Fuzzy Min–Max Neural Networks—Part 1: Classification

Patrick K. Simpson

Abstract—**A supervised learning neural network classifier that utilizes fuzzy sets as pattern classes is described. Each fuzzy set is an aggregate (union) of fuzzy set hyperboxes. A fuzzy set hyperbox is an n-dimensional box defined by a min point and a max point with a corresponding membership function. The min–max points are determined using the fuzzy min–max learning algorithm, an expansion–contraction process that can learn nonlinear class boundaries in a single pass through the data and provides the ability to incorporate new and refine existing classes without retraining. The use of a fuzzy set approach to pattern classification inherently provides degree of membership information that is extremely useful in higher level decision making. This paper will describe the relationship between fuzzy sets and pattern classification. It explains the fuzzy min–max classifier neural network implementation, it outlines the learning and recall algorithms, and it provides several examples of operation that demonstrate the strong qualities of this new neural network classifier.**

I. Introduction

PATTERN classification is a key element to many engineering solutions. Sonar, radar, seismic, and diagnostic applications all require the ability to accurately classify a situation. Control, tracking, and prediction systems will often use classifiers to determine input–output relationships. Because of this wide range of applicability, pattern classification has been studied a great deal [13], [15], [19]. This paper describes a neural network classifier that creates classes by aggregating several smaller fuzzy sets into a single fuzzy set class. This technique, introduced in [42] as an extension of earlier work [41], can learn pattern classes in a single pass through the data, it can add new pattern classes on the fly, it can refine existing pattern classes as new information is received, and it uses simple operations that allow for quick execution.

Fuzzy min–max classification neural networks are built using hyperbox fuzzy sets. A hyperbox defines a region of the n-dimensional pattern space that has patterns with full class membership. A hyperbox is completely defined by its min point and its max point, and a membership function is defined with respect to these hyperbox min–max points. The min–max (hyperbox) membership function combination defines a fuzzy set, hyperbox fuzzy sets are aggregated to form a single fuzzy set class, and the resulting structure fits naturally into a neural network framework; hence this classification system is called a fuzzy min–max classification neural network. Learning in the fuzzy min–max classification neural network is performed by properly placing and adjusting hyperboxes in the pattern space.

Manuscript received May 19, 1991; revised Sept. 3, 1991. This work was supported by the General Dynamics Electronics Division Independent Research and Development Project 91015131.

The author is with the Orincon Corporation, 9363 Towne Center Drive, San Diego, CA 92121.

IEEE Log Number 9201458.

Fuzzy min–max classification neural network recall consists of computing the fuzzy union of the membership function values produced from each of the fuzzy set hyperboxes. Each of these subjects will be described in greater detail later in the paper.

There are several properties that a pattern classifier should possess. Each of these properties has motivated a portion of the development of the fuzzy min–max classification neural network. These properties are (in no particular order):

On-Line Adaptation: A pattern classifier should be able to learn new classes and refine existing classes quickly and without destroying old class information. This property is sometimes referred to as on-line adaptation or on-line learning. Grossberg [21] specifically identifies this property as a key problem associated with neural network design and refers to it as the stability–plasticity dilemma. Simply stated, the dilemma concerns the design of neural networks that can remain plastic enough to learn yet still be able to stabilize and recall stored pattern information. Many of the popular neural network and traditional pattern classification techniques utilize off-line adaptation. Each time new information is added to the classification system it requires a complete retraining of the system with both the old and the new information. As such, off-line adaptation can place huge demands on the memory requirements and it can lead to increasingly longer training times.

Nonlinear Separability: A pattern classifier should be able to build decision regions that separate classes of any shape and size, a property that we will refer to as nonlinear separability (of which linear separability is a special case). It is of historical value to note that the inability to perform nonlinear separation was considered to be a severe handicap of neural networks [31]. Fortunately, this handicap no longer exists. The advent of neural networks such as the Boltzmann machine [36], backpropagation [36], [47] the reduced coulomb energy network [34], and many others [40] provide this capability.

Overlapping Classes: In addition to pattern classes being nonlinearly separable, they also tend to overlap. A pattern classifier should have the ability to form a decision boundary that minimizes the amount of misclassification for all of the overlapping classes. The most prevalent method of minimizing misclassification is the construction of a Bayes classifier. Unfortunately, to build a Bayes classifier requires knowledge of the underlying probability density function for each class. This is information that is quite often unavailable. This means that the probability density functions, or their equivalents, must be found on the fly and, in the on-line adaptation case, constantly tuned to represent the current state of the data being received.

Training Time: One of the most crippling elements of nonlinear classification algorithms is the amount of time required to learn the decision boundaries of the classes. Some of the most successful algorithms are based upon either the minimization or the maximization of an objective function which can take from hundreds to thousands to sometimes millions of passes through the data set. Examples of such algorithms include back-propagation [36], [47] cascade-correlation [16], and the Boltzmann machine [36]. A very desirable property of a pattern classification approach able to learn nonlinear decision boundaries is a short training time. This property, when combined with the aforementioned on-line adaptation property, poses a formidable problem. Recently there have been a few neural networks that have had varying levels of success when addressing this problem, including the probabilistic neural network [44] and its recent extensions [45], and the reduced coulomb energy network [34].

Soft and Hard Decisions: A pattern classifier should be able to provide both soft and hard classification decisions. A hard, or crisp, decision is either 0 or 1. A pattern is either in a class or it is not. A soft decision provides a value that describes the degree to which a pattern fits within a class. There are many real-world problems that require soft decisions. As an example, a pattern classifier for object recognition might not be trained to recognize colors. It is simply not possible to train a classification system with every possible combination of the three primary colors, but it is possible to train a system to recognize the classes of the primary colors and then use a soft decision process to determine the degree to which each of the three colors is present.

Verification and Validation: It is important that a classifier, neural or traditional, have a mechanism for verifying and validating its performance in some way. Contour plots, scatter plots, and closed-form solutions have all been used to perform this function.

Tuning Parameters: A classifier should have as few parameters to tune in the system as possible. Ideally, a classifier system will have no parameters that need to be tuned during training. If there are parameters, the effect these parameters have on the system should be well understood.

Nonparametric Classification: Parametric classifiers assume *a priori* knowledge about the underlying probability density functions of each class. If this information is available, it is possible to construct very reliable pattern classifiers, but often this information is not available. When the classifier does not have *a priori* information available, it is called a nonparametric classifier. If the classifier is nonparametric, it should be able to describe the underlying distribution of the data in a way that still provides reliable class boundaries.

These properties became strict design considerations during the development of the fuzzy min–max classification neural network. This development has been an evolutionary process that began by trying to develop unsupervised learning clustering neural networks that were fast, efficient, and reliable. These developments led to a neural network paradigm that combined fuzzy logic with the ART1 neural network of Carpenter and Grossberg [8] to produce a fast and reliable analog pattern clustering system called fuzzy adaptive resonance theory (fuzzy ART) [39], which has since been extended and improved (see the companion paper [43] for a detailed description of this improved clustering neural network). During the period of time between the introduction of fuzzy ART and the writing of this paper, a second fuzzy ART has been independently introduced [9] that has many similar properties.

While enhancing the original fuzzy ART, it was realized that a similar approach could be taken for pattern classification. Both the original fuzzy ART network and the fuzzy min–max classification neural network utilize hyperbox fuzzy sets as the fundamental computing element. In this paper the aggregate of hyperbox fuzzy sets is used to define pattern classes. In the companion paper the hyperbox fuzzy sets are used to define pattern clusters. To eliminate any potential confusion between the original fuzzy ART [39] and the later fuzzy ART [9], to provide a firm relationship between the classification neural network presented here and the improved clustering neural network [43], and to emphasize the importance of the fuzzy (hyperbox) min–max structure of these two neural networks, the improved version of the original fuzzy ART neural network has been renamed the fuzzy min–max clustering neural network [43].

Following this introduction is a brief review of fuzzy sets and a brief survey of the work done in fuzzy sets and pattern classification. The third section describes fuzzy min–max classification, hyperbox fuzzy sets, and the way in which hyperbox fuzzy sets are aggregated to form pattern classes. Section IV explains the neural network implementation of fuzzy min–max classifiers. Section V outlines the fuzzy min–max classification learning algorithm, provides a brief discussion of decision boundaries, and offers some examples of operation. Section VI provides three examples of fuzzy min–max classification operation that includes overlapping classes, nonlinear classification, and a comparison with classical pattern classification techniques. Section VII briefly describes the relationship between the fuzzy min–max classification neural network and other neural network classifiers. The final section contains a summary of the fuzzy min–max classification neural network as well as a discussion of future work.

II. Fuzzy Sets and Pattern Classification

Fuzzy sets were introduced by Zadeh [51] as a means of representing and manipulating data that were 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 gave pattern classification and control engineers the ability to measure the degree to which a pattern was present or a situation was occurring. On the contrary, 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 if an event will occur, measuring the chance with which a given event is expected to occur. In situations such as the 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 from 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.

Pattern classification partitions $\mathcal{R}^n$ into categories (sets) and then assigns a point $x \in \mathcal{R}^n$ to one of those categories. If x does not fit directly within a category, a "goodness of fit" is usually reported. If it is assumed that the space of all set objects is $\mathcal{R}^n$, the notion of a fuzzy set as a pattern class is becomes clear. By viewing each category as a fuzzy set and identifying the assignment operation with the membership function, a direct relationship between fuzzy sets and pattern classification is realized. There are many data sets that have built in ambiguities and there are never enough data to completely describe all the possible patterns that will be seen. By employing fuzzy sets as pattern classes, it is possible to describe the degree to which a pattern belongs to one class or another. This relationship was realized very early in the development of fuzzy sets. In Zadeh's seminal paper [51] he describes a preceding technical memorandum that applied fuzzy sets to pattern recognition [1]. This paper was published later [2].

In this paper, a fuzzy set formed from the union of a collection of fuzzy sets is used to describe each class. The fuzzy set class is constructed by aggregating together the collection of hyperbox fuzzy sets that belong to each class. In the following sections are both historical and rigorous definitions of fuzzy sets and some of their common operations.

A. Fuzzy Set Definition

The following definition has been paraphrased from Zadeh's seminal paper [51] to illustrate the simplicity and the significance of fuzzy sets as well as to emphasize the relationship between pattern classes 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 provide 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.

A more formal definition of fuzzy sets has been presented by many researchers and theoreticians. The definition offered here was abstracted from Kandel [25]. 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.

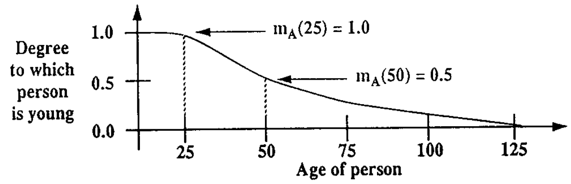


Fig. 1. This membership function describes the relationship between a person's age and the degree to which a person is considered to be young. This membership function determines that a 25-year-old person belongs to A twice as much as a 50-year-old person.

As an example, let $\mathcal{X}$ represent the ages of all people. The subset A of $\mathcal{X}$ that represents those people that are young is a fuzzy set with the membership function shown in Fig. 1.

B. Fuzzy Set Operations

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 $\mathcal{X}$ to be the universe of discourse, $A \in \mathcal{X}$ and $B \in \mathcal{X}$, these operations are defined as follows.

Comparison: Is $A = B$?

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

Containment: Is $A \subset B$?

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

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. [14] and [50]), the most common, and by far the simplest, union is defined as

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

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

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

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

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

In addition to these operations, De Morgan's laws, the distributive laws, algebraic operations such as addition and multiplications, and the notion of convexity have fuzzy set equivalents [51].

C. Fuzzy Sets and Pattern Classes

The combination of fuzzy sets and pattern classification has been examined by many people. As mentioned earlier, Zadeh [51] had stated the relationship between pattern classes and fuzzy sets in the first paper on fuzzy sets. In the same paper, Zadeh went on to discuss how hyperplanes could be used to separate fuzzy sets. In a much later paper by Keller and

Hunt [26], fuzzy hyperplane decision boundaries replaced the crisp decision boundaries of the perceptron neural network. Other work, by Bellman *et al.* [2] and Wee [46], described fuzzy pattern classification. Ruspini [37] has been credited with many of the insights that have led to effective and efficient fuzzy pattern classification algorithms [5]. Bezdek [3] has studied fuzzy objective functions for pattern classification in great detail. In particular, Bezdek developed a fuzzy k-nearest-neighbor classifier [4] that performed well with various data sets and was shown to subsume crisp k-nearest-neighbor classifiers. Bezdek has also developed an extensive body of work dealing with the fuzzy c-means algorithm [3], [5]. Others who have worked with fuzzy k-nearest-neighbor classifiers include Keller *et al.* [26] and Jozwik [24].

Recently there has been a great amount of interest in the synergistic combination of neural networks and fuzzy systems [6]. Some of this work was done over a decade ago [30], [46] and some has appeared recently [23], [26], [32], [48]. Many of these efforts have focused upon methods for implementing fuzzy rules in a neural network framework and techniques for parallelizing successful fuzzy control system applications. The work presented here illustrates that fuzzy sets and neural networks can be effectively merged by utilizing neural network nodes as fuzzy sets and using fuzzy set operations during learning and recall.

III. Fuzzy Set Classes as Aggregates of Hyperbox Fuzzy Sets

Pattern classification can be described by Fig. 2, which has been abstracted from Duda and Hart [15]. An input pattern A_h is passed through each of the m discriminant, or characteristic, functions and the characteristic function with the largest value (and inherently the class the discriminant function represents) is selected. There have been many different methods of implementing the discriminant functions including the use of rules—syntatic pattern classification [18], probability density functions—Bayes classification [15], pattern exemplars with a corresponding distance function—nearest-neighbor classification [12], linear functions—linear discriminant analysis [15], fuzzy sets, and, most recently, neural networks [39].

Within the field of neural networks there have been several different approaches taken toward producing effective discriminant functions. Some neural networks make use of the statistical mean and variance of data clusters to define the center and size of pattern classes, such as the probabilistic neural network [44], learning vector quantization [29], and radial basis functions [7]. Some neural networks create linear discriminant functions that partition the n-dimensional pattern space, such as perceptrons [35] and the adaline [49]. Some neural networks minimize or maximize a cost function that gives an input–output mapping that in turn produces nonlinear decision boundaries, such as back-propagation [36], [47], the Boltzmann machine [36], and cascade-correlation [16]. Some neural networks build decision boundaries by creating subsets of the pattern space, such as the related coulomb energy/Reilly–Cooper–Elbaum (RCE) neural network, and the hyperspherical attractor neural network [28].

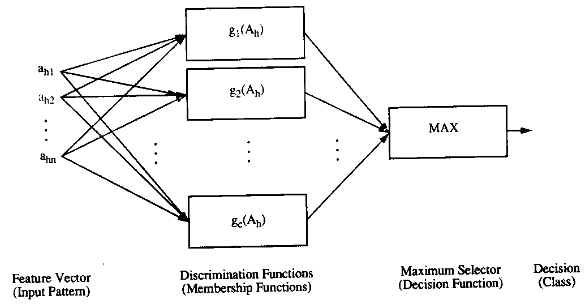


Fig. 2. A pattern classification system is shown. An input pattern A_h is passed through each of the m discriminant functions where each discriminant function represents a pattern class. The values of the discriminant functions are compared, and the one with the largest value is used to identify the pattern class.

The fuzzy min–max classification neural network is an example of this last type of neural network classifier. Hyperboxes, defined by pairs of min–max points, and their corresponding membership functions are used to create fuzzy subsets of the n-dimensional pattern space. One of the most important properties of this approach is that the majority of the processing is concerned with finding and fine-tuning the boundaries of the classes. By relegating these operations to simple compare, add, and subtract operations, the resulting learning algorithm is extremely efficient.

A. Min–Max Points, Hyperbox Fuzzy Sets, and Pattern Classes

An illustration of the min and max points in a three-dimensional hyperperbox 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 the 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 $\mathcal{R}^n$ to I^n destroys a small amount of absolute information when the range of each dimension is known prior to rescaling and it preserves all of the relative information between the various dimensions. In addition, by constraining the pattern space to I^n, it automatically becomes a compact space amenable to powerful function approximation theorems such as the Stone–Weierstrass theorem [11].

The aggregation of several hyperboxes in I^2 is illustrated for a two-class problem in Fig. 4. Let each hyperbox fuzzy set, B_j, be defined by the ordered set

$$B_j = \{X, V_j, W_j, f(X, V_j, W_j)\} \qquad \forall\ X \in I^n.$$

Using this definition of a hyperbox fuzzy set, the aggregate

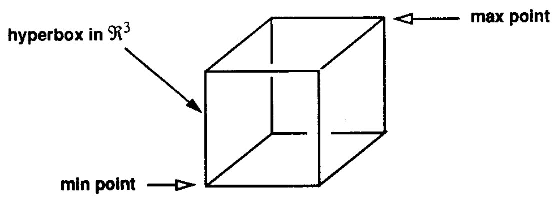


Fig. 3. The min–max hyperbox $B_j = \{V_j, W_j\}$ in $\mathcal{R}^3$ is shown. The min and max points are all that are required to define the hyperbox. A membership function is associated with the hyperbox that determines the degree to which any point $x \in \mathcal{R}^3$ is contained within the box. A collection of these boxes forms a pattern class.

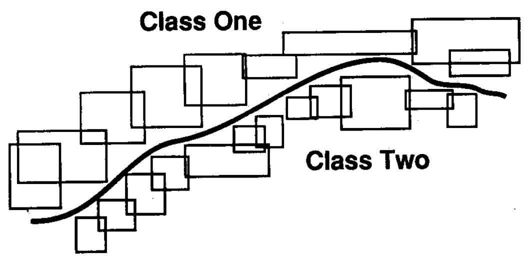


Fig. 4. An example of fuzzy min–max hyperboxes placed along the boundary of a two-class problem is illustrated. Note that the hyperboxes between the classes are nonoverlapping.

fuzzy set that defines the kth pattern class C_k is defined as

$$C_k = \bigcup_{j \in K} B_j,$$

where K is the index set of those hyperboxes associated with class k. Note that the union operation in fuzzy sets is typically the max of all of the associated fuzzy set membership functions.

The learning algorithm described below allows overlapping hyperboxes from the same class and eliminates the overlap between hyperboxes from separate classes. This does not mean that the fuzzy sets do not overlap, only that those portions of the fuzzy set representing full membership are nonoverlapping. Using this configuration, it is possible to define crisp class boundaries as a special case. These class boundaries are defined as those points where the membership values are equal.

B. Hypercube Membership Function

The membership function for the jth hyperbox $b_j(A_h)$, $0 \le b_j(A_h) \le 1$, must measure the degree to which the hth input pattern A_h falls outside of the hyperbox B_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. Also, as $b_j(A_h)$ approaches 1, the point should be more contained by the hyperbox, with the value 1 representing complete hyperbox containment. The function that meets all these criteria is the sum of two complements—the average amount of max point violations and the average amount of min point violations. The resulting membership function is defined as

$$b_j(A_h) = \frac{1}{2n}\sum_{i=1}^{n}[\max(0, 1 - \max(0, \gamma \min(1, a_{hi} - w_{ji}))) + \max(0, 1 - \max(0, \gamma \min(1, v_{ji} - a_{hi})))], \quad (1)$$

where $A_h = (a_{h1}, a_{h2}, \cdots, a_{hn}) \in I^n$ is the hth input pattern, $V_j = (v_{j1}, v_{j2}, \cdots, v_{jn})$ is the min point for B_j, $W_j = (w_{j1}, w_{j2}, \cdots, w_{jn})$ is the max point for B_j, and γ is the sensitivity parameter that regulates how fast the membership values decrease as the distance between A_h and B_j increases. An example of this membership function for two dimensions is shown in Fig. 5.

IV. Implementing the Fuzzy Min–Max Classifier Neural Network

Implementing the fuzzy min–max classifier as a neural network, it is possible to immediately exploit the parallel nature of the classifier and provide a mechanism for fast and efficient implementations. The neural network that implements the fuzzy min–max classifier is shown in Fig. 6. The topology of this neural network grows to meet the demands of the problem. Both F_B and F_C. Each F_B node in this three-layer neural network represents a hyperbox fuzzy set where the F_A to F_B connections are the min–max points and the F_B transfer function is the hyperbox membership function defined by (1). The min points are stored in the matrix V and the max points are stored in the matrix W. The connections are adjusted using the learning algorithm described in Section V. A detailed view of the jth F_B node is shown in Fig. 7. The connections between the F_B and F_C nodes are binary valued and stored in the matrix U. The equation for assigning the values to the F_B to F_C connections is

$$u_{jk} = \begin{cases} 1 & \text{if } b_j \text{ is a hyperbox for class } c_k \\ 0 & \text{otherwise,} \end{cases}$$

where b_j is the jth F_B node and c_k is the kth F_C node. Each F_C node represents a class. The output of the F_C node represents the degree to which the input pattern A_h fits within the class k. The transfer function for each of the F_C nodes performs the fuzzy union of the appropriate hyperbox fuzzy set values. This operation is defined as

$$c_k = \max_{j=1}^{m} b_j u_{jk} \quad (2)$$

for each of the q F_C nodes. There are two main ways that the outputs of the F_C class nodes can be utilized. If a soft decision is required, the outputs are utilized directly. If a hard decision is required, the F_C node with the highest value is located and its node value is set to 1 to indicate that it is the closest pattern class, and the remaining F_C node values are set to 0. This last operation is commonly referred to as a winner-take-all response [29].

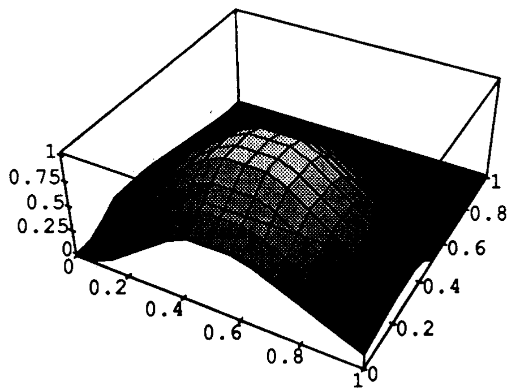


Fig. 5. The membership function for two dimensions is shown. The min point is at $V_j = (0.4, 0.2)$ and the max point is at $W_j = (0.6, 0.4)$. The sensitivity parameter $\gamma = 4$ is set to produce a moderately quick decrease from full membership to no membership.

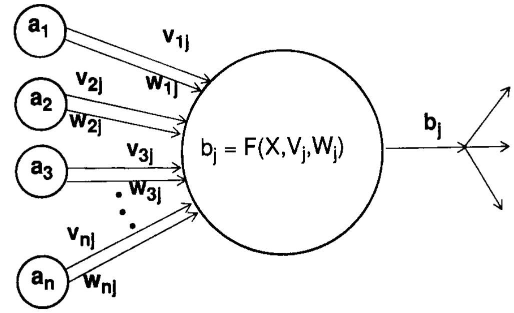


Fig. 7. The implementation of a hyperbox and its associated membership function as a neural network assembly is shown for the jth F_B node b_j. The input nodes accept each dimension of the hth input A_h. There are two connections from each input node to the output node, one connection represents the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

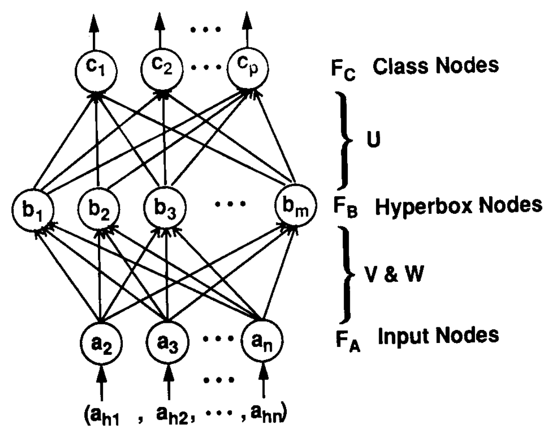


Fig. 6. The three-layer neural network that implements the fuzzy min–max neural network is shown. The input layer $F_A = (a_1, a_2, \cdots, a_n)$ has n processing elements, one for each of the n dimensions of the input pattern A_h. There are two sets of connections between each input node and each of the m hyperbox fuzzy set nodes found in the layer $F_B = (b_1, b_2, \cdots, b_m)$. These dual connections are adjusted using the fuzzy min–max classification learning algorithm. There are two sets of connections that emanate from F_A and about the jth F_B node—the min vector V_j and the max vector W_j. The connections between the F_B nodes and the p output nodes $F_C = (c_1, c_2, \cdots, c_p)$ are binary valued and are determined as each F_B node is added during learning. Each F_C node represents a pattern class.

V. Fuzzy Min–Max Classification Learning Algorithm

Fuzzy min–max learning is an expansion/contraction process. The training set $\mathcal{D}$ consists of a set of M ordered pairs $\{X_h, d_h\}$, where $X_h = (x_{h1}, x_{h2}, \cdots, x_{hn}) \in I^n$ is the input pattern and $d_h \in \{1, 2, \cdots, m\}$ is the index of one of the m classes. Note that X_n and A_n are both used to represent input patterns. The learning process begins by selecting an ordered pair from $\mathcal{D}$ and finding a hyperbox for the same class that can expand (if necessary) to include the input. If a hyperbox cannot be found that meets the expansion criteria, a new hyperbox is formed and added to the neural network. This growth process allows classes to be formed that are nonlinearly separable (see the example in subsection VI-B), it allows existing classes to be refined over time, and it allows new classes to be added without retraining. One of the residuals of hyperbox expansion is overlapping hyperboxes. Hyperbox overlap is not a problem when the overlap occurs between hyperboxes representing the same class. When the overlap occurs between hyperboxes that represent different classes, the overlap is eliminated using a contraction process. Note that the contraction process only eliminates the overlap between those portions of the fuzzy set hyperboxes from separate classes that have full membership. There is still overlap between nonunit valued members of each of the fuzzy set hyperboxes.

In summary, the fuzzy min–max classification learning algorithm is a three-step process:

1) *Expansion:* Identify the hyperbox that can expand and expand it. If an expandable hyperbox cannot be found, add a new hyperbox for that class.
2) *Overlap Test:* Determine if any overlap exists between hyperboxes from different classes.
3) *Contraction:* If overlap between hyperboxes that represent different classes does exist, eliminate the overlap by minimally adjusting each of the hyperboxes.

The following three subsections describe these operations in greater detail.

A. Hyperbox Expansion

Given an ordered pair $\{X_h, d_h\} \in D$, find the hyperbox B_j that provides the highest degree of membership, allows expansion (if needed), and represents the same class as d_h. The degree of membership is measured using (1). The maximum size of a hyperbox is bounded above by $0 \leq \theta \leq 1$, a user-defined value. For the hyperbox B_j to expand to include X_h, the following constraint must be met:

$$n\theta \geq \sum_{i=1}^{n}(\max(w_{ji}, x_{hi}) - \min(v_{ji}, x_{hi})). \tag{3}$$

If the expansion criterion has been met for hyperbox B_j, the min point of the hyperbox is adjusted using the equation

$$v_{ji}^{\text{new}} = \min(v_{ji}^{\text{old}}, x_{hi}) \qquad \forall\, i = 1, 2, \cdots, n, \tag{4}$$

and the max point is adjusted using the equation

$$w_{ji}^{\text{new}} = \max(w_{ji}^{\text{old}}, x_{hi}) \qquad \forall\, i = 1, 2, \cdots, n. \tag{5}$$

Note that these expansion operations are similar to the fuzzy intersection and fuzzy union operations, but should not be interpreted as such. Also, Moore [33] has used (4) in a variant of ART1.

B. Hyperbox Overlap Test

As previously stated, overlapping hyperboxes that represent the same class do not present a class formation problem; it is necessary to eliminate overlap between hyperboxes that represent different classes.

To determine if this expansion created any overlap, a dimension by dimension comparison between hyperboxes is performed. If, for each dimension, at least one of the following four cases is satisfied, then overlap exists between the two hyperboxes. Assume that the hyperbox B_j was expanded in the previous step and that the hyperbox B_k represents another class and is being tested for possible overlap. While testing for the overlap, the smallest overlap along any dimension and the index of the dimension is saved for use during the contraction portion of the learning process. Assuming $\delta^{\text{old}} = 1$ initially, the four test cases and the corresponding minimum overlap value for the ith dimension are as follows.

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

$$\delta^{\text{new}} = \min(w_{ji} - v_{ki}, \delta^{\text{old}}).$$

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

$$\delta^{\text{new}} = \min(w_{ki} - v_{ji}, \delta^{\text{old}}).$$

Case 3: $v_{ji} < v_{ki} < w_{ki} < w_{ji}$,

$$\delta^{\text{new}} = \min(\min(w_{ki} - v_{ji}, w_{ji} - v_{ki}), \delta^{\text{old}}).$$

Case 4: $v_{ki} < v_{ji} < w_{ji} < w_{ki}$,

$$\delta^{\text{new}} = \min(\min(w_{ji} - v_{ki}, w_{ki} - v_{ji}), \delta^{\text{old}}).$$

If $\delta^{\text{old}} - \delta^{\text{new}} > 0$, then $\Delta = i$ and $\delta^{\text{old}} = \delta^{\text{new}}$, signifying that there was overlap for the Δth dimension and overlap testing will proceed with the next dimension. If not, the testing stops and the minimum overlap index variable is set to indicate that the next contraction step is not necessary, i.e., $\Delta = -1$.

C. Hyperbox Contraction

If $\Delta > 0$, then the Δth dimensions of the two hyperboxes are adjusted. Only one of the n dimensions is adjusted in each of the hyperboxes to keep the hyperbox size as large as possible and minimally impact the shape of the hyperboxes being formed. It is felt that this minimal disturbance principle will provide more robust pattern classification. To determine the proper adjustment to make, the same four cases are examined.

Case 1: $v_{j\Delta} < v_{k\Delta} < w_{j\Delta} < w_{k\Delta}$,

$$w_{j\Delta}^{\text{new}} = v_{k\Delta}^{\text{new}} = \frac{w_{j\Delta}^{\text{old}} + v_{k\Delta}^{\text{old}}}{2}.$$

Case 2: $v_{k\Delta} < v_{j\Delta} < w_{k\Delta} < w_{j\Delta}$,

$$w_{k\Delta}^{\text{new}} = v_{j\Delta}^{\text{new}} = \frac{w_{k\Delta}^{\text{old}} + v_{j\Delta}^{\text{old}}}{2}.$$

Case 3a: $v_{j\Delta} < v_{k\Delta} < w_{k\Delta} < w_{j\Delta}$ and $(w_{k\Delta} - v_{j\Delta}) < (w_{j\Delta} - v_{k\Delta})$,

$$v_{j\Delta}^{\text{new}} = w_{k\Delta}^{\text{old}}.$$

Case 3b: $v_{j\Delta} < v_{k\Delta} < w_{k\Delta} < w_{j\Delta}$ and $(w_{k\Delta} - v_{j\Delta}) > (w_{j\Delta} - v_{k\Delta})$,

$$w_{j\Delta}^{\text{new}} = v_{k\Delta}^{\text{old}}.$$

Case 4a: $v_{k\Delta} < v_{j\Delta} < w_{j\Delta} < w_{k\Delta}$ and $(w_{k\Delta} - v_{j\Delta}) < (w_{j\Delta} - v_{k\Delta})$,

$$w_{k\Delta}^{\text{new}} = v_{j\Delta}^{\text{old}}.$$

Case 4b: $v_{k\Delta} < v_{j\Delta} < w_{j\Delta} < w_{k\Delta}$ and $(w_{k\Delta} - v_{j\Delta}) > (w_{j\Delta} - v_{k\Delta})$,

$$v_{k\Delta}^{\text{new}} = w_{j\Delta}^{\text{old}}.$$

D. Examples of the Expansion–Contraction Process

Fig. 8 illustrates how the four two-dimensional patterns shown in Fig. 8(a) are used to construct a two-hyperbox classifier with a maximum hyperbox size of $\theta = 0.3$. This step-by-step example illustrates how hyperboxes are formed and how overlapping hyperboxes are adjusted.

VI. Examples of Operation

Three examples have been selected to illustrate the operation and performance of the fuzzy min–max classification neural network. The first data set, a four-class problem with all four of the classes overlapping, will illustrate the fuzzy min–max classification network's performance on overlapping class data. The second data set, a two-class nested spiral type of data set, will illustrate how this network performs on nonlinearly separable classification problems. These first two examples utilize data sets from I^2 to allow visual verification of class formation. The third data set comprises Fisher's iris data [17]. The performance on this data set is compared with results of other classifiers and clustering algorithms reported elsewhere. Each of these examples will be covered in separate subsections that follow.

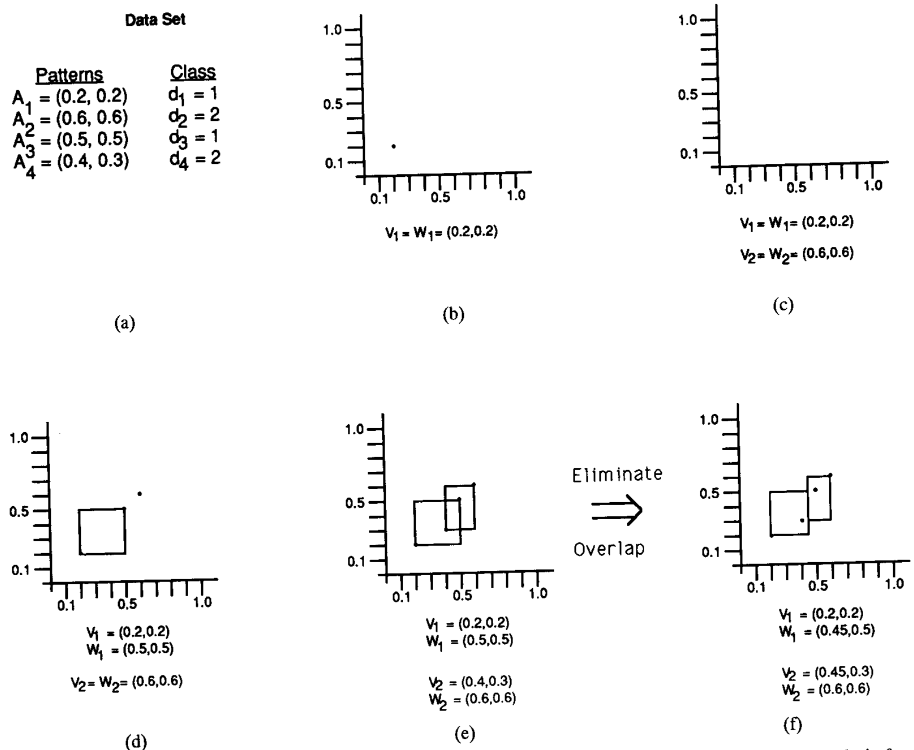


Fig. 8. An example illustrating the learning algorithm for a two classes problem is shown. The data set used for this example is found in panel (a). The growth parameter, the maximum hyperbox size, is $\theta = 0.3$. The storage sequence is as follows. (a) Initially, there are no hyperboxes, so the first input pattern, $A_1 = (0.2, 0.2) \in d_1$, creates a set of min and max connections $V_1 = W_1 = (0.2, 0.2)$ for the hyperbox node b_1 and a class node c_1 with a unit valued connection from b_1 to c_1. (b) The second input pattern, $A_2 = (0.6, 0.6) \in d_2$, creates a second hyperbox node b_2 with the min and max connection values $V_2 = W_2 = (0.6, 0.6)$ and a second class node c_2 with a unit valued connection from b_2 to c_2. (c) The third input pattern, $A_3 = (0.5, 0.5) \in d_1$, is tested for inclusion in hyperbox b_1 and found to be acceptable and it is expanded to include this input pattern, resulting in a set of b_1 min connections $V_1 = (0.2, 0.2)$ and max connections $W_1 = (0.5, 0.5)$. (d) The fourth input pattern, $A_4 = (0.4, 0.3) \in d_2$, is tested for inclusion in hyperbox b_2 and found to be acceptable to be expanded to include this input pattern, resulting in a set of b_2 min connections $V_2 = (0.4, 0.3)$ and max connections $W_2 = (0.6, 0.6)$. (e) The last expansion created an overlap of hyperboxes b_1 and b_2 and, since they are representing different classes, this overlap must be eliminated. Because b_2's min point overlapped with bs_1's max point, these two points are manipulated to remove the overlap, resulting in the new min and max points $V_2 = (0.45, 0.3)$ and $W_1 = (0.45, 0.5)$.

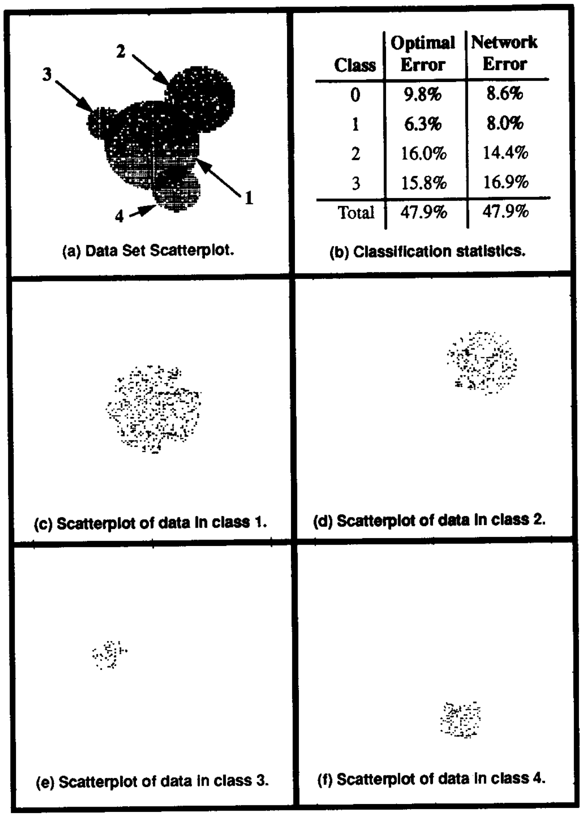
Example 1: Four Overlapping Classes

Four classes of data were randomly generated using an equiprobable distribution. Ten thousand data points were generated for training and another 1000 were generated for testing. A scatter plot of the training set is shown in Fig. 9(a). Note that each of these circular classes has different sizes and different overlaps. The "optimal" misclassification error was numerically computed by randomly generating points in each of the circles and computing how many of the total points between each set of overlapping classes there were, dividing this number by 2, and then dividing that number by the total number of points collected for each of the classes. Runs of 100 000 randomly sampled points were used to collect these statistics, which are shown in Fig. 9(b).

The fuzzy min–max classification neural network was used to classify the same data. A single pass through 10 000 training points with a growth parameter of $\theta = 0.15$ was used. A separate test set of 1000 samples was used both to collect classification performance statistics and to illustrate where the class boundaries were formed. Fig. 9(b) provides the performance for the fuzzy min–max classification network and panels (c) through (f) of Fig. 9 illustrate the class boundaries. Note that the total error generated by the network was identical to the calculated optimal error, but the errors for each class were different. Some of the boundaries formed resulted in fewer errors for one class relative to the other. As an example, class 0, the large circle that overlaps with the other three circular classes, and class 2, the smallest of the four circular classes, produced less error than the computed "optimal," but the other two classes created slightly more error.

Example 2: Nested Spiral Type Classes

The data set used in this example was motivated by the cover of Minsky and Papert's book *Perceptrons* [31], which was extremely influential in the late 1960's and through most of the 1970's. Like the preceding example, 10 000 data points were equiprobably randomly selected for training. A scatter plot of this data set is shown in Fig. 10(a). The test set consisted of another 10 000 equiprobably randomly selected data points. The training was performed in a single pass through the data set and testing produced perfect recall. The growth parameter was $\theta = 0.175$, which resulted in the formation of 90 hyperboxes. Scatter plots of the points associated with classes 1 and 2 during the test pass are shown in panels (b) and (c) of Fig. 10. To understand where the



Class	Optimal Error	Network Error
0	9.8%	8.6%
1	6.3%	8.0%
2	16.0%	14.4%
3	15.8%	16.9%
Total	47.9%	47.9%

Fig. 9. The results of the classification of four overlapping classes are shown. The data set consisted of 10 000 equiprobably selected points in I^2 shown in the scatter plot in panel (a). An "optimal" misclassification error was computed for each of the classes and was compared with the misclassification error produced by the network. A table of these statistics is shown in panel (b). One thousand samples were equiprobably randomly selected for the testing. Scatter plots of where those data points were classified illustrates where the class boundaries (defined as those points that have a higher degree of membership in one class than they do in any other). The scatter plots of these classifications are shown in panels (c) through (f).

decision boundaries are for this problem, a separate test set of 10 000 equiprobably randomly selected data points across all of I^2 was passed through the classifier and these points were associated with their best matching class. The scatter plots of the results of this test for classes 1 and 2 are shown in panels (d) and (e) of Fig. 10.

Example 3: Iris Data Comparative Performance

The third data set comprised the Fisher iris data [17]. This data set was selected because of the tremendous number of results available from a wide range of classification techniques that will provide a measure of relative performance. The iris data consisted of 150 four-dimensional feature vectors (patterns) in three separate classes, 50 for each class. The test set used to determine the performance of the fuzzy min–max classification neural network used 25 randomly selected patterns from each class for training and the remaining 75 for testing. The growth parameter was $\theta = 0.0175$ and the number of hyperboxes built was 48. Training was performed in a single pass through the data set. Table I compares the

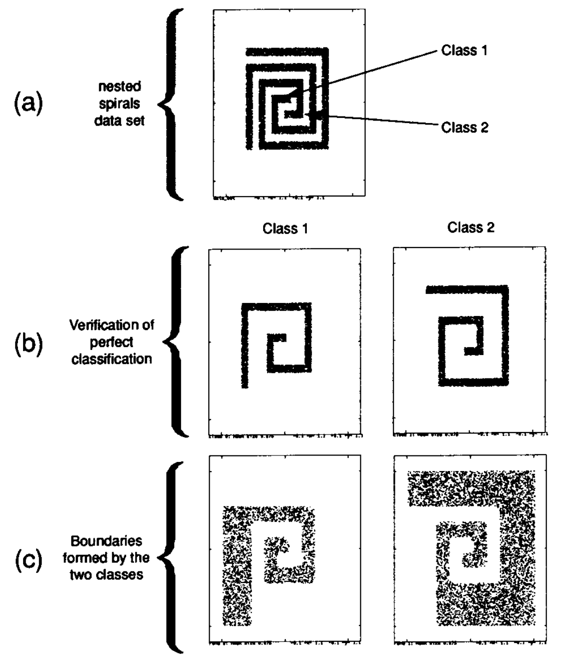


Fig. 10. The results of the classification of two nested spirals; (a) shows the data, (b) and (c) show the classification results, and (d) and (e) show the class boundaries.

performance of the fuzzy min–max classifier neural network with several other neural, fuzzy, and traditional classifiers on the same data set. This comparative performance test demonstrates that this classification technique performs at least as well as other traditional, fuzzy, and neural network classifiers.

VII. Comparing the Fuzzy Min–Max Classification Neural Net with Other Neural Networks

It is useful to compare the similarities and differences between the fuzzy min–max classification neural network and other classifiers. The one element that differs between the fuzzy min–max neural network classifier and each of the classifiers described below is the relationship between fuzzy sets as pattern classes and its direct implementation as a neural network.

A. *k-Nearest-Neighbor Classifier*

The fuzzy min–max neural network classifier collapses to the k-nearest-neighbor classifier when the size of the hyperboxes are set to 0 (i.e., $\theta = 0$). It might be that the expansion of a point (pattern exemplar) to a hyperbox has several computational advantages, with little loss in classification performance. This relationship is the subject of future work and will be examined in greater detail.

B. *Probabilistic Neural Network*

The probabilistic neural network (PNN) [44] is similar

TABLE I
A Comparison of the Classification Performance of Various Traditional, Fuzzy, and Neural Classifiers

Technique	No. Wrong	Remarks
Bayes classifier [16][1]	2	Very amenable to this type of data
k-nearest neighbor [16][1]	4	Scales up poorly
Fuzzy k-NN [4][2]	4	Allows fuzzy labels for data points
Fisher ratios [16][1]	3	Limited to linear discrimination
Ho–Kashyap [16][1]	2	Requires matrix inversion
Perceptron [27][3]	3	Limited to linear discrimination
Fuzzy perceptron [27][3]	2	Fuzzifies linear boundaries
Fuzzy min–max network	2	Single pass learning, learns on-line

[1] Training set comprised 75 data points (25 from each class) and test set comprised the remaining data points.

[2] Training data comprised 36 data points (12 from each class) and test set comprised another 36 data points. The results were then scaled up to 150 total points for the comparison.

[3] Training and testing data were the same.

to the fuzzy min–max neural network in that it associates membership functions (in this case Gaussian density functions) with pattern classes, it utilizes a union operation, and it grows to meet the needs of the problem. The differences between the PNN and the fuzzy min–max neural network classifier are that (1) the PNN stores each data set pattern in the network and the fuzzy min–max neural network classifier utilizes hyperboxes, (2) the PNN uses a Euclidian distance metric and the probability density function and the fuzzy min–max neural network classifier uses a Hamming-distance-based membership function, and (3) the PNN normalizes its data to unit length, which destroys the relative magnitude information, and the fuzzy min–max neural network classifier only rescales the data and retains the relative magnitude information.

C. Reduced Coulomb Energy Network

The reduced Coulomb energy (RCE) network [34] is similar to the fuzzy min–max neural network in that it utilizes an expansion contraction process to adjust subspaces of the n-dimensional feature space and that it grows to fit the needs of the problem. The differences between the RCE network and the fuzzy min–max neural network classifier are that (1) the RCE network utilizes hyperspheres and the fuzzy min–max neural network classifier uses hyperboxes, (2) the RCE network adjusts each dimension of the overlapping n-dimensional hyperspheres while the fuzzy min–max neural network adjusts the one dimension of the overlapping n-dimensional hyperboxes that will produce the minimum amount of change, and (3) the RCE network uses a Euclidian distance function during its operation and the fuzzy min–max neural network classifier uses a Hamming-distance-based membership function. These differences are also germane to the hyperspherical attractor neural network [28], a neural network classifier that is very similar to the RCE network.

D. Nested Generalized Exemplar Theory

Nested generalized examplars is a machine learning theory developed by Salzberg [38] that utilizes hyperboxes (referred to as hyperrectangles by Salzberg) to represent classes. The differences between nested generalized exemplar (NGE) theory and the fuzzy min–max neural network classifier are that (1) the focus of NGE theory is the ability to represent hierarchical relationships between data items through overlapping and nested hyperboxes and the fuzzy min–max neural network classifier focuses on providing the best possible classification performance with the least amount of computational effort, and (2) the learning algorithm for NGE theory uses a Euclidean-distance-based similarity measure that looks for closest and next closest matches and the fuzzy min–max neural network classifier provides a degree of match for each class represented by the system. Clearly, there are many similarities between NGE theory, which has been developed within the machine learning community, and the fuzzy min–max neural network classifier, which has grown from developments in the neural network community. The relationships and possible synergisms between these two classifiers are the subject of future research.

E. ARTMAP

There are similarities between how ARTMAP [10] and the fuzzy min–max neural network classifier process information. Each of these networks uses an expanding set of processing elements that grow to fit the needs of the problem and are constrained by a growth parameter and they use a similar mapping mechanism that links processing elements together. The differences between ARTMAP the fuzzy min–max neural network classifier are that (1) ARTMAP processes binary valued patterns and the fuzzy min–max neural network classifier processes analog patterns, (2) ARTMAP is a pattern matching neural network that links two clustering neural networks together through a set of map field connections and the fuzzy min–max neural network classifier performs a fuzzy union to perform class associations, and (3) ARTMAP does not attempt to directly resolve overlaps between processing elements that represent different classes and the fuzzy min–max neural network classifier does.

VIII. Summary and Further Work

A neural network classifier that utilizes min–max hyperboxes as fuzzy sets that are aggregated into fuzzy set classes is introduced. This learning algorithm has the ability to learn on-line and in a single pass through the data. Examples are provided that demonstrate the ability of the fuzzy min–max classification neural network to find reasonable decision boundaries in overlapping classes, learn highly nonlinear decision boundaries, and provide results on a standard data set that was equivalent to other neural and traditional classifiers.

The operations in the fuzzy min–max classifier are primarily adds and compares that can be implemented using relatively low single precision arithmetic operations. This simple design provides excellent opportunities for quick execution in parallel hardware.

Future work includes a mathematical description of the decision boundaries that are formed, a study of the effects of the maximum hyperbox size θ on classification accuracy, further exercises using higher dimensional real-world data, and the extension of this classification technique to control and prediction problems.

Acknowledgment

The author would like to thank Dr. L. Firestone, Dr. J. Bezdek, and Dr. R. C. Eberhart for many insightful suggestions that helped in properly representing this work. The author also wishes to recognize J. H. McBeth, S. A. Luse, and R. O. Deich for many helpful comments.

References

[1] R. Bellman, R. Kalaba, and L. Zadeh, "Abstraction and pattern classification," *RAND Memorandum,* RM-4307-PR, Oct. 1964.
[2] R. Bellman, R. Kalaba, and L. Zadeh, "Abstraction and pattern classification," *J. Math. Anal. Appl.,* vol. 13, pp. 1–7, 1966.
[3] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms.* New York: Plenum Press, 1981.
[4] J. Bezdek, S. Chuah, and D. Leep, "Generalized k-nearest neighbor rules," *Fuzzy Sets and Systems,* vol. 18, pp. 237–256, 1986.
[5] J. Bezdek, "Some nonstandard clustering algorithms," in *NATO ASI Series, vol. G14, Developments in Numerical Ecology,* P. Legendre and L. Legendre, Eds. Berlin: Springer-Verlag, 1987.
[6] J. Bezdek, "Fuzzy models and computational neural networks," presented at *IEEE Conference on Neural Networks for Ocean Engineering,* Washington, DC, Aug. 15, 1991.
[7] D. Broomhead and D. Lowe, "Multivariable function interpolation and adaptive networks," *Complex Systems,* vol. 2, pp. 321–355, 1988.
[8] 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.
[9] 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.
[10] G. Carpenter, S. Grossberg, and J. Reynolds, "ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network," *Neural Networks,* vol. 4, pp. 565–588.
[11] N. Cotter, "The Stone–Weierstrass theorem and its relationship to neural networks," *IEEE Trans. Neural Networks,* vol. 1, pp. 290–295, 1991.
[12] T. Cover and P. Hart, "Nearest-neighbor pattern classification," *IEEE Trans. Inform. Theory,* vol. 13, pp. 21–27, 1967.
[13] P. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach.* Englewood Cliffs, NJ: Prentice-Hall, 1982.
[14] D. Dubois and H. Prade, "A review of fuzzy sets aggregation connectives," *Inform. Sci.,* vol. 36, pp. 85–121, 1985.
[15] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* New York: Wiley, 1973.
[16] S. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," Carnegie Mellon University, School of Computer Science, Tech. Rep. CMU-CS-90-100, Feb. 1990.
[17] R. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics,* vol. 7, part II, pp. 179–188, 1936.
[18] S. Fu and P. Swain, "On syntactic pattern recognition," in *Software Engineering,* vol. 2, J. Tou, Ed. New York: Academic Press, 1971, pp. 155–182.
[19] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* New York: Academic Press, 1972.
[20] G. Granath, "Application of fuzzy clustering and classification to evaluate provenance of glacial till," *J. Math. Geology,* vol. 16, pp. 283–301, 1984.
[21] S. Grossberg, "How does a brain build a cognitive code?," *Pshychol. Rev.,* vol. 87, pp. 1–51, 1980.
[22] Y-C. Ho and R. Kashyap, "An algorithm for linear inequalities and its applications," *IEEE Trans. Electron. Comput.,* vol. EC-14, pp. 683–688, 1965.
[23] T. Huntsberger and P. Ajjimarangesee, "Parallel self-organizing feature maps for unsupervised pattern recognition," *Int. J. Intelligent Systems,* vol. 16, pp. 357–372, 1990.
[24] A. Jozwik, "A learning scheme for a fuzzy k-NN rule," *Pattern Recog. Lett.,* vol. 1, pp. 287–289, 1983.
[25] A. Kandel, *Fuzzy Mathematical Techniques with Applications.* Reading, MA: Addison-Wesley, 1986.
[26] J. Keller, M. Gray, and J. Givens, "A fuzzy k-nearest neighbor algorithm," *IEEE Trans. Syst., Man, Cybern.,* vol. SMC-15, pp. 580–585, 1985.
[27] J. Keller and D. Hunt, "Incorporating fuzzy membership functions into the perceptron algorithm," *IEEE Trans. Patt. Anal. Mach. Intell.,* vol. 7, pp. 693–699, 1985.
[28] R. Kennedy, "The hyperspherical attractor neural network: Concepts, theory, and application," in *Proc. WNN-AIND 91,* 1991, pp. 241–271.
[29] T. Kohonen, *Self-Organization and Associative Memory.* Berlin: Springer-Verlag, 1984.
[30] S. Lee and E. Lee, "Fuzzy neural networks," *Math. Biosciences,* vol. 23, pp. 151–177, 1975.
[31] M. Minsky and S. Paper, *Perceptropns: An Introduction to Computational Geometry.* Cambridge, MA: MIT Press, 1969.
[32] S. Mitra and S. Pal, "Self-organizing neural network as a fuzzy classifier," *IEEE Trans. Syst., Man, Cybern.,* to be published.
[33] B. Moore, "ART 1 and pattern clustering," in *Proc. 1988 Connectionist Summer School,* pp. 174–185.
[34] D. Reilly, L. Cooper, and C. Elbaum, "A neural model for category learning," *Biol. Cybern.,* vol. 45, pp. 35–41, 1982.
[35] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms.* Washington, DC: Spartan Books, 1962.
[36] D. Rumelhart and J. McClelland, Eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* Cambridge, MA: MIT Press, 1986.
[37] E. Ruspini, "A new approach to clustering," *Inform. and Control,* vol. 15, pp. 22–32, 1969.
[38] S. Salzberg, "A nearest hyperrectangle learning method," *Machine Learning,* vol. 6, pp. 251–276.
[39] P. Simpson, "Fuzzy adaptive resonance theory," presented at the Southern Illinois University Neuroengineering Workshop, Sept. 6–7, 1990, Carbondale, IL. Also in *Proc. GD AI 90* (Ft. Worth, TX), Nov. 7–8, 1990, and published as General Dynamics Tech. Rep. GDE-ISG-PKS-11.
[40] P. Simpson, *Artificial Neural Systems: Foundations, Paradigms, Applications and Implementations.* Elmsford, NY: Pergamon Press, 1990.
[41] P. Simpson, "Fuzzy min–max classification with neural networks," *Heuristics,* vol. 4, no. 1, pp. 1–9, 1991.
[42] P. Simpson, "Fuzzy min–max neural networks," in *Proc. 1991 Int. Joint Conf. Neural Networks* (Singapore).
[43] P. Simpson, "Fuzzy min–max neural networks—Part 2: Clustering, submitted to *IEEE Trans. Fuzzy Syst.*
[44] D. Specht, "Probabilistic neural networks," *Neural Networks,* vol. 3, no. 1, pp. 109–118, 1990.
[45] R. Streit and T. Luginbuhl, "Maximum likelihood training of probabilistic neural networks," *IEEE Trans. Neural Networks,* in press.
[46] 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.
[47] P. Werbos, "Beyond regression: New tools for prediction and analysis in the behavioral sciences," Ph.D. thesis, Harvard University, Cambridge, MA, 1974.
[48] P. Werbos, "Neurocontrol and fuzzy logic—Connections and designs," *Int. J. Approximate Reasoning,* to be published.
[49] B. Widrow and M. Hoff, "Adaptive switching circuits," in *IRE Conv. Rec.,* part IV, 1960, pp. 96–104.
[50] R. Yager, "A measurement informational discussion on fuzzy union and intersection," *Int. J. Man–Machine Studies,* vol. 11, pp. 189–200, 1979.
[51] L. Zadeh, "Fuzzy sets," *Inform. and Control,* vol. 8, pp. 338–353, 1965.