Project Assignment
System Identification 2016

Logistics

This MATLAB-based project assignment is a compulsory part of the System Identification course in the Control Engineering B.Sc. program of the Technical University of Cluj-Napoca. It will be graded and the mark counts for 20% in the final grade of the course. The assignment is carried out in groups of two students, and should take around 10 hours per person to solve, depending on your experience with MATLAB. Each group will receive their own data sets. To receive the files, send as soon as possible an e-mail to the lecturer Lucian Buşoniu at `lucian@busoniu.net`. Please, mention the names of the two members of your group and their e-mail addresses. The assignment must be worked out in the form of a short written report (in English, one report per group).

Please email the report **in PDF format** to the lecturer at the same address. Do not forget to include your names on the title page of the report. It is required to include in the report complete listings of the MATLAB code (functions and scripts) that you developed for solving the assignment problems. In addition to that, your software must also be sent as a ZIP file, with the same email in which you send the report. Use MATLAB version R2006b or higher and mention the version you used.

The deadline for the full report and code is **Friday 16 December 2016, 24:00 (midnight) at the latest**. In addition, an intermediary version of the report and code with the first problem solution must be sent by **Friday 11 November 2016, 24:00**. In case of delays, each newly entered day of delay results in a 1 point decrease in the maximum grade for the assignment (for instance, delivering the full report on 18 December at 00:10 AM leads to a maximum grade of 8 since the second day of delay has been entered). Note that it is strictly forbidden to take over results from other students, as well as copy text from online resources. Failing to obey this rule constitutes plagiarism and will lead to your solution being disqualified and graded with 0.

The assignment consists of two problems, both using feedforward neural networks. In the first problem, the neural network is used to model the behavior of an unknown function. The second problem concerns data-driven black-box modeling of an unknown dynamical system.

Fitting an unknown function

Given is a data set of input-output pairs, where the outputs are generated by an unknown, nonlinear but static function f. The outputs are corrupted by noise, which is assumed to be additive and zero-mean Gaussian. The function has two input variables and one output variable. Your will have to develop a model for this function, using a feedforward neural network. A second data set generated using the same function is provided for validating the developed model. The two data sets will be given as a MATLAB data file, containing one structure for each set. The training data structure is named `id` and the validation data structure `val`. Each of these structures contains the following fields:

- A set of grid coordinates `X` for the inputs, where `X` is a cell array of two vectors, each vector `X{d}` containing n grid points for input dimension d.
- A set of corresponding outputs `Y`, a matrix of size $n \times n$, where `Y(i,j)` is equal to the value of f at point `(X{1}(i),X{2}(j))`.
- The same data is provided in a different, 'flattened' format: the inputs `Xflat`, a wide matrix containing the n^2 input points, one on each column, and the corresponding row vector of outputs `Yflat`.

The 'structured' variants X and Y are easier to use for creating graphs, while the 'flat' variants are more suitable to neural network training and simulation.

Create a neural net to model the unknown function (appropriately choosing the parameters, such as number of layers, number of neurons on each layer, neuron activation functions, training epochs etc.) Train the neural network using the training data set, and validate the obtained model on the different, validation set. Report the mean squared errors for both sets and show a representative plot for the fit on the training and the validation data sets (e.g. true values compared to neural network outputs). Discuss the results, including the choice of parameters and the quality of the model fit on the two data sets. In addition, vary the number of neurons in the network, and study the evolution of the error on the identification and validation datasets, as the number of neurons grows. Discuss the results, relating them to the discussion during lectures on model choice and overfitting in regression.

It is recommended that you use the Neural Network toolbox of MATLAB to solve this problem. The `feedforwardnet` and `train` functions respectively create and train a feed-forward neural network (or you might use `fitnet` which directly produces a network suitable for the fitting problem). Use `doc nnet` to get started with the Neural Network toolbox.

Before working on this assignment, you will need to learn some basic background about neural networks. You can use the following references:

1. R. Babuška, *Artificial Neural Networks*, Chapter 7 of the lecture notes on *Knowledge Based Control Systems*, Delft University of Technology, 2005.

2. A. Jain, J. Mao, and K. Mohuddin, *Artificial Neural Networks: A Tutorial*, Computer 29(3), 1996.

3. C. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press 1996.

The third reference is an advanced textbook, recommended only for in-depth study.

Black-box system identification

Before working on this part of the project, you will need background on *linear* ARX models. These will be taught in lecture 6.

Given is a data set measured on an unknown **dynamic system** with one or more inputs and one output. The order of the dynamics is not larger than three, and the dynamics may be nonlinear while the output may be affected by noise. Your task is to develop a black-box model for this system, using a feedforward neural network. A second data set measured on the same system is provided for validating the developed model. The two data sets will be given in a MATLAB data file, with variables `id` and `val` containing the two sets as objects of type `iddata` from the System Identification toolbox. Recall that the input, output, and sampling time are available on fields `u, y, Ts` respectively. As a backup in case the system identification toolbox is not installed on the computer, `id_array` and `val_array` contain the same two datasets but now in an array format, with the structure: time values on the first column, output on the last column, and inputs on the second column if the system is single-input, and otherwise on the second to next-to-last columns.

Train a nonlinear, neural-network ARX model using the training data, and validate it on the validation data. Choose carefully the model orders and the delay, as well as the neural network parameters. Report the one-step-ahead prediction error, and the simulation error for both sets (use the mean squared error). Show a representative plot for the fit on the training and the validation data sets, for both simulation and prediction. Note that one-step-ahead prediction uses knowledge of the real previous outputs of the system, whereas simulation may not use this knowledge and can only use previous outputs of the model itself. **Therefore, prediction and simulation are two different experiments that you must run.** Discuss the results, including the quality of the model fit on the two data sets.

Like in the fitting problem, using the Neural Network toolbox is recommended. You may develop your own functions for training and testing the nonlinear ARX model, or you may use standard System Identification functions for nonlinear models.

Matlab programming, logistics, and other remarks

Strive for a compact and elegant MATLAB code, avoid the use of loops (`for`, `while`, etc.) and also `if-then` for vector operations are easier and more readable. Search for "vectorization" in the MATLAB help system for helpful tips on the proper MATLAB programming style. However, do not exaggerate with applying vectorization: if the code is clearer andmore readable with loops or if statements, don't be afraid to use them.

If you are less familiar with programming in MATLAB, the following pointers may help. Type `doc` at the command line to access the documentation. A good place to start is the *Getting Started* node of the documentation. Focus especially on *Matrices and Arrays* and *Programming*. A minimal knowledge of *Plotting* is also required in order to present your results in a graphical form.

It is preferable that each group brings their own laptop to the project classes, and always works on the same laptop for the project. This is to prevent incompatibilities between the Matlab versions installed on their computers and the University computers. If this is not possible for you, please approach the lecturer to identify an appropriate solution for your case.