KERNEL: a system for Knowledge Extraction and Refinement by NEural Learning

Giovanna Castellano, Ciro Castiello, Anna Maria Fanelli
Dipartimento di Informatica, Università degli Studi di Bari,
via Orabona 4, 70125 Bari, Italy.

Abstract. This paper presents KERNEL, a neuro-fuzzy system for the extraction of knowledge directly from data. The KERNEL system conforms to the KBN approach which concerns the use and representation of explicit knowledge within the neurocomputing paradigm. 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. For the implementation of the system, a toolbox developed in the Matlab environment is proposed. A well-known classification benchmark is used as illustrative example.

1 Introduction

Recent research efforts extend to combine the processing capabilities of the neural networks with the readability of the structured representation of knowledge. In this context, there has been a considerable interest in the Knowledge-Based Neurocomputing (KBN) approach which is a discipline concerning methods to address the explicit representation and processing of knowledge where a neurocomputing system is involved [6]. In particular, neuro-fuzzy systems are a powerful trade off in terms of readability and efficiency beetwen human-like representations of knowledge and fast learning methods [8], [9], [10].

In this paper we describe KERNEL (Knowledge Extraction and Refinement by NEural Learning): a KBN system which is able to extract and refine knowledge directly from data in the form of fuzzy rules through neural learning. We have presented this system in other previous works [5]. In particular, in this paper we present a toolbox for KERNEL, developed in the Matlab environment.

2 The KERNEL System and its architecture

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

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

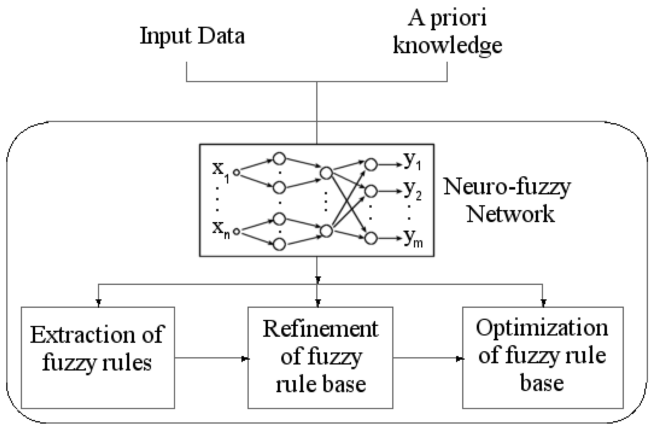

Figure 1: Scheme of the KERNEL system.

where $x_1, \dots, x_n$ are the input variables, $y_1, \dots, y_m$ are the output variables, A_i^k are fuzzy sets and b_{jk} are fuzzy singletons. Fuzzy sets A_i^k are defined by Gaussian membership functions $\mu_{ik}(x_i) = \exp(-(x_i - c_{ik})^2/\sigma_{ik}^2)$, where c_{ik}, σ_{ik} are the center and the width of the Gaussian function, respectively. For every input vector $\mathbf{x} = (x_1, \dots, x_n)$, the output vector $\mathbf{y} = (y_1, \dots, y_m)$ can be obtained by

$$y_j = \frac{\sum_{k=1}^{K} \mu_k(\mathbf{x}) b_{jk}}{\sum_{k=1}^{K} \mu_k(\mathbf{x})}, \quad j = 1, \dots, m, \tag{2}$$

where $\mu_k(\mathbf{x}) = \prod_{i=1}^{n} \mu_{ik} x_i$ is the activation strength of the kth rule.

This fuzzy inference model is encoded in a specific neural network, whose topology reflects the parameters and the organization of the fuzzy rule base. The neuro-fuzzy network is composed of three layers:

1. In the first 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.

2. The second layer is composed of K units (corresponding to the K rules of the fuzzy model). Each node evaluates the fulfilment degree of every rule.

3. The nodes of the third layer supply the final output of the system according to (2).

The neuro-fuzzy network described above constitutes the core of KERNEL system (see Fig. 1), which is organized into three distinct components:

- the first component extracts knowledge from data in the form of a fuzzy rule base. A clustering of data is performed by carrying out an unsupervised learning of the neuro-fuzzy network.

- The second component improves the accuracy of the fuzzy rule base performing a supervised learning of the neuro-fuzzy network.

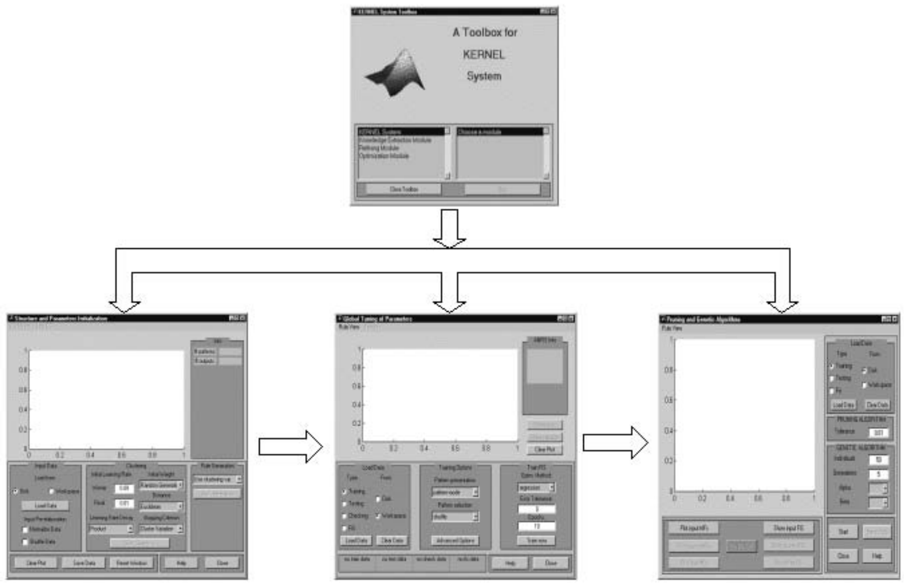

Figure 2: The toolbox for the KERNEL system.

- The third component improves the interpretability of the fuzzy rule base. This is achieved by reducing the number of rules through a structure reduction of the neural network and by enforcing constraints on the input membership functions via a genetic algorithm.

Details about theoretical aspects and implementation of each component of KERNEL can be found in our previous publications ([1], [2], [3], [4]).

3 A Matlab toolbox for KERNEL

A toolbox for the KERNEL system has been developed in the MATLAB ver 5.3 environment[1]. The toolbox is composed of three modules corresponding to the three components of the KERNEL system (Fig. 2). The three modules have been conceived to work either sequentially (starting from the extraction of knowledge from input data to the final improvement of the fuzzy rule base) or in an unrelated way (according to the user needs). The toolbox provides a graphical representation of multidimensional data in 2D and a tool to visualize the fuzzy-rules and test the accuracy of the available model.

The first module of the toolbox initializes the structure and the set of parameters of the model. By means of a graphical interface, the user can load the input data and define the initial set of parameters required to perform the unsupervised learning. In particular, two kinds of clustering can be selected: the first is based on Euclidean distance to perform hyperspherical clustering, the latter uses Mahalanobis distance for hyperelliptical clustering. A peculiarity of the proposed learning process consists in providing automatically the appropriate number of clusters (corresponding to the number of rules for the fuzzy model). Refer to [1] and [2] for more details about the learning algorithms involved in this module.

[1]The Matlab toolbox can be downloaded at www.di.uniba.it\ ∼cilab

Table 1:

	number of rules	% classification	
		Training set	Test set
Unsupervised phase	5	95.6	93.4
Supervised phase	5	97.1	100
Optimization phase	3	87.7	93.3

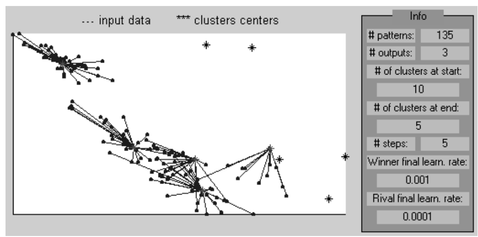

Figure 3: The elliptical clustering.

A global tuning of fuzzy rule parameters is performed by the second module of the toolbox. After selecting the training options, it is possible to start the supervised learning of the neural network. According to the specific task the system has to deal with, i.e. regression or classification problems, two cost functions are available [4].

The third module has been designed to optimize the final structure of the fuzzy inference model [3]. Two algorithms are sequentially applied:

- a pruning algorithm to simplify the structure of the neuro-fuzzy network and, consequently, the fuzzy rule base;
- a genetic algorithm which works on the possible configurations of membership functions to improve the readability of the fuzzy rule base.

The application of these algorithms can be iterated until an accuracy criterion is satisfied.

4 Illustrative example

To illustrate the use of the proposed toolbox, we have considered a well-known benchmark problem concerning classification of Iris flowers [7]. Three species of Iris flowers (setosa, versicolor and virginica) are known. There are 150 samples for this problem, 50 of each class. A sample is a four-dimensional pattern vector representing four attributes of the Iris flower (sepal length, sepal width, petal length, and petal width).

Table 1 summarizes the results achieved on a random-chosen split of the data set in a training set and a test set. Figures 3, 4 and 5 illustrate, respectively, the result of the elliptical clustering (empty clusters are also shown), the trend of the error during the supervised learning and the plot of the final fuzzy rule base.

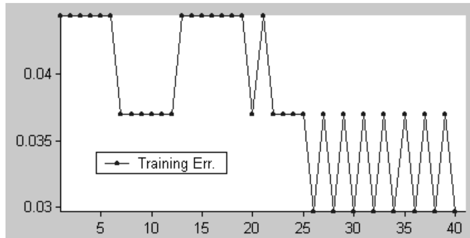

Figure 4: The error plot during the supervised learning process.

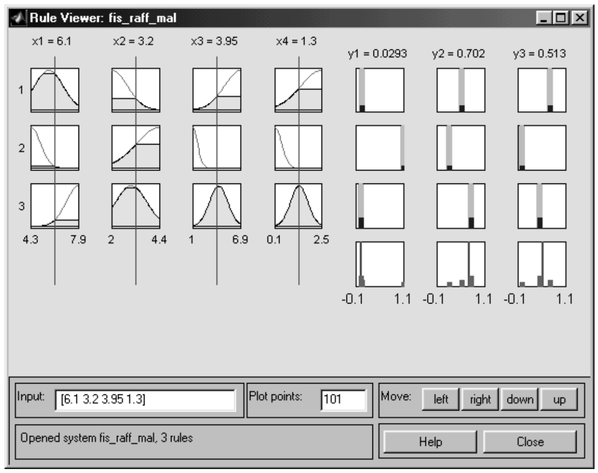

Figure 5: The final fuzzy rule base.

References

[1] G. Castellano, A. M. Fanelli, Fuzzy inference and rule extraction using a neural network, Neural Network World Journal, **3**, (2000) 361–371.

[2] G. Castellano, A. M. Fanelli, A knowledge-based neurocomputing approach to extract linguistic refined rules from data, In E. Esposito, ed., Lecture Notes in Artificial Intelligence - AI*IA 2001: Advances in Artificial Intelligence, Springer Verlag Berlin Heidelberg **2175** (2001) 40–50.

[3] G. Castellano, A. M. Fanelli, Complex system modeling by an hybrid approach, International Journal of Knowledge-Based Intelligent Engineering Systems, **3(1)**, (1999) 42–50.

[4] G. Castellano, A. M. Fanelli, C. Mencar, A new empirical risk functional for a neuro-fuzzy classifier, In Proc. of ESIT 2000, Ahachen, Germany, (2000) 429–436.

[5] G. Castellano, C. Castiello, A.M. Fanelli, KERNEL: A Matlab toolbox for Knowledge Extraction and Refinement by NEural Learning, Lecture Notes in Computer Science, **2329(1)**, Springer Verlag Berlin Heidelberg, (2002) 970–979.

[6] I. Cloete, J. M. Zurada, Knowledge-based neurocomputing, The MIT Press, Cambridge, Massachussets (2000).

[7] R. A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugen., **7**, (1936) 179–188.

[8] D. Nauck, Neuro-Fuzzy systems: review and prospects, Proc. of EUFIT97, (1997) 1044–1053.

[9] C. T. Sun, Rule-base structure identification in an adaptive-network-based fuzzy inference system, IEEE Trans. on Fuzzy Systems, **2(1)**, (1994) 64–73.

[10] L.X. Wang, J. Mendel, Generating fuzzy rules by learning from examples, IEEE Trans. Syst., Man, and Cyb., **22**, (1992) 1414–1427.