Fuzzy Lattice Neural Network (FLNN) : A Hybrid Model for Learning[*]

Vassilios Petridis and Vassilis G. Kaburlasos[†]

Abstract

This paper proposes two hierarchical schemes for learning, one for clustering and the other for classification problems. Both schemes can be implemented on a fuzzy lattice neural network (FLNN) architecture, to be introduced herein. The corresponding two learning models draw on adaptive resonance theory (ART) and min-max neurocomputing principles but their application domain is a mathematical lattice. Therefore they can handle more general types of data in addition to N-dimensional vectors. The FLNN neural model stems from a cross-fertilization of lattice theory and fuzzy set theory. Hence a novel theoretical foundation is introduced in this paper, that is the *framework of fuzzy lattices* or *FL-framework*, based on the concepts *fuzzy lattice* and *inclusion measure*. Sufficient conditions for the existence of an inclusion measure in a mathematical lattice are shown. The performance of the two FLNN schemes, that is for clustering and for classification, compares quite well with other methods and it is demonstrated by examples on various data sets including several benchmark data sets.

Index Terms - ART neural networks, clustering methods, decision support systems, fuzzy lattice theory, fuzzy neural networks, learning systems, pattern classification, pattern recognition.

1 INTRODUCTION

Artificial neural networks are a technology rooted in many disciplines. Numerous popular models have been proposed to-date inspired from physics, chemistry, geometry, statistics, biology, neurobiology, psychology [1], [2], [4], [11], [15], [21], [27], [32] to name but a few. The artificial neural networks are endowed with unique attributes such as universal input-output mapping, the ability to learn from and adapt to their environment [14]. A most prominent point of vantage is their ability for massive parallel processing.

Connectionist schemes have been proposed to process information using representations other than the numerical one. For instance there have been proposed neural networks for extracting symbolic knowledge [28], for utilizing expert knowledge represented by fuzzy if-then rules [16], [33]. Cross-fertilization of inferencing networks with conventional neural networks such as the Kohonen's model have been reported [25], and wavelet networks were also considered [8].

Nevertheless the applicability domain of all previous neural schemes is more or less restricted. More specifically the data being processed are frequently in the Euclidean space or they are treated as such. For example the problem of grammatical inference is treated in [28] by processing strings of "0's" and "1's" of positive and negative examples as real numbers. For the binary ART neural network [4] inputs of features are represented by strings of "0's" and "1's" and are treated as real numbers. Moreover even in [16], where human knowledge and numerical data are integrated by propagating fuzzy numbers through a neural network, other types of data can not be treated in addition.

In an effort to define connectionist schemes able to cope with a wide gamut of disparate data we came to grips with mathematical lattices. The work herein reports on both theoretical and practical results in this direction. Our contribution can be summarized as follows. We propose a sound mathematical ground, that is

[*] This work was supported in part by Brite Euram project BE7470

[†] The authors are with the Department of Electrical and Computer Engineering at Aristotle University of Thessaloniki, GR-54006 Thessaloniki, Greece. Email: petridis@vergina.eng.auth.gr, vgkabs@egnatia.ee.auth.gr

the *fuzzy lattice framework* or *FL-framework*. Based on the FL-framework we show a connectionist architecture, namely fuzzy lattice neural network (FLNN) [20], [29], which can treat with mathematical consistency and jointly such disparate data as conventional R^N vectors, fuzzy sets, symbols, propositions, etc. Next, two learning schemes implementable by the FLNN are shown, one scheme for clustering and the other scheme for classification. These two FLNN schemes are applied quite successfully on several benchmark data sets for learning and recognition.

Employment of mathematical lattices is not new in engineering. They appear either explicitly or implicitly in many instances. For example lattices are employed explicitly in [6] for controlling discrete event systems. Complete lattices are employed in [23] in deductive databases. In [35] and [12] it is explained how a lattice may generalize the notion of a fuzzy set while in [35], in particular, it is noted that "fuzzy sets (over a universe of discourse) constitute a distributive lattice with 0 and 1". In neurocomputing, lattices appear implicitly in the fuzzy ART [5] and in the min-max neural nets [30], [31] since both types of neural networks deal with hyperboxes in the conventional set R^N; note that the set of hyperboxes is a lattice. Regarding neurocomputing in lattices the oldest attempt known to the authors is noted in [19], where the work is oriented more toward medical applications than toward theoretical substantiation.

According to the neurocomputing approach to learning we have taken, sets of lattice elements are specified by sets of finitely many overlapping and/or non-overlapping lattice intervals. In this sense the FLNN approach to learning is similar to the one taken by the biologically motivated adaptive resonance theory (ART) [5] and the min-max neural networks [30], [31]. Nevertheless, even though it is originated in the adaptive resonance as well as the min-max neural networks the FLNN proceeds far beyond. It generalizes both of them by employing the theory of lattices [3] which is cross-fertilized with the theory of fuzzy sets [35] as explained throughout this paper. Note that the work herein is not merely a combination of lattice theory and fuzzy set theory but instead it is a fruitful cross-fertilization of the two theories.

The layout of this paper is as follows. Section 2 outlines a novel theoretical foundation for learning which employs mathematical lattices. Section 3 introduces the FLNN architecture as well as two learning schemes, one for clustering and the other for classification. Section 4 shows sufficient conditions for applying an FLNN scheme on lattices of intervals. Section 5 reports on the capacity of the FLNN for pattern recognition of various data sets, including five benchmark data sets. Finally section 6 summarizes the principal achievements.

2 A NEW THEORETICAL FOUNDATION FOR LEARNING

The goal of this section is to introduce a new a viable theoretical framework for hybrid learning with a wide application domain.

2.1 Fuzzy lattices

Recall that a lattice is a partly ordered set any two of whose elements have a greatest lower bound or *meet* denoted by $x \wedge y$ and a least upper bound or *join* denoted by $x \vee y$. A lattice $\mathbb{L}$ is *complete* when each of its subsets has a least upper bound and a greatest lower bound in $\mathbb{L}$. A non-void complete lattice contains a least and a greatest element denoted respectively by O and I [3].

Let $\mathbb{L}$ be a lattice, either complete or not. Consider the relation R of the induced partial ordering in $\mathbb{L}$, and let $x,y \in \mathbb{L}$. Then either x and y are *comparable*, that is $(x,y) \in R$ or $(y,x) \in R$, or x and y are *incomparable*, that is none of the previous ordered pairs belongs to R. The novel notion *fuzzy lattice* is introduced in order to extend the crisp, if existent, lattice relation of partial ordering to any pair (x,y) of the set $S=\{(x,y)\text{: } x,y \in \mathbb{L}\}$. The practical significance for such an extension is that it may associate any two lattice elements. Hence to every $(x,y) \in S$ a real number $\mu_P(x,y) \in [0,1]$ is attached to indicate the degree of inclusion of x in y. Formally an extension (relation) P can be defined [36] by

$$P = \{((x,y),\mu_P(x,y)) : x,y\in \mathbb{L},\ \mu_P(x,y)\in [0,1]\},$$ where

the real function $\mu_P(x,y)$ is a fuzzy membership function on the universe of discourse $S=\{(x,y): x,y\in \mathbb{L}\}$.

To keep the extension P compatible with the original lattice's $\mathbb{L}$ partial ordering relation R, P is defined under the constraint that $\mu_P(x,y)=1$ if and only if $x\leq y$ in $\mathbb{L}$. The definition of a fuzzy lattice ensues.

Definition 1

A *fuzzy lattice* is a pair $(\mathbb{L}, \mu_P(x,y))$, where $\mathbb{L}$ is a conventional lattice and μ_P: $S\rightarrow[0,1]$ is a fuzzy membership function on the universe of discourse $S=\{(x,y): x,y\in \mathbb{L}\}$. It is $\mu_P(x,y)=1$ if and only if $x\leq y$ in $\mathbb{L}$.

□

We remark that real function $\mu_P(x,y)$ can be interpreted as specifying the degree of inclusion of x in y. We will refer to the collection of all fuzzy lattices by the term *FL-framework*. Learning in the FL-framework is achieved by specifying sets of lattice elements by intervals. To this end rendering "fuzzy" a conventional lattice is convenient as the latter enables association of lattice elements to one another, even when those lattice elements are incomparable. Note that in this work we deal solely with complete lattices.

2.2 Inclusion measure

A fuzzy lattice can ensue from a conventional one by an inclusion measure defined next.

Definition 2

Let $\mathbb{L}$ be a complete lattice with least and greatest elements O and I respectively. An *inclusion measure* is a mapping σ: $S=\{(x,y): x,y\in \mathbb{L}\}\rightarrow[0,1]$ such that $\sigma((x,u))\equiv\sigma(x\leq u)$ satisfies the following three conditions

(C1) $\sigma(u\leq O) = 0$, $u\neq O$. (C2) $\sigma(u\leq u) = 1$, $\forall u\in \mathbb{L}$.

(C3) $u\leq w \Rightarrow \sigma(x\leq u) \leq \sigma(x\leq w)$, $x,u,w\in \mathbb{L}$ - Consistency Property

□

Conditions (C2) and (C3) jointly imply $u\leq w \Rightarrow \sigma(u\leq u) \leq \sigma(u\leq w) \Rightarrow \sigma(u\leq w)=1$, $u,w\in \mathbb{L}$. The latter furthermore implies $\sigma(u\leq I)=1$, where I is the greatest element in $\mathbb{L}$. It can be argued that $\sigma(x\leq u)$ shows the degree of truth of the lattice inclusion relation $x\leq u$, that is the degree of inclusion of x in u.

In order an inclusion measure to define of a fuzzy lattice out of a conventional one it suffices to be $\sigma(u\leq w)=1 \Rightarrow u\leq w$, according to Definition 1. To this end we considered a real-valued *function-h* defined below.

Definition 3

A *function-h*, h: $\mathbb{L}\rightarrow$R on a complete lattice $\mathbb{L}$ satisfies the following three properties:

(P1) $h(O)= 0$, where O is the least element in $\mathbb{L}$, (P2) $u<w \Rightarrow h(u)<h(w)$, $u,w\in \mathbb{L}$, and

(P3) $u\leq w \Rightarrow h(x\vee w)-h(x\vee u)\leq h(w)-h(u)$, $x,u,w\in \mathbb{L}$.

□

We remark that if $h(O)\neq 0$, then a function $h_0(.)$ with $h_0(O)=0$ can be defined out of h(.) by subtracting h(O) from all h(x), $x\in \mathbb{L}$. A function-h does not exist necessarily in a lattice. Nevertheless when it exists it can be shown that $k(x,y) = \dfrac{h(y)}{h(x\vee y)}$, $x,y\in \mathbb{L}$ where h(.) is a function-h, defines an inclusion measure in $\mathbb{L}$.

Theorem 1

Let $\mathbb{L}$ be a complete lattice. Then the existence of a function-h on $\mathbb{L}$ is sufficient for the function $k(x \le u) = \dfrac{h(u)}{h(x \vee u)}$ to be an inclusion measure in $\mathbb{L}$, that is $k(x \le u) = \sigma(x \le u) = \dfrac{h(u)}{h(x \vee u)}$.

□

The proof of theorem 1 is given in the Appendix.

Note that $k(x \le u) = \dfrac{h(u)}{h(x \vee u)} = 1 \Leftrightarrow h(u) = h(x \vee u) \Leftrightarrow u = x \vee u \Leftrightarrow x \le u$. Hence $(\mathbb{L}, k(x \le u))$ is a fuzzy lattice. Function $k(x \le y)$ will be employed as an activation function by the neurons of an FLNN neural model.

Example-2.1

Consider the lattice $\mathbb{I}=[0,1]$, that is the closed interval of real numbers between 0.0 and 1.0. This lattice is called *chain* [3] because it is characterized by a *total* instead of a partial ordering. It is clear that $\mathbb{I}=[0,1]$ is a complete lattice with least and greatest elements 0.0 and 1.0 respectively. The corresponding lattice meet ($\wedge$) and join ($\vee$) operations are given respectively by $x \wedge y = \min\{x,y\}$ and $x \vee y = \max\{x,y\}$, $x,y \in \mathbb{I}=[0,1]$.

Any monotone increasing real function h(.) with h(0.0)=0 is a function-h because properties (P1) and (P2) of Definition 3 are obviously true. Property (P3) can be shown by considering the three cases: 1) $x \le u \le w$, 2) $u \le x \le w$, and 3) $u \le w \le x$. Case 1) implies $x \le u \le w \Rightarrow h(x \vee w) - h(x \vee u) = h(w) - h(u)$, case 2) implies $u \le x \le w \Rightarrow h(x \vee w) - h(x \vee u) = h(w) - h(x) \le h(w) - h(u)$, and case 3) implies $u \le w \le x \Rightarrow h(x \vee w) - h(x \vee u) = 0 \le h(w) - h(u)$. Hence (P3) holds.

For all examples herein drawn from the Euclidean space the function-h: $h(x)=x$ was employed.

□

2.3 A hierarchy of fuzzy lattices

It holds that the product of N lattices $\mathbb{L}_1, \ldots, \mathbb{L}_N$, that is $\mathbb{L} = \mathbb{L}_1 \times \ldots \times \mathbb{L}_N$, is lattice with an induced partial ordering relation defined by $(x_1, \ldots, x_N) \le (y_1, \ldots, y_N) \Leftrightarrow x_1 \le y_1, \ldots, x_N \le y_N$ [3]. This partial ordering relation is called herein *relation r_1*. Lattice $\mathbb{L} = \mathbb{L}_1 \times \ldots \times \mathbb{L}_N$ is called *product lattice*, and each one of the $\mathbb{L}_i$, $i \in \{1, \ldots, N\}$ is called *constituent lattice*. The product lattice meet ($\wedge$) and join ($\vee$) are defined respectively by $x \wedge y = (x_1, \ldots, x_N) \wedge (y_1, \ldots, y_N) = (x_1 \wedge y_1, \ldots, x_N \wedge y_N)$ and $x \vee y = (x_1, \ldots, x_N) \vee (y_1, \ldots, y_N) = (x_1 \vee y_1, \ldots, x_N \vee y_N)$ [3]. Note that if the constituent lattices $\mathbb{L}_i$, $i \in \{1, \ldots, N\}$ are all complete with least and greatest elements $O_1, \ldots, O_N$ and $I_1, \ldots, I_N$ respectively, then the product lattice $\mathbb{L} = \mathbb{L}_1 \times \ldots \times \mathbb{L}_N$ will be a complete lattice with least element $O = (O_1, \ldots, O_N)$ and greatest element $I = (I_1, \ldots, I_N)$. Regarding functions-h the following statement holds.

Lemma 1

Let $\mathbb{L} = \mathbb{L}_1 \times \ldots \times \mathbb{L}_N$ be the product of N complete constituent lattices with function(s)-h : $h_1(.), \ldots, h_N(.)$ respectively. Then $h(x_1, \ldots, x_N) = h_1(x_1) + \ldots + h_N(x_N)$ defines a function-h on the product lattice $\mathbb{L} = \mathbb{L}_1 \times \ldots \times \mathbb{L}_N$.

□

The proof of Lemma 1 is given in the Appendix.

Remark that Lemma 1 substantiates FL-framework's claim for a "disparate data fusion capacity", suffices the data be elements of complete constituent lattices where a function-h is available on each constituent lattice. Lemma 1 also partly substantiates the claim for "an hierarchical nature" of the FLNN. An additional conceptual level in the hierarchy of lattices is incurred by considering the set of intervals of lattice elements as shown in the sequel. The set in question augmented by a least element is a complete lattice, denoted herein by $\mathbb{V}_L$. The following definition suggests a "measure of the magnitude" of an interval in $\mathbb{V}_L$ with respect to a function-h defined on $\mathbb{L}$.

Definition 4

Let $\mathbb{L}$ be a complete lattice with O and I its least and greatest elements, respectively, and let h(.) be a function-h on $\mathbb{L}$. Then the *size* of $[a,b]\in\mathbb{V}_L$ with respect to h(.) is defined by Z([a,b])=h(b)-h(a).

□

At this point it is necessary to cite a convention regarding the notation employed. Specifically let $\mathbb{L}$ denote a product lattice, and let $\mathbb{L}_1,\ldots,\mathbb{L}_N$, denote its constituent lattices. We agree to denote an element of the product lattice $\mathbb{L}$ by a letter without any subscript, for instance $x\in\mathbb{L}$. On the other hand, an element of a constituent lattice $\mathbb{L}_i$ is denoted by a letter with a subscript, for instance $x_i\in\mathbb{L}_i$. Same subscripts imply elements of the same constituent lattice and vice-versa.

Example-2.2

Consider the product lattice $\mathbb{U}=[0,1]\times[0,1]$, that is the unit-square. In this case there exist two identical constituent lattice-chains $\mathbb{I}=[0,1]$. By Lemma 1 we infer that function h(x)=x on lattice-chain $\mathbb{I}=[0,1]$ implies function-h: $h(x_1,x_2)= h(x_1)+h(x_2)=x_1+x_2$ in the product lattice $\mathbb{U}=[0,1]\times[0,1]$.

In lattice-chain $\mathbb{I}=[0,1]$ the size of an interval $[x_1,y_1],x_1,y_1\in[0,1]$ with $x_1\leq y_1$ is defined by $Z([x_1,y_1])=h(y_1)-h(x_1)= y_1-x_1$. On the other hand an interval in the product lattice $\mathbb{U}=[0,1]\times[0,1]$ corresponds to a rectangle in the unit-square and its size equals half the circumference of the corresponding rectangle.

□

2.4 Lattices $\mathbb{PL}_1$ and $\mathbb{PL}_2$

Let $\mathbb{L}$ be a complete product lattice and let O and I denote, respectively, the least and greatest element of $\mathbb{L}$. In line with last section’s analysis it can be inferred that $\mathbb{L}\times\mathbb{L}$ implies a complete lattice, which will be denoted by $\mathbb{PL}_1$. Lattice $\mathbb{PL}_1$ is defined on the set $S=\{(a,b): a,b\in\mathbb{L}\}$ and the relation of its partial ordering is the same *relation* r_1, as before, implied by product lattice $\mathbb{L}=\mathbb{L}_1\times\ldots\times\mathbb{L}_N$. That is $(a,b)\leq(c,d)$ if and only if $a\leq c$ and $b\leq d$. The join and the meet in $\mathbb{PL}_1$ are defined by

$$(a,b)\vee(c,d) = (a\vee c,b\vee d), \text{ and } (a,b)\wedge(c,d) = (a\wedge c,b\wedge d), \text{ where } a,b,c,d\in\mathbb{L}$$

The least element of the complete lattice $\mathbb{PL}_1$ is (O,O) whereas its greatest element is (I,I).

By virtue of Lemma 1 it can be shown that if h(.) is a function-h on a lattice $\mathbb{L}$, then H((a,b))= h(a)+h(b) defines a function-h on $\mathbb{PL}_1$.

In the sequel another lattice is defined on the set $S=\{(a,b): a,b\in\mathbb{L}\}$, that is lattice $\mathbb{PL}_2$ of *generalized intervals* in $\mathbb{L}$. But lattice $\mathbb{PL}_2$ is defined with a different partial ordering. Note that the set $S=\{(a,b): a,b\in\mathbb{L}\}$ will also be denoted by $S=\{[a,b]: a,b\in\mathbb{L}\}$.

Theorem 2

A complete lattice $\mathbb{PL}_2$ can be defined on the set $S=\{[a,b]: a,b\in\mathbb{L}\}$ where $\mathbb{L}$ is a complete lattice, with two binary operations between elements of $\mathbb{PL}_2$ defined as follows :

$$[a,b]\wedge[c,d]= [a\vee c,b\wedge d], \text{ and } [a,b]\vee[c,d]= [a\wedge c,b\vee d], \text{ where } a,b,c,d\in\mathbb{L}$$

because the following laws L1-L4 are satisfied, for $a,b,c,d,e,f\in\mathbb{L}$ [3].

L1.	$[a,b]\wedge[a,b] = [a,b]$, and $[a,b]\vee[a,b] = [a,b]$	(Idempotent)
L2.	$[a,b]\wedge[c,d] = [c,d]\wedge[a,b]$, and $[a,b]\vee[c,d] = [c,d]\vee[a,b]$	(Commutative)
L3.	$[a,b]\wedge([c,d]\wedge[e,f])=([a,b]\wedge[c,d])\wedge[e,f]$, and $[a,b]\vee([c,d]\vee[e,f])=([a,b]\vee[c,d])\vee[e,f]$	(Associative)
L4.	$[a,b]\wedge([a,b]\vee[c,d]) = [a,b]\vee([a,b]\wedge[c,d]) = [a,b]$	(Absorption)

□

The proof of theorem 2 is given in the Appendix.

The implied relation $[a,b] \leq [c,d]$ of partial ordering in $\mathbb{PL}_2$ is equivalent to $[a,b]\wedge[c,d] = [a,b] \Leftrightarrow a\vee c=a,\ b\wedge d=b \Leftrightarrow c\leq a,\ b\leq d$, and it will be called (partial ordering) *relation r_2*. Note that $[I,O]$ is the least element and $[O,I]$ is the greatest element in the complete lattice $\mathbb{PL}_2$.

An element of $\mathbb{PL}_2$ is called *generalized interval* because of the way lattice $\mathbb{PL}_2$ was defined. Note that lattice $\mathbb{V}_L$ of intervals is a sublattice of $\mathbb{PL}_2$. Recall that “a sublattice of a lattice $\mathbb{L}$ is a subset $\mathbb{X}$ of $\mathbb{L}$ such that $a,b\in\mathbb{X}$ imply $a\wedge b\in\mathbb{X}$ and $a\vee b\in\mathbb{X}$” [3]. As the least element in $\mathbb{V}_L$ we may take the least element of $\mathbb{PL}_2$, that is the generalized interval $[I,O]$. Furthermore note that parentheses () are used to embrace an element of $\mathbb{PL}_1$, for instance $(a,b)\in\mathbb{PL}_1$, whereas brackets [] are embracing elements of $\mathbb{PL}_2$, for instance $[c,d]\in\mathbb{PL}_2$, intentionally, to underscore the two different lattice orderings on the same set $\mathbb{S}$.

Example-2.3

Regarding the unit-square product lattice $\mathbb{U}=[0,1]\times[0,1]$ an element of $\mathbb{PU}_1$ or $\mathbb{PU}_2$ is given by a vector of 4 numbers. These vectors, as elements of the two distinct lattices $\mathbb{PU}_1$ and $\mathbb{PU}_2$, differ as to the manner in which their meet ($\wedge$) and the join ($\vee$) are calculated. For instance consider the elements (0.1,0.7,0.3,0.4) and (0.5,0.8,0.5,0.2) of $\mathbb{PU}_1$. Their meet and join are, respectively

$(0.1,0.7,0.3,0.4)\wedge(0.5,0.8,0.5,0.2)=(0.1\wedge0.5,0.7\wedge0.8,0.3\wedge0.5,0.4\wedge0.2)=(0.1,0.7,0.3,0.2)$, and

$(0.1,0.7,0.3,0.4)\vee(0.5,0.8,0.5,0.2)=(0.1\vee0.5,0.7\vee0.8,0.3\vee0.5,0.4\vee0.2)=(0.5,0.8,0.5,0.4)$.

The same vectors of numbers considered as elements of $\mathbb{PU}_2$, are combined as follows:

$[0.1,0.7,0.3,0.4]\wedge[0.5,0.8,0.5,0.2]=[0.1\vee0.5,0.7\wedge0.8,0.3\vee0.5,0.4\wedge0.2]=[0.5,0.7,0.5,0.2]$, and

$[0.1,0.7,0.3,0.4]\vee[0.5,0.8,0.5,0.2]=[0.1\wedge0.5,0.7\vee0.8,0.3\wedge0.5,0.4\vee0.2]=[0.1,0.8,0.3,0.4]$ where it was assumed that [0.1,0.7], [0.3,0.4], [0.5,0.8], and [0.5,0.2] specify generalized intervals in the constituent lattices. Note that a single point $(x_1,y_1)\in\mathbb{U}$ can be represented in $\mathbb{PU}_2$ by the trivial interval $[x_1,x_1,y_1,y_1]$.

□

3 FUZZY LATTICE NEURAL NETWORK (FLNN)

The goal in this section is to employ the FL-framework presented in the previous section in order to introduce 1) a neural architecture, that is the fuzzy lattice neural network (FLNN), and 2) two learning schemes, one for clustering and the other for classification problems which (schemes) can be applied by the FLNN. Some more notions need to be introduced in the following subsection.

3.1 Families of lattice intervals

The elements of the set $\mathbb{V}_L$, by being intervals of lattice elements, define sets of “adjacent” lattice $\mathbb{L}$ elements. The FLNN deals with certain families of lattice intervals denoted by $\{w_i\}$, where $w_i\in\mathbb{V}_L$ and i is in a finite index set. The FLNN aims at identifying sets c_k of lattice elements, namely *classes*, which can be represented by the set-union of a finite number of $\mathbb{V}_L$ intervals, that is $c_k=\bigcup_i w_{k,i}$. Note that more than one families of intervals may specify the same class c_k.

Instrumental to the applicability of the FLNN in lattice $\mathbb{V}_L$ is the existence of an inclusion measure in $\mathbb{V}_L$. Throughout this section we assume the existence of an inclusion measure in $\mathbb{V}_L$. In the next section we show sufficient conditions for its existence. The degree of inclusion of an interval $x\in\mathbb{V}_L$ in a class c_k is defined as follows.

Definition 5

If $x\in\mathbb{V}_L$ and $c_k=\bigcup_i w_{k,i}$ then the *degree of inclusion* of x in class c_k is defined as $\sigma(x\leq c_k):=\max_i \sigma(x\leq w_{k,i})$.

□

Remark that sometimes $\sigma(x \le c_k) := \sigma(x \le \bigcup_i w_{k,i})$ will be denoted instead by $\sigma(x \le \{w_{k,i}\})$. Moreover, when it happens to be $\sigma(x \le c_k)=1$ we will write $x \le c_k$ and say : "x is *in(side)* class c_k". The definition of *connected* classes will be useful.

Definition 6

A class $c=\bigcup_i w_i$ is called *connected* if and only if for any two intervals p,q inside c there exists a sequence of intervals $t_0,\ldots,t_{N-1}$ inside class c, from p to q. That is $t_0 \wedge p=p$, $t_{N-1} \wedge q=q$, and $t_i \wedge t_{i+1} \neq O$, $i=0,\ldots,N-2$.

□

Note that a family is called *connected* if and only if it defines a connected class.

Decision making in the fuzzy lattice neural network (FLNN) is driven by the degree of true of proposition " $x \le c_k$" as expressed by $\sigma(x \le c_k)$, where x is an input lattice interval that excites the system and the c_k's are learned classes kept in the system's memory.

Let F be the collection of all families that can represent a concrete class c in lattice $\mathbb{L}$. Note that the cardinality of a family $\{w_i\}$ of intervals used to represent class c need not be constant. We say that a family $\{p_m\}$ in F is "smaller than or equal to" another family $\{q_n\}$ in F symbolically $\{p_m\} \le \{q_n\}$, if and only if $\forall$p in $\{p_m\}$ there exists a q in $\{q_n\}$ such that $p \le q$. Moreover, a family $\{p_m\}$ is "strictly smaller" than another family $\{q_n\}$ and we write $\{p_m\} < \{q_n\}$ if and only if it is $\{p_m\} \le \{q_n\}$ and either there is a p in $\{p_m\}$ and a q in $\{q_n\}$ such that $p<q$ or the cardinality of family $\{p_m\}$ is strictly smaller than the cardinality of family $\{q_n\}$. In conclusion F is a partially ordered set. Consider the following lemma.

Lemma 2

The collection F of families representing a concrete class $c=\bigcup_i w_i$ has a maximum element, namely *quotient* of class c, and denoted by $Q(F) = Q(\{w_i\})$.

□

The proof of Lemma 2 is given in the Appendix.

A quotient $Q(F)$ is the maximum element in the set F. In fact, it is easy to show that the set F is a lattice, and it constitutes another conceptual "pinnacle" in the hierarchy of lattices implied by the FL-framework. There is a concrete benefit in replacing a family $\{w_i\}$ which represents a class by the quotient $Q(\{w_i\})=Q(F)$. That is for any input interval x it holds $\sigma(x \le Q(F)) = \max \sigma(x \le \{w_j\})$, where $\{w_j\} \in F$. In other words the quotient $Q(F)$ maximizes the degree $\sigma(x \le c)$ of inclusion of x in the class $c=\bigcup_i w_i$.

Example-3.1

The utility of the technique of maximal expansions is illustrated in Fig. 1. Note that the complete lattice in this case is the unit-square defined as the product $\mathbb{U}=[0,1]\times[0,1]$ of the two lattices-chains $\mathbb{I}=[0,1]$. Recall that an interval in lattice unit-square $\mathbb{U}$ is a rectangle. Consider the class $c=\{w_1,w_2\}$ and let the rectangle x consist solely of points of w_1 and w_2 (Fig.1(a)). Then it is reasonable to expect $\sigma(x \le c)=1$. But this is not the case in Fig. 1(a) because both $w_1 < x \vee w_1$ and $w_2 < x \vee w_2$ hold, therefore it is $\sigma(x \le c) = k(x \le c) < 1$, according to Theorem 1. The technique of maximal expansions comes to restore the expected equality relation by replacing class $c=\{w_1,w_2\}$ by its quotient, that is the family $\{w_1,w'_2\}$ of the maximal rectangles of c (Fig.1(b)). Hence any rectangle containing solely points of the class c is contained in at least one of the quotient members w_1, w'_2, and equality $\sigma(x \le c)=1$ is guaranteed.

□

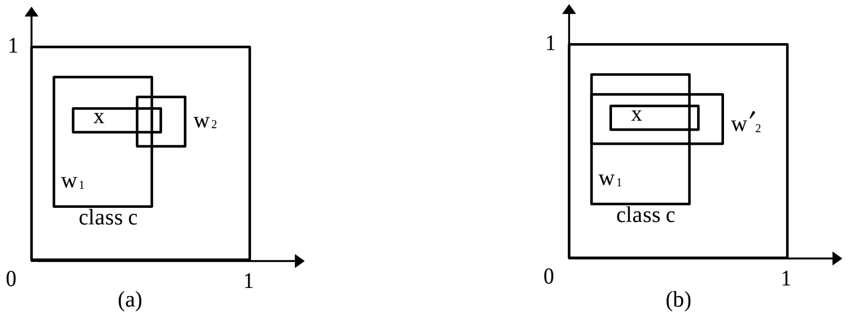

Figure 1 (a) Despite the fact that rectangle x is inside class $c=w_1 \cup w_2$ it is neither $x \leq w_1$ nor $x \leq w_2$, therefore it follows $\sigma(x \leq c)<1$.

(b) The technique of maximal expansions finds the quotient $\{w_1, w'_2\}$ of class c and it guarantees $\sigma(x \leq c)=1$ when x is inside class $c=w_1 \cup w_2 = w_1 \cup w'_2$.

3.2 The FLNN architecture

Fig. 2 shows the basic FLNN architecture which can be employed for learning and decision making in the FL-framework including the conventional set R^N. Notice the resemblance to the two layer ART neural network [4] which was introduced in the mid-1970s to solve some problems in sensory coding. The many variations of the ART have modeled a variety of neurobiological and behavioral data and they were also applied in a range of engineering applications [7], [13]. Nevertheless ART is limited to the set R^N

Like ART, the FLNN proposes a modifiable recurrent architecture for clustering in two layers. One layer, called *category layer*, is cognitively "higher" than the other called the *input layer*. Single nodes at the category layer encode patterns of node activities from the input layer. On the one hand the category layer consists of L artificial neurons specifying L intervals that define M classes (Fig.2); it is $L \geq M$. On the other hand the input layer consists of N artificial neurons used for buffering and matching. The two layers are fully interconnected by lattice-weighted bidirectional links which may filter -up or -down the activities of the corresponding layer. A "reset" node (Fig.2) operates much the same way as the orienting subsystem of the ART does [4].

A key difference between FLNN and ART, besides FLNN's own activation function $\sigma(x \leq w)$, is the applicability of FLNN on fuzzy lattices which could imply significant technological benefits stemming from a much wider application domain than the conventional Euclidean space. Another vantage point of the FLNN compared to the ART is that the inputs $I_1, I_2, \ldots, I_N$ to the FLNN (Fig.2) are intervals, that is hyperboxes in the set R^N in particular. Hence we may compensate for the uncertainty of the measurements by feeding to the FLNN a neighborhood of values as defined by an interval rather than feeding a single point. A final advantage of the FLNN is the technique of *maximal expansions* which enhances overlapping of two intervals by enlarging their lattice meet to the maximum and in all possible ways. The basic FLNN architecture of Fig.2 can be employed for learning by either clustering or classification.

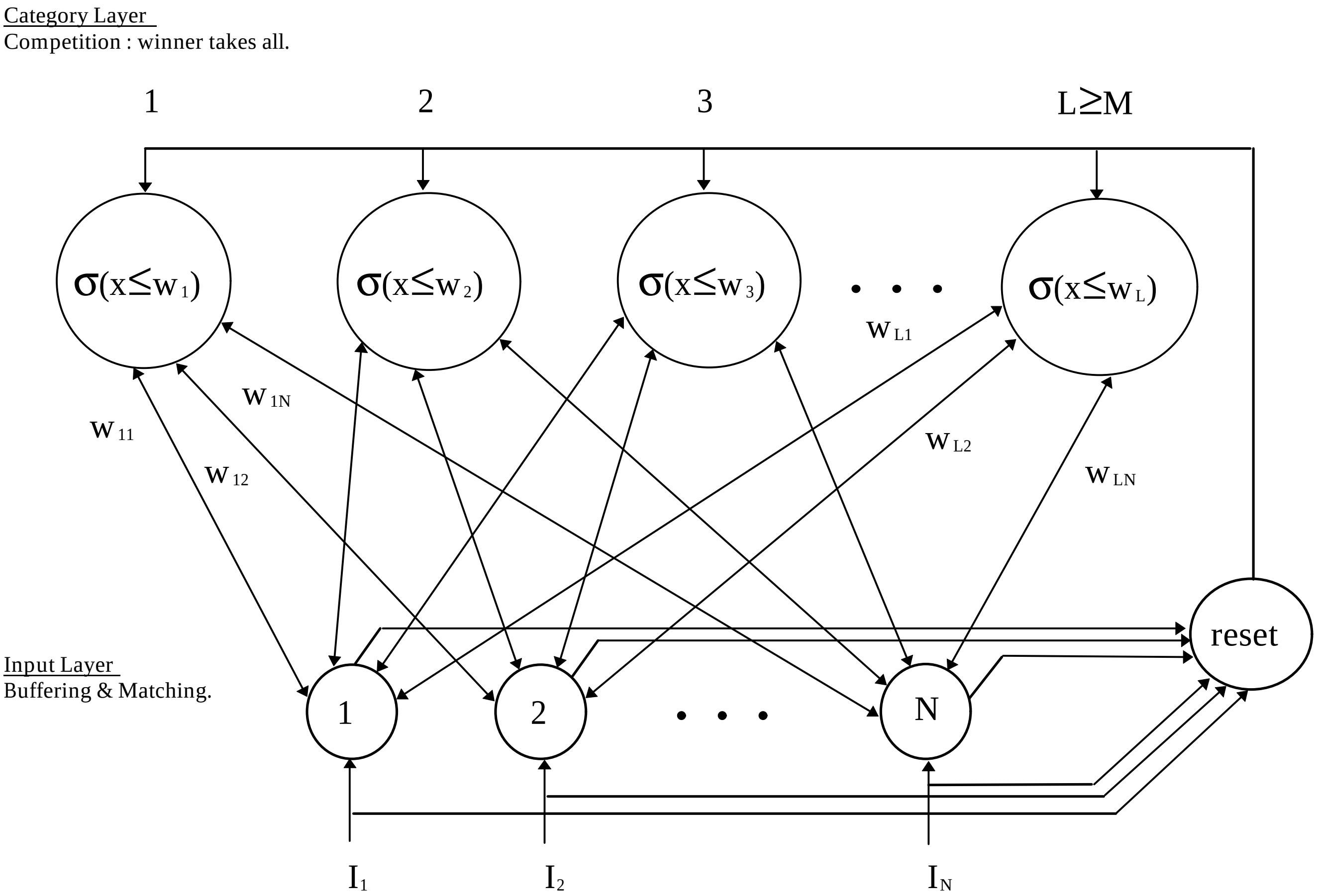

Figure 2 The basic two layer FLNN architecture for learning in the lattice $\mathbb{V}_L$ of intervals. L is the number of category layer neurons which equals the total number of intervals used to define M classes. The category layer neurons employ the lattice inclusion measure σ(.) as their activation functions. N is the number of input layer neurons. The two layers are fully interconnected by lattice $\mathbb{V}_L$ weighted links that filter -up or -down activity of a layer. A "reset" node is used for resetting nodes in the category layer.

3.3 FLNN for learning - Clustering

The FLNN scheme for unsupervised leaning is described below next.

0. The first input is stored in the memory, that is FLNN's category layer. From then on learning proceeds as follows (at an instant t there are M=M(t) known classes c_k, k=1,…,M stored in the memory).
1. Present an input interval x to the initially "set" classes c_1,…,c_M.
2. Calculate $\sigma(x\le c_k)$ for all c_k, k=1,…M that have not yet been "reset", where $c_k=\bigcup_i w_{k,i}$ and $\sigma(x\le c_k) = \sigma(x\le \bigcup_i w_{k,i}) := \max_i \sigma(x\le w_{k,i})$.
3. Competition among the classes c_k : select c_J such that $\sigma(x\le c_J) = \max_k \sigma(x\le c_k)$, where J is the index of corresponding winner and $c_J=\bigcup_i w_{J,i}$.

4. The maximum-size-test (*assimilation condition* or *matching test*) : Is the size of $x \vee w$ less than a user defined threshold Z ? (where w corresponds to the $\max_{i} \sigma(x \le w_{J,i})$).
5. If the maximum-size-test is successful, then incorporate x into c_J by replacing w by $x \vee w$ and then by calculating the new quotient $Q(\{w_{J,i}\})$.
6. If the maximum-size-test fails then "reset" c_J. That is, while the current input x is present make c_J inaccessible during subsequent quests for a winner.
7. Completion-test : Are all the classes $c_1,\ldots,c_M$ "reset" ? If the completion-test fails go to step 2 to look for another winner.
8. If the completion-test is successful then memorize $x_{M+1}c=x$.

The previous algorithm corresponds to the learning phase of the FLNN scheme for clustering where learning is "on" and the classes are updated continually. As soon as learning is over the testing data are applied, and then only the degrees of inclusion $\sigma(x \le c_k)$ are calculated of an input datum x to all classes c_k, k=1,…,M stored in FLNN's category layer (Fig.2) and x is assigned to the winner class. During the testing phase no class is updated whatsoever.

Example-3.2

This example illustrates graphically the mechanics of the FLNN for clustering on the plane and it also demonstrates the technique of maximal expansions. Assume that the FLNN has already stored two distinct classes $c_1=\{w_1\}$ and $c_2=\{w_2\}$ in its category layer and let a new input x, that is a rectangle in the general case, be presented to the system as in Fig. 3(a). The two classes c_1 and c_2 compete with one another by comparing the inclusion measures $\sigma(x \le c_1)$ and $\sigma(x \le c_2)$, and let c_1 be the winner class. Assume that the maximum-size-test is not met then class c_1 is reset. Searching for a winner class continues, c_2 is tested next (Fig.3(b)), and let c_2 satisfy the maximum-size-test. Then w_2 is replaced by $w'_2=x \vee w_2$. Note that rectangles w_1 and w'_2 overlap. The FLNN for clustering assumes that now w_1 and w'_2 are in the same family of lattice intervals that defines a single class, say c_1, and it triggers the technique of maximal expansions. This technique considers intersection $w_1 \wedge w'_2$ and it expands it maximally in both dimensions (Fig.3(c)). Finally, one class c_1 is specified consisting of four rectangles $c_1=\{w_1,w'_2,w_3,w_4\}$. Rectangle w_1 is specified by its four corners 1-2-3-4, rectangle w'_2 is specified by its corners 5-6-7-8, rectangle w_3 is specified by 9-2-10-8, and rectangle w_4 by 11-6-12-4. The collection of the maximal rectangles for a given family of overlapping intervals, as in Fig. 3(c), is *the quotient* of the corresponding class. The degree of inclusion of a new input y in class c_1 as shown in Figure 3(d), is given by $\max\{\sigma(y \le w_1), \sigma(y \le w'_2), \sigma(y \le w_3), \sigma(y \le w_4)\}$. Note that input y could be a trivial interval, that is a point in the unit-square $[0,1]\times[0,1]$.

□

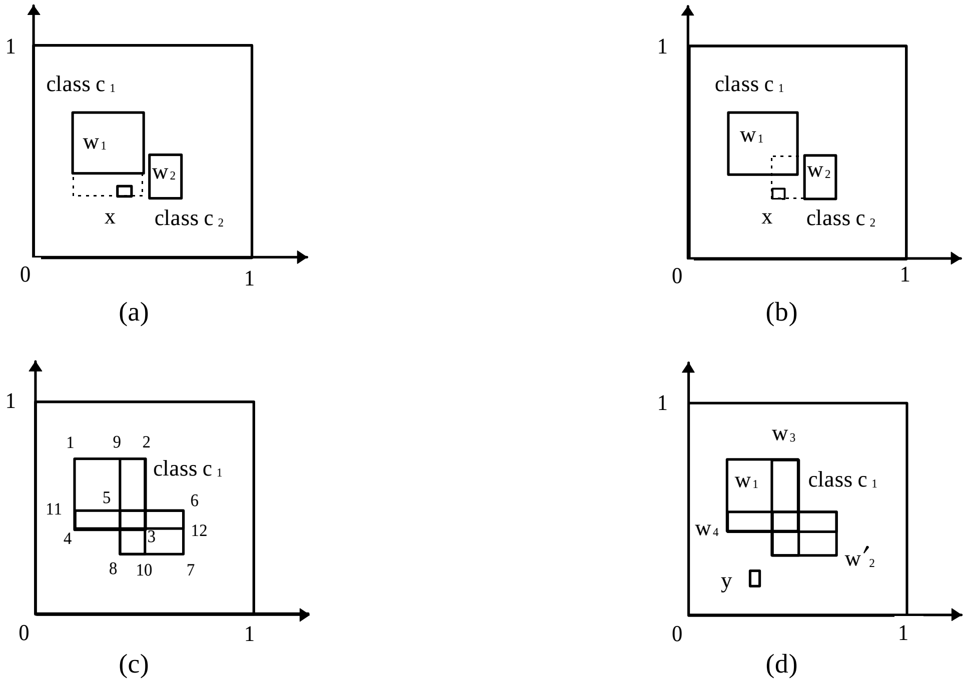

Figure 3 Learning by fuzzy lattice neural network (FLNN) for clustering. The two classes $c_1=\{w_1\}$ and $c_2=\{w_2\}$ compete over input x.

(a) Class c_1 is the winner because $\sigma(x\leq c_1)$ is larger that $\sigma(x\leq c_2)$, but it is reset because it does not satisfy the maximum-size-test.

(b) class c_2 is selected as the new winner that satisfies the maximum-size-test.

(c) rectangle w_2 is replaced by $w'_2=x\vee w_2$. The overlapping rectangles w_1 and w'_2 define now one class enhanced by rectangles w_3 and w_4 produced by the technique of maximal expansions, and

(d) a new input rectangle y appears and the learning cycle repeats.

3.4 FLNN for learning - Classification

In this case a different learning algorithm is implemented on the FLNN architecture. In particular a training set is employed explicitly consisting of N pairs (x_i,g_i), $i\in\{1,\ldots,N\}$ of data, where $x_i\in\mathbb{L}$ is an input pattern and element of a lattice $\mathbb{L}$, and g_i is an index for the category of x_i. That is the range of g_i is the set $\{1,\ldots,K\}$ where K is the number of categories. Note that the basic FLNN architecture of Fig. 2 has to be enhanced so as to accommodate information regarding the index g_i of a category. Such an accommodation can be achieved 1) by allowing for storage of an index g_i in the category layer, 2) by augmenting the input layer by one node so as to be able to store the index g_i of a category, and 3) by fully interconnecting the two layers. The FLNN scheme for supervised learning follows.

1. For i=1 to N consider training pair (x_i,g_i).
2. Set $x=x_i$.
3. For j-1 to N consider the next training pair (x_j,g_j) such that $g_j=g_i$.
4. Set $x'=x\vee x_j$.
5. For all (x_k,g_k), $k\in\{1,\ldots,N\}$ with $g_k\neq g_i$, test whether $\sigma(x_k\leq x')<1$.
6. If the test in Step 5) succeeds then replace x by x'.
7. If j<N go to Step 3).
8. Store x in the category layer of the FLNN together with g_i indicating x's category.

The above algorithm for training the FLNN terminates. In particular there exist three processing loops at Steps 1), 3), and 5) implying that the complexity is $O(N^3)$, where N is the number of training data. Note that the technique of maximal expansions was not employed for training.

Regarding the testing phase, the testing set is applied and the degrees of inclusion $\sigma(x\leq c_k)$ are calculated for each datum x to all classes c_k, k=1,…,M in FLNN's category layer (Fig.2). Finally x is assigned to the winner class, that is the class which provides with the largest degree of inclusion.

For both the clustering and the classification schemes, FLNN allows overlapping of intervals. The idea for interval overlapping was borrowed from [18] where mutually nonexclusive classes are treated and the mechanics of human pattern recognition are modeled with good learning results in benchmark data sets.

At this point it is worth referring to the neural architecture in [16] which maps fuzzy input vectors to fuzzy outputs. The architecture in question is able to integrate knowledge, presented by fuzzy if-then rules, and numerical data into a single information processing system. This is effected by employing an extension of the conventional backpropagation algorithm which (extended algorithm) can handle h-level sets of fuzzy numbers. Despite its enhanced scope of applicability the architecture in [16] can not accommodate additional types of data, for instance symbols and/or propositions. On the other hand the FLNN is universal in scope and it can handle in principle any type of data suffices the data in question be lattice-ordered and both a dual-automorphism (to be discussed in the sequel) and a function-h be available.

4 DEFINITION OF AN INCLUSION MEASURE IN THE LATTICE $\mathbb{V}_L$ OF INTERVALS

In the previous section we assumed existence of an inclusion measure in the lattice $\mathbb{V}_L$ of intervals. Herein we show sufficient conditions for this existence. The only way known to the authors for constructing an inclusion measure is via a function-h. Nevertheless we could not find such a function in the lattice $\mathbb{V}_L$ of intervals. Therefore we decided to search for an inclusion measure $\sigma(.)$ in lattice $\mathbb{PL}_2$. Note that such a $\sigma(.)$ would be a valid inclusion measure in lattice $\mathbb{V}_L$ as well, because $\mathbb{V}_L$ is a sublattice of $\mathbb{PL}_2$. Unfortunately we could not define an inclusion measure in $\mathbb{PL}_2$ because we could not find a function-h in $\mathbb{PL}_2$. Nevertheless we already know that a function-h in lattice $\mathbb{L}$ would imply a function-h in lattice $\mathbb{PL}_1$ by virtue of Lemma 1. Therefore we directed our efforts to finding an isomorphism between lattices $\mathbb{PL}_2$ and $\mathbb{PL}_1$. Recall that an isomorphism between two lattices is a bijection (one-one correspondence) between them such that the image of the meet equals the meet of the images and likewise for the join [3]. A sufficient condition was found for the existence of an isomorphism between lattices $\mathbb{PL}_1$ and $\mathbb{PL}_2$.

We have found that the existence of a dual-automorphism in $\mathbb{L}$ is sufficient for establishing an isomorphism between lattices $\mathbb{PL}_1$ and $\mathbb{PL}_2$. Recall that a dual-automorphism in a lattice $\mathbb{L}$ is a bijection (one-one correspondence) $\theta:\mathbb{L}\rightarrow\mathbb{L}$ such that $x\leq y \Leftrightarrow \theta(x)\geq\theta(y)$, [3]. To show that a dual-automorphism in $\mathbb{L}$ is sufficient for an isomorphism between $\mathbb{PL}_1$ and $\mathbb{PL}_2$ assume that there exists a dual-automorphism in the complete lattice $\mathbb{L}$. Then $[a,b]\leq[c,d]$ in $\mathbb{PL}_2$ implies $c\leq a$ and $b\leq d \Rightarrow \theta(c)\geq\theta(a)$ and $b\leq d \Rightarrow (\theta(a),b)\leq(\theta(c),d)$ in $\mathbb{PL}_1$, and vice-versa. Therefore an isomorphism is implied between lattices $\mathbb{PL}_1$ and $\mathbb{PL}_2$, and elements $(a,b)\in\mathbb{PL}_1$ and $[\theta(a),b]\in\mathbb{PL}_2$ are isomorphic. A dual-automorphism $\theta(.)$ in $\mathbb{L}$ implies $\theta(O)=I$ and $\theta(I)=O$, hence the least element (O,O) of $\mathbb{PL}_1$ maps to the least element $[\theta(O),O]=[I,O]$ of $\mathbb{PL}_2$. Likewise, the greatest element (I,I) of $\mathbb{PL}_1$ maps to the greatest element $[\theta(I),I]=[O,I]$ of $\mathbb{PL}_2$.

To recapitulate, an inclusion measure can be defined in the lattice $\mathbb{V}_L$ of intervals as follows.

1. A function-h in lattice $\mathbb{L}$ can define a function-h in lattice $\mathbb{PL}_1$ by virtue of Lemma 1.
2. Function-h in lattice $\mathbb{PL}_1$ implies an inclusion measure $\sigma(.)$ in $\mathbb{PL}_1$ by virtue of Theorem 1.
3. A dual-automorphism $\theta(.)$ in $\mathbb{L}$ can define an isomorphism between $\mathbb{PL}_1$ and $\mathbb{PL}_2$, as it was shown above.

4. In conclusion, an inclusion measure between two intervals $[a,b],[c,d]\in\mathbb{V}_L$ is given by $\sigma([a,b]\leq[c,d])=\sigma((\theta(a),b)\leq(\theta(c),d))$, where $(\theta(a),b)$ and $(\theta(c),d)$ are in $\mathbb{PL}_1$ and $\sigma(.)$ is the inclusion measure in $\mathbb{PL}_1$.

Regarding computation of the meet ($\wedge$) and join ($\vee$) in lattice $\mathbb{V}_L$ note that the join $[a,b]\vee[c,d]=[a\wedge c,b\vee d]$ is straightforward because it is always $a\wedge c\leq b\vee d$. However, computation of the meet $[a,b]\wedge[c,d]$ is attained only after the following *inclusion test* in $\mathbb{V}_L$:

$a\vee c\leq b\wedge d$ (true) $\Rightarrow [a,b]\wedge[c,d] = [a\vee c,b\wedge d]$, and

$a\vee c\leq b\wedge d$ (false) $\Rightarrow [a,b]\wedge[c,d] := [I,O]$

That is any generalized interval $[x,y]$, for which it is not $x\leq y$, is mapped to the least element in the lattice $\mathbb{V}_L$ of intervals. Since all versions of the FLNN presented in this paper use solely the lattice join ($\vee$) operator to update intervals during learning, we did not employ the aforementioned inclusion test herein.

Example-4.1

Let's consider our familiar lattice-chain $\mathbb{I}=[0,1]$. Then function $\theta(x)=1-x$ defines a dual-automorphism in lattice $\mathbb{I}=[0,1]$. The isomorphic of an interval, say $[0.1,0.3]\in\mathbb{V}_I\subset\mathbb{PI}_2$ is $(\theta(0.1),0.3)=(1-0.1,0.3)=(0.9,0.3)$ in lattice $\mathbb{PI}_1$. Therefore the degree of inclusion of an interval, say [0.1,0.3], to another one, say [0.2,0.5], is calculated as follows:

$$\sigma([0.1,0.3]\leq[0.2,0.5])=\sigma((\theta(0.1),0.3)\leq(\theta(0.2),0.5)=\sigma((0.9,0.3)\leq(0.8,0.5))=\frac{h(0.8,0.5)}{h((0.9,0.3)\vee(0.8,0.5))}\cong 0.928,$$

where $h(x)=x$. Conversely the degree of inclusion of [0.2,0.5] in [0.1,0.3] is $\sigma([0.2,0.5]\leq[0.1,0.3])\cong 0.857$.

To keep building on the plane consider the unit-square lattice $\mathbb{U}=[0,1]\times[0,1]$, where an interval in the unit-square is a rectangle. Consider two rectangles in the unit-square, say $[0.1,0.5]\times[0.2,0.3]$ and $[0.3,0.6]\times[0.1,0.6]$ as shown in Fig. 4. Rectangle u: $[0.1,0.5]\times[0.2,0.3]$ corresponds to element $[0.1,0.5,0.2,0.3]$ of $\mathbb{PU}_2$ whose isomorphic is $(\theta(0.1),0.5,\theta(0.2),0.3)= (0.9,0.5,0.8,0.3)$ in $\mathbb{PU}_1$, whereas the isomorphic of rectangle w: $[0.3,0.6]\times[0.1,0.6]$ is $(\theta(0.3),0.6,\theta(0.1),0.6)= (0.7,0.6,0.9,0.6)$ in $\mathbb{PU}_1$. The degree of inclusion of rectangle $[0.1,0.5]\times[0.2,0.3]$ in rectangle $[0.3,0.6]\times[0.1,0.6]$ is computed to be $\sigma([0.1,0.5,0.2,0.3] \leq [0.3,0.6,0.1,0.6]) \cong 0.933$, whereas the degree of inclusion of rectangle $[0.3,0.6]\times[0.1,0.6]$ in rectangle $[0.1,0.5]\times[0.2,0.3]$ is $\sigma([0.3,0.6,0.1,0.6]\leq [0.1,0.5,0.2,0.3])\cong 0.833$.

For the N-dimensional space an interval is an N-dimensional hyperbox, or *hyperbox* for short.

□

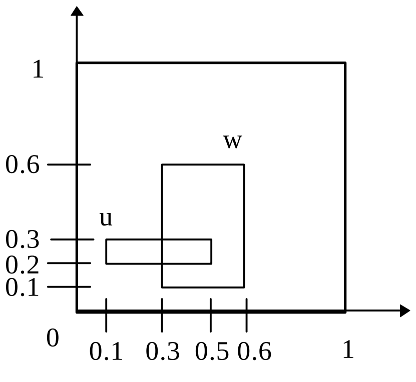

Figure 4 In the unit-square it can be calculated that rectangle u in included in rectangle w more than the other way around when functions $h(x)=x$ and $\theta(x)=1-x$ are selected, respectively, as function-h and dual-automorphism in the constituent lattices $\mathbb{I}=[0,1]$.

5 EXPERIMENTS

Our goal is to demonstrate the viability of the FLNN as an effective tool for clustering and classification. Moreover, results of other methods are cited from the relevant references in order to compare performance on the same data sets. Since the bulk of the benchmark data are drawn from the conventional Euclidean space we dealt with the Euclidean space in most of the examples in the sequel.

Five benchmark and one synthetic data set were processed by the FLNN. The benchmark data sets can be obtained either from the Carnegie Mellon University's neural net benchmark collection [9], or from the University of California Irvine repository of machine learning databases [24]. For some of the benchmarks a training set and a testing set are given explicitly. In this case only the training set is employed for learning by either clustering or classification. On the other hand when the data are not given separated into a training set and a testing set, a training set was formed either by leaving 1 datum out or by leaving randomly 25% of the data out. For leave-1-out, the experiment was repeated so as to leave, in turn, most data out. For leave-25%-out ten trials were carried out and the average performance is reported. The success of an experiment is expressed by the percentage of correctly recognized testing data.

We introduce two complete lattices, namely the unit hypercube and the convex fuzzy sets.

5.1 The Unit Hypercube

We summarize in this subsection our discussions regarding lattice-chain $\mathbb{I}$=[0,1]. The lattice-chain $\mathbb{I}$= [0,1] is complete where 0.0 and 1.0 are respectively its least and greatest elements. A function-h and a dual-automorphism are given by h(x)=x and θ(x)=1-x, respectively. Hence the corresponding lattices $\mathbb{PI}_1$ and $\mathbb{PI}_2$ are isomorphic to each other. By employing Lemma 1 one can go to more dimensions and hence consider hyperboxes in the unit hypercube. Note that employment of the unit hypercube instead of the whole Euclidean space is acceptable in engineering practice [22]. Finally note that the dual-automorphism θ(.) is a set theoretic interpretation in the context of the FL-framework of the so called "complement coding" technique employed by the fuzzy adaptive resonance theory (fuzzy ART) [5].

5.2 Fuzzy Numbers

Let $\mathbb{C}$ be the collection of convex fuzzy sets defined over a subset of a linear space; the latter space is typically a subspace of R^N. It can be shown that $\mathbb{C}$ is a lattice under the regular containment $A\subset B \Leftrightarrow f_A \leq f_B$, where f_A and f_B are, respectively, membership (characteristic) functions for fuzzy sets A and B [35]. To this end we will show that if A,B∈$\mathbb{C}$ then both $A\wedge B$ and $A\vee B$ exist in $\mathbb{C}$. It is known from [35] that $A\cap B$ is the largest set contained in both A and B, and moreover $A\cap B$ is convex; hence we define $A\wedge B := A\cap B$. On the other hand let $\mathbb{C}_{AB}$ be the collection of convex fuzzy sets that include both A and B. Consider the fuzzy set M= $\bigcap_{X\in C_{AB}} X$. By employing the "completeness axiom" from real analysis it can be shown that $\bigcap_{X\in C_{AB}} X$ is the minimum fuzzy set in $\mathbb{C}_{AB}$. Hence we define $A\vee B = \bigcap_{X\in C_{AB}} X$. Following the terminology in [35] the smallest convex fuzzy set containing $A\cup B$ is called the *convex hull* of $A\cup B$ and it is denoted by conv($A\cup B$), that is $A\vee B :=$ conv($A\cup B$). In conclusion, the set $\mathbb{C}$ constitutes a lattice.

Remark that the collection of all fuzzy sets (including convex fuzzy sets) is a lattice under the ∩ and ∪ operations [35]. Nevertheless we opted for the lattice $\mathbb{C}$ of convex fuzzy sets in particular because of an important potential advantage : that is the generalization implied by replacing A and B by conv($A\cup B$) instead of replacing them by $A\vee B$.

□

In all the learning experiments in the sequel both a training set and a testing set are employed. Learning is effected on the training set in either supervised or unsupervised fashion, and then learning is rated by applying the testing set. Due to the aforementioned similarities between supervised and unsupervised learning experiments we decided to put results of both types of experiments in the same table for one benchmark data set.

5.3 Experiment-5.1

The Fisher IRIS benchmark [10], [24] was processed because its familiarity to the scientific research community may allow an assessment of the relative performance. This data set comprises 4 attributes for three classes of iris flowers. Fifty iris flowers per class result in a total of 150 data. The aim was the correct classification to one of the three iris types. A training set is not given explicitly.

Table 1 cites the percentage of correct classifications of the testing set by different methods. Backpropagation and structural learning with forgetting (SLF) were employed in [17] with a randomly selected training set and averages over five trials. For SLF a statistical overall optimum of 5.0 errors is reported when the training set contained 30 data. For backpropagation the optimal performance was an average of 4.8 errors when the training set contained 60 data. A constrained multiple objective criterion (CMOC) neural network [26] misclassified 6 of the testing data when the first half instances in each IRIS class were employed for training. The min-max neural network for clustering [31] reports an optimum 10% plus 12% misclassification rate for two IRIS classes that corresponds to 11 misclassifications, for a hyperbox size equal to 0.10. The success percentage rates for the FLNN are shown in Table 1.

In addition, for this benchmark data set, a 2-level FLNN for clustering was employed as follows. The data were presented to the FLNN for clustering with a size Z of 1.71 resulting in fifteen (15) 4-D hyperboxes. Then a second level clustering was effected by determining the degree of inclusion of each hyperbox in turn in all other, in order to associate a hyperbox with another one which corresponded to the largest inclusion measure. Thus the 15 hyperboxes were partitioned to three groups. In conclusion the Fisher's IRIS data were clustered in three classes with only three (3) data misclassifications (Table 1).

Table 2 compares the performances of the min-max neural network [31] with the 2-level FLNN for clustering. Besides FLNN's smaller misclassification percentage, it is also important to note that the FLNN tracks down 3 classes after two consecutive levels of clustering without a teacher, whereas the min-max neural network identifies a total of 14 clusters in one level of clustering which are then assigned to the 3 IRIS classes by an external teacher.

TABLE 1 Recognition results for the IRIS benchmark by various neural networks.

Neural Network	% Right on Testing
FLNN for classification, leave-1-out	99.34
FLNN for classification, leave-25%-out	99.12 (*)
2-level FLNN for clustering	98.00
SLF	95.84
FLNN for clustering, leave-25%-out	95.04 (*)
Back Propagation	94.67
FLNN for clustering, leave-1-out	94.00
Min-max neural net for clustering	92.67
CMOC	92.00

(*) Average in 10 random trials

TABLE 2 Comparing performances of the min-max neural net versus the 2-level FLNN for clustering for the IRIS benchmark.

	Misclassifications for			Number of clustering levels	Number of classes identified
	Class-1	Class-2	Class-3		
Min-max neural net for clustering	-	12 %	10%	1	14
2-level FLNN for clustering	-	6 %	-	2	3

5.4 Experiment-5.2

The SONAR benchmark was employed from the Carnegie Mellon University collection of neural net benchmarks [9]. The data are 60-dimensional vectors that correspond to sonar signals bounced off either a metal cylinder or a cylindrical rock. 104 vectors are provided for training and another 104 for testing. The aim was the correct classification of sonar testing signals to one of the two classes: “Mine” or “Rock”.

Table 3 summarize results by various methods. Back propagation and nearest neighbor results are reported in the documentation that accompanies the SONAR data. Results by K-nearest neighbor and the fuzzy adaptive resonance associative map (ARAM) with voting across five simulations are from [34].

Regarding FLNN for clustering the SONAR training data were fed in for various values of the size Z. Table 4 details the results of learning with different sizes of Z, and the additional column “Right on Training Set” is listed to facilitate the comparison with the results cited in Table 3. In particular, 5999 processing cycles were effected for size Z values from 0 to 5.998 in steps of 0.001. In every processing cycle the training data were fed only once and a number of clusters was identified. Each cluster was labeled either “Rock” or “Mine” according to the majority of the data it encoded. Then the SONAR testing data were fed in and the results in Table 4 came out. Note that selection of a good value for Z can be regarded as a “parameter tuning problem” and it depends on the specific problem at hand.

TABLE 3 Recognition results for the SONAR benchmark by various methods.

Method	% Right on Training Set	% Right on Testing Set
FLNN for classification	100.0	100.00
FLNN for clustering	100.0	94.23
Fuzzy ARAM	100.0	94.20
K-Nearest Neighbor	100.0	91.60
BackProp: Angle-Dep. (1)	99.8	90.40
BackProp: Angle-Dep.	99.4	89.30
BackProp: Angle-Dep.	100.0	89.20
BackProp: Angle-Dep.	98.1	87.60
BackProp: Angle-Dep.	96.2	85.70
Nearest Neighbor	100.0	82.70
BackProp: Angle-Ind. (2)	89.4	77.10

(1) Angle Dependent data ordering.
(2) Angle Independent data ordering.

TABLE 4 Performance of the FLNN-for-clustering in classifying SONAR returns for various values of the size Z. The selection of a “good ” value for Z depends on the specific recognition problem.

size Z range	% Right on Training Set	% Right on Testing Set
0-3.077	100.00	93.27
3.078-3.329	100.00	94.23
3.33-3.763	100.00	90.38
3.764-4.333	100.00	91.35
4.334-4.751	100.00	90.38
4.752-4.802	100.00	89.42
4.803-4.804	100.00	85.58
4.805-5.243	100.00	84.62
5.244-5.363	99.03	87.50
5.364-5.379	99.03	90.38
5.38-5.434	99.03	87.50
5.435-5.636	98.07	85.58
5.637-5.892	98.07	88.46
5.893-5.998	98.07	90.38

5.5 Experiment-5.3

The WINE benchmark was employed from the UCI repository of machine learning databases [24]. The data are 13-dimensional vectors that correspond to various wine constituents determined by chemical analysis. 178 data vectors are given distributed by 59, 71, and 48 in three wine types. The aim was the correct classification to one of the three wine types. A training set is not given explicitly.

Table 5 summarizes the classification results by different methods. Results by Regularized Discriminant Analysis (RDA), Quadratic Discriminant Analysis (QDA), Linear Discriminant Analysis (LDA), and 1-Nearest-Neighbor (1NN) are reported in the documentation that accompanies the WINE benchmark, however no details on the training and testing sets are given in there. Procedure CLUSTER from the SAS/STAT package, and Rprop (Resilient Backpropagation) in Table 5 are the “best” among clustering and classification methods respectively reported in [18] for the WINE data set, where two-thirds of the data were used for learning.

TABLE 5 Recognition results for the WINE benchmark by various methods including Rprop (Resilient Backpropagation), RDA (Regularized Discriminant Analysis), QDA (Quadratic Discriminant Analysis), LDA (Linear Discriminant Analysis), and 1NN (1-Nearest-Neighbor).

Method	% Right on Testing
FLNN for classification, leave-1-out	100.00
Rprop	100.00
RDA	100.00
FLNN for classification, leave-25%-out	99.75 (*)
QDA	99.40
LDA	98.90
FLNN for clustering, leave-25%-out	96.55 (*)
1NN	96.10
FLNN for clustering, leave-1-out	96.10
CLUSTER	84.09

(*) Average in 10 random trials

5.6 Experiment-5.4

The GLASS benchmark was employed from the UCI repository of machine learning databases [24]. The data are 9-dimensional vectors that specify various chemical elements in two types of glass, these are float processed and non-float processed window glasses. 87 vectors are given for float processed and 76 vectors are given for non-float processed window glasses. The aim was the correct identification of a glass in one of the two classes. No training set is given explicitly.

Table 6 summarizes the classification results by different methods. Results by Beagle (that is a rule-based system), Nearest Neighbor, and Discriminant Analysis are reported in the documentation that accompanies the GLASS data, nevertheless no details are provided regarding the training and testing sets. Procedure FASTCLUS from the SAS/STAT package, and Rprop (Resilient Backpropagation) are respectively the "best" among clustering and classification methods reported in [18] for the GLASS benchmark, where two-thirds of the data were used for learning.

TABLE 6 Recognition results for the GLASS benchmark by various methods including Rprop (Resilient Backpropagation), FASTCLUS (a procedure from the SAS/STAT software package), and Beagle (a rule-based system).

Method	% Right on Testing
FLNN for classification, leave-1-out	99.38
FLNN for classification, leave-25%-out	98.88 (*)
Rprop	95.23
FASTCLUS	89.25
Nearest Neighbor	82.82
Beagle	82.20
FLNN for clustering, leave-25%-out	81.84 (*)
FLNN for clustering, leave-1-out	80.98
Discriminant Analysis	73.62

(*) Average in 10 random trials

5.7 Experiment-5.5

The IONOSPHERE benchmark was employed from the UCI repository of machine learning databases [24]. The aim was the correct identification of the type of radar returns in one of two classes from a vector of 34 attributes. The classes are "good" radar returns which show evidence of some type of structure in the ionosphere, and "bad" radar returns which do not. 200 instances are given for training and 151 instances are given for testing.

Table 7 summarizes the classification results by various methods. Results by linear perceptron, non-linear perceptron, back-propagation, nearest neighbor, Quinlan's C4 algorithm, and Aha's IB3 (an instance based algorithm) are reported in the documentation that accompanies the IONOSPHERE data set. The multiresolution algorithm (a variant of the min-max neural net), and Rprop are, respectively, the "best" among clustering and classification methods reported in [18] for the IONOSPHERE data set, where approximately two-thirds of the data were employed for learning.

□

TABLE 7 Recognition results for the IONOSPHERE benchmark by various methods including IB3 (a an instance-based algorithm), Rprop (Resilient Backpropagation), and Quinlan's C4 algorithm.

Method	% Right on Testing
FLNN for classification	100.00
IB3	96.70
Rprop	96.23
FLNN for clustering	96.02
Backpropagation	96.00
C4	94.00
Multiresolution Algorithm	92.87
Nearest Neighbor	92.10
Non-linear Perceptron	92.00
Linear Perceptron	90.70

Here concludes our FLNN processing of benchmark data sets. For all benchmarks it was noted a good performance of an FLNN scheme which (performance) is attributed, first, to the specific decision function employed by an FLNN scheme, that is the inclusion measure $k(x \leq u)$ of Theorem 1, and second, to the specific FLNN schemes for clustering and classification presented in section 3.

In general, the FLNN for classification performed better that the FLNN for clustering because of the presence of a teacher. Regarding the FLNN for classification a leave-1-out experiment performed better than a leave-25%-out experiment because for a leave-1-out experiment there existed more data in the training set. Nevertheless with regards to clustering, a leave-25%-out experiment provided consistently with a better result than a leave-1-out experiment. This led us to the conclusion that the FLNN for clustering generalizes better when more outliers are removed from the data. Finally note that an FLNN scheme for clustering may perform better than other classification methods, for instance with the Fisher IRIS benchmark (Table 1).

5.8 Experiment-5.6

We considered the collection $\mathbb{F}$ of fuzzy numbers over the domain [0,20]. In line with the analysis in subsection 5.2 $\mathbb{F}$ is a lattice. In particular lattice $\mathbb{F}$ is complete; its least element O equals 0 on [0,20], whereas its greatest element I equals 1 on [0,20]. On [0,20] a function-h was defined by the integral of a membership function over [0,20]. A dual-automorphism $\theta(.)$ was not identified; therefore we worked with intervals $[O,F_S]$, where O is the least element in $\mathbb{F}$ and $F_S \in \mathbb{F}$. Recall that for any dual-automorphism $\theta(.)$ it is $\theta(O)=I$; hence it follows $h(I)=20$. To calculate the degree of inclusion of an interval $[O,F_S]$ to another interval we need to compute the number $h\theta(O),F_S)= h(I)+h(F_S)= 20+h(F_S)$.

We will illustrate only the FLNN scheme for clustering on synthetic data. Let two fuzzy numbers $u_1,u_2 \in \mathbb{F}$ enter the FLNN (Fig.5(a)), where a size threshold Z=3 has been defined. Because $Z=3 < \text{size}(u_1 \vee u_2)= 4.75$, u_1 and u_2 were not replaced by $u_1 \vee u_2$. The technique of maximal expansions was applied. Each one of the u_1 and u_2 was expanded maximally to define fuzzy numbers u^*_1 and u^*_2, respectively, and the non-convex fuzzy set shown in Fig.5(b) was formed. Fuzzy numbers u^*_1 and u^*_2 can be discerned more clearly in Fig.5(c) and (d), respectively. A new "input" fuzzy number x enters the FLNN. Fig. 5 (c) and (d) show how the degrees of inclusion $\sigma(x \leq u^*_1)$ and $\sigma(x \leq u^*_2)$ can be calculated via $x \vee u^*_1$ and $x \vee u^*_2$, respectively. Because $\text{size}(x \vee u^*_1)>3$ and $\text{size}(x \vee u^*_2)>3$, the fuzzy number x is not incorporated into neither u^*_1 nor u^*_2 but it defines a new class of its own. Finally note that more lattices of fuzzy numbers can be considered jointly along additional dimensions.

On the other hand regarding the FLNN for classification note that it can be applied in principle, likewise.

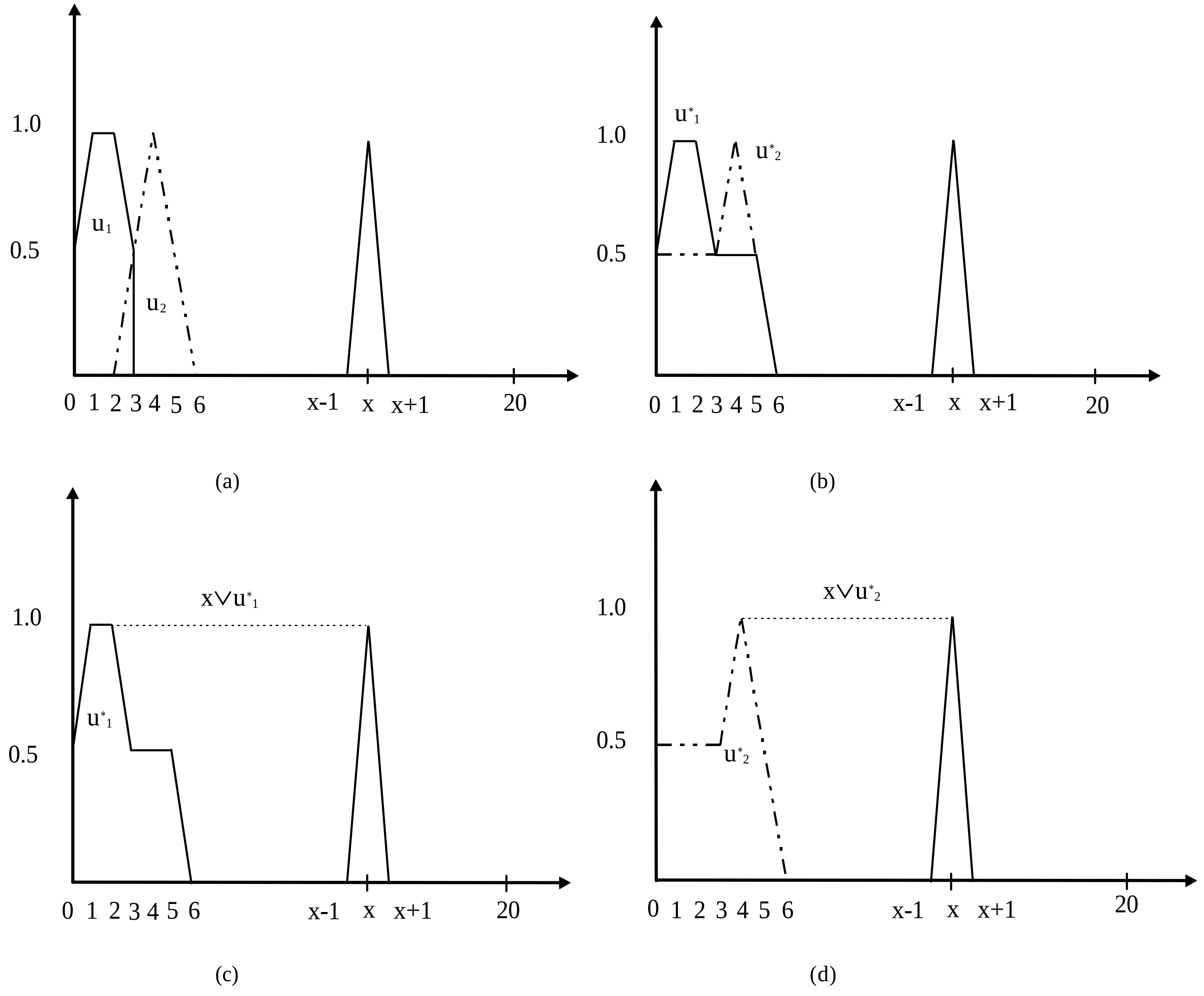

Figure 5 (a) Three fuzzy numbers enter the FLNN for clustering in the order u_1, u_2, and x.

(b) Because $Z=3<size(u_1 \vee u_2)=4.75$, u_1 and u_2 were not replaced by $u_1 \vee u_2$. Instead, u_1 and u_2 were expanded maximally to u^*_1 and u^*_2, and

(c) The inclusion measure $\sigma(x \leq u^*_1)$ is calculated by considering the $x \vee u^*_1$.

(d) The inclusion measure $\sigma(x \leq u^*_2)$ is calculated by considering the $x \vee u^*_2$.

6 DISCUSSION & CONCLUSION

The novel fuzzy lattice neural network (FLNN) for clustering and for classification was introduced with a wide domain due to its applicability on mathematical lattices. The FLNN emerges as a promising neural architecture for hybrid learning on disparate lattice data. The efficiency of FLNN for pattern recognition was demonstrated on five benchmark data sets. In all cases it compared well with other recognition methods. One additional example demonstrated the applicability of the FLNN beyond the conventional set R^N in one lattice of fuzzy numbers.

The FLNN draws on adaptive resonance theory (ART) [4], [5] and min-max neural network principles [30], [31]. Nevertheless the FLNN improves on both of these neurocomputing paradigms in a number of ways such as its applicability on mathematical lattices, the novel technique of maximal-expansions, and its capacity for treating intervals, which could compensate for the uncertainty of measurements.

We went one step further from demonstrating an efficient connectionist scheme for hybrid learning. Hence a new perspective to machine learning and decision making, namely the FL-framework, was delineated and novel formalism was introduced. Sound theoretical results were demonstrated such as the fuzzy degree of inclusion of a lattice interval into another one which can be calculated by a novel tool-concept, namely "inclusion measure".

Finally note that an efficient hardware implementation of the FLNN as shown in Fig.2 should address accordingly the "representation problem". That is the question of how a lattice element is stored and processed by the FLNN. Hence if one lattice element, like for example a fuzzy number, could be represented as a whole object in an analog form and if the lattice join ($\vee$) and meet ($\wedge$) operations could be performed in one step, then quite an efficient hardware implementation of the FLNN would result in.

Appendix

Proof of Theorem 1

1. $k(u \leq O) = \dfrac{h(O)}{h(u \vee O)} = \dfrac{h(O)}{h(u)} = 0$, assuming $u \neq O$.

2. $k(u \leq u) = \dfrac{h(u)}{h(u \vee u)} = \dfrac{h(u)}{h(u)} = 1$, $\forall u \in \mathbb{L}$.

3. $u \leq w \Rightarrow h(x \vee w)-h(x \vee u) \leq h(w)-h(u) \Rightarrow h(x \vee w) \leq h(w) - h(u) + h(x \vee u)$. For $w \neq O$ it holds $\dfrac{h(u)}{h(w)} h(x \vee w) \leq \dfrac{h(u)}{h(w)} [h(w) - h(u) + h(x \vee u)] = \dfrac{h(w)-h(u)}{h(w)} h(u) + \dfrac{h(u)}{h(w)} h(x \vee u) \leq \dfrac{h(w)-h(u)}{h(w)} h(x \vee u) + \dfrac{h(u)}{h(w)} h(x \vee u) = h(x \vee u) \Rightarrow \dfrac{h(u)}{h(x \vee u)} \leq \dfrac{h(w)}{h(x \vee w)} \Rightarrow k(x \leq u) \leq k(x \leq w)$.

Proof of Theorem 2

L1. $[a,b] \wedge [a,b] = [a \vee a, b \wedge b] = [a,b]$.

L2. $[a,b] \wedge [c,d] = [a \vee c, b \wedge d] = [c \vee a, d \wedge b] = [c,d] \wedge [a,b]$.

L3. $[a,b] \wedge ([c,d] \wedge [e,f]) = [a,b] \wedge [c \vee e, d \wedge f] = [a \vee (c \vee e), b \wedge (d \wedge f)] =$
$= [(a \vee c) \vee e, (b \wedge d) \wedge f] = [a \vee c, b \wedge d] \wedge [e,f] = ([a,b] \wedge [c,d]) \wedge [e,f]$.

In cases L1-L3 the truth for the joint operation $\vee$ may be shown dually.

L4. $[a,b] \wedge ([a,b] \vee [c,d]) = [a,b] \wedge [a \wedge c, b \vee d] = [a \vee (a \wedge c), b \wedge (b \vee d)] = [a,b]$, and
$[a,b] \vee ([a,b] \wedge [c,d]) = [a,b] \vee [a \vee c, b \wedge d] = [a \wedge (a \vee c), b \vee (b \wedge d)] = [a,b]$.

Proof of Lemma 1

Suffices to show that function $h(x_1,\ldots,x_N)=h_1(x_1)+\ldots+h_N(x_N)$ satisfies (P1) - (P3) of Definition 3.

(P1) Let $h_i(O_i)=0$, for $i \in \{1,\ldots,N\}$, where O_i is the least element of lattice $\mathbb{L}_i$. Then $h(O)=h(O_1,\ldots,O_N)= h_1(O_1)+\ldots+h_N(O_N)=0$, where $O=(O_1,\ldots,O_N)$ is the least element of lattice $\mathbb{L}=\mathbb{L}_1 \times \ldots \times \mathbb{L}_N$.

(P2) Since $h_i(.), i \in \{1,\ldots,N\}$ is a function-h it holds $u_i<w_i \Rightarrow h_i(u_i)<h_i(w_i)$, where $u_i,w_i \in \mathbb{L}_i$. On the other hand, condition $u=(u_1,\ldots,u_N)<(w_1,\ldots,w_N)=w$ is equivalent to : $u_i \leq w_i$ for all $i \in \{1,\ldots,N\}$ and $u_i<w_i$ for at least one $i \in \{1,\ldots,N\}$, hence $h(u_i)<h(w_i)$ for at least one $i \in \{1,\ldots,N\}$. Therefore $(u_1,\ldots,u_N)<(w_1,\ldots,w_N) \Rightarrow h_1(u_1)+\ldots+h_N(u_N)<h_1(w_1)+\ldots+h_N(w_N)$, that is $u<w \Rightarrow h(u)<h(w)$, $u,w \in \mathbb{L}=\mathbb{L}_1 \times \ldots \times \mathbb{L}_N$.

(P3) Let $u_i \leq w_i \Rightarrow h(x_i \vee w_i)-h(x_i \vee u_i) \leq h(w_i)-h(u_i)$, $x_i,u_i,w_i \in \mathbb{L}._i$, for $i \in \{1,\ldots,N\}$, and
let $x=(x_1,\ldots,x_N)$, $u=(u_1,\ldots,u_N)$, $w=(w_1,\ldots,w_N)$, and $u=(u_1,\ldots,u_N)<(w_1,\ldots,w_N)=w$. Then
$h(x \vee w)-h(x \vee u) = h(x_1 \vee w_1,\ldots,x_N \vee w_N)-h(x_1 \vee u_1,\ldots,x_N \vee u_N) = [h_1(x_1 \vee w_1)+\ldots+h_N(x_N \vee w_N)]-[h_1(x_1 \vee u_1)+\ldots+h_N(x_N \vee u_N)] = [h_1(x_1 \vee w_1)-h_1(x_1 \vee u_1)]+\ldots+[h_N(x_N \vee w_N)-h_N(x_N \vee u_N)] \leq [h(w_1)-h(u_1)]+\ldots+[h(w_N)-h(u_N)] = [h_1(w_1)+\ldots+h_N(w_N)]-[h_1(u_1)+\ldots+h_N(u_N)] = h(w)-h(u)$.

Proof of Lemma 2

Let $\{w_i\}$ be a connected family of lattice intervals. A *maximal expansion* of $\{w_i\}$ is defined to be another family $\{q_i\}$ in the collection F of families that represent class c such that $\{w_i\}<\{q_i\}$. The intervals in $\{q_i\}$ are called *maximal intervals*. We will delineate a method for constructing an ever (strictly) larger maximal expansion of a family. This construction process will be shown to terminate and a global maximum will be reached, that is the quotient $Q(F) = Q(\{w_i\})$.

The truth of Lemma 2 will be shown by induction. Let family $\{w_i\}$ contain exactly two connected intervals, say w_1 and w_2. In order to progressively construct maximal expansions of family $\{w_1,w_2\}$, assume it is $w_1=[w_{11},\ldots,w_{1L}]$ and $w_2=[w_{21},\ldots,w_{2L}]$, where w_{1i} and w_{2i}, $i=1,\ldots,L$ are intervals "along" each constituent lattice and L is the total number of constituent lattices. The maximal expansion "along" the first constituent lattice is determined by inserting to the family $\{w_1,w_2\}$ the interval $[\max(w_{11}\wedge w_{21}),w_{12}\wedge w_{22}, \ldots,w_{1L}\wedge w_{2L}]$, where $\max(w_{11}\wedge w_{21})$ is the largest interval in the first constituent lattice which contains $w_{11}\wedge w_{21}$ and consists of elements of w_{11} or w_{21}. The latter is a trivial problem framed within the first constituent lattice. In the sequel consider the maximal expansions "along" the rest of the constituent lattices, these are at the most another L-1 maximal expansions. The set-union $me(w_1,w_2)$ of the intervals in all maximal expansions "along" all the constituent lattices has to be the maximum element in F that is the quotient $Q(F)$.

To prove the latter statement consider any interval u which contains only elements of $w_1\cup w_2$. If u contains exclusively elements of w_1 OR exclusively elements of w_2 then it will be $u\leq w_1$ or $u\leq w_2$ respectively, hence $u\leq me(w_1,w_2)$. On the other hand, suppose that $u=[u_1,\ldots,u_L]$ contains exclusive elements of $w_1=[w_{11},\ldots,w_{1L}]$ AND exclusive elements of $w_2=[w_{21},\ldots,w_{2L}]$, where "exclusive element" means that it belongs only to one interval and not to the other. This implies that for at least one constituent lattice interval u_i, $i=1,\ldots,L$ it will be $w_{1i}\wedge w_{2i}<u_i$. But such a strict inequality can be true for at most one constituent lattice interval; otherwise u would contain elements that do not belong to neither w_1 nor w_2. Because of the way the set $me(w_1,w_2)$ was constructed it follows that $u\leq me(w_1,w_2)$. In conclusion, $me(w_1,w_2)$ is the maximum family in $\{w_i\}$, $i\in\{1,2\}$, that is the quotient $Q(\{w_1,w_2\})=me(w_1,w_2)=Q(F)$.

Consider now a third interval w_3 such that $\{w_3\}\cup Q(\{w_1,w_2\})$ is connected. Assume the maximal expansions $me(w_3,w_1)$ and $me(w_3,w_2)$. Then any interval u containing only elements of one of w_1, w_2, w_3, w_1 and w_2, w_2 and w_3, w_3 and w_1 will be included in $me(w_1,w_2)\cup me(w_2,w_3)\cup me(w_3, w_1)$. In addition to that and in order to consider intervals containing exclusive elements of w_1 AND w_2 AND w_3, if any, the following maximal expansions will have to be considered : $me(w_1, me(w_2,w_3))$, $me(w_2, me(w_3,w_1))$, and $me(w_3, me(w_1,w_2))$. Note that one of the latter maximal expansions, say, $me(w_3, me(w_1,w_2))$ between an interval w_3 and the family of intervals $me(w_1,w_2)$ is calculated by finding the maximal expansions between w_3 and all the intervals in $me(w_1,w_2)$. Finally the set-union of the intervals in all the resulting maximal expansions is the quotient $Q(\{w_1,w_2,w_3\})$. Apparently the problem becomes a combinatorial one and the truth of Lemma 2 follows, in general, by mathematical induction.

□

The aforementioned algorithm which calculates the quotient $Q(\{w_i\})$ of a connected family $\{w_i\}$ of intervals, where i belongs to a finite index set, is called *algorithm-Q*. Note that the order in which the intervals of $\{w_i\}$ are selected to calculate the maximal expansions is not important. Remark that algorithm-Q is not computationally efficient due to the exponential explosion of the required operations as a family's cardinality increases. Nevertheless, it does find the quotient $Q(\{w_i\})$ and it was proven handy in the classification examples presented in this paper where families of relatively small cardinality (a few tens of intervals, at the most) were identified.

Finally note that any interval not connected to the rest intervals in a family does not have to be considered in any maximal expansion because such an interval is by itself a maximal interval. Likewise any connected group of intervals, but detached from the rest intervals in a class, is treated by itself when calculating the maximal expansion of the class the group in question belongs to.

REFERENCES

[1] D.H. Ackley, G.E. Hinton, and T.J. Sejnowski, “A Learning Algorithm for Boltzmann Machines”, *Cognitive Science*, vol. 9, pp. 147-169, 1985.

[2] S.I. Amari, “Dualistic Geometry of the Manifold of Higher-Order Neurons”, *Neural Networks*, vol. 4, no. 4, pp. 443-451, 1991.

[3] G. Birkhoff, *Lattice Theory*, American Mathematical Society, Colloquium Publications, vol. 25, 1967.

[4] G. Carpenter, and S. Grossberg, “A massively parallel architecture for self-organizing neural pattern recognition machine”, *Computer Vision, Graphics and Image Understanding*, vol. 37, pp. 54-115, 1987.

[5] 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.

[6] D.D. Cofer, and V.K. Garg, “Supervisory Control of Real-Time Discrete-Event Systems Using Lattice Theory”, *IEEE Trans. Automatic Control*, vol. 41, no. 2, pp. 199-209, 1996.

[7] R. Desimone, *Neural Networks for Vision and Image Processing*, G.A. Carpenter and S. Grossberg Eds, Cambridge, MA, MIT Press, pp. 343-364, 1992.

[8] H. Dickhaus, and H. Heinrich, “Classifying Biosignals with Wavelet Networks”, *IEEE Engineering in Medicine and Biology magazine*, vol. 15, no. 5, pp. 103-111, 1996.

[9] S.E. Fahlman, and M.White, Carnegie Mellon University (CMU) Repository of Neural Net Benchmarks, [http://www.boltz.cs.cmu.edu/].

[10] R. Fisher, “The use of Multiple Measurements in Taxonomic Problems”, *Annals of Eugenics*, vol. 7 (II), pp. 179-188, 1936.

[11] K. Fukushima, S. Miyake, and T. Ito, “Neocognitron: A Neural Network Model for a Mechanism of Visual Pattern Recognition”, *IEEE Trans. Systems, Man, and Cybernetics*, vol. 13, no. 5, pp. 826-834, 1983.

[12] J.A. Goguen , “L-Fuzzy Sets”, *Journal of Mathematical Analysis and Applications*, vol. 18, pp. 145-174, 1967.

[13] S. Grossberg, “The Attentive Brain”, *American Scientist*, vol. 83, pp. 438-449, 1995.

[14] S. Haykin, *Neural Networks*, Macmillan College Publishing Company, Inc., 1994.

[15] J.J. Hopfield, “Neural networks and physical systems with emergent collective computational ability”, *Proc. of the National Academy of Sciences*, vol. 79, pp. 2554-2558, 1982.

[16] H. Ishibuchi, R. Fujioka, and H. Tanaka, “Neural Networks that learn from Fuzzy If-Then Rules”, *IEEE Trans. Fuzzy Systems*, vol. 1, no. 2, pp. 85-97, 1993.

[17] M. Ishikawa, “Structural Learning with Forgetting”, *Neural Networks*, vol. 9, no. 3, pp. 509-521, 1996.

[18] A. Joshi, N. Ramakrishman, E.N. Houstis, and J.R. Rice, “On Neurobiological, Neuro-Fuzzy, Machine Learning, and Statistical Pattern Recognition Techniques”, *IEEE Trans.Neural Networks*, vol. 8, no. 1, pp. 18-31.

[19] V.G. Kaburlasos, “Adaptive Resonance Theory with Supervised Learning and Large Database Applications”, Ph.D. Dissertation, Dept. Electrical Engineering, Univ. of Nevada, Reno, 1992.

[20] V.G. Kaburlasos, and V. Petridis, “Fuzzy Lattice Neurocomputing (FLN)”, *Proc. ISCA 5th Int’l Conf. on Intelligent Systems*, Reno, Nevada, June 19-21, pp. 56-60, 1996.

[21] T. Kohonen, *Self-Organizing Maps*, Springer-Verlag, Berlin Heidelberg, 1995.

[22] B.C. Lovell, and A.P. Bradley, “The Multiscale Classifier”, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp. 124-137, 1996.

[23] J.J. Lu, A. Nerode, and V.S. Subrahmanian, “Hybrid Knowledge Bases”, *IEEE Trans. Knowledge and Data Engineering* vol. 8, no. 5, pp. 773-785, 1996.

[24] Merz, C.J., & Murphy, P.M. (1996). UCI Repository of machine learning databases. [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

[25] S. Mitra, and S.K. Pal, “Fuzzy Self-Organization, Inferencing, and Rule Generation”, *IEEE Trans. Systems, Man, and Cybernetics - Part A*, vol. 26, no. 5, pp. 608-620, 1996.

[26] M.M. Moya, and D.R. Hush, "Network Constraints and Multi-objective Optimization for One-class Classification",*Neural Networks*, vol. 9, no. 3, pp. 463-474, 1996.

[27] B.A. Olshausen, C.H. Anderson, and D.C. Van Essen, "A Neurobiological Model of Visual Attention and Invariant Pattern Recognition Based on Dynamic Routing of Information", *The Journal of Neuroscience*, vol. 13, no. 11, pp. 4700-4719, 1993.

[28] C.W. Omlin, and C.L. Giles, "Extraction of Rules from Discrete-time Recurrent Neural Networks", *Neural Networks*, vol. 9, no. 1, pp. 41-52, 1996.

[29] V. Petridis, and V.G. Kaburlasos, "FLN: A Fuzzy Lattice Neural Network Scheme for Clustering", *Proc. World Congress on Neural Networks 1996 (WCNN'96)*, San Diego, California, September 15-18, pp. 942-945, 1996.

[30] P.K. Simpson, "Fuzzy Min-Max Neural Networks - Part1: Classification", *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 776-786, 1992.

[31] P.K. Simpson, "Fuzzy Min-Max Neural Networks - Part2: Clustering", *IEEE Trans. Fuzzy Systems*, vol. 1, no. 1, pp. 32-45, 1993.

[32] D.F. Specht, "Probabilistic Neural Networks and the Polynomial Adaline as Complementary Techniques for Classification", *IEEE Trans. Neural Networks*, vol. 1, no. 1, pp. 111-121, 1990.

[33] C.-T. Sun, "Rule-Base Structure Identification in an Adaptive-Network-Based Fuzzy Inference System", *IEEE Trans. Fuzzy Systems*, vol. 2, no.1, pp. 64-73, 1994.

[34] A-H. Tan, "Adaptive Resonance Associative Map", *Neural Networks*, vol. 8, no. 3, pp. 437-446, 1995.

[35] L.A. Zadeh, "Fuzzy Sets", *Information and Control*, vol. 8, pp. 338-353, 1965.

[36] H.J. Zimmermann, *Fuzzy Set Theory and Its Applications*, Kluwer Academic Publishers, Norwell, MA, 1991.