

Fuzzy Min-Max Neural Networks—Part 2: Clustering

Patrick K. Simpson, *Member, IEEE*

Abstract— In an earlier companion paper [56] a supervised learning neural network pattern classifier called the fuzzy min-max classification neural network was described. In this sequel, the unsupervised learning pattern clustering sibling called the fuzzy min-max clustering neural network is presented. Pattern clusters are implemented here as fuzzy sets using a membership function with a hyperbox core that is constructed from a min point and a max point. The min-max points are determined using the fuzzy min-max learning algorithm, an expansion-contraction process that refines the author's earlier Fuzzy Adaptive Resonance Theory neural network [50]. The fuzzy min-max clustering neural network stabilizes into pattern clusters in only a few passes through a data set; it can be reduced to hard cluster boundaries that are easily examined without sacrificing the fuzzy boundaries; it provides the ability to incorporate new data and add new clusters without retraining; and it inherently provides degree of membership information that is extremely useful in higher level decision making and information processing.

This paper will provide some background concerning the development of the fuzzy min-max clustering neural network and provide a comparison with similar work that has recently emerged. A brief description of fuzzy sets, pattern clustering, and their synergistic combination is presented. The fuzzy min-max clustering neural network will be explained in detail and examples of its clustering performance will be given. The paper will conclude with a description of problems that need to be addressed and a list of some potential applications.

I. INTRODUCTION

UNLIKE pattern classification, which provides class labels with pattern exemplars and seeks to find the decision boundary between classes that minimizes misclassification, pattern clustering has unlabeled pattern data and attempts to find natural groupings amongst the exemplars. The pattern clusters are formed according to some predefined metric or criterion [16, 19, 23]. As Bezdek [5] points out in his seminal book *Pattern Recognition With Fuzzy Objective Functions*, there are many possible clustering criteria that can be used, including distance, angle, curvature, symmetry, connectivity, and intensity.

The choice of the proper grouping metric is only one aspect of the clustering problem. The fundamental question of how many clusters also remains. To illustrate this point, Figure 1 shows a set of two-dimensional patterns (data points) that have no class labels. In this four small clusters as shown in Fig. 1(a)? Is it two large clusters like that shown in Fig. 1(b)? Or is there another clustering scheme that is more appropriate.

Manuscript received July 9, 1992; revised September 1, 1992.
The author is with the ORINCON Corporation, 9363 Towne Centre Drive, San Diego, CA 92121.
IEEE Log Number 9205334.

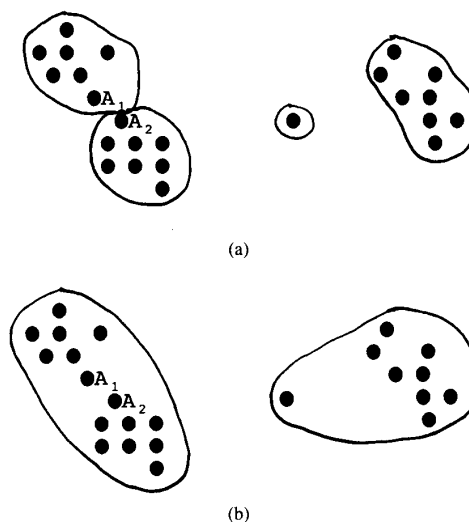


Fig. 1. An illustration of the different types of clusters that are possible. Panel (a) shows the data grouped into four clusters, and panel (b) shows the same data grouped into two clusters. Which one is correct depends a great deal on the data and the application.

The answer is not clear at all, rather, it is fuzzy. Continuing the example, assume the clusters found in Fig. 1(a) were formed by some clustering technique. There are some data points that completely belong to one cluster, but can also partially belong to another. Consider A_1 and A_2 : it seems reasonable that A_2 should have a high degree of membership in the cluster that A_1 is associated with, and vice versa. The fuzzy min-max clustering neural network addresses these concerns.

This paper describes the fuzzy min-max clustering neural network, a neural network that creates and refines pattern clusters in a fashion similar to adaptive resonance theory (ART) [9], [10] and the leader-cluster algorithm [19]. The fuzzy min-max clustering neural network makes the natural connection between clusters and fuzzy sets. This work is based upon an earlier neural network entitled fuzzy ART that was introduced by the author [50]. Since the introduction of fuzzy ART, another fuzzy ART neural network has been introduced into the literature [11], [12]. To eliminate any potential confusion and allow mutual recognition with the sibling classification network, the title fuzzy min-max clustering neural network has been selected.

The fuzzy min-max clustering neural network has several appealing attributes:

- 1) The fuzzy min-max clustering neural network does not

bound the number of clusters, rather the number grows to meet the demands of the problem.

- 2) The operations used by the fuzzy min-max clustering neural network are very simple, requiring only complement, compare, and accumulate operations which makes the implementation of this algorithm in fixed point hardware both feasible and efficient.
- 3) It is possible to reduce the fuzzy min-max clustering neural network to a set of hard cluster boundaries. This allows the verification and validation process to be performed quickly and easily.
- 4) There are only two parameters that need to be adjusted. One governs the maximum size of the hyperbox, and the other controls the fuzziness of the cluster membership.

The fuzzy min-max clustering neural network is constructed using hyperbox fuzzy sets. A hyperbox defines a region of the n -dimensional pattern space, and all patterns contained within the hyperbox have full cluster membership. A hyperbox is completely defined by its min point and its max point. The combination of the min-max points and the hyperbox membership function defines a fuzzy set (cluster). The resulting hyperbox fuzzy set fits naturally into a neural network framework (hence its name). Learning in the fuzzy min-max clustering neural network consists of creating and adjusting hyperboxes in pattern space as they are received. Once the fuzzy min-max clustering neural network is trained, it is operated by presenting a pattern and computing the membership value that pattern has in each of the currently existing fuzzy sets.

The remainder of this paper is organized as follows. Section II provides some background concerning the development of the fuzzy min-max clustering neural network and presents a brief overview of fuzzy sets, clustering techniques, and the motivation for their synergistic combination. Section III describes the fuzzy min-max clustering neural network in detail, including learning, recall, its neural network implementation, possible applications, and examples of its operation using different size hyperboxes. Section IV compares the fuzzy min-max clustering neural network with other fuzzy clustering neural networks, including the fuzzy c -means neural network, the Carpenter-Grossberg-Rosen fuzzy ART and the adaptive fuzzy leader cluster neural network. Section V presents some future issues that need to be addressed.

II. BACKGROUND

A. The Original Fuzzy ART

The development of the fuzzy min-max clustering neural network has been an evolutionary process that had as its goal the development of an unsupervised learning clustering neural network that is fast, efficient, reliable, and still capable of revealing the structure of the partitions being formed. These developments led to a neural network paradigm that combined fuzzy logic with the ART networks of Carpenter and Grossberg [9] into an analog pattern clustering system called fuzzy adaptive resonance theory (fuzzy ART) [50]. This original development had several flaws (none of which were crippling):

- 1) The relationship to fuzzy sets was not appropriately identified. In the original fuzzy ART description, the fuzzy portion of the neural network clusterer was attributed to the use of min and max operations for adjusting the min and max points and the use of data points that were rescaled to the range $[0, 1]$. In reality, the use of min and max operations in this context had nothing to do with fuzzy sets. Similarly, just because data points were constrained to the range $[0, 1]$ did not make the system fuzzy. There was, however, one important element of the original fuzzy ART that was properly described as fuzzy—the membership function.
- 2) The original fuzzy ART had a membership function that utilized the so-called fuzzy subthreshold measure. The fuzzy subthreshold measure is based upon an interpretation of fuzzy sets as points in the unit hypercube [38], and was not appropriate for this setting. This measure has been since replaced with a more appropriate membership function.
- 3) The hyperboxes could overlap, which allowed one data point to have full membership in more than one set. In retrospect, this significantly weakens the use of fuzzy sets. It is more plausible to utilize non-overlapping clusters which guarantees that a data point has the full membership in only one cluster unless it is a point along the boundary between two abutting hyperboxes.
- 4) The original fuzzy ART utilized a maximum (hyperbox) cluster size parameter. Although this was the only parameter in the entire clustering system, it is not reasonable to assume that one parameter is sufficient for the entire system (rather it would be a function of each cluster being formed). In reality, the clustering operation should be conducted in two phases. In the first phase, low-level clusters would be found and in the second phase they would be hierarchically grouped. This paper will not examine the hierarchical grouping, it will only focus on low-level clustering.
- 5) An extremely desirable attribute of any clustering system (neural, fuzzy, or otherwise) is to process new data immediately without having to retrain or refer to any of the previous training data. This attribute, referred to here as on-line learning, is present in the fuzzy min-max clustering neural network. Unfortunately, on-line learning typically results in a system that becomes order dependant during training, meaning if the same data points are processed in a different order during training, it is possible (and actually common) to have different pattern clusters emerge. The fuzzy min-max clustering neural network also shares this property. Utilizing the previously mentioned hierarchical grouping of low-level clusters could alleviate this problem. This is the subject of future research.

Despite these shortcomings, the original fuzzy ART demonstrated excellent performance on several data sets [50]. This paper describes improvements to the original fuzzy ART system that eliminate all but the last two of these flaws. These issues are by far the most difficult to tackle for any clustering technique and will remain the subject of future research.

During the period of time between the introduction of fuzzy ART and the writing of this paper, there have been several similar systems that have emerged. Carpenter, Grossberg, and Rosen have independently introduced a separate version of fuzzy ART [11], [12] that has many similar properties. Also a year after the introduction of the original fuzzy ART, Newton and Mitra [45] introduced the adaptive fuzzy leader cluster algorithm. As Lippmann [41], Moore [44], Burke [32], and Baruah and Holden [2] have pointed out, there are many similarities between adaptive resonance theory and classical leader cluster algorithm. In addition, there have been several other fuzzy neural network clustering techniques that have surfaced, including a neural network implementation of the fuzzy *c*-means algorithm [15] and a fuzzy Kohonen network [8]. In Section V the similarities, differences, strengths, and weaknesses of these algorithms relative to the fuzzy min-max clustering neural network will be discussed.

B. Fuzzy Sets Review

Fuzzy sets were introduced by Zadeh [62] as a means of representing and manipulating data that was not precise, but rather fuzzy. Zadeh's extension of set theory provided a mechanism for representing linguistic constructs such as "many," "few," "often," and "sometimes," and it provided some new tools that could be applied to pattern recognition and control by allowing the degree to which a pattern was present or a situation was occurring to be measured. In comparison, traditional set theory describes crisp events, events that either do, or do not, occur. There is no middle ground. Traditional sets use probability theory to explain whether an event will occur, measuring the chance with which a given event is expected to occur. In situations such as a flip of a coin or death, probability theory plays a role. These are situations that do not have much middle ground. In contrast, fuzzy theory measures the degree to which an event occurs. The degree to which a person is bald is very different than the probability that a person is bald. Probability theory states unequivocally that a person either is, or is not, bald. Fuzzy theory simply states that a person is somewhat bald, or a little bald, or quite bald, or sort of bald, and so on.

The following definition has been paraphrased from Zadeh's seminal paper [62] to illustrate the simplicity and the significance of fuzzy sets as well as to emphasize the relationship between pattern clusters and fuzzy sets that was made obvious in his definition. Zadeh's definition of a fuzzy set (with one minor editorial insertion and some emphasis provided in *italics*) is the following:

Let \mathcal{X} be a space of points (objects), with a generic element of \mathcal{X} denoted by x . [\mathcal{X} is often referred to as the universe of discourse.] A fuzzy set (*class*) A in \mathcal{X} is characterized by a membership (*characteristic*) function $m_A(x)$ which associates with each point in \mathcal{X} a real number in the interval $[0, 1]$, with the value of $m_A(x)$ representing the "grade of membership" of x in A . Thus, the nearer the value of $m_A(x)$ to unity, the higher the grade of membership of x in A .

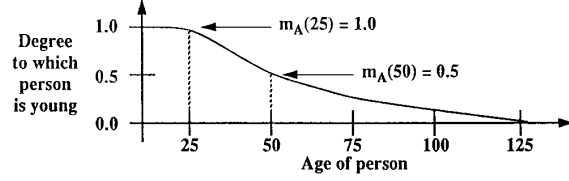


Fig. 2. This membership function describes the relationship between a person's age and the degree to which a person is considered to be young. This membership function determines that a 25 year old person belongs to A twice as much as a 50 year old person.

A more formal definition of fuzzy sets has been presented by many researchers and theoreticians. The formal definition offered below was abstracted from Kandel [34]. A fuzzy set A is a subset of the universe of discourse \mathcal{X} that admits partial membership. The fuzzy set A is defined as the ordered pair

$$A = \{x, m_A(x)\},$$

where $x \in \mathcal{X}$ and $0 \leq m_A(x) \leq 1$. The membership function $m_A(x)$ describes the degree to which the object x belongs to the set A , where $m_A(x) = 0$ represents no membership, and $m_A(x) = 1$ represents full membership.

As an example, let \mathcal{X} represent the ages of all people. The subset A of \mathcal{X} that represents those people who are young is a fuzzy set with the membership function shown in Fig. 2.

The operations on fuzzy sets are extensions of those used for traditional sets. Some of the common operations include *comparison*, *containment*, *intersection*, *union*, and *complement*. Assuming \mathcal{X} is the universe of discourse, $A \in \mathcal{X}$ and $B \in \mathcal{X}$, these operations are defined as follows.

Comparison: Is $A = B$?

$$A = B \text{ iff } m_A(x) = m_B(x) \quad \forall x \in \mathcal{X}$$

Containment: Is $A \subset B$?

$$A \subset B \text{ iff } m_A(x) < m_B(x) \quad \forall x \in \mathcal{X}$$

Union: The union of two fuzzy sets A and B , $A \vee B$, is found by combining the membership functions of A and B . Although there have been several different union operations defined (cf. [18] and [61]), the most common and by far the simplest union is defined as

$$m_{A \vee B}(x) = \max(m_A(x), m_B(x)) \quad \forall x \in \mathcal{X}.$$

Intersection: Like the union, the intersection of two fuzzy sets A and B , $A \wedge B$, is found by combining the membership functions of A and B and is defined as

$$m_{A \wedge B}(x) = \min(m_A(x), m_B(x)) \quad \forall x \in \mathcal{X}.$$

Complement: The complement of the fuzzy set A , \bar{A} , is defined as

$$m_{\bar{A}}(x) = 1 - m_A(x) \quad \forall x \in \mathcal{X}.$$

In addition to these operations, De Morgan's laws, the distributive laws, algebraic operations such as addition and multiplication, and the notion of convexity have fuzzy set equivalents [62].

C. Pattern Clustering Review

In many pattern recognition and decision making tasks, there is often little prior information available about the data that needs to be utilized. Pattern clustering uses the minimum amount of information to organize data into categories such that patterns within a cluster are more similar to each other than patterns belonging to other clusters. Given a data set $\mathcal{A} = (A_1, A_2, \dots, A_m)$, where each pattern A_k is an n -dimensional feature vector, the clustering partitions \mathcal{A} into c -many categories, where c is the number of clusters, a number that is either predetermined or determined on the fly. There are many different techniques that have been offered for solving this problem. In the following sections is a brief review of some traditional, fuzzy, and neural network clustering techniques.

1) *Traditional Clustering*: There are many clustering algorithms that have been developed to date, including ISODATA, FORGY, WISH, and CLUSTER [17], many of which are commercially sold. Jain [33] has reduced these clustering techniques to two popular methods:

- 1) *Hierarchical Clustering*: A hierarchical clustering technique imposes a hierarchical structure on the data which consists of a sequence of clusters. The result is a tree with the data patterns at the leaves and successively higher branches representing larger clusters.
- 2) *Partitional Clustering*: A partitional clustering technique organizes patterns into a small number of clusters by labeling each pattern in some way. Unlike hierarchical clustering, which offers several partitions of the data, partitional clustering finds a single cluster partition.

In addition to the two techniques cited above, there are also combinations of the two clustering approaches that are employed. There are many excellent books that describe classical approaches to pattern clustering, including Anderberg [1], Everitt [21], Hartigan [29], and Duda and Hart [19].

2) *Fuzzy Clustering*: Fuzzy sets brings a new dimension to traditional clustering systems by allowing a pattern to belong to multiple clusters to different degrees. Bezdek [6] has organized fuzzy clustering algorithms into five categories:

- 1) *Relation Criterion Functions*: Clustering driven by optimization of a criterion function which assesses partitions according to some global property of the grouped data. Ruspini [48] was the first to utilize this technique in the fuzzy community and he [49] and Bezdek [5, 6] have since considerably extended this pioneering work.
- 2) *Object Criterion Functions*: Clustering directly on the data set \mathcal{A} in the n -dimensional feature space according to some objective function is the most popular form of fuzzy pattern clustering. The fuzzy c -means and fuzzy ISODATA algorithms introduced by Dunn [20] and generalized by Bezdek [5], are the most popular technique for this class of fuzzy clustering algorithms.
- 3) *Convex Decompositions*: The decomposition of a fuzzy partition (a set of fuzzy clusters) into a combination of convex sets. The use of the convex decompositions may provide added insights into data structure that might otherwise be lost. Bezdek & Harris [4] describe three algorithms that can perform this decomposition.

4) *Numerical Transitive Closures*: The extraction of crisp equivalence relations from fuzzy transitive similarity relations. This technique is closely related to hierarchical methods based on graph-theoretic models.

5) *Generalized Nearest Neighbor Rules*: Although the nearest neighbor algorithm is used mostly for classification, there is a clustering version as well. This technique is primarily used once a data set has already been partitioned using another clustering algorithm such as fuzzy c -means.

3) *Neural Network Clustering*: Neural network clustering offers the ability to determine the size, shape, number, and placement of pattern clusters adaptively while intrinsically operating in parallel. In addition, the use of clustering to form sensory maps has strong biological support. Although there are a large number of neural networks available today (cf. [51]), there are only two primary neural clustering techniques currently in widespread use:

- 1) *Competitive Learning*: Similar to the c -means clustering algorithm, competitive learning finds the centroids of decisions regions in the n -dimensional pattern space. Although this form of neural network learning seems to have been introduced by Grossberg [25], [26] and von der Malsburg [42], it has been most successfully championed by Kohonen [37], who has extended the neural dynamics to include topographic constraints.
- 2) *Adaptive Resonance Theory*: Similar to the leader cluster algorithm, adaptive resonance theory nondestructively creates pattern "codes" (clusters). The concept of adaptive resonance was introduced by Grossberg [27] and was first cast into a neural network formalism by Carpenter and Grossberg [9]. There have been numerous extensions and refinements since [13].

D. Fuzzy Neural Networks

Recently there has been a great amount of interest in the synergistic combination of neural networks and fuzzy systems [7]. Some of this work was done over a decade ago [39], [57], and some of this work has appeared recently [31], [35], [43], [59]. Many of these efforts have focused upon methods for implementing fuzzy rules in a neural network framework and techniques for parallelizing the successful fuzzy control system applications. The work presented here illustrates very clearly that fuzzy sets and neural networks can be effectively merged by utilizing neural network processing elements as fuzzy sets.

The synergism of neural networks and fuzzy sets seems natural. Neural networks, such as the competitive learning and adaptive resonance networks, compare an input pattern with a set of stored exemplars, codes, or reference vectors. The closer the input matches the data stored in the network, the higher the output values of the corresponding processing element. Assume subsets of the processing elements in a neural network represent separate categories (or clusters), such as the output processing elements of an adaptive resonance network. The closer an input pattern A_h fits within a category, the higher the corresponding value of that processing element. By viewing each category as a fuzzy set and identifying the processing

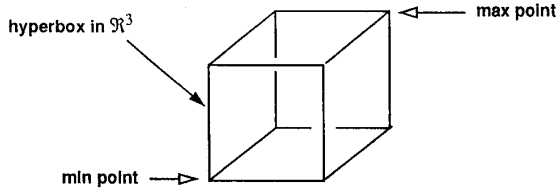


Fig. 3. The min-max hyperbox $B_j = \{V_j, W_j\}$ in \mathbb{R}^3 is shown. The min and max points are all that are required to completely define the hyperbox. A membership function is associated with the hyperbox; it determines the degree to which any point $A_h \in \mathbb{R}^3$ is contained within the box. A cluster is defined as one hyperbox fuzzy set.

element transfer function with the membership function, a direct relationship between fuzzy sets and pattern clustering is realized.

Data sets tend to have inherent ambiguities (especially real-world data), and there never seem to be enough data to describe completely all the possible patterns that will be seen in a real-world applications. By employing fuzzy sets as pattern clusters, it is possible to describe the degree to which a pattern belongs to one cluster or another. This relationship was realized very early in the development of fuzzy sets. In Zadeh's seminal paper [62] he describes a technical memorandum that preceded his paper that applied fuzzy sets to pattern recognition [3].

III. FUZZY MIN-MAX CLUSTERING NEURAL NETWORK

In the following sections the fuzzy min-max clustering neural network is described. The fuzzy set hyperbox membership function is presented first. The fuzzy min-max learning algorithm for clustering is then explained, and a simple example is provided. Next, the neural network implementation of the fuzzy min-max clustering neural network is described, and potential applications are offered. Finally, examples of fuzzy min-max clustering using various hyperbox sizes are offered.

Fuzzy Hyperbox Membership Function

An illustration of the min and max points in a three-dimensional hyperbox is shown in Fig. 3. Although it is possible to use hyperboxes that have an arbitrary range of values in any dimension, this paper will use values that range from 0 to 1 along each dimension. Hence, the pattern space will be the n -dimensional unit cube I^n . The membership function for each hyperbox fuzzy set must describe the degree to which a pattern fits within the hyperbox. In addition, it is typical to have fuzzy membership values range between 0 and 1. Note that the range of membership values has no direct relationship to the range of pattern values along each dimension. These ranges of values were selected because they made the computations simpler. Also note that rescaling any n -dimensional input pattern from \mathbb{R}^n to I^n destroys a minimal amount of absolute information when the absolute range of each dimension is known prior to rescaling, and it preserves all of the relative information between the various dimensions.

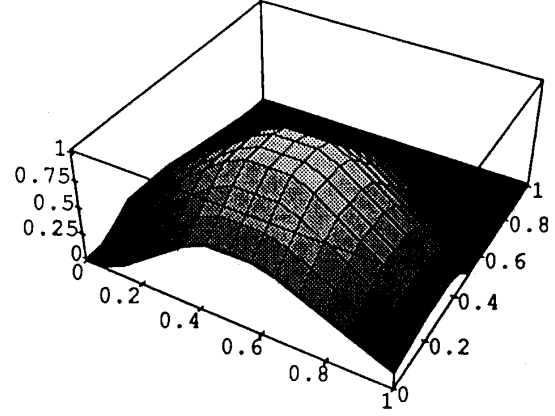


Fig. 4. A two-dimensional fuzzy set is illustrated using the membership function described by eq. (1) with the min point at (0.4, 0.2) and the max point at (0.6, 0.4). The sensitivity parameter γ is set at 4 to produce a moderately quick decrease from full membership to no membership.

Let the j th hyperbox fuzzy set, B_j , be defined by the ordered set

$$B_j = \{A_h, V_j, W_j, b_j(A_h, V_j, W_j)\}$$

for all $h = 1, 2, \dots, m$, where $A_h = (a_{h1}, a_{h2}, \dots, a_{hn}) \in I^n$ is the h th pattern in the data set, $V_j = (v_{j1}, v_{j2}, \dots, v_{jn})$ is the min point for the j th hyperbox, $W_j = (w_{j1}, w_{j2}, \dots, w_{jn})$ is max point for the j th hyperbox, and the membership function for the j th hyperbox is $0 \leq b_j(A_h, V_j, W_j) \leq 1$. The membership function measures the degree to which the h th input pattern A_h falls within the hyperbox formed by the min point V_j and the max point W_j . On a dimension-by-dimension basis, this can be considered a measurement of how far each component is greater (less) than the max (min) point value along each dimension that falls outside the min-max bounds of the hyperbox. As A_h approaches the hyperbox, $b_j(A_h, V_j, W_j)$ should approach 1. When the point is contained within the hyperbox, $b_j(A_h, V_j, W_j) = 1$. The function that meets all these criteria is the complement of the average amount of max point violations and the average amount of min point violations. The resulting membership function (shown in Fig. 4 for two-dimensions) is defined as

$$b_j(A_h, V_j, W_j) = \frac{1}{n} \sum_{i=1}^n [1 - f(a_{hi} - w_{ji}, \gamma) - f(v_{ji} - a_{hi}, \gamma)], \quad (1)$$

where $f()$ is the two-parameter ramp threshold function

$$f(x, \gamma) = \begin{cases} 1 & \text{if } x\gamma > 1 \\ x\gamma & \text{if } 0 \leq x\gamma \leq 1 \\ 0 & \text{if } x\gamma < 0 \end{cases}$$

The parameter γ is the sensitivity parameter that regulates how fast the membership values decrease when an input pattern is separated from the hyperbox core. When γ is large, the fuzzy set becomes more crisp, and when γ is small the fuzzy set becomes less crisp. By making γ large, it is possible to define crisp cluster boundaries as a special case of fuzzy boundaries. The learning algorithm described in the

next section does not allow overlapping hyperboxes. This does not mean that the fuzzy sets do not overlap, only that those portions of the fuzzy set representing full membership are nonoverlapping. Cluster boundaries are defined as those points where the membership values between fuzzy sets are equal.

IV. FUZZY MIN-MAX CLUSTERING ALGORITHM

Learning in the fuzzy min-max clustering neural network is an expansion/contraction process. Assume the training set is defined as $\mathcal{A} \in \{A_h | h = 1, 2, \dots, m\}$, where $A_h = (a_{h1}, a_{h2}, \dots, a_{hn}) \in I^n$ is the h th pattern. The learning process begins by selecting an ordered pair from \mathcal{A} and finding the closest hyperbox to that pattern that can expand (if necessary) to include the pattern. If a hyperbox cannot be found that meets the expansion criteria, a new hyperbox is formed and added to the system. This growth process allows existing clusters to be refined over time, and it allows new clusters to be added without retraining.

One of the residuals of hyperbox expansion is overlapping hyperboxes (note: *hyperbox overlap* is referring to the overlap of those portions of the fuzzy set that admit full membership—the crisp cluster boundaries, not partial membership). Hyperbox overlap causes ambiguity. It is reasonable to assume that a pattern can have the same partial membership in more than one cluster. It is not reasonable to assume that a pattern can completely belong to more than one cluster. A much more desirable situation would be to ensure that there are nonoverlapping crisp clusters. The fuzzy min-max clustering neural network utilizes a contraction process to eliminate any hyperbox overlap.

A. Fuzzy Min-Max Learning

The fuzzy min-max classification learning algorithm is a four-step process:

- 1) *Initialization*: Initialize all the min-max points prior to any learning.
- 2) *Expansion*: Identify the hyperbox closest to the input pattern that can be expanded and expand it. If an expandable hyperbox cannot be found, add a new hyperbox.
- 3) *Overlap Test*: Determine whether the recent expansion caused any overlap between hyperboxes.
- 4) *Contraction*: If the expansion test identified any overlapping hyperboxes, contract the hyperboxes to eliminate the overlap. The contraction process is performed immediately after an overlap is found between two hyperboxes to eliminate any potentially wasted or erroneous contractions that might occur if all of the overlaps were detected first and the overlaps were then eliminated.

Each of these steps is described in greater detail in the following subsections.

1) *Hyperbox Initialization*: There are two sets of clusters used in the fuzzy min-max clustering algorithm, the *committed set* \mathcal{C} and the *uncommitted set* \mathcal{U} . The committed set of clusters are those clusters that have had their min-max points adjusted, and the uncommitted set of clusters are those clusters that are waiting to be committed. Before any learning has occurred the

system has an empty set of committed clusters and an arbitrary number of clusters in the uncommitted set. Prior to presenting the first pattern, the hyperboxes held in the uncommitted set have their min-max points initialized. The min points are initially

$$V_j = \underline{1} \quad (2)$$

and the max points are initially

$$W_j = \underline{0} \quad (3)$$

for all $B_j \in \mathcal{U}$, where $\underline{1}$ is the n -dimensional vector of all ones and $\underline{0}$ is the n -dimensional vector of all zeroes.

The purpose of the initialization process is to ensure that the first pattern committed to a hyperbox results in a single point that is identical to the input pattern. As an example, assume that the j th hyperbox B_j had just been withdrawn from \mathcal{U} and will be adjusted for the first time using the pattern A_h . Then, following the learning process, the min and max points would be

$$V_j = W_j = A_h. \quad (4)$$

By performing this initialization procedure, the expansion portion of the learning algorithm performs this operation automatically.

2) *Hyperbox Expansion*: Given a pattern $A_h \in \mathcal{A}$, find the hyperbox $B_j \in \mathcal{C}$ that provides the highest degree of membership and allows expansion (if needed). The degree of membership $b_j(A_h, V_j, W_j)$ is measured using (1). The maximum size of a hyperbox is bounded by $0 \leq \theta \leq 1$, a user-defined value. For the hyperbox B_j to expand to include A_h , the following constraint must be met:

$$\sum_{i=1}^n (\max(w_{ji}, a_{hi}) - \min(v_{ji}, a_{hi})) \leq \theta. \quad (5)$$

If (5) is satisfied, the min point of the hyperbox is adjusted using the equation

$$v_{ji}^{new} = \min(v_{ji}^{old}, a_{hi}) \quad \forall i = 1, 2, \dots, n, \quad (6)$$

and the max point is adjusted using the equation

$$w_{ji}^{new} = \max(w_{ji}^{old}, a_{hi}) \quad \forall i = 1, 2, \dots, n. \quad (7)$$

If all $B_j \in \mathcal{C}$ are exhausted without any possible expansions, then select a hyperbox $B_j \in \mathcal{U}$ and apply (6) and (7).

It is important to note that the use of the max and min operations does resemble the fuzzy set union and intersection, but that interpretation does not apply here. For these operations to be correctly interpreted as fuzzy operations, they would have to be applied to the fuzzy set membership values $b_j(A_h, V_j, W_j)$, not the parameters of the fuzzy set membership function V_j and W_j .

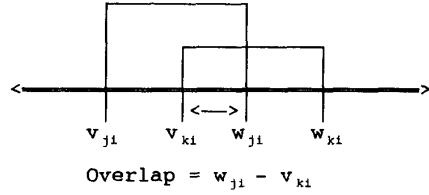
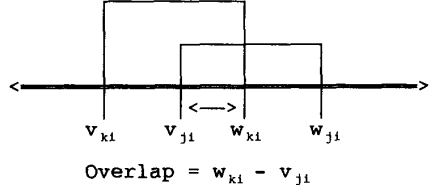
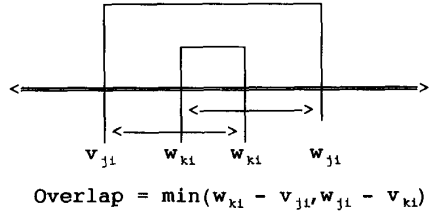
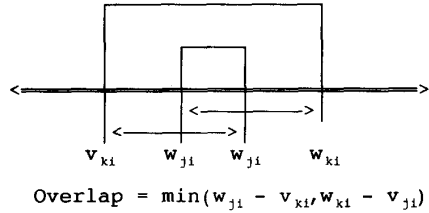
(a) CASE 1: Max of B_j overlaps Min of B_k (b) CASE 2: Min of B_j overlaps Max of B_k (c) CASE 3: B_k contained within B_j (d) CASE 4: B_j contained within B_k 

Fig. 5. The four cases that occur during overlap testing. Each graph shows the relative values of two hyperboxes along one dimension.

3) *Hyperbox Overlap Test*: Assume that the hyperbox $B_j \in \mathcal{C}$ is expanded in the previous step. To determine overlap, a dimension-by-dimension comparison is conducted between B_j and the remaining $B_k \in \mathcal{C}$. The expansion creates an overlap between B_j and B_k if one of the four cases is satisfied for each of the n dimensions:

Case 1: $v_{ji} < v_{ki} < w_{ji} < w_{ki}$

Case 2: $v_{ki} < v_{ji} < w_{ki} < w_{ji}$

Case 3: $v_{ji} < v_{ki} \leq w_{ki} < w_{ji}$

Case 4: $v_{ki} < v_{ji} \leq w_{ji} < w_{ki}$

Each of these cases is illustrated in Fig. 5 with the corresponding overlap measurement shown. These overlaps are eliminated using hyperbox contraction as described in the next subsection.

4) *Hyperbox Contraction*: If the hyperboxes B_j and B_k are overlapping, the overlap is eliminated on a dimension-by-dimension basis. Using the four cases previously described, the overlapping hyperboxes are contracted as follows:

Case 1: If $v_{ji} < v_{ki} < w_{ji} < w_{ki}$,

$$v_{ki}^{new} = w_{ji}^{new} = \frac{v_{ki}^{old} + w_{ji}^{old}}{2}. \quad (8)$$

Case 2: If $v_{ki} < v_{ji} < w_{ki} < w_{ji}$,

$$v_{ji}^{new} = w_{ki}^{new} = \frac{v_{ji}^{old} + w_{ki}^{old}}{2}. \quad (9)$$

Case 3: If $v_{ji} < v_{ki} \leq w_{ki} < w_{ji}$ then the contraction is performed on the smaller of the two overlaps. If $w_{ki} - v_{ji} < w_{ji} - v_{ki}$ then contract using the assignment

$$v_{ji}^{new} = v_{ki}^{old}; \quad (10)$$

otherwise use the assignment

$$w_{ji}^{new} = v_{ki}^{old}. \quad (11)$$

Case 4: If $v_{ki} < v_{ji} \leq w_{ji} < w_{ki}$, then, by symmetry, the same assignments under the same conditions in case 3 are applied here.

The overlap test and contraction procedures are different from those used in the fuzzy min-max classification neural network [56] in one important respect. In classification, the emphasis is placed on accurately placing decision boundaries between classes. To accomplish this, the classification overlap test and the corresponding contraction procedure are designed to make the minimal change possible to eliminate class overlap. To do this, the one dimension with the smallest possible overlap is eliminated. Here, the clusters are adjusted to eliminate overlap in every dimension so that the resulting clusters are more compact.

Steps 1–4 are repeated for each pattern in the data set until cluster stability is achieved. Cluster stability is defined as all hyperbox min and max points not changing during successive presentations of the data set in the same order.

B. Illustration of the Expansion–Contraction Process

Figure 6 illustrates how four two-dimensional patterns are clustered into two hyperboxes with a maximum hyperbox size of $\theta = 0.4$. The step-by-step example illustrates how hyperboxes are formed and how overlapping hyperboxes are contracted.

C. Implementing the Fuzzy Min-Max Clustering Neural Network

Implementing the fuzzy min-max clusterer as a neural network makes it possible to immediately exploit the parallel nature of the algorithm. Although the learning portion of the algorithm is not necessarily neural (there is no biological basis for the use of the expansion–contraction process), the recall operation fits immediately into a neural network framework. Please note that in the following description the label for the j th F_B PE, b_j , is also used to represent the value produced by the j th F_B PE.

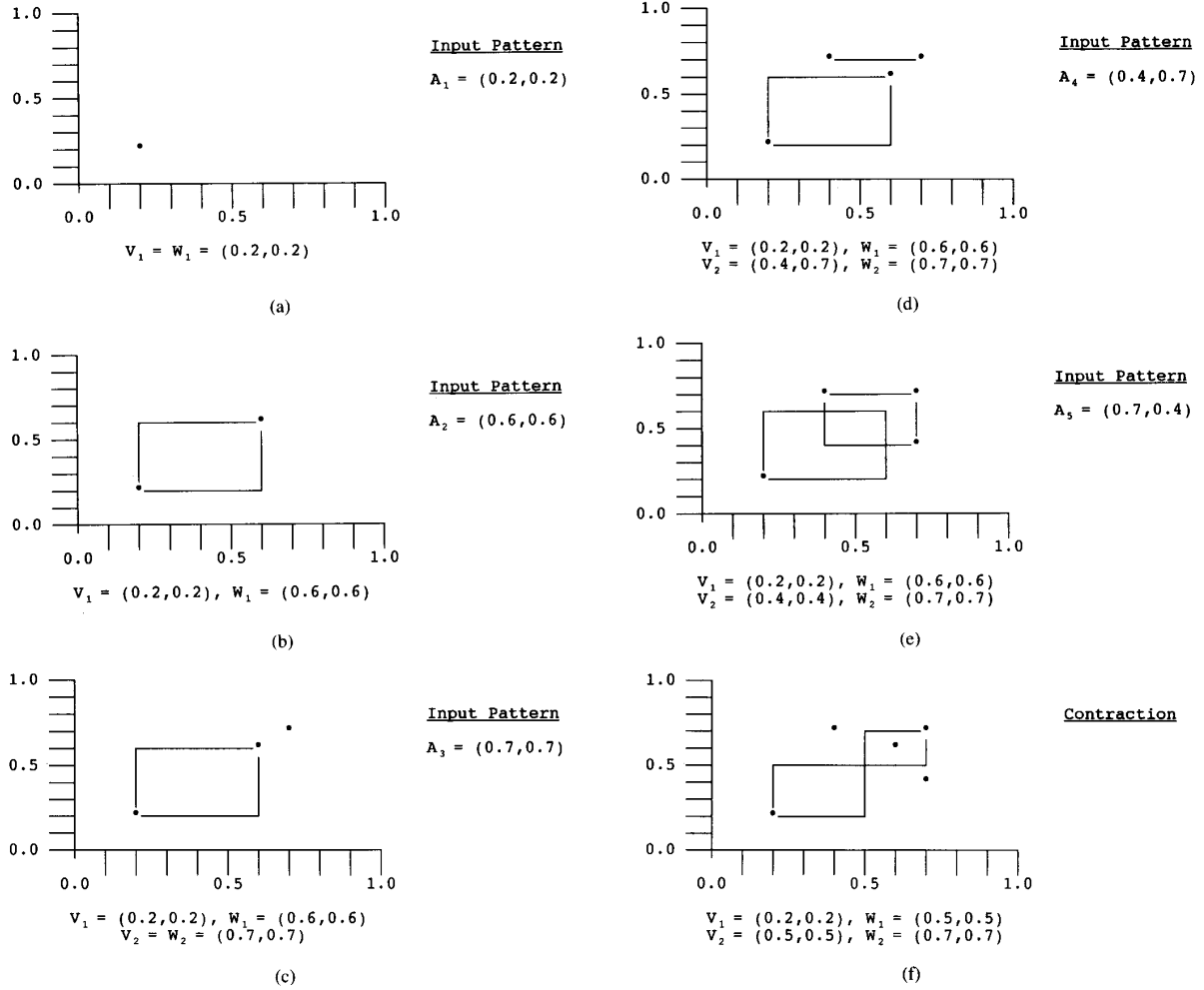


Fig. 6. An example illustrating the fuzzy min-max clustering learning algorithm for a set of two-dimensional data is shown. The maximum hyperbox size is $\theta = 0.4$. The storage sequence is as follows. (a) Initially, there are no hyperboxes, so the first input pattern, $A_1 = (0.2, 0.2)$, creates a set of min and max connections $V_1 = W_1 = (0.2, 0.2)$ for the hyperbox B_1 . (b) The second input pattern, $A_2 = (0.6, 0.6)$, is able to fit in an expanded B_1 hyperbox so B_1 is expanded to create the new hyperbox with the min-max points $V_1 = (0.2, 0.2)$ and $W_1 = (0.6, 0.6)$. (c) The third input pattern, $A_3 = (0.7, 0.7)$, will not fit into an expanded B_1 hyperbox, so it is used to create a new hyperbox B_2 with the min-max points $V_2 = W_2 = (0.7, 0.7)$. (d) The fourth input pattern, $A_4 = (0.4, 0.7)$, is closest to B_1 , but it cannot be expanded to include this new point. The next closest hyperbox B_2 can expand to include A_4 , resulting in the min-max points $V_2 = (0.4, 0.4)$ and $W_2 = (0.7, 0.7)$. (e) The input pattern A_5 is found to be closest to B_1 , which cannot expand, and is then found to be the next closest to B_2 , which causes B_2 to expand, resulting in the min-max points $V_2 = (0.4, 0.4)$ and $W_2 = (0.7, 0.7)$. Up to this point, there has not been any hyperbox overlap, but now there is. Using the equations described in the text, the hyperboxes are contracted, resulting in the min-max points $V_1 = (0.2, 0.2)$, $W_1 = (0.5, 0.5)$ and $V_2 = (0.5, 0.5)$, $W_2 = (0.7, 0.7)$.

The fuzzy min-max clusterer can be implemented as a two-layer neural network where the input layer, F_A , that consists of n processing elements (PE's) (one PE for each dimension of the patterns being clustered) and the output layer, F_B , consists of m PE's (one for each cluster). There are two connections from each F_A PE to each F_B PE (Fig. 7). This topology is best seen as two vectors abutting each F_B PE b_j —a min vector V_j and a max vector W_j . This is not the first neural network to employ a dual-connection strategy. Other neural networks that have used dual connections include excitatory-inhibitory cooperative-competitive networks and mean-variance multilayer perceptrons (cf. [54]). Each F_B

PE utilizes the membership function described by (1). This topology directly implements each hyperbox fuzzy set as a F_B PE, meaning that the fuzzy set hyperbox B_j is exactly the same as the F_B PE b_j .

The output from the F_B PE's represents the degree to which the input pattern A_h belongs to each of the p clusters. The F_B PE with the largest value represents the cluster with the greatest degree of membership. Unless an input pattern falls directly on a point that separates two hyperboxes, the full degree of membership ($b_j = 1$) can only occur for one hyperbox fuzzy set. There are three possible ways that the F_B cluster membership information can be applied:

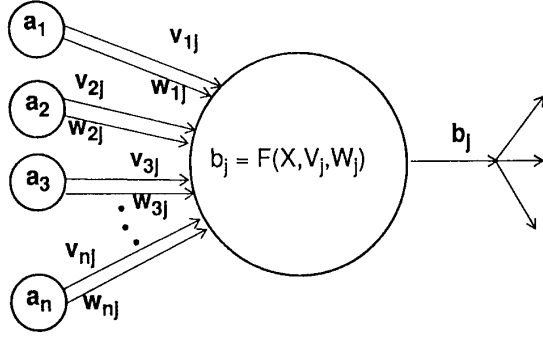
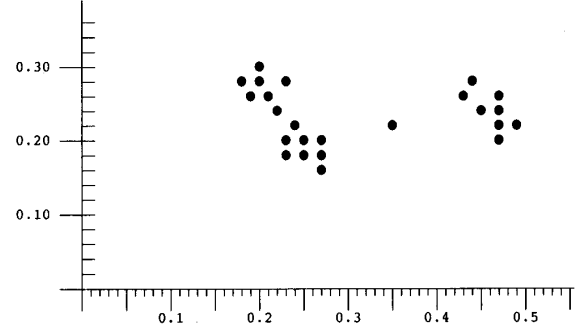


Fig. 7. The implementation of a hyperbox and its associated membership function as a neural network assembly is shown for the j th F_B node b_j . The input nodes accept each dimension of the k th input A_k . There are two connections from each input node to the output node. One connection represents the min value for that dimension, and the other connection represents the max value for that dimension. The connections between i th input node and the j th output node are v_{ji} and w_{ji} . The min point for the j th F_B is the vector $V_j = (v_{j1}, v_{j2}, \dots, v_{jn})$ and the max point is $W_j = (w_{j1}, w_{j2}, \dots, w_{jn})$. Assuming the input pattern A_h , b_j 's output value $b_j(A_h, V_j, W_j)$ is computed using eq. (1). This entire neural assembly represents a hyperbox fuzzy set.



$A_1 = (0.18, 0.28)$	$A_{10} = (0.23, 0.20)$	$A_{17} = (0.43, 0.26)$
$A_2 = (0.19, 0.26)$	$A_{11} = (0.25, 0.20)$	$A_{18} = (0.44, 0.28)$
$A_3 = (0.20, 0.28)$	$A_{12} = (0.25, 0.18)$	$A_{19} = (0.45, 0.24)$
$A_4 = (0.18, 0.30)$	$A_{13} = (0.27, 0.20)$	$A_{20} = (0.47, 0.26)$
$A_5 = (0.22, 0.26)$	$A_{14} = (0.27, 0.18)$	$A_{21} = (0.47, 0.22)$
$A_6 = (0.22, 0.24)$	$A_{15} = (0.27, 0.16)$	$A_{22} = (0.47, 0.20)$
$A_7 = (0.23, 0.28)$	$A_{16} = (0.34, 0.22)$	$A_{23} = (0.49, 0.22)$
$A_8 = (0.24, 0.22)$		$A_{24} = (0.49, 0.22)$

Fig. 8. The data used for the two-dimensional clustering example are shown. There are 24 data points that have values that range from 0.18 to 0.49 along one dimension and from 0.16 to 0.30 along the other dimension. Note that this data set does not clearly reveal where the clusters should be.

- 1) **Classification:** Each cluster can be considered a separate pattern class. Although the validity of the clusters is in question, this can be done. In this situation, the largest activation value would be found, and the corresponding cluster would be declared the class for the input pattern. This use of the fuzzy min-max neural network resembles the popular competitive learning algorithms of Kohonen and Grossberg (cf. [51]).
- 2) **Feature Extraction:** The F_B PE activation values can be treated as features in a preprocessing stage of a hierarchical neural network. As an example, Rajapakse *et al.* [46] and Li and Wee [40] have used ART networks as feature extraction networks for hierarchical data fusion (cf. [55]). In this setting, the low level feature detectors would be fuzzy min-max clustering neural networks with small hyperboxes. Intermediate layers of the hierarchical system would cluster the membership values produced from the lower levels and the output network could consist of a fuzzy min-max classification neural network [56]. By regulating the γ value in the transfer functions, it is possible to contrast enhance the response of each layer, which adds a powerful tuning mechanism to the information processing scheme.
- 3) **Pattern Matching:** The F_B PE's can be treated as hidden units in a three-layer pattern matching neural network in a manner similar to the counterpropagation network [30]. In this setting, a third layer of F_C PE's would be added with a set of connections from the F_B PE's to the F_C PE's, where each F_C PE corresponds to a dimension of the output pattern. The connections between the F_B and F_C PE's could be adjusted in many different ways, but the use of the LMS algorithm [60] is the most likely candidate.

D. Examples of Operation

Example 1—Two-Dimensional Clustering: A two-dimensional data set consisting of 24 points was constructed to show how the fuzzy min-max clustering neural network performs with various hyperbox sizes. Fig. 8 shows a scatter plot of the data used for this example as well as the values of the data points. The order of the data presentation was exactly as that shown. This data set was constructed to illustrate two common clustering dilemmas: 1) there is a group of data points that could be considered either one cluster or two, and 2) there is an outlier in the middle of the two primary groups of data.

The data were clustered six times with six different hyperbox sizes ranging from 0.25 to 0.065 as shown in Fig. 9. The number of clusters created ranges from 1 (when the hyperbox size is the largest) to 5 (when the hyperbox size is the smallest). The cluster arrangement that seems most appropriate was with a hyperbox size of either 0.075 or 0.065 (panels (e) and (f)). As this example shows, it is important to have a feel for the appropriate hyperbox size and be attentive to the order of pattern presentation.

Example 2—Iris Data Clustering: The second data set used was the Fisher Iris data [22]. This data set was selected because its familiarity to the pattern recognition community allows a measure of relative performance. The Iris data consist of 150 four-dimensional feature vectors (patterns) in three separate classes, 50 for each class. Each datum (pattern dimension) has been rescaled to lie between 0 and 1 by dividing each component of each pattern by 8. The clustering performance of the fuzzy min-max clustering neural network was determined by first clustering the data and then identifying each cluster with a class to determine how well the fuzzy min-max clustering neural network was able to find the underlying structure of the data using four different hyperbox sizes. The results of this experiment are shown in the confusion matrices found in Fig. 10 for hyperboxes that range in size from 0.25 to 0.10. The

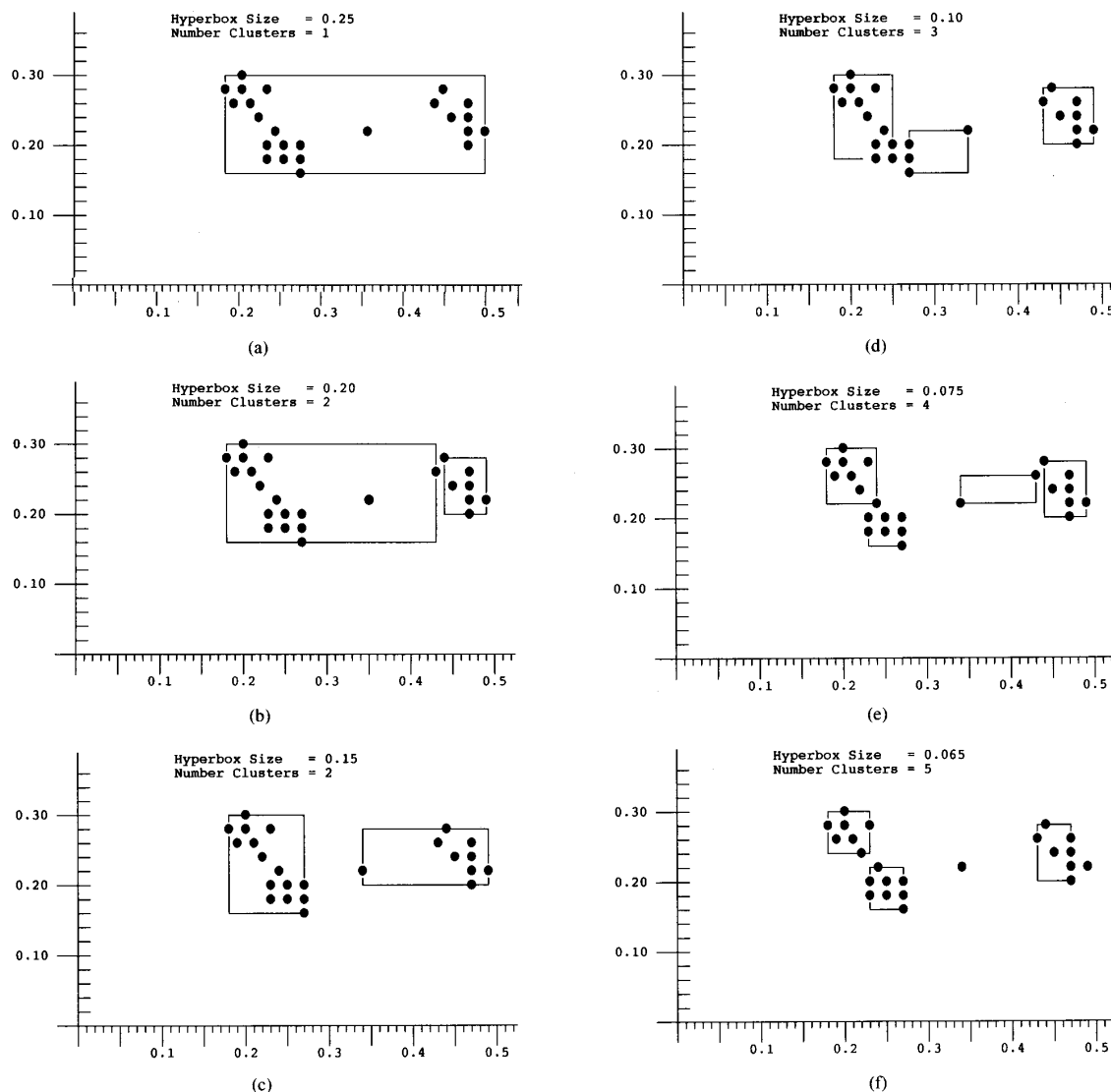


Fig. 9. A demonstration of the clustering performance of the fuzzy min-max clustering neural network with six different hyperbox sizes. The hyperboxes that were formed are shown with the exception of panel (f), where two points were never expanded.

best performance was found with the hyperbox size of 0.10, which produced 14 clusters. The number of passes through the data to achieve cluster stability was less than 10 in each instance.

V. COMPARING THE FUZZY MIN-MAX CLUSTERING NEURAL NET WITH OTHER CLUSTERING TECHNIQUES

There have been many clustering algorithms presented in the pattern recognition, fuzzy, and neural network literature. Examples of similar techniques include the Kohonen self-organizing feature map [37], D-ART [36], (which also uses min-max points for clusters), and fuzzy-shells [14]. In the following three subsections is a comparison of three fuzzy neural network clustering techniques: (1) the fuzzy *c*-means

clustering neural network, (2) the Carpenter–Grossberg–Rosen fuzzy ART, and (3) adaptive fuzzy leader clustering.

A. Fuzzy *c*-Means Clustering Neural Network

The fuzzy *c*-means clustering algorithm is arguably the best known and possibly the best performing of all the fuzzy clustering algorithms. The *c*-means clustering algorithm assumes there are *c* clusters in the data set and then minimizes an objective function that results in the mean for each of the *c* clusters. There are two primary differences between the fuzzy *c*-means algorithm and the fuzzy min-max clustering neural network:

- 1) The fuzzy *c*-means algorithm assumes the number of clusters is known in advance and the fuzzy min-max

(a) Hyperbox size = 0.25 Number clusters = 3				
	1	2	3	
1	100%			
2		100%		
3		40%	60%	

(b) Hyperbox size = 0.20 Number clusters = 3				
	1	2	3	
1	100%			
2		94%	6%	
3		30%	70%	

(c) Hyperbox size = 0.15 Number clusters = 7				
	1	2	3	
1	98%	2%		
2		96%	4%	
3		22%	78%	

(d) Hyperbox size = 0.10 Number clusters = 14				
	1	2	3	
1	100%			
2		88%	12%	
3		10%	90%	

Fig. 10. Four confusion matrices using four different hyperbox sizes are shown for clustering the four-dimensional Iris data. Classes were determined by identifying which clusters were associated with each class. Class information was not provided for this data until after the clusters were formed.

clustering neural network determines the number of clusters dynamically. To determine the proper number of clusters, both algorithms need to cluster with different parameter sets (e.g., number of means or size of hyperbox) and use some additional cluster validity criteria to determine which set of parameters yields the best results.

- 2) The fuzzy c -means minimizes an objective function to find the best set of clusters. As such, fuzzy c -means is not order dependent and, assuming the number of clusters is correctly selected, it can find "better" clusters. The fuzzy min-max clustering neural network is order dependent and does not minimize any known objective function.

For more information on the fuzzy c -means algorithm, refer to Bezdek [4]–[6], who has provided the greatest body of results, extensions, and analysis of the fuzzy c -means algorithm.

Recently a neural network implementation of the fuzzy c -means algorithm has been introduced [15]. This neural network uses the fuzzy c -means objective function in the multilayer perceptron framework and shows that the cluster centers can be found using the generalized delta rule.

B. Carpenter–Grossberg–Rosen Fuzzy ART

The Carpenter–Grossberg–Rosen (CGR) fuzzy ART [11], [12] mostly closely resembles the fuzzy min-max clustering neural network. As was mentioned earlier, this should not be surprising since they both grew out of the fuzzification of the ART-1 neural network. Despite the similar origin, there are several differences between these two fuzzy neural network clustering techniques:

- 1) From a fundamental perspective, the CGR fuzzy ART carries the same flaw that the original fuzzy ART had concerning the relationship between fuzzy sets and neural network. The CGR fuzzy ART explains the relation-

ship between fuzzy sets and ART-1 as follows ([12], p. 762):

Figure 3 [in [12]] summarizes how the ART1 operations of category choice, matching, search, and learning translate into Fuzzy ART operations by replacing the set theory intersection operator (\cap) of ART 1 by the fuzzy set theory conjunction operator, or MIN operator (\wedge).

As was pointed out earlier, this analogy is only appropriate when the values of the patterns are fuzzy set membership values. No such relationship is either claimed or evident for the CGR Fuzzy ART.

- 2) The fast learning operation used in the CGR fuzzy ART is

$$U_j^{new} = U_j^{old} \wedge X_h, \quad (12)$$

where $X_h = (a_{h1}, a_{h2}, \dots, a_{hn}, 1 - a_{h1}, 1 - a_{h2}, \dots, 1 - a_{hn})$ is the $2n$ -dimensional input pattern, A_h is the equivalent of a fuzzy min-max clustering neural network input pattern, the \wedge operator is the vector form of the pairwise min operation, and U_j is a $2n$ -dimensional weight vector. The behavior of this learning rule, although described in the same terms as min-max point hyperbox terminology, is very different from the fuzzy min-max cluster learning equations. To illustrate this point, Fig. 11 shows how the fast learning CGR fuzzy ART clusters the first two patterns found in Fig. 6. By expressing the CGR fuzzy ART weight vector as the concatenation of the fuzzy min-max clustering neural network's min and max points, $U_j = (V_j, W_j)$, it is possible to describe the difference mathematically as

$$W_j^{new} = W_j^{old} \vee A_h \neq W_j^{old} \wedge A_h^c,$$

where \vee is the vector form of the pairwise max operation.

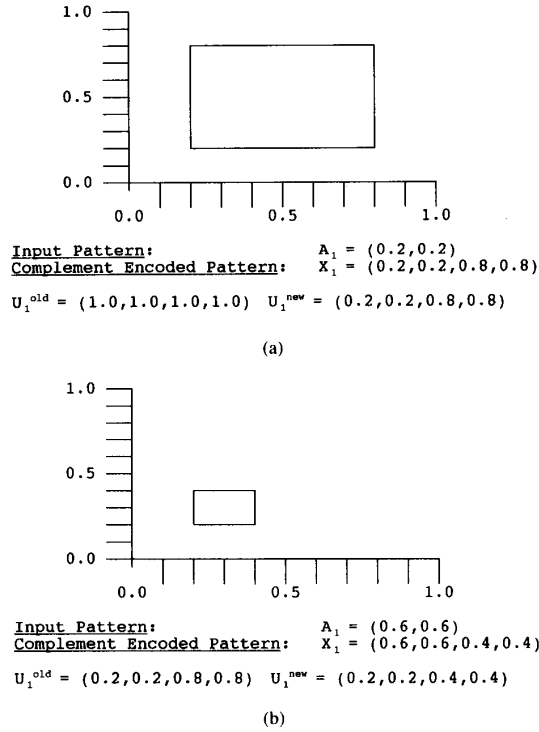


Fig. 11. Using the first two data points from the earlier example of fuzzy min-max clustering (Fig. 6), the behavior of the CGR fuzzy ART learning rule is shown. As this example shows, the behavior of the two learning rules is significantly different. Note that the CGR fuzzy ART concatenates the min and max points into a single weight vector. This figure has separated the four-dimensional weight vector into its two two-dimensional vectors for plotting purposes so that the differences between the two learning rules could be visually compared.

- 3) The CGR fuzzy ART allows hyperbox clusters to overlap, which results in pattern cluster ambiguity—a pattern can have full membership in more than one cluster. The fuzzy min-max clustering neural network does not share this problem.
- 4) The CGR fuzzy ART bounds the size of hyperboxes through the matching function ([11, pp. 765–766]). The fuzzy min-max clustering neural network uses an explicit calculation of the size of the hyperbox to bound the size (eq. (5)). In effect, these are two separate approaches toward achieving the same result.

C. Adaptive Fuzzy Leader Clustering

The relationship between the leader cluster algorithm and adaptive resonance theory was first pointed out by Lippmann [41]. Several people have since analyzed this relationship [2], [32], [44]. The leader cluster algorithm works as follows:

- 1) The leader cluster system begins with a cluster centered at the location of the first input pattern, A_1 . This cluster is called the leader.
- 2) For each of the remaining patterns, $A_h, h = 1, 2, \dots, m$, the following is done.
 - a. Find the cluster closest to A_h (usually using Euclidean distance).

- b. If the ratio of the distance between the cluster center and A_h is less than a prespecified tolerance value, the cluster center is updated to include A_h in its cluster by averaging the location of A_h with the cluster center.
- c. If the closest cluster ratio is too large, then this cluster is eliminated from the set of candidate clusters and control returns to step 2a).
- d. If the clusters are exhausted, add a new cluster with A_h as the cluster center.

Recently, Newton and Mitra [45] have developed a fuzzy adaptive leader cluster algorithm that utilizes a fuzzy membership function to determine the degree to which the input pattern belongs to each cluster. In this sense, the fuzzy min-max clustering neural network and adaptive fuzzy leader clustering are similar (although the membership functions are very different). In addition, the adaptive fuzzy leader cluster algorithm has a neural network implementation that leverages the same relationship between neural networks and fuzzy sets that is described in this paper—a fuzzy set can be implemented as a single neuron and its associated weights for fast parallel execution.

There are three primary differences between the adaptive fuzzy leader cluster neural network and the fuzzy min-max clustering neural network:

- 1) The key difference between these two clustering techniques is the method of representing clusters. The adaptive fuzzy leader cluster algorithm utilizes a single point to represent the center of each cluster and the fuzzy min-max clustering neural network utilizes a set of min-max points to represent the hard (crisp) boundaries of the cluster.
- 2) The membership functions between these two algorithms are different. The adaptive fuzzy leader cluster neural network uses Euclidean distance from a cluster prototype and the fuzzy min-max clustering neural network uses a fuzzy Hamming distance from the edges of an n -dimensional hyperbox.
- 3) The method of adapting the clusters is also different. The adaptive fuzzy leader cluster neural network tries to optimally position a cluster prototype and the fuzzy min-max clustering neural network adjusts the boundaries.

VI. SUMMARY AND FURTHER WORK

Several extensions and refinements of the author's earlier introduction of the fuzzy ART neural network have been presented. Earlier in the paper there were several flaws with the original fuzzy ART that have been corrected here. Two key problems still remain: (1) adaptively sizing the hyperboxes, and (2) hierarchically clustering the hyperbox fuzzy sets to capture more of the structure of the data. In addition, there are several analytical issues that need to be addressed, including (1) proving that the expansion contraction process eliminates hyperbox overlap, (2) proving that the hyperbox fuzzy sets reduce to crisp hyperbox sets, (3) and proving that the clusters stabilize for a fixed but arbitrarily ordered data set. These and other issues are the subject of future work.

ACKNOWLEDGMENT

The extensions of the original Fuzzy ART that have been presented in this paper were performed on the author's own time without any funding. The views expressed in this paper are those of the author and do not reflect the views or opinions of ORINCON Corporation.

I would like to thank Dr. L. Firestone, Dr. J. Bezdek, Dr. R. C. Eberhart, and J. H. McBeth for many insightful suggestions that helped me to properly represent this work. The author also wishes to recognize Steven L. Biafore, Stephen A. Luse, and Russell O. Deich for many helpful comments. Most of all, I would like to thank my wife, Christalyn, for her unending support and love.

REFERENCES

- [1] M. Anderberg, *Cluster Analysis for Applications*. New York: Academic Press, 1973.
- [2] A. Baruah and A. Holden, "Adaptive resonance theory and the classical leader algorithm: Similarities and additions," in *Proc. Artificial Neural Networks in Engineering—ANNIE 91*, 1991.
- [3] R. Bellman, R. Kalaba, and L. Zadeh, "Abstraction and pattern classification," RAND Memorandum, RM-4307-PR, Oct. 1964.
- [4] J. Bezdek and J. Harris, "Convex decompositions of fuzzy partitions," *J. Mathematical Analysis and Applications*, vol. 67, pp. 490–512, 1979.
- [5] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press, 1981.
- [6] J. Bezdek, "Some non-standard clustering algorithms," NATO ASI Series, vol. G14, *Developments in Numerical Ecology*, P. Legendre and L. Legendre, Eds. Berlin: Springer-Verlag, 1987.
- [7] J. Bezdek, "Fuzzy models and computational neural networks," Tutorial at *IEEE Conf. Neural Networks for Ocean Engineering* (Washington, DC), Aug. 15, 1991.
- [8] J. Bezdek, E. Tsao, and N. Pal, "Fuzzy Kohonen clustering network," *IEEE Trans. Neural Networks*, to be published.
- [9] G. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Computer Vision, Graphics, and Image Understanding*, vol. 37, pp. 54–115, 1987.
- [10] G. Carpenter and S. Grossberg, "ART 2: Stable self-organization of pattern recognition codes for analog pattern recognition," *Appl. Opt.*, vol. 26, pp. 4919–4930, 1987.
- [11] G. Carpenter, S. Grossberg, and D. Rosen, "Fuzzy ART: An adaptive resonance algorithm for rapid, stable classification of analog patterns," in *Proc. 1991 Int. Joint Conf. Neural Networks* (Seattle), vol. II, 1991, pp. 411–416.
- [12] G. Carpenter, S. Grossberg, and D. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Networks*, vol. 4, pp. 759–771, 1991.
- [13] G. Carpenter and S. Grossberg, Eds., *Pattern Recognition By Self-Organizing Neural Networks*. Cambridge, MA: MIT Press, 1991.
- [14] N. Dave, "Fuzzy-shell clustering and applications to circle detection in digital images," *Int. J. General Systems*, vol. 16, pp. 343–355, 1990.
- [15] J. Davis, T. Warms, and W. Winters, "A neural network implementation of the fuzzy *c*-means clustering algorithm," in *Proc. Int. Joint Conf. Neural Networks* (Seattle), vol. II, 1991, p. A-953.
- [16] P. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
- [17] R. Dubes and A. Jain, "Clustering techniques: The user's dilemma," *Pattern Recognition*, vol. 8, pp. 247–260, 1976.
- [18] D. Dubois and H. Prade, "A review of fuzzy sets aggregation connectives," *Inform. Sci.*, vol. 36, pp. 85–121, 1985.
- [19] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: John Wiley, 1973.
- [20] J. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybern.*, vol. 3, pp. 32–57, 1974.
- [21] B. Everitt, *Cluster Analysis*. New York: John Wiley, 1974.
- [22] R. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, part II, pp. 179–188, 1936.
- [23] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York: Academic Press, 1972.
- [24] I. Gath and A. Geva, "Unsupervised optimal fuzzy clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, pp. 773–781, 1989.
- [25] S. Grossberg, "Neural expectation: Cerebellar and retinal analogues of cells fired by unlearnable and learnable pattern classes," *Kybernetik*, vol. 10, pp. 49–57, 1972.
- [26] S. Grossberg, "Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural detectors," *Biol. Cybern.*, vol. 23, pp. 121–134, 1976.
- [27] S. Grossberg, "Adaptive pattern classification and universal recoding: II. Feedback, oscillation, olfaction, and illusions," *Biol. Cybern.*, vol. 23, pp. 187–207, 1976.
- [28] S. Grossberg, "How does a brain build a cognitive code?" *Psychol. Rev.*, vol. 87, pp. 1–51, 1980.
- [29] J. Hartigan, *Clustering Algorithms*. New York: John Wiley, 1975.
- [30] R. Hecht-Nielsen, "Counterpropagation networks," *Appl. Opt.*, vol. 26, pp. 4979–4985, 1987.
- [31] T. Huntsberger and P. Ajjimarangese, "Parallel self-organizing feature maps for unsupervised pattern recognition," *Int. J. Intelligent Systems*, vol. 16, pp. 357–372, 1990.
- [32] L. Burke, "Clustering characterization of adaptive resonance," *Neural Networks*, vol. 4, pp. 485–492, 1991.
- [33] A. Jain, "Cluster analysis," in *Handbook of Pattern Recognition and Image Processing*, T. Young and K.-S. Fu, Eds., 1986, pp. 33–57.
- [34] A. Kandel, *Fuzzy Mathematical Techniques with Applications*. Reading, MA: Addison-Wesley, 1986.
- [35] J. Keller and D. Hunt, "Incorporating fuzzy membership functions into the perceptron algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 7, pp. 693–699, 1985.
- [36] K. Kizan and T. Caudell, "D-ART: A pattern recognition system based on adaptive resonance and algebraic metric space theories," *Adaptive Neural Systems: The 1990 IR&D Technical Report*, Boeing Computer Services, July 1991.
- [37] T. Kohonen, *Self-Organization and Associative Memory*. Berlin: Springer-Verlag, 1984.
- [38] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Englewood Cliffs, NJ: Prentice-Hall, 1992.
- [39] S. Lee and E. Lee, "Fuzzy neural networks," *Math. Biosci.*, vol. 23, pp. 151–177, 1975.
- [40] D. Li and W. Wee, "A new neocognitron structure modified by ART and backpropagation," in *Proc. 1990 Int. Joint Conf. Neural Networks* (Washington, DC), vol. 1, pp. 420–423, 1990.
- [41] R. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Magazine*, vol. 61, pp. 4–22, 1987.
- [42] C. v. d. Malsburg, "Self-organization of orientation sensitive cells of the striate cortex," *Kybernetik*, vol. 14, pp. 85–100, 1973.
- [43] S. Mitra and S. Pal, "Self-organizing neural network as a fuzzy classifier," *IEEE Trans. Syst., Man, Cybern.*, to be published.
- [44] B. Moore, "ART1 and pattern clustering," in *Proc. 1988 Connectionist Summer School*, pp. 174–185.
- [45] S. Newton and S. Mitra, "Self-organizing leader clustering in a neural network using a fuzzy learning rule," *SPIE Proc. 1565: Adaptive Signal Processing*, July 1991.
- [46] J. Rajapakse, O. Jakubowicz, and R. Acharya, "A real-time ART-1 based vision system for distortion invariant recognition and auto-association," in *Proc. 1990 Int. Joint Conf. Neural Networks* (Washington, DC), vol. 2, 1990, pp. 298–301.
- [47] D. Rumelhart and J. McClelland, Eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA: MIT Press, 1986.
- [48] E. Ruspini, "A new approach to clustering," *Inform. Contr.*, vol. 15, pp. 22–32, 1969.
- [49] E. Ruspini, "New experimental results in fuzzy clustering," *Inform. Sci.*, vol. 6, pp. 273–284, 1973.
- [50] P. Simpson, "Fuzzy adaptive resonance theory," *Southern Illinois University Neuroengineering Workshop*, Sept. 6–7, 1990, Carbondale, IL; also in *Proc. GD AI 90* (Ft. Worth, TX), Nov. 7–8, 1990; also published as General Dynamics Technical Report, GDE-ISG-PKS-11, April 1990.
- [51] P. Simpson, *Artificial Neural Systems: Foundations, Paradigms, Applications and Implementations*. Elmsford, NY: Pergamon Press, 1990.
- [52] P. Simpson, "Fuzzy min-max classification with neural networks," *Heuristics*, vol. 4, no. 1, pp. 1–9, 1991.
- [53] P. Simpson, "Fuzzy min-max neural networks," in *Proc. 1991 Int. Joint Conf. Neural Networks* (Singapore), Nov. 18–21, 1991, pp. 1658–1669.
- [54] P. Simpson, "Neural network paradigms," *AGARD Lecture Series 179: Artificial Neural Networks for Guidance and Control*, AGARD-LS-179, paper 2, NATO Publication: London, UK.
- [55] P. Simpson, "Neural networks for multisensor data fusion," *AGARD Lecture Series 179: Artificial Neural Networks for Guidance and Control*, AGARD-LS-179, paper 9, NATO Publication: London, UK.

- [56] P. Simpson, "Fuzzy min-max neural networks—Part 1: Classification," *IEEE Trans. Neural Networks*, vol. 3, pp. 776–786, Sept. 1992.
- [57] W. Wee, "On generalizations of adaptive algorithms and applications of the fuzzy sets concept to pattern classification," Ph.D. thesis, Purdue University, Lafayette, IN, 1967.
- [58] P. Werbos, "Beyond regression: New tools for prediction and analysis in the behavioral sciences, Ph.D. thesis, Harvard University, Cambridge, MA, 1974.
- [59] P. Werbos, "Neurocontrol and fuzzy logic—Connections and designs," *Int. J. Approximate Reasoning*, to be published.
- [60] B. Widrow and M. Hoff, "Adaptive switching circuits," *IRE Convention Record*, part IV, pp. 96–104, 1960.
- [61] R. Yager, "A measurement informational discussion on fuzzy union and intersection," *Int. J. Man-Machine Studies*, vol. 11, pp. 189–200, 1979.
- [62] L. Zadeh, "Fuzzy sets," *Inform. Contr.*, vol. 8, pp. 338–353, 1965.

Patrick K. Simpson (M'87) is a principal engineer with ORINCON Corporation, a small-business dedicated to the application of intelligent systems to difficult defense-related problems. Mr. Simpson is an active member of the IEEE, having served on the organization and program committees of several national and international conferences as well as being the current IEEE Neural Networks Council Vice-President. He was the Program Chairman of the first IEEE Conference on Neural Networks for Ocean Engineering and is an associate editor of the *Transactions on Neural Networks* and a recent NATO Lecture Series participant. {Mr. Simpson has written over 60 papers on the theory and application of neural networks, fuzzy systems, and artificial intelligence and is the author of the book *Artificial Neural Systems: Foundations, Paradigms, Applications, and Implementations*.