ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 49-54

The Extraction of Sugeno Fuzzy Rules from Neural Networks

Adelmo L. Cechin [1], Ulrich Epperlein [2],
Wolfgang Rosenstiel [1], Bernhard Koppenhoefer [2]

(1) Universität Tübingen, Lehrstuhl für Technische Informatik,
Sand 13, D 72076 Tübingen, Germany
(2) Universität Tübingen, Institut für Organische Chemie,
Auf der Morgenstelle 18, D 72076 Tübingen, Germany

Abstract. This paper describes an algorithm for the extraction of Sugeno fuzzy rules from feedforward neural networks without a dedicated architecture. Three application examples of the algorithm are presented: exclusive or, Iris classification and prediction of chemical properties. The algorithm exposes the strategy used by the network to solve the problem and provides insight into the linearities in the data set. The number of rules generated by the algorithm is $O(number\ of\ patterns)$.

1. Introduction

Feedforward Neural Networks are often considered as black-box models because it is difficult to extract knowledge from them in a comprehensible way [1] [2].

Why should someone want to extract knowledge of a neural network? First as a way to validate the network; second to discover salient features in input data; third to improve the generalization of the network [3]; fourth to connect it to other systems and fifth in applications where security is a mandatory condition like automobilism and power generation.

For those reasons several algorithms were developed which extract knowledge in basically two forms: a) expert rules [1] [3] and b) fuzzy rules [4] . Expert rules are more appropriate for classification problems but not for approximation ones. Most algorithms that extract fuzzy rules from neural networks are based on special architectures [4]. The problem is that these networks contain many hidden layers using neurons with special activation functions that are not adequate for standard training algorithms like backpropagation [5]. Thus, training takes a long time or the networks have convergence problems.

There are several types of fuzzy rules; the main types are the Mamdani type rule and the Sugeno type rule [6] [2]. Compared to the Mamdani rules, the Sugeno rules have the advantage that they implement more efficiently the defuzzyfication. The defuzzyfication is the operation that converts the outputs of all rules in the fuzzy system into one value. This is the most time consuming operation in fuzzy systems.

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 49-54

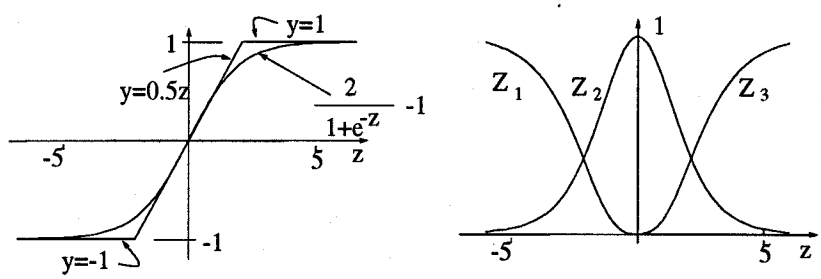

Figure 1: Sigmoid function and consequent part of the neuron rules (left); ideal membership function for each region of the sigmoid neuron (right).

This paper is subdivided basically into two parts: the algorithm to extract Sugeno rules from a neural network and the application of it to different data sets. To the knowledge of the authors this is the first time an algorithm is proposed to extract Sugeno rules from neural networks without a dedicated architecture, so an explicit comparison with other methods was not possible.

2. Fuzzy Rule Extraction

The output of a sigmoid neuron is computed by the weighted sum of the input signals and the application of sigmoid function. For the purpose of this article, the sigmoid function will be defined as $sig(z) = \frac{2}{1+e^{-z}} - 1$ (Fig. 1 (left)), where z is the weighted sum of the input signals.

A Sugeno rule in its simplest form is "IF $z \in Z$ THEN $y = az + b$", where z is the input of the rule, Z is a fuzzy set, y the output of the consequence part, a and b constants. Z is defined by a membership function $\mu_Z(z)$. Both are mathematically different but for the purposes of this paper $\mu_Z(z) = Z(z)$ (for ex.$Z_1(z)$ in Fig. 1(right)). This simplifies the notation.

$z \in Z$ is the premisse of the rule and can be computed by the application of the defining membership function on z.

To compute the output of the rule, the multiplication between premisse and consequence can be used (so called fuzzy implication).

$$output\_of\_rule = (z \in Z)(az + b) = Z(z)(az + b) \tag{1}$$

The premisse of the rule indicates *where* the linear equation in the consequence part is valid and the consequence part indicates *which* linear relation is valid between input and output.

For a sigmoid neuron it is possible to express the activation function mathematically as being composed of 3 rules (see the membership functions in Fig. 1).

$$\begin{aligned} rule_1 &: \texttt{IF } z \in Z_1 \texttt{ THEN} y = -1 \\ rule_2 &: \texttt{IF } z \in Z_2 \texttt{ THEN } y = 0.5z \\ rule_3 &: \texttt{IF } z \in Z_3 \texttt{ THEN } y = 1 \end{aligned} \tag{2}$$

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 49-54

2.1. Algorithm to Extract Fuzzy Rules

The following algorithm is based on two fundamental ideas: first the combination of the membership functions of each sigmoid neuron with the multiplication operation (an "*and*" fuzzy operation). This is done at the step "$premisse_p \leftarrow \bar{z} \in \prod_n membership(n)$". Second, the fact that if all neurons can be expressed in a linear way, then the whole network can be reduced to a single linear relation between inputs and outputs [5]. This happens when the premisse of a rule has the value 1. This linear relation is expressed by the matrices $\bar{\bar{A}}_p$ and $\bar{B}_p$.

The algorithm generates $O(number\ of\ patterns)$ rules. This is a pessimistic estimation. In practice, many patterns tend to cluster in one fuzzy set so that the number of rules is much smaller.

Extraction_algorithm
  Input: network NN, training patterns $(\bar{x_p}, \bar{y_p})$
  Output: rule set R
Train the network NN with the patterns $(\bar{x_p}, \bar{y_p})$
Rule set $R \leftarrow \emptyset$
Foreach input pattern $\bar{x_p}$
  Propagates the input through the network
  (compute the values z_n (Fig. 1(left) for each sigmoid neuron n)
  For each sigmoid neuron n
    $j \leftarrow arg\ max_i\ Z_i(z_n)$ (one on the 3 membership functions in Fig. 1(right))
    $membership(n) \leftarrow\ Z_j$ (Fig. 1(right))
  $premisse_p \leftarrow \bar{z} \in \prod_n membership(n)$
  $consequence_p \leftarrow y = \bar{\bar{A}}_p \bar{x} + \bar{B}_p$
  $candidate\_rule \leftarrow$ IF $premisse_p$ THEN $consequence_p$
  If $candidate\_rule \notin R$
    $R \leftarrow R \cup candidate\_rule$

3. Experimental Comparison

The algorithm was implemented and tested on 3 examples: exclusive or [5], iris data set (this is a classic in the field of pattern recognition) [7], and on chemical data (function approximation problem) [8].

3.1. Exclusive OR

The patterns for the exclusive-or problem are {([0,0],0), ([1,0],1), ([0,1],1), ([1,1],0)}. This pattern set is not linearly separable, that means, we cannot expect to solve the problem with one fuzzy rule (remember that a Sugeno rule has as consequence part a linear relation). The network after 100 epochs is shown in Fig. 2(left) and the correspondent output in Fig. 2(right). In the Fig. 2(left), the neurons marked with i are input neurons, s are sigmoid neurons and + a linear neuron.

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 49-54

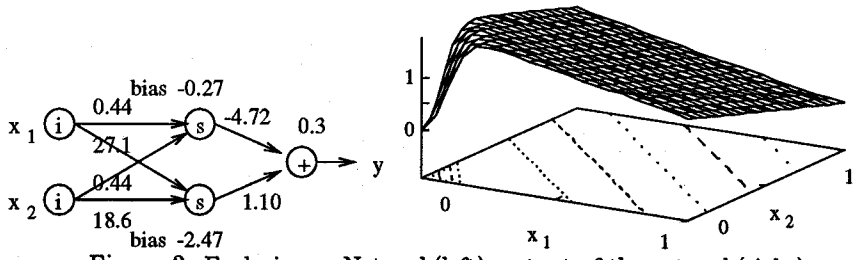

Figure 2: Exclusive-or Network(left); output of the network(right).

The algorithm described above was used to extract the fuzzy rules from the trained network. Two rules were obtained:

1. IF $(x_1, x_2) \in M_1$ THEN $y = -0.19$
2. IF $(x_1, x_2) \in M_2$ THEN $y = -0.74x_1 - 0.83x_2 + 2.01$

Like the network, the system above computes a real value that shall be post-processed by a threshold (≥ 0.5) to get the binary values $0, 1$.

The fuzzy membership functions M_1 and M_2 are shown in the Fig. 3(left) and Fig. 3(right), respectively.

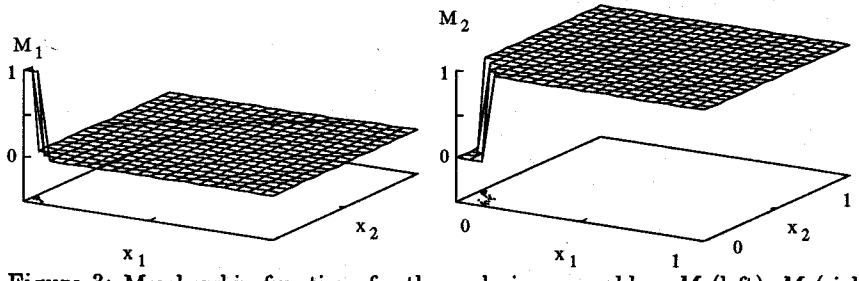

Figure 3: Membership functions for the exclusive or problem M_1(left); M_2(right).

3.2. Iris Data Set

The Iris data set is composed of 150 Patterns. Each pattern contains four parameters describing the iris plant and one value that indicates the species (*Iris setosa*, *Iris versicolor* or *Iris virginica*). The problem is to recognize the species based on the four describing parameters.

Three networks with the same structure (4 input neurons, 6 nonlinear hidden and 1 output) were trained, each one to recognize one of the plants. Here only the application of the algorithm to the first two networks will be described.

The patterns for *Iris setosa* are linearly separable from the other patterns, so there is hope that a single rule will solve the problem. After training the network for 20 epochs the extraction algorithm was applied and the following rule obtained (valid in the whole space).

IF True THEN $y = -0.11x_1 + 0.39x_2 - 0.09x_3 - 0.20x_4 + 0.52$

The second network was employed to recognize *Iris versicolor*. After training for 160 epochs the response of the network can be seen in Fig. 4(left), the fuzzy sets extracted in Fig. 4(right) and the corresponding fuzzy rules below:

IF $(\bar{x}) \in M_1$ THEN $y = 0.25x_1 + 0.27x_2 - 0.32x_3 - 0.83x_4 + 1.30$
IF $(\bar{x}) \in M_2$ THEN $y = 0.12x_1 - 0.70x_2 + 0.95x_3 + 1.11x_4 - 0.57$
IF $(\bar{x}) \in M_3$ THEN $y = 0.27x_1 + 0.11x_2 - 0.20x_3 - 0.21x_4 - 1.54$
IF $(\bar{x}) \in M_4$ THEN $y = 0.25x_1 + 0.27x_2 - 0.32x_3 - 0.83x_4 - 1.77$

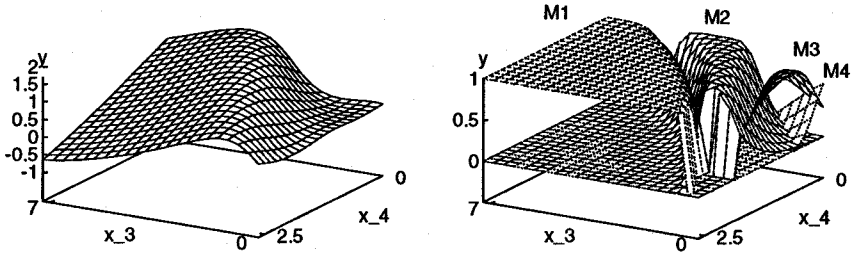

Figure 4: Response of the network (left); fuzzy sets extracted (right).

The interpretation of the rules obtained exhibits some strange behaviour of the network:

i) A linear function (first rule) is used to separate the *Iris versicolor* patterns from those of *Iris virginica*. Both are under the fuzzy set M_1 in Fig. 4(right), corresponding to the big plane on the left side of the surface in Fig. 4(left). However the patterns of these two species are not linearly separable, what explains the missclassification of three patterns by the network.

ii) In contrast, three rules (M_2, M_3 and M_4) are used by the network to represent the *Iris setosa* patterns. These 3 rules could be simplified to IF $(\bar{x}) \in M_2 \cup M_3 \cup M_4$ THEN *Iris setosa*. Though the network was trained to recognize *Iris versicolor* pattens only, the application of the algorithm to the network permits the recognition of both other species, too. This happens because the patterns of *Iris versicolor* are mainly between *Iris setosa* and *Iris virginica*.

3.3. Chemical Data

This example deals with the prediction of the separation factor α in gas chromatography. It is desired to predict this property directly from the molecular structure, that was numerically coded and used as input to the neural network. A set of 33 substances were used to generate the training patterns (a fairly high number compared to other projects in chemistry). The substances were coded using 8 parameters: length of the carbon chains (2 parameters), length of the ramifications (5 parameters) and molecular weight.

A network with 8 input neurons, 5 hidden and 1 output neuron was trained for 500 epochs and the fuzzy rules extracted. Four rules were obtained but for space saving only one rule is shown:

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 49-54

1. IF $(\bar{x}) \in M_1$ THEN $y = -0.068x_1 + 0.57x_2 - 0.39x_3 - 0.23x_4 + 0.12x_5 + 0.17x_6 - 0.079x_7 - 0.34x_8 + 0.55$

The membership functions are surfaces in an 8-dimensional (number of inputs) space, what is very difficult to represent graphically. Nevertheless, the membership functions can be approximately represented by the pattern at the maximum of the membership function (its center, e.g. $(3, 0, 0, 0, 3, 2, 0, 187.2583)$ for the first rule). Near to each center in the 8-dimensional space, the linear relation in the correspondent rule is valid. Hence, the chemical compounds were classified into four groups with a linear relationship inside each group. Additionally, it is possible to determine which parameters are responsible for the nonlinearity of the relationships between property and chemical structure and which parameters influence the property inside each group.

4. Conclusion

A method was presented to extract Sugeno fuzzy rules from a neural network without a dedicated architecture. This method was successfully applied to three data sets conducting to an interpretation of the network employed.

References

[1] G.Towell and J.W.Shavlik: *Refining Symbolic Knowledge Using Knowledge-Based Neural Networks, Machine Learning, 13, (1993) 71-101*

[2] D.Nauck and F.Klawonn and R.Kruse: *Neuronale Netze und Fuzzy-Systeme,* Vieweg Verlag, Braunschweig, (1994)

[3] C.McMillan and M.C.Mozer and P.Smolensky: *The Connectionist Scientist Game: Rule Extraction and Refinement in a Neural Network,* Technical Report CU-CS-530-91, University of Colorado, (1991)

[4] J.S.R.Jang: *Self-Learning Fuzzy Controllers Based on Temporal Backpropagation,* IEEE Transactions on Neural Networks, Vol.3, no.5, September (1992) 714-723

[5] D.E.Rumelhart and G.E.Hinton and R.J.Williams: *Learning Internal Representations by Error Propagation,* in Parallel Distributed Processing, D.E.Rumelhart and J.L.McClelland, The MIT Press, Chapter 8, (1986)

[6] T.Takagi and M.Sugeno: *Fuzzy Identification of Systems and Its Application to Modelling and Control,* IEEE Transactions on Systems, Man, Cybernetics, Vol.15 (1985) 116-132

[7] R.A.Fisher: *The Use of Multiple Measurements in Taxonomic Problems,* in Annual Eugenics, II, Vol.7, p. 179-188. John Wiley, NY, (1950)

[8] B.Koppenhoefer and H.Allmendinger and G.J.Nicholson and E.Bayer: *Direct Resolution of Enantiomers of 2- and 3-Hydroxy Acid Alkyl Esters by Fused-Silica Capillary Gas Chromatography,* Journal of Chromatography, Vol.260 (1983) 63-73

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 49-54