A Method for Fuzzy Rules Extraction Directly from Numerical Data and Its Application to Pattern Classification

Shigeo Abe, *Senior Member, IEEE*, and Ming-Shong Lan

Abstract—**In this paper, we discuss a new method for extracting fuzzy rules directly from numerical input–output data for pattern classification. Fuzzy rules with variable fuzzy regions are defined by activation hyperboxes which show the existence region of data for a class and inhibition hyperboxes which inhibit the existence of data for that class. These rules are extracted from numerical data by recursively resolving overlaps between two classes. Then, optimal input variables for the rules are determined using the number of extracted rules as a criterion. The method is compared with neural networks using the Fisher iris data and a license plate recognition system for various examples.**

I. Introduction

In theory, neural networks, and fuzzy systems are equivalent in that they are convertible [1], yet in practice each has its own advantages and disadvantages. For neural networks, knowledge is automatically acquired by the backpropagation algorithm [2], [3], but the learning process is relatively slow and analysis of the trained network is difficult. On the other hand, since the input space of fuzzy systems must be divided into fuzzy regions, it is very difficult to apply fuzzy systems to problems in which the number of input variables is large. Moreover, another difficulty arises in knowledge acquisition through interviewing experts. For fuzzy systems, however, once the knowledge is acquired, how the systems work is relatively easily understood. To fill the gap of knowledge acquisition between the two technologies, several methods for extracting fuzzy rules from numerical data have been developed. One approach uses neural networks to extract fuzzy rules. For example, in [4], neural networks are extended to automatically extract fuzzy rules from numerical data. The major restriction of this method is that the number of divisions of each input variable must be defined in advance. (Division of an output variable is also necessary. But this is not a disadvantage since it can be determined easily based on the approximation accuracy.) Another approach extracts fuzzy rules directly from numerical data. For example, in [5], fuzzy rules are derived by dividing the input space into fuzzy regions and the output space into regions, and by determining in which fuzzy region each numerical input datum is included and in which output region the corresponding output datum is included. Although the above approach is very straightforward, again the input space needs to be divided in advance.

In [6], fuzzy rules with variable fuzzy regions are extracted for classification problems. This approach solves the problems of fuzzy systems described above. The input region of each class is represented by a set of hyperboxes, in which overlaps among hyperboxes for the same class are allowed, but no overlaps are allowed between different classes. If a datum is in the hyperbox belonging to a class, it is judged to be that class. The learning algorithm dynamically expands, split and contract hyperboxes when several classes overlap.

Our study aims at developing a fuzzy classification system which:

1) has a classification ability comparable to that of neural networks with much less computational burden of learning; and,
2) can handle large-scale classification problems.

In this paper, we discuss how to extract fuzzy rules with variable fuzzy regions directly from numerical data. First, we define fuzzy rules for pattern classification recursively; each rule is composed of an activation hyperbox which defines the existence region of a class and, if necessary, an inhibition hyperbox which inhibits the existence of data in that activation hyperbox. Then, we discuss their inference mechanism. Furthermore, we discuss how to delete redundant input variables based on the number of fuzzy rules created. Finally, we apply the fuzzy rule classifier to the Fisher iris data and a license plate recognition system in which the classification power is compared with that of neural networks.

II. Classification by Fuzzy Rules with Variable Fuzzy Regions

Representation of an existence region of data for a class by a set of hyperboxes which was discussed in [6] is efficient to handle classification problems with a large number of input variables. But since only one type of hyperbox was defined in that approach, hyperboxes between different classes cannot overlap, although overlaps of hyperboxes among the same class are allowed. Thus to resolve overlaps between different classes, compaction or splitting of hyperboxes is necessary.

To resolve overlaps between different classes, we introduce two types of hyperboxes: activation hyperboxes which define the existence regions for classes, and inhibition hyperboxes which inhibit the existence of data within the activation hy-

Manuscript received November 6, 1992; revised April 24, 1994.

S. Abe is with the Hitachi Research Laboratory, Hitachi, Ltd., 7-1-1 Omika, Hitachi, Ibaraki 319-12 Japan.

M.-S. Lan is with Asahi Diamond Industrial Co., Ltd., The New Otani Garden Court, 4–1, Kioi-cho, Chiyoda-ku, Tokyo 102 Japan.

IEEE Log Number 9406654.

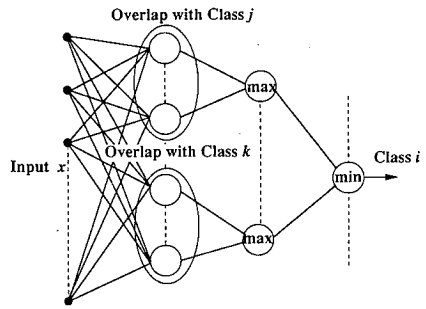

Fig. 1. Architecture of a fuzzy classification system (only the network for class i is shown).

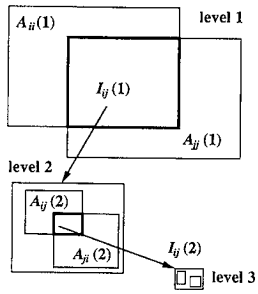

Fig. 2. Recursive definition of activation and inhibition hyperboxes.

perboxes. These hyperboxes are defined recursively. Namely, first we determine activation hyperboxes by calculating the minimum and maximum values of data for each class. If the activation hyperbox for class i overlaps with the activation hyperbox for class j, the overlapping region is defined as an inhibition hyperbox. If in the inhibition hyperbox data for classes i and j exist, we define additional activation hyperboxes for these classes. Again, if an overlap exists between these activation hyperboxes, we further define the overlapping region as the inhibition hyperbox. In this way the overlap of activation hyperboxes is resolved recursively.

We define a fuzzy rule based on an activation hyperbox or based on an activation hyperbox and its corresponding inhibition hyperbox, if generated. A neural network-like inference architecture for our proposed fuzzy system is shown in Fig. 1 in which only the portion for class i is shown for simplicity. Different classes have different numbers of units for the second to fourth layers and there is no connection among units of different classes. The second layer units consist of fuzzy rules and they calculate the degrees of membership for an input vector $\mathbf{x}$. The third layer units take the maximum values of inputs from the second layer, which are the degrees of membership generated by resolving overlaps between two classes. The number of third layer units for class i is determined by the number of classes that overlap with class i. Therefore, if there is no overlap between class i and any other classes, the network for class i reduces to two layers. The fourth layer unit for class i takes the minimum value among the maximum values; each of them is associated with a two-class overlap. Therefore, if class i overlaps with only one class, the network for class i reduces to the three layers; in other words, the "min" node in the fourth layer is not necessary. Calculation of a minimum in the fourth layer resolves overlaps among more than two classes. Thus in the process of generating hyperboxes, we need to resolve only the overlap between two classes. (The validity of the architecture shown in Fig. 1 is fully discussed in Section II-B.)

A. Fuzzy Rule Representation

This section describes how to create fuzzy rules based on a set of input data X_i for class i, where $i = 1, \cdots, n$, for classifying an m-dimensional input vector $\mathbf{x}$ into one of n classes. First, using X_i, an activation hyperbox of level 1, denoted as A_{ii} (1), is defined, which is the maximum region of class i data

$$A_{ii}(1) = \{\mathbf{x} \mid v_{iik}(1) \le x_k \le V_{iik}(1), \quad k = 1, \cdots, m\} \quad (1)$$

where

x_k: the kth element of input vector $\mathbf{x}$;

$v_{iik}(1)$: the minimum value of x_k for $\mathbf{x} \in X_i$; and

$V_{iik}(1)$: the maximum value of x_k of $\mathbf{x} \in X_i$.

If there is no overlap between activation hyperboxes $A_{ii}(1)$ and $A_{jj}(1)$ $(j \neq i, j = 1, \cdots, n)$, we obtain a fuzzy rule of level 1 for class i as follows

$$\text{If } \mathbf{x} \text{ is } A_{ii}(1) \text{ then } \mathbf{x} \text{ is class } i. \quad (2)$$

If the activation hyperboxes $A_{ii}(1)$ and $A_{jj}(1)$ overlap, we resolve the overlap recursively as illustrated in Fig. 2 in which we define the overlapping region as the inhibition hyperbox of level 1 denoted as $I_{ij}(1)$

$$I_{ij}(1) = \{\mathbf{x} \mid w_{ijk}(1) \le x_k \le W_{ijk}(1), \quad k = 1, \cdots, m\} \quad (3)$$

where $v_{iik}(1) \le w_{ijk}(1) \le W_{ijk}(1) \le V_{iik}(1)$. The minimum and maximum values of inhibition hyperbox $I_{ij}(1)$ are given by (cf. Fig. 3)

i) For $v_{jjk}(1) \le v_{iik}(1) \le V_{jjk}(1) < V_{iik}(1)$

$$w_{ijk}(1) = v_{iik}(1), W_{ijk}(1) = V_{jjk}(1) \quad (4)$$

ii) For $v_{iik}(1) < v_{jjk}(1) \le V_{iik}(1) \le V_{jjk}(1)$

$$w_{ijk}(1) = v_{jjk}(1), W_{ijk}(1) = V_{iik}(1) \quad (5)$$

iii) For $v_{jjk}(1) \le v_{iik}(1) \le V_{iik}(1) \le V_{jjk}(1)$

$$w_{ijk}(1) = v_{iik}(1), W_{ijk} = V_{iik}(1) \quad (6)$$

iv) For $v_{iik}(1) < v_{jjk}(1) \le V_{jjk}(1) < V_{iik}(1)$

$$w_{ijk}(1) = v_{jjk}(1), W_{ijk}(1) = V_{jjk}(1). \quad (7)$$

Then we define a fuzzy rule of level 1 with inhibition by

$$\text{If } \mathbf{x} \text{ is } A_{ii}(1) \text{ and } \mathbf{x} \text{ is not } I_{ij}(1) \text{ then } \mathbf{x} \text{ is class } i. \quad (8)$$

If $A_{ii}(1)$ is included in $A_{jj}(1)$, i.e., (6) holds for all $k, k = 1, \cdots, m$, $A_{ii}(1)$ coincides with $I_{ij}(1)$. In this case (8) is a void rule, which is not created, since there is no $\mathbf{x}$ that satisfies (8).

If some data belonging to X_i exist in $I_{ij}(1)$, we define the activation hyperbox of level 2 denoted as $A_{ij}(2)$ within the

inhibition hyperbox $I_{ij}(1)$ by calculating the minimum and maximum values of x_k based on the data in $I_{ij}(1)$

$$A_{ij}(2) = \{\mathbf{x} \mid v_{ijk}(2) \le x_k \le V_{ijk}(2), \quad k = 1, \cdots, m\} \tag{9}$$

where $\mathbf{x} \in X_i$ and $\mathbf{x}$ is in $I_{ij}(1)$

$v_{ijk}(2)$: the minimum value of x_k where $\mathbf{x} \in X_i$ and $\mathbf{x}$ is in $I_{ij}(1)$

$V_{ijk}(2)$: the maximum value of x_k where $\mathbf{x} \in X_i$ and $\mathbf{x}$ is in $I_{ij}(1)$

$$w_{ijk}(1) \le v_{ijk}(2) \le x_k \le V_{ijk}(2) \le W_{ijk}(1). \tag{10}$$

If there is no overlap between the activation hyperboxes of level 2, we define a fuzzy rule of level 2 for class i by

$$\text{If } \mathbf{x} \text{ is } A_{ij}(2) \text{ then } \mathbf{x} \text{ is class } i. \tag{11}$$

If $A_{ij}(2)$ and $A_{ji}(2)$ overlap, we define the overlapping region as the inhibition hyperbox of level 2 denoted as $I_{ij}(2)$

$$I_{ij}(2) = \{\mathbf{x} \mid w_{ijk}(2) \le x_k \le W_{ijk}(2), \quad k = 1, \cdots, m\} \tag{12}$$

where $v_{ijk}(2) \le w_{ijk}(2) \le W_{ijk}(2) \le V_{ijk}(2)$. Then we define a fuzzy rule of level 2 with inhibition

$$\text{If } \mathbf{x} \text{ is } A_{ij}(2) \text{ and } \mathbf{x} \text{ is not } I_{ij}(2) \text{ then } \mathbf{x} \text{ is class } i. \tag{13}$$

Similarly, we define fuzzy rules of levels higher than two if an overlap still remains. In a general form, we define the fuzzy rule $r_{ij}(l)$ of level $l(\ge 1)$ without inhibition as follows

$$\text{If } \mathbf{x} \text{ is } A_{ij}(l) \text{ then } \mathbf{x} \text{ is class } i \tag{14}$$

where $i = j$ for $l = 1$ and $i \ne j$ for $l \ge 2$. Or we define the fuzzy rule $r_{ij}(l)$ of level l with inhibition as follows

$$\text{If } \mathbf{x} \text{ is } A_{ij'}(l) \text{ and } \mathbf{x} \text{ is not } I_{ij}(l) \text{ then } \mathbf{x} \text{ is class } i \tag{15}$$

where $j' = i$ for $l = 1$ and $j' = j$ for $l \ge 2$.

The recursion process for defining fuzzy rules terminates when there is no overlap between $A_{ij'}(l)$ and $A_{ji'}(l)$ or $A_{ij'}(l) = A_{ji'}(l) = I_{ji}(l-1)$ holds. In the latter case, since the overlap cannot be resolved by the recursive process, instead of defining $A_{ij'}(l)$ and $A_{ji'}(l)$ by (9), for each datum of class i and/or j in $I_{ij}(l-1)$ we define an activation hyperbox which includes only that datum. In this case we no longer define inhibition and activation hyperboxes with levels higher than l, because as long as no identical data exist in classes i and j, no overlap exists between the activation hyperboxes of level l.

B. Fuzzy Rule Inference

For pattern classification, a natural assumption is that the degree of membership of $\mathbf{x}$ for a fuzzy rule given by (14) is one if $\mathbf{x}$ is in the activation hyperbox $A_{ij}(l)$, and the degree of membership decreases as $\mathbf{x}$ moves away from the activation hyperbox. Namely, if all the input variables are normalized to the same scale, e.g., [0, 1], the contour surface, which has the same degree of membership, is parallel to, and lies at an equal distance from the surface of the activation hyperbox as shown in Fig. 4. To realize a membership function with this

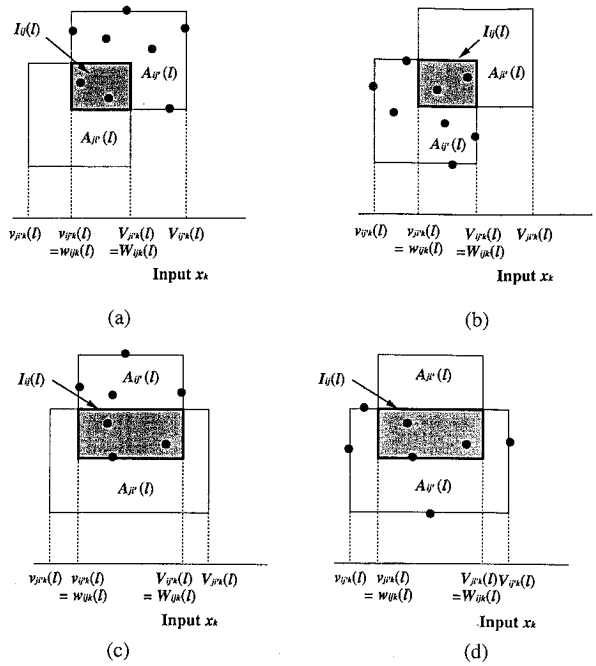

Fig. 3. Definition of activation and inhibition hyperboxes ($j' = i$ and $i' = j$ for $l = 1$, $j' = j$ and $i' = i$ for $l \ge 2$).

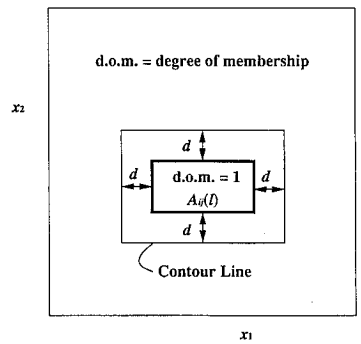

Fig. 4. The contour line of membership function for the activation hyperbox (two-dimensional case).

characteristic we use the following function which is similar to that proposed in [6]

$$m_{A_{ij}(l)}(\mathbf{x}) = \min_{k=1,\cdots,m} m_{A_{ij}(l)}(\mathbf{x}, k), \tag{16}$$

$$m_{A_{ij}(l)}(\mathbf{x}, k) = [1 - \max(0, \min(1, \gamma_k(v_{ijk}(l) - x_k)))] \times [1 - \max(0, \min(1, \gamma_k(x_k - V_{ijk}(l))))] \tag{17}$$

where γ_k is the sensitivity parameter for the kth input variable x_k. Even if we normalize all the input variables, it is still possible to tune the recognition rate by setting different values for each sensitivity parameter γ_k. But in the following we assume that x_k's are normalized and $\gamma_k = \gamma$ for $k = 1, \cdots, m$. The one-dimensional membership function $m_{A_{ij}(l)}(\mathbf{x}, k)$ becomes as shown in Fig. 5. The parameter γ serves to control the generalization region.

In the following, we show that (16) and (17) give a contour surface which is parallel to, and lies at an equal distance from the surface of the activation hyperbox. For $\mathbf{x} = (x_1, \cdots, x_m)$,

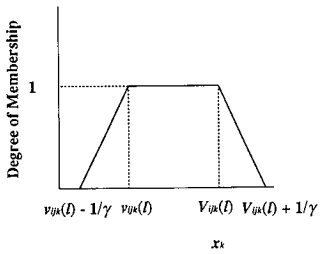

Fig. 5. One-dimensional membership function of the activation hyperbox $A_{ij}(l)$.

if

$$v_{ijk}(l) \leq x_k \leq V_{ijk}(l) \tag{18}$$

holds $m_{A_{ij}(l)}(\mathbf{x}, k) = 1$. Thus, the degree of membership is determined by x_k whose $\min(|v_{ijk}(l) - x_k|, |x_k - V_{ijk}(l)|)$ is the maximum among those of $x_1, \cdots x_m$ which do not satisfy (18). Now without loss of generality we can assume that $x_k, k = 1, \cdots, k'$ satisfy (18) while $x_k, k = k'+1, \cdots, m (m - k' - 1 > 0)$ do not satisfy (18) but they satisfy

$$\min(|v_{ijk}(l) - x_k|,\ |x_k - V_{ijk}(l)|) = d (> 0). \tag{19}$$

From (16) and (17), $m_{A_{ij}(l)}(\mathbf{x}) = 1 - \gamma d$. Then the degree of membership $m_{A_{ij}(l)}(\mathbf{x})$ does not change even if any x_k is changed between

$$v_{ijk}(l) - d \leq x_k \leq V_{ijk}(l) + d \tag{20}$$

which means that the contour surface is parallel to, and lies at an equal distance from the surface of the activation hyperbox.

Thus the degree of membership of x for a fuzzy rule $r_{ij}(l)$ given by (14) is

$$d_{r_{ij}(l)}(\mathbf{x}) = m_{A_{ij}(l)}(\mathbf{x}). \tag{21}$$

The degree of membership of $\mathbf{x}$ for a fuzzy rule given by (15) is 1 when $\mathbf{x}$ is in the activation hyperbox but not within the inhibition hyperbox, i.e., $\mathbf{x}$ is in $\overline{A_{ij'}(l) - I_{ij}(l)}$, where $\overline{S}$ denotes the closure of set S and $j' = i$ for $l = 1$ and $j' = j$ for $l \geq 1$. If $\mathbf{x}$ moves away from this region the degree of membership decreases. Namely, in this case it is also favorable that the contour surface is parallel to, and lies at an equal distance from the surface of $\overline{A_{ij'}(l) - I_{ij}(l)}$ as shown in Fig. 6. (If $A_{ij'}(l) = I_{ij}(l)$, i.e., if the rule is void, we do not calculate the degree of membership for this rule.) To realize this membership function we first define the region $H_{ij}(l)$ associated with $A_{ij'}(l)$ and $I_{ij}(l)$ as follows (c.f. Fig. 3)

$$\begin{aligned} H_{ij}(l) = \\ \{\mathbf{x} \mid\ & x_k \leq W_{ijk}(l) && \text{for } v_{ji'k}(l) \leq v_{ij'k}(l) \leq V_{ji'k}(l) \\ & && \quad < V_{ij'k}(l), \\ & x_k \geq w_{ijk}(l) && \text{for } v_{ij'k}(l) < v_{ji'k}(l) \leq V_{ij'k}(l) \\ & && \quad \leq V_{ji'k}(l), \\ & -\infty < x_k < \infty && \text{for } v_{ji'k}(l) \leq v_{ij'k}(l) \leq V_{ij'k}(l) \\ & && \quad \leq V_{ji'k}(l), \\ & w_{ijk}(l) \leq x_k \leq W_{ijk}(l) && \text{for } v_{ij'k}(l) < v_{ji'k}(l) \leq V_{ji'k}(l) \\ & && \quad < V_{ij'k}(l), \\ & && \quad k = 1, \cdots, m\} \end{aligned} \tag{22}$$

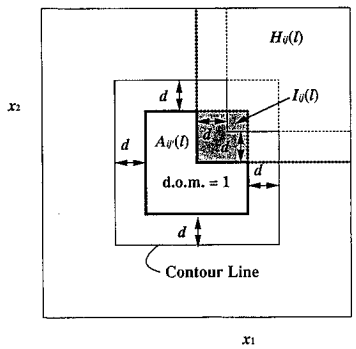

Fig. 6. The contour line of membership function for the activation and inhibition hyperboxes (two-dimensional case).

where $j' = i$ and $i' = j$ for $l = 1, j' = j$ and $i' = i$ for $l \geq 2$, and $H_{ij}(l)$ and $H_{ji}(l)$ are in general different. According to the definition

$$H_{ij}(l) \supset I_{ij}(l). \tag{23}$$

The region $H_{ij}(l)$ defines an input region where the inhibition hyperbox affects the degree of membership of the rule given by (15). If $\mathbf{x} \notin H_{ij}(l)$, the degree of membership for a fuzzy rule $r_{ij}(l)$ given by (15) is the same as (21). Then for $\mathbf{x} \in I_{ij}(l)$ the degree of membership $m_{I_{ij}(l)}(\mathbf{x})$ is given by

$$m_{I_{ij}(l)}(\mathbf{x}) = \max_{k=1,\cdots,m} m_{I_{ij}(l)}(\mathbf{x}, k) \tag{24}$$

where $m_{I_{ij}(l)}(\mathbf{x}, k)$ is the degree of membership of x_k and is calculated by:

1) For $v_{ji'k}(l) \leq v_{ij'k}(l) \leq V_{ji'k}(l) < V_{ij'k}(l)$ (c.f. Figs. 3(a) and 7(a))

$$m_{I_{ij}(l)}(\mathbf{x}, k) = 1 - \max(0, \min(1, \gamma(W_{ijk}(l) - x_k))) \tag{25}$$

2) For $v_{ij'k}(l) < v_{ji'k}(l) \leq V_{ij'k}(l) \leq V_{ji'k}(l)$ (c.f. Figs. 3(b) and 7(b))

$$m_{I_{ij}(l)}(\mathbf{x}, k) = 1 - \max(0, \min(1, \gamma(x_k - w_{ijk}(l)))) \tag{26}$$

3) For $v_{ji'k}(l) \leq v_{ij'k}(l) \leq V_{ij'k}(l) \leq V_{ji'k}(l)$.
Since $x_k = v_{ij'k}(l)$ and $x_k = V_{ij'k}(l)$ do not constitute the surface of $\overline{A_{ij'}(l) - I_{ij}(l)}$, we need not define the membership in the x_k axis. Thus we get

$$m_{I_{ij}(l)}(\mathbf{x}, k) = 0. \tag{27}$$

Equation (27) holds for all $k, k = 1, \cdots, m$ only when $A_{ji'}(l) \supset A_{ij'}(l) = I_{ij}(l)$, which corresponds to a void rule. Thus, the x_k axis is neglected when calculating the degree of membership using (27) and (24).

4) For $v_{ij'k}(l) < v_{ji'k}(l) \leq V_{ji'k}(l) < V_{ij'k}(l)$ (c.f. Figs. 3(d) and 7(c))

$$\begin{aligned} & m_{I_{ij}(l)}(\mathbf{x}, k) = \\ & \quad 1 - \max(0, \min(1, \gamma(W_{ijk}(l) - x_k))) \\ & \quad \text{for } (w_{ijk}(l) + W_{ijk}(l))/2 \leq x_k \leq W_{ijk}(l), \\ & \quad 1 - \max(0, \min(1, \gamma(x_k - w_{ijk}(l)))) \\ & \quad \text{for } w_{ijk}(l) \leq x_k \leq (w_{ijk}(l) + W_{ijk}(l))/2. \end{aligned} \tag{28}$$

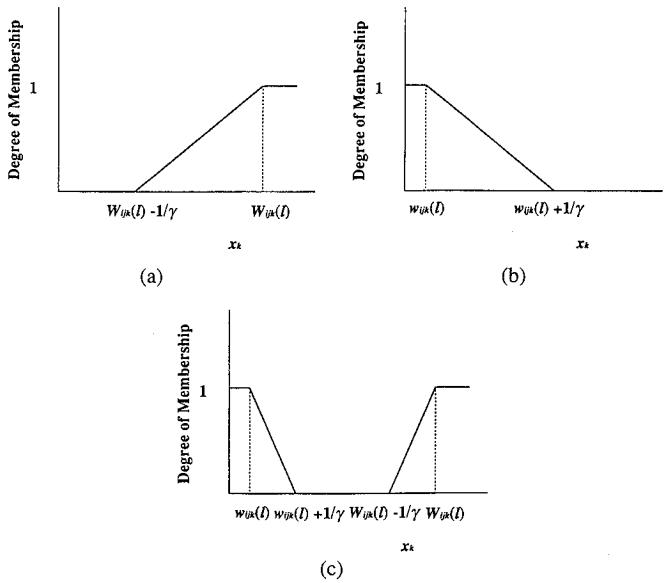

Fig. 7. One-dimensional membership function with respect to the inhibition hyperbox $I_{ij}(l)$.

To see that (24) gives the contour surface which is parallel to, and lies at an equal distance from the surface of $\overline{A_{ij'}(l) - I_{ij}(l)}$, assume $m_{I_{ij}(l)}(x, k)$ gives the maximum value among $m_{I_{ij}(l)}(\mathbf{x}, k'), k' = 1, \cdots, m$. Namely (c.f. Figs. 3(a), (b), and (d))

$$\begin{aligned} &|W_{ijk}(l) - x_k| = d && \text{for } v_{ji'k}(l) \le v_{ij'k}(l) \le V_{ji'k}(l) \\ &&& < V_{ij'k}(l), \\ &|x_k - w_{ijk}(l)| = d && \text{for } v_{ij'k}(l) < V_{ji'k}(l) leq V_{ij'k}(l) \\ &&& \le V_{ji'k}(l), \\ &|W_{ijk}(l) - x_k| = d && \text{for } v_{ij'k}(l) < v_{ji'k}(l) \le V_{ji'k}(l) \\ &&& < V_{ij'k}(l) \\ &\text{or } |x_k - w_{ijk}(l)| = d && < V_{ij'k}(l) \end{aligned} \tag{29}$$

where d is the maximum distance from the surface of $I_{ij}(l)$. Then the degree of membership $m_{I_{ij}(l)}(\mathbf{x})$ does not change even if $x_{k'}, k' = 1, \cdots, k-1, k+1, \cdots, m$, moves to the point until it satisfies any of (29) for k'. (If $v_{ji'k}(l) \le v_{ij'k}(l) \le V_{ij'k}(l) \le V_{ji'k}(l)$ holds, the change of $x_{k'}$ does not affect the degree of membership.) Thus the contour surface is parallel to, and lies at an equal distance from the surface of $\overline{A_{ij'}(l) - I_{ij}(l)}$.

The degree of membership for $\mathbf{x} \in H_{ij}(l)$ and $\mathbf{x} \notin I_{ij}(l)$ is obtained by calculating both $m_{A_{ij}(l)}(\mathbf{x})$ and $m_{I_{ij}(l)}(\mathbf{x})$, and taking the minimum, i.e., $\min(m_{A_{ij}(l)}(\mathbf{x}), m_{I_{ij}(l)}(\mathbf{x}))$. To show this, assume that for $\mathbf{x} \in H_{ij}(l)$ and $\mathbf{x} \notin I_{ij}(l)$, $m_{I_{ij}(l)}(\mathbf{x}) = m_{I_{ij}(l)}(\mathbf{x}, k) = m_{A_{ij}(l)}(\mathbf{x}) = m_{A_{ij}(l)}(\mathbf{x}, k')$ holds. Let k and k' be different. Then, if $x_{k'}$ is changed until it satisfies (29) for k', the value of $m_{A_{ij}(l)}(\mathbf{x})$ increases but that of $m_{I_{ij}(l)}(\mathbf{x})$ does not change. Thus, the resulting degree of membership does not change. This means that the contour surface is parallel to the surface of $I_{ij}(l)$ which is included in $\overline{A_{ij'}(l) - I_{ij}(l)}$. Also if x_k is changed so that the value d of (29) is decreased, $m_{I_{ij}(l)}(\mathbf{x})$ is increased but $m_{A_{ij}(l)}(\mathbf{x})$ is constant. Thus, the resulting membership function does not change. This means that $\mathbf{x}$ lies on one edge of the contour surface and it is parallel to the surface of $A_{ij}(l)$.

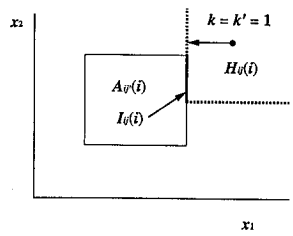

Fig. 8. Special case of the inhibition hyperbox.

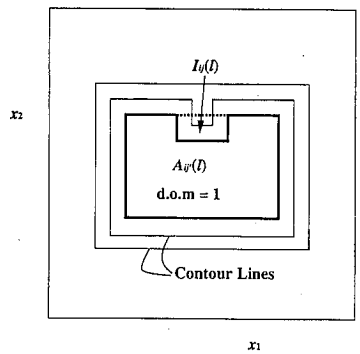

Fig. 9. The contour lines of membership function for the activation and inhibition hyperboxes (one dimension of the hyperbox is completely included).

The subscripts k and k' are the same when 1) $v_{ji'k}(l) \le v_{ij'k}(l) = V_{ji'k}(l) < V_{ij'k}(l)$ (cf. Fig. 3(a)), or 2) $v_{ij'k}(l) < v_{ji'k}(l) = V_{ij'k}(l) \le V_{ji'k}(l)$ (cf. Fig. 3(b)) holds. For 1) and 2) (see Fig. 8), the inhibition hyperbox does not work at all. This situation is avoided by expanding the inhibition hyperbox as discussed in Section II-D.

Thus $d_{r_{ij}(l)}(\mathbf{x})$ for (15) is given by

$$\begin{aligned} d_{r_{ij}(l)}(\mathbf{x}) = &\; m_{A_{ij}(l)}(\mathbf{x}) && \text{for } \mathbf{x} \notin H_{ij}(l) \\ &\; m_{I_{ij}(l)}(\mathbf{x}) && \text{for } \mathbf{x} \in I_{ij}(l) \\ &\; \min(m_{A_{ij}(l)}(\mathbf{x}), m_{I_{ij}(l)}(\mathbf{x})) && \text{for } \mathbf{x} \in H_{ij}(l) \\ &&& \text{and } \mathbf{x} \notin I_{ij}(l). \end{aligned} \tag{30}$$

Since $m_{A_{ij}(l)}(\mathbf{x}) = 1$ for $\mathbf{x} \in I_{ij}(l)$, (30) can be rewritten as follows

$$\begin{aligned} d_{r_{ij}(l)}(\mathbf{x}) = &\; m_{A_{ij}(l)}(\mathbf{x}) && \text{for } \mathbf{x} \notin H_{ij}(l) \\ &\; \min(m_{A_{ij}(l)}(\mathbf{x}), m_{I_{ij}(l)}(\mathbf{x})) && \text{for } \mathbf{x} \in H_{ij}(l). \end{aligned} \tag{31}$$

Fig. 9 shows an example of contour lines when $A_{ij'}(l)$ includes $A_{ji'}(l)$ in the x_1 axis (cf., Fig. 3(d)). As the distance from the surface of $\overline{A_{ij'}(l) - I_{ij}(l)}$ becomes large the effect of concave of the hyperbox is diminished.

Thus, the final degree of membership of $\mathbf{x}$ for a set of fuzzy rules $\{r_{ij}(l) \mid l = 1, \cdots\}$ denoted as $d_{r_{ij}}(\mathbf{x})$ is given by

$$d_{r_{ij}}(\mathbf{x}) = \max_{l=1,\cdots}(d_{r_{ij}(l)}(\mathbf{x})). \tag{32}$$

We take the maximum because the activation hyperbox $A_{ij}(l+1)$, if it exists, is included in the inhibition hyperbox $I_{ij}(l)$, and thus each fuzzy rule in $\{r_{ij}(l) \mid l = 1, \cdots\}$ is exclusive of one another. (Equation (32) corresponds to the "max" operator in the third layer of the architecture shown in Fig. 1.)

Now the degree of membership of $\mathbf{x}$ for class i denoted as $d_i(\mathbf{x})$ is given by

$$d_i(\mathbf{x}) = \min_{\substack{j \neq i, j=1,\cdots,n, \\ A_{ii}(1) \cap A_{jj}(1) \neq \emptyset}} (d_{r_{ij}}(\mathbf{x})). \tag{33}$$

When the activation hyperbox of class i overlaps with those of classes j and k, we resolve the conflict, independently, first between classes i and j, then between classes i and k. This process is reflected by taking the minimum in (33). For example, if $d_{r_{ij}}(\mathbf{x}) = 1$ and $d_{r_{ik}}(\mathbf{x}) = 0$, this means that $\mathbf{x}$ is in the region inhibited by the inhibition hyperbox between classes i and k and thus $\mathbf{x}$ should not be classified as class i. (Equation (33) corresponds to the "min" operator in the fourth layer of the architecture shown in Fig. 1.)

The input $\mathbf{x}$ is then classified as class i if $d_i(\mathbf{x})$ is the maximum among $d_j(\mathbf{x}), j = 1, \cdots, n$.

C. Class Boundaries

In this section, we explain the class boundaries of the fuzzy classifier using a two-class classification problem in which two input variables are considered.

First, consider the simple case where there is no overlap. Fig. 10 shows the classification regions, in which a given datum can be classified into a class, when the sensitivity parameter γ is large, in other words, the generalization region for each class is small. In this case, a region exists in which data cannot be classified because the degrees of membership for both classes are zero. By using a small sensitivity parameter γ, thus increasing the generalization region of each class, all the data in the input space can be classified as shown in Fig. 11. Since the contour lines are parallel to the surface of the activation hyperboxes, the class boundary becomes as shown in Fig. 11. If we make the sensitivity parameter γ small enough so that the degree of membership for any given point in the input space is greater than 0, in this case, the class boundary does not change even if we make the sensitivity parameter γ smaller. This means that as the sensitivity parameter γ is decreased from a large value to a small one the recognition rate of test data increases and reaches a plateau. If each input variable is normalized as [0, 1], it is sufficient to set the sensitivity parameter smaller than 1 to obtain the maximum recognition rate. (Suppose there is a fuzzy rule given by (14). Then the degree of membership $m_{A_{ij}(l)}(\mathbf{x}, k)$ is nonzero in $(v_{ijk}(l) - 1/\gamma, V_{ijk}(l) + 1/\gamma) \supset [0, 1]$.) Or if we want to know whether the input data are used for training or not, we may set the sensitivity parameter large. If a datum is not classified because the degree of membership is zero, we know that data which are near to this datum are not used for training.

Fig. 12 shows the class boundary when two classes overlap but no data are included in the inhibition hyperbox, including its surfaces, and the sensitivity parameter is small. If data exist on the surface of the inhibition hyperbox a problem arises as shown in Fig. 13. According to the definition of membership function, the degree of membership for one class is one on the surface of the inhibition hyperbox which is nearer to the activation hyperbox associated with that class. Thus the degrees of membership of class 2 datum which is on the surface of the inhibition hyperbox are one for both classes 1 and 2. And there is a region where a datum cannot be classified because the degree of membership is the same for two classes. This situation arises because on the boundary of inhibition hyperbox, data of two classes are allowed to exist. This can be avoided if we expand the inhibition hyperbox. In the next section we discuss how to avoid this situation and achieve a 100% recognition rate for the training data.

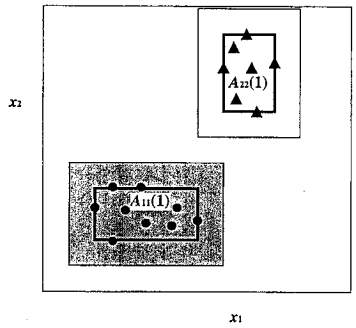

Fig. 10. Classification region when sensitivity parameter is large.

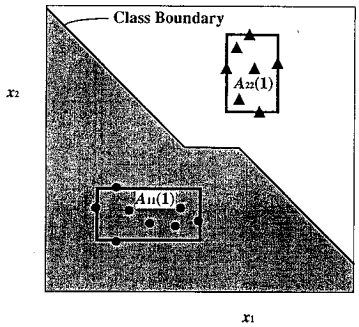

Fig. 11. Classification region when sensitivity parameter is small.

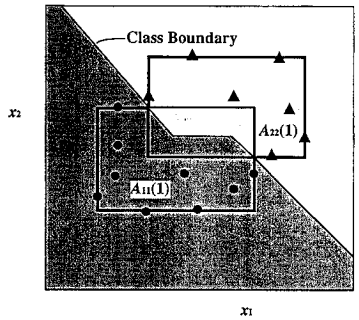

Fig. 12. Class boundary when two classes overlap (no data exist in the inhibition hyperbox).

D. Expansion of Inhibition Hyperboxes

To overcome the problem when data exist on the surface of the inhibition hyperbox, we expand the inhibition hyperbox $I_{ij}(l)$ associated with $A_{ij}(l)$ as shown in Fig. 14. We denote the resulting expanded inhibition hyperbox as $J_{ij}(l) = \{\mathbf{x} \mid u_{ijk}(l) \leq x_k < U_{ijk}(l), k = 1, \cdots, m\}$. The expanded inhibition hyperboxes for $A_{ij'}(l)$ and $A_{ji'}(l)$ are $J_{ij}(l)$ and $J_{ji}(l)$, respectively, which are different. The expanded inhibition hyperbox $J_{ij}(l)$ is defined as follows (cf. Fig. 3).

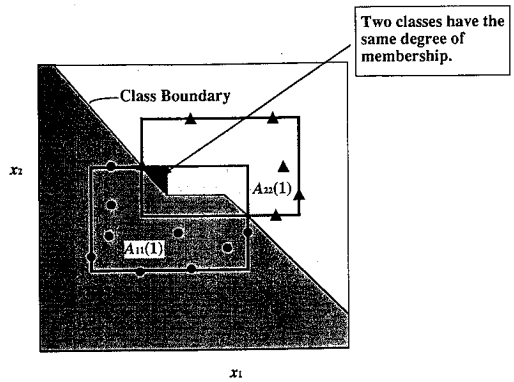

Fig. 13. Class boundary when two classes overlap (a datum exists on the boundary of the inhibition hyperbox).

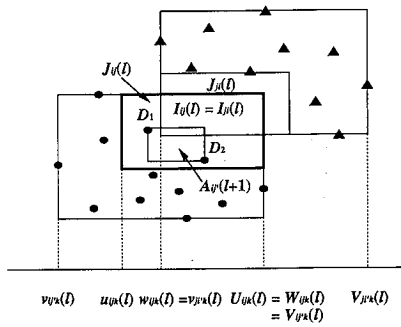

Fig. 14. Expansion of the inhibition hyperbox.

1) For $v_{ji'k}(l) \le v_{ij'k}(l) \le V_{ji'k}(l) < V_{ij'k}(l)$

$$u_{ijk}(l) = v_{i'jk}(l)$$
$$U_{ijk}(l) = V_{ji'k}(l) + \alpha(V_{ij'k}(l) - V_{ji'k}(l)) \quad (34)$$

where $\alpha(>0)$ is an expansion parameter.

2) For $v_{ij'k}(l) < v_{ji'k}(l) \le V_{ij'k}(l) \le V_{ji'k}(l)$

$$u_{ijk}(l) = v_{ji'k}(l) - \alpha(v_{ji'k}(l) - v_{ij'k}(l))$$
$$U_{ijk}(l) = V_{ij'k}(l). \quad (35)$$

3) For $v_{ji'k}(l) \le v_{ij'k}(l) \le V_{ij'k}(l) \le V_{ji'k}(l)$. In this case we do not expand the inhibition hyperbox for class i since we need not calculate the degree of membership for the x_k axis. Namely

$$u_{ijk}(l) = v_{ij'k}(l)$$
$$U_{ijk}(l) = V_{ij'k}(l). \quad (36)$$

4) For $v_{ij'k}(l) < v_{ji'k}(l) \le V_{ji'k}(l) < V_{ij'k}(l)$

$$u_{ijk}(l) = v_{ji'k}(l) - \alpha(v_{ji'k}(l) - v_{ij'k}(l))$$
$$U_{ijk}(l) = V_{ji'k}(l) + \alpha(V_{ij'k}(l) - V_{ji'k}(l)). \quad (37)$$

Now the rule shown in (15) is modified as

If $\mathbf{x}$ is $A_{ij'}(l)$ and $\mathbf{x}$ is not $J_{ij}(l)$ then x is class i (38)

where $j' = i$ for $l = 1$ and $j' = j$ for $l \ge 2$.

We define the activation hyperbox in the expanded inhibition hyperbox. Namely, if data for class i exist in $J_{ij}(l)$, we define the activation hyperbox $A_{ij'}(l+1)$. For example, in Fig. 14, an additional hyperbox $A_{ij'}(l+1)$ is defined for data D_1 and D_2.

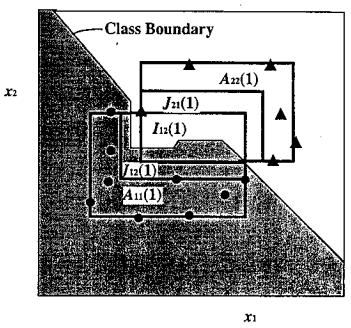

Fig. 15. Class boundary when two classes overlap (expansion of the inhibition hyperbox).

The inference method for the rules with expanded inhibition hyperboxes is the same as discussed in Section II-B.

Fig. 15 shows the class boundary of Fig. 13 when expanded inhibition hyperboxes are used. With this expansion, 100% recognition for the training data set is assumed.

E. Selection of Input Variables

As long as there are no identical data between different classes for a given set of training data, the fuzzy classifier can classify the training data 100%. But considering the generalization ability for a classifier, the smaller the number of fuzzy rules the better. Therefore, if the number of rules obtained for one class is larger than that for another, the former class is considered more difficult to classify than the latter. (The sensitivity parameter γ does not affect the number of rules generated but the expansion parameter α does. Since α is introduced only to avoid that any of the training data which exist on the surface of the inhibition hyperbox fail to be classified, we set α small enough to avoid creating additional rules.) Next, we discuss the method to select input variables using the number of fuzzy rules as a selection criterion.

Let M be the set of input variables, and $r_i(M)$ be the number of fuzzy rules obtained for class i with M and a given set of training data. We set the maximum number of rules r_s to obtain enough generalization ability. If for some i

$$r_i(M) \le r_s \quad (39)$$

does not hold, we conclude that the set M is insufficient to get enough generalization ability.

If (39) holds for all i, an attempt to eliminate redundant input variables can be performed as follows. Let M' be the set in which one input variable is deleted from M. Then acquire new fuzzy rules using the training data set with M'. If

$$r_i(M) = r_i(M') \quad (40)$$

holds for all i, delete the variable deleted from M' also from M. Repeat the above procedure until testing is finished for all the input variables initially included in the set M.

III. Performance Evaluation

A. Iris Data

We used the Fisher iris data [7], which are widely used for comparing various pattern classification methods, to compare the performance of the fuzzy classifier with that of the fuzzy

min-max neural networks [6]. The iris data included 150 data with four input features and three classes. Both training and test data sets were the same as those used in [6]. Namely, for each class, the first 25 data and the remaining 25 data were included in the training and test data, respectively. Two test data were misclassified by the fuzzy min-max neural network with 48 hyperboxes (fuzzy rules).

Table I shows the number of rules generated and number of misclassified test data for various α and $\gamma = 1$ using the method described in this paper. There were no misclassification for the training data and also for class 1 of the test data. As α increased, the number of misclassfied data decreased, and when α was equal to or greater than 0.9, the number of misclassfied test data was the same as that obtained by using the fuzzy min-max neural network. Although the number of rules increased as α increased, the number of rules generated for our fuzzy classifier was still less than that in the fuzzy min-max neural network. Since the number of rules for class 1 was always 1, that clearly indicated that no overlap existed between classes 1 and 2, and between classes 1 and 3.

For $\alpha = 0.001$, we applied the input selection method discussed in Section II-E. As seen in Table II, when we deleted input feature 1, the number of rules increased. Thus we kept input feature 1 in the input variable set. When we deleted input feature 2, the number of rules was the same as that when four input features were used. Therefore, we removed the input feature from the input variable set. When input features 2 and 3, or 2 and 4 were deleted, the number of rules increased. Thus, we could delete only input feature 2 without increasing the number of rules. Table II also shows the numbers of misclassified test data for different α values. The performance of the classifier using input features 1, 3, and 4 is comparable to that using all four input features.

TABLE I
PERFORMANCE FOR THE IRIS TEST DATA BY THE FUZZY CLASSIFIER ($\gamma = 1$)

α	No. of Rules	No. of Wrongs
0.001	5	6
0.1	7	5
0.2	7	5
0.3	9	5
0.4	9	4
0.5	9	4
0.6	11	4
0.7	11	3
0.8	13	3
0.9	17	2
0.99	17	2

TABLE II
SELECTION OF INPUT VARIABLES FOR THE IRIS DATA ($\gamma = 1$)

Input Deleted	No. of Wrongs			
	$\alpha = 0.001$	$\alpha = 0.3$	$\alpha = 0.6$	$\alpha = 0.99$
None	6 (5)	5	4	2
1	6 (6)	6	3	2
2	5 (5)	3	3	4
2, 3	8 (7)	8	8	9
2, 4	4 (8)	4	4	5

() indicates the number of rules generated for each case.

B. A License Plate Recognition System

A license plate recognition system [8], [9], which was originally developed using a decision tree algorithm, recognizes 10 numbers using 12 input features extracted from the images of running cars as taken by a TV camera. (The features used are the number of holes, the depth from the left–hand side, the curvature of some point, etc. [9]). The numbers were distorted and covered with dirt. In our study a total of 1630 data were divided into two combinations of training and test data: 1) 200 training data and 1430 test data, and 2) 810 training data and 820 test data. We compared the classification performance with that of a three-layered neural network with six hidden units. The number of hidden units was determined using the statistical method discussed in [10]. Since the performance of the network varies according to the initial values of the weights, we trained the network 100 times with initial values randomly assigned between -0.1 and 0.1, and calculated the average recognition rate.

Effect of Sensitivity Parameters: Using a 16 MIPS workstation, fuzzy rules based on 200 or 810 training data were generated in less than one second. For each, except class 7 for the 810 training data, one rule was extracted; for class 7 two rules were generated using the 810 training data, and the recognition rate for the same set of data was 100% irrespective of the sensitivity parameter γ and of the positive expansion parameter α. Then, we measured the recognition rates for different values of the sensitivity parameters using 1430 and 820 test data with $\alpha = 0.001$; the results are shown in Tables III and IV, respectively. As the sensitivity parameter becomes larger, the generalization region for each class becomes smaller. Thus, for $\gamma = 100$, the rules are considered to be crisp. Since the recognition rate was only 42.80% for the 1430 test data, the training data and test data were very different from each other; while for the 820 test data, this indicates that the training and test data were very similar. As the values of the sensitivity parameters decreased, the recognition rate improved and reached a plateau of 97.06% at $\gamma = 2$ for the 1430 test data and 99.63% at $\gamma = 8$ for the 820 test data. Therefore, for the study described in the next section, we set $\gamma = 2$ for both the 1430 and 820 test data. The recognition rate for the 1430 test data was constant even if we changed the expansion parameter α. Also the recognition rate of the 820 test data did not change much. This is because class regions did not overlap. So in the following we used $\alpha = 0.001$. When class regions overlapped because of the deletion of the input variables, we changed the expansion parameter α and checked the change of the recognition rate.

TABLE III
RECOGNITION RATE FOR THE 1430 TEST DATA BY THE FUZZY CLASSIFIER (200 TRAINING DATA USED, $\alpha = 0.001$)

γ	Recognition Rate in %
100	42.80
10	84.90
8	90.70
6	92.87
4	95.66
2	97.06
1	97.06

TABLE IV
RECOGNITION RATE FOR THE 820 TEST DATA BY THE FUZZY CLASSIFIER (810 TRAINING DATA USED, $\alpha = 0.001$)

γ	Recognition Rate in %
100	94.13
10	99.63
8	99.63
6	99.63
4	99.63
2	99.63
1	99.63

Training a six-hidden-unit neural network using the 200 training data took an average 11.7 seconds on a 31 MIPS mainframe computer. The average recognition rate for 1430 test data was 96.54% with maximum and minimum recognition rates of 98.25% and 95.17%, respectively. Thus, fuzzy rule acquisition described in this paper is more than 20 times faster than that of neural networks. Also, the recognition rate of the fuzzy rule inference proposed in this paper is higher than the average recognition rate by the neural network.

As for the 810 training data, training the six-hidden-unit neural network took an average 2.63 minutes on the 31 MIPS computer. The average recognition rate for the 820 test data was 99.41% with maximum and minimum recognition rates of 99.76% and 98.90%, respectively. Thus, fuzzy rule acquisition is more than 300 times faster than that of neural networks. Meanwhile, the recognition rate by the fuzzy classifier is better than the average recognition rate by the neural network.

Selection of Input Features: We selected input features according to the method discussed in Section II-E, in which the input features were deleted from the first to the twelfth features. Table V shows the number of rules generated using 200 training data when one or more than one input feature is deleted from a total of 12 input features. The recognition rate

TABLE V
SELECTION OF INPUT VARIABLES FOR THE 200 TRAINING DATA (DELETED FROM THE FIRST INPUT FEATURE, THE 1430 TEST DATA, $\alpha = 0.001$)

Input Deleted	Number of Rules	Recognition Rates in % Fuzzy Classifier Training	Fuzzy Classifier Test	Neural Network Training	Neural Network Test
None	1	100	97.06	100	96.54
1	2 for Classes 5 & 9	100	93.08	99.67	86.59
	1 for Others		(93.85)		
2	1	100	97.06	100	96.82
2,3	1	100	95.94	100	96.37
2,3,4	1	100	95.94	100	97.82
2,3,4,5	1	100	95.94	100	96.82
2,3,4,5,6	2 for Class 5	100	92.59	99.77	93.57
	1 for Others		(93.71)		
2,3,4,5,7	1	100	96.08	99.96	97.60

(): the maximum recognition rate by varying α
γ= 100 for the training data and γ= 2 for the test data

of the fuzzy classifier and the average recognition rate of the neural networks are also included in the table. As illustrated in Table V, when deleting the first feature, two rules were generated for classes 5 and 9. Therefore, the deleted feature was restored in the set of input features used in the fuzzy rules. Since the number of rules for each class by deleting the second feature was one, we deleted this feature. In doing so, five features were deleted and the recognition rate for the test data was slightly lower than that of using all 12 input features.

As for the training data the recognition rate of the fuzzy classifier was always 100%, while the average recognition rate of the neural network was less than 100% for three cases. As for the test data the average recognition rate of the neural network increased while the recognition rate of the fuzzy classifier decreased. Also the maximum difference of performance was 1.88% when only nine input features were used. This can be explained as follows. The performance degradation occurred when the number of rules generated for each class was one. Namely, there was no overlap between classes, or even if there was an overlap, no data existed in the inhibition hyperbox. In other words, the class boundaries of the fuzzy classifier were determined by the single activation hyperbox for each class. Thus the approximation of the separation boundaries became coarser than that of neural networks which were determined by the linear combinations of input features [11].

Since there is no reason to start deleting from the first input feature, we also deleted from the twelfth to first input features. Table VI shows the results. In this case only three input features were deleted but the recognition rate for the test data by the fuzzy classifier was better than that in Table V. Also the performance of the fuzzy classifier was comparable to that of the neural network.

Tables VII and VIII show the number of rules generated using 810 training data when one or more input features were deleted from the first input feature and from the twelfth input feature, respectively. In Table VII three features were deleted

TABLE VI
SELECTION OF INPUT VARIABLES FOR THE 200 TRAINING DATA (DELETED FROM THE TWELFTH INPUT FEATURE, THE 1430 TEST DATA, $\alpha = 0.001$)

Input Deleted	Number of Rules	Recognition Rates in %			
		Fuzzy Classifier		Neural Network	
		Training	Test	Training	Test
None	1	100	97.06	100	96.54
12	1	100	97.20	100	96.91
11,12	1	100	96.92	100	97.06
10,11,12	1	100	96.92	100	97.10

γ= 100 for the training data and γ= 2 for the test data

TABLE VII
SELECTION OF INPUT VARIABLES FOR THE 810 TRAINING DATA (DELETED FROM THE FIRST INPUT FEATURE, THE 820 TEST DATA, $\alpha = 0.001$)

Input Deleted	Number of Rules	Recognition Rates in %			
		Fuzzy Classifier		Neural Network	
		Training	Test	Training	Test
None	2 for Class 7 1 for Others	100	99.63	99.99	99.41
1	8 for Class 9 4 for Class 2 2 for Classes 5 & 7 1 for Others	99.51	98.66	97.54	96.60
2	2 for Classes 1,5,7 1 for others	100	99.51	99.98	99.26
3	2 for Class 7 1 for others	100	99.63	99.99	99.49
3,4	2 for Class 7 1 for others	100	99.63	99.99	99.49
3,4,5	2 for Class 7 1 for others	100	99.63	99.98	99.49

γ= 100 for the training data and γ= 2 for the test data

TABLE VIII
SELECTION OF INPUT VARIABLES FOR THE 810 TRAINING DATA (DELETED FROM THE TWELFTH INPUT FEATURE, THE 820 TEST DATA, $\alpha = 0.001$)

Input Deleted	Number of Rules	Recognition Rates in %			
		Fuzzy Classifier		Neural Network	
		Training	Test	Training	Test
None	2 for Class 7 1 for Others	100	99.63	99.41	99.41
12	2 for Class 7 1 for Others	100	99.51	99.98	99.28

γ= 100 for the training data and γ= 2 for the test data

while in Table VIII one feature was deleted. But for both cases the recognition rate of the fuzzy classifier was always better than the average recognition rate of the neural network for both the training and test data.

Training Time Comparison: We compared the training capabilities between our method and the neural network under severe convergence conditions. Namely, we set the input features so that the recognition rate for the training data by the fuzzy classifier was 100% but if any of the input features was deleted the recognition rate was no longer 100%.

In this case, we used the features from the ninth to twelfth as input features for the 200 training data. For classes 4, and 8, two rules were extracted; for class 9, three rules were extracted; for class 5, four rules were extracted; and one rule was extracted for each of the remaining classes. The recognition rate for the 200 training data was 100%. For the 1430 test data, the recognition rate was 72.10% for $\alpha = 0.001$. The maximum recognition rate of 75.52% was obtained for $\alpha = 0.6$. The training time was less than one second using the 16 MIPS computer. Using the neural network, it converged only four times out of 100 trials with the maximum number of epochs of 10000. The average training time was 4.49 minutes using a 31 MIPS computer. Thus the fuzzy rule extraction was at least 500 times faster than training the neural network (the average recognition rate for the 200 training data was 98.55%), and the average recognition rate for the 1430 test data was 74.50% (the maximum and minimum recognition rates were 77.76% and 71.33%, respectively).

For the 810 training data, we used the 1st and the 8th to twelfth features as input features. For classes 2, 4, 5 and 7, two rules were extracted; for class 10, three rules were extracted; and one rule was extracted for each of the remaining classes. The recognition rate for the 810 training data was 100%. For the 820 test data, the recognition rate was 98.05% for $\alpha = 0.001$. The maximum recognition rate of 98.17% was obtained for $\alpha = 0.2$. The training time was less than one second using the 16 MIPS computer. Using the neural network, the average recognition rate for the 810 training data was 99.88%. It converged 22 times out of 100 trials with the maximum number of epochs of 10000. The average training time was 13.90 minutes using the 31 MIPS computer. Thus, extracting fuzzy rules was at least 1500 times faster than training the neural network. The average recognition rate for the 820 test data was 97.78% (the maximum and minimum recognition rates were 98.66% and 96.83%, respectively).

IV. DISCUSSION

The difference between our fuzzy classification system and the fuzzy min-max neural networks discussed in [6] is that fuzzy rule extraction by our method is easier but the inference procedure is more complicated. In addition, our method tends to define larger hyperboxes or smaller number of fuzzy rules. When using the training data for testing, both methods give a recognition rate of 100% if there are no identical data presented in different classes. As for the generalization ability, we cannot say which is better. That means the performance may vary according to applications. Thus, two methods can be said to be alternative methods to realize a fuzzy classifier.

The advantages of our fuzzy classification system over neural networks are as follows:

1) The network structure is automatically determined through the acquisition of fuzzy rules according to the overlap between classes. Namely, the network is

two layers if there is no overlap between classes, three if each class overlaps with only one class at most, and four if one class overlaps with more than one class.

2) Knowledge acquisition or training is very fast. As long as there are no identical data presented in different classes, we can obtain a recognition rate of 100% for training data. Thus, retraining according to misclassification is not a problem.
3) Misclassification can be easily analyzed by fuzzy rules. Thus modification of rules is also possible.
4) Generalization ability can be directly controlled by modifying the sensitivity parameter γ. If a test datum is in the region where no activation hyperbox is in the neighborhood, the system can determine that the input is not classifiable.
5) Implementation is relatively easy since activation and inhibition hyperboxes can be determined recursively.

The disadvantages as compared to neural networks are as follows:

1) Generalization ability may be lower than that of neural networks when only one rule per class is extracted and when the characteristics of the training and test data are very different.
2) Since the overlapping is resolved by considering two classes each time, the network shown in Fig. 1, created by using our method may be larger then the conventional backpropagation networks for difficult classification problems.

There are two advantages of our fuzzy classification system over conventional fuzzy systems.

1) Fuzzy rules can be easily obtained from numerical data.
2) The classifier can be generated even for a large number of input variables.

V. Conclusion

In this paper, we discussed how to extract fuzzy rules directly from numerical data for pattern classification. Fuzzy rules with variable fuzzy regions were defined by activation hyperboxes which show the existence region of data for a class and inhibition hyperboxes which inhibit the existence of the data for that class. These rules were extracted from numerical data by recursively resolving overlaps between two classes. Then according to the number of rules extracted, we developed an approach to the selection of optimal input variables. The method described in this paper was compared with neural networks using the Fisher iris data and a license plate recognition system.

Acknowledgment

We are grateful to anonymous reviewers for their invaluable comments leading to improvements in the manuscript. Thanks also to P. Simpson of Orincon Corporation for providing the Fisher Iris data and H. Naya of Hitachi Research Laboratory for help in debugging the program.

References

[1] J. J. Buckley, Y. Hayashi, and E. Czogala, "On the equivalence of neural networks and fuzzy expert systems," in *Proc. IJCNN-92*, Baltimore, MD, vol. 2, 1992, pp. 691–695.
[2] D. E. Rumelhart *et al.*, "Learning internal representations by error propagation," *Parallel Distributed Processing: Explorations in the Microstructures of Cognition, Vol 1: Foundations.* Cambridge, MA: MIT Press, 1986, pp. 318–362.
[3] P. Simpson, *Artificial Neural Systems: Foundations, Paradigms, Applications and Implementation.* Elmsford, NY: Pergamon, 1990.
[4] C. T. Lin and C. S. G. Lee, "Neural network-based fuzzy logic control and decision system," *IEEE Trans. Comput.*, vol. 40, no. 12, pp. 1320–1336, 1991.
[5] L.-X. Wang and J. M. Mendel, "Generating fuzzy rules from numerical data with applications," Univ. of S. California, SIPI Report No. 169, 1991.
[6] P. K. Simpson, "Fuzzy min-max neural networks—Part 1: Classification," *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 776–786, Sept. 1992.
[7] R. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, Part II, vol. 7, pp. 179–188, 1936.
[8] M. Takatoo, M. Kanasaki, T. Mishima, T. Shibata, and H. Ota, "Gray scale image processing technology applied to vehicle license number recognition system," in *Proc. International Workshop on Industrial Applications of Machine Vision and Machine Intelligence*, Feb. 1987, pp. 76–79.
[9] H. Takenaga *et al.*, "Input layer optimization of neural networks by sensitivity analysis and its application to recognition of numerals," *T. IEE Japan* (in Japanese), vol. 111-D, no. 1, pp. 36–44, 1991. (Translated into English by Scripta Technica, Inc. *Elec. Eng. Japan*, vol. 111, no. 4, pp. 130–138, 1991).
[10] M. Kayama *et al.*, "Constructing optimal neural networks by linear regression analysis," in *Proc. Neuro-Nimes '90*, 1990, pp. 363–376.
[11] S. Abe, M. Kayama, and H. Takenaga, "How neural networks for pattern recognition can be synthesized," *J. Inform. Process.*, vol. 14, no. 3. pp. 344–350, 1991.

Shigeo Abe (M'79–SM'83) received the B.S. degree in electronics engineering, the M.S. degree in electrical engineering, and the Doctor of Engineering degree, all from Kyoto University, Kyoto, Japan, in 1970, 1972, and 1984, respectively.

Since 1972, he has been with Hitachi Research Laboratory, Hitachi, Ltd., and is now a Chief Researcher. From 1978 to 1979, he was a Visiting Research Associate at The University of Texas at Arlington, Arlington, TX.

His current research interests are in power system analysis, development of a vector processor, a Prolog processor, neural network theories, and fuzzy system models.

Dr. Abe was awarded an outstanding paper prize from the Institute of Electrical Engineers of Japan in 1984. He is a member of the International Neural Network Society, the Institute of Electrical Engineers of Japan, the Information Processing Society of Japan, the Institute of Electronics, Information and Communication Engineers of Japan, and the Society of Instrument and Control Engineers of Japan.

Ming-Shong Lan received the B.S in mechanical engineering from National Taiwan University (1976) and the Ph.D. degree in mechanical engineering from University of California at Berkeley (1983).

Dr. Lan is a Technical Staff Member at Rockwell International's Science Center where he has worked in the in-process monitoring and control of manufacturing processes, and applications of AI technology in design and manufacturing. His current research interests include neural networks, fuzzy logic, neurofuzzy systems, adaptive control, and intelligent control.