Learning Fuzzy Rules from Data

Frank Klawonn[1] and Annette Keller[2]

[1] Fachbereich Elektrotechnik und Informatik
Fachhochschule Ostfriesland
Constantiaplatz 4
D-26723 Emden, Germany
[2] Deutsches Zentrum für Luft- und Raumfahrt
Institut für Flugführung
Lilienthalplatz 7
D-38108 Braunschweig, Germany

Abstract. This paper briefly reviews techniques for learning fuzzy rules. In many applications fuzzy if-then rules are not interpreted as implications, but in a procedural way that corresponds to a conjunction or Cartesian product. For this type of rules many approaches to rule learning have been proposed. However, for more complex fuzzy systems based on logical implications, there is still a need for suitable learning algorithms.

1 Introduction

When L.A. Zadeh introduced the notion of a fuzzy set in 1965, his intention was to incorporate vague knowledge in a model of a system. A fuzzy set admits a gradual membership between zero and one. The first applications of fuzzy sets in the field of control were completely in this spirit. Although the idea to describe a vague concept in terms of a fuzzy sets seemed to be appealing, it turned out that the specification of optimal fuzzy sets is a tedious task. Therefore, methods for tuning and learning fuzzy rules from data were developed.

In this paper we briefly review techniques for learning fuzzy rules from data. We distinguish between two types of fuzzy rules. In most application, especially in fuzzy control, fuzzy if-then rules are understood in a procedural way and are therefore interpreted as a Cartesian product, i.e. from the viewpoint of logic in a conjunctive manner. For this type of fuzzy systems various learning algorithms were developed on the basis of neural networks, evolutionary algorithms, cluster analysis, and regression. But these systems are not suited for complex knowledge that involves chaining of rules.

However, for the type of rules that can cope with more complex knowledge, there is still a need for appropriate learning techniques.

The paper is organized as follows. Section 2 provides a short introduction to fuzzy sets and fuzzy logic. In section 3 we take a closer look at fuzzy logic in the narrow sense. Sections 4 and 5 are devoted to learning techniques for Cartesian product-like rules and logical implications, respectively.

2 Fuzzy Rules and Cartesian Products

A fuzzy set is defined is mapping $\mu : X \to [0,1]$ from the domain X to the unit interval. The value $\mu(x)$ is understood as the membership degree of the element x to the fuzzy set μ. If we interpret the fuzzy set μ as a predicate on X, then $\mu(x)$ corresponds to the truth value of the statement $x \in \mu$ in a $[0,1]$-valued logic. We denote the truth value of a logical formula φ by $[\![\varphi]\!]$. If we think of fuzzy sets in this sense, we can define operations like intersection, union, or complement for fuzzy sets on the basis of the corresponding many-valued logical connectives conjunction, disjunction, and negation, respectively.

We do not discuss the problems connected to this logic-based view of fuzzy sets, especially the consequences of truth-functionality. For a more detailed discussion to fuzzy sets in general and their interpretation we refer to [16].

In fuzzy logic a variety of truth functions for the logical connectives is allowed. For the conjunction generally a t-norm – a binary operation on the unit interval that is commutative, associative, increasing in both arguments with one as unit – is chosen. The most common one is the minimum, but also the product, and the Łukasiewicz t-norm defined by $[\![\varphi_1 \wedge \varphi_2]\!] = \max\{[\![\varphi_1]\!] + [\![\varphi_2]\!] - 1, 0\}$ are used. Disjunctions are represented by t-conorms, having the same properties as t-norms, except that they have zero as unit instead of one. Typical t-conorms are the maximum, the bounded sum ($[\![\varphi_1 \vee \varphi_2]\!] = \min\{[\![\varphi_1]\!] + [\![\varphi_2]\!], 1\}$), and the dual to the Łukasiewicz conjunction ($[\![\varphi_1 \vee \varphi_2]\!] = [\![\varphi_1]\!] + [\![\varphi_2]\!] - [\![\varphi_1]\!][\![\varphi_2]\!]$).

The Gödel implication

$$[\![\varphi_1 \to \varphi_2]\!] = \begin{cases} [\![\varphi_2]\!] & \text{if } [\![\varphi_2]\!] < [\![\varphi_2]\!] \\ 1 & \text{otherwise} \end{cases}$$

and the Łukasiewicz implication $[\![\varphi_1 \to \varphi_2]\!] = \min\{1-[\![\varphi_1]\!]+[\![\varphi_2]\!], 1\}$ are the most common implications in fuzzy logic. They are the corresponding residuated implications to the conjunctions minimum and Łukasiewicz t-norm, respectively. An implication is residuated with respect to a conjunction, if

$$[\![\varphi_1 \wedge \varphi_2]\!] \leq [\![\varphi_3]\!] \iff [\![\varphi_1]\!] \leq [\![\varphi_2 \to \varphi_3]\!].$$

In this case we have $[\![\varphi_1 \to \varphi_2]\!] = \sup_{\varphi_3}\{[\![\varphi_3]\!] \mid [\![\varphi_1 \wedge \varphi_3]\!] \leq [\![\varphi_2]\!]\}$.

In applications usually rules of the form

$$\text{If } \xi_1 \text{ is } \mu_1 \text{ and } \ldots\ \xi_n \text{ is } \mu_n \text{ then } \xi \text{ is } \mu \tag{1}$$

are used. $\xi_1, \ldots, \xi_n$ and ξ are variables on the domains $X_1, \ldots, X_n$, and X, respectively. In the rules instead of the fuzzy sets usually linguistic terms like *approximately zero*, *very large* etc. associated with the fuzzy sets appear. For reasons of a simplified notation we do not distinguish between the fuzzy sets and the associated linguistic terms.

It should be noted that for applications it is necessary to distinguish between classification rules and interpolative rules. The difference lies in the structure of the domain X of the variable ξ. If X is a discrete set then usually the fuzzy set μ is a crisp set with just one element. This means that in this case the rules are of the form

$$\text{If } \xi_1 \text{ is } \mu_1 \text{ and } \ldots\ \xi_n \text{ is } \mu_n \text{ then class is } x \tag{2}$$

where $x \in X$. The reason for the distinction between classification and interpolation rules is that the fuzzy rules provide fuzzy information on the output variable ξ and in the case of a continuous output domain X, it makes sense to apply a kind of interpolation technique to the fuzzy outputs obtained from the single rules. But if X contains discrete classes, then interpolation between these classes is not applicable.

The fuzzy rules (1) are usually not interpreted in the sense of logical implications, but in the sense of a piecewise defined (fuzzy) function, i.e. in the sense

$$\xi \;\approx\; \begin{cases} \mu^{(1)} & \text{if } \xi_1 \approx \mu_1^{(1)} \text{ and } \ldots \text{ and } \xi_n \approx \mu_n^{(1)} \\ \vdots & \\ \mu^{(r)} & \text{if } \xi_1 \approx \mu_1^{(r)} \text{ and } \ldots \text{ and } \xi_n \approx \mu_n^{(r)} \end{cases} \tag{3}$$

Therefore, if we have r fuzzy rules of this type, they represent the following fuzzy relation or fuzzy predicate on the product space $X_1 \times \ldots \times X_n \times X$:

$$\mu_{\text{cart}}(x_1, \ldots, x_n, x) \;=\; \bigvee_{k \in \{1,\ldots,r\}} \mu_1^{(k)}(x_1) * \ldots * \mu_n^{(k)}(x_n)$$

where $\bigvee$ indicates the supremum and $*$ is the truth function of a $[0,1]$-valued conjunction. Thus, when the values $x_1, \ldots, x_n$ are known, we obtain a fuzzy set (or fuzzy predicate) on the domain X:

$$\mu_{\text{out}}^{(x_1,\ldots,x_n)}(x) \;=\; \mu_{\text{cart}}(x_1, \ldots, x_n, x)$$

This fuzzy set has to be defuzzified, in order to have crisp decision. In case of classification rules, i.e. when the output domain x is discrete, one simply chooses the output value (class) $x \in X$ with the highest membership degree to the fuzzy set $\mu_{\text{out}}^{(x_1,\ldots,x_n)}$. For continuous output domains X usually a kind of 'interpolation' between the outputs of the single rules is carried out by taking the centre of gravity of the fuzzy set $\mu_{\text{out}}^{(x_1,\ldots,x_n)}(x)$. A detailed discussion and justification of defuzzification methods would lead to far in this paper.

It should be noted that this kind of evaluation of a system of fuzzy if-then rules has nothing to do with logical inference in the classical sense. The following section is devoted to the concept how logical inference is carried in fuzzy logic.

3 Fuzzy Logic in the Narrow Sense

In order to understand the basic ideas of fuzzy logic in the narrow sense as it was introduced for propositional logic by Pavelka [23] and later on for first order logic by Novák [22], we compare them to the concepts in classical propositional logic. The syntax of classical propositional logic is based on

- variables for propositions $\varphi, \psi, \ldots$
- logical connectives $\wedge, \vee, \neg, \rightarrow, \leftrightarrow, \ldots$
- brackets (,).

For the semantics an interpretations are introduced that assign to each propositional variable the value *true* (1) or *false* (0). Given truth tables for the logical connectives like

$[\![\varphi]\!]$	$[\![\psi]\!]$	$[\![\varphi \wedge \psi]\!]$
1	1	1
1	0	0
0	1	0
0	0	0

an interpretation I can be extended to any logical formula that is composed out of propositional variables. We denote the truth value of a logical formula under interpretation I by $[\![\varphi]\!]_I$ or $[\![\varphi]\!]$, when it is obvious to which interpretation we refer.

In order to model a special 'world' or situation, a set of special axioms $\mathcal{A}$ of logical formulae is assumed to be true. A model of $\mathcal{A}$ is an interpretation I such that I assigns 1 to all formulae in $\mathcal{A}$. We write $\mathcal{A} \models \varphi$, if φ is true in all models of $\mathcal{A}$, and call φ a tautology if $\emptyset \models \varphi$ (or, for short $\models \varphi$) holds.

Proof theory or logical deduction provides rules to derive other true formulae from formulae that are already known to be true. The most simple ones are

$$\textit{modus ponens} \qquad \frac{\varphi, \quad \varphi \to \psi}{\psi}$$

and *substitution*:

> If φ is established to be true then one may assert as true any other formula φ' obtained by uniformly replacing all occurrences in φ of some propositional variable by any formula whatsoever.

In addition to the rules a set of tautologies or general axioms is needed in order to have a starting point for the inference or deduction rules. Classical propositional logic (as well as classical first order logic) is known to be sound and complete, i.e. $\mathcal{A} \vdash \varphi \iff \mathcal{A} \models \varphi$ holds.

Fuzzy logic can be seen as a (partially) valuated [0,1]-valued logic. First of all we have to fix the logical connectives including their truth functions that we want to incorporate in our system of fuzzy logic.

In classical logic a special 'situation' or 'world' under consideration is specified by a set of special axioms $\mathcal{A} \subseteq L$, where L is the formal language (set of well formed formulae) of propositional logic.

We can also represent $\mathcal{A}$ by its characteristic function $a : L \to \{0,1\}$, where $a(\varphi) = 1 \Leftrightarrow \varphi \in \mathcal{A}$. $a(\varphi) = 1$ means φ is true, whereas $a(\varphi) = 0$ does not imply that φ is necessarily false, but it means that φ is not known to be true, it might be true or false. For fuzzy logic the language is enriched with the set of truth values $[0,1]$. We denote the set of such formulae by L^*. Note that this set is not even countable. However, this problem can be amended in some cases by introducing a new unary logical connective $\sqrt{\ }$ [7]. But this is not important for our discussion here.

In fuzzy logic the set of special axioms in classical logic (or its characteristic function) is replaced by fuzzy set of axioms $\mu : L^* \to [0,1]$. Again, as in classical logic, $\mu(\varphi)$ does not specify the truth value of the formula φ, but it provides only a lower bound for $[\![\varphi]\!]$.

An model of the set of special axioms $\mathcal{A}$ in classical logic is a $\{0,1\}$-valued interpretation I such that at least the formulae in $\mathcal{A}$ are true, i.e. $\mathcal{A} \subseteq \mathcal{F}_I$, where $\mathcal{F}_I$ is the set of true formulae under interpretation I. A model of the fuzzy set of special axioms μ is a [0,1]-valued interpretation I such that $\mu(\varphi) \leq [\![\varphi]\!]_I$ holds. A logical formula can be semantically inferred from a set $\mathcal{A}$ of special axioms in classical logic, if φ is true in all models of $\mathcal{A}$, i.e.

$$\mathcal{A} \models \varphi \iff \varphi \in \bigcap_{I \text{ is a model of } \mathcal{A}} \{\psi \text{ is true in } I\}.$$

In fuzzy logic a formulae φ can be semantically inferred from a fuzzy set of axioms μ to a certain degree, namely, to the lowest truth degree that is assigned to φ in the models of μ:

$$[\![\mu \models \varphi]\!] \;=\; \bigwedge \{[\![\varphi]\!]_I \mid I \text{ model of } \mu\}$$

Table 1 illustrates the relations between fuzzy logic and classical logic that we have discussed so far. However, we still have to discuss the role of deduction or proof theory in fuzzy logic.

Since logical formulae are pre-evaluated in fuzzy logic by the fuzzy set of axioms, the lower bounds for the truth value of a formula has to be taken into account in deduction in fuzzy logic. Let us consider the example of modus ponens in fuzzy logic. We assume that we use the Łukasiewicz implication as the truth function for the implication.

classical logic	**fuzzy logic**
set of special axioms $\mathcal{A} \subseteq L$ mapping $a : L \to \{0,1\}$, where $a(\varphi) = 1 \Leftrightarrow \varphi \in \mathcal{A}$	fuzzy set of axioms $\mu : L^* \to [0,1]$
$a(\varphi) = 1$ means φ is true $a(\varphi) = 0$ means φ is not known to be true, it might be true or false	$\mu(\varphi)$ lower bound for $[\varphi]$ L^*: L extended with logical constants for the truth values
model	
$\{0,1\}$-valued interpretation I s.t. at least the formulae in $\mathcal{A}$ are true, i.e. $\mathcal{A} \subseteq \mathcal{F}_I$, where $\mathcal{F}_I$ is the set of true formulae under interpretation I	[0,1]-valued interpretation I s.t. $\mu(\varphi) \leq [\varphi]_I$
$\models$	
$\mathcal{A} \models \varphi$ iff $\varphi \in \bigcap_{I \text{ is a model of } \mathcal{A}} \{\psi \text{ is true in } I\}$	$[\mu \models \varphi] = \bigwedge \{[\varphi]_I \mid I \text{ model of } \mu\}$

Table 1. Classical and fuzzy logic

Assume, we already know $[\![\varphi]\!] \geq \alpha$ and $[\![\varphi \to \psi]\!] \geq \beta$ (for instance, because we have $\mu(\varphi) = \alpha$ and $\mu(\varphi \to \psi) = \beta$). This means that

$$\beta \leq [\![\varphi \to \psi]\!] = \min\{1 - [\![\varphi]\!] + [\![\psi]\!], 1\}$$

holds and this implies

$$[\![\psi]\!] \geq [\![\varphi]\!] + \beta - 1 \geq \alpha + \beta - 1.$$

Thus we have that from $[\![\varphi]\!] \geq \alpha$ and $[\![\varphi \to \psi]\!] \geq \beta$ we can derive $[\![\psi]\!] \geq [\![\varphi \wedge (\varphi \to \psi)]\!]$ where the conjunction is evaluated by the Łukasiewicz conjunction.

More generally: If the conjunction in fuzzy logic is evaluated by a continuous t-norm and the implication by the corresponding residuated implication, then starting from $[\![\varphi]\!] \geq \alpha$ and $[\![\varphi \to \psi]\!] \geq \beta$ we can infer $[\![\psi]\!] \geq [\![\varphi \wedge (\varphi \to \psi)]\!]$.

Therefore, modus ponens for fuzzy logic (with Łukasiewicz implication) is represented in the following way:

$$\frac{\varphi,\ \varphi \to \psi}{\psi} \left(\frac{\alpha, \beta}{\max\{\alpha + \beta - 1, 0\}} \right)$$

The modus ponens now involves in addition to the formulae also there (actual) lower bounds for their truth values. The general form of an inference rule in fuzzy logic is therefore of the following form:

$$\frac{\varphi_1, \ldots, \varphi_n}{r^{\text{syn}}(\varphi_1, \ldots, \varphi_n)} \left(\frac{\alpha_1, \ldots, \alpha_n}{r^{\text{sem}}(\alpha_1, \ldots, \alpha_n)} \right)$$

A proof for the formula φ to the degree α in fuzzy logic is then obtained by applying such inference rules until we finally obtain φ (α). This leads to the definition of the degree of provability of a formula, given the fuzzy set of axioms:

$$[\![\mu \vdash \varphi]\!] \ = \ \bigvee \{\alpha \mid \text{ there is a proof for } \varphi \text{ to the degree } \alpha\}$$

Under quite general conditions and assuming that the truth function of the implication is continuous, it can be proved that fuzzy logic is sound and complete with respect to the above definitions of $\models$ and $\vdash$.

It should be emphasized that this is essentially only approximative completeness, since the supremum can be a proper one and therefore there is not always a proof yielding the truth degree in the semantic sense ($\models$). It can only be guaranteed that there is a sequence of (valuated) proofs, that converges to the semantic truth degree.

4 Learning Cartesian Product Rules

Learning fuzzy rules from data involves two tasks:

1. the identification of the rules themselves and
2. the adaptation of the fuzzy sets appearing in the rules.

The first approaches to learning fuzzy rules from data were based on neural network concepts. There are a number of neuro-fuzzy systems, for instance [2, 6, 20, 21], for an overview see [19], and it is not possible to describe them here in detail. The underlying principal of all these models, is that they need an evaluation of the output that the fuzzy rules represented by the neural network produce for a given input. This can be an error measure or simply a binary good-or-bad judgement. Then the models aim at improving the output by changing the fuzzy sets appearing in the rules in a suitable way. This change is usually derived by some backpropagation technique or simply by a heuristic algorithm. They are usually designed for the adaptation of the fuzzy sets, but sometimes also involve a simple ad-hoc strategy for choosing an initial set of rules, adding or deleting rules while learning. They strongly rely on the simple architecture that is induced by the interpretation of the fuzzy rules in the spirit of a Cartesian product. This allows them to identify, how changes in the fuzzy sets effect the output. Therefore, they are quite suited for tuning the fuzzy sets in this type of rules, but are not designed for learning rules and it is not obvious how they could be extended to more complex rule structures.

Another popular approach to learning fuzzy rules is based on evolutionary algorithms. Evolutionary algorithms are a quite general technique for parameter optimization, that are motivated by concepts of biological evolution. They start with an initial 'population' (set) of possible solutions of the optimization problem, that is usually randomly generated. Then they change some of the values/parameters of the population ('mutation') and apply 'biological' operators like crossover that exchanges values between different solutions. Then the best solutions are selected to form the next generation which is treated in the same way until a maximum number of generations is computed or a solution with a desired quality appears.

The fuzzy rules interpreted in the sense of (3) can be encoded in an $(n+1) \times r$-array where for each rule $(n+1)$ fuzzy sets (or corresponding linguistic terms) have to be chosen. This discrete problem is a task for a genetic algorithm, whereas the optimization of the fuzzy sets can be carried out by an evolution strategy, when each fuzzy set is encoded by a set of real-valued parameters. Approaches in a similar spirit are described in [9, 10, 17].

Evolutionary algorithms require a parameterized formulation of the problem with a quite limited number of parameters. Therefore, they can be applied to fuzzy systems where the number of possible rules is strictly limited and only a few parameterized fuzzy sets have to be learned. They are not suited for learning more complex fuzzy rule bases.

An alternative approach to learning fuzzy rules interpreted in the Cartesian product sense is fuzzy clustering. In terms of (3) each fuzzy if-then rule specifies a vague point of the graph of the described function. If for instance triangular membership functions are used, the point or vector with the coordinates of the tips of the triangles is a 'typical' point on the function. With increasing distance, points in its neighbourhood are less 'typical' and have therefore a decreasing membership degree to the vague point defined by the Cartesian product of the fuzzy sets appearing in the rule. In this sense, a fuzzy rule can be characterized by a typical point (on the function) and a (multidimensional) membership function that is decreasing with increasing (transformed [11]) distance to the typical points. Many fuzzy clustering algorithms are exactly pursuing the same strategy: A fuzzy cluster is represented by a typical element – usually the cluster centre – and the membership degree of a datum to the cluster is decreasing with increasing, sometimes transformed distance to the cluster centre.

Many fuzzy clustering approaches characterize each cluster by a set of parameters, the so-called prototype. Given a set $X = \{x_1, \ldots, x_N\} \subset$

$\mathbb{R}^p$ of sample data, the aim of objective function based fuzzy clustering [3] is to determine the prototypes in such a way that the objective function

$$J(X,U,v) \;=\; \sum_{i=1}^{c}\sum_{k=1}^{N}(u_{ik})^m d^2(v_i,x_k) \tag{4}$$

is minimized. u_{ik} stands for the membership degree of datum x_k to cluster i, $d(v_i,x_k)$ is the distance of datum x_k to the cluster i, represented by the prototype v_i. c is the number of clusters. To avoid the trivial minimum $u_{ik}=0$ for all i,k, either the probabilistic constraint

$$\sum_{i=1}^{c} u_{ik} \;=\; 1 \qquad \text{for all } k \in \{1,\dots N\} \tag{5}$$

has to be assumed or the term

$$\sum_{i=1}^{c}\eta_i\sum_{k=1}^{N}(1-u_{ik})^m$$

has to be added to (4) leading to the concept of possibilistic clustering [15]. It is obvious that in order to minimize (4) data points x_k with a small distance $d(v_i,x_k)$ to the cluster i should be assigned a high membership degree whereas data with larger distances should have low membership degrees.

A standard approach to obtain a concrete algorithm is to derive necessary conditions for the membership degrees u_{ik} and the prototypes v_i in order to obtain a (local) minimum of the objective function (4). The clustering algorithm simply starts with a random initialization and updates the u_{ik} and the v_i alternatingly in an iterative procedure. We do not go into the details of the algorithms, since they are not of importance for our considerations in this paper. For an overview on various fuzzy clustering algorithms and their detailed description we refer to [8].

The principal idea to apply fuzzy clustering in order to derive if-then rules from data is that each cluster induces a rule by projecting the cluster to the corresponding coordinate spaces [14, 24]. The projection of a cluster to the i-th domain is obtained by taking the i-th coordinate of each data point and assigning to it the membership degree of the original data point to the cluster. In this way a discrete fuzzy set is defined on the i-th coordinate space. To extend this fuzzy set to the whole i-th domain, a piecewise linear fuzzy set can be defined on the basis of these discrete points, an enveloping fuzzy set or a suitable approximation by a parameterized fuzzy set like a triangular or trapezoidal membership function can be chosen [24].

It is obvious that fuzzy clustering relies on the interpretation of the rules in terms of Cartesian products, and is therefore well suited for learning such rules. However, fuzzy clustering does not yield optimal fuzzy sets due to the loss of information caused by the projection of clusters. And it is definitely not applicable to learning other types of fuzzy rules.

5 Neural Fuzzy Logic Programs

Because of the interpretation of the rules as Cartesian products a simple identification of the effect that a new rule or a change of a fuzzy set within a rule has on the output is not very difficult. This allows the application of quite simple learning techniques. In the following we discuss an approach that, although it is restricted to propositional logic, demonstrates an application of the backpropagation technique of neural networks to sets of logical rules that allow chaining.

The underlying idea is based on the observation that the network input of a neuron in a neural network can be interpreted as the implementation of modus ponens in fuzzy logic. A neuron u obtains the weighted sum of the outputs of the neurons that are connected to u as network input:

$$\sum_v W(v,u)o_v \tag{6}$$

$W(v,u) \in \mathrm{I\!R}$ is the weight that connects the neurons v and u, $o_v \in [0,1]$ is the output of neuron v.

Let us now consider the logical formula $\varphi_1 \vee \ldots \vee \varphi_n \to \varphi$. We assume that we know $[\![\varphi_i]\!] \geq \alpha_i$ for $i = 1,\ldots,n$ and $[\![\varphi_1 \vee \ldots \vee \varphi_n \to \varphi]\!] \geq \beta$. We evaluate the disjunction by the bounded sum, i.e. $[\![\varphi_1 \vee \varphi_2]\!] = \min\{[\![\varphi_1]\!] + [\![\varphi_2]\!], 1\}$ and the implication by the Goguen implication

$$[\![\varphi_1 \to \varphi_2]\!] \;=\; \begin{cases} 1 & \text{if } [\![\varphi_1]\!] \le [\![\varphi_2]\!] \\ \frac{[\![\varphi_2]\!]}{[\![\varphi_1]\!]} & \text{otherwise,} \end{cases}$$

the residuated implication to the product conjunction. The modus ponens in fuzzy logic then yields

$$[\![\varphi]\!] \;\ge\; \min\left\{\sum_{i=1}^{n} \beta\alpha_i, 1\right\}.$$

The similarity to the network input formula (6) is obvious. The lower bounds α_i for the truth values of the formula φ_i correspond to the outputs of the neurons 'φ_i', the weights are all identical, being equal to the lower bound of the truth value for the rule $\varphi_1 \vee \ldots \vee \varphi_n \to \varphi$.

This relation between neural networks and fuzzy logic has been exploited in [4] to derive a learning algorithm for propositional fuzzy Prolog programs. This paper generalizes from the truth functions that were the basis for the motivation to quite general truth functions. As a consequence, the network input might differ from the classical formula (6). As in multilayer perceptrons a training algorithm based on a modified backpropagation technique can be derived that allows learning of the weights that represent (the lower bounds for) the truth values of the implication rules. It should be noted that the transformation of a set of rules to neural network requires some modifications of the set of the rules. Nevertheless, this modification leads to a set of formulae that is equivalent to the original one.

Although this approach allows the application of learning from data to fuzzy systems incorporating chaining of rules, the restriction to propositional logic is drastic and larger sets of formulae cannot be handled, since the transformation to a neural network involves a significant increase of the number of formulae.

6 Conclusions

Today, applications of fuzzy systems seem to be restricted mainly to simple input-output systems. Simple does not mean that the modelled system is simple, but that the structure of the fuzzy systems requires a more or less direct connection between inputs and outputs. Because of this simple structure it is not very difficult to develop suitable learning algorithms for such fuzzy systems. However, these algorithms cannot be extended to fuzzy systems with a more complicated underlying structure.

The specification of membership or truth degrees is however much more tedious for complex rule bases and it seems almost impossible to define a consistent systems without adjusting the parameters by appropriate learning techniques. The problem of good experts estimations of parameters appears also in probabilistic settings, for instance in Bayesian networks. For a more complex network it is almost impossible to specify the necessary conditional probability distributions simply on the basis of experts experience. However, in the probabilistic setting the parameters, i.e. the probabilities, have a concrete interpretation in the sense of relative frequencies so that they can be derived and verified on the basis of sample data by corresponding statistical methods. For fuzzy systems involving truth or membership degrees such an interpretation of the numbers allowing to derive them directly from data is still missing. Membership degrees are usually understood in an intuitive way that does not provide the basis for direct learning techniques. There are approaches to define a more concrete interpretation of the membership degrees in terms of possibility distributions [5], preference [1], similarity [11], partial contradiction [13, 18], or (pessimistic) probabilities [12]. However, except for the possibility setting no learning algorithms were developed on the basis of these interpretations.

References

1. Bellman, R.E., Zadeh, L.A.: Decision Making in a Fuzzy Environment. Management Sciences 17 (1970), 141-164
2. Berenji, H.R., Khedkar, P.: Learning and Tuning Fuzzy Logic Controllers through Reinforcement Learning. IEEE Trans. Neural Networks 3 (1992), 724-740
3. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
4. Eklund, P., Klawonn, F.: Neural Fuzzy Logic Programming. IEEE Trans. Neural Networks 3 (1992), 815-818
5. Gebhardt, J., Kruse, R.: The Context Model – An Integrating View of Vagueness and Uncertainty. Approximate Reasoning 9 (1993), 283-314

6. Halgamuge, S.K., Glesner, M.: The Fuzzy Neural Controller FuNe II with a New Adaptive Defuzzification Strategy Based on CBAD Distributions. In: Proc. EUFIT'93, Aachen (1993), 167-173
7. Höhle, U.: M-Valued Sets and Sheaves over Integral Commutative Monoids. In: Rodabough, S.E., Klement, E.P., Höhle, U. (eds.): Applications of Category Theory to Fuzzy Subsets. Kluwer, Dordrecht (1992), 33-72
8. Höppner, F., Klawonn, F., Kruse, R.: Fuzzy–Clusteranalyse: Verfahren für die Bilderkennung, Klassifikation und Datenanalyse (in German). Vieweg, Braunschweig (1997)
9. Karr, C.: Fuzzy Control of pH Using Genetic Algorithms. IEEE Trans. Fuzzy Systems 1 (1993), 46-53
10. Kinzel, J., Klawonn, F., Kruse, R.: Modifications of Genetic Algorithms for Designing and Optimizing Fuzzy Controllers. Proc. First IEEE Conf. on Evolutionary Computation. IEEE, Piscatawy, NJ (1994), 28-33
11. Klawonn, F.: Fuzzy Sets and Vague Environments. Fuzzy Sets and Systems 66 (1994), 207-221
12. Klawonn, F.: Prolog Extensions to Many-Valued Logics. In: Höhle, U., Klement, E.P. (eds.): Non-Classical Logics and their Applications to Fuzzy Subset. Kluwer, Dordrecht (1995), 271–289
13. Klawonn, F., Kruse, R.: A Łukasiewicz Logic Based Prolog. Mathware and Soft Computing 1 (1994), 5-29
14. Klawonn, F., Kruse, R.: Clustering Methods in Fuzzy Control. In: Gaul, W., Pfeifer, D. (eds.): From Data to Knowledge: Theoretical and Practical Aspects of Classification, Data Analysis and Knowledge Organization. Springer-Verlag, Berlin (1995), 195-202
15. Krishnapuram, R., Keller, J.: A Possibilistic Approach to Clustering. IEEE Trans. on Fuzzy Systems 1 (1993), 98-110
16. Kruse, R., Gebhardt, J., Klawonn, F.: Foundations of fuzzy systems. Wiley, Chichester (1994)
17. Lee, M., Takagi, H.: Integrating Design Stages of Fuzzy Systems Using Genetic Algorithms. Proc. 2nd IEEE Intern. Conf. on Fuzzy Systems. IEEE, Piscatawy, NJ (1993), 612-617
18. Mundici, D.: Ulam Games, Łukasiewicz Logic, and AF C^*-Algebras. Fundamenta Informaticae 18 (1993), 151-161
19. Nauck, D., Klawonn, F., Kruse, R.: Foundations of Neuro-Fuzzy Systems. Wiley, Chichester (1997)
20. Nauck, D., Kruse, R.: A Fuzzy Neural Network Learning Fuzzy Control Rules and Membership Functions by Fuzzy Error Backpropagation. Proc. IEEE Intern. Conf. on Neural Networks 1993, San Francisco (1993), 1022-1027
21. Nauck, D., Kruse, R.: NEFCLASS – A Neuro-Fuzzy Approach for the Classification of Data. In: George, K., Carrol, J.H., Deaton, E., Oppenheim, D., Hightower, J. (eds.): Proc. 1995 ACM Symposium on Applied Computing, ACM Press, New York (1995), 461-465
22. Novák, V.: On the Syntactico-Semantical Completeness of First Order Fuzzy Logic, Part I: Syntax and Semantics. Kybernetica 26 (1990), 47-66
23. Pavelka, J.: On Fuzzy Logic I, II, III. Zeitschr. Math. Logik Grundl. Math. 25 (1979), 45-52, 119-134, 447-464
24. Sugeno, M., Yasukawa, T.: A Fuzzy-Logic-Based Approach to Qualitative Modeling. IEEE Trans. Fuzzy Systems 1 (1993), 7-31