

Decompositional Rule Extraction from Artificial Neural Networks and Application in Analysis of Transformers

M. A. B. Amora, O. M. Almeida, A. P. S. Braga, F. R. Barbosa, S. S. Lima, L. A. C. Lisboa

Abstract--The artificial neural networks represent efficient computational models that are widely used to solve problems of difficult solution in Artificial Intelligence. The greatest difficulty associated with the use of Artificial Neural Networks (ANN) is in obtaining knowledge about its behavior, because of that ANNs are also considered as black-box methods. This paper presents a brief history of methods of extraction of knowledge, and in detail a method of interpreting the behavior of an artificial neural network by establishing a relation of equality between certain classes of neural networks and systems based on fuzzy rules, with modifications that allow the acquisition of rules coherent with the domain of the variables of the problem. An example of application is used to illustrate the method, considering the identification of incipient faults in transformers by using data from gas dissolved in transformer oil.

Index Terms-- Knowledge rule extraction, neural networks, fuzzy rule-based systems, transformer failure diagnosis.

I. INTRODUCTION

Artificial Neural Networks (ANNs) are used to solve complex problems of artificial intelligence and are known for their characteristic of parallelism. However, they are considered as "black box", since determining why a particular solution obtained from an ANN is a difficult task.

Fuzzy rule-based systems (FRBSs) are a powerful tool for solving problems of control (fuzzy control) and to model knowledge.

Researches for the extraction of knowledge from ANNs have been growing in importance. This paper presents a brief history of methods for obtaining knowledge rules.

In the example presented in this paper we illustrated how to obtain rules easier to be interpreted in an ANN designed to classify incipient faults in a power transformer, using information from gases dissolved in transformer oil. In this example of application, we will be using the method proposed in [1] of similarities between certain types of ANNs and

FRBSs, allowing us to represent the knowledge of an ANN like fuzzy sentences, easier to be interpreted.

To ensure a higher degree of understanding of obtained rules, we apply a modification proposed in [2] to obtain coherent fuzzy rules.

In section two we present an analysis on methods of extraction of information. Comments about ANNs and FRBSs are presented in section three. And similarities between FRBSs and ANNs are discussed in section four. The achievement of coherent fuzzy propositions and the use of fuzzy logic operator \otimes_n^u are discussed in section five. In section six is presented the example to illustrate the usefulness of the method of similarities between ANNs and FRBSs. Finally, the conclusion of the article is presented in section 7.

II. KNOWLEDGE EXTRACTION METHODS

A. Knowledge Representation

Various forms of representation of knowledge were used by the techniques of learning in ANNs. A summary is presented below [3].

Conventional rules: It is an expression in two parts. The first part contains the background and the second part the consequence, i.e., the conclusion that the background is true. The form of these rules is:

$$IF (x_i \leq t_i) AND \dots AND (x_p \geq t_p) THEN C;$$

where x_i are continuous variables, t_i are real variables and C is a class designating a concept. Note that the discrete variables are a special case of continuous variables.

M-of-N rules: A simple rule M-of-N is equivalent to a conventional set of rules:

$$IF M \text{ of } (x_j \geq t_j) AND \dots AND (x_N \leq t_N) ARE TRUE THEN C;$$

Oblique rules: Are rules in the form:

$$IF (c_{01} | c_{11} x_1 | \dots | c_{n1} x_n \geq 0) AND \dots AND$$

$$(c_{0m} | c_{1m} x_1 | \dots | c_{nm} x_n \geq 0) THEN C$$

M. A. B. Amora, O. M. Almeida, A. P. S. Braga, F. R. Barbosa, e S. S. Lima are with Department of Electric Engineering of Federal University of Ceará, PO Box 6001 - Campus do Pici- Bloco 705 - 60.455-760 Fortaleza - CE - Brazil. E-mails: {marcio,fabio,sergio,arthurp,otacilio}@dee.ufc.br.

L. A. C. Lisboa work in Companhia Hidro Elétrica do São Francisco (CHESF), Rua Delmiro Gouveia, 333 - Bongí, 50761-901, Recife -PE, Brazil. E-mail: lcalmon@chesf.gov.br.

where x_i represents an input neuron and c_{ij} are real number. Compared with the conventional rules, the oblique rule are more difficult to understand, however allow us to create a separation edge in the input space.

Fuzzy rules: These rules use pertinence functions for dealing with partial truths, acting in a manner different to Boolean logic that only accepts situations totally True or totally False. The fuzzy rules have the form:

$$IF (x_1 \acute{e} f_i) AND \dots AND (x_p \acute{e} f_p) THEN C;$$

where f_i and C are defined as fuzzy sets with linguistic description.

Finite-state automata: It provides the simplest model of a computing device. It has a central processor of finite capacity and is based on the concept of state. Grammars are formalisms that generate languages. They provide the set of rules to enable to form correct sentences.

B. Taxonomy of Rule Extraction Algorithms

In this section, we use a taxonomy presented in [4] which uses three criteria for classification of rule extraction algorithms: scope of use, type of dependency with the method of solution of the type "black-box", and format of the extracted rules.

Concerning to the criterion scope of use, the algorithms can be a *regression* or *classification* algorithms. There are some algorithms that can be applied to both cases, such as the G-REX [4].

On the second criterion, an algorithm is considered *independent* if it is totally independent of the model type black-box used (ANN, Support Vector Machines, and others). The algorithms that use information of the black-box methods are called *dependent* methods.

Regarding the format of the extracted rules, the methods can be classified into *descriptive* and *predictive*. The *predictive* algorithms perform extraction of rules that allow the expert to make an easy prediction for each possible observation from input space. If this analysis can not be made directly, the algorithms are known only as *descriptive*.

Table 1 shows the classification for some algorithms to extract knowledge.

TABLE 1
TAXONOMY OF SOME RULE EXTRACTION ALGORITHMS

	Independent		Dependent	
	Classification	Regression	Classification	Regression
Predictive	CART, C4.5, TREPAN [7], G-REX [8], BIO-RE [9]	ITER, G-REX [8], CART, ANN-DT [14]	Barakat [15], Fung [16], FERNN [17], NeuroLinear [18], RE-RX [19]	REFANN [22], RN2 [23]
Descriptive	STARE [10], RAFNE [11], GEX [12], BUR [13]		SVM [20], VIA [21], Castro et al. [2]	

III. ARTIFICIAL NEURAL NETWORKS AND FUZZY RULE-BASED SYSTEMS

In the Fig. 1 is shown an example of feed-forward ANN with one hidden layer, and layers of input and output. Where there are n neurons (x_1, \dots, x_n) in the input layer, hidden neurons h (z_1, \dots, z_h), and m neurons in the output layer (y_1, \dots, y_m); w_{ij} is the weight of the connection between a neuron x_i of the input layer and one neuron z_j , and β_{jk} the weight of the connection between the neuron z_j and the neuron y_k . In this network are considered too, τ_j as the value of the bias for the neuron z_j of the hidden layer, and φ_k as the bias for the output neuron y_k .

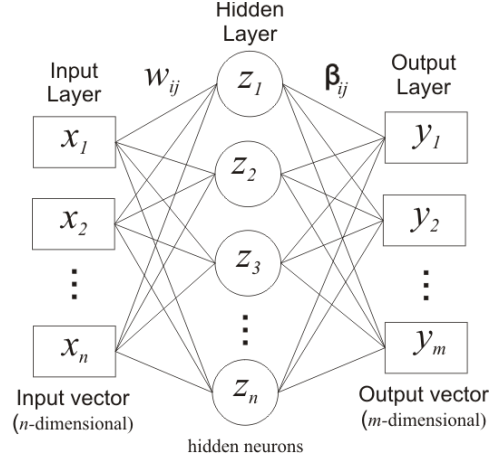


Fig. 1. RNA feedforward [6].

The activation functions of this given network are g_A and f_A , for the neurons of the hidden and output layer respectively. These functions are usually continuous, non decreasing and non-linear.

The ANN will represent a network function that:

$$F : \mathfrak{R}^n \rightarrow \mathfrak{R}^m; F(x_1, \dots, x_n) = (y_1, \dots, y_m)$$

$$\text{with: } y_k = g_A \left(\sum_{j=1}^h z_j \beta_{jk} + \varphi_k \right) \quad (1)$$

$$z_j = f_A \left(\sum_{i=1}^n x_i w_{ij} + \tau_j \right) \quad (2)$$

It can be demonstrated that an ANN with only one hidden layer has the ability to approximate any function with certain accuracy, regardless of their complexity. However, many researchers have criticized the use of ANNs because of its behavior described as black-box since the networks do not provide a satisfactory explanation for its behavior.

The fuzzy logic supports modes of reasoning that are approximate rather than exact, being based on the theory of fuzzy sets. In the FRBS the knowledge is represented using linguistic expressions related to numeric figures, and therefore more accessible to human understanding.

As shown in Figure 2, an FRBS is composed of four parts: fuzzification, knowledge base, inference machine, and defuzzification.

The process of fuzzification converts real values to fuzzy values, defined by fuzzy sets. The functions that define the ranges of linguistic variables are represented in the database, along with the fuzzy rules. The inference machine calculates the fuzzy output. The defuzzification transforms the value of fuzzy output of the system in a crisp value.

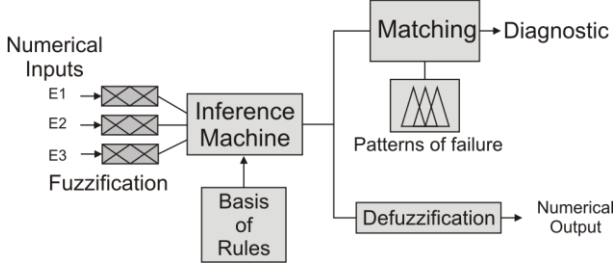


Fig. 2. The block diagram of a FRBS [6].

The fuzzy rules have the following form:

R_i : IF x_1 is A_1i AND x_2 is A_2i AND ... AND x_n is A_ni

THEN $y \in B_i$ (3)

where x_1, \dots, x_n are the inputs, y is the output, and A_1^i, \dots, A_n^i and B_i are linguistic variables. The Fuzzy Additive Systems (FASs) has as output a linear function of the inputs.

IV. EQUIVALENCE BETWEEN ANNS AND FRBSs

When the activation functions of an ANN are continuous, the calculated function of the network will be continuous. Then an ANN can be approximated by a FRBS, and vice versa [1].

The equivalence between ANNs and FRBSs has been studied by several authors in the past. Many of these studies establish this equivalence through processes of approximation. However, to this equivalence occur with a high degree of accuracy requires a large number of rules for a FRBS approximates an ANN.

However, in [1] is shown that a FAS can represent the same function of an ANN with a finite number of rules and equality in mathematical results. Following a brief demonstration will be presented in the reference.

If we consider a feedforward neural network of three layers with a logistic activation function in the hidden layer and an identity function for the neurons of the output layer. Then there is a FAS that computes the same function of the neural network.

To describe a fuzzy system is required only the basis of rules. Employing standard rules of Takagi-Sugeno-Kang (TSK), for each pair of neurons between the hidden and output layers, $(z_j \in y_k)$ is added:

$$R_{jk} : IF \sum_{i=1}^n x_i \cdot w_{ij} + \tau_j \in A THEN y_k = \beta_{jk} \quad (4)$$

where A is a fuzzy value in \mathfrak{R} , and the function of pertinence is simply a function of activation f_A of the neurons in the hidden layer.

Since this is a fuzzy additive system, the firing strength for rule R_{jk} , v_{jk} , will be given by $A\left(\sum_{i=1}^n x_i w_{ij} + \tau_j\right)$, and the output of the system is a vector with components given by:

$$y_k = \sum_{j=1}^h A\left(\sum_{i=1}^n x_i w_{ij} + \tau_j\right) \cdot \beta_{jk} \quad (5)$$

Through (4) and (5), we can easily check that the output y_k of FAS is exactly the same obtained with an ANN.

In this development, we used simple rules like “IF z is A THEN $y = v$ ” where $v \in \mathfrak{R}$ and z are new variables obtained by a change of variable in the n inputs. Also, the fuzzy sets A can be understood as “greater than approximately r ” where r is a positive real number obtained from a cut-off pre-established value. Since the logistics functions can vary from 0 to 1, assynthotically, is usual in the literature consider the values (levels) 0.1 and 0.9 for total absence of activation and full activation, respectively.

The rules obtained can also be modified to allow an easier interpretation. For this, we can make the decomposition of the premises of the rules, which can be rewritten:

$$R_{jk} : SE x_1 \in A_{jk}^1 \theta x_2 \in A_{jk}^2 \theta \dots \theta x_n \in A_{jk}^n \\ ENTÃO y_k = \beta_{jk} \quad (6)$$

where θ is a logical connective and A_{jk}^i are new values of the fuzzy set obtained from A , the weights w_{ij} and bias τ_j .

V. COHERENT FUZZY PROPOSITIONS

The method described in the previous section allows us to obtain fuzzy rules from a trained ANN, however many times these rules will not be in the domain of the input variables, making its interpretation difficult. In order to solve this problem we used in this paper a methodology for changing the fuzzy rules described in [2] that allow us to obtain new fuzzy rules according to the domain of the input variables of the problem.

This method consists in change the ranges of definition of fuzzy variables of the problem, using a new logical operator in the connection of the premises of fuzzy rules obtained from the ANN. This new operator offer us series of properties, described in [2], useful to the purpose of obtain knowledge from fuzzy rules acquired from an ANN. The operator is defined by:

$$\otimes_n^u(a_1, \dots, a_n) = \frac{a_1^{u-1} \dots a_n^{u-1}}{a_1^{u-1} \dots a_n^{u-1} + (1-a_1)^{u-1} \dots (1-a_n)^{u-1}} \quad (7)$$

If we consider a multilayer ANN with a single hidden layer as illustrated in the Fig. 1, also adopting that some of the weights w_{ij} are positive and others negative, disregard the loss of generalization, so that: $w_{ij} < 0$ for $1 \leq i \leq p$ and $w_{ij} > 0$ for $p \leq i \leq n$. So the procedure to transform the fuzzy rules for the domain of input variables of the problem consists of the following steps:

- 1) Transform the domains of the inputs variables. To obtain fuzzy rules with the same operator \otimes_n^u , a common value u must be used for all the transformations. Hence, this step consists of calculating:

$$\tau_0 = \frac{\min\{|w_{ij}|, 1 \leq i \leq n, 1 \leq j \leq h\}}{2}, \text{ and } u = \frac{4}{\tau_0}.$$

Now, the function $T(x) = u \cdot x$ and the operator \otimes_n^u are common for all the rules.

- 2) For each output neuron y_k , a fuzzy rule " R_{0k} : If TRUE Then $y_k = \varphi_k$ " is added to the rule base.
- 3) For each pair of neurons (z_j, y_k) , the following rule is added:

$$R_{jk}: \text{ If } -T(x_i, w_{ij} + \tau_0) \text{ is not greater approximately } 2,2 \otimes_{n+1}^u \dots -T(x_p, w_{pj} + \tau_0) \text{ is not greater approximately } 2,2 \otimes_{n+1}^u T(x_{(p+1)}, w_{(p+1)j} - \tau_0) \text{ is greater approximately } 2,2 \otimes_{n+1}^u \dots T(x_n, w_{nj} - \tau_0) \text{ is greater approximately } 2,2 \otimes_{n+1}^u \gamma_j \text{ Then } y_k = \beta_{jk}.$$

where:

- a) $-T(x_i, w_{ij} + \tau_0)$ is not greater approximately $2,2 \equiv x_i$ is not greater approximately $(2,2 + u \cdot \tau_0) / (-u \cdot w_{ij}) = 6,2 / (-u \cdot w_{ij}) = \lambda_{ij}$, and μ is not greater approximately $\lambda_{ij}(x) = f_A((6,2 / \lambda_{ij}) \cdot x - 4)$.
- b) $T(x_i, w_{ij} - \tau_0)$ is greater approximately $2,2 \equiv x_i$ is greater approximately $(2,2 + u \cdot \tau_0) / (u \cdot w_{ij}) = 6,2 / (u \cdot w_{ij}) = \lambda_{ij}$, and μ is not greater approximately $\lambda_{ij}(x) = f_A((6,2 / \lambda_{ij}) \cdot x - 4)$.
- c) $\gamma_j = f_A(T(\tau_j - p \cdot \tau_0 + (n-p) \cdot \tau_0))$.

Therefore:

$$R_{jk}: \text{ If } x_i \text{ is not greater approximately } \lambda_{ij} \otimes_{n+1}^u \dots x_p \text{ is not greater approximately } \lambda_{pj} \otimes_{n+1}^u \dots x_{(p+1)} \text{ is greater approximately } \lambda_{(p+1)j} \otimes_{n+1}^u \dots x_n \text{ is greater approximately } \lambda_{nj} \otimes_{n+1}^u \gamma_j \text{ Then } y_k = \beta_{jk}.$$

Two kinds of biases appear in the multilayer ANN of Fig. 1: biases φ_k ($k = 1, \dots, m$) of the output neurons and biases τ_j ($j = 1, \dots, h$) of the hidden neurons.

The biases φ_k generate the rules " R_{0k} : If TRUE Then $y_k = \varphi_k$ " in the fuzzy rule-based system. These rules provide a default value for each output y_k . If the remaining rules are fired, they only modify this default output value. This in straight connection with the human reasoning process where a default value is modified as new information is considered.

The biases τ_j generates the constants γ_j that appears in the antecedents of the fuzzy rules R_{jk} . The constants γ_j provide a default value, $\otimes_1^u(\gamma_j)$, for the firing strength of the antecedents. The remaining fuzzy propositions in the

antecedents only modify this default value.

VI. ANALYSIS OF TRANSFORMERS

To illustrate the procedure shown in this article we will consider an example of application based on the analysis of incipient faults in transformers, using data collected from gases dissolved in insulating oil of the transformer.

The methods of diagnosis in transformers based on DGA (Dissolved Gas Analysis) are widely used. These methods are based on the analysis of the concentration and rate of production of gases generated and dissolved in transformer oil, and they try to relate the type of failure with the gas present. For example, electric discharges lead to the generation of acetylene, as the presence of carbon dioxide is linked to the overheating of the cellulose. Conventionally, the diagnosis is performed through the interpretation of laboratory results by an expert.

From a database of samples of gases collected from power transforms we trained an ANN with 3 inputs, 10 neurons in hidden layer and one neuron in output layer. For the neurons of hidden layer we used a logistic function and the for the output neuron a linear function. Therefore, we used a topology consistent with the method shown in this article.

Of the data set used, consisting of 135 samples, 95 samples were used for training the neural network and 40 for its validation. The input of the ANN is trained through the ratios of gases: $R_1 = \text{CH}_4 / \text{H}_2$, $R_2 = \text{C}_2\text{H}_2 / \text{C}_2\text{H}_4$ e $R_5 = \text{C}_2\text{H}_4 / \text{C}_2\text{H}_6$. The output of the ANN can indicate the numerical code from 1 to 5, where 1 designates thermal failure of low temperature, 2 designates thermal failure of high temperature, 3 designates discharge of low energy, 4 designates discharge of high energy and 5 designates cellulose degradation.

After training and validation of the network were obtained the following values of weights and bias for the neurons and connections of the ANN:

$$W^t = [w_{ij}]^t = \begin{bmatrix} -0,131158 & -0,181829 & -0,18724 \\ -1,480831 & -6,776136 & 3,80546 \\ 1,7633864 & 7,110725 & -3,697832 \\ 0,3544984 & 0,565215 & 0,214268 \\ -5,274797 & -5,327245 & -0,271477 \\ 3,8866752 & 4,33527 & -0,807768 \\ -3,868009 & -4,331852 & 0,777081 \\ -0,275689 & -0,110015 & -0,125491 \\ 0,9477073 & 0,378771 & -2,054748 \\ -0,041112 & 2,022798 & 0,288286 \end{bmatrix}; \tau = [\tau_j] = \begin{bmatrix} -0,1935 \\ 6,1058 \\ -1,4821 \\ -0,7320 \\ 1,7463 \\ -1,1268 \\ 1,1154 \\ -0,1386 \\ 0,9239 \\ -1,3250 \end{bmatrix}$$

$$B' = [\beta_{jk}] = [16,8912 \quad -3,9712 \quad -0,0585 \quad -0,8484 \quad 0,0336 \\ -1,3175 \quad -1,3725 \quad 5,7971 \quad -0,0009 \quad 0,0284]; \\ \varphi = [\varphi_k] = [-4,7624]$$

Applying the methodology presented in this article, a FRBS was built, equivalent to trained ANN, containing eleven rules. As example we present below three of these extracted rules:

Rule 1: IF TRUE THEM Y = -4.7624.

Rule 2: IF R_1 is not greater than approximately 0.24293 $\otimes_4^{194.59}$ R_2 is not greater than approximately 0.17523 $\otimes_4^{194.59}$ R_5 is not greater than approximately 0.17017 $\otimes_4^{194.59}$ $4.96 \cdot 10^{-24}$ THEN $Y = 16.8912$.

Rule 3: IF R_1 is not greater than approximately 0.02152 $\otimes_4^{194.59}$ R_2 is not greater than approximately 0.00448 $\otimes_4^{194.59}$ R_5 is greater than approximately 0.0084 $\otimes_4^{194.59}$ 1 THEN $Y = -3.97118$.

Rule 4: IF R_1 is not greater than approximately 0.0181 $\otimes_4^{194.59}$ R_2 is greater than approximately 0.0045 $\otimes_4^{194.59}$ R_5 is not greater than approximately 0.0086 $\otimes_4^{194.59}$ $6.33 \cdot 10^{-133}$ THEN $Y = -0.0585$.

Rule 5: IF R_1 is greater than approximately 0.0899 $\otimes_4^{194.59}$ R_2 is greater than approximately 0.0564 $\otimes_4^{194.59}$ R_5 is not greater than approximately 0.1487 $\otimes_4^{194.59}$ $1.56 \cdot 10^{-69}$ THEN $Y = -0.8484$.

Rule 6: IF R_1 is not greater than approximately 0.006 $\otimes_4^{194.59}$ R_2 is not greater than approximately 0.006 $\otimes_4^{194.59}$ R_5 is not greater than approximately 0.1174 $\otimes_4^{194.59}$ 1 THEN $Y = 0.0336$.

Rule 7: IF R_1 is greater than approximately 0.0082 $\otimes_4^{194.59}$ R_2 is greater than approximately 0.0073 $\otimes_4^{194.59}$ R_5 is not greater than approximately 0.0394 $\otimes_4^{194.59}$ $6.66 \cdot 10^{-103}$ THEN $Y = -1.3175$.

Rule 8: IF R_1 is not greater than approximately 0.0082 $\otimes_4^{194.59}$ R_2 is not greater than approximately 0.0074 $\otimes_4^{194.59}$ R_5 is greater than approximately 0.0410 $\otimes_4^{194.59}$ 1 THEN $Y = -1.3725$.

Rule 9: IF R_1 is not greater than approximately 0.1156 $\otimes_4^{194.59}$ R_2 is not greater than approximately 0.2896 $\otimes_4^{194.59}$ R_5 is not greater than approximately 0.2539 $\otimes_4^{194.59}$ $2.19 \cdot 10^{-19}$ THEN $Y = 5.7971$.

Rule 10: IF R_1 is greater than approximately 0.0336 $\otimes_4^{194.59}$ R_2 is greater than approximately 0.0841 $\otimes_4^{194.59}$ R_5 is not greater than approximately 0.0155 $\otimes_4^{194.59}$ 1 THEN $Y = -0.0009$.

Rule 11: IF R_1 is not greater than approximately 0.775 $\otimes_4^{194.59}$ R_2 is greater than approximately 0.0152 $\otimes_4^{194.59}$ R_5 is not greater than approximately 0.1105 $\otimes_4^{194.59}$ $1.19 \cdot 10^{-119}$ THEN $Y = 0.0284$.

The value of $\mu = 194.59$ and the values associated with the independent terms (bias) in the rules were calculated as shown in section 5.

Table 2 presents some results. The **Values** column shows examples of the data (R_1 , R_2 and R_5) collected from power transforms. The numerical values obtained from the trained

ANN and from **FRBS** are always equal, proving the effectiveness of the method of equality used in this paper. The **Results** column shows the values of the RNA without regard to standardization, and **Goal** indicates the expected values. Therefore, the ANN and the FRBS was able to generate correctly the numerical result of diagnostic analysis of the power transform.

TABLE 2
SIMULATION RESULTS

Values	RNA	SFA	Results	Goal
R1=-1.2283; R2=0.6245; R5=0.9797	-0,3772	-0,3772	4,0155	4,0000
R1=-0.6242; R2=-0.6288; R5=1.4721	-0,2190	-0,2190	1,9956	2,0000
R1=-0.6955; R2=-0.7154; R5=-0.0050	1,4159	1,4159	3,0016	3,0000

Using the method presented, the premises of the rules obtained present values in the range of the input variables of the problem. This way, we can improve the understanding of the rules.

Analyzing the values of the bias obtained through the training of the ANN and that influence the γ terms of the rules extracted, we observed that values of bias that resulted in very small γ values indicate a low influence of this rule to the numerical result of the application of the set of rules. This influence can be evaluated in a simple way through the relation $\otimes_1^{194.59} \gamma$. Making an analysis of the influence of the values of bias in the rules extracted we observed that three rules of the original set could be omitted.

Also, looking at the output value of the rule extracted through the ninth neuron of hidden layer of trained ANN, it shows the value -0.0009 (value of the weight which connects the neuron with the output of the ANN), which represents a very small value compared to other available outputs, so it can be omitted.

Removing these four rules that have a small influence on the outcome of the fuzzy system, a new test was performed for the values listed in Table 2 and the new results are presented in Table 3

TABLE 3
NEW SIMULATION RESULTS

Valores	RNA	SFA
R1=-1.2283; R2=0.6245; R5=0.9797	-0,3772	-0,3659
R1=-0.6242; R2=-0.6288; R5=1.4721	-0,2191	-0,2213
R1=-0.6955; R2=-0.7154; R5=-0.0050	1,4159	1,4160

As we can see, comparing the results of Tables 2 and 3, the exclusion of the four rules of the original set did not significantly alter the outcome of the FRBS, which is almost equal to that obtained from the ANN, demonstrating a small influence of the omitted rules.

VII. CONCLUSION

In this paper we presented a brief discussion of methods for extraction of knowledge rules.

It was also demonstrated, in particular, a method for obtaining a FRBS from a trained ANN, with modifications that allow us to acquire consistent values with the domain of the input variables of the problem to the range of evaluation in premises of the rules.

Two important conclusions can be observed from the

equality between FRBSs and ANNs. First of all, everything discovered in a model can be applied to the other model. Second, the knowledge of an ANN associated with the connections and synaptic weights can be expressed as fuzzy rules, allowing a better understanding of this knowledge.

We can also see the influence of the terms associated with the values of bias of neurons in the rules extracted. This indicates that certain rules can be omitted without any loss to the numerical result.

VIII. ACKNOWLEDGMENT

The authors are grateful to the support provided by Companhia Hidro Elétrica do São Francisco (CHESF) in this paper.

IX. REFERENCES

- [1] J. M. Benitez, J. L. Castro, and I. Requena, "Are artificial neural networks black boxes?", *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 1156–1164, Sep. 1997.
- [2] J. L. Castro, C. J. Mantas, and J. M. Benitez, "Interpretation of artificial neural networks by means of fuzzy rules", *IEEE Trans. Neural Netw.*, vol. 13, no. 1, pp. 101–116, Jan. 2002.
- [3] G. Bologna, "A model for single and multiple knowledge based networks", *ELSEVIER Artificial Intelligence in Medicine* no. 28, pp. 141–163, 2003.
- [4] J. Huysmans, B. Baesens and J. Vanthienen. "Using rule extraction to improve the comprehensibility of predictive models". *Katholieke Universiteit Leuven. Department of Decision Sciences and Information Management. Leuven, Belgium, 2006.*
- [5] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators" *Neural Networks*, vol. 2, pp. 359–366, 1989.
- [6] Lima S. E. U. "Diagnóstico Inteligente de Falhas Incipientes em Transformadores de Potência Utilizando Análise dos Gases Dissolvidos em Óleo", *Dissertação de Mestrado, UFC/CT/DEE, Fortaleza-CE, 2005.*
- [7] M.W. Craven and J.W. Shavlik. "Extracting tree-structured representations of trained networks". In David S. Touretzky, Michael C. Mozer and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 24–30. The MIT Press, 1996.
- [8] U. Johansson, R. König and L. Niklasson. "Rule extraction from trained neural networks using genetic programming". In *Joint 13th International Conference on Artificial Neural Networks and 10th International Conference on Neural Information Processing, ICANN/ICONIP 2003*, pages 13–16, 2003.
- [9] I. Taha and J. Ghosh. "Symbolic interpretation of artificial neural networks". *IEEE Transactions on Knowledge and Data Engineering*, 11(3):448–463, 1999.
- [10] Z.-H. Zhou, S.-F. Chen and Z.-Q. Chen. "A statistics based approach for extracting priority rules from trained neural networks". *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, 3, 2000.
- [11] Z.-H. Zhou, Y. Jiang and S.-F. Chen. "Extracting symbolic rules from trained neural network ensembles". *AI Communications*, 16(1):3–15, 2003.
- [12] U. Markowska-Kaczmar and M. Chumieja. "Discovering the mysteries of neural networks". *International Journal of Hybrid Intelligent Systems*, 1(3–4):153–163, 2004.
- [13] F. Chen. "Learning accurate and understandable rules from SVM classifiers". *Master's thesis, Simon Fraser University, 2004.*
- [14] G.P.J. Schmitz, C. Aldrich and F.S. Gouws. "ANN-DT: An algorithm for extraction of decision trees from artificial neural networks". *IEEE Transactions on Neural Networks*, 10(6):1392–1401, 1999.
- [15] N. Barakat and J. Diederich. "Eclectic rule-extraction from support vector Machines". *International Journal of Computational Intelligence*, 2(1):59–62, 2005.
- [16] G. Fung, S. Sandilya and R.B. Rao. "Rule extraction from linear support vector machines". In *11th ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 32–40, 2005.
- [17] R. Setiono and W.K. Leow. "FERNN: An algorithm for fast extraction of rules from neural networks". *Applied Intelligence*, 12(1–2):15–25, 2000

- [18] R. Setiono and H. Liu. "Neurolinear: From neural networks to oblique decision rules". *Neural Computing*, 17(1):1–24, 1997.
- [19] R. Setiono and B. Baesens. "Risk management using recursive neural network rule extraction". *Submitted to Management Science, 2006.*
- [20] H. Núñez, C. Angulo and A. Català. "Rule extraction from support vector machines". In *European Symposium on Artificial Neural Networks (ESANN)*, pages 107–112, 2002.
- [21] S. Thrun. "Extracting provably correct rules from artificial neural networks". *Technical report iai-tr-93-5, Universität Bonn, Institut für Informatik III, 1993.*
- [22] R. Setiono, W.K. Leow and J.M. Zurada. "Extraction of rules from artificial neural networks for nonlinear regression". *IEEE Transactions on Neural Networks*, 13(3):564–577, 2002.
- [23] K. Saito and R. Nakano. "Extracting regression rules from neural networks". *Neural Networks*, 15(10):1279–1288, 2002.

X. BIOGRAPHIES

Márcio A. B. Amora was born in Fortaleza-CE, Brazil, in 1972. Has degree in Electrical Engineering at Federal University of Pará (1997), master degree in Electrical Engineering at Federal University of Pará (2001). He is currently assistant professor at the Federal University of Ceará. He has experience in Electrical Engineering with emphasis on Applied Computational Intelligence, working mainly in the following areas: diagnosis of incipient faults in transformers, analysis of load of power transforms, generation of wind energy, analysis of transient stability.

Otaclio M. Almeida has degree in Electrical Engineering at Federal University of Ceará (1987), Masters in Electrical Engineering at Universidade Estadual de Campinas (1990) and doctorate degree in Electrical Engineering at Universidade Federal de Santa Catarina (2002). He is currently an adjunct professor at the Federal University of Ceará. He has experience in Electrical Engineering with emphasis on Automotive Electronics and Electrical Industrial Process, working mainly in the following areas: industrial control, process control, PID controller, electrical industrial machinery and smart controllers.

Arthur P. S. Braga was born in Natal-RN, Brazil, in 1971. Has degree in Electrical Engineering at Federal University of Ceará (1995), masters degree in Electrical Engineering at University of São Paulo (1998), doctorate degree in Electrical Engineering the University of São Paulo[S. Carlos] (2004), post-doctorate at University of São Paulo / São Carlos - USP / SC (2006). He is currently an adjunct professor at Federal University of Ceará. He has experience in Electrical Engineering with emphasis on Artificial Neural Networks, working mainly in the following areas: reinforcement learning, autonomous agents, neural networks, artificial intelligence, Self Organizing Maps.

Fábio R. Barbosa was born in Fortaleza-CE, Brazil, in 1979. He has medium technical level in Electrotechnics at Federal Center of Technological Education of Ceará (1999) and degree in Electrical Engineering at Federal University of Ceará (2004), masters degree in Electrical Engineering at Federal University of Ceará (2008). He has experience in research in Artificial Intelligence applied to Electrical Engineering and he had consultancy activities in the area of industrial automation and programmable logic control with the National Service for Industrial Apprenticeship of Ceará - SENAI. Currently, he is a post-graduate scholarship from CAPES and develops research in the field of monitoring and diagnosis of power transformer immersed in insulating oil as a doctoral student of the Graduate Program of the Department of Electrical Engineering at Federal University of Ceará.

Sergio S. Lima was born in Fortaleza-CE, Brazil, in 1973. He has degree in Computer Science at University of Fortaleza – UNIFOR (2000). Currently, he is student of the master program in Electrical Engineering at Federal University of Ceará and scholarship of the Laboratory of the Group of Research on Automation and Robotics (GPAR) at Department of Electrical Engineering of the Federal University of Ceará, and develops research in the field of monitoring and diagnosis of power transformer immersed in insulating oil.

Luciano A. C. Lisboa has degrees in Electronic Engineering at Universidade Federal de Pernambuco (2002). Currently he is an Engineer at Hydro Electric Company of San Francisco. He has experience in Electrical Engineering with emphasis on Industrial Electronics and Electronic Systems and Controls, Sistemas e Controles Eletrônicos.