Neurocomputing 151 (2015) 955–962

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Rule generation of fuzzy logic systems using a self-organized fuzzy neural network

Juan C. Figueroa-García*, Cynthia M. Ochoa-Rey, José A. Avellaneda-González

[a] Universidad Distrital Francisco José de Caldas, Bogotá - Colombia

ARTICLE INFO

Article history:
Received 7 October 2013
Received in revised form
20 July 2014
Accepted 12 September 2014
Available online 23 October 2014

Keywords:
Fuzzy systems
Rule generation
Neural networks

ABSTRACT

This paper proposes an algorithm for creating rules of a fuzzy logic system using a neuro-fuzzy approach. The proposal is based on the results of Juang and Tsao who use a Fuzzy Neural Network *(FNN)* to generate rules and fuzzy sets from input data. A time series example is solved using our proposal, which is tested using statistical analysis of the residuals of the model.

1. Introduction

One of the most important aspects of a Fuzzy Logic System (FLS) regards to the definition of its *rule base*. As FLSs have many parameters to be defined e.g., the shapes of fuzzy sets, rules, weights and defuzzification methods, its design becomes a complex activity. In this way, the use of optimization techniques to define all those parameters arises as an interesting field for application.

The use of intelligent algorithms for defining the rule base of an FLS seems to be efficient for obtaining accurate results. In our proposal we generate rules of an FLS by using an intelligent method based on a fuzzy neural network, built on the idea that a new data leads to a new rule if it shows a different behavior regarding the existent rules, so if a new data is not involved by any existent rule, then a new rule is generated using back propagation principles (see [1]).

The paper is organized as follows: a Section 1 introduces the problem. Section 2 presents some basics on fuzzy sets. Section 3 presents the proposed methodology. In Section 4, an application example is solved using our proposal and finally in Section 5, some concluding remarks of the proposal are shown.

1.1. Related work

Neuro-fuzzy methods for FLS generation are popular among practitioners. *ANFIS* methodology was proposed by Jyh-Shing Roger Jang [2,3] becoming a useful architecture which gave many ideas to other scientists. Ben et al. [4] used ANFIS in biometric applications. Wang and Fsas [5] designed an improved method for ANFIS rule generation. Figueroa-García and Soriano [6] have compared ANFIS to neural networks in boolean defuzzification applied to time series forecasting problems. Radial basis neural networks are also useful in identification problems (See [7,8]), including human face recognition (See [9]) and polynomial root detection (See [10,11]).

Intelligent algorithms can be applied to either generate or cut-off rules of an FLS. Most of the applications in this field are oriented to generate fuzzy input sets having a full rule-base. The seminal paper of Wang and Mendel [12,13] proposed a method that uses predefined fuzzy sets for defining the combination of inputs given data pairs. Obayashi et al. [14] proposed a neuro fuzzy architecture for generating fuzzy input sets of an FLS while having a full rule base. Numberger and Kruse [15] proposed the use of neuro-fuzzy systems to optimize hierarchical FLSs. Wu and Goo [16] applied neuro-FLSs to time series forecasting problems, and Figueroa-García et al. [17,18] applied neuro Type-2 FLSs to time series forecasting problems.

We focus on the proposal of Juang and Tsao [19] who designed a self-organized neural network for Type-2 fuzzy logic systems, with implementations on FPGAs. Their proposal consists in two parts: a clustering algorithm to generate the rules of the FLS and another one which synchronize the parameters of the fuzzy sets.

Figueroa-García et al. [20] proposed a Type-1 version of the Type-2 version of Juang and Tsao [19], which was tested by Figueroa-García et al. [21]. In this paper we present a full explanation of the proposal with extended technical details, more figures and an extended discussion of its behavior and results.

* Corresponding author.
E-mail addresses: jcfigueroag@udistrital.edu.co (J.C. Figueroa-García), cmochoar@correo.udistrital.edu.co (C.M. Ochoa-Rey), jaavellaneda@correo.udistrital.edu.co (J.A. Avellaneda-González).

http://dx.doi.org/10.1016/j.neucom.2014.09.079
0925-2312/

2. Basic definitions of FLSs

A fuzzy set (A) is a generalization of a *Crisp* or Boolean set. It is defined on an universe of discourse X and is characterized by a *membership function* namely $\mu_A(x)$ that takes values in the interval [0,1]. Then the set A may be represented as a set of ordered pairs of a generic element x and its grade of membership function, $\mu_A(x)$, i.e.,

$$A = \{(x, \mu_A(x)) | \; x \in X\} \tag{1}$$

Now, x can be defined using multiple sets $\{A_1, A_2 \cdots, A_m\}$, each one defined by a membership function $\{\mu_{A_1}(x), \mu_{A_2}(x); \cdots, \mu_{A_m}(x)\}$, where $\mu_A(x)$ is a membership degree of x regarding any fuzzy set F. A is a *linguistic label* which defines the sense of the fuzzy set through the word A. This word defines how an expert perceives the variable X and the shape of A. These concepts lead us to give the following definitions.

Definition 1 (*Membership function* $\mu_A(x)$). The Membership Function of a set A called $\mu_A(x)$ is a function which provides a measure of degree of similarity of an element in X to the fuzzy set A. It takes values in the interval [0,1], that is:

$$\mu_A(x) : X \to [0, 1] \tag{2}$$

After defining the inputs and their fuzzy sets of an FLS, the analyst needs to define a rule base which is a way to represent the knowledge of an expert about the system. Each rule denoted by R^j relates the input variables to a consequence of its occurrence (represented by an output fuzzy set). In this way, each rule R^j can be represented as follows See Mendel [22], Klir & Folger [23]):

$$R^i = \text{IF } x_1 \text{ is } A_1^i \text{ and} \cdots \text{and } x_n \text{ is } A_n^i, \quad \text{THEN } \hat{y} \text{ is } G^i; \quad i = 1, \ldots, M \tag{3}$$

where G^i represents the fuzzified output of the FLS before defuzzification.

In Mandami FLSs, inference is made using *T-norms* and *T-conorms* to represent *and* ($\wedge$) and *or* ($\vee$) operators. After firing each rule, the combination of all operators leads to a single set G^i which is defined as a singleton, as described as follows:

$$\mu_G(x) = \begin{cases} 1 & \text{for } x \\ 0 & \text{for } X \notin x \end{cases} \tag{4}$$

Thus, the last step called *Defuzzification* projects fuzzy inference into a crisp measure. When using singletons as outputs, the most used defuzzification method is the *Center of sets* which is the average of the outputs. It is defined as follows:

$$\hat{y} = \frac{\sum_{i=1}^{M} G^i w_i}{\sum_{i=1}^{M} w_i} = \sum_{i=1}^{M} G^i / M \tag{5}$$

It is important to note that in this kind of Type-1 FLSs, each rule leads to a singleton output. This means that an FLS has as many outputs as rules are defined, so the design of both input fuzzy sets and rule-base become into a important aspect when using FLS as inference method.

Sometimes when having many fuzzy input sets, the analyst can either define useless rules, conflicting rules, and/or duplicated rules which lead to non-desired outputs, and finally lead to an inappropriate response of the FLS. These problems can be avoided using cut-off strategies and testing methods to contrast the output of the FLS against known outputs.

3. A neuro-fuzzy approach for rule generation

As usual, an FLS can be composed by many rules depending on the amount of input–output pairs and their associated fuzzy sets. The natural way to generate rules to link all fuzzy input–output pairs is by asking an expert. Other common way to generate rules is simply by composing all the possible combinations of logical statements, trying to fit a desirable response.

The main idea of this paper is to generate rules of an FLS using intelligent algorithms (either heuristic or optimization methods), in this case neuro-fuzzy methods. Our focus is rule generation using FNNs instead of other intelligent approaches, since FNNs have known properties such as low memory consumption, easiness of implementation, recursive search, and relatively good convergence properties. The proposed FLS is displayed in Fig. 1

Our method is based on the paper of Juang and Tsao [19] who proposed a five-layer neural network, where its main characteristic is the use of Interval Type-2 fuzzy sets (*IT2FS*) as fuzzifiers of the FLS. Now, we propose the use of Type-1 fuzzy sets as fuzzifiers of the FLS in order to reduce its complexity and computing time. The proposed configuration of a five-layer neural network for synchronizing Type-1 FLSs is presented in Fig. 2.

In Fig. 2, input data is represented by n vectors $\{X_1, \ldots, X_n\}$ of size k, and the output data (a.k.a. goal or desired response) is defined as $\hat{y}$ where $(X_j, \hat{y}) \in \mathbb{R}$. Now, the proposed structure is described as follows:

Layer 1 – *Normalization*: In this layer, all input data $x_1, \ldots, x_n$ should be normalized to any of two choices: either the interval of $[-1,1]$ or [0,1].

Layer 2 – *Fuzzification*: This layer defines a Type-1 fuzzy set A for the j_{th} input, A_j^i namely μ_{ji}. The set A_j^i should be derivable, usually defined as a gaussian membership function μ_{ji}.

Layer 3 – *Intersection*: Each node of this layer is a rule of the FLS, and each one uses a t-norm to compose the intersection operation. The output of a node is defined as its

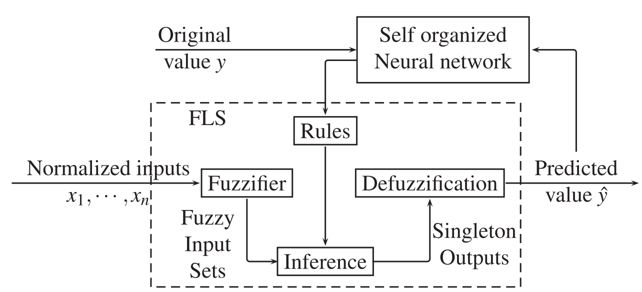

Fig. 1. Neuro-FLS design methodology.

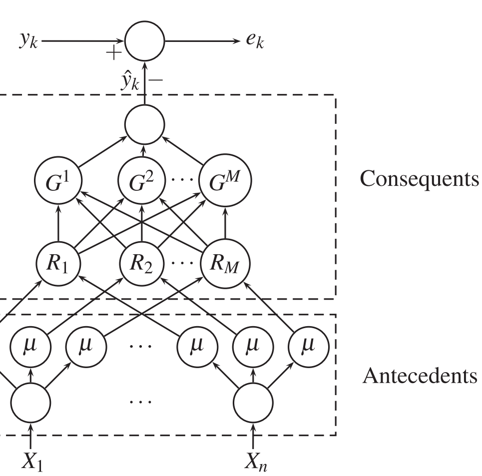

Fig. 2. Proposed neuro-fuzzy architecture.