**Abstract**

This code tests the statistical significance of inputs by using an artificial neural network with a flexible structure. The code tries to find the best number of neurons for the neural network for testing the significance of inputs. First of all, create an empty vector y and an empty matrix X by the following commands:

>>y=[];

>>X=[];

Copy and paste the values of the series y and matrix X (values of the independent variable(s)) by opening and  respectively. When a user runs the code by copying and pasting the first line of the code in the command window of MATLAB:

[pvalesallm,Results]=AnnInptSigTest(y,x)

following menu will appear:



The maximum number of neurons in the first and second layers can be any integer value(for the second layer user can choose zero as well). If a user sets the number of neurons to large numbers running programs will take a longer time, but the result will be more reliable. The activation function for the first and second layers can be hyperbolic tangent sigmoid(tansig), Logistic sigmoid(logsig), Gaussian, or any activation function defined in MATLAB. Training algorithm can be training with Bayesian regularization(trainbr), training with Levenberg-Marquardt(trainlm), traingdm, traingda, traindx, etc. Nrepit is for stabilizing results. Because of different initial values for the neural network, the test results changes at each time of running. A Higher number of Nrepit may stabilize the results, but running time will be longer. If you do not have any idea about the number of neurons, please press the OK button for running with default values.

Example:

>> x6=randn(1000,1);

>> x7=randn(1000,1);

>> x8=randn(1000,1);

>> x9=randn(1000,1);

>> y(1,1)=0.2+0.4\*0+0.74\*sin(0)+0.8\*x6(1,1)+0.25\*x7(1,1)^2+randn(1);

>> y(2,1)=0.2+0.4\*y(1,1)+0.74\*sin(0)+0.8\*x6(2,1)+0.25\*x7(2,1)^2+randn(1);

>> for i=3:1000

y(i,1)=0.2+0.4\*y(i-1,1)+0.74\*sin(y(i-2,1))+0.8\*x6(i,1)+0.25\*x7(i,1)^2+randn(1);

end

>> X=[lagmatrix(y,1:5) x6 x7 x8 x9];

>> [pvalesallm,Results]=AnnInptSigTest(y(6:1000,1),X(6:1000,:));



After pressing OK bottom and completion of the wait bar menu, you will see



In the results table X1 and X2 are y(i-1,1) and y(i-2,1) variables which are lagged values of dependent variables. As you can see, the variables(both linear and nonlinear) used for the generation of data(y) are statistically significant, and the irrelevant variables are not significant.

Note: Results in your computer may be slightly different because of the different initial weights of the neural networks chosen randomly.

Reference:

1- Mohammadi, S.(2018), A new test for the significance of neural network inputs, Neurocomputing 273 (2018) 304-322.

Copyright(c) Shapour Mohammadi, University of Tehran, 2018. shmohmad@ut.ac.ir

Keywords: Neural Networks, Input Significance Test, Nonlinear models, Nuisance parameters, Variable selection, Pruning of Neural Networks.