KERNEL: A Matlab Toolbox for Knowledge Extraction and Refinement by NEural Learning

Giovanna Castellano, Ciro Castiello and Anna Maria Fanelli

Dipartimento di Informatica,
Università degli Studi di Bari
Via E.Orabona, 4 - I70126 Bari ITALY
[castellano, castiello, fanelli]@di.uniba.it

Abstract. In this paper we present KERNEL, a neuro-fuzzy system for the extraction of knowledge directly from data, and a toolbox developed in the Matlab environment for its implementation. The KERNEL system belongs to the novel approach which concerns the use and representation of explicit knowledge within the neurocomputing paradigm: the Knowledge Based Neurocomputing. A specific neural network is designed, that reflects in its topology the structure of the fuzzy inference model on which is based the KERNEL system. A well-known system identification benchmark is used as illustrative example.

1 Introduction

In recent years there has been a growing interest in combining neural networks and fuzzy systems (see for instance [6], [8], [9]). This approach belongs to the field known as Knowledge-Based Neurocomputing (KBN), which is a discipline concerning methods to address the explicit representation and processing of knowledge where a neurocomputing system is involved [5]. The KBN approach combines the powerful processing capabilities of the neural networks with the explanatory advantages of symbolic representation of knowledge, such as Boolean functions, automata, linguistic rules.

This paper proposes a toolbox for KERNEL (Knowledge Extraction and Refinement by NEural Learning) system for use with Matlab. KERNEL is a neuro-fuzzy system based on KBN approach, in which the explicit representation and the manipulation of knowledge is provided by fuzzy inference methods. As known [10], the fuzzy approach uses human-like reasoning mechanisms, thus the knowledge in the system is expressed in form of linguistic rules (IF-THEN). The final task of KERNEL is to obtain a fuzzy rule base directly from input data and to improve its efficiency and readability. The mathematical description of the neuro-fuzzy model and the presentation of the learning algorithms underlying the KERNEL system are out of the intents of this paper and can be found in our previous papers [2], [3], [4]. The focus of this paper is to describe in detail the functionalities of the Matlab user-friendly environment that has been developed to allow easy utilization of the KERNEL system.

P.M.A. Sloot et al. (Eds.): ICCS 2002, LNCS 2329, pp. 970–979, 2002.

The outline of the paper is the following: in section 2 we briefly describe the KERNEL system and its architecture. Section 3 is dedicated to the presentation of the MATLAB toolbox and its peculiarities. Finally, an example of system identification shows the utilization of the toolbox and the applicability of the proposed approach.

2 The KERNEL System and its architecture

The KERNEL system has been designed to exploit both the benefits deriving from the use of neural networks and the advantages in term of knowledge representation offered by fuzzy reasoning models. In particular, two essential characteristics should be preserved: accuracy and interpretability. To cope with these requirements, the architecture of the system is organized (see Fig. 1) into three distinct components:

- the first component extracts knowledge in form of fuzzy rules directly from data;
- the second component performs a refinement process of the fuzzy rule base in order to improve the modeling accuracy;
- the third component increases the interpretability of the fuzzy rule base through an optimization procedure.

According to the KBN approach, the fuzzy inference model is translated into a neural network: the neuro-fuzzy network (NFN). Therefore, the structure and parameters of the fuzzy rule base correspond to the structure and parameters of the NFN, respectively.

The main feature of the KERNEL system, which distinguishes it from other neuro-fuzzy systems proposed in the literature, resides in the cability of determining the structure of the rule base in an automatic fashion, through learning from data.

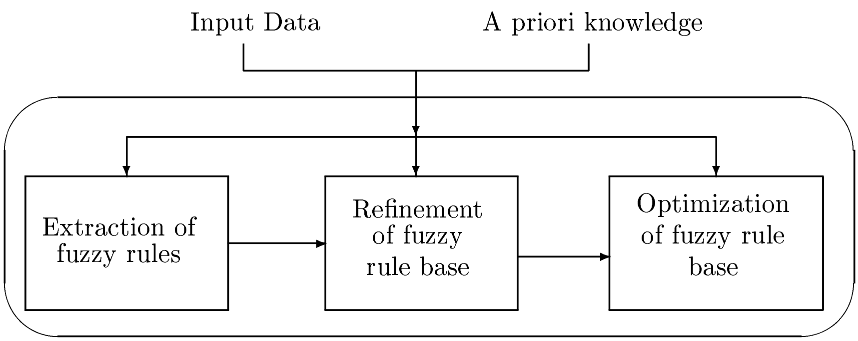

Fig. 1. Scheme of the KERNEL system

2.1 The fuzzy inference model

The KERNEL system is based on a fuzzy inference model which adopts K fuzzy rules of type:

$$\textbf{IF } x_1 \text{is } A_1^k \textbf{ AND} \ldots \textbf{AND } x_n \text{is } A_n^k \textbf{THEN } y_1 \text{is } b_{1k} \textbf{AND } \ldots \textbf{AND } y_m \text{is } b_{mk}, \tag{1}$$

where $x_1, \ldots, x_n$ are the input variables, $y_1, \ldots, y_m$ are the output variables, A_i^k are fuzzy sets and b_{jk} are fuzzy singletons. The membership functions related to the fuzzy sets A_i^k are Gaussian functions in the following form

$$\mu_{ik}(x_i) = \exp(-(x_i - c_{ik})^2/\sigma_{ik}^2), \tag{2}$$

where c_{ik}, σ_{ik} are the center and the width of the Gaussian function, respectively. For every input vector $\mathbf{x} = (x_1, \ldots, x_n)$, the output vector $\mathbf{y} = (y_1, \ldots, y_m)$ can be obtained calculating first the fulfillment of each rule by

$$\mu_k(\mathbf{x}) = \prod_{i=1}^{n} \mu_{ik} x_i, \quad k = 1, \ldots, K, \tag{3}$$

then using the following equation:

$$y_j = \frac{\sum_{k=1}^{K} \mu_k(\mathbf{x}) b_{jk}}{\sum_{k=1}^{K} \mu_k(\mathbf{x})}, \quad j = 1, \ldots, m. \tag{4}$$

2.2 The neuro-fuzzy network

The NFN is composed of four layers (see Fig. 2):

1. The nodes in the first layer just supply the input values $x_i (i = 1, ..., n)$.
2. In the second layer nodes are collected in K groups (corresponding to the K rules), each composed of n units (corresponding to the n fuzzy sets of every rule). These nodes estimate the values of the Gaussian membership functions.
3. The third layer is composed of K units (corresponding to the K rules of the fuzzy model). Each node evaluates the fulfillment degree of every rule.
4. The nodes of the fourth layer supply the final output of the system according to (4).

The NFN described above constitutes the core of KERNEL and it is involved in the three modules mentioned above in the following way:

– in the first module a clustering of input data is performed by carrying out an unsupervised learning of the NFN;
– in the second module the accuracy of the fuzzy rules base is improved performing a supervised learning of the NFN;
– in the third module a pruning algorithm on the network structure and a genetic algorithm on the configuration of the membership functions are performed for the benefit of the final interpretability of the fuzzy rules base.

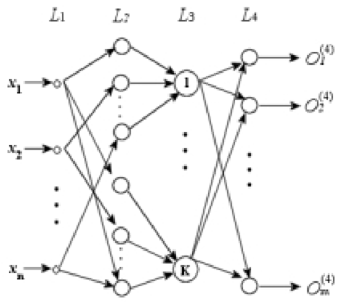

Fig. 2. The neuro-fuzzy network

3 The toolbox

The toolbox has been developed in the Matlab (ver.5.3) environment and it is designed to provide an interface to the system described in the previous section. It is composed of three modules corresponding to the three components of the KERNEL system (see Fig 3). The three modules are endowed with Graphical User Interfaces (GUIs) and have been conceived to work either sequentially (starting from the extraction of knowledge from input data to the final improvement of the fuzzy rules base) or in an unrelated way (according to the results the user needs to obtain).

The fuzzy inference model is represented by a FIS variable containing all the information related to the model and organized in a hierarchic structure. The input data can be loaded from disk or from the Matlab workspace in the form of numerical matrices.

The toolbox provides a 2D graphical representation of multidimensional data and a tool to visualize the fuzzy rules and test the accuracy of the model available at the moment.

3.1 Structure and parameters initialization

The first module of the toolbox is devoted to the automatic creation of an initial fuzzy rule base starting from the input data. The number of the rules and the parameters of the antecedents and the consequents of the rules are obtained by a clustering process. This is performed through the training of the NFN using a competitive learning algorithm [2]. A peculiarity of this clustering procedure consists in providing automatically the appropriate number of clusters, corresponding to the number of rules for the fuzzy model.

The interface realized for this first module (see Fig. 4) allows the user to load the input data and to represent them in a 2D space. An input pre-elaboration can be performed to normalized and/or shuffle the data before starting the clustering process, which can be carried out calculating two kinds of distance between the input patterns and the clusters centers: the Euclidean and the Mahalanobis

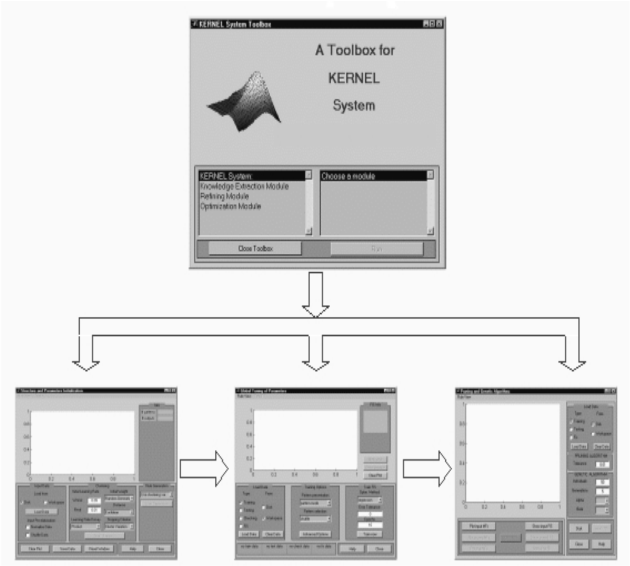

Fig. 3. The toolbox scheme

distance. The first distance performs a spherical clustering, the latter produces an elliptical one.

Once the unsupervised learning of the NFN is completed, the user can generate the initial FIS on the basis of the values obtained after the clustering. The module allows to visualize the fuzzy rule base and to test it on the training or the test set in case it is available.

3.2 Global tuning of parameters

To improve the accuracy of the fuzzy rule base, a supervised learning of the NFN must be performed. The GUI of the second module (see Fig. 5) allows the user to determine the set of parameters for the learning process.

According to the type of problem the system has to deal with (regression or classification tasks), two training algorithms can be chosen which differ in the cost functions [4]. The training of the NFN is carried out using the training set of data and, to avoid a possible overfitting, a checking set can be loaded to perform an early-stopping (see Fig. 6)

3.3 Optimization of the model

The causes of a poor readability of the fuzzy rule base can be found both in a high number of rules and in overlapping membership functions. The third module of the toolbox improves the interpretability of the fuzzy inference model while preserving the accuracy degree achieved in the previous steps. In order to realize

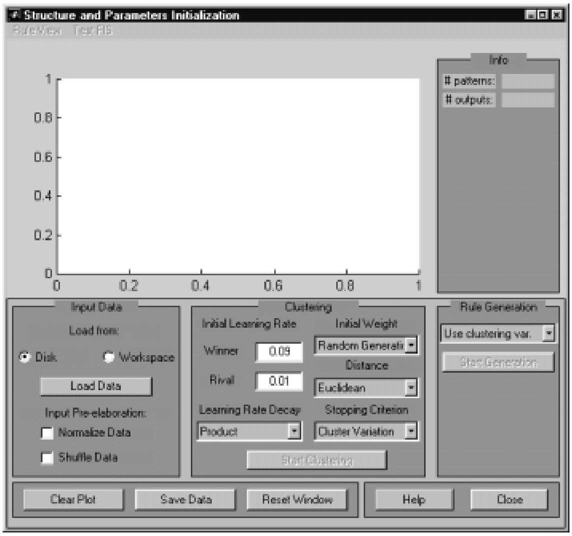

Fig. 4. GUI of the module for knowledge extraction

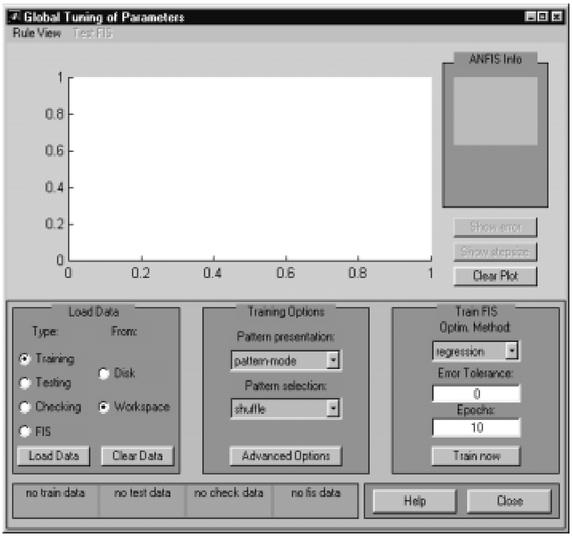

Fig. 5. GUI of the module for knowledge refinement

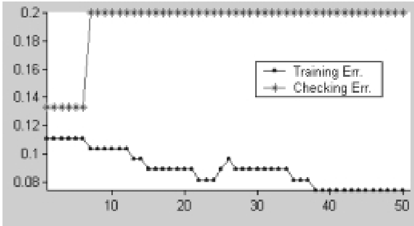

Fig. 6. An example of overfitting reached after six epochs

this task, a two step process is carried out which provides for a reduction of the model structure and a rearrangment of the membership function parameters by a pruning and genetic algorithm, respectively. The first works on the topology of the NFN to simplify it, the latter is based on population of possible membership functions configurations [3].

The user can fix the parameters for the algorithms described above by the GUI implemented for this module (see Fig. 7).

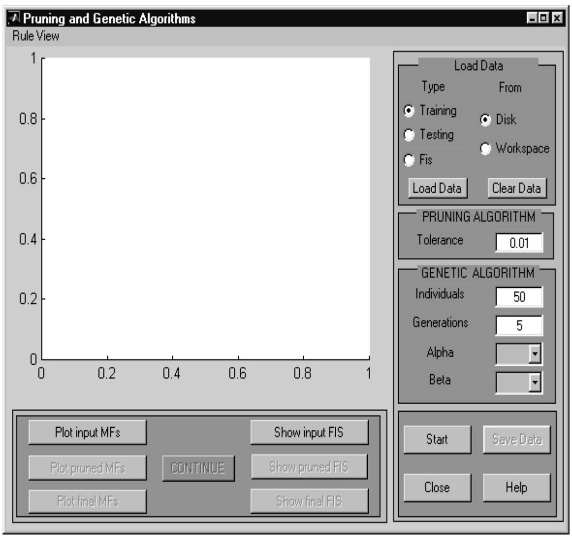

Fig. 7. GUI of the module for knowledge optimization

4 An illustrative example

To illustrate the use of the toolbox, we employ an example concerning the identification of a dynamical nonlinear system using the gas furnace data (series J) of Box and Jenkins [1].

The process is a gas furnace with a single input $u(t)$ representing the gas flow rate into the furnace, and a single output $y(t)$ representing the CO_2 concentration in the outlet gases. The data set consists of 296 pairs of input-output measurements taken with a sampling interval of 9 s. Since the process is a dynamical one, there are different values of the variables that can be considered as candidates to affect the present output $y(t)$. Typically, ten input candidates are considered, i.e. $y(t-1), \ldots, y(t-4), u(t-1), \ldots, u(t-6)$. As in [7], we considered only three input variables: $y(t-1), u(t-3)$ and $u(t-4)$. The produced CO_2 at time t, i.e. $y(t)$, is the output variable. Hence the original 296 data pairs $\langle u(t); y(t) \rangle$ were converted into 290 training points of the form $\langle y(t-1), u(t-3), u(t-4); y(t) \rangle$.

Our experiment starts with the competitive learning phase which performs a spherical clustering of the input data (see Fig. 8). The competitive learning was run with 25 initial clusters randomly generated and produced a network structure with a number of 13 nodes in the third layer (corresponding to 13 fuzzy rules for the initial knowledge base).

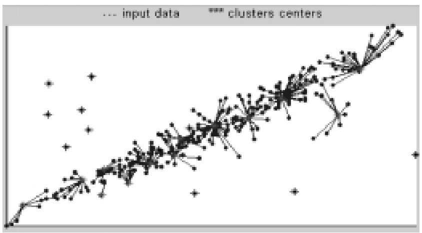

Fig. 8. Partitioning of input data with 13 clusters (the empty clusters will be discarded)

The second module realizes a global tuning of the parameters to improve the accuracy of the knowledge base. In Fig. 9 the output of the fuzzy model is compared with the one of the actual process.

In the end an optimization of the knowledge base was performed by the third module: the number of rules was reduced from 13 to 5 and the readability of the fuzzy membership functions was enhanced as can be seen in the comparison reproduced in Fig. 10.

The results obtained at each step are summed up in Table 1, in terms of numbers of rules and MSE.

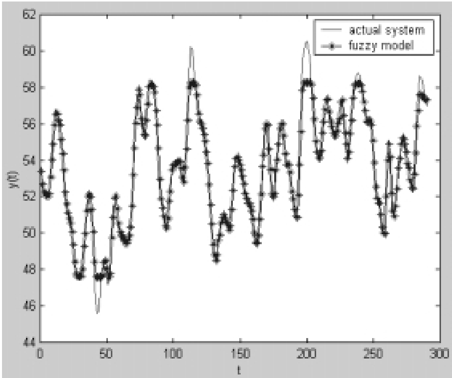

Fig. 9. Comparison between the actual output and the output of the fuzzy model

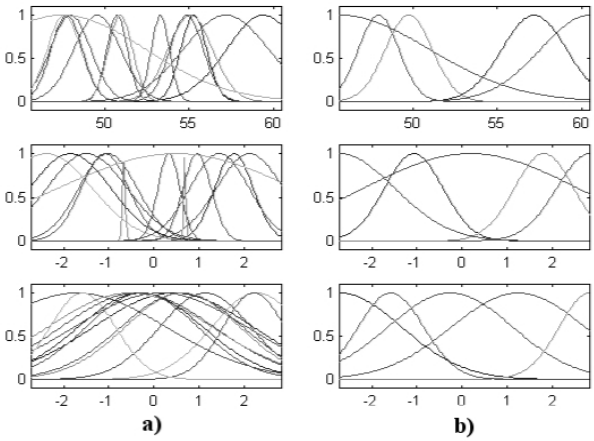

Fig. 10. Comparison of membership functions: initial (a), at the end of the third step (b).

Table 1. Results obtained at each stage of the KERNEL System

Modeling stage		rules number	MSE
stage1: structure and parameters initialization		13	1.3540
stage2: global parameter tuning		13	0.1400
stage3: first iteration	structure reduction	8	0.1459
	local parameter tuning	8	0.1421
second iteration	structure reduction	5	0.1511
	local parameter tuning	5	0.1503

References

1. Box, G. E., Jenkins, G. M.: Time Series Analysis: Forecasting and Control, 2nd ed. Holden Day, San Francisco, CA 1976.
2. Castellano, G., Fanelli, A. M.: Fuzzy inference and rule extraction using a neural network, Neural Network World Journal, 3:361-371, 2000.
3. Castellano, G., Fanelli, A. M.: Complex system modeling by a hybrid approach, International Journal of Knowledge-Based Intelligent Engineering Systems, 3(1): 42-50, 1999.
4. Castellano, G., Fanelli, A. M., Mencar, C.: A new empirical risk functional for a neuro-fuzzy classifier, In Proc. of ESIT 2000, Ahachen, Germany, pp. 429-436, 2000.
5. Cloete, I., Zurada, J. M.: Knowledge-based neurocomputing, The MIT Press, Cambridge, Massachussets, 2000.
6. Nauck, D.: Neuro-Fuzzy systems: review and prospects, Proc. of EUFIT97, pp. 1044-1053, 1997.
7. Sugeno, M., Yasukawa, T.: A Fuzzy-Logic-Based Approach to Qualitative Modeling, IEEE Trans. on Fuzzy Systems, 1:7-31, 1993.
8. Sun, C. T.: Rule-base structure identification in an adaptive-network-based fuzzy inference system, IEEE Trans. on Fuzzy Systems, 2(1):64-73, (1994).
9. Wang, L. X., Mendel, J.: Generating fuzzy rules by learning from examples, IEEE Trans. Syst., Man, and Cyb., 22: 1414-1427, (1992).
10. Zadeh, L. A.: Fuzzy sets, Information and control, 8: 338-353, (1965).