plots

% Uncomment these lines to enable various plots.

%figure, plotperform(tr)

%figure, plottrainstate(tr)

%figure, plotfit(net,x,t)

%figure, plotregression(t,y)

%figure, ploterrhist(e)

% Deployment

% Change the (false) values to (true) to enable the following code blocks.

if (false)

% Generate MATLAB function for neural network for application deployment

% in MATLAB scripts or with MATLAB Compiler and Builder tools, or simply

% to examine the calculations your trained neural network performs.

genFunction(net,'myNeuralNetworkFunction');

y = myNeuralNetworkFunction(x);

end

if (false)

% Generate a matrix-only MATLAB function for neural network code

% generation with MATLAB Coder tools.

genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');

y = myNeuralNetworkFunction(x);

end

if (false)

% Generate a Simulink diagram for simulation or deployment with.

% Simulink Coder tools.

gensim(net);

end