**Universidade de Coimbra**

*Departamento de Engenharia Electrotécnica e de Computadores*

**Controlo Difuso e Aprendizagem - 2020/2021**

Practical Assignment Nº2

**Estimation Using Neural Networks**

## Objectives

The goal of this practical assignment is to learn to work and apply neural networks [Haykin, 1999]. The application context is the estimation of NOx in an oil refinery.

## Work to be Performed

In this work a neural network should be developed to estimate the NOx (oxidation number) in an oil refinery, more specifically in a debutanisation column. The estimation will be based on the key variables that make up the system. Table 1 describes the variables. A multi-layer feedforward neural

| Variable | Description |
|---|---|
| $u_1$ | Temperature at the top of the column |
| $u_2$ | Pressure at the top of the column |
| $u_3$ | Reflux flow |
| $u_4$ | Flow to the next process |
| $u_5$ | Temperature at the base of the column |
| $u_6$ | Temperature 1 at the base of the column |
| $u_7$ | Temperature 2 at the base of the column |
| $y$ | Oxidation number, NOx |

Table 1: Main variables that make up the debutanisation column system.

network should be employed. The base method for training of the neural network will be the error backpropagation algorithm, but specifically a variant of this base method will be explored. To perform the learning of the neural network, a database obtained from the system will be used, including the variables listed in Table 1. A computational simulation tool should be employed to perform the work. The tool can be MATLAB/Simulink, C, C++, a neural networks toolbox, or other tool to be agreed with the professor.

The work to be performed consists of the following steps:

1. Download of the database from Nónio (`https://inforestudante.uc.pt/`).

2. Using variable selection and reduction techniques it is possible to verify what is the most important data on the database. Employing linear and nonlinear techniques, namely the Pearson coefficient, and mutual information, respectively, it is possible to verify that variables $u_6$ e $u_7$ have a strong linear relation among them, and variable $u_2$ is dispensable in the estimation process due to the small amount of information that it contains. In this way, in a next step PCA (*Principal Components Analysis*) can be used to obtain the following variable $u_{6,7}$ that makes the synthesis of the information contained in variables $u_6$ and $u_7$:

$$u_{6,7} = 0.4878\, u_6 + 0.5261\, u_7. \tag{1}$$

   Taking this information into account:

   - Remove variable $u_2$ from the database.

- Substitute variables $u_6$ and $u_7$ by the approximating variable $u_{6,7}$ specified in equation (1).

3. Divide the database in a training dataset and a test dataset.

4. Choose a neural network architecture with one hidden layer. This step involves, for example, the selection of: number of nodes of the input layer; number of nodes of the hidden layer; number of nodes of the output layer; activation function of the neurons of each layer; and the initialisation of the network weights. Train the neural network using the Levenberg-Marquardt Backpropagation (LMB) neural network training algorithm [Hagan *et al.*, 1996] and accompany this step with the choice of the parameters of the algorithm, in order to obtain the best performance for the system.

   To infer the error or the similarity between the real output signal of the neural network and the desired output signal for the neural network there are several possibilities, such as the use of the mean squares error (MSE) [Araújo, 2006] or the use of the correlation coefficient [Papoulis, 1991]. Calculate the MSE for the training and test datasets generated in Step 3.

5. Investigate and present in a succinct form the Extreme Learning Machine (ELM) neural network architecture and training algorithm [Huang *et al.*, 2006], [Liang *et al.*, 2006].

6. Choose a neural network architecture with one hidden layer. This step involves, for example, the selection of: number of nodes of the input layer; number of nodes of the hidden layer; number of nodes of the output layer; activation function of the neurons of each layer; and the initialisation of the network weights. Train the neural network using the ELM neural network training algorithm [Huang *et al.*, 2006], [Liang *et al.*, 2006] and accompany this step with the choice of the parameters of the algorithm, in order to obtain the best performance for the system. Calculate the MSE for the training and test datasets generated in Step 3. The ELM Matlab toolbox [ELM, 2017] can be used and/or adapted.

7. Make an incremental analysis concerning the neural network architectures chosen in Steps 4 and 6. Specifically, vary the following parameters: number of neurons on the hidden layer between 0 and 20; activation function (test two types); and initialisation of the weights (test three methods). Make two separate incremental analyses, one for each of the LMB and ELM algorithms. For each of the LMB and ELM algorithms, train a neural network using each possible combination of the indicated parameters. Calculate the MSE on the training and test datasets. Compare the results of the best neural network architectures. Compare the training times of the neural networks for both the LMB and ELM algorithms.

8. Choose the ten best neural network architectures generated in Step 7 (jointly considering the LMB and ELM results). Make a combination of the outputs generated by their models using the mean. Compare the results, namely with the results of Steps 4, 6, and 7.

### Report and Materials to Deliver

A report should be delivered to the professor in electronic PDF format. The report should contain the following information in the first page: name of the course ("disciplina"), title and number of the practical assignment, names of the students, number of the class ("turma"), number of the group. It should be submitted through the Nónio system (`https://inforestudante.uc.pt/`) area of the "Controlo Difuso e Aprendizagem" course in a single file in `zip` format that should contain all the relevant files that have been involved in the realisation of the practical assignment. The name of the submitted `zip` file should be "`tXgYrZ-cda21.zip`", where "`X`", "`Y`" e "`Z`" are characters that represent the numbers of the class ("turma"), group, and practical assignment, respectively.

## References

[Araújo, 2006] Rui Araújo. *Controlo Inteligente*. Departamento de Engenharia Electrotécnica e de Computadores, Faculdade de Ciências e Tecnologia, Universidade de Coimbra, 2006.

[ELM, 2017] ELM. Extreme Learning Machine: Source Codes. 2017. Nanyang Technological University, Singapore. `http://www3.ntu.edu.sg/home/egbhuang/elm_codes.html`.

[Hagan *et al.*, 1996] Martin T. Hagan, Howard B. Demuth, and Mark Beale. *Neural Network Design*. PWS Publishing, Boston, MA, USA, 1996.

[Haykin, 1999] Simon Haykin. *Neural Networks - A Comprehensive Foundation*. Prentice-Hall, Inc, Upper Saddle River, NJ, USA, second ed., 1999.

[Huang *et al.*, 2006] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme Learning Machine: Theory and Applications. *Neurocomputing*, vol. 70, no. 1-3, pages 489–501, December 2006.

[Liang *et al.*, 2006] Nan-Ying Liang, Guang-Bin Huang, P. Saratchandran, and N. Sundararajan. A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks. *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pages 1411–1423, November 2006.

[Papoulis, 1991] Athanasius Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, Inc, New York, NY, USA, 1991.