Extracting Heuristically Acceptable Information from Fuzzy/Neural Architectures via Heuristic Constraint Enforcement, Part I: Foundation

Mo-Yuen Chow  IEEE Senior Member
Sinan Altug  IEEE Student Member
H. Joel Trussell  IEEE Fellow

Department of Electrical and Computer Engineering
North Carolina State University
Raleigh, NC 27695-7911

Abstract
Knowledge extraction from systems where the existing knowledge is limited is a difficult task. Using fuzzy/neural architectures to extract heuristic information from systems has received increasing attention. In most cases, using output error measures to validate extracted knowledge is not sufficient; extracted knowledge may not make heuristic sense even if the output error may meet the specified criterion. Using the principles of set theoretic estimation, this paper proposes a method for enforcing heuristic constraints on the membership functions of fuzzy/neural architectures. The proposed method ensures that the final membership functions conform to a priori heuristic knowledge. Although the method is described on a specific fuzzy/neural architecture, it is applicable to other realizations of fuzzy inference systems including adaptive or static implementations. The organized yet flexible characteristic of the heuristic constraint enforcement method enables its application to a wide range of problems.

1. Introduction

Using fuzzy/neural architectures to extract heuristic information from systems have received increasing attention recently. For this purpose, fuzzy/neural architectures and knowledge extraction methods have been analyzed [1-6]. In this paper, a method to incorporate a priori information about the extracted knowledge into the training and knowledge extraction procedure for fuzzy/neural architectures is proposed. This method is called the heuristic constraints enforcement method. The foundations of the proposed method are given in part I. The techniques for implementation and integration into the training are given in part II, together with application of the proposed method to a motor fault detection problem.

1.1. Issues Regarding Information Extraction via Fuzzy/Neural Architectures

The aim of using a fuzzy/neural architecture (FZ/NN) is to find, through training, a fuzzy inference system (FIS) which represents the underlying system, while also satisfying an error criterion. This fuzzy inference system can be visualized as being composed of i) rules, ii) fuzzy sets used in these rules, and iii) fuzzy inference system specifications (e.g. method of intersection). This conceptual fuzzy inference system is implemented using an architecture equivalent to a neural network. The conceptual problem of finding membership functions and rules of the underlying FIS can be interpreted as finding parameters that represent the membership functions and the relative importance of the rules of the FIS, respectively. The training algorithms of fuzzy/neural architectures do not involve any information about the heuristic expectations about the solution in terms of membership functions and rules. Therefore the extracted knowledge may not make heuristic sense even if the output error may meet the specified criteria. Using the heuristic constraint enforcement method proposed in this paper, a priori information about the expectations from the solution can be incorporated into the training and information extraction procedures, yielding heuristically acceptable solutions.

1.2. Set Theoretic Framework

Set theoretic estimation has been used in a wide range of research areas, and has proven to be an effective tool to incorporate information into numerous problems [7-10]. In the set theoretic formulation of a problem, every piece of information, including information about the system, the solution and external factors are represented with sets. The intersection of all these sets form the set or the family of acceptable solutions, referred as the feasibility set [7]. Therefore, in set theoretic estimation, the optimal solution

0-7803-4859-1/98 $10.00©1998 IEEE 1250
is not sought, rather, a set of solutions is defined as a class of objects that is consistent with all the information searched for. The incorporated information includes information acquired from the observed data as well as a priori information [10]. One of the most powerful features of set theoretic formulation is the fact that the ‘solution space’ may be in various different forms depending on the object that is being estimated [10].

Using principles of set theoretic estimation, a methodology for enforcing heuristic constraints on membership functions for information extraction from a fuzzy/neural architecture is developed in this paper. Moreover, a procedure for integrating these constraints into the training of the fuzzy/neural network is also proposed, so that the extracted membership functions are guaranteed to conform to the heuristic knowledge that is embodied into the formulation in the form of constraint sets. Also, with this method, the FZ/NN training is carried out only in the region where the constraints are satisfied, which is likely to improve convergence.

The foundations of the proposed method are given in this paper. Techniques for implementation and incorporation of the method into the training algorithm of the fuzzy/neural architecture, as well as experimental results are given in Part II of this paper.

2. Description of the Fuzzy/Neural Architecture and Space Definitions

While the proposed method of constraint enforcement is not architecture dependent, the FZ/NN utilized in Parts I and II of this paper is based on the architecture described in [4, 11]. The FZ/NN is a fuzzy inference system implemented on an adaptive network structure. The inputs of the FZ/NN are defined on $\Omega_a$ and the outputs are defined on $\Omega_y$, with the input vector as $x = [x_1, \ldots, x_n]^T$ and the corresponding output vector as $y = [y_1, \ldots, y_m]^T$. There exist two types of parameters for the FZ/NN architecture: membership function parameters, $p$, used to parameterize the membership functions, and the rule parameters, $q$, associated with the relative importance of the rules and represented by a layer of connection weights for the underlying FIS. The corresponding Membership Function Parameter Space and Rule Parameter Space are represented as $\Omega_p$ and $\Omega_q$, respectively.

Gaussian type of membership functions are used for the fuzzy sets defined in the form:

$$\mu_j^i (x, a, b) = e^{-\frac{(x-a)^2}{2b^2}}$$  \hspace{1cm} (1)

where the parameters $a$ and $b$ represent the mean and standard deviation of the function, respectively, and $c$ is the scaling factor which is assumed to be equal to one. Thus, each membership function can be represented with the parameter pair $(a, b)$ in the parameter vector $p$.

The membership functions using the above parameters are represented in Membership Function Space $\Omega_m$. A total of $M_j$ fuzzy sets are defined on each of the input and output subspaces $\Omega_a$, and labeled as $X_j^i$, where $j = 1, \ldots, M_j$. Each $X_j^i$ has a corresponding membership function $\mu_j^i$, where

$$\mu_j^i : \Omega_a \rightarrow [0,1], \text{ for } j = 1, \ldots, M_j. \hspace{1cm} (2)$$

Using the index set $J = \{1, 2, \ldots, M_j\}$, for each membership function $\mu_j^i$, $j \in J$, the set of points where the membership function attains its maximum is labeled as $x_j^i$:

$$\mu_j^i (x_j^i) = \max_x \{ \mu_j^i (x) \}, \forall j \in J. \hspace{1cm} (3)$$

In this context, the FZ/NN can be visualized as a mapping $F : \Omega_x \times \Omega_p \times \Omega_q \rightarrow \Omega_y$, which maps the input $x$ to $\hat{y}$, the estimate of $y$, using the membership function parameters $p$ and rule parameters $q$:

$$\hat{y} = F(x, p, q). \hspace{1cm} (4)$$

For any input $x$, the fuzzy neural output estimate $\hat{y}$ is obtained using $(p, q)$ through (4).

The training is composed of two main parts. Following initialization of the membership functions (representing fuzzy partitions), and connection weights (representing rules), a modified competitive learning algorithm is applied to extract the rule parameters, $q$. The second part of the training is an optimal adjustment of the membership function parameters $p$, with an effort to minimize the error measure defined on the output. A gradient descent algorithm is used to update both antecedent and consequence membership function parameters. For details the reader should refer to [3, 11].

The FZ/NN architecture mimics a fuzzy inference system, yet the FZ/NN has extra degrees of freedom through continuous values of $q$ used to represent the rule structures. Any FIS can be represented through the FZ/NN. Higher values of $p$ for FZ/NN indicate the relative importance of the corresponding rule in the underlying rulebase, whereas in a conventional rulebase where each rule is considered to have equal importance, or $q=1$, for all antecedent-consequence combinations.

As mentioned, the training of the FZ/NN is equivalent to adapting parameters $p \in \Omega_p$ and $q \in \Omega_q$. These parameters are then used to implement membership functions and the connection weights corresponding to rule structures in the FZ/NN. The conceptual diagram of the FZ/NN architecture is depicted in Fig. 1, where $\Omega_a$, $\Omega_p$, $\Omega_m$, $\Omega_q$, and $\Omega_k$ represent the topographical locations of the elements in input, output, membership function,
membership function parameter and rule parameter spaces, respectively.

Fig. 1. The conceptual diagram of a) the fuzzy/neural architecture, b) the fuzzy inference system.

3. Enforcing Heuristic Constraints on Membership Functions

As discussed above, fuzzy/neural architectures (FZ/NN) are utilized to find, through training, a fuzzy inference system (FIS) which represents the underlying system, while also satisfying an error criterion. In most of the cases, using output error measures to validate extracted information is not sufficient, as extracted knowledge may not make heuristic sense, even if the output error may meet the specified criteria.

On the other hand, although it is assumed that a priori information about the rulebase does not exist, information about heuristic expectations from the general properties of the solution, may still be available. Via heuristic constraint enforcement method, such information is incorporated into the FZ/NN architecture.

The aim of enforcing heuristic constraints is to achieve heuristically acceptable solutions for membership functions in the input and output spaces, while also satisfying defined error criteria in the output space. The enforcement of constraints on membership functions, when integrated into the process of finding a rulebase, results in heuristically acceptable solutions for the membership functions representing fuzzy sets.

The process of incorporating heuristic information include the following steps:

i. Defining the heuristic constraints from a priori knowledge;
ii. Constructing constraint sets corresponding to the constraints;
iii. Enforcing constraints to find acceptable solutions;
iv. Integrating the constraint enforcement technique into the FZ/NN architecture.

These steps will be discussed in detail parts I and II.

3.1. Defining Heuristic Constraints on Membership Functions

The information about the properties of the solution will be used in determining whether the solution is acceptable. As described in [10], a priori information about the properties of a solution can be used to determine the acceptability of a solution. Analogously, we define constraints related to heuristic expectations from the fuzzy partitions of the input and output spaces; each of the constraints reflect a piece of information that is used to evaluate the heuristic acceptability of the extracted information. For example, in a fuzzy set labeled as "low", if the grade of membership of \( x_i \) is \( \mu_s \), it is expected that the grade of membership of a point of lower value \( x_i < x_a \) is higher than the grade of membership of \( x_i \), or that \( \mu_b > \mu_s \). This expectation is due to our heuristic understanding of the low label. In this manner, we express each piece of a priori information in mathematical form, and call them as constraints.

Using a priori information, the following constraints are defined in this paper to illustrate the proposed method. However, it is possible to add or remove constraints according to the requirements from the solution, which are determined by the nature of the specific problem at hand. The constraint sets corresponding to each of the following constraints are given in part II.

1) Existence of prototype point constraint: At least one prototype point [12] exists for each fuzzy set on \( X_i \), which can be used to describe each fuzzy set. In other words, each membership function attains a maximum value of 1 in the universe of discourse:

\[
\max_{x_i \in \Omega_{x_i}} (\mu_i^j (x_i)) = 1; \forall j \in J. \tag{C1}
\]

2) Convexity constraint: The membership functions must be convex, following Zadeh's definition [13]. That is, the membership functions should be either non-decreasing, non-increasing, or have a non-decreasing portion followed by a non-increasing portion:

\[
x_a \leq x_r \leq x_b \Rightarrow \min \{ \mu_i^j (x_a), \mu_i^j (x_b) \} \leq \mu_i^j (x_r); \\
\forall x_a, x_b, x_r \in \Omega_{x_i}; \forall j \in J. \tag{C2}
\]
3) **Leftmost membership function constraint:** The "leftmost" membership function, i.e., the membership function with the smallest \( x_i \) value as given in (3), is required to attain its maximum at the smallest value of \( x_i \) in \( \Omega_{x_i} \), which is \( x_{iL} \); i.e., maximum occurs at the smallest input value:

\[
x_i^L = x_{iL}; \forall x_i^L : \exists(x_i^L < x_i^L) \land (j,k \in J, k \neq j);
\]  

(C3)

4) **Rightmost membership function constraint:** The "rightmost" membership function, or the membership function with the largest \( x_i \) value given in (3), is required to attain its maximum at the largest value of \( x_i \) in \( \Omega_{x_i} \), which is \( x_{iU} \):

\[
x_i^r = x_{iU}; \forall x_i^r : \exists(x_i^r > x_i^r) \land (j,k \in J, k \neq j);
\]  

(C4)

Rightmost and leftmost membership function constraints are defined for the purpose of eliminating occurrences similar to the example discussed in the preceding section. Several membership functions that do not agree with constraints C1, C2, C3 or C4 are shown in Fig. 2.

![Membership functions violating constraints C1–C4.](image)

**Fig. 2. Membership functions violating constraints C1–C4.**

5) **Overlap constraint:** The overlap between the fuzzy sets should be in an interval \([L,U]\):

\[
0 < L \leq \frac{\text{diam}(X_i^L \cap X_i^U)}{\text{diam}(X_i^L)} \leq U < 1
\]

\[\forall j, j+1 \in J \text{ or } j, j-1 \in J; \quad \text{(C5)}\]

where \( j \) and \( j+1 \) are indices representing adjacent fuzzy sets, \( X_i^L \) is the \( \alpha \)-cut set of \( X_i^L \), and \( \text{diam} \) is the diameter function \([10]\) for the crisp \( \alpha \)-cut sets. (C5) is the overlap constraint for \( X_i^L \). It is noted that the choice of \( L \) and \( U \) is important.

The regions corresponding to the sets given in (C5) are depicted in Fig. 3. Very high values of \( U \) (or very small values of \( L \)) corresponding to large (or small) overlap between adjacent membership functions may be undesirable in certain system modeling and control applications \([14]\). On the other hand, for specific problems, the information about low (or high) overlap may be used to make inferences about necessity of architectural changes (e.g., adding or removing membership functions). Consequently, for problems where enforcing a lower and/or upper limit on the overlap is not advantageous, this constraint may be relaxed.

![The regions of interest for the overlap constraint.](image)

**Fig. 3. The regions of interest for the overlap constraint.**

6) **Characterization constraint:** Every point \( x_i \) in the universe of discourse \( X_i \), must have a grade of membership of at least \( \mu_{\text{min}}>0 \), in at least one of fuzzy sets \( X_i^L \) defined on \( X_i \). So that every point in the universe can be characterized using (at least) one linguistic label attached to the fuzzy sets, i.e., every point in the universe of discourse can be considered to be a member of at least one fuzzy set with a minimum grade of membership of \( \mu_{\text{min}} \):

\[
\min_{j \in J} \mu_j^L(x_i) = \mu_{\text{min}}; \forall x_i \in X_i.
\]  

(C6)

Fig. 4 presents a region R in \( X_i \) where (C6) is violated. Such regions are not desirable in general, because the elements in R cannot be characterized as an element of any fuzzy set defined in this universe of discourse.
The elements of the intersection of these two sets are depicted Fig. 5c. As the membership functions must agree with both constraints (C1) and (C2), each membership function must be in both $C_1$ and $C_2$. Consequently, it is concluded that any element that agrees with each constraint must actually be a member of the intersection of all the constraint sets, or equivalently, $\mu \in \bigcap_{i=1}^{S} C_i$.

![Figure 4](image1.png)

Fig. 4. A region $R \subset X_1$, where $x_i \in R$ violates $C6$.

The above list of constraints is subjective. According to the expectations from the solution, new constraints can be added, or some constraints can be deleted. When defining the constraints, it must be remembered that the more constraints enforced, the more a priori heuristic information is being incorporated into the problem.

3.2. Defining Constraints on the Solution in $\Omega_p$

The magnitude of the error in the output space depends on the membership function parameters, as the output of the FZ/NN is a function of the parameters. The definition of the total solution set in the membership function parameter space, the elements of which provide a specified input-output mapping within a specified accuracy, is given as follows:

**Definition (Total solution set)** Given a mapping $(x,y)$, and the rule parameter vector $q$, the total solution set on $\Omega_p$ is defined as $\Theta_{\mu, \varepsilon} \subset \Omega_p$, the elements of which satisfy the output criterion set forth on $\hat{y}$ in the output space $\Omega_q$:

$$\Theta_{\mu, \varepsilon} = \{ p \mid || F(x, p, q) - y || \leq \varepsilon ; (x,y) \text{ in training set} \}. \tag{5}$$

The size of the total solution set depends on the choice of $\varepsilon$. The FZ/NN training strategy by itself does not guarantee convergence for arbitrary $\varepsilon$. Furthermore, a $p \in \Theta_{\mu, \varepsilon}$ satisfying (5) is not guaranteed to conform to the heuristic knowledge in general. The motive for enforcing constraints is to extract information from FZ/NN that conforms to a priori information (e.g., heuristics about the problem), while satisfying (5).

3.3. Constraint Sets and Candidate Solutions

Each constraint defines a constraint set of elements satisfying the constraint, indicated by $C_i$, $1 \leq i \leq S$, where $S$ is the number of constraints. Several candidate elements of $C_i$, defined by constraint (C1), are depicted in Fig. 5b.

![Figure 5](image2.png)

Fig. 5. Several elements of the constraint sets a) $C_1$ b) $C_2$, c) $C_1 \cap C_2$.

**Definition (Candidate Solution Set)** Following the definition of [7], we define a candidate solution in the membership function parameter space $\Omega_p$ as a $p \in \Omega_p$, which conforms to all the heuristic knowledge incorporated in the form of constraints. The candidate solution set $\Gamma$ is the set that contains all such elements in $\Omega_p$, or $\Gamma = \bigcap_{i=1}^{S} C_i$.

Ideally, the candidate solution set $\Gamma$ is a subset of the total solution set $\Theta_{\mu, \varepsilon}$ so that any solution in $\Gamma$ is also in the total solution set $\Theta_{\mu, \varepsilon}$:

$$\Gamma \subset \Theta_{\mu, \varepsilon} \subset \Omega_p. \tag{6}$$

However, we cannot generally assume that (6) holds. Moreover, given any $\varepsilon$, convergence is not guaranteed; even if convergence is achieved, the point of convergence is not guaranteed to be in $\Theta_{\mu, \varepsilon}$. Still, although $p \in \Gamma$ may not be in $\Theta_{\mu, \varepsilon}$ every element $p \in \Gamma$ is a candidate solution. Thus, if a choice of $\varepsilon$ enables convergence in $\Theta_{\mu, \varepsilon}$, then the training with heuristic constraint enforcement is ensured to converge to a candidate solution in $\Theta_{\mu, \varepsilon}$. Because every element of the candidate solution set $\Gamma$ complies with all constraints, every element of $\Gamma$ is also an element of all the constraint sets, $C_i$, $1 \leq i \leq S$. Consequently, the problem of finding a point $p \in \Omega_p$ that satisfies all the constraints is actually equivalent to finding a common point of the constraint sets $C_i$. 1254
4. Discussion

A novel method of incorporating heuristic information into the training of and information extraction from fuzzy neural architectures was presented in this paper. The method, called *heuristic constraint enforcement*, leads to extraction of heuristically acceptable information from fuzzy neural architectures. The proposed method is not problem or architecture dependent, it can be applied to other realizations of fuzzy inference systems including adaptive or static implementations. The collection of constraints is problem specific, and is chosen according to the particular requirements from the solution for any specific problem. The flexibility of the method enables the application of it to a wide range of problems. Part II of this paper presents method of constructing constraint sets from the constraints, finding the candidate solutions, and the techniques of integrating the enforcement methodology with the training of the fuzzy neural architecture.

5. Acknowledgments

The authors of this paper would like to acknowledge the support of National Science Foundation grant ECS-9521609 and Electric Power Research Center grant 5-39907.

6. References


