Are Artificial Neural Networks Black Boxes?

J. M. Benítez, J. L. Castro, and I. Requena

***Abstract*— Artificial neural networks are efficient computing models which have shown their strengths in solving hard problems in artificial intelligence. They have also been shown to be universal approximators. Notwithstanding, one of the major criticisms is their being black boxes, since no satisfactory explanation of their behavior has been offered. In this paper, we provide such an interpretation of neural networks so that they will no longer be seen as black boxes. This is stated after establishing the equality between a certain class of neural nets and fuzzy rule-based systems. This interpretation is built with fuzzy rules using a new fuzzy logic operator which is defined after introducing the concept of f-duality. In addition, this interpretation offers an automated knowledge acquisition procedure.**

***Index Terms*— Equality between neural nets and fuzzy rule-based systems, f-duality, fuzzy additive systems, interpretation of neural nets, i-or operator.**

I. Introduction

ARTIFICIAL neural networks (ANN's) are well-known massively parallel computing models which have exhibited excellent behavior in the resolution of complex artificial intelligence problems. However, many researchers refuse to use them because of their shortcoming of being "black boxes," that is, determining why an ANN makes a particular decision is a difficult task. This is a significant weakness, for without the ability to produce comprehensible decisions, it is hard to trust the reliability of networks addressing real-world problems.

On the other hand, fuzzy rule-based systems (FRBS's), developed using fuzzy logic, have become a field of active research during the last few years. These algorithms have proved their strengths in tasks such as the control of complex systems, producing fuzzy control. But fuzzy set theory also provides an excellent way of modeling knowledge.

The relation between both worlds (ANN's and FRBS's) has been extensively studied. Indeed, this is a close relation since equivalence results have been obtained. However, all of these results are approximative. In this paper, we go one step further by establishing not just the equivalence but the equality between some kinds of ANN's and FRBS's that use comprehensible fuzzy rules. This connection yields two immediate and important conclusions. First, we can apply what has been discovered for one of the models to the other. Second, we can translate the knowledge embedded in the neural network into a more cognitively acceptable language, fuzzy rules. In other words, we obtain an understandable interpretation of neural nets. We are mainly concerned in explaining how an ANN works, not just using a neural net as a tool to translate the knowledge underlying a data set into rules. Nevertheless, the interpretation leads to a method for extracting rules from an ANN. Rule-extraction from ANN's constitutes a large and growing area of research [1]–[5].

This paper is organized as follows. Section II is devoted to describing the kind of ANN's we consider, and Section III introduces FRBS's. Next, we present the main (equality) result linking both models. In Section V, the concept of f-duality and some immediate properties are presented. This is a crucial concept on which we define the interactive-or operator, which enables us to offer an interpretation of ANN's based on fuzzy rules, as explained in Section VI. This interpretation is illustrated with a simple sample. We finish the paper by discussing conclusions and stating final remarks. Proofs of results presented throughout the paper are found in the Appendix.

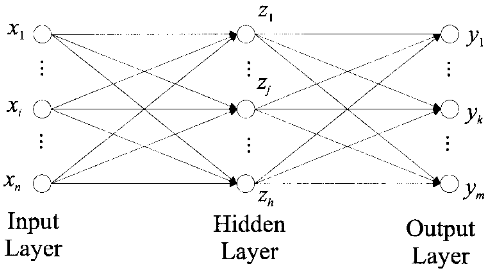

Fig. 1. Multilayer neural network.

II. Artificial Neural Networks

Many different models of neural nets have been proposed [6], [7], however, multilayered feedforward ANN's are especially interesting since they are the most common. In these nets, neurons are arranged in layers and there are only connections between neurons in one layer to the following. Let us consider an ANN with input, hidden, and output layers. Let us suppose that the net has n input neurons $(x_1, \cdots, x_n)$, h hidden neurons $(z_1, \cdots, z_h)$, and m output neurons $(y_1, \cdots, y_m)$. Let τ_j be the bias for neuron z_j. Let w_{ij} be the weight of the connection from neuron x_i to neuron z_j and, β_{jk} the weight of the connection from neuron z_j to neuron y_k. Fig. 1 shows the general layout of these nets. The

Manuscript received February 2, 1996; revised August 12, 1996 and March 8, 1997.

The authors are with the Department of Computer Science and Artificial Intelligence, E.T.S. Ingeniería Informática., University of Granada, 18071 Granada, Spain.

Publisher Item Identifier S 1045-9227(97)05241-7.

1045–9227/97$10.00 © 1997 IEEE

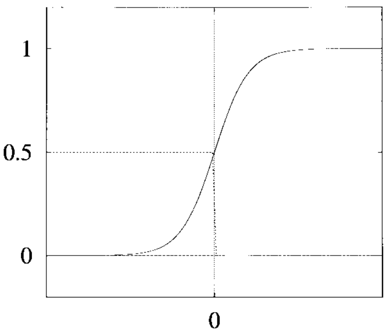

Fig. 2. Logistic function $f_A(x) = 1/(1+e^{-x})$.

function the net calculates is

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

$$y_k = g_A\left(\sum_{j=1}^{h} z_j \beta_{jk}\right)$$

with

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

where g_A and f_A are activation functions, which are usually continuous, bounded, nondecreasing, nonlinear functions. The usual choice is the logistic function: $f_A(x) = 1/(1+e^{-x})$, whose graph is shown in Fig. 2. Linear functions, even the identity, are also common for g_A.

One of the most interesting properties of the ANN's is that they are universal approximators, that is, they can approximate to any desired degree of accuracy any real-valued continuous function or one with a countable number of discontinuities between two compact sets. In [8], this result is shown for feedforward ANN's with a single hidden layer and a sufficient number of units in it whose activation function is continuous and nonlinear, and taking the identity as activation function for output units. This result establishes that we can successfully use ANN's for learning and faithfully reproducing a system's behavior from a sufficiently large set of samples. The result has increased the already widespread use of ANN's.

However, many people refuse to use ANN's because of their most criticized feature: there is no satisfactory interpretation of their behavior. ANN's are devices working as black boxes: they capture "hidden" relations between inputs and outputs with a highly accurate approximation, but no definitive answer is offered for the question of how they work.

III. Fuzzy Rule-Based Systems

Rules, in general, represent in a natural way causality relationships between inputs and outputs of a system, corresponding to the usual linguistic construction "IF *a set of conditions is satisfied,* THEN *a set of consequences is inferred.*" Fuzzy logic [9], [10] provides a natural tool to model and process uncertainty, hence, fuzzy rules have the additional advantage over classical production rules of allowing a suitable management of vague and uncertain knowledge. They represent knowledge using linguistic labels instead of numeric values, thus, they are more understandable for humans and may be easily interpreted.

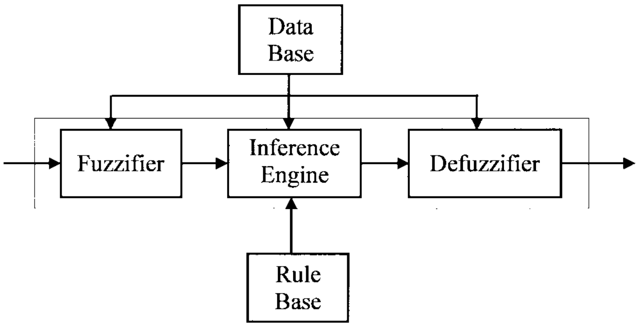

Fig. 3. Structure of a fuzzy rule-based system.

Systems using fuzzy rules are termed FRBS's [11], [12]. These systems map n-dimensional spaces into m-dimensional spaces. As depicted in Fig. 3, they are composed of four parts: fuzzifier, knowledge base, inference engine, and defuzzifier.

The fuzzifier converts real valued inputs into fuzzy values (usually, in singleton fuzzy sets).

The knowledge base includes the fuzzy rule base and the database. Fuzzy rules have the form

$$\begin{aligned} R_i&: \text{If } x_1 \text{ is } A_1^i \text{ and } x_2 \text{ is } A_2^i \text{ and } \cdots \text{ and } x_n \text{ is } A_n^i \\ &\quad \text{then } y \text{ is } B_i \end{aligned}$$

where $x_1, \cdots, x_n$ are the inputs, y is the output and $A_1^i, \cdots, A_n^i$, and B_i are linguistic labels. Membership functions of these linguistic terms are contained in the database.

The inference engine calculates fuzzy output from fuzzy inputs by applying a fuzzy implication function. Finally, the defuzzifier yields a real-value output from the inferred fuzzy output.

The most successful application of FRBS's is fuzzy control, which is devoted to the management of complex systems, which are very hard to model using classical mathematics. Yet they have some additional interesting properties, the most important being universal approximation. In [13]–[15], it is demonstrated that wide classes of fuzzy controllers are universal approximators. One of these is the class of fuzzy additive systems (FAS's) [15], which employ rules with the following expression, known as TSK rules [16]:

$$\begin{aligned} R_{jk}&: \text{If } x_1 \text{ is } A_{jk}^1 \text{ and } x_2 \text{ is } A_{jk}^2 \text{ and } \cdots \text{ and } x_n \text{ is } A_{jk}^n \\ &\quad \text{then } y_k \text{ is } p_{jk}(x_1, \cdots, x_n) \end{aligned}$$

where $p_{jk}(x_1, \cdots, x_n)$ is a linear function on the inputs. In FAS's, the inference engine works as follows: For each rule, the fuzzified inputs are matched against the corresponding antecedents in the premises giving the rule's firing strength (or weight). It is obtained as the t-norm (usually the minimum operator) of the membership degrees on the rule if-part. The overall value for output y_k is calculated as the weighted sum of relevant rule outputs. Let us suppose a system with n inputs, m outputs, and multiinput single output (MISO) fuzzy rules, having l_k of them for kth output. Then y_k is computed as

$$y_k = \sum_{j=1}^{l_k} v_{jk} \cdot p_{jk}(x_1, \cdots, x_n)$$

where v_{jk} is the firing strength of jth rule for kth output. FAS's are very similar to the more common Sugeno systems. The only difference between them is the way they obtain the output: while FAS's output is a single weighted sum, Sugeno systems' output is a weighted average.

When the activation functions in an ANN are continuous, the function the net calculates is continuous. So an ANN can be approximated by an FRBS. And conversely, an ANN can approximate a continuous FRBS. Thus, it is easy to check that the ANN's and FRBS's we have described are equivalent. We study this relation in depth in the following section.

IV. Artificial Neural Networks Are Fuzzy Rule-Based Systems

The equivalence between ANN's and FRBS's has been studied by different authors [17], [18]. Most of their results establish the equivalence through an approximation process. But this process works in such a way that if a sufficiently high degree of accuracy is required, the number of rules needed to approximate the ANN or the number of neurons needed to approximate the FRBS becomes too high. Hence, it is just a purely theoretical solution.

However, Jang and Sun's work [18] is different. They give an equivalence result between radial basis function networks and fuzzy additive systems. In fact, it is an equality relation that only requires a finite number of rules or neurons. Here, considering a different and more frequently used ANN model, we provide a similar result.

In this paper, we show that building an FRBS that calculates *exactly the same function* as an ANN like those employed by Hornik *et al.* [8], not just approximated but the same, is trivial. We need no results requiring an unbounded number of rules. The basic ideas were already indicated in [19].

Theorem 1: Let N be a three-layer feedforward neural network with a logistic activation function in hidden neurons and identity in output neurons. Then there exists a fuzzy additive system that calculates the same function as the net does.

Proof: We provide a constructive proof. To describe the fuzzy system we only need to give the rule base. We employ TSK type rules. A fuzzy rule R_{jk} per pair of neurons (hidden, output), (z_j, y_k), is added

$$R_{jk}\text{: If } \sum_{i=1}^{n} x_i w_{ij} + \tau_j \text{ is } A \text{ then } y_k = \beta_{jk} \qquad (1)$$

where A is a fuzzy set on $\mathbb{R}$ whose membership function is simply the activation function of hidden neurons f_A.

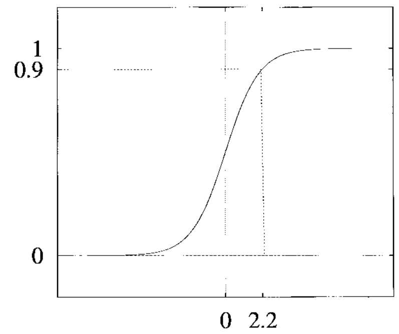

Fig. 4. Membership function for "approximately greater than 2.2."

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

$$y_k = \sum_{j=1}^{h} A\left(\sum_{i=1}^{n} x_i w_{ij} + \tau_j\right) \cdot \beta_{jk}. \qquad (2)$$

By (1) and (2), one can easily check that each output y_k of the fuzzy system is exactly the same as the corresponding output for the network. ■

A couple of remarks can be noted. First, we use simple rules "If z is A, then $y = v$" where $v \in \mathbb{R}$ and z are new variables obtained by a single change of variable in the n inputs. Second, the A fuzzy set may be understood as "greater than approximately r," where r is a positive real number obtained from a preestablished α-cut. Since the logistic function can reach zero and one only asymptotically, the usual convention in neural literature is to consider activation levels of 0.1 and 0.9, for total absence of activation and full activation, respectively. Thus we can set an α-cut for $\alpha = 0.9$ and interpret the fuzzy set with f_A as membership function (see Fig. 4) as "greater than approximately 2.2" since $f_A(2.2) = 0.9$.

By considering the stated fuzzy rule set, the inference method, and the identity function for g_A, we have shown the existence of a system based on fuzzy rules that calculates exactly the same as a neural net with a total approximation degree. This is the theoretical aspect of the result: since the equivalence-by-approximation was already proven we have established the *equality* between ANN's and FRBS's.

Two important conclusions may be drawn from the equivalence between ANN's and FRBS's. On the one hand, everything discovered for one of the models may be applied to the other. On the other hand, the knowledge an ANN encodes into its synaptic weights may be expressed in a more comprehensible form for humans, thus making it more transparent and easier to interpret. This leads to a couple of interesting ideas, an interpretation of ANN's is possible, as well as a method for automatic knowledge acquisition.

The rules used in the system are fuzzy ones. They have a clear meaning for a mathematician. Nevertheless, we intend to go one step further and provide fuzzy rules that can be easily interpreted. Hence, we proceed to find such a decomposition of the premise of rules so that we might rewrite them as

$$R_{jk}\text{: If } x_1 \text{ is } A^1_{jk}\,\theta\, x_2 \text{ is } A^2_{jk}\,\theta \cdots \theta\, x_n \text{ is } A^n_{jk} \text{ then } y_k = \beta_{jk} \tag{3}$$

where θ is a logic connective and A^i_{jk} are fuzzy sets obtained from A, the weights, w_{ij}, and the biases τ_j.

The first approach would be to try a decomposition of the obtained rules to the most common and simple fuzzy rule type, that is, rules with AND joining the antecedents. Clearly, this should be the ideal situation, but this decomposition is not possible. It is easy to check the negative answer. Fuzzy rules may be seen as fuzzy relations. The decomposition we are trying means that the relation corresponding to each rule (1) must be expressed as a composition of relations. But Yager [20] showed that this is not possible in every case. As to the OR connective, a similar problem arises. So a different type of connective must be considered.

The solution comes from the concept of f-duality, which we define in the next section. This concept implies very powerful representation properties and leads to a fuzzy logic operator that enables us to write more comprehensible and suitable fuzzy rules to describe the knowledge embedded in an artificial neural network.

V. f-Duality and Interactive-or

We now introduce the concept of f-duality, which will be used to define a logical operator that enables us to give a proper interpretation of ANN.

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

$$f(x_1 \oplus x_2) = f(x_1) \circledast f(x_2). \tag{4}$$

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

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

Now, let us consider the operation $+$ in $\mathbb{R}$ and the logistic function f_A. The latter is a bijective function from $\mathbb{R}$ to (0, 1). Thus we have the following.

Lemma 2: The f_A-dual of + is $*$, defined as

$$a * b = \frac{a \cdot b}{(1-a)\cdot(1-b) + a \cdot b}. \tag{5}$$

Definition 2: We call the operator defined in the previous lemma the **interactive-or** operator, i-or.

We proceed by studying some of the straightforward properties of the i-or operator in the following.

Lemma 3: Let $*$ be the f_A-dual of +. Then $*$ verifies the following.

1) $*$ is commutative.
2) $*$ is associative.
3) There exists a neutral element e for $*$. It is $e = \frac{1}{2}$.
4) Existence of inverse elements. $\forall a \in (0, 1)\ \exists_1 a' \in (0, 1)$ such that $a * a' = e$. $a' = 1 - a$.

Corollary 1: Let $*$ be the f_A-dual of +. Then $[(0, 1), *]$ is an abelian group.

Lemma 4: The f_A-dual of + extends easily to n arguments

$$a_1 * a_2 * \cdots * a_n = \frac{a_1 \cdot a_2 \cdot \cdots \cdot a_n}{(1-a_1)\cdot(1-a_2)\cdot \cdots \cdot(1-a_n) + a_1 \cdot a_2 \cdot \cdots \cdot a_n}. \tag{6}$$

By writing $a_1 * a_2 * \cdots * a_n$ instead of $*(a_1, a_2, \cdots, a_n)$ we abuse the notation, but this does not cause any problem as $*$ is both associative and commutative.

Lemma 5: The f_A-dual of +, $*$, verifies the following.

1) $\lim_{a_i \to 0} a_1 * a_2 * \cdots * a_n = 0 \quad \forall a_1, \cdots, a_n \in (0, 1) \quad \forall i \in \{1, \cdots, n\}$.
2) $\lim_{a_i \to 1} a_1 * a_2 * \cdots * a_n = 1 \quad \forall a_1, \cdots, a_n \in (0, 1) \quad \forall i \in \{1, \cdots, n\}$.
3) $*$ is strictly increasing in every argument.

From a purely mathematical point of view, the operator interactive-or has some interesting and undoubtedly elegant properties. This operator endows (0, 1) with a fine group structure having nice symmetries like $1/2$ being the neutral element and $1 - a$ being the inverse of a.

The operator we have just defined works on (0, 1) and hence, can be seen as a fuzzy logic operator, that is, an operator that takes several truth values corresponding to different propositions and produces an overall truth value.[1] This value is obtained as an aggregation whose result increases with the inputs.

But the most interesting point about interactive-or is that it has a natural interpretation. This meaningfulness is supported by the clear logical interpretation of properties stated in Lemmas 3 and 5. It can be best exposed through an example.

Let us consider the evaluation of a scientific paper. In order to decide whether accepting it for publication or not, the editor sends it to, say, two reviewers. After an in-depth examination, each one assigns the paper a number in (0, 1), reflecting his/her opinion as to whether to accept or reject the paper. The closer the number is to one, the better the opinion. The editor makes the final decision. This decision may be modeled using the interactive-or. Let us see why. Since both reviewers' criterion is equally good, both receive the same consideration, hence the symmetry. If one of the referees has no clear opinion, he will be neutral and will give a 0.5. The final decision would be made from the other referee's evaluation. Here, the role of the neutral element for i-or, namely 0.5, is obvious. In case the examinators' opinions are opposed, the more emphasized one, that is, the closer to one of the extremes, would prevail. This conforms to the behavior in limits as indicated in Lemma 5. However, if the opinions are opposed but of equal "strength," they would provide little help to the editor, who would be confused. In other words, the overall evaluation of the referees would be 0.5. This is explained by the inverse elements.

We can find, i-or applicability in many other situations: the quality of game developed by two tennis players in a double

[1] Obviously, we refer to a fuzzy logic in which membership values belong to (0, 1) instead of the more common [0, 1].

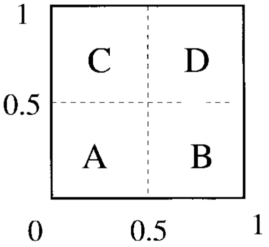

Fig. 5. Regions defining the behavior of i-or.

tennis match, the evaluation of students, the role of forward and goalkeeper in a soccer match, etc.

In addition, the i-or operator can model situations in which more than two elements are involved. It also covers, by using weights, cases where the importance of some factors is higher than others.

The interactive-or operator is an hybrid between both a t-norm and a t-conorm. To understand this, see Fig. 5, where the unit square is divided into four quadrangular regions: A, B, C, and D. The i-or works as a very conservative t-norm (i.e., lesser than most of them) in A region; it works as a very conservative t-conorm (i.e., greater than most of them) in region D, which is the dual of the t-norm in A; and in regions B and C a continuous transition from a t-norm to a t-conorm, or vice versa, takes place.

To give the reader a clearer idea of i-or behavior, we include Figs. 6 and 7. They represent the surfaces defined by i-or and min, respectively, when applied to two triangular fuzzy sets. The aspect of the surfaces suggests that when it comes to model nonlinear surfaces, i-or is much more suitable than linear minimum. We feel that the nonlinear behavior of i-or can be shaped to fit a nonlinear surface easier than min, where only the interpolation of several rules would provide the approximating behavior.

To end this section, we observe that, from Proposition 1, it is straightforward that many new operators may be obtained by considering different activation functions (f_A) or different aggregating operators (+). Properties of these new operators depend mainly on those from the originating aggregating operator and, in a second place, on the activation function. What you can take for granted is that whenever you have a bijective function you get the structure of the original set duplicated on the image set. Hence, you find in the image set operators with the same operations than those defined in the original set. This opens a new and interesting line of research, which is under development.

VI. Interpretation of Artificial Neural Networks

As proposed in Section IV, in order to improve the understandability of fuzzy rules, we have reformulated them by decomposing their premises. This section is devoted to explaining how this decomposition is achieved. To make the exposition easier to follow, we describe the interpretation in two steps. First we consider ANN's without biases. Then we indicate the role of biases.

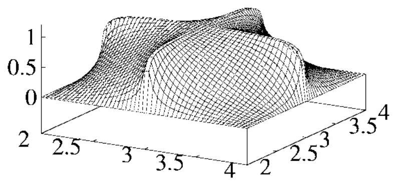

Fig. 6. Interactive-or operator.

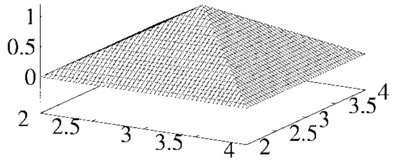

Fig. 7. Minimum operator.

A. Artificial Neural Networks Without Biases

In this section, we are concerned with ANN's without biases, thus rules of the corresponding FAS are a special case of (1) where $\tau_j = 0$. These rules are translated into rules of form (3), using the i-or operator as the logic connective θ. In effect, since the logistic is a bijective function, we can obtain the f_A-dual of +. This operation is $*$, as discussed in the previous section. The resulting rules are

$$R_{jk}\text{: If } x_1w_{1j} \text{ is } A * x_2w_{2j} \text{ is } A * \cdots * x_nw_{nj} \text{ is } A \text{ then } y_k = \beta_{jk}. \tag{7}$$

And accepting that "x_iw_{ij} is A" might be interpreted as "x_i is A/w_{ij}," then we can rewrite them as indicated in (3)

$$R_{jk}\text{: If } x_1 \text{ is } A^1_{jk} * x_2 \text{ is } A^2_{jk} * \cdots * x_n \text{ is } A^n_{jk} \text{ then } y_k = \beta_{jk}. \tag{8}$$

The A^i_{jk} are fuzzy sets obtained from A and w_{ij}. Their membership function is given by

$$\mu_{A^i_{jk}}(x) = \mu_A(xw_{ij}).$$

Then it is easy to check that

$$x_1 \text{ is } A^1_{jk} * x_2 \text{ is } A^2_{jk} * \cdots * x_n \text{ is } A^n_{jk} = \sum_{i=1}^{n} x_iw_{ij} \text{ is } A.$$

Obviously, their interpretation is built out of that of A and modified by the weights w_{ij}. If w_{ij} is positive, then "x_i is A^i_{jk}" can be read as "x_i is greater than approximately r/w_{ij}," where r is a positive real number obtained from a previously established α-cut (e.g., $\alpha = 0.9$). If w_{ij} is negative, then "x_i is A^i_{jk}" can be understood as "x_i is lower than approximately r/w_{ij}." As to the absolute weight value, when it increases the proposition becomes crisper and the reference point, r/w_{ij}, becomes smaller. The effect is converse when the absolute weight value decreases.

Moreover, when the weight is negative, the proposition may also be expressed in terms of a "greater-than" relationship.

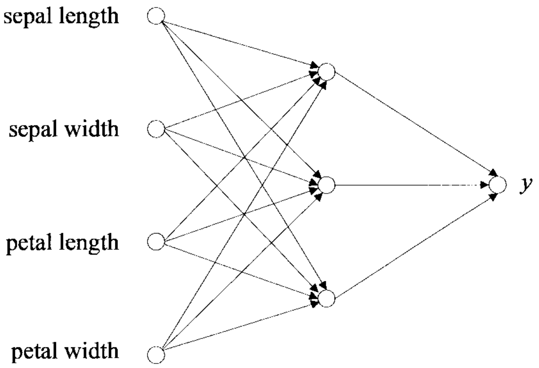

Fig. 8. A feedforward neural network for the iris plant problem.

This is obtained by writing the negated proposition, "not x_i is $\neg A^i_{jk}$," where $\neg A^i_{jk}$ is the complement of A^i_{jk}, whose membership function is

$$\mu_{\neg A^i_{jk}}(x) = 1 - \mu_{A^i_{jk}}(x)$$

It is easy to check that $\mu_{\neg A^i_{jk}}$ is a logistic function (see next Lemma 6). This way, "x_i is lower than approximately r" is the same that "x_i is not greater than approximately $-r$."

Lemma 6: Let $f(x) = 1/(1 + e^{-xw})$ then $f(-x) = 1 - f(x)$.

These rules provide a meaningful expression to knowledge extracted from examples by the neural net. Hence, they offer an interpretation of ANN's.

One of the most interesting properties of this interpretation is its independence from the learning algorithm. The training method employed to obtain the weights does not matter. The only relevant point is the knowledge the net has learned.

An Example: To illustrate the process, let us consider a simple example. This is the well-known iris plant problem, whose data set has been used in results described in many papers and can be found in a number of artificial intelligence repositories spread across the Internet. The goal is to recognize the type of the iris plant to which a given individual belongs. The data set is composed of 150 instances, equally distributed between the three classes: 50 for each of the three types of plants: setosa, versicolor, and virginica. One class is linearly separable from the other two; while the latter are not linearly separable from each other. Each instance features four attributes (petal length, petal width, sepal length, and sepal width), which take continuous values.

A small adaption of data was necessary. We coded the three possible classes as three values in (0, 1): 0.1, 0.5, and 0.9, respectively. Hence, a single output neuron is required.

In order to obtain three rules, we used the data to train a feedforward network with four, three, and one neurons in each layer. The structure of the network is shown in Fig. 8. Its weights after learning are

$$\begin{aligned} W^t &= [w_{ij}]^t \\ &= \begin{bmatrix} 0.096 & -0.016 & 0.157 & 0.123 \\ -0.085 & -0.012 & 0.131 & 0.021 \\ -0.502 & -0.836 & 0.898 & 1.002 \end{bmatrix} \\ B^t &= [\beta_{jk}]^t \\ &= [13.92,\ -23.179,\ 2.143]. \end{aligned}$$

By applying the rule building process we obtained the following rules:

R_1 If *sepal-length* is greater than approximately 22.916 i-or
sepal-width is not greater than approximately 137.500 i-or
petal-length is greater than approximately 14.013 i-or
petal-width is greater than approximately 17.886,
then $y = 13.92$.

R_2 If *sepal-length* is not greater than approximately 25.882 i-or
sepal-width is not greater than approximately 183.333 i-or
petal-length is greater than approximately 16.794 i-or
petal-width is greater than approximately 104.762,
then $y = -23.179$.

R_3 If *sepal-length* is not greater than approximately 4.382 i-or
sepal-width is not greater than approximately 2.631 i-or
petal-length is greater than approximately 2.450 i-or
petal-width is greater than approximately 2.195,
then $y = 2.143$.

To classify a given instance, it is matched against the three rule premises. Each rule is fired to a certain degree v_j. The overall output is the sum of theses degrees multiplied by the rule weight β_j namely

$$y = 13.92v_1 - 23.179v_2 + 2.143v_3.$$

The class the instance is assigned to is chosen as that with the closest numerical value to y.

The somewhat out of range values of rules R_1 and R_2 is indicative of a small influence in the network classification task. To check this extreme we observed the performance of the FAS using only rule R_3. It was very close to that obtained using the three rules. To get a definitive confirmation we proceed to train a multilayered feedforward net with four, one, and one neurons. And indeed, the new net with a single hidden neuron managed to learn all the knowledge by reaching the same performance as the previous bigger network. The rule extracted from this last net was

If *sepal-length* is not greater than approximately 5.447 i-or
sepal-width is not greater than approximately 4.118 i-or
petal-length is greater than approximately 3.567 i-or
petal-width is greater than approximately 2.61,
then $y = 1.227$.

B. Artificial Neural Networks with Biases

The interpretation of ANN's without biases offered in the preceding section can be extended to explain ANN's whose hidden neurons have biases. The interpretation is again built

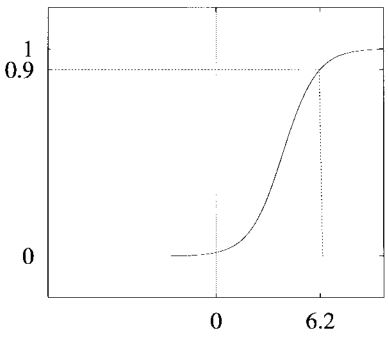

Fig. 9. Membership function for "approximately greater than 6.2."

by using the i-or operator. Some minor changes have to be considered when decomposing rules of type (1) into rules of type (3). Indeed, these changes only affect the definition of the A^i_{jk} fuzzy sets, the rules themselves keep their form (3).

First, it is easy to check that

$$f_A\left(\sum_{i=1}^{n} x_i w_{ij} + \tau_j\right) = f_A(x_1 w_{1j} + \tau'_j) * f_A(x_2 w_{2j} + \tau'_j) * \\ * \cdots * f_A(x_n w_{nj} + \tau'_j).$$

where $\tau'_j = \tau/n$. In this equality, the first term corresponds to the fuzzy proposition "$\sum_{i=1}^{n} x_i w_{ij} + \tau_j$ is A." Likewise, $f_A(x_i w_{ij} + \tau'_j)$ corresponds to proposition "$x_i w_{ij} + \tau'_j$ is A." The latter expression can also be written as "$x_i w_{ij}$ is $A - \tau'_j$," which, following the interpretation for A, can be understood as "$x_i w_{ij}$ is greater than approximately $2.2 - \tau'_j$." Hence the bias term means a sheer translation. Consider the membership function for the $A+4$ fuzzy set, plotted in Fig. 9, and compare it with Fig. 4. The former plot is the same than the latter shifted four units to the left. This gives a clear idea of what is the purpose of using biases in neural networks. They are not to be considered as another weight. They have a special role allowing the neuron to translate the area of influence to its activation state.

Now, the A^i_{jk} fuzzy sets has to be redefined for them to account for both the weight, w_{ij}, and the bias, τ'_j. Their membership function is defined by

$$\mu_{A^i_{jk}}(x) = \mu_A[(x + \tau'_j) w_{ij}].$$

VII. A Method for Knowledge Acquisition

The interpretation of ANN's we have proposed translates knowledge from a cryptic representation (network) into a human-friendly representation (rules). The powerful advantages of neural nets as their capacity to learn and generalize may be exploited to extract knowledge from a set of examples. Now this knowledge may be expressed in terms of fuzzy rules, which have the advantage of permitting an explanation for the answer an FRBS gives. So this is a knowledge acquisition method which captures knowledge from a great deal of samples and converts it into rules.

The proof for Theorem 1 also provides an automated procedure for knowledge elicitation. More specifically, it is a method to extract rules from a network. Many papers dealing with the subject can be found [2]–[4]. However, what is new is the logic connective. The i-or representation power is more condensed than classical "and" and "or" connectives. Moreover, it allows representing information which may not be expressed with neither "and" nor "or."

Therefore, in comparison with other methods, ours yields more complex rules but their number is usually smaller. On the other hand, once you have trained the network, the method is very fast. Since the rule building is straightforward, its computational efficiency is as low as a linear function on the number of hidden neurons.

However, the method can be improved. Rules are simplified if their premises are smaller, with less variables. This can be accomplished by using a feature selection procedure, which detects what inputs are relevant for each output. We have proposed in [2] an extension to handle fuzzy information of the methodology described in [21].

Now, the complete procedure for knowledge acquisition would be as follows.

1) Apply the feature selection procedure and pick relevant inputs for each output.
2) For each output:
   a) A neural network with only influential inputs, an appropriate number of hidden units, and one single output is trained. The training set is derived by projection from the original set.

   b) Extract rules from the network according to the method presented in the paper. (Proof for Theorem 1 and decomposition of rules as explained in Section VI.)

VIII. Conclusions and Final Remarks

We have worked on the problem of interpretation of the most widely known and used model of ANN, the multilayered perceptron. Our research took into account another well-known model, the FRBS, which shares with nets the property of being universal approximators. A connection has been established between both models. An equivalence relation had previously been proven by other authors, while we have shown a stronger relation, equality. This is obtained after a trivial transformation from a network to a set of fuzzy rules of a particular type.

In the quest for a reformulation of rules that makes them more human-friendly, we have defined the concept of f-duality. This concept permits us to obtain an operation in a set induced by an operation in another set by a bijective map. By applying f-duality to the ANN logistic activation function, we found a fuzzy logic operator, which we have called interactive-or. This operator enables us to reformulate fuzzy rules into a more comprehensible form, which can be easily understood. Then the networks may be seen as FBRS's. This way, we can explain clearly the knowledge a net acquires after the learning process. This constitutes an interpretation of ANN's, so they can no longer be considered as black boxes.

Furthermore, the equality result proof yields a method for knowledge acquisition. Compared to other methods based on neural networks, this is one of the most efficient, since the building of the rules is straightforward and its efficiency order is linear. The quality and complexity of rules generated could be improved by including a feature selection procedure.

The concept of f-duality is general and could be used to produce other operators than i-or, which could have interesting applications. This is a promising line of research which will be the object of future works.

In addition, the same approach described here could be applied to other models of neural nets, and so produce an interpretation of those ANN's.

Appendix
Proof of Results

A. Proposition 1

Proof: Let $a, b \in Y$. Then we define $\circledast$ as follows:

$$a \circledast b = f[f^{-1}(a) \oplus f^{-1}(b)].$$

Since f is a bijective function, for each $a \in Y$ there is only one point in X, say x_1, such that $f(x_1) = a$. The same holds for b and a certain point $x_2 \in X$, $f(x_2) = b$. As $\oplus$ is an operation in X then $x_1 \oplus x_2$ is unique and so is $f(x_1 \oplus x_2)$. Hence, the definition of $\circledast$ is all correct.

The uniqueness of the operation is very easy to check. Let $\odot$ be another operation on Y such that

$$f(x_1 \oplus x_2) = f(x_1) \odot f(x_2).$$

Then we have

$$\begin{aligned} f(x_1) \circledast f(x_2) &= f\{f^{-1}[f(x_1)] \oplus f^{-1}[f(x_2)]\} \\ &= f(x_1 \oplus x_2) \\ &= f(x_1) \odot f(x_2). \end{aligned} \tag{9}$$

This concludes the proposition. However, due to its construction, this operation $\circledast$ verifies the very same properties as $\oplus$ does. This implies that f copies the structure of X onto Y and becomes an isomorphism. ■

B. Lemma 1

Proof: It is trivial. ■

C. Lemma 2

Proof: Let $a, b \in (0, 1)$. Let $x_1, x_2 \in \mathbb{R}$ such that $a = f_A(x_1)$, $b = f_A(x_2)$. The logistic function is defined as

$$f_A(x) = \frac{1}{1 + e^{-x}}.$$

Hence

$$\begin{aligned} x &= -\ln \frac{1 - f_A(x)}{f_A(x)} \\ x_1 &= -\ln \frac{1-a}{a}, \\ x_2 &= -\ln \frac{1-b}{b} \\ x_1 + x_2 &= -\ln \frac{1-a}{a} - \ln \frac{1-b}{b} \\ &= -\ln \left(\frac{1-a}{a} \cdot \frac{1-b}{b} \right) \\ &= -\ln \left[\frac{(1-a)(1-b)}{ab} \right]. \end{aligned} \tag{10}$$

By definition of f_A-dual of + (4)

$$\begin{aligned} a * b &= f_A(x_1) * f_A(x_2) \\ &= f_A(x_1 + x_2) \\ x_1 + x_2 &= -\ln \frac{1 - f_A(x_1 + x_2)}{f_A(x_1 + x_2)}. \end{aligned} \tag{11}$$

From (10) and (11) we have

$$f_A(x_1 + x_2) = \frac{a \cdot b}{(1-a)(1-b) + a \cdot b}$$

which leads to

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

D. Lemma 3

Proof: The pair $(\mathbb{R}, +)$ is an abelian group. Hence, by Proposition 1, $[(0, 1), *]$ is an abelian group too, and f_A becomes an isomorphism between both groups. This makes trivial Lemma 3 and its corollary.

The neuter element for $*$ is $f_A(0)$, namely, 1/2.

The inverse element of $a \in (0, 1)$ is $1 - a$. We can check this by noting that $f_A(-x) = 1 - f_A(x)$ (see Lemma 6). ■

E. Lemma 4

Proof: Trivial by associativity of $*$. ■

F. Lemma 5

Proof:

1) When $a_1 \to 0$ then $a_1 \cdot a_2 \cdot \cdots \cdot a_n \to 0$ and $(1 - a_1) \cdot (1 - a_2) \cdot \cdots \cdot (1 - a_n) \neq 0$. Thus, $a_1 * a_2 * \cdots * a_n \to 0$.
2) When $a_1 \to 1$ then $(1 - a_1) \cdot (1 - a_2) \cdot \cdots \cdot (1 - a_n) \to 0$ and hence, $a_1 * a_2 * \cdots * a_n \to 1$.
3) First, we prove the assertion for a_1. Let us consider $a_2, \cdots, a_n$ with fixed values. The expression for $a_1 * a_2 * \cdots * a_n$ reduces to

$$\frac{ka_1}{(1 - a_1)c + ka_1}$$

with k and c being positive constants. Let $a, a' \in (0, 1)$ with $a < a'$. Then we have $1 - a > 1 - a'$ and $ka < ka'$ which implies

$$kaka' + ka(1 - a')c < kaka' + ka'(1 - a)c.$$

This leads to

$$\frac{ka}{(1-a)c + ka} < \frac{ka'}{(1-a')c + ka'}$$

namely $a * a_2 * \cdots * a_n < a' * a_2 * \cdots * a_n$.

Next, by considering commutativity we obtain the result. ■

G. Lemma 6

Proof:

$$
\begin{aligned}
f(-x) &= \frac{1}{1+e^{xw}} \\
1 - f(x) &= 1 - \frac{1}{1+e^{-x}} \\
&= \frac{1+e^{-xw}-1}{1+e^{-xw}} \\
&= \frac{1}{\dfrac{1+e^{-xw}}{e^{-xw}}} \\
&= \frac{1}{1+e^{xw}.}
\end{aligned}
$$
■

References

[1] J. M. Benítez, A. Blanco, and I. Requena, "An empirical procedure to obtain fuzzy rules using neural networks," in *Proc. 6th IFSA Congr.*, Campinas, Brazil, vol. 2, July 1995, pp. 663–666.

[2] J. M. Benítez, A. Blanco, M. Delgado, and I. Requena, "Neural methods for obtaining fuzzy rules," *Mathware Soft Compu.g,* vol. 3, pp. 371–382, 1996.

[3] S. Horikawa, T. Furuhashi, S. Okuma, and Y. Uchikawa, "A fuzzy controller using a neural network and its capability to learn expert's control rules," in *IIZUKA'90,* pp. 103–106.

[4] T. Sudkamp and R. J. Hammell, II, "Interpolation, completion, and learning fuzzy rules," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, pp. 332–342, 1994.

[5] H. Takagi and I. Hayashi, "NN-driven fuzzy reasoning," *Int. J. Approximate Reasoning,* vol. 5, no. 3, pp. 191–212, 1991.

[6] R. P. Lippman, "An introduction to computing with neural nets," *IEEE ASSP Mag.,* vol. 4, pp. 4–22, 1987.

[7] P. D. Wasswerman, *Advanced Methods in Neural Computing.* New York: Van Nostrand Reinhold, 1993.

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

[9] L. A. Zadeh, "Fuzzy logic and approximate reasoning," *Synthese,* vol. 30, pp. 407–428, 1975.

[10] D. Dubois and H. Prade, *Fuzzy Sets and Systems.* New York: Academic, 1980.

[11] A. Kandel, Ed., *Fuzzy Expert Systems.* Boca Raton, FL: CRC, 1991.

[12] H. J. Zimmermann, *Fuzzy Set Theory and Its Applications,* 2nd ed. Boston, MA: Kluwer, 1991.

[13] J. L. Castro, "Fuzzy logic controllers are universal approximators," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, Apr. 1995.

[14] J. L. Castro and M. Delgado, "Fuzzy systems with defuzzification are universal approximators," *IEEE Trans. Syst., Man, Cybern.,* Feb. 1996.

[15] B. Kosko, "Fuzzy systems as universal approximators," *IEEE Trans. Comput.,* vol. 43, pp. 1324–1333, 1994.

[16] M. Sugeno and M. Nishida, "Fuzzy control of model car," *Fuzzy Sets Syst.,* vol. 16, pp. 103–113, 1985.

[17] J. J. Buckley, Y. Hayashi, and E. Czogala, "On the equivalence of neural nets and fuzzy expert systems," *Fuzzy Sets Syst.,* no. 53, pp. 129–134, 1993.

[18] J.-S. R. Jang and C.-T. Sun, "Functional equivalence between radial basis function networks and fuzzy inference systems," *IEEE Trans. Neural Networks,* vol. 4, pp. 156–158, 1992.

[19] J. M. Benítez, J. L. Castro, and I. Requena, "Translation of artificial neural networks into fuzzy additive systems," in *Proc. IPMU96–Information Processing and Management of Uncertainty in Knowledge-Based Systems,* Granada, Spain, July 1996, vol. I, pp. 156–168.

[20] R. R. Yager, "The representation of fuzzy relational production rules," *Appl. Intell.,* vol. 1, no. 1, pp. 35–42, 1991.

[21] S. Sestito and T. Dillon, "Knowledge acquisition of conjunctive rules using multilayered neural networks," *Int. J. Intell. Syst.,* vol. 8, pp. 779–805, 1993.

J. M. Benítez received the M.S. degree in computer science from the University of Granada, Spain, in 1994. He is currently working toward the Ph.D. degree with a doctoral dissertation on obtaining fuzzy rules using neural networks.

He is a Professor in the Department of Computer Science and Artificial Intelligence (DECSAI) of the University of Granada. His research interests include neural networks, fuzzy rule-based systems, artificial intelligence, and software engineering.

Mr. Benítez was awarded the 1994 First National Prize for Computer Science university students.

J. L. Castro received the M.S. degree in 1988 and the Ph.D. degree in 1991, both in mathematics, from the University of Granada, Spain. His doctoral dissertation was on logical models for artificial intelligence.

He is currently a Research Professor in the Department of Computer Science and Artificial Intelligence (DECSAI) at the University of Granada and is a member of the Group of Approximate Reasoning in this department. He has published more than 30 papers and is the author of three books on computer science. His research interests include fuzzy logic, nonclassical logics, approximate reasoning, knowledge-based systems, neural networks, and related applications.

Dr. Castro serves as a reviewer for some international journals and conferences.

I. Requena received the M.S. degree in 1974 and the Ph.D. degree in 1992, both in mathematics, from the University of Granada, Spain. His doctoral dissertation was in neural networks for decision problems with fuzzy information.

He was a Secondary School Teacher in Spain from 1975 to 1989. He is currently a Research Professor in the Department of Computer Science and Artificial Intelligence (DECSAI) in the University of Granada and is a member of the group of Approximate Reasoning in this Department. He has published more than 15 technical papers and has coauthored two books on computer science and another on mathematics for secondary school. His research interests include neural networks, fuzzy rules extraction, knowledge-based systems, and related applications, principally in the economic area.

Dr. Requena serves as a reviewer for some international journals and conferences.