Available online at www.sciencedirect.com
ScienceDirect
International Journal of Approximate Reasoning
43 (2006) 202–221

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING

www.elsevier.com/locate/ijar

Extraction of similarity based fuzzy rules from artificial neural networks

C.J. Mantas [a,*], J.M. Puche [a], J.M. Mantas [b]

[a] *Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain*
[b] *Department of Software Engineering, University of Granada, 18071 Granada, Spain*

Received 27 October 2005; received in revised form 3 April 2006; accepted 4 April 2006
Available online 3 May 2006

Abstract

A method to extract a fuzzy rule based system from a trained artificial neural network for classification is presented. The fuzzy system obtained is equivalent to the corresponding neural network. In the antecedents of the fuzzy rules, it uses the similarity between the input datum and the weight vectors. This implies rules highly understandable. Thus, both the fuzzy system and a simple analysis of the weight vectors are enough to discern the *hidden* knowledge learnt by the neural network. Several classification problems are presented to illustrate this method of knowledge discovery by using artificial neural networks.
© 2006 Elsevier Inc. All rights reserved.

Keywords: Artificial neural networks; Fuzzy systems

1. Introduction

Artificial neural networks (ANNs) [1–3] are learning models that have been successfully used in many areas such as automatic control, weather forecasting, etc. However, they are *black boxes* and consequently, it is very difficult to understand how an ANN has solved a problem. There are a comprehensive literature about the extraction of knowledge from learning models based on black boxes [4–14].

* Corresponding author.
E-mail address: cmantas@decsai.ugr.es (C.J. Mantas).

0888-613X/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.ijar.2006.04.003

The principle of *Zadeh's incompatibility* [15] established "*the complexity of a system and the precision with which it can be analyzed bear a roughly inverse relation to one another*". This principle can be applied to the ANNs . They are systems with high complexity which can achieve a good approximation to the solution of a problem. However, it is very difficult to analyze their performance. According to this principle, the methods to understand the action carried out by a trained ANN can be included in one of the two following groups:

- To obtain a comprehensible system that approximates the behavior of the ANN (more comprehension → less complexity → less accuracy). In this case, any rule extraction method in [5,9–12] can be used.
- To describe the exact action of the ANN as understandably as possible (same complexity → same accuracy → same comprehension, but with other words). In this case, the methods presented in [6,8,13,14] can be used.

  In [14] a fuzzy rule based system (FRBS) with 2^m fuzzy rules, where m is the number of hidden neurons, equivalent to an ANN is presented. This fact involves that the FRBS obtained is very complex. Therefore it is very difficult to discern knowledge from it. On the other hand, the FRBS presented in this work is also equivalent to the corresponding ANN and it has only 2 rules with straightforward antecedents.

  Other equivalent FRBS is proposed in [6,13]. It has m fuzzy rules, where m is also the number of hidden neurons of the ANN. The antecedents of these rules use the same inputs which are used in the ANN. Therefore, each antecedent has n fuzzy propositions aggregated with the *i-or* operator. Thus, the FRBS has $m \times n$ propositions. This number of propositions makes the comprehension of the fuzzy system complex. In this paper, the method proposed obtains antecedents with only m fuzzy propositions. Hence, the number of propositions in the fuzzy system is equal to $2 \times m$. This reduced number makes the extraction of knowledge from the FRBS easier.

The interested reader can find out a survey of several rule extraction methods in [16–18]. The approach which is proposal here, is able to extract a FRBS from an ANN that carries out classification. This FRBS has the following features:

- The fuzzy rules express exactly the input–output mapping of the ANN. Thus, a more comprehensible description of the ANN action is achieved.
- Their inputs are the similarity between the input vector and the weight vectors of the ANN. Thus, the FRBS only contains 2 rules with m propositions in each antecedent. This implies that the FRBS is very simple. Therefore, it is easy to examine the ANN behavior by studying the FRBS.

Later, the weight vectors are analyzed to inspect the influence of each input variable on the classification performed by the network. Thereby, the knowledge which is learnt by the network can be understood.

The paper is structured as follows. The initial sections introduce several concepts which are necessary to describe our proposal, in particular: ANNs for classification, TSK FRBSs and Uninorm operators. After that, the equivalence between ANNs and TSK FRBSs is shown. Next, a method to extract understandable fuzzy rules from ANNs is presented. Finally, some classification problems are used to illustrate the extraction

method proposed in this paper. Proofs of results presented throughout the paper are found in Appendix A.

2. Artificial neural networks for classification

Multilayer feedforward ANNs for binary classification are the most common and general model of neural nets, hence they are studied in this work. A standard network is illustrated in Fig. 1.

Let us suppose that the net has $n-1$ input values $\mathbf{x}_{\text{initial}} = (x_1, \ldots, x_{n-1})$ and m neurons in its only hidden layer. The output provided by the ANN is:

$$h(\mathbf{x}_{\text{initial}}) = \sum_{j=1}^{m} v_j \cdot \text{Sigm}\left(\sum_{i=1}^{n-1} x_i \cdot w_{ij} + 1 \cdot w_{nj}\right) + b, \tag{1}$$

where w_{nj} is the bias of the hidden neuron j, b is the bias of the output neuron and $\text{Sigm}(x)$ is the Sigmoid activation function defined as

$$\text{Sigm} : \mathbb{R} \to (0,1)$$
$$x \mapsto \text{Sigm}(x) = \frac{1}{1+\mathrm{e}^{-x}}.$$

To carry out the binary classification (output equal to 1 or -1) the sign function is used, which is defined as follows:

$$\text{sgn}(h(\mathbf{x}_{\text{initial}})) = \begin{cases} 1 & \text{if } h(\mathbf{x}_{\text{initial}}) >= 0, \\ -1 & \text{elsewhere.} \end{cases}$$

On the other hand, the initial input vector $\mathbf{x}_{\text{initial}}$ is transformed into $\mathbf{x} = (\mathbf{x}_{\text{initial}}, 1)$ when the bias input is overlapped. By doing so, the former expression $\sum_{i=1}^{n-1} x_i \cdot w_{ij} + 1 \cdot w_{nj}$ in the output of an ANN is

$$\sum_{i=1}^{n} x_i \cdot w_{ij} = \langle \mathbf{x}, \mathbf{w_j} \rangle, \tag{2}$$

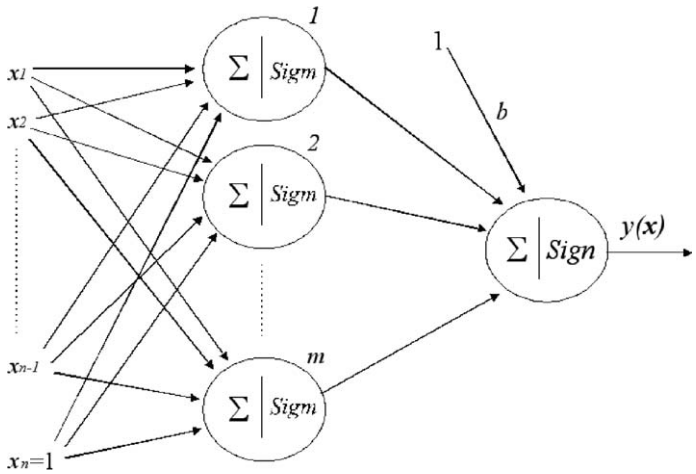


Fig. 1. Multilayer neural network for binary classification.

where $x_n = 1$, $\mathbf{x}$ is the new input vector, $\mathbf{w_j}$ is the weight vector of the hidden neuron j and $\langle,\rangle$ is the typical inner product in $\mathbb{R}^n$. By substituting the Eq. (2) in (1), the ANN output is the following:

$$h(\mathbf{x}) = \sum_{j=1}^{m} v_j \cdot \mathrm{Sigm}\left(\sum_{i=1}^{n} x_i \cdot w_{ij}\right) + b = \sum_{j=1}^{m} v_j \cdot \mathrm{Sigm}(\langle \mathbf{x}, \mathbf{w}_j \rangle) + b. \tag{3}$$

Finally, every input vector $\mathbf{x}^0$ is normalized to have unit length ($\|\mathbf{x}\| = 1$), that is:

$$\mathbf{x} = \left(\frac{x_1^0}{\|\mathbf{x}^0\|}, \ldots, \frac{x_{n-1}^0}{\|\mathbf{x}^0\|}, \frac{x_n^0}{\|\mathbf{x}^0\|}\right).$$

Hence,

$$\langle \mathbf{x}, \mathbf{w_j} \rangle = \|\mathbf{x}\| \cdot \|\mathbf{w_j}\| \cdot \cos(\mathbf{x}, \mathbf{w_j}) = \|\mathbf{w_j}\| \cdot \cos \alpha_j,$$

where α_j is the angle between the input vector $\mathbf{x}$ and the weight vector $\mathbf{w_j}$. The cosine function is a similarity angular measure. For instance, this measure is widely used in areas as important as information retrieval [19].

The normalization applied to the input vectors avoids to collapse two vectors having the same direction but different magnitude, because the bias value equal to 1 is inserted into the vector $\mathbf{x}_{\text{initial}}$ [20].

3. TSK fuzzy rule based systems

Takagi–Sugeno–Kang fuzzy rule based systems (TSK FRBSs) [21] usually have the following structure:

$$R_k : \text{if } X_1 \text{ is } A_1 \ * \ldots * X_n \text{ is } A_n, \text{ then } Y_k = p_0 + \cdots + p_n \cdot X_n,$$

where X_i are the system input variables, A_i are labels with associated fuzzy sets and Y is the output variable. The output Y of a FRBS with m TSK rules is computed as the weighted average of the individual rule outputs Y_i ($i = 1,\ldots,m$) as follows:

$$Y = \frac{\sum_{i=1}^{m} Y_i \cdot g_i}{\sum_{i=1}^{m} g_i},$$

where $g_i = T(A_1(x_1),\ldots,A_n(x_n))$ is the matching degree between the antecedent part of the rule and the current system inputs, T is usually a t-norm and $\mathbf{x} = (x_1,x_2,\ldots,x_n)$ is the system input.

This kind of fuzzy system will be used in the implementation of the FRBS which is proposed in this paper. The fuzzy rules will be implemented by using the operator T like an uninorm [22,23] and the THEN-part only contains the term p_0.

4. Uninorms

Typically, fuzzy sets are combined using t-norms (fuzzy intersection, *and* connective) or t-conorms (fuzzy union, *or* connective) [24,25]. When these operators are used, no compensation between small and large degrees of membership takes place [26–28].

T-norms do not allow low values to be compensated by high values, and t-conorms do not allow high values to be compensated by low values [23]. To describe this fact, we suppose the following example:

"We have evaluated n features of a car (security, comfort, acceleration, ...). We have obtained n values $x_i \in [0,1]$. Each value indicates the quality of a feature (0 is bad quality — 0.5 neuter quality — 1 is good quality). We need to aggregate these values x_i to obtain a global value $y \in [0,1]$ about the car quality."

If we use a t-norm ($x_1 \mathrm{AND} \ldots \mathrm{AND} x_n$) to aggregate the features of the car, only one low value will produce a low final conclusion about the car quality, regardless of the rest of the features were good (high values).

In the same way, if we use a t-conorm ($x_1 \mathrm{OR} \ldots \mathrm{OR} x_n$), a single high value will yield a high final conclusion. Notwithstanding the remainder of the features were bad (low values).

To solve this problem that arises when several degrees of membership are aggregated in some real situations, the *uninorm* operators were defined. Formally, an *uninorm* is a function

$$U : [0,1] \times [0,1] \to [0,1]$$

that has the following properties:

- commutativity,
- monotonicity (increasing),
- associativity and
- an neuter element $e \in [0,1]$.

The most interesting property of the uninorms is its different behavior on particular subdomains (see Fig. 2). The uninorms behave as a t-norm on the interval $([0,e] \times [0,e])$, as a t-conorm on $([e,1] \times [e,1])$ and they have a compensation behavior on $([0,e[\times]e,1] \cup]e,1] \times [0,e[)$. When the uninorm operators are used to aggregate: (A) two small degrees of membership, then they are scaled down, (B) two large degrees, then they are graded

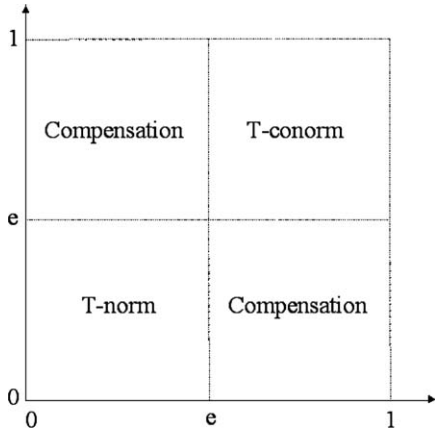


Fig. 2. Behavior of the uninorm operators.

up and (C) finally, some compensation takes place if small and large degrees are aggregated [27].

This behavior is coherent with some real situations. For example, if we aggregate the n values x_i about the car quality with an uninorm, the following reasoning can be carried out:

- The car shortcomings ($x_i \in [0, 0.5[$) are aggregated with a t-norm, obtaining only one low value $y_{\text{shortcomings}}$.
- The advantages of the car ($x_i \in]0.5,1]$) are aggregated with a t-conorm, obtaining only one high value $y_{\text{advantages}}$.
- The neuter features ($x_i = 0.5$) do not influence on the conclusion.
- Finally, a compensation between shortcomings ($y_{\text{shortcomings}} \in [0, 0.5[$) and advantages ($y_{\text{advantages}} \in]0.5, 1]$) of the car is made. In this way, a final value y is obtained.

A particular uninorm is the *symmetric sum* [29] defined as follows:

$$a * b = \frac{a \cdot b}{a \cdot b + (1-a) \cdot (1-b)}.$$

Its domain is the unit square with the exception of the two points (0,1) and (1,0). The neuter element of this operator is 0.5 ($e = 0.5$).

As illustrative example, the *symmetric sum* has the following behavior when combining the values $\alpha,\beta \in]0.0,0.5[$, $\chi,\delta \in]0.5,1.0[$ and $\varepsilon = 0.5$: where:

$$\alpha * \beta * \chi * \delta * \varepsilon = \alpha * \beta * \chi * \delta = \boxed{\begin{array}{c}(\chi\ OR\ \delta)\\ compensation\\ (\alpha\ AND\ \beta)\end{array}} = \boxed{\begin{array}{c}(\chi\ OR\ \delta)\\ \approx\approx\\ (\alpha\ AND\ \beta)\end{array}}$$

- The AND operator acts on]0,0.5[like a t-norm with neuter element equal to 0.5 instead of 1.0.
- The OR operator acts on]0.5,1.0[like a t-conorm with neuter element equal to 0.5 instead of 0.0.
- The $\approx\approx$ operator carries out a *compensation* between a low value

$$\eta = (\alpha \text{ AND } \beta),\ \eta \in]0.0, 0.5[$$

and a high value

$$\varphi = (\chi \text{ OR } \delta),\ \varphi \in]0.5, 1.0[,$$

that is:

- $(\eta \approx\approx \varphi)$ is lower than 0.5 if "$(0.5 - \eta) > (\varphi - 0.5)$".
- $(\eta \approx\approx \varphi)$ is greater than 0.5 if "$(0.5 - \eta) < (\varphi - 0.5)$".
- $(\eta \approx\approx \varphi)$ is equal to 0.5 if "$(0.5 - \eta) = (\varphi - 0.5)$".

The *symmetric sum* operator will be used to aggregate the fuzzy propositions in the antecedents of the rules obtained from ANNs.

5. ANNs for classification are fuzzy rule based systems

Let be a trained ANN to solve a binary classification problem. Let us suppose the weights w_{ij}, v_j and the bias b are constants after the training of the network. We have the following decision function:

$$f(\mathbf{x}) = \operatorname{sgn}(h(\mathbf{x}))$$

where

$$h(\mathbf{x}) = \sum_{j=1}^{m} v_j \cdot \operatorname{Sigm}\left(\sum_{i=1}^{n} x_i \cdot w_{ij}\right) + b = \sum_{j=1}^{m} v_j \cdot \operatorname{Sigm}(\langle \mathbf{x}, \mathbf{w_j} \rangle) + b.$$

Theorem 1. *Let be an ANN with the following decision function*:

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{j=1}^{m} v_j \cdot \operatorname{Sigm}\left(\langle \mathbf{x}, \mathbf{w_j} \rangle\right) + b\right) = \operatorname{sgn}(h(\mathbf{x})).$$

It is equivalent to the following TSK FRBS:

R_1 : **If** $h(\mathbf{x})$ is $I_{(0,\infty)}(x)$, **Then** $Y_1 = 1$
R_2 : **If** $h(\mathbf{x})$ is $I^*_{(0,\infty)}(x)$, **Then** $Y_2 = -1$

where

$$I_{(0,\infty)}(x) = \begin{cases} 1 & \text{if } x \in (0,\infty) \\ 0 & \text{if } x \in (-\infty, 0) \end{cases} \quad \text{and} \quad I^*_{(0,\infty)}(x) = 1 - I_{(0,\infty)}(x).$$

Even thought we have found a FRBS that fires the same output that an ANN, the interpretability has not been improved. To reach it, we will build another FRBS that approximates the system above and it will let us to extract knowledge in an easy way.

Theorem 2. *Let be an ANN with the following decision function*:

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{j=1}^{m} v_j \cdot \operatorname{Sigm}(\langle \mathbf{x}, \mathbf{w_j} \rangle) + b\right) = \operatorname{sgn}(h(\mathbf{x})).$$

It is equivalent to the following TSK FRBS:

R_1 : **If** $h(\mathbf{x})$ is $\operatorname{Sigm}(\lambda \cdot x)$, **Then** $Y_1 = 1$
R_2 : **If** $h(\mathbf{x})$ is $\operatorname{Sigm}^*(\lambda \cdot x)$, **Then** $Y_2 = -1$ *when* $\lambda \to \infty$

where $\operatorname{Sigm}(\lambda \cdot x) = \frac{1}{1+e^{-\lambda \cdot x}}$ *and* $\operatorname{Sigm}^*(\lambda \cdot x) = 1 - \operatorname{Sigm}(\lambda \cdot x) = \operatorname{Sigm}(-\lambda \cdot x)$.

Definition 1. A FRBS which fulfills the features shown in Theorem 2 is called λ-FRBS.

When a λ-FRBS is implemented, the value λ is moderately high. Thereby, the sigmoid function is not quickly saturated to zero or one according to the limited precision of the computer.

We have obtained a λ-FRBS from an ANN with only one proposition in the antecedent of the fuzzy rules. To improve its interpretability, the antecedent of the fuzzy rules will be transformed to get several fuzzy propositions.

5.1. Several fuzzy propositions in the antecedents

In order to obtain several fuzzy propositions in the antecedents of the rules from a λ-FRBS, we need the following result.

Proposition 1

$$\mathrm{Sigm}\left(\beta \cdot \sum_{j=1}^{l} z_j\right) = \mathrm{Sigm}(\beta \cdot z_1) * \ldots * \mathrm{Sigm}(\beta \cdot z_l)$$

where $\beta, z_j \in \mathbb{R}$, $l \in \mathbb{N}$ *and the operator* $*$ *is the symmetric sum uninorm aforementioned in Section 4 [28,29]. Its definition is given by*

$$a_1 * \cdots * a_l = \frac{a_1 \cdots a_l}{a_1 \cdots a_l + (1-a_1)\cdots(1-a_l)}.$$

This result is held for to the function $Sigm^*(x)$ as it is shown below:

$$\begin{aligned}\mathrm{Sigm}^*\left(\beta \cdot \sum_{j=1}^{l} z_j\right) &= \mathrm{Sigm}\left(-\beta \cdot \sum_{j=1}^{l} z_j\right) = \mathrm{Sigm}(-\beta \cdot z_1) * \cdots * \mathrm{Sigm}(-\beta \cdot z_l) \\ &= \mathrm{Sigm}^*(\beta \cdot z_1) * \cdots * \mathrm{Sigm}^*(\beta \cdot z_l).\end{aligned}$$

Taking into account the nature of the output of the ANN $h(\mathbf{x})$ (see Eq. (3)), we have:

$$\begin{aligned}&R_1: \textbf{ If } \left(\sum_{j=1}^{m} v_j \cdot \mathrm{Sigm}(\langle \mathbf{x}, \mathbf{w_j} \rangle) + b\right) \text{ is } \mathrm{Sigm}(\lambda \cdot x), \textbf{ Then } Y_1 = 1 \\ &R_2: \textbf{ If } \left(\sum_{j=1}^{m} v_j \cdot \mathrm{Sigm}(\langle \mathbf{x}, \mathbf{w_j} \rangle) + b\right) \text{ is } \mathrm{Sigm}^*(\lambda \cdot x), \textbf{ Then } Y_2 = -1\end{aligned}$$

According to the former Proposition, the λ-FRBS above can be transformed into a modified λ-FRBS with several fuzzy propositions, as follows:

$$\begin{aligned}R_1: \textbf{ If } & v_1 \cdot \mathrm{Sigm}(\langle \mathbf{x}, \mathbf{w_1} \rangle) \text{ is } \mathrm{Sigm}(\lambda \cdot x) * \ldots \\ & v_m \cdot \mathrm{Sigm}(\langle \mathbf{x}, \mathbf{w_m} \rangle) \text{ is } \mathrm{Sigm}(\lambda \cdot x) * \\ & b \text{ is } \mathrm{Sigm}(\lambda \cdot x), \textbf{ Then } Y_1 = 1 \\ R_2: \textbf{ If } & v_1 \cdot \mathrm{Sigm}(\langle \mathbf{x}, \mathbf{w_1} \rangle) \text{ is } \mathrm{Sigm}^*(\lambda \cdot x) * \ldots \\ & v_m \cdot \mathrm{Sigm}(\langle \mathbf{x}, \mathbf{w_m} \rangle) \text{ is } \mathrm{Sigm}^*(\lambda \cdot x) * \\ & b \text{ is } \mathrm{Sigm}^*(\lambda \cdot x), \textbf{ Then } Y_2 = -1\end{aligned}$$

which is equivalent to:

$$
\begin{aligned}
R_1:\ &\textbf{If } \langle \mathbf{x}, \mathbf{w_1} \rangle \text{ is } \mathrm{Sigm}(\lambda \cdot v_1 \cdot \mathrm{Sigm}(x)) * \ldots \\
&\quad \langle \mathbf{x}, \mathbf{w_m} \rangle \text{ is } \mathrm{Sigm}(\lambda \cdot v_m \cdot \mathrm{Sigm}(x)) * \\
&\quad b \text{ is } \mathrm{Sigm}(\lambda \cdot x), \textbf{ Then } Y_1 = 1 \\
R_2:\ &\textbf{If } \langle \mathbf{x}, \mathbf{w_1} \rangle \text{ is } \mathrm{Sigm}^*(\lambda \cdot v_1 \cdot \mathrm{Sigm}(x)) * \ldots \\
&\quad \langle \mathbf{x}, \mathbf{w_m} \rangle \text{ is } \mathrm{Sigm}^*(\lambda \cdot v_m \cdot \mathrm{Sigm}(x)) * \\
&\quad b \text{ is } \mathrm{Sigm}^*(\lambda \cdot x), \textbf{ Then } Y_2 = -1
\end{aligned}
$$

Since the vector $\mathbf{x}$ is normalized to have unit lenght, we have the following expression:

$$\langle \mathbf{x}, \mathbf{w_j} \rangle = \|\mathbf{w_j}\| \cdot \cos \alpha_j = \|\mathbf{w_j}\| \cdot \mathrm{sim}(\mathbf{x}, \mathbf{w_j}).$$

As we mentioned before, the *cosine* function is used as a similarity measure in several application areas, i.e. information retrieval. From now on, the *cosine* function will be denoted as sim. This is to notice the fact that the *cosine* function is a similarity measure. The interpretation of its output is the following:

- When its output is close to -1, then the vectors compared have practically opposite directions. Hence, both vectors hardly bear resemblance.
- When its output is close to 1, then the inspected vectors have nearly the same direction. As a consequence, we consider they are just about alike.

In this way, we obtain the following FRBS:

$$
\begin{aligned}
R_1:\ &\textbf{If } \|\mathbf{w_1}\|\mathrm{sim}(\mathbf{x}, \mathbf{w_1}) \text{ is } \mathrm{Sigm}(\lambda \cdot v_1 \cdot \mathrm{Sigm}(x)) * \ldots \\
&\quad \|\mathbf{w_m}\|\mathrm{sim}(\mathbf{x}, \mathbf{w_m}) \text{ is } \mathrm{Sigm}(\lambda \cdot v_m \cdot \mathrm{Sigm}(x)) * \\
&\quad b \text{ is } \mathrm{Sigm}(\lambda \cdot x), \textbf{ Then } Y_1 = 1 \\
R_2:\ &\textbf{If } \|\mathbf{w_1}\|\mathrm{sim}(\mathbf{x}, \mathbf{w_1}) \text{ is } \mathrm{Sigm}^*(\lambda \cdot v_1 \cdot \mathrm{Sigm}(x)) * \ldots \\
&\quad \|\mathbf{w_m}\|\mathrm{sim}(\mathbf{x}, \mathbf{w_m}) \text{ is } \mathrm{Sigm}^*(\lambda \cdot v_m \cdot \mathrm{Sigm}(x)) * \\
&\quad b \text{ is } \mathrm{Sigm}^*(\lambda \cdot x), \textbf{ Then } Y_2 = -1
\end{aligned}
$$

which is equivalent to the next one:

$$
\begin{aligned}
R_1:\ &\textbf{If } \mathrm{sim}(\mathbf{x}, \mathbf{w_1}) \text{ is } \mathrm{Sigm}(\lambda \cdot v_1 \cdot \mathrm{Sigm}(\|\mathbf{w_1}\| \cdot x)) * \ldots \\
&\quad \mathrm{sim}(\mathbf{x}, \mathbf{w_m}) \text{ is } \mathrm{Sigm}(\lambda \cdot v_m \cdot \mathrm{Sigm}(\|\mathbf{w_m}\| \cdot x)) * \\
&\quad b \text{ is } \mathrm{Sigm}(\lambda \cdot x), \textbf{ Then } Y_1 = 1 \\
R_2:\ &\textbf{If } \mathrm{sim}(\mathbf{x}, \mathbf{w_1}) \text{ is } \mathrm{Sigm}^*(\lambda \cdot v_1 \cdot \mathrm{Sigm}(\|\mathbf{w_1}\| \cdot x)) * \ldots \\
&\quad \mathrm{sim}(\mathbf{x}, \mathbf{w_m}) \text{ is } \mathrm{Sigm}^*(\lambda \cdot v_m \cdot \mathrm{Sigm}(\|\mathbf{w_m}\| \cdot x)) * \\
&\quad b \text{ is } \mathrm{Sigm}^*(\lambda \cdot x), \textbf{Then } Y_2 = -1
\end{aligned}
$$

Furthermore, if we consider the following properties:

- $\mathrm{Sigm}(\lambda \cdot v_j \cdot \mathrm{Sigm}(\|\mathbf{w_j}\| \cdot x)) \in]0.5,1[$ when $v_j > 0$,
- $\mathrm{Sigm}^*(\lambda \cdot v_j \cdot \mathrm{Sigm}(\|\mathbf{w_j}\| \cdot x)) \in]0,0.5[$ when $v_j > 0$,
- $\mathrm{Sigm}(\lambda \cdot v_j \cdot \mathrm{Sigm}(\|\mathbf{w_j}\| \cdot x)) \in]0,0.5[$ when $v_j < 0$,
- $\mathrm{Sigm}^*(\lambda \cdot v_j \cdot \mathrm{Sigm}(\|\mathbf{w_j}\| \cdot x)) \in]0.5,1[$ when $v_j < 0$,

and we suppose, without loss of generality, that:

- $v_j > 0$ for $k = 1,\ldots,p$,
- $v_j < 0$ for $k = (p+1),\ldots,m$,
- $b < 0$,

then, we obtain the following λ-FRBS which takes into account the behavior of the *symmetric sum* operator (see Section 4):

R_1 : **If** $\mathrm{sim}(\mathbf{x},\mathbf{w_1})$ is $\mathrm{Sigm}(\lambda \cdot v_1 \cdot \mathrm{Sigm}(\|\mathbf{w_1}\| \cdot x))$ OR ...
$\mathrm{sim}(\mathbf{x},\mathbf{w_p})$ is $\mathrm{Sigm}(\lambda \cdot v_p \cdot \mathrm{Sigm}(\|\mathbf{w_p}\| \cdot x))$
$\approx\approx$
$\mathrm{sim}(\mathbf{x},\mathbf{w_{p+1}})$ is $\mathrm{Sigm}(\lambda \cdot v_{p+1} \cdot \mathrm{Sigm}(\|\mathbf{w_{p+1}}\| \cdot x))$ AND ...
$\mathrm{sim}(\mathbf{x},\mathbf{w_m})$ is $\mathrm{Sigm}(\lambda \cdot v_m \cdot \mathrm{Sigm}(\|\mathbf{w_m}\| \cdot x))$ AND
b is $\mathrm{Sigm}(\lambda \cdot x)$, **Then** $Y_1 = 1$

R_2 : **If** $\mathrm{sim}(\mathbf{x},\mathbf{w_1})$ is $\mathrm{Sigm}^*(\lambda \cdot v_1 \cdot \mathrm{Sigm}(\|\mathbf{w_1}\| \cdot x))$ OR ...
$\mathrm{sim}(\mathbf{x},\mathbf{w_p})$ is $\mathrm{Sigm}^*(\lambda \cdot v_p \cdot \mathrm{Sigm}(\|\mathbf{w_p}\| \cdot x))$
$\approx\approx$
$\mathrm{sim}(\mathbf{x},\mathbf{w_{p+1}})$ is $\mathrm{Sigm}^*(\lambda \cdot v_{p+1} \cdot \mathrm{Sigm}(\|\mathbf{w_{p+1}}\| \cdot x))$ AND ...
$\mathrm{sim}(\mathbf{x},\mathbf{w_m})$ is $\mathrm{Sigm}^*(\lambda \cdot v_m \cdot \mathrm{Sigm}(\|\mathbf{w_m}\| \cdot x))$ AND
b is $\mathrm{Sigm}^*(\lambda \cdot x)$, **Then** $Y_2 = -1$

5.2. Linguistic interpretation of the fuzzy propositions

Next, we give a linguistic interpretation to the fuzzy propositions included in the λ-FRBS which is extracted from the corresponding ANN. In these λ-FRBSs, we find the following fuzzy propositions:

(1) "*b is* $\mathrm{Sigm}(\lambda \cdot x)$" can be interpreted as "*b is approximately larger than* 0" (see Fig. 3 with $\lambda = 50$).
(2) "*b is* $\mathrm{Sigm}^*(\lambda \cdot x)$" can be interpreted as "*b is not approximately larger than* 0" as it is shown:
$$\mathrm{Sigm}^*(\lambda \cdot x) = 1 - \mathrm{Sigm}(\lambda \cdot x) \equiv \mathrm{Not}(\mathrm{Sigm}(\lambda \cdot x)). \quad (4)$$
The last expression is equivalent to "*b is approximately smaller than* 0" (see Fig. 4 with $\lambda = 50$).

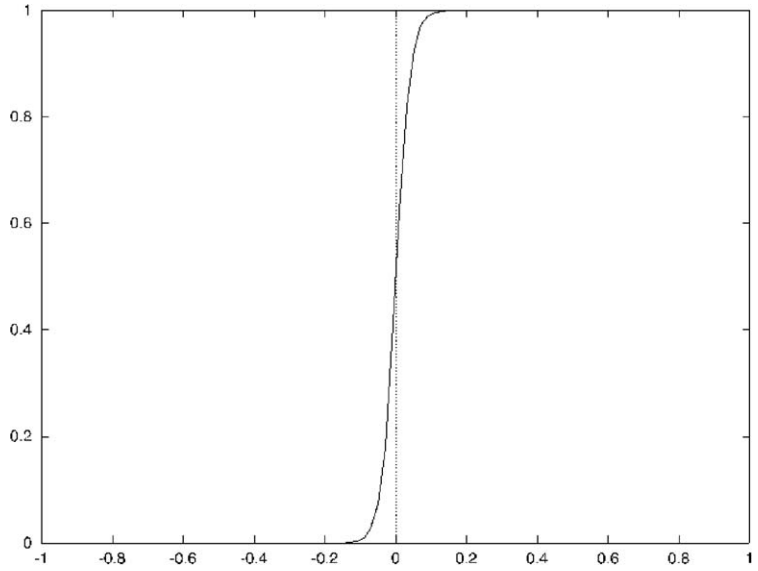


Fig. 3. b is approximately larger than 0.

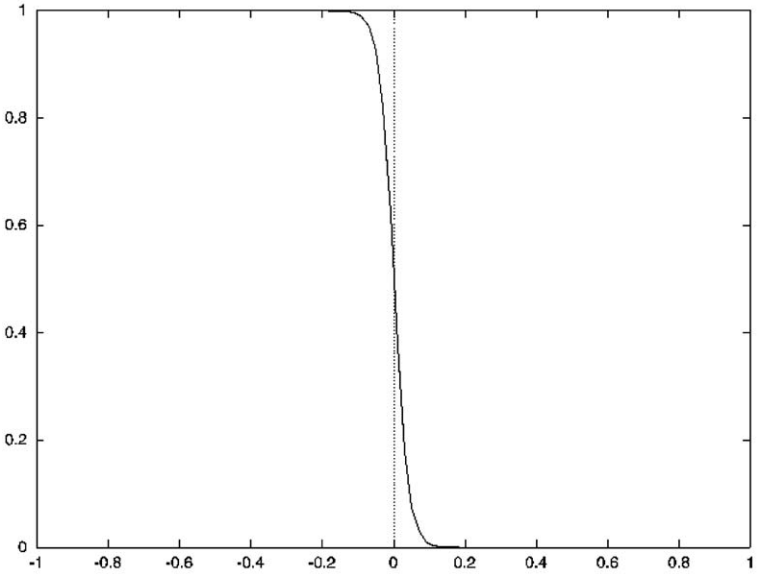


Fig. 4. b is approximately smaller than 0.

(3) "sim($\mathbf{x}$,$\mathbf{w_j}$) *is* Sigm($\lambda \cdot v_j \cdot$ Sigm($\|\mathbf{w_j}\| \cdot x$))" with $v_j > 0$ can be interpreted as "sim($\mathbf{x}$,$\mathbf{w_j}$) *is approximately larger*$_{\mathrm{OR}}$ *than* β" (see Fig. 5 with $\lambda = 50$, $v_j = 1$ and $\|\mathbf{w_j}\|$) where

$$\beta = \frac{\mathrm{Sigm}^{-1}\left(\frac{\mathrm{Sigm}^{-1}(0.75)}{\lambda \cdot |v_j|}\right)}{\|\mathbf{w_j}\|}$$

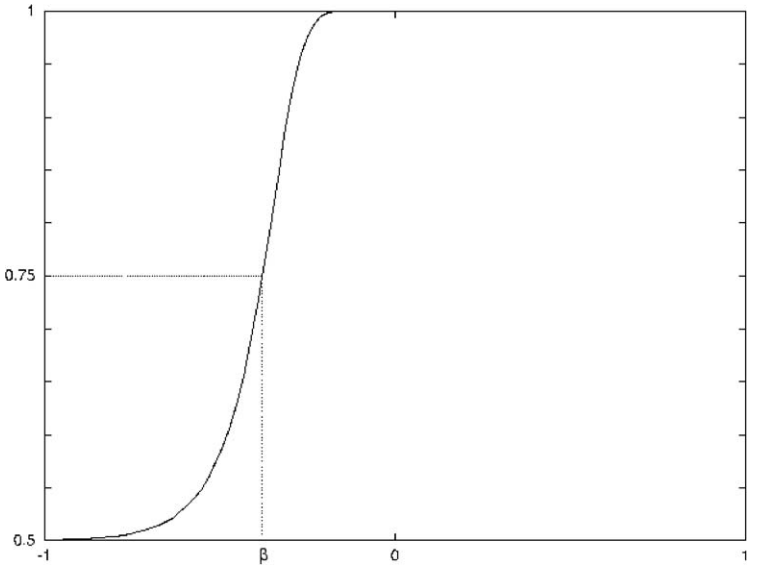


Fig. 5. $\mathrm{sim}(\mathbf{x}, \mathbf{w_j})$ is approximately $\mathrm{larger}_{\mathrm{OR}}$ than β.

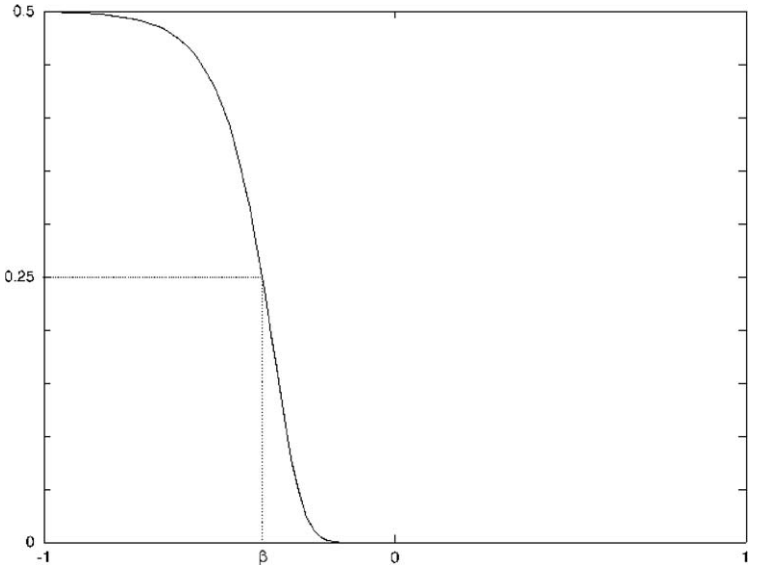


Fig. 6. $\mathrm{sim}(\mathbf{x}, \mathbf{w_j})$ is approximately $\mathrm{smaller}_{\mathrm{AND}}$ than β.

and

$$\mathrm{Sigm}^{-1}(x) = -\ln\left(\frac{1-x}{x}\right).$$

We can use the word **larger**$_{\mathrm{OR}}$ in the last expression, because it has only a valid meaning in the aggregation of fuzzy propositions with the t-conorm provided by the symmetric sum.

(4) "sim($\mathbf{x}$,$\mathbf{w_j}$) *is* Sigm($\lambda \cdot v_j \cdot$ Sigm($\|\mathbf{w_j}\| \cdot x$))" with $v_j < 0$ can be interpreted as "sim($\mathbf{x}$,$\mathbf{w_j}$) *is not approximately larger*$_{\mathrm{OR}}$ *than* β" since

$$\begin{aligned}\mathrm{Sigm}(\lambda \cdot v_j \cdot \mathrm{Sigm}(\|\mathbf{w_j}\| \cdot x)) &= 1 - \mathrm{Sigm}(\lambda \cdot |v_j| \cdot \mathrm{Sigm}(\|\mathbf{w_j}\| \cdot x)) \\ &\equiv \mathrm{Not}(\mathrm{Sigm}(\lambda \cdot |v_j| \cdot \mathrm{Sigm}(\|\mathbf{w_j}\| \cdot x))).\end{aligned}$$

The former expression is equivalent to "sim($\mathbf{x}$,$\mathbf{w_j}$) *is approximately smaller*$_{\mathrm{AND}}$ *than* β" (see Fig. 6 with $\lambda = 50$, $v_j = -1$ and $\|\mathbf{w_j}\| = 10$).

As happened with the linguistic term larger$_{\mathrm{OR}}$, the word **smaller**$_{\mathrm{AND}}$ is used in this expression, since it has only a valid meaning if we aggregate the fuzzy propositions with the t-norm given by the symmetric sum operator.

(5) Finally, "sim($\mathbf{x}$,$\mathbf{w_j}$) *is* Sigm*($\lambda \cdot v_j \cdot$ Sigm($\|\mathbf{w_j}\| \cdot x$))" is equivalent to "sim($\mathbf{x}$,$\mathbf{w_j}$) *is not Sigm*($\lambda \cdot v_j \cdot$ Sigm($\|\mathbf{w_j}\| \cdot x$))" by using (4) and it has been interpreted in the above paragraphs.

6. Examples

Next, we consider two binary classification problems to show the design of a λ-FRBS from an ANN. The two problems are (A) the XOR problem and (B) the PIMA diabetes problem.

6.1. The XOR problem

This problem [30] has the following training data:

$$\{(\mathbf{x^1} = (-1,-1), y^1 = -1), (\mathbf{x^2} = (-1,1), y^2 = 1), (\mathbf{x^3} = (1,-1), y^3 = 1), (\mathbf{x^4} = (1,1), y^4 = -1)\}.$$

These input vectors joined to the bias are normalized to have unit length. Thus, the training data are transformed into the following ones:

$$\left\{\left(\mathbf{x^1} = \left(\frac{-1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right), y^1 = -1\right), \left(\mathbf{x^2} = \left(\frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right), y^2 = 1\right), \left(\mathbf{x^3} = \left(\frac{1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right), y^3 = 1\right), \left(\mathbf{x^4} = \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right), y^4 = -1\right)\right\}.$$

A multilayer feedforward ANN with one hidden layer composed by two neurons that solves the XOR problem is figured out in the Fig. 7. The weight vectors of this network are:

i →	1 (x_1)	2 (x_2)	3 (x_3)
w_{i1} →	$24\sqrt{3}$	$24\sqrt{3}$	$-8\sqrt{3}$
w_{i2} →	$24\sqrt{3}$	$24\sqrt{3}$	$8\sqrt{3}$

It is important to note that these weight vectors are similar. They have only the third component different ($w_{31} \neq w_{32}$). However, it is the least important component when the scalar products with the input vectors are computed. This component is only important when the inputs x_1 and x_2 are very small or when they fulfill that ($x_1 = -x_2$). These

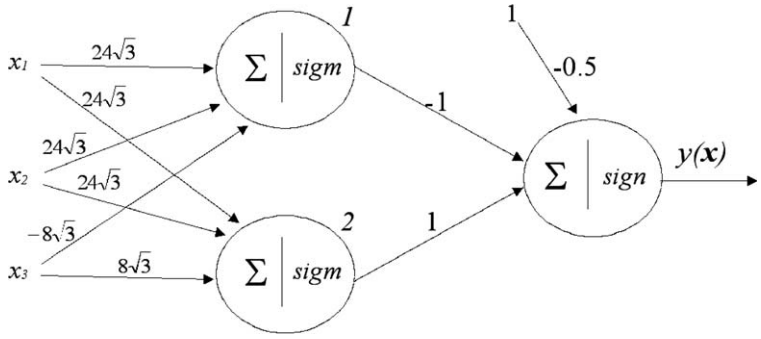


Fig. 7. The ANN trained to solve the XOR problem.

facts will be useful to understand the obtained λ-FRBS from the ANN that solves the problem considered.

The corresponding λ-FRBS with $\lambda = 50$, extracted from the ANN displayed in Fig. 7, is the following:

$$
\begin{aligned}
R_1:\ & \textbf{If } \mathrm{sim}(\mathbf{x},\mathbf{w_2}) \text{ is } \mathrm{Sigm}(50\cdot \mathrm{Sigm}(60.4\cdot x)\\
& \qquad \approx\approx\\
& \quad \mathrm{sim}(\mathbf{x},\mathbf{w_1}) \text{ is } \mathrm{Sigm}(-50\cdot \mathrm{Sigm}(60.4\cdot x)) \text{ AND}\\
& \quad -0.5 \text{ is } \mathrm{Sigm}(50\cdot x),\ \textbf{Then } Y_1 = 1\\
R_2:\ & \textbf{If } \mathrm{sim}(\mathbf{x},\mathbf{w_1}) \text{ is } \mathrm{Sigm}^*(-50\cdot \mathrm{Sigm}(60.4\cdot x)) \text{ AND}\\
& \quad -0.5 \text{ is } \mathrm{Sigm}^*(50\cdot x)\\
& \qquad \approx\approx\\
& \quad \mathrm{sim}(\mathbf{x},\mathbf{w_2}) \text{ is } \mathrm{Sigm}^*(50\cdot \mathrm{Sigm}(60.4\cdot x),\ \textbf{Then } Y_2 = -1
\end{aligned}
$$

This can be linguistically translated as:

$$
\begin{aligned}
R_1:\ & \textbf{If } \mathrm{sim}(\mathbf{x},\mathbf{w_2}) \text{ is aproximately larger}_{\mathrm{OR}} \text{ than } -0.06\\
& \qquad \approx\approx\\
& \quad \mathrm{sim}(\mathbf{x},\mathbf{w_1}) \text{ is approximately smaller}_{\mathrm{AND}} \text{ than } -0.06 \text{ AND}\\
& \quad -0.5 \text{ is approximately larger than } 0, \textbf{Then } Y_1 = 1\\
R_2:\ & \textbf{If } \mathrm{sim}(\mathbf{x},\mathbf{w_1}) \text{ is approximately larger}_{\mathrm{OR}} \text{ than } -0.06 \text{ OR}\\
& \quad -0.5 \text{ is approximately smaller than } 0\\
& \qquad \approx\approx\\
& \quad \mathrm{sim}(\mathbf{x},\mathbf{w_2}) \text{ is approximately smaller}_{\mathrm{AND}} \text{ than } -0.06,\ \textbf{Then } Y_2 = -1
\end{aligned}
$$

From an analysis of the former λ-FRBS, we can note the following conclusions:

- The fuzzy proposition "-0.5 *is approximately smaller than* 0" is nearly true in rule R_2. Thereby, the output of the λ-FRBS is equal to -1 except when the input datum is similar to $\mathbf{w_2}$. In this case, the fuzzy proposition "$\mathrm{sim}(\mathbf{x},\mathbf{w_2})$ *is approximately smaller*$_{\mathrm{AND}}$

than -0.06'' is almost false in rule R_2. Thus, this expression is compensated with the fuzzy proposition obtained from the bias weight.

However, it is not enough to change the output value of rule R_2 when the input sample is also similar to the weight vector $\mathbf{w_1}$. In this case, the following proposition wins the compensation

$$\begin{array}{l}\mathrm{sim}(\mathbf{x},\mathbf{w_1}) \text{ is approximately larger}_{\mathrm{OR}} \text{ than } -0.06\\ \quad \mathrm{OR}\\ -0.5 \text{ is approximately smaller than } 0\end{array}$$

to the opposite fuzzy proposition of $\mathbf{w_2}$ in the rule R_2:

$$\mathrm{sim}(\mathbf{x},\mathbf{w_2}) \text{ is approximately smaller}_{\mathrm{AND}} \text{ than } -0.06.$$

Therefore, the output value of the λ-FRBS is equal to 1 when the input datum $\mathbf{x}$ accomplishes the following:

$$\mathbf{x} \text{ is similar to } \mathbf{w_2} \text{ and } \mathbf{x} \text{ is not similar to } \mathbf{w_1}. \tag{5}$$

- Since the weight vectors $\mathbf{w_1}$ and $\mathbf{w_2}$ are similar with the only exception of the third component, we need that $(x_1 = -x_2)$ to fulfill the condition established in (5). Thus, the two first components of the weight vectors are cancelled and the similarity is measured using only the third component.

After analyzing the obtained λ-FRBS, we have extracted the following knowledge:

$$\begin{array}{l}\text{The output of the network is equal to } -1,\\ \text{except when ``}x_1 \text{ is approximately equal to } -x_2\text{''}.\end{array}$$

This conclusion is coherent with a feasible solution of the XOR problem.

6.2. The PIMA diabetes problem

This problem [31] consists of discerning whether a person is diabetic or not. The data set includes 768 data. They are divided into two sets of 576 and 192 elements that will be used as training and test folds, respectively. Each datum is composed by eight continuous variables, normalized in the unit interval, that is $\mathbf{x}_i \in [0,1]$ $(i = 1,\ldots,8)$ and one output with two possible values ($y = -1 \rightarrow$ No diabetes; $y = 1 \rightarrow$ Diabetes). The input variables are:

- $\mathbf{x_1}$: Number of pregnancies.
- $\mathbf{x_2}$: Plasma glucose concentration a two hours in an oral glucose tolerance test.
- $\mathbf{x_3}$: Diastolic blood pressure (mm Hg).
- $\mathbf{x_4}$: Triceps skin fold thickness (mm).
- $\mathbf{x_5}$: Serum insulin (vU/ml).
- $\mathbf{x_6}$: Body mass index (weight in kg/(height in m)^2).
- $\mathbf{x_7}$: Diabetes pedigree function.
- $\mathbf{x_8}$: Age in years.

A multilayer feedforward ANN with one hidden layer composed by two neurons has been trained with the Backpropagation algorithm [32] to solve the problem considered. The input vectors have been previously transformed to have unit length. The success of the trained network is 77.60% and 80.21% on the training and test data set, respectively. It is figured out in Fig. 8.

The λ-FRBS extracted from this ANN with $\lambda = 50$ is:

$$
\begin{aligned}
R_1:\ & \textbf{If } \mathrm{sim}(\mathbf{x},\mathbf{w_1}) \text{ is } \mathrm{Sigm}(98.75 \cdot \mathrm{Sigm}(22.3 \cdot x)\\
& \approx\approx\\
& \mathrm{sim}(\mathbf{x},\mathbf{w_2}) \text{ is } \mathrm{Sigm}(-95 \cdot \mathrm{Sigm}(24.48 \cdot x)) \text{ AND}\\
& -0.5 \text{ is } \mathrm{Sigm}(50 \cdot x), \textbf{ Then } Y_1 = 1\\
R_2:\ & \textbf{If } \mathrm{sim}(\mathbf{x},\mathbf{w_2}) \text{ is } \mathrm{Sigm}^*(-95 \cdot \mathrm{Sigm}(24.48 \cdot x)) \text{ AND}\\
& -0.5 \text{ is } \mathrm{Sigm}^*(50 \cdot x)\\
& \approx\approx\\
& \mathrm{sim}(\mathbf{x},\mathbf{w_1}) \text{ is } \mathrm{Sigm}^*(98.75 \cdot \mathrm{Sigm}(22.3 \cdot x), \textbf{ Then } Y_2 = -1
\end{aligned}
$$

It can be linguistically interpreted as:

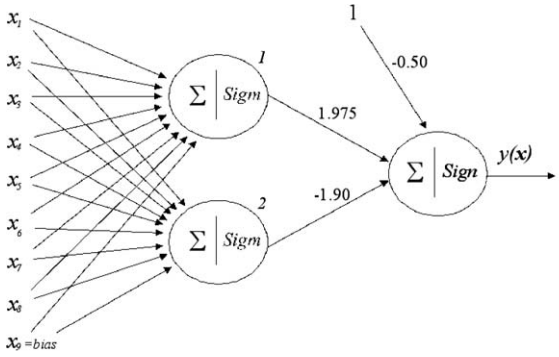
$$
\begin{aligned}
R_1:\ & \textbf{If } \mathrm{sim}(\mathbf{x},\mathbf{w_1}) \text{ is aproximately larger}_{\mathrm{OR}} \text{ than } -0.2\\
& \approx\approx\\
& \mathrm{sim}(\mathbf{x},\mathbf{w_2}) \text{ is approximately smaller}_{\mathrm{AND}} \text{ than } -0.18 \text{ AND}\\
& -0.5 \text{ is approximately larger than0, } \textbf{Then } Y_1 = 1\\
R_2:\ & \textbf{If } \mathrm{sim}(\mathbf{x},\mathbf{w_2}) \text{ is approximately larger}_{\mathrm{OR}} \text{ than } -0.18 \text{ OR}\\
& -0.5 \text{ is approximately smaller than 0}\\
& \approx\approx\\
& \mathrm{sim}(\mathbf{x},\mathbf{w_1}) \text{ is approximately smaller}_{\mathrm{AND}} \text{ than } -0.2, \textbf{ Then } Y_2 = -1
\end{aligned}
$$

and the weight vectors $\mathbf{w_1}$ and $\mathbf{w_2}$ are the followings:

i →	1 (x_1)	2 (x_2)	3 (x_3)	4 (x_4)	5 (x_5)	6 (x_6)	7 (x_7)	8 (x_8)	9 (Bias)
w_{i1} →	3.3	12.8	−1.85	−0.27	0.75	9.3	5.8	−2.3	−13.7
w_{i2} →	3.4	−0.84	3.7	−2.6	6.0	−1.8	−0.9	−22.62	3.6

If an analysis of the former λ-FRBS and their weight vectors is made, we can claim the following:

- The fuzzy proposition "-0.5 *is approximately smaller than* 0" is almost true in rule R_2. This fact establishes a default value equal to -1 (No diabetes) in the λ-FRBS.
- In the same rule, the fuzzy proposition "$\mathrm{sim}(\mathbf{x},\mathbf{w_2})$ *is approximately larger*$_{\mathrm{OR}}$ *than* -0.18" determines that when the input vector $\mathbf{x}$ is similar to the weight vector $\mathbf{w_2}$, the output ($y = -1$) (No diabetes) is fired. Hence, we can conclude that the weight vector $\mathbf{w_2}$ is the prototype to the No diabetes samples. If we observe the features of the weight vector $\mathbf{w_2}$, we can see that the values with high magnitude and different from the ones of $\mathbf{w_1}$



$\mathbf{w}_{ij}$	1	2
1	3.3	3.4
2	12.8	-0.84
3	-1.85	3.7
4	-0.27	-2.6
5	0.75	6.0
6	9.3	-1.8
7	5.8	-0.9
8	-2.3	-22.62
9	-13.7	3.6

Fig. 8. The ANN trained to solve the PIMA problem.

(prototype to the Diabetes samples) are $w_{52} = 6$ and $w_{82} = -22.6$. As the inputs x_i are positive, we can claim that the prototype defined by the weight vector $\mathbf{w_2}$ is essentially characterized by high values of x_5 (High Serum Insuline) and low values of x_8 (Low Age).

- In rule R_1, the fuzzy proposition "$sim(\mathbf{x},\mathbf{w_1})$ *is approximately larger*$_{\mathrm{OR}}$ *than* -0.2" determines that when the input vector $\mathbf{x}$ is similar to the weight vector $\mathbf{x_1}$, the output ($y = 1$) (Diabetes) is fired. Thus, we can deduce that the weight vector $\mathbf{w_1}$ is the prototype to the Diabetes data.

  Let us see the features of the weight vector $\mathbf{w_1}$, we can see that the values with high magnitude and different from the ones of $\mathbf{w_2}$ (the prototype to the No diabetes data) are $w_{21} = 12.8$, $w_{61} = 9.3$ and $w_{71} = 5.8$. As the inputs x_i are positive, we can conclude that the prototype defined from the weight vector $\mathbf{w_1}$ is mainly characterized to the high values of x_2 (High Glucose Concentration), x_6 (High Body Mass Index) and x_7 (High Diabetes Pedigree).

The knowledge discerned from the λ-FRBS is intuitively correct.

7. Comparison with other extraction methods

Finally, we compare the λ-FRBS extracted by using the proposed method against the FRBSs obtained when the approaches presented in [6,14] are considered. The comparison is made by means of a subset of standard criteria proposed in [16,33]. Next, we explain them:

- The *expressive power* of the extracted rules (types of rules obtained).
- The *fidelity* which describes how well the rules can mimic the behavior of the ANN.
- Lastly, the *comprehensibility* that is determined by measuring the number of rules and the number of antecedents per rule.

Considering the *expressive power*, the presented method and the one proposed in [14] extract TSK rules with constant output. On the other hand, we obtain additive fuzzy rules by using [6]. If we focus on the *fidelity*, the three approaches yield a FRBS with the same behavior as the corresponding ANN used to extract the rules. Finally, with respect to the *comprehensibility*, the method presented in [14] extracts 2^m fuzzy rules with m propositions

per rule antecedent (where m is the number of neurons in the ANN hidden layer) and the one proposed in [6] obtains m rules with n fuzzy propositions (where n is the number of input attributes) within each fuzzy rule antecedent. Furthermore, the proposed method yields a FRBS with only *2* rules with m fuzzy propositions in the antecedent per rule.

8. Conclusions

A method to extract FRBSs from trained ANNs has been proposed. It carries out the same classification as the original ANN.

To classify new data, the fuzzy rules of the system evaluate the similarity between the input sample and the weight vectors. As a result, they are highly understandable.

Once the FRBS has been built, a simple analysis of the weight vectors is necessary to understand the action of the network. Thus, classification problem knowledge can be discovered using ANNs. This method of knowledge discovery has been illustrated by means of classification problems. The knowledge extracted from the trained ANNs to solve the former problems is coherent with the standard knowledge avalaible about the problem considered.

Appendix A. Proof of results

A.1. Proof (Theorem 1)

Let be $\mathbf{x_0}$ a vector belonging to the input feature space. It is tested by using the λ-FRBS proposed above:

- If $h(\mathbf{x_0}) \in (-\infty,0)$, then the output fired by the λ-FRBS is given by

$$Y = \frac{\sum_{i=1}^{m} Y_i \cdot g_i}{\sum_{i=1}^{m} g_i} = \frac{Y_1 \cdot I_{(0,\infty)}(h(\mathbf{x_0})) + Y_2 \cdot I^*_{(0,\infty)}(h(\mathbf{x_0}))}{I_{(0,\infty)}(h(\mathbf{x_0})) + I^*_{(0,\infty)}(h(\mathbf{x_0}))} = \frac{Y_1 \cdot 0 + Y_2 \cdot 1}{0+1} = Y_2 = -1,$$

  which is equal to the output yielded by $f(\mathbf{x_0})$, since $f(\mathbf{x_0}) = \text{sign}(h(\mathbf{x_0})) = -1$.

- If $h(\mathbf{x_0}) \in (0,\infty)$, then the output fired by the λ-FRBS is as follows:

$$Y = \frac{\sum_{i=1}^{m} Y_i \cdot g_i}{\sum_{i=1}^{m} g_i} = \frac{Y_1 \cdot I_{(0,\infty)}(h(\mathbf{x_0})) + Y_2 \cdot I^*_{(0,\infty)}(h(\mathbf{x_0}))}{I_{(0,\infty)}(h(\mathbf{x_0})) + I^*_{(0,\infty)}(h(\mathbf{x_0}))} = \frac{Y_1 \cdot 1 + Y_2 \cdot 0}{1+0} = Y_1 = 1,$$

  so it is equal to the output provided by $f(\mathbf{x_0}) = \text{sign}(h(\mathbf{x_0})) = 1$. □

A.2. Proof (Theorem 2)

If we consider that the following expression is equivalent to $I_{(0,\infty)}(x)$

$$\lim_{\lambda\to\infty}(\text{Sigm}(\lambda \cdot x)) = \lim\left(\frac{1}{1+e^{-\lambda \cdot x}}\right) = \begin{cases} 1 & x \in (0,\infty) \\ 0 & x \in (-\infty,0), \end{cases}$$

and the next one to $I^{*}_{(0,\infty)}(x)$,

$$\lim_{\lambda\to\infty}(\mathrm{Sigm}^{*}(\lambda\cdot x)) = \lim(1-\mathrm{Sigm}(\lambda\cdot x)) = \begin{cases} 0 & x\in(0,\infty) \\ 1 & x\in(-\infty,0). \end{cases}$$

Then, we have that the following FRBS is equivalent to the FRBS presented in Theorem 1 when $\lambda\to\infty$.

R_1 : **If** $h(\mathbf{x})$ is $\mathrm{Sigm}(\lambda\cdot x)$, **Then** $Y_1 = 1$

R_2 : **If** $h(\mathbf{x})$ is $\mathrm{Sigm}^{*}(\lambda\cdot x)$, **Then** $Y_2 = -1$

In this way, as the Theorem 1 FRBS is equivalent to the decision function $f(\mathbf{x})$, the FRBS presented in this Theorem is also equivalent to $f(\mathbf{x})$. □

A.3. Proof (Proposition 1)

See the proof of results included in [6]. □

References

[1] R. Lippmann, An introduction to computing with neural nets, IEEE ASP Magazine April (1987) 4–22.
[2] P. Wasserman, Advanced methods in neural computing, Van Nostrand Reinhold, 115, Fifth Avenue, New York, NY, 10003, 1993.
[3] S. Haykin, Neural Networks: A Comprenhensive Foundation, McMillan College Publishing Company, New York, 1994.
[4] J.-S. Jang, C.-T. Sun, Functional equivalence between radial basis function networks and fuzzy inference systems, IEEE Transactions on Neural Networks 4 (1999) 156–158.
[5] L. Fu, Rule generation from neural networks, IEEE Transactions on Systems, Man and Cybernetics 24 (8) (1994) 1114–1124.
[6] J. Benítez, J. Castro, I. Requena, Are artificial neural networks black boxes? IEEE Transactions on Neural Networks 8 (5) (1997) 1156–1164.
[7] M. Figueiredo, F. Gomide, Design of fuzzy systems using neurafuzzy networks, IEEE Transactions on Neural Networks 10 (4) (1999) 815–827.
[8] F. Maire, Rule-extraction by backpropagation of polyhedra, Neural Networks 12 (1999) 717–725.
[9] J. Alexander, M. Mozer, Template-based procedures for neural network interpretation, Neural Networks 12 (1999) 479–498.
[10] I. Taha, J. Ghosh, Symbolic interpretation of artificial neural networks, IEEE Transactions on Knowledge and Data Engineering 11 (3) (1999) 448–463.
[11] R. Setiono, Extracting m-of-n rules from trained neural networks, IEEE Transactions on Neural Networks 11 (2) (2000) 512–519.
[12] H. Tsukimoto, Extracting rules from trained neural networks, IEEE Transactions on Neural Networks 11 (2) (2000) 377–389.
[13] J. Castro, C. Mantas, J. Benítez, Interpretation of artificial neural networks by means of fuzzy rules, IEEE Transactions on Neural Networks 13 (1) (2002) 101–116.
[14] E. Kolman, M. Margaliot, Are artificial neural networks white boxes? IEEE Transactions on Neural Networks 16 (4) (2005) 844–852.
[15] L. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, part 1, Information Sciences 8 (1975) 199–249.
[16] R. Andrews, J. Diederich, A. Tickle, Survey and critique of techniques for extracting rules from trained artificial neural networks, Knowledge-Based Systems 8 (6) (1995) 373–389.
[17] S. Mitra, Y. Hayashi, Neuro-fuzzy rule generation: survey in soft computing framework, IEEE Transactions on Neural Networks 11 (3) (2000) 748–768.

[18] H. Jacobson, Rule extraction from recurrent neural networks: a taxonomy and review, Neural Computation 17 (6) (2005) 1223–1263.
[19] R.R. Korfhage, Information Storage and Retrieval, John Wiley & Sons, 1997.
[20] R. Duda, P. Hart, D. Stork, Pattern Classification, second ed., Wiley Interscience, 2000.
[21] Y. Takagi, M. Sugeno, Fuzzy identification of systems and its application to modeling and control, IEEE Transactions on Systems Man and Cybernetics 15 (1985) 116–132.
[22] R. Yager, A. Rybalov, Uninorm aggregation operators, Fuzzy Sets and Systems 80 (1996) 111–120.
[23] M. Detyniecki, Mathematical aggregation operators and their application to video querying, Ph.D. thesis, Laboratoire d'Informatique de Paris, 2000.
[24] H. Zimmermann, Fuzzy Set Theory and its Applications, 2nd ed., Kluwer Academic Publishers, Dordrecht-Boston, MA, 1991.
[25] G. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice Hall, 1995.
[26] H. Zimmermann, P. Zysno, Latent connectives in human decision making, Fuzzy Sets and Systems 4 (1980) 37–51.
[27] P. Klement, R. Mesiar, E. Pap, On the relationship of associative compensatory operators to triangular norms an conorms, International Journal of Uncertainty, Fuzziness and Knowledge-based Systems 4 (2) (1996) 129–144.
[28] R. Yager, A. Rybalov, Full reinforcement operators in aggregation techniques, IEEE Transactions on Systems Man and Cybernetics 28 (1998) 757–769.
[29] W. Silvert, Symmetric summation: a class of operations on fuzzy sets, IEEE Transactions on Systems, Man and Cybernetics SMC-9 (1979) 657–659, Reprinted in: D. Dubois, H. Prade, R. Yager (Eds.), Reading in fuzzy sets for intelligent systems, Morgan Kaufman, San Mateo, CA, 1993, pp. 77–79.
[30] M. Minsky, S. Papert, Perceptrons, MIT Press, 1969.
[31] Using the ADAP learning algorithm to forecast the outset of diabetes mellitus, IEEE Computer Society Press, 1988.
[32] D. Rumelhart, G. Hinton, R. Williams, Parallel Distributed Processing: Explanations in the Microstructure of Cognition, vol. 1: Foundation, Cambridge, MIT Press, MA, 1986 (Chapter: Learning internal representations by error propagation, pp. 318–362).
[33] W. Duch, R. Adamczak, K. Grąbczewski, A new methodology of extraction, optimization and application of crisp and fuzzy logical rules, IEEE Transactions on Neural Networks 12 (2) (2001) 277–306.