Knowledge Discovery in Neural Networks With Application to Transformer Failure Diagnosis

Adriana Rosa Garcez Castro, *Student Member, IEEE,* and Vladimiro Miranda, *Senior Member, IEEE*

***Abstract*—The paper describes a new methodology for mapping a neural network into a rule-based fuzzy inference system. This mapping makes explicit the knowledge implicitly captured by the neural network during the learning stage, by transforming it into a set of rules. The method is applied in transformer fault diagnosis using dissolved gas-in-oil analysis. Studies on transformer failure diagnosis are reported, illustrating the good results obtained and the knowledge discovery made possible.**

***Index Terms*—Fault diagnosis, fuzzy logic, neural networks.**

I. Introduction

TRANSFORMER faults, mainly in the form of overheating, arcing or partial discharge, develop certain gaseous hydrocarbons, which are retained by the insulating oil as dissolved gases. Their concentration, relative proportion and generation rate have been extensively used for the estimation of the condition of a transformer, and Dissolved gas-in-oil Analysis (DGA) methods such as Dornenburg Ratios, Rogers Ratios, and IEC Ratios are commonly used by utilities and manufacturers [1], [2]. However, the characterization achieved so far still has a large margin for improvement.

Fuzzy inference systems (FIS) [3] have been tried in developing fault diagnosis systems. These have been built according to DGA methods and the efficiency of the models developed depended on the completeness of the knowledge of the specialist. Also, the rules in fuzzy logic based models could not be automatically adjusted through a self-learning process when new knowledge was acquired.

Also, artificial neural networks (ANN) have been proposed to deal with transformer fault diagnosis [4], [5]. However, it is often argued that ANNs do not have explaining capability and behave like black boxes. This is a drawback, because human understanding would be greatly enhanced if the relations between the variables were explicit, and engineers or technicians would also gain more confidence in the diagnoses produced. In short, ANNs adapt well to a problem of classification but the knowledge they've captured remains hidden, and fuzzy systems make an explicit display of knowledge but it is basically the knowledge of experts and not really learned from the problem.

This paper shows how we may build an ANN that captures knowledge from a transformer fault diagnosis data and how we may transform it into an equivalent FIS and expose, in the form of rules, the knowledge captured by the ANN. This allows knowledge discovery by human specialists and helps in understanding how the neural network arrives at a particular result. The usefulness of this approach is illustrated with its application to incipient transformer failure diagnosis. The necessary theory will be presented and then results will be discussed. The model corresponds to an evolution of a prior attempt [6], which still did not exhibit the desired property of transparency of the rule base as now achieved. Some concepts will be repeated for the sake of clarity.

The work presented does not constitute a comprehensive diagnosis system for incipient transformer failures. It does not take in account evolving rates for dissolved gases, nor other type of information that is not suited to be represented by if-then rules. It also assumes that there is a single major failure in a transformer and does not take in account the possibility of multiple simultaneous failures, whose mixed symptoms would eventually blur one another and confuse the diagnosis procedure. However, it represents an important theoretical step into building better and more robust diagnosis systems. And the fact that, at the present level, the methodology has already allowed obtaining better results than by using IEC 60 599 publication is encouraging. The reason why it gives better results may be investigated by analyzing the rules generated by the process of knowledge extraction described, and will be available for further research.

II. Neural Networks and Fuzzy Systems

The multilayer feedforward neural network, also known as multilayer perceptron (MLP), is well known. We will summarize its characteristics in what is required for the swift understanding of the developments further described. A MLP basically consists of a finite number of successive layers (Fig. 1), each having a finite number of processor units called neurons. Each neuron of every layer is connected to every neuron of a following layer through synaptic weights. Every neuron in a hidden layer calculates

$$s_j = f\left(\sum_{i=1}^{n} x_i w_{ij} + \theta_j\right) \quad (1)$$

where x_i is the ith input to the net, w_{ij} is the weight of the connection from input neuron i to hidden neuron j, θ_j is the bias of the jth hidden neuron, and $f(.)$ is the activation function of the neuron.

Manuscript received April 5, 2004; revised September 18, 2004. The work of A. R. G. Castro was supported by CAPES-Brazilian Government. Paper no. TPWRS-00810-2004.

A. R. G. Castro is with INESC Porto, Porto, Portugal, and also with NESC/UFPA—Federal University of Pará, Belém PA, Brazil (e-mail: acastro@inescporto.pt).

V. Miranda is with INESC Porto, Portugal, and also with FEUP, Faculty of Engineering of the University of Porto, Porto 4200–465, Portugal (e-mail: vmiranda@inescporto.pt).

Digital Object Identifier 10.1109/TPWRS.2005.846074

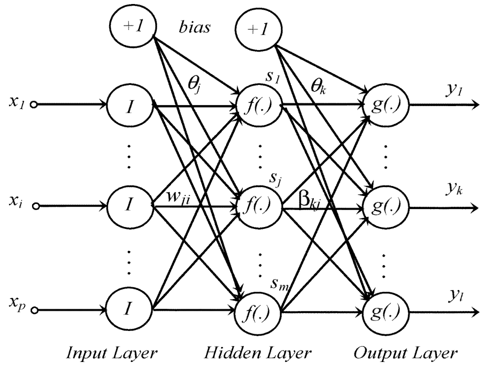

Fig. 1. Multilayer feedforward neural network structure.

For the output layer, each neuron calculates:

$$y_k = g\left(\sum_{j=1}^{m} \beta_{jk} s_j + \theta_k\right) \tag{2}$$

where

β_{jk} weight of the connection from hidden neuron j to output neuron k;

y_k *kth* output of the net;

θ_k bias of the *kth* output neuron;

$g(.)$ activation function of the neuron.

ANNs are universal approximators. It has been extensively demonstrated that a MLP working with arbitrary squashing functions in hidden neurons can approximate virtually any function of interest to any desired degree of accuracy [7].

In problems where the concern is mainly with results, such as in a control problem, then ANNs are satisfactory. However, in problems where knowledge is important, the black box nature of ANNs may undermine the confidence of specialists or system operators in their results.

On the other hand, FIS or fuzzy rule based systems, unlike ANN, are systems that have precisely the desired characteristics of an explicit form of knowledge representation. In Takagi-Sugeno (TS) fuzzy inference systems, the relationship between variables of the system is represented by fuzzy IF-THEN rules in the form

$$\begin{aligned}\textbf{Rule } \boldsymbol{R_l}:\ &\text{If } x_1 \text{ is } C_1^l \text{ and } \ldots \text{ and } x_n \text{ is } C_n^l\\ &\text{Then } y^l = t(x_1,\ldots,x_n)\end{aligned} \tag{3}$$

where

C_i^l fuzzy sets that may represent linguistic values;

x input vector of the system;

t function of the inputs.

The consequent of the rule is an affine linear or nonlinear function of the input variables and the output of the TS model is computed as the weighted average of y^l. When $y^l = t(x)$ is a constant, the fuzzy inference system is called a zero-order TS fuzzy model. Fig. 2 illustrates the reasoning mechanism for such a model.

In spite of their capacity of explanation, TS-FIS have some drawbacks, which greatly restrict their application.

1) There is no systematic method for the transformation of expert knowledge or experience into the rule base of a fuzzy inference system.

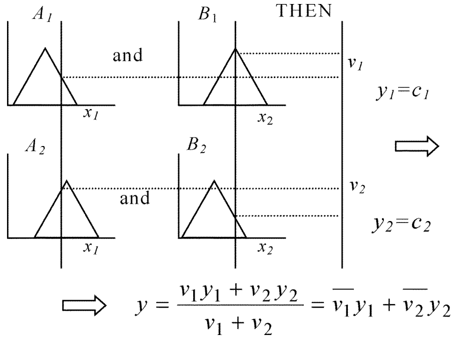
A_1
B_1
THEN
and
v_1
$y_1=c_1$
x_1
x_2
A_2
B_2
and
v_2
$y_2=c_2$
x_1
x_2
$\Rightarrow$
$\Rightarrow\ y = \dfrac{v_1 y_1 + v_2 y_2}{v_1 + v_2} = \overline{v_1} y_1 + \overline{v_2} y_2$

Fig. 2. Zero-order TS model.

2) Even when human specialists exist, their knowledge is often incomplete and episodic rather then systematic.

3) They suffer from the *curse of dimensionality*, meaning that the number of rules of the system grows exponentially when the number of inputs increases and computational complexity in the implementation for practical problems increases accordingly.

III. Rule Extraction From Neural Networks

Rule extraction from ANN may be organized under five primary classification criteria [8].

- The expressive power of the rules extracted.
- The quality of the extracted rule.
- Translucency.
- Algorithm complexity.
- Portability or generality.

In particular, translucency refers to the rule extraction technique based on the granularity of the underlying ANN; rule extraction from ANN can be categorized as decompositional, pedagogical, and eclectic:

- The *decompositional approach* regards rule extraction as a search process that maps the internal structure of a neural network to a set of rules. The analysis of numerical values of the network such as activation values of hidden and output neurons and weights of connections between them are used to extract the rules directly. We find examples of this approach in [9]–[12], just to name a few.
- The *pedagogical approach* does not disassemble the architecture of the trained neural network. Instead, it regards the ANN as an entity and tries to extract rules that could explain its function. The ANN is treated as a "black-box", where the extracted rules describe the global relationship between the variables of the input and output of the ANN. Examples of this approach are in [13]–[16].
- The *eclectic approach* incorporates elements of both the decompositional and the pedagogical models.

No method is exact and most of them have identified drawbacks, such as a curse of dimensionality (explosion of the number of rules), approximation degree, limitations in their applicability, etc. This paper presents an approach that may be classified as decompositional and that relies on an exact mathematical correspondence between ANN and TS-FIS—which is an advantage—no approximation is involved.

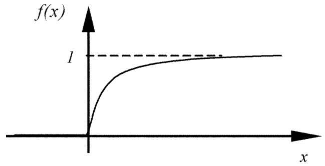

Fig. 3. Positive-sigmoid function.

IV. Mapping Neural Networks Into a Fuzzy Inference Model

A. Definition of the Topology of the ANN

In this paper we will consider an ANN with one hidden layer and a single output neuron. We recall that it has been shown that there is always an ANN with a single hidden layer equivalent to another ANN with arbitrary number of hidden layers. The activation function of the output neuron is linear and the hidden neurons have the following particular activation sigmoidal function, which we will call *positive sigmoid* and whose graphic is shown in Fig. 3

$$f(x)=\begin{cases}1-e^{-x}, & x\geq 0\\ 0, & x<0\end{cases}. \tag{4}$$

B. Applying the Concept of f-Duality

To produce the mapping of an ANN such as defined above into a TS-FIS, the concept of f-duality will be used [17]. This concept allows us to consider a transform providing an equivalent mathematical operation to (1)—the operation performed by the hidden neurons in the ANN. The following propositions and lemmas are useful:

Proposition 1: Let $f:X\rightarrow Y$ be a bijective function and let $\oplus$ be an operation defined in X, the domain of f. Then there is one and only one operation $\otimes$, defined in the range of f, Y, verifying

$$f\left(\overset{n}{\underset{i=1}{\oplus}} x_i\right)=\overset{n}{\underset{i=1}{\otimes}} f(x_i). \tag{5}$$

Definition 1: Let f be a bijective function and let $\oplus$ be an operation defined in domain of f. The operation $\otimes$ whose existence is proven in proposition 1 is called f-dual of $\oplus$.

Lemma 1: If $\otimes$ is the f-dual of $\oplus$ then $\oplus$ is the f^{-1} dual of $\otimes$.

Proofs of Proposition 1 and Lemma 1 are in [17].

Applying (5) to the positive sigmoid function f (4) and having $\oplus$ as the operation $+$ in $\Re$, we are lead to:

Lemma 2 (f-Duality): The f-dual of $+$ is $*$ and is defined as

$$\begin{aligned}&f(x_1+x_2+\ldots+x_n)\\&\quad=f(x_1)*f(x_2)*\ldots*f(x_n)\\&\quad=1-(1-f(x_1))(1-f(x_2))\ldots(1-f(x_n))\\&\qquad\text{if } x_1+x_2+\ldots+x_n\geq 0\quad\text{and}\quad x_i\geq 0.\end{aligned} \tag{6}$$

Proof of Lemma 2 is in the Appendix.

These proposition and lemmas allow us to write a new but equivalent expression for the output of the ANN, which can easily receive an interesting interpretation.

Applying these concepts to (1), without bias θ_j, where the activation function f is as in (4) and considering $\sum_{i=1}^{n} x_iw_{ij}\geq 0$, the output signal of the hidden neurons can be calculated by

$$\begin{aligned}s_j&=f\left(\sum_{i=1}^{n} x_iw_{ij}\right)=f(x_1w_{1j})*\ldots*f(x_nw_{nj})\\&=1-(1-f(x_1w_{1j}))\ldots(1-f(x_nw_{nj}))\\&\quad\text{if }\sum_{i=1}^{n} x_iw_{ij}\geq 0\quad\text{and}\quad x_iw_{ij}\geq 0.\end{aligned} \tag{7}$$

We recognize in (7) a logic operator, well known in fuzzy logic as an algebraic sum, which is an S-norm (OR-operator).

Function $f(x_iw_{ij})$ would be considered in Fuzzy Systems as a membership function. However, function $f(x)$ can only reach 1 asymptotically. To give it a linguistic translation, we arbitrarily take the α-cut for $\alpha=0.9$ as the limit above which the linguistic concept is fulfilled. Thus, $f(x_iw_{ij})$ may be considered as the membership function of a fuzzy set representing "x_i is greater than $2.3/w_{ij}$", where $f(2.3/w_{ij})=0.9$.

C. Extracting Rules From ANN

Using (7) and $f(x_iw_{ij})$, a neural network (with some constraints) can be mapped into a rule-based system.

In an ANN as shown in Fig. 1, having the hidden neurons without bias, $\sum_{i=1}^{n} x_iw_{ij}\geq 0$ and $x_iw_{ij}\geq 0$, for each neuron in hidden layer, one may state its output in fuzzy rule form as

$$\textbf{Rule } R_j:\ \textbf{If } \sum_{i=1}^{n} x_iw_{ij} \text{ is } A \textbf{ then } y_j=\beta_j \tag{8}$$

where A is a fuzzy set whose membership function is the positive-sigmoid function.

There is nothing new here, except the aspect of (8), which is a fuzzy set interpretation of (1) and (2). In fact, this is a rule characteristic of a zero-order TS-FIS, because the weighting function is a constant β. The output of the rule will be the product of β with the membership value of the antecedent of the rule, which is given by the activation function of the neuron—thus, the result is the same as in the ANN.

According to (7), rules as in (8) can be written as

$$\begin{aligned}\textbf{Rule } R_j:\ &\textbf{If } (x_1w_{1j}\text{ is } A)*\ldots*(x_iw_{ij}\text{ is } A)\\&*\ldots*(x_nw_{nj}\text{ is } A)\\&\textbf{then } y_j=\beta_j.\end{aligned} \tag{9}$$

Expression "x_iw_{ij} is A" may also be interpreted as "x_i is A_{ij}", defining the fuzzy set A_{ij} by a membership function $\mu(A_{ij})=f(x_iw_{ij})$, with the weight w_{ij} as a scaling factor of the slope of $f(.)$. Once the operation $*$ is the algebraic sum operator (OR), we may rewrite (9) as

$$\begin{aligned}\textbf{Rule } R_j:\ &\textbf{If } (x_1 \text{ is } A_{1j}) \textbf{ or}\ldots\textbf{or } (x_i\text{ is } A_{ij})\\&\textbf{or}\ldots\textbf{or } (x_n\text{ is } A_{nj})\\&\textbf{then } y_j=\beta_{\text{j}}\end{aligned} \tag{10}$$

where the firing strength of such rule is calculated by the algebraic sum operator, as follows:

$$\mathrm{v_j}=\mu(\mathrm{A_{1j}})*\ldots*\mu(\mathrm{A_{nj}})=1-((1-\mu(\mathrm{A_{ij}}))\ldots(1-\mu(\mathrm{A_{nj}})). \tag{11}$$

Finally, from the output neuron in Fig. 1, the output of the fuzzy system can be expressed as

$$y = \sum_{j=1}^{m} \beta_j s_j. \tag{12}$$

Since $s_j = v_j$ and $\beta_j = y_j$, (12) can be rewritten

$$y = \sum_{j=1}^{m} y_j v_j. \tag{13}$$

This has the form of an inference system, extracted from the neural net, similar to a zero-order Takagi-Sugeno model, with the difference that here the fuzzy logic operator used to calculate the firing strength of each rule is a S-norm (OR) and not a T-norm (AND).

However, for each S-norm there is a T-norm "associated" with it, where "associated" means that there is a fuzzy complement such that the two together satisfy the DeMorgan's Law [18].

The T-norm associated with the algebraic sum operator $S(a,b) = 1-(1-a)(1-b)$ is the algebraic product operator $T(a,b) = ab$. Therefore, the rule system extracted in (10) can be transformed into

$$\begin{aligned}\textbf{Rule } R_j:\ & \textbf{If } (x_1 \text{ is Not } A_{1j}) \textbf{and} \ldots \textbf{and} (x_i \text{ is Not } A_{ij}) \\ & \textbf{and} \ldots \textbf{and} (x_n \text{ is Not } A_{nj}) \\ & \textbf{then } y_j = \beta_j \end{aligned} \tag{14}$$

where the firing strength for each R_j rule is now calculated by the algebraic product operator (AND operator) and the system output is as follows:

$$y = \sum_{j=1}^{m} \beta_j (1 - v_j). \tag{15}$$

Rearranging (16) leads to the output of the fuzzy system as

$$y = \sum_{j=1}^{m} \beta_j (1 - v_j) = \sum_{j=1}^{m} \beta_j - \sum_{j=1}^{m} y_j v_j \tag{16}$$

where $\sum_{j=1}^{m} \beta_j$ is a default value of the fuzzy system output.

D. ANN With Bias

If a bias input is used in the hidden neurons, then (7) is rewritten as

$$\begin{aligned} s_j &= f\left(\sum_{i=1}^{n} x_i w_{ij} + \theta_j\right) \\ s_j &= f(x_1 w_{1j}) * \ldots * f(x_n w_{nj}) * f(\theta) \\ s_j &= 1 - ((1-\mu(A_{1j})) \ldots . (1-\mu(A_{nj}))\,(1 - f(\theta_j)) \\ & \text{if } \sum_{i=1}^{n} x_i w_{ij} \geq 0, \quad x_i w_{ij} \geq 0 \text{ and } \theta_j > 0. \end{aligned} \tag{17}$$

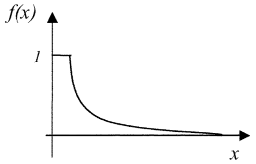

Fig. 4. New membership function extracted.

Considering the output of the system and (17), we have

$$\begin{aligned} y &= \sum_{j=1}^{m} \beta_j s_j \\ &= \sum_{j=1}^{m} \beta_j \left(1 - ((1-\mu(A_{1j}))\right. \\ &\qquad \left. \ldots . (1-\mu(A_{nj}))\,(1-f(\theta_j))\right) \\ &= \sum_{j=1}^{m} \beta_j - \sum_{j=1}^{m} \beta_j \left(1 - f(\theta_j)\,(1-\mu(A_{1j}))\right. \\ &\qquad \left. \ldots (1-\mu(A_{nj}))\right) \\ &= \sum_{j=1}^{m} \beta_j - \sum_{j=1}^{m} \beta_j\,(1-f(\theta_j))\,v_j \\ y &= \sum_{j=1}^{m} \beta_j - \sum_{j=1}^{m} \beta'_j v_j. \end{aligned} \tag{18}$$

If a bias is used in the hidden neuron, then the consequent of rule R_j will change from $y_j = \beta_j$ to $y_j = \beta'_j = \beta_j(1 - f(\theta_j))$. If a bias (θ_{out}) is used in the output neuron, the output of the system will change to

$$y = \sum_{j=1}^{m} \beta_j - \sum_{j=1}^{m} y_j v_j + \theta_{\text{out}} \tag{19}$$

where $\sum_{j=1}^{m} \beta_j + \theta_{\text{out}}$ is the new default value of the rule.

E. Comments

The process explained so far contains the basic idea to produce the mapping of ANNs into FIS. However, for the rule antecedents extracted from ANNs to be meaningful and subject to interpretation, we must be able to represent then in linguistic form. Consider the following condition:

Condition 2: If the negation (NOT) is applied to the extracted membership $\mu(A_i) = f(x_i w_{ij})$, we will have in (7) a new membership defined as (Fig. 4)

$$f(x_i w_{ij}) = \begin{cases} e^{-x_i w_{ij}}, & x_i w_{ij} \geq 0 \\ 1, & x_i w_{ij} < 0 \end{cases}. \tag{20}$$

Weight w_{ij} acts as a scaling factor of $f(.)$. Taking the α-cut for $\alpha = 0.999$, we can approximate (20) to

$$f(x_i) = \begin{cases} e^{-x_i w_{ij}}, & x_i \geq \frac{0.001}{w_{ij}} \\ 1, & x_i < \frac{0.001}{w_{ij}} \end{cases} \tag{21}$$

where $f(0.001/w_{ij}) = 0.999 \approx 1$.

The linguistic interpretation of this new set fuzzy is "smaller than $0.001/w_{ij}$" and it will only make sense if $0 \leq 0.001/w_{ij} \leq 1$, which leads to the need that $w_{ij} \geq 0.001$.

This is related with the usual practice of training an ANN with normalized inputs; therefore all memberships functions extracted have to be defined for the respective input interval.

With $w_{ij} \geq 0.001$, $0 \leq \mathrm{x_i} \leq 1$ and $\theta \geq 0$, the correct use of (8) is guaranteed since we will always have $\sum_{i=1}^{n} x_i w_{ij} \geq 0$ and $x_i w_{ij} \geq 0$. However, during the training of the neural net the bias values can assume any value in $[-\infty\, +\infty]$. To overcome this problem, in the next section we show how to enforce the constraints $w_{ij} \geq 0.001$ and $\theta_j \geq 0$ in such a way that we can extract rules from the neural network as presented above.

F. Constrained Neural Network

In Section IV-C, we have seen that if we have $\sum_{i=1}^{n} x_i w_{ij} \geq 0$ and $x_i w_{ij} \geq 0$ we will extract rules as in (14), and in Section IV-E that, if $w_{ij} \geq 0.001$ and $\theta_j \geq 0$, we will always extract rules that may make sense. To take in account the constraints $w_{ij} \geq 0.001$ and $\theta_j \geq 0$, let us transform the weights and bias of (1) using the exponential function

$$s_j = f\left(\sum_{i=1}^{n} x_i(0.001 + e^{w_{ij}}) + e^{\theta_j}\right). \qquad (22)$$

Using this transformation, the new weight $w'_{ij} = 0.001 + e^{w_{ij}}$ will always be greater than 0.001 and the new bias $\theta'_j = e^{\theta_j}$ will be greater than zero. These transformations will not change the backpropagation algorithm commonly used for training a neural network. The algorithm will adjust normally w_{ij} and θ_j between $[-\infty\ +\infty]$ and the restrictions will be guaranteed through the exponentiation of w_{ij} and θ_j.

G. Extraction of a Transparent Fuzzy System

The interpretation of a fuzzy system is possible only if it satisfies conditions of transparency. According to [19], a FIS is transparent if all rules in the rule base are transparent. A rule is considered transparent if at firing strength

$$v_j = \bigcap_{i=1}^{n} \mu_{ij}(x_i) = 1 \qquad (23)$$

the system output is

$$y = y_j \qquad (24)$$

where y_j is the centre of the output membership associated with the rule. For the case of zero-order Takagi-Sugeno models, y_j is equal to the constant consequent. This means that the effect of a single rule may be isolated.

The transparency conditions are defined based on the overlapping degree of the input membership functions. For a fuzzy system to be transparent, the overlapping of the input membership functions must be smaller than 50%. This guarantees the existence of transparency checkpoints, which are points in the input-output space where an explicit contribution of a given rule takes place and where it is fully activated. Fig. 5 gives an example of a transparent fuzzy system, where the asterisks denote transparency checkpoints.

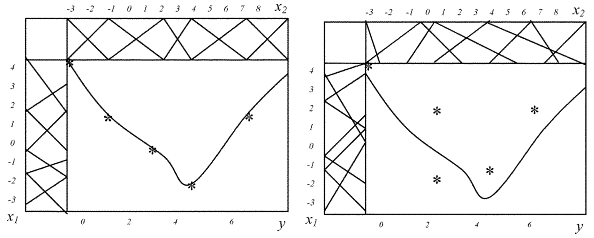
-3 -2 -1 0 1 2 3 4 5 6 7 8 x_2
4 3 2 1 0 -1 -2 -3
x_1
0 2 4 6 y
-3 -2 -1 0 1 2 3 4 5 6 7 8 x_2
4 3 2 1 0 -1 -2 -3
x_1
0 2 4 6 y

Fig. 5. Left: transparent fuzzy system [19]. X, Y axis represent two variables and the fuzzy partition of their domains with triangular fuzzy sets. When a single rule is activated, with both variables at membership level 1, its output (line) coincides with a real case (asterisk). Right: nontransparent fuzzy system. The activation of a single rule does not generate meaningful answers and interpolation of rules is necessary.

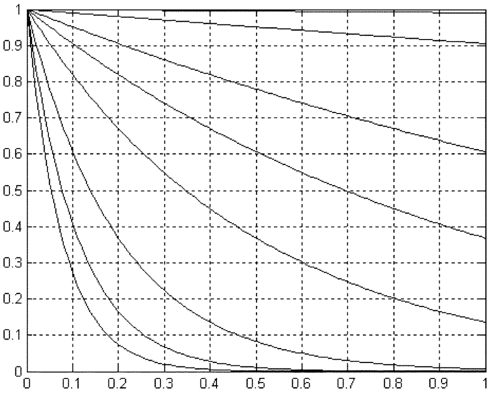

Fig. 6. Example of membership extracted for one input. X axis: normalized input values. Y axis: membership values.

If the overlapping is greater than 50%, however, at least two rules contribute simultaneously for any given input, thus the output is always the result of interpolation. This makes the contribution of a given rule invisible in system output and the fuzzy system cannot be considered transparent. Fig. 5 gives also an illustration of a nontransparent fuzzy system.

Although the methodology presented so far in this paper provided the extraction of an interpretable rule-based system, the rule base obtained cannot be considered transparent. This can be easy verified, e.g., by considering an ANN with seven hidden neurons. For each input, seven memberships will be extracted like the ones showed in Fig. 6. The extracted rule-based system will have at least two rules activated simultaneously, which leads to the case of nontransparent fuzzy system.

In order to provide the desired transparency for the extracted rule-based system, an approximation process needs to be carried out on all membership extracted from the ANN. In this work, this process is performed by using a combination of the five membership functions shown in Fig. 7. The membership functions for "extremely small" and "very small" have been selected to maintain the accuracy of the result. This option is not usual but nothing prevents one to adopt it—in our case, we have verified that it was a necessary option.

The approximation given by each membership function is

$$\mu(x) = a_1\mu_{smll}(x) + a_2\mu_{med}(x) + a_3\mu_{high}(x) + a_4\mu_{verysmll}(x) + a_5\mu_{extsmll}(x) \qquad (25)$$

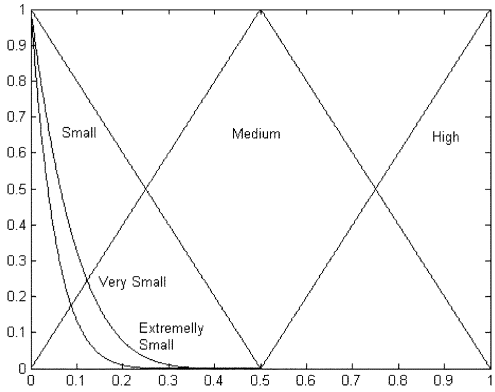

Fig. 7. New membership functions for each input.

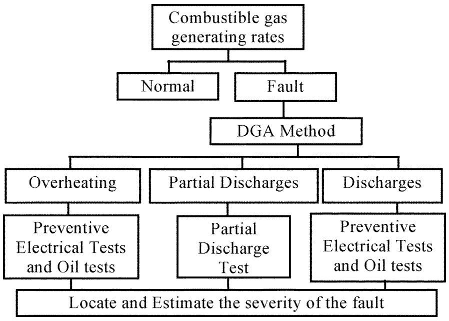
Combustible gas generating rates
Normal
Fault
DGA Method
Overheating
Partial Discharges
Discharges
Preventive Electrical Tests and Oil tests
Partial Discharge Test
Preventive Electrical Tests and Oil tests
Locate and Estimate the severity of the fault

Fig. 8. Transformer fault diagnosis flowchart.

where $x \in [0, 1]$ and $[a_1, a_2, \ldots a_5]$ are parameters that have to be identified. In this work, the recursive least-squares algorithm is used for this task.

For each rule, the approximated membership functions for each input are then combined. As result, a new rule-based system with a total number of rules equal to 5^{n} is formed, where n is the number of inputs of the system. Verifying (23) and (24), this new extracted fuzzy system may be considered a transparent fuzzy system.

V. Transformer Fault Diagnosis

A. The Transformer Fault Diagnosis System Proposed

The detection of incipient faults on transformers follows, in general terms, the flow-chart presented in Fig. 8. The process begins with the observation of the evolution of rates of combustible gases that exceed "normal" quantities. If the evolution rate per day is greater than a determined level then the transformer is suspected to have an active internal fault. The possible fault is investigated by DGA methods.

After guessing the possible fault, in order to obtain confirmation and more detailed information, such as the location of the fault, other tests are needed. Many techniques for the detection of possible faults of transformer using the measurement of gases have been established. Table I presents the IEC 60 599 criteria, which is widely used by utilities to interpret the DGA [2]. In spite of all the criteria already developed, the search for a more reliable method using DGA is still a topic of interest in many utilities.

When applying an ANN to transformer incipient fault diagnosis, the diagnosis can be reduced to an association process of inputs (pattern gases concentration) and output (fault type) since it does not need a physical model. Neural networks are capable of acquiring experiences from training data and interpolate from it. However, for a proper training, the database has to be plentiful and consistent.

TABLE I
IEC 60599 Criteria for the Interpretation of DGA Method

Case	Characteristic fault	$\frac{C_2H_2}{C_2H_4}$	$\frac{CH_4}{H_2}$	$\frac{C_2H_4}{C_2H_6}$
PD	Partial discharge	NS	<0.1	<0.2
D1	Discharges of low energy	>1	0.1-0.5	>1
D2	Discharges of high energy	0.6-2.5	0.1-1	>2
T1	Thermal fault T < 300 ^{0}C	NS	>1 , NS	<1
T2	Therm. f. 300 <T < 700 ^{0}C	<0.1	>1	1-4
T3	Thermal fault T > 700 ^{0}C	<0.2	>1	>4

NS = Non-significant whatever the value

TABLE II
Classification Results

	TR %	***T1%***
ANN	100	97.84 (3 errors)
IEC 599	94.86 (1 error and 14 *NI**)	94.96 (7 *NI*)

TR% - percentage of correct diagnosis for the training set
T1% - percent of correct diagnosis for the testing set
NI - non identified fault

In our work, we trained a neural network to receive as input data the percentage of concentration of the gases Hydrogen $(\mathrm{H_2})$, Ethylene $(\mathrm{C_2H_4})$, and Acetylene $(\mathrm{C_2H_2})$ and then classify the transformer fault as discharges, partial discharges or thermal fault. Other data could have been used, such as other gas concentrations or the gradients or increasing rates of concentration of key gases but, as far as the classification system is concerned, these three gases were enough as input for the ANN to give good classification results.

The database of faulty equipment inspected in service, used in Publication IEC 60 599 [2], [20], was used for training the ANN. Additionally, a database derived from the literature and a database obtained from CELPA (Power Stations of Pará, SA-Brazil) were also used in the ANN training.

The ANN was trained with a common backpropagation algorithm. It had 25 hidden neurons and three normalized inputs for gas concentration: $\mathrm{C_2H_2}$, $\mathrm{C_2H_4}$, and $\mathrm{H_2}$, and one output neuron. The output of the ANN is a real value. The target values defined, discriminating the type of fault, were: 0—thermal fault, 1—partial discharge, and 2—discharge fault. Around each value, a crisp band has been defined of size ±0.5, and an output result falling in a band was labeled according to its central value.

We have used 292 training and 139 testing patterns. The average square error (MSE) for training patterns was 0.0054 and for testing was 0.017. The control of the convergence of the ANN has taken in account not only the MSE but also the actual success in classification. In fact, the ANN performs an interpolation and the real objective is not so much reducing MSE but achieving accurate classification results.

B. ANN Results

Table II shows the results of the ANN trained with the restrictions necessary for the rule extraction process, namely $w_{ij} \geq 0.001$ and $\theta \geq 0$.

The result presented corresponds to the best one after some training. The table also shows, for comparison, the diagnosis results using the IEC criteria. One may observe that the IEC

TABLE III
TEST RESULTS

H_2	CH_4	C_2H_2	C_2H_4	C_2H_6	ANN	IEC	I
13	17	0.3	4.7	4.2	T	T	T
39	1.7	0.1	0.1	0.6	PD	PD	PD
1600	3600	0	14	670	PD	NI	PD
95	10	39	11	0	D	NI	D
1570	1110	1830	1780	175	D	D	D
41	112	4536	254	0	D	T	D
835	76	16	10	29	T	NI	D
10	13	0.1	25	7	T	T	T
33046	619	0	2	58	PD	PD	PD

T=Thermal Fault, PD=Partial Discharge, D=Discharge, NI=not identified
ANN= Classification with ANN, IEC= Classification according to IEC 60599
I= results of inspection — All values of gases are in ppm

method fails to identify a certain percentage of faults, but these have all been correctly classified by the ANN. Also, the IEC method is not exempt of error and ANN provided a more reliable classification of faults. Table III shows a sample of the results for the testing data, each line representing a case. The reader will notice some cases where the ANN made a correct identification while the IEC table failed, and one case of failure of the ANN where also the IEC table was not useful.

C. Rule Extraction From the Neural Network

Once the ANN trained, the process of extraction of rules from the ANN could be initiated.

The extracted FIS has, as the ANN, three normalized inputs (percentage of gases concentration: C_2H_2, C_2H_4, and CH_4) and one output (fault); as the trained ANN has 25 neurons in its hidden layer, 25 rules were extracted. Each rule extracted is expressed as

$$\begin{aligned} R_i : &\ \mathbf{IF}\ (\mathrm{C_2H_2 \text{ is smaller than }} a) AND \\ &(\mathrm{C_2H_4 \text{ is smaller than }} b) AND (\mathrm{H_2 \text{ is smaller than }} c) \\ &\mathbf{then}\ y_i = d. \end{aligned}$$

To guarantee the transparency of the fuzzy systems, all membership values extracted from the ANN were approximated by the combination of the five membership functions shown in Fig. 5. With this combination, the number of the rules is $5^3 = 125$, and each input has five membership values associated.

After membership approximation (as in Section IV-G), the new average squared error of the system for training patterns is 0.0056 and for testing pattern is 0.0201. By comparison with the results obtained by the ANN, we may conclude that the fidelity between the ANN and the extracted FIS is guaranteed.

Examples of the approximate rules are

$$\begin{aligned} R_1 : &\ \mathbf{IF}\ (\mathrm{H_2 \text{ is Small}})\ AND\ (\mathrm{C_2H_2 \text{ is Small}}) \\ &AND\ (\mathrm{C_2H_4 \text{ is small}})\ \mathbf{THEN}\ y_1 = 3.89 \\ R_2 : &\ \mathbf{IF}\ (\mathrm{H_2 \text{ is Small}})\ AND\ (\mathrm{C_2H_2 \text{ is Small}}) \\ &AND\ (\mathrm{C_2H_4 \text{ is Medium}})\ \mathbf{THEN}\ y_2 = 4.53 \\ R_3 : &\ \mathbf{IF}\ (\mathrm{H_2 \text{ is Small}})\ AND\ (\mathrm{C_2H_2 \text{ is Small}}) \\ &AND\ (\mathrm{C_2H_4 \text{ is High}})\ \mathbf{THEN}\ y_3 = 4.38 \\ R_4 : &\ \mathbf{IF}\ (\mathrm{H_2 \text{ is Small}})\ AND\ (\mathrm{C_2H_2 \text{ is Small}}) \\ &AND\ (\mathrm{C_2H_4 \text{ is Extremely Small}})\ \mathbf{THEN}\ y_4 = -0.19 \\ &\ldots \\ R_{125} : &\ \mathbf{IF}\ (\mathrm{H_2 \text{ is Small}})\ AND\ (\mathrm{C_2H_2 \text{ is Very Small}}) \\ &AND\ (\mathrm{C_2H_4 \text{ is Very Small}})\ \mathbf{THEN}\ y_{125} = 0.002. \end{aligned}$$

The output of the fuzzy system is given by

$$y = 4.226\,888 - \sum_{j=1}^{125} v_j y_j. \tag{26}$$

If we take in account the FIS default value, we can uncover the meaning of a rule. For instance, take rule R_3: if it could be fired isolated, in a condition where its antecedent would have membership value of 1, the FIS output would be of $\mathrm{y} = 0.15$, clearly indicating a thermal fault (value close to 0). This conclusion, by the way, is not in contradiction with IEC 60 599 criteria in Table I. What is surprising is that we could reach better quality results with less information (using a smaller number of gases).

The new rules extracted may now be subject to examination by experts. Because transparency is assured, an expert may examine the individual merits of each rule. Two consequences may derive: the rule may match previous expert knowledge, and therefore the soundness of the action of the ANN/FIS is confirmed; or the rule represents new knowledge, or a new way of presenting it, and therefore we face a knowledge discovery moment. In any case, the process allows one to make explicit the hidden knowledge captured by an ANN.

VI. CONCLUSION

This paper has been inspired by the problem of early diagnosis of transformer incipient faults by dissolved gas analysis. Although there are guidelines from IEC (rules organized in a table) in order to help in classifying faults, many cases are still subject to doubt and error and, therefore, this problem is still a concern for utilities and manufacturers.

We have shown that ANNs could give good results in such task. In fact, the results in the paper demonstrate that in 431 cases analyzed, from a diversity of origins, we had only three errors in classification, compared to 22 cases nonclassified or wrongly classified by applying IEC 60 599 recommendation.

But we have provided an answer to another concern: the black box characteristic of an ANN. In fact, the paper described how to build a mathematical transform for an ANN and represent it by an equivalent zero order TS-FIS, with explicit rules of the IF-THEN type. The model presented, inspired in the f-duality concept developed by other researchers, evolves in a different direction and represents a further advance because it allows a clear and classic linguistic representation of rules with the use of the AND connective.

Once rules are available, knowledge is made explicit. However, its interpretability can only be fully achieved if the rule base exhibits the property of transparency, allowing the inspection of the merits of each rule individually. The paper presents a solution to this aspect.

The application of the method to the transformer fault diagnosis problem showed that not only an ANN could be used to produce better results than IEC standard method but also that rules governing fault classification could be extracted from the ANN. We have not shown that the TS-FIS generated from the ANN gives exactly the same results as the ANN itself, because this is an obvious result from the fact that they are mathematically equivalent. We have, however, shown that the FIS with a transparent fuzzy rule base is a very good approximation to the ANN.

The work with transformer fault diagnosis is not complete. In fact, we believe that this paper may open a new path of research and eventually allow the development of better and more accurate diagnosis systems. In real-world applications, one should not rely on a single technique, and a clever blend usually is the best engineering solution.

However, the work reported opens a new way to knowledge discovery. One may now follow the procedure of first, training an ANN on a problem, then using our f-duality transform to convert it into a fuzzy inference system, apply the described technique to generate transparent rules and subject these rules to expert examination in order to confirm established knowledge and discover new knowledge.

Appendix

Proof of Lemma 2

Only two input variables in domain of X will be considered to prove Lemma 2.

Let $a, b \in [0, 1[$. Let $x_1, x_2 \in \Re$ such that $a = f(x_1)$ and $b = f(x_2)$. For the positive-sigmoid function defined in (6), we have then

$$\begin{aligned}
\text{For}\quad & x = x_1, x_1 = -\ln(1 - f(x_1)) = -\ln(1-a)\\
& x = x_2, x_2 = -\ln(1 - f(x_2)) - = -\ln(1-b)\\
\text{then}\quad & x_1 + x_2 = -\ln(1-a) - \ln(1-b)\\
& \qquad\quad = -\ln(1-a)(1-b)\\
\text{For}\quad & x = x_1 + x_2, \quad x_1 + x_2 = -\ln(1 - f(x_1 + x_2))\\
\text{then}\quad & -\ln(1-a)(1-b) = -\ln(1 - f(x_1 + x_2))\\
& \qquad (1-a)(1-b) = 1 - f(x_1 + x_2)\\
\text{and}\quad & f(x_1 + x_2) = 1 - (1-a)(1-b)\\
\text{thus}\quad & f(x_1 + x_2) = f(x_1) * f(x_2) = a * b = 1 - (1-a)(1-b)
\end{aligned}$$

and generalizing for n inputs, we have proved Lemma 2.

Acknowledgment

The authors are grateful to GRICES (Portugal) and CAPES (Brazil) for the support brought to the NESC/INESC cooperation protocol, which allowed one of the authors, A. R. G. Castro, to develop the research reported.

References

[1] IEEE Guide of Gases Generated in Oil-Immersed Transformer, IEEE Power Engineering Society, 1992.

[2] Interpretation of the Analysis of Gases in Transformers and Other Oil-Filled Electrical Equipment in Service, Mar. 1999.

[3] K. Tomsovic, M. Tapper, and T. Ingvarsson, "A fuzzy information approach to integrating different transformer diagnostic methods," *IEEE Trans. Power Del.*, vol. 8, no. 3, pp. 1638–1644, Jul. 1993.

[4] Y.-C. Huang, "Evolving neural nets for fault diagnosis of power transformer," *IEEE Trans. Power Del.*, vol. 18, no. 3, pp. 843–848, Jul. 2003.

[5] Y. Zhang, X. Ding, Y. Liu, and P. J. Griffin, "An artificial neural approach to transformer fault diagnosis," *IEEE Trans. Power Del.*, vol. 11, no. 4, pp. 1836–1841, Oct. 1996.

[6] A. Castro and V. Miranda, "Mapping neural networks into rule sets and making their hidden knowledge explicit—application to spatial load forecasting," in *Proc. PSCC02—14th Power Systems Computation Conf.*, Sevilla, Spain, Jun. 2002.

[7] K. Hornik, M. Stincchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Network*, vol. 2, pp. 359–366, 1989.

[8] R. Andrews, J. Diederich, and A. B. Tickle, "A survey and critique of techniques for extracting rules from trained artificial neural networks," *Knowledge-Based Syst.*, vol. 8, no. 6, pp. 373–389, 1995.

[9] C. MacMillan, M. C. Mozer, and P. Smolensky, "The connectionist scientist game: rule extraction and refinement in a neural network," in *Proc. 13th Annu. Conf. Cognitive Science Society*, Hillsdale, NJ, 1991.

[10] R. Andrews and S. Geva, "Rule extraction from a constrained error backpropagation MLP," in *Proc. 5th Australian Conf. Neural Networks*, Brisbane, Qld., Australia, 1994, pp. 9–12.

[11] R. Setiono, "Extracting M-of-N rules from trained neural networks," *IEEE Trans. Neural Netw.*, vol. 11, no. 2, pp. 512–519, Mar. 2000.

[12] R. Setiono, W. K. Leow, and J. M. Zurada, "Extraction of rules from artificial neural networks for nonlinear regression," *IEEE Trans. Neural Netw.*, vol. 13, no. 3, pp. 564–577, May 2002.

[13] K. Saito and R. Nakano, "Medical diagnostic expert system based on PDP model," in *Proc. IEEE Int. Conf. Neural Networks*, vol. 1, San Diego, CA, 1988, pp. 255–262.

[14] S. Thrun, "Extracting rules from artificial neural networks with distributed representations," in *Advances in Neural Information Processing Systems 7*, G. Tesauro, D. S. Touretzky, and T. Leen, Eds. Cambridge, MA: MIT Press, 1995.

[15] G. P. J. Schmitz *et al.*, "ANN-DT: an algorithm for extraction of decision trees from artificial neural networks," *IEEE Trans. Neural Netw.*, vol. 10, no. 6, pp. 1392–1401, Nov. 1999.

[16] Z.-H. Zhou *et al.*, "Extracting symbolic rules from trained neural networks ensembles," *AI Commun.*, vol. 16, no. 1, pp. 3–5, 2003.

[17] 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, Sept. 1997.

[18] L.-X. Wang, *A Course in Fuzzy Systems and Control*. Englewood Cliffs, NJ: Prentice-Hall, 1997.

[19] A. Riid and E. Rustern, "Transparent fuzzy systems and modeling with transparency protection," in *Proc. IFAC Symp. Artificial Intelligence in Real Time Control*, Oct. 2000, pp. 229–235.

[20] M. Duval and A. Pablo, "Interpretation of gas-in-oil analysis using new IEC publication 60 599 and IEC TC10 databases," *IEEE Elect. Insul. Mag.*, vol. 17, no. 2, pp. 31–41, Mar./Apr. 2001.

Adriana Rosa Garcez Castro graduated and received the M.Sc. degree in electrical engineering from the Federal University of Pará (UFPA), Pará, Brazil, in 1992 and 1995, respectively, and the Ph.D. degree from INESC Porto and the Faculty of Engineering, University of Porto (FEUP), Porto, Portugal, in 2004.

She is currently a Lecturer at UFPA. Her research interests are in power systems and control and the application of computational intelligence techniques.

Vladimiro Miranda graduated and received the Ph.D. and Agregado degrees from the Faculty of Engineering, University of Porto, Portugal (FEUP), Porto, Portuga, in 1977, 1982, and 1991, respectively, all in electrical engineering.

In 1981, he joined FEUP and currently holds the position of Professor Catedrático. He is also currently Director of INESC Porto. He has authored many papers and been responsible for many projects in areas related with the application of computational intelligence to power systems.