Fuzzy Lattice Neurocomputing (FLN) : A Novel Connectionist Scheme for Versatile Learning and Decision Making by Clustering*

Vassilis G. Kaburlasos, Vassilios Petridis

Aristotle University of Thessaloniki
Dept. of Electrical and Computer Engineering, GR-54006 Thessaloniki, Greece
email : vgkabs@egnatia.ee.auth.gr petridis@vergina.eng.auth.gr

Abstract

This paper proposes a fundamentally new and inherently hierarchical approach to neurocomputing that is fuzzy lattice neurocomputing (FLN). The application domain of the FLN is a general mathematical lattice. The notion *fuzzy lattice* is introduced herein to be a lattice whose inclusion-relation (≤) has been fuzzified. The FLN proposes a synergistic combination of fuzzy sets and lattice theory with adaptive resonance theory (ART) and min-max neurocomputing. The new theoretical notion *(lattice) inclusion measure* is defined and it is employed practically for learning and decision making by the FLN. Necessary conditions for the existence of a specific inclusion measure in a lattice are shown. The merits of the FLN are demonstrated practically in the set R^N on synthetic and benchmark data sets, where the FLN outperforms other neurocomputing schemes. An example is shown in the lattice of fuzzy sets over a universe of discourse. Lattices of disparate objects may be treated jointly with mathematical consistency. Another novelty is that the inputs to the FLN do not have to be individual lattice elements but they may be intervals of lattice elements.

Key words: Competitive neural networks, lattice theory, fuzzy set theory, clustering, pattern recognition, parallel processing.

1 Introduction and the Origins

A scheme for machine learning and decision making is typically crafted, either explicitly or implicitly, in the Euclidean space or the set R^N. With regards to neurocomputing in particular, the goal during the "learning phase" is to separate effectively that domain into regions, namely *categories* or *classes* $c_1,\ldots,c_K$, with respect to certain constraints. To this effect, a function $f_i(.)$ i=1,…,K is defined during learning for each category c_i, i=1,…,K such that when a new and hitherto unknown input, typically an N-dimensional point, is presented during the "testing phase" to be correctly classified to its corresponding class.

In this work we tried to go beyond the Euclidean space and to develop a neurocomputing scheme that can be applied on less structured sets; namely complete lattices. The goal meant to be a versatile scheme applicable on a wide domain. The result has been a novel neural scheme for learning and decision making, namely fuzzy lattice neurocomputing (FLN) scheme, which currently is used only for clustering. Different aspects of the FLN have been reported recently [9, 10]. Note that in the context of lattices the Euclidean space is only one alternative application domain.

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

According to the approach we have taken, a class of lattice elements is defined by a finite set of overlapping or non-overlapping lattice intervals. This implies that when the lattice R^N is employed in particular, then N-dimensional hyperboxes will be used to define a class. In this sense, our approach is similar to the one taken by the biologically motivated adaptive resonance theory (ART) [2, 3] and the min-max neural networks [11, 12]. But even though it is originated in the adaptive resonance as well as the min-max neural networks, fuzzy lattice neurocomputing (FLN) proceeds far beyond these approaches. It generalizes both of them by employing the theory of lattices [1] which is blended efficiently with the theory of fuzzy sets [13]. Thus, the activation function employed by FLN's neurons is lattice applicable instead of being strictly applicable to the set of real numbers.

A theoretical by-product of this research has been the introduction of the novel concept of *fuzzy lattice*, that is a lattice that fuzzifies its conventionally binary-valued inclusion-relation (≤). The *fuzzy lattice framework* or *FL-framework* is introduced herein with the aspiration to form a basis for the development of intelligent machine schemes. A fuzzy lattice is not to be confused with an L-fuzzy set [5]. The latter is a mapping from a universe of discourse onto a lattice and as such it is a generalization of a fuzzy set. On the other hand, the FLN is a neurocomputing scheme which approximates sets of lattice elements by lattice intervals and, as such, it is a generalization of the min-max neural networks [12] and the ART [3]. Lattice theory also provides for the hierarchical feature of the FLN since the product of lattices is a lattice [1]. Our oldest known attempt to fuzzy lattice neurocomputing (FLN) is noted in [8]. However, the work herein significantly systematizes and enhances it. Another important innovation introduced herein is that an input to be learned or to be classified can be a lattice interval. For the lattice R^N in particular, this remark implies that an input to the FLN can be a hyperbox instead of solely an N-dimensional point.

The domain of the FLN includes not only the Euclidean space but it includes other sets of objects retaining a "loose" relation to each other as in a lattice - for instance, the set of all waveforms and that of all images as well as their respective Fourier transforms. Another illustration is the set of all fuzzy sets defined on a universe of discourse, the corresponding Boolean set being one specific case. Additional examples is the set of hyperboxes inside the unit hypercube and the set of hyperspheres inside the unit hypersphere. Conventional N-dimensional vectors have been treated to-date by different neurocomputing paradigms without resorting to lattice theory. However, paradigms, like the adaptive resonance theory (ART) [2] which deals with binary vectors, the fuzzy ART [3] and the min-max neural networks [11, 12] which deal with analog vectors that describe hyperboxes, may be regarded in the context of FLN as its domain-specific variations.

Section 2 presents the new theoretical perspective for learning and decision making which is based on the theory of lattices. The mathematics involved is out of the usual track made for neurocomputing. Hence we avoided diligently any diversions into various obscure theoretical issues. Instead we dealt only with those aspects of the theory indispensable to reveal the hierarchical nature, the capacity for handling disparate data, and the consistency of learning and decision making. The proofs of the theorems are given in the appendix. Section 3 introduces the fuzzy lattice neurocomputing (FLN) scheme. New concepts are defined again wherever necessary. Section 4 reports on the capacity of the FLN for pattern discrimination and recognition on various data sets, and it compares it with that of other neural networks. Finally a brief discussion in section 5 accentuates the technological potential implied by an efficient hardware implementation of the FLN.

2 A Novel Framework to Learning and Decision Making

2.1 A hierarchy of fuzzy lattices

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 called *complete* when each of its subsets *X* has a least upper bound and a greatest lower bound. Moreover, a non-void complete lattice contains a least and a greatest element denoted by O and I respectively [1].

Let $\mathbb{L}$ be a lattice which may or may not be complete. Consider the relation R of the induced partial ordering in $\mathbb{L}$, and let $x,y \in \mathbb{L}$. Then it is known that $(x,y) \in R$ or $(y,x) \in R$ or none of the previous ordered pairs belongs to R. The novel notion *fuzzy lattice* is introduced at this very point in order to extend the crisp lattice relation, if it exists, of partial ordering to any pair (x,y) of the space $\{(x,y): x,y \in \mathbb{L}\}$. Such an extended relation is denoted by P and it is, in fact, a fuzzy inclusion relation. That is, to every $(x,y) \in \{(x,y): x,y \in \mathbb{L}\}$ a real number from the interval [0,1] is attached to denote the degree of inclusion of x in y, that is the degree of truth of $x \leq y$, where $x,y \in \mathbb{L}$. Formally the relation P can be defined [15] by

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

the real function $\mu_P(x,y)$ specifies the degree of inclusion of x in y.

To keep the extension P meaningful and compatible with the original lattice's $\mathbb{L}$ partial ordering relation R, the fuzzy relation 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 naturally.

Definition 1

A *fuzzy lattice* is a pair $(\mathbb{L}, \mu_P(x,y))$, where $\mathbb{L}$ is a conventional lattice and $\mu_P(x,y)$ is a fuzzy relation defined on $\{(x,y): x,y \in \mathbb{L}\}$ which specifies the degree of inclusion of x in y. It is $\mu_P(x,y)=1$ if and only if $x \leq y$ in $\mathbb{L}$.

The notion "fuzzy lattice" is meant to complement computationally the set of real numbers in the design of intelligent schemes for automated machine learning and decision making. We will refer to the set of all fuzzy lattices by the term *fuzzy lattice framework* or *FL-framework* in short.

To define a fuzzy lattice out of a conventional lattice $\mathbb{L}$, a real number will be attached to each element of lattice $\mathbb{L}$ by a function $v: \mathbb{L} \rightarrow \mathbb{R}$. In the set of all such functions that may be defined on $\mathbb{L}$ only the subset of the *positive valuations* will be considered. A valuation on a lattice $\mathbb{L}$ is a real-valued function v(.) on $\mathbb{L}$ which satisfies [1]

$$v(x) + v(y) = v(x \vee y) + v(x \wedge y).$$

A valuation is *isotone* if and only if $x \leq y$ implies $v(x) \leq v(y)$, and *positive* if and only if $x < y$ implies $v(x) < v(y)$.

A reason for selecting the set of the positive valuations is that in a lattice $\mathbb{L}$ with a positive valuation v(.), a distance function $d(x,y) = v(x \vee y) - v(x \wedge y)$ can be defined and hence $\mathbb{L}$ becomes a metric lattice [1]. Recall that the *distance* or *metric* is a nonnegative function d(x,y) for all (x,y) which satisfies (1) $d(x,y)=0$, if and only if $x=y$, (2) $d(x,y)=d(y,x)$, and (3) $d(x,y)+d(y,z) \geq d(x,z)$

(triangle inequality). Another implication of the existence of a positive valuation in $\mathbb{L}$ is that an inclusion measure can be defined in this lattice as well as in the lattice of its closed intervals as it will be shown in this section.

A lattice property states 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 anew with an induced partial ordering relation defined by $(x_1,x_2,\ldots,x_N) \leq (y_1,y_2,\ldots,y_N) \Rightarrow x_1\leq y_1, x_2\leq y_2,\ldots, x_N\leq y_N$ [1]. 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 a *constituent lattice*. Note that if the constituent lattices $\mathbb{L}_i$, $i\in\{1,\ldots,N\}$ are all complete then apparently the composite lattice $\mathbb{L}=\mathbb{L}_1\times\ldots\times\mathbb{L}_N$ will as well be a complete lattice. With regards to positive valuations in the constituent lattices $\mathbb{L}_1,\ldots,\mathbb{L}_N$ the following statement can be made.

Lemma 1

Let $\mathbb{L}=\mathbb{L}_1\times\ldots\times\mathbb{L}_N$ be the product of N lattices with positive valuations $v_1(.),\ldots,v_N(.)$ respectively. Then the function $v(x_1,\ldots,x_N)=v_1(x_1)+\ldots+v_N(x_N)$ defines a positive valuation in $\mathbb{L}=\mathbb{L}_1\times\ldots\times\mathbb{L}_N$.

∎

Lemma 1 is easily proven. We omit the proof for lack of space.

At this point it is necessary to cite a convention with regards to 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 plain 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 $x_i\in\mathbb{L}$. Same subscripts imply elements of the same constituent lattice and vice-versa.

2.2 The lattices $\mathbb{PL}_1$ and $\mathbb{PL}_2$ on the product set $\mathbb{L}\times\mathbb{L}$

To keep building up the hierarchy of lattices consider a complete product lattice $\mathbb{L}$ with O and I denoting its least and its greatest element respectively. In line with the previous analysis it can be inferred that if $\mathbb{L}$ is a complete lattice then the product $\mathbb{L}\times\mathbb{L}$ is itself a complete lattice, which will be denoted by $\mathbb{PL}_1= \mathbb{L}\times\mathbb{L}$. Furthermore recall that the implied relation of partial ordering in $\mathbb{PL}_1$, that is $(a,b)\leq(c,d)$ if and only if $a\leq c$ and $b\leq d$, is the same *relation* r_1 as before implied by the product lattice $\mathbb{L}=\mathbb{L}_1\times\ldots\times\mathbb{L}_N$. The join and the meet in $\mathbb{PL}_1$ are defined as follows

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

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 v(.) is a valuation in a lattice $\mathbb{L}$, then the function $V((a,b))= v(a)+v(b)$ defines a derived valuation in $\mathbb{PL}_1$. Moreover, it can also be easily shown that if the valuation v(.) is isotone (positive) in $\mathbb{L}$, then likewise is V(.) in $\mathbb{PL}_1$. Hence it is noted that when there exists a positive valuation in $\mathbb{L}$ then there exists another one in $\mathbb{PL}_1$. In the sequel another lattice, that is the lattice $\mathbb{PL}_2$ of *generalized intervals* in $\mathbb{L}$, is defined on the same product set $\mathbb{L}\times\mathbb{L}$ as it did lattice $\mathbb{PL}_1$ above. But the lattice $\mathbb{PL}_2$ of generalized intervals is defined with a different partial ordering.

Theorem 1

Let the set $\mathbb{PL}_2$ be $\mathbb{PL}_2$= {[a,b]: a,b∈$\mathbb{L}$} where $\mathbb{L}$ is a complete product lattice, and let two binary operations between elements of $\mathbb{PL}_2$ be defined as follows :

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

Then $\mathbb{PL}_2$ is a complete lattice because the following laws L1-L4 are satisfied, for a,b,c,d,e,f in $\mathbb{L}$ [1].

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. $[a,b]\wedge([c,d]\wedge[e,f]) = ([a,b]\wedge[c,d])\wedge[e,f]$
$[a,b]\vee([c,d]\vee[e,f]) = ([a,b]\vee[c,d])\vee[e,f]$ (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 1 is given in the Appendix.

The implied relation [a,b] ≤ [c,d] of partial ordering in $\mathbb{PL}_2$ is equivalent to $[a,b]\wedge[c,d] = [a,b]$ ⇔ $a\vee c=a$, $b\wedge d=b$ ⇔ c≤a, b≤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 $\mathbb{PL}_2$ is defined. Note that lattice $\mathbb{PL}_2$ considers the set {(x,y): x,y∈$\mathbb{L}$}, as it does lattice $\mathbb{PL}_1$. Nevertheless, parentheses ( ) are used to embrace the elements of $\mathbb{PL}_1$, whereas brackets [ ] are embracing the elements of $\mathbb{PL}_2$ intentionally, to underline the two different lattice orderings on the same product set $\mathbb{L}\times\mathbb{L}$.

It can be proven easily that if v(.) is a valuation in a lattice $\mathbb{L}$, then the function V([a,b]) = v(a)+v(b) defines a derived valuation in $\mathbb{PL}_2$. However note that an isotone valuation v(.) in $\mathbb{L}$ does not imply an isotone valuation of the form V([a,b])=v(a)+v(b) in $\mathbb{PL}_2$ as it did before in $\mathbb{PL}_1$. This can be shown by considering the relation [a,b]= x≤y =[c,d] ⇔ c≤a and b≤d. If v(.) is isotone in $\mathbb{L}$ then it holds c≤a ⇒ v(c)≤v(a) and b≤d ⇒ v(b)≤v(d). Hence the inequality relation between V(x)=v(a)+v(b) and V(y)=v(c)+v(d) is case dependent.

The lattice $\mathbb{PL}_2$ of generalized intervals is significant because it contains all the conventional intervals whom families could be employed to represent a set of lattice $\mathbb{L}$ elements. It would be quite advantageous to have a positive valuation function in $\mathbb{PL}_2$. Recall that the importance of the existence of a positive valuation in $\mathbb{PL}_2$ lies first of all in the fact that a metric could be defined in it. Second, a positive valuation function in $\mathbb{PL}_2$ would imply an *inclusion measure* as shown in the sequel, which is instrumental to applying the FLN scheme in the lattice in question. Nevertheless such a positive valuation could not be found in lattice $\mathbb{PL}_2$, provided one in lattice $\mathbb{L}$.

Instead of searching for a positive valuation in lattice $\mathbb{PL}_2$, which may not exist, we direct our efforts to finding an isomorphic relation between lattices $\mathbb{PL}_2$ and $\mathbb{PL}_1$. Such a search is fruitful and a sufficient condition can be found for the existence of an isomorphism between $\mathbb{PL}_1$ and $\mathbb{PL}_2$. Recall that an isomorphism between two lattices implies that the join of two elements in one of them corresponds to the join of their images in the other lattice; a similar correspondence exists for the meet. On the other hand, when two lattices are dual-isomorphic to each other, then the join of two elements in one of them corresponds to the meet of their images in the other lattice; further the meet of two elements in one of them corresponds to the join of their images in the other lattice. A (dual-)isomorphic mapping whose domain is identical to its range is called a (*dual*-) *automorphism* [1].

We find that the existence of a dual-automorphism in $\mathbb{L}$ can establish an isomorphism between the already defined lattices $\mathbb{PL}_1$ and $\mathbb{PL}_2$. To prove this statement, assume that there exists indeed a dual-automorphism in the complete lattice $\mathbb{L}$, that is an one-to-one correspondence $\theta(.)$: $\mathbb{L}\to\mathbb{L}$ such that $x\leq y \Rightarrow \theta(x)\geq\theta(y)$ $x,y\in\mathbb{L}$. 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 $\mathbb{PL}_1$. Further, $(a,b)\leq(c,d)$ in $\mathbb{PL}_1$ implies $a\leq c$ and $b\leq d \Rightarrow \theta(a)\geq\theta(c)$ and $b\leq d \Rightarrow$ $[\theta(a),b]\leq[\theta(c),d]$ in $\mathbb{PL}_2$. In conclusion there is a bijection (an one-to-one correspondence) between $\mathbb{PL}_2$ and $\mathbb{PL}_1$ which is order preserving. Therefore, in this case, we say that the lattices $\mathbb{PL}_2$ and $\mathbb{PL}_1$ are *isomorphic to each other under q(.)*.

Note that the dual-automorphism $\theta(.)$ in $\mathbb{L}$ implies $\theta(O)=I$ and $\theta(I)=O$, and the least element (O,O) of $\mathbb{LP}_1$ maps into the least element $[\theta(O),O]=[I,O]$ of $\mathbb{LP}_2$. Likewise, the greatest element (I,I) of $\mathbb{LP}_1$ maps into the greatest element $[\theta(I),I]=[O,I]$ of $\mathbb{LP}_2$. Note in addition that a dual-automorphism $\psi(.)$, other than $\theta(.)$, in $\mathbb{L}$ establishes an isomorphism between $\mathbb{PL}_1$ and $\mathbb{PL}_2$ different than the one under $\theta(.)$. In fact, a dual-automorphism in $\mathbb{L}$ is only a sufficient and not a necessary condition to establish an isomorphism between lattices $\mathbb{PL}_1$ and $\mathbb{PL}_2$. Nevertheless, the task of identifying a dual-automorphism $\theta(.)$ in a lattice $\mathbb{L}$ systematizes the definition of an isomorphism between the lattices $\mathbb{PL}_1$ and $\mathbb{PL}_2$.

2.3 Lattice intervals and the "inclusion test"

Let us formally define crisp sets of lattice $\mathbb{L}$ elements. We will start by dealing with the closed interval $[a,b]$ which is the set of all the elements $x\in\mathbb{L}$ such that $a\leq x\leq b$. The set $\mathbb{V}_L$ of all closed intervals, or simply *intervals* $t=[a,b]$, $a\leq b$ of lattice $\mathbb{L}$ elements augmented by the least element $[I,O]$ of $\mathbb{PL}_2$ is defined by

$$\mathbb{V}_L = \{[a,b] : a,b\in\mathbb{L} \text{ and } a\leq b \} \cup \{[I,O]\}$$

It can be shown that $\mathbb{V}_L\subset\mathbb{PL}_2$ is a complete lattice in its own right by virtue of theorem I.6 in [1]. Note that lattice $\mathbb{V}_L$ topes the hierarchy of lattices that may be implied by the FL-framework. A single element x of the original lattice $\mathbb{L}$ corresponds to $[x,x]$ in $\mathbb{V}_L$ and the latter is called an *atom*. In the complete lattice $\mathbb{V}_L$ the implied binary relation $\leq$ of partial ordering, as in $[a,b]\leq[c,d]$ $\Leftrightarrow$ $c\leq a\leq b\leq d$, signifies the fact that $[c,d]$ contains at least the elements of $[a,b]$. The following definition suggests a "measure of the magnitude" of an interval in $\mathbb{V}_L$ with regards to a positive valuation $v(.)$ defined on $\mathbb{L}$.

Definition 2

Let $\mathbb{L}$ be a complete lattice with O and I its least and greatest elements respectively, let $v(.)$ be a positive valuation on $\mathbb{L}$, and $[a,b]$ be an interval in $\mathbb{V}_L$. Then the *size* of $[a,b]$ with respect to $v(.)$ is defined by the non-negative function $Z([a,b])=v(b)-v(a)$ (it is tacitly assumed that the size of the least element $[I,O]$ in $\mathbb{V}_L$ is zero, that is $Z([I,O])=0$).

□

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

$$a\vee c\leq b\wedge d \text{ (true)} \Rightarrow [a,b]\wedge[c,d] = [a\vee c,b\wedge d]$$
$$a\vee c\leq b\wedge d \text{ (false)} \Rightarrow [a,b]\wedge[c,d] = [I,O]$$

Finally note that an implication of the existence of a dual-automorphism $\theta(.)$ in $\mathbb{L}$ is that a distance between two intervals of $\mathbb{V}_L\subset\mathbb{PL}_2$ can be determined. Moreover the distance between any two generalized intervals of $\mathbb{PL}_2$, say [a,b] and [c,d], is taken to be equal to the distance of their isomorphic images in $\mathbb{PL}_1$. That is

$$d([a,b],[c,d]) := d((\theta(a),b),(\theta(c),d)) = V((\theta(a),b)\vee(\theta(c),d)) - V((\theta(a),b)\wedge(\theta(c),d)) =$$
$$= V(\theta(a)\vee\theta(c),b\vee d) - V(\theta(a)\wedge\theta(c),b\wedge d) = [v(\theta(a)\vee\theta(c)) - v(\theta(a)\wedge\theta(c))] + [v(b\vee d)-v(b\wedge d)] =$$
$$= [v(\theta(a))+v(\theta(c))] + [v(b)+v(d)] = [v(\theta(a))+v(b)] + [v(\theta(c))+v(d)] = d(\theta(a),b) + d(\theta(c),d).$$

It has already been explained that a positive valuation function occasions the definition of a distance function in $\mathbb{L}$. Underneath is shown that such a positive valuation furthermore occasions the definition of an inclusion measure in the lattice in question.

2.4 Inclusion measure

Definition 3

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

(C1) $\sigma(x\leq O) = 0$, $x\neq O$.

(C2) $\sigma(x\leq I) = 1$, $\forall x\in\mathbb{L}$.

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

It can be argued that $\sigma(x\leq u)$ indicates the degree of truth of the lattice inclusion relation $x\leq u$, or in other words $\sigma(x\leq u)$ indicates the degree of inclusion of x in u. Note that when the complete lattice $\mathbb{L}$ is the lattice $\mathbb{V}_L$ of intervals of some complete lattice, whether it be a constituent or a product lattice, then the Consistency Property above states that if an interval u is included in another interval w, then any interval x is included in u less than it is in w. Note also in definition 3 that an inclusion measure $\sigma(x\leq u)$ defines two "antipodal" fuzzy sets in $\mathbb{L}$, each one resulting in when one of the x, u is kept constant while the other runs the whole of lattice $\mathbb{L}$. Finally, it should be noted that the definition of the inclusion measure above has been inspired from the definition of the *possibility measure* [14, 15] and adapted to a lattice context.

[**] Here $\mathbb{L}$ can be either a product lattice or a constituent lattice.

Consider the function $k(x,y) = \frac{v(y)}{v(x \vee y)}$, where $x,y \in \mathbb{L}$ and v(.) is a positive valuation on lattice $\mathbb{L}$ with v(O)=0. Note that if for the positive valuation v(.) it is v(O)≠0, then a positive valuation $v^+(.)$ with $v^+(.)=0$ can always be defined out of v(.) by simply subtracting v(O) from all v(x), where $x \in \mathbb{L}$. In the sequel it is tacitly assumed for all positive valuations v(.) considered that v(O)=0. It can be shown that the function k(x≤u)=k(x,y) defines an inclusion measure in $\mathbb{L}$.

Theorem 2

Let $\mathbb{L}$ be a complete lattice. Then the existence of a positive valuation function v(.) in $\mathbb{L}$ (with v(O)=0) is a sufficient condition for the function $k(x \leq u) = \frac{v(u)}{v(x \vee u)}$ to be an inclusion measure in $\mathbb{L}$.

□

The proof of theorem 2 is given in the Appendix.

Note that the function k(x≤y), called also *characteristic function*, is equal to 1 if and only if it is x≤y. Note that k(x≤y) defines a fuzzy set for each lattice element y; moreover it defines another fuzzy set for each lattice element x. The highest possible degree of membership is 1 (full membership in y) attained for those $x \in \mathbb{L}$ for which it holds x≤y; the set of all such x(s) is called the *core* of the fuzzy set defined by y [13].

Theorem 2 entails some further reaching implications when applied in lattice $\mathbb{PL}_1$. Recall that when v(.) is a positive valuation in $\mathbb{L}$ then so is V((a,b)) = v(a)+v(b) in $\mathbb{PL}_1$. Hence, the function k(x≤y), $x,y \in \mathbb{PL}_1$, defines an inclusion measure in $\mathbb{PL}_1$. Furthermore recall that the existence of a dual-automorphism θ(.) in $\mathbb{L}$ suffices to establish an isomorphism between lattices $\mathbb{PL}_1$ and $\mathbb{PL}_2$. Therefore, the existence of both a positive valuation v(.) and a dual-automorphism θ(.) in a complete lattice $\mathbb{L}$, is a sufficient condition to define an inclusion measure in the lattice $\mathbb{V}_L$ of closed intervals of $\mathbb{L}$ elements. Note that the core [13] of the fuzzy set defined by y=[a,b] and with characteristic function k(x≤y), $x,y \in \mathbb{V}_L$, (that is, the collection of intervals x=[c,d] such that σ(x=[c,d]≤y=[a,b]) = 1) includes all the atoms [e,e] such that $e \in \mathbb{L}$ and a≤e≤b. To calculate the inclusion measure σ(x≤u) of a lattice interval x into another lattice interval u, first we map both x and u where $x,u \in \mathbb{V}_L \subset \mathbb{PL}_2$ to their isomorphic x′ and u′, respectively, where $x',u' \in \mathbb{PL}_1$. Finally, the inclusion measure σ(x′≤u′)= k(x′≤u′) is employed as the inclusion measure σ(x≤u) for all intervals $x,u \in \mathbb{V}_L$.

Note that the characteristic function k(x≤u) always provides a non-zero measure of inclusion of an interval x inside interval u, where u≠O. A non-zero inclusion measure is important when it is attempted to assign an interval x to one of the non-overlapping with x intervals u and w, as happens frequently in the FLN scheme.

These notions have led us to a connectionist scheme, namely fuzzy lattice neurocomputing (FLN), where learning occurs by defining and enhancing crisp lattice intervals which are in fact the cores of fuzzy sets in $\mathbb{V}_L$. In this sense, the FLN is a generalization of the min-max neural networks [11, 12] which have emanated likewise from the adaptive resonance theory (ART) [2]. A detailed comparison of the FLN on one hand with both the ART and the min-max neural networks on the other hand, will be given elsewhere, for lack of space.

3 Description of the Fuzzy Lattice Neurocomputing (FLN) Scheme

The FLN deals with elements of $\mathbb{V}_L$ which are called *intervals*, *codes*, or *nodes*. The elements of the set $\mathbb{V}_L$, by being closed intervals, define sets of akin, in the sense of "adjacent", lattice $\mathbb{L}$ elements. Intelligent schemes can be crafted in $\mathbb{V}_L$ for defining sets of lattice elements which can be represented by the set-union of $\mathbb{V}_L$ intervals. Furthermore the existence of a positive valuation in $\mathbb{L}$ was shown to enable the definition of an inclusion measure for lattice $\mathbb{PL}_1$, and hence this lattice becomes tractable for mathematical operations. Finally, the existence of a dual-automorphism in $\mathbb{L}$ "bridges the gap" between lattice $\mathbb{PL}_2 \supset \mathbb{V}_L$ and lattice $\mathbb{PL}_1$ by establishing an isomorphism between $\mathbb{PL}_1$ and $\mathbb{PL}_2$. A concrete connectionist scheme for a crisp set identification will be shown in this section, that is the fuzzy lattice neurocomputing (FLN) scheme.

The FLN operates similarly to an adaptive resonance theory (ART) model [2, 3]. In fact, a neural implementation of the FLN for Euclidean space applications resembles that of the two-layer ART [2]. In addition, the FLN proposes the following enhancements. Instead of dealing exclusively with Euclidean vectors that define hyperrectangles [3], the FLN deals instead with codes in the complete lattice $\mathbb{V}_L$. It is important to note that even when only the Euclidean domain is considered, the FLN is superior to an ART-based model in two points. The first point is that an input to the FLN might be an N-dimensional hyperrectangle instead of only an N-dimensional point. Such a capacity is due to the nature of the lattice-applicable activation function $\sigma(x \le w)$ employed by the FLN's upper layer neurons, where $\sigma(.)$ is an inclusion measure, x is an input code to the FLN, and w is a learned code. The other point is the technique of *maximal expansions* which enhances the overlapping of two codes by enlarging their *lattice meet* to the maximum and in all possible ways. A detailed description of the latter technique is given below. Therefore, it can be claimed that a traditional ART-based neurocomputing scheme becomes a domain-specific variation of the more general FLN scheme.

Let $\{w_{k,i}\}$ be a family of intervals and c_k be a class defined by the union of the family $\{w_{k,i}\}$, as explained in [6]; that is, $c_k = \bigcup_i w_{k,i}$. We say that a trivial interval or point, [a,a] is inside a class $c_k = \bigcup_i w_{k,i}$ if and only if there is an interval $w_{k,i}$ for some value of the index i such that $[a,a] \le w_{k,i}$. Moreover if x is a single interval, then the degree of inclusion of x in c_k is defined to be $\sigma(x \le c_k) = \sigma(x \le \bigcup_i w_{k,i}) := \max_i \sigma(x \le w_{k,i})$. Note that sometimes $\sigma(x \le \bigcup_i w_{k,i})$ will be denoted instead by $\sigma(x \le \{w_{k,i}\})$. Furthermore, when $\sigma(x \le c_k)=1$ we will write $x \le c_k$ and say that "x is in(side) the class c_k".

Definition 4

A family $c=\{w_i\}$ of lattice intervals is called *connected* if and only if, for any two intervals p,q inside $c = \bigcup_i w_i$ there exists a sequence of intervals $t_0, \ldots, t_{N-1}$ solely containing points of $\{w_i\}$ 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,…,N-2. In other words, a family $\{w_i\}$ is connected if and only if there exists a path of intervals between any two intervals of $\{w_i\}$ where the path lies exclusively in $\{w_i\}$.

The decision making in the fuzzy lattice neurocomputing (FLN) scheme is driven by the degree of true of the proposition $x \le c_k$, where x is a lattice interval that excites the system and the c_k's are learned classes stored in the system's memory. Note that any such a family $\{w_{k,i}\}$ is *simplified* in the sense that there are no two *constituent intervals* $w_{k,m}$, $w_{k,n}$ in $\{w_{k,i}\}$ such that $w_{k,m} \le w_{k,n}$.

Let F be the collection of all families that can represent a class c in lattice $\mathbb{L}$. Note that the cardinality of a family $\{w_i\}$ of intervals used to represent class c need not to be a 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 simplified family $\{p_m\}$ is "strictly smaller" than another simplified 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 $\{q_n\}$. Note that F is a partially ordered set. The following lemma holds,

Lemma 2

A collection F of families that can represent a class c has a maximum element, namely *quotient* 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 "pinnacle" in the hierarchy of lattices.

The technique of maximal expansions calculates the maximum of the set F, that is the quotient $Q(F)$. A computational method towards this end is algorithm-Q shown in the proof of Lemma 2. There is a benefit in the context of fuzzy lattice neurocomputing (FLN) for replacing a family $\{w_i\}$ which represents a class c by the quotient $Q(\{w_i\}) = Q(F)$. The benefit stems from the fact that a lattice inclusion measure $\sigma(x \le c)$ is used by the FLN scheme as an activation function of its upper layer neurons. Because $Q(F)$ is the maximum family in F, it may be inferred that, for any constituent interval w in a family $\{w_i\}$, there exists an interval W in $Q(F)$ such that $w \le W$. Hence, for any input interval x, it is $\sigma(x \le Q(F)) = \max \sigma(x \le \{w_j\})$, 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 $c = \bigcup_i w_i$.

The Fuzzy LatticeNeurocomputing (FLN) Scheme

Figure 1 shows the two layer FLN architecture which can be employed for learning and decision making in the FL-framework which includes the set R^N. Notice the resemblance with the ART [2] neural network architecture. A key difference with the ART, besides FLN's own activation function $\sigma(x \le w)$, is its applicability to fuzzy lattices. This could imply significant technological benefits as explained in section 4. Another advantage of the FLN compared to the ART is that an input to the FLN might be an interval, that is, a hyperbox in the set R^N.

Figure 2 shows a flow-chart of the processing cycle of the fuzzy lattice neurocomputing (FLN) for learning and decision making. The processing cycle of Figure 2 can be implemented by the FLN architecture of Figure 1. The flow-chart blocks in Figure 2 are explained below.

0. The first input is memorized. From then on learning and decision making proceeds as follows (assume that at any instant, there are M known and learned classes - that is M families of intervals c_k, k=1,…,M stored in the memory).
1. Present an input interval x to the initially "set" database of classes $c_1,\ldots,c_M$.
2. Calculate $\sigma(x\leq c_k)$ for all c_k, k=1,…M that have not yet been "reset", where $c_k=\bigcup_i w_{k,i}$ and

   $\sigma(x\leq c_k) = \sigma(x\leq \bigcup_i w_{k,i}) := \max_i \sigma(x\leq w_{k,i})$
3. Competition among the classes c_k : select c_J such that $\sigma(x\leq c_J) = \max_k \sigma(x\leq c_k)$, where J is the index of the corresponding winner and $c_J=\bigcup_i w_{J,i}$.
4. The maximum-size-test (*assimilation condition* or *matching test*) : Test to see whether the size $x\vee w$ is less than a critical threshold Z, where w corresponds to the $\max_i \sigma(v\leq w_{J,i})$.
5. If the maximum-size-test is successful, then incorporate x into c_J by replacing w by $x\vee w$ and calculate the new quotient $Q(\{w_{k,i}\})$.
6. If the maximum-size-test fails, then "reset" c_J.
7. Completion-test : Test whether all the classes $c_1,\ldots,c_M$ have been "reset". If the completion-test fails go to step 2 to look for another winner.
8. If the completion-test is successful, then store x : c_{M+1}=x.

Note that step 8 assumes that there exists sufficient system memory available. If no more memory is available, then the accommodation of the input x is turned down. In all, the FLN can be regarded as a competitive scheme that handles and manipulates fuzzy sets.

The specific distinction between the FLN and an ART-based neural network which (distinction) contributes to an overwhelming utility of the FLN emanates from its applicability to lattices other than the Euclidean space. Finally note that the FLN is inherently hierarchical due to a mathematical property which states that the product of lattices is lattice [1], therefore the FLN promotes data fusion.

4 Experiments

In most of the experiments, conventional N-dimensional vectors were used because the employment of such data for learning and decision making is widespread in the pattern recognition community. Hence the corresponding performance of the FLN can be compared with the performance of other clustering algorithms. One simple example is also cited beyond the Euclidean space.

The set R^N is a lattice and it may be regarded as the product of N lattices (continuous chains). An element of the lattice R^N is an N-dimensional point, whereas an interval in lattice R^N is an N-dimensional hyperrectangle also called a *hyperbox* herein. More specifically, the complete lattice employed was the unit hypercube in R^N with least element O=(0,…,0) and greatest element

I=(1,…,1). The corresponding N complete lattice chains are simply the unit intervals [0,1]. Furthermore for each one of the N lattice chains, the positive valuation used was the value x of the corresponding vector entry; that is $v_i(x)=x$, i=1,…,N. Recall that by Lemma 1 a positive valuation in the product N-dimensional lattice R^N can be defined by the sum of the positive valuations of its N constituent lattice chains. Note also that for each complete lattice chain [0,1] there exists a dual-automorphic function θ(.) given by θ(x)=1-x. Hence in line with the theoretical analysis of section 2, an inclusion measure is induced in the lattice V_L of hyperboxes inside the unit hypercube in R^N.

A hyperbox specified by its N edges $[x_{2i},x_{2i+1}]$, i=0,…,N-1, is represented conventionally in the 2N-dimensional space by the vector $x=(1-x_0, x_1, \ldots, 1-x_{2N-2},x_{2N-1})$ with a positive valuation $v(x)= 1-x_0 + x_1 + \ldots + 1-x_{2N-2} + x_{2N-1}$. In all of the following examples, the learning was always stable and required only a few passes through the data.

4.1 Example-1

The two dimensional data set shown in Figure 3(a) consisting of 112 points was synthesized to test the ability of the FLN to separate non-linearly separable data sets. The order of data presentation was from left to right and from bottom up starting with the cluster above and proceeding to the one below. The product 2-D lattice size Z was 0.0202. The FLN placed the data into the clusters shown in Figure 3(b) using 78 nodes, assigning 39 nodes to each cluster. Note that a node on the plane in Figure 3(b) is a rectangle and often a trivial one, that is a line segment.

4.2 Example 2

The two-dimensional data set shown in Figure 4(a) and consisting of 84 points was constructed to test the ability of the FLN to identify a non-convex data set. Starting from the top-left corner, the order of the data presentation was to the right and downwards. The product 2-D lattice size Z was set to 0.0408. The neural network placed all the data into one cluster consisting of 10 overlapping nodes as shown in Figure 4(b). The boxes marked by an × are not inside any node (rectangle). Note that the interior of the identified 2-D cluster shown in Figure 4(b) does <u>not</u> belong to the cluster itself.

4.3 Example 3

This time a benchmark data set was processed. The Fisher IRIS data set [4] was chosen because its familiarity to the scientific community may allow an assessment of the relative performance.

The data were presented in their original order, that is, as cited in [4] and with a product 4-D lattice size of 1.71. As a result, 15 classes were formed; each class is defined by the set-union of 4-D hyperrectangles (codes or nodes). In the sequel, a second level clustering was performed by determining the degree of inclusion of each class within all the others. Such a degree of inclusion of a class into another one was determined by finding the maximum value for the inclusion measure of a first class' node to a second class' node. Hence, the square matrix shown in Table 1 resulted where only the two most significant decimals are shown. The second level clustering is as follows : each first level class may be associated with at most one other class, that class with the largest inclusion measure, suffices this maximum inclusion measure is over a threshold. In our case, the threshold was set to 0.60. Table 1 shows that there is indeed a maximum inclusion value above the threshold for every first level class which is not on the main diagonal. The main diagonal is of no practical interest because the values along it are always equal to 1 demonstrating the trivial fact that any class contains itself.

Figure 5 shows how the association of first level classes implies a partition into three classes. Raising the threshold value above 0.64 causes more than three classes to appear in the second level of clustering, while lowering the threshold to less than 0.57 causes fewer than three classes to appear. In conclusion, without the presence of a teacher and using only FLN clustering, the Fisher's IRIS data were clustered in three classes with only three data misclassifactions, as summarized in Table 2.

Table 3 shows the best results obtained using the min-max neural network [12]. Besides FLN's superior classification results shown in Table 2, it is also important to note that the FLN tracks down three classes after two consecutive stages of clustering, whereas the min-max neural network identifies a total of 14 clusters in one stage of clustering which are then assigned to the 3 IRIS classes by an external teacher. In conclusion, the FLN outperforms the min-max neural network in this clustering example.

More recent classification results of the IRIS benchmark are reported in [7] by different supervising neural networks including backpropagation and structural learning with forgetting (SLF). A statistical overall optimum of 5.0 errors is reported for the SLF, occurring when the training set contains either 30 or 90 randomly selected data; the rest of the data constitute the testing set. The corresponding optimal performance for the backpropagation results in an average of 4.8 errors. Note again that the superior discriminatory capacity of the FLN compared with both the backpropagation and the SLF, is further accentuated by the fact that the FLN identified the three underlying IRIS classes without the help of an external teacher.

It should also be mentioned that additional benchmark data sets have been processed by the FLN. For example the SONAR and the VOWEL data sets, obtained from the Carnegie Mellon University's collection of neural net benchmarks have been processed. And in both cases the classification results by the FLN were better than those reported in the aforementioned collection of benchmarks. However, due to the lack of space here, these results will be reported elsewhere.

4.4 Example-4

A simple example is cited below to illustrate the applicability of the FLN beyond the Euclidean space. We will consider the collection of all fuzzy sets specified by (1) trapezoidal membership functions of height = 1, and (2) non-zero area, over an inteval [a,b]. This collection of trapezoids is a lattice denoted by $\mathbf{T}$. Specifically if $t_1,t_2 \in \mathbf{T}$ then the lattice-join $t_1 \vee t_2$ is the smallest trapezoid that includes both t_1 and t_2, whereas the lattice-meet $t_1 \wedge t_2$ is the largest trapezoid included in both t_1 and t_2, that is, the intersection of t_1 and t_2. When the intersection of t_1 and t_2 is either the empty set or it is not a trapezoid with the given specifications, then $t_1 \wedge t_2$ is defined to be the least element O of the lattice in question. On the other hand the membership function of the largest trapezoid I equals 1 everywhere on [a,b]. A function $\tau(.)$ was defined on the lattice $\mathbf{T}$ that maps a trapezoid to its area. The function $\tau(.)$ satisfies the conditions (R1)-(R3) of the "Remark" in the proof of Theorem 2, therefore $k(x \leq u) = \dfrac{t(u)}{t(x \vee u)}$ is an inclusion measure in T. Another feature of lattice $\mathbf{T}$ is that a dual-automorphism $\theta(.)$ was not required to apply the FLN in $\mathbf{T}$, because a $t \in \mathbf{T}$

with non-zero area implies all the fuzzy sets $x \in \mathbb{T}$: $x \leq t$. That is a fuzzy set t implicitly implies the interval [O,t]. Therefore the technique of maximal expansions is still applicable. Due to both the absence of a positive valuation in T and to the consideration of a $t \in \mathbb{T}$ as the interval [O,t], the "size" of a fuzzy set t, required in step-4 of the FLN scheme, was specified by the value $t(t)$.

As a concrete numerical example consider the fuzzy "lattice of weights", denoted by $\mathbb{T}_W$, which was defined over the interval [0, 22 N] (Fig.6). Assume the linguistic variables "light (weight)" and "heavy (weight)" defined by two different experts as shown in Figure 6(a). The problem is to assign sensibly an object x with a triangular membership function (Fig.6(b)) to one of the classes "light" or "heavy".

The FLN addressed this problem. In particular, learning occurred by combining the definitions given by the two experts and hence the FLN formed one "light" class and one "heavy" class as shown in Figure 6(b). For the data values of Figure 6(b) the degrees of inclusion of x in the classes "light" and "heavy" were calculated to be $\sigma(x \leq \text{"light"}) = \dfrac{t(\text{"light"})}{t(x \vee \text{"light"})} = \dfrac{4}{7} \cong 0.57$, and $\sigma(x \leq \text{"heavy"}) = \dfrac{t(\text{"heavy"})}{t(x \vee \text{"heavy"})} = \dfrac{4}{12} \cong 0.33$. Therefore according to the FLN, the belief that the object x is "light" is approximately twice as large as the belief of being "heavy".

It is important to note that additional lattices could have been considered jointly, for example the "lattice of lengths", or the "lattice of colors". In such as case instead of specifying an object by the 3-D Euclidean vector (weight, length, color) we could specify it by a triple of fuzzy sets. Consequently the FLN can be applied in principle.

5 Discussion

Regarding the examples presented in Subsections 4.1 through 4.3 it should be noted that those examples only demonstrate the capacity of the FLN for pattern discrimination and recognition in the set R^N and when the inputs to the FLN are simply points. The previous examples do not demonstrate FLN's unique ability to handle N-dimensional hyperboxes, or intervals of lattice elements in general. From a practical standpoint such a feature could be quite advantageous in cases where need to compensate for the uncertainty of the measurements by feeding to the learning and decision making system a neighborhood of values as defined by an N-dimensional interval rather than as defined by a precise single point value. Moreover the previous examples do not demonstrate the one quality which is fundamentally unique to the FLN, that is, its capacity to handle elements in a lattice. Note that an input to the FLN can be a lattice element, for instance an image or its Fourier transform, a waveform or its spectrum, a fuzzy set, a real number, or an element of another lattice as well as any combination of these inputs. The later implies the FLN's capacity for fusing disparate sensory data.

In subsection 4.4 a simple example demonstrated the applicability of the FLN beyond the Euclidean space in one lattice of fuzzy sets. Certain sufficient conditions had to be relaxed, such as the existence of a positive valuation. Nevertheless the FLN was still applicable in principle as well as the technique of maximal expansions.

It should be noted that an efficient hardware implementation of the FLN, as shown in Figure 1, in order to live up to its theoretical expectations it should address efficiently the "representation problem". That is the question of how a lattice element is stored and processed by the FLN's architecture. Selecting a digital computer implementation implies the employment of the set R^N for the representation of a lattice element. The amount of computation required for the calculation of the lattice join ($\vee$) and meet ($\wedge$) operations depends on N. Such a way of processing may be quite time-demanding especially in cases of a large N value (e.g. images). On the other hand, the representation problem could be relieved if there exists a way to represent one lattice element, for example, an image or a fuzzy set, as a whole object in an analog form. Then assuming the capacity for an analog calculation of the lattice join ($\vee$) and meet ($\wedge$) operations between whole lattice elements, we could achieve a real efficient hardware implementation of the FLN scheme.

6 Conclusion

The fuzzy lattice neurocomputing (FLN), a novel knowledge acquisition and decision making connectionist scheme, was introduced. Moreover a novel theoretical perspective to machine learning and decision making was introduced and new theoretical results were shown in this paper. The wide domain of the FLN is due to its applicability to lattices. Neurocomputing paradigms like the fuzzy adaptive resonance theory (fuzzy ART) and the min-max neural networks may be described as domain-specific variations of the more general FLN scheme. In addition, the fuzzy degree of inclusion of a lattice interval into another one may be calculated by a novel tool-concept, namely "inclusion measure". However, for the FLN, the order of data presentation is again significant as it is the case in an ART model (fuzzy or not) and in the min-max neural networks.

In this paper the FLN was employed for clustering in pattern recognition problems primarily in the conventional set R^N on synthetic data sets and the Fisher IRIS benchmark. Its outstanding capacity for learning and recognition was illustrated. An additional example demonstrated the applicability of the FLN beyond the Euclidean space in one lattice of fuzzy sets. The true potential of FLN lies, we believe, in its ability to be applied in different lattices, to consolidate hierarchically lattices of disparate objects, and to treat them with mathematical consistency in order to attain a sophisticated decision making capacity.

References

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

[2] G. Carpenter, and S. Grossberg,"A massively parallel architecture for self-organizing neural pattern recognition machine", *Computer Vision, Graphics and Image Understanding*, 37: 54-115, 1987.

[3] G. Carpenter, S. Grossberg, and D. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system", *Neural Networks*, 4: 759-771, 1991.

[4] R. Fisher, "The use of Multiple Measurements in Taxonomic Problems", *Annals of Eugenics*, 7(II): 179-188, 1936.

[5] J.A. Goguen , "L-Fuzzy Sets", *Journal of Mathematical Analysis and Applications*, 18: 145-174, 1967.

[6] P.R. Halmos, *Naive Set Theory*, Van Nostrand, New York, 1960.

[7] M. Ishikawa, "Structural Learning with Forgetting", *Neural Networks*, 9(3): 509-521, 1996.

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

[9] V.G. Kaburlasos, and V. Petridis, "Fuzzy Lattice Neurocomputing (FLN)", *Proc. ISCA 5th Intl Conf. on Intelligent Systems*, Reno, Nevada, June 19-21, pp. 56-60, 1996.
[10] V. Petridis, and V.G. Kaburlasos, "FLN: A Fuzzy Lattice Neurocomputing Scheme for Clustering ", *Proc. WCNN*, San Diego, California, September 15-18, 1996.
[11] P.K. Simpson, "Fuzzy Min-Max Neural Networks - Part1: Classification", *IEEE Transactions on Neural Networks*, 3(5): 776-786, 1992.
[12] P.K. Simpson, "Fuzzy Min-Max Neural Networks - Part2: Clustering", *IEEE Trans. on Fuzzy Systems*, 1(1): 32-45, 1993.
[13] L.A. Zadeh, "Fuzzy Sets", *Information and Control*, 8: 338-353, 1965.
[14] L.A. Zadeh, "Fuzzy Sets as a Basis for a Theory of Possibility", *Fuzzy Sets and Systems*, 1: 3-28, 1978.
[15] H.J. Zimmermann, *Fuzzy Set Theory and Its Applications*, Kluwer Academic Publishers, Norwell, MA, 1991.

Appendix

Proof of Theorem 1

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

Moreover, note that $\mathbb{PL}_2$ is a complete lattice because $\mathbb{L}$ is a complete lattice.

Proof of Theorem 2

1. $k(x\le O) = \dfrac{v(O)}{v(x\vee O)} = \dfrac{v(O)}{v(x)} = 0$, assuming $x\ne O$

   Note that this condition is true if v(.) is a positive, and not only an isotone, valuation with v(O)=0.

2. $k(x\le I) = \dfrac{v(I)}{v(x\vee I)} = \dfrac{v(I)}{v(I)} = 1$

3. In any lattice the operation of join is isotone [1], that is, $u\le w \Rightarrow x\vee u \le x\vee w$.

   Provided a positive valuation v(.) in a lattice $\mathbb{L}$. A distance function is defined in $\mathbb{L}$ by $d(u,w) = v(u\vee w) - v(u\wedge w)$ [1]. Noting the isotone relation above, implies both $d(x\vee u, x\vee w) = v(x\vee w) - v(x\vee u) \ge 0$ and $d(u,w) = v(w) - v(u) \ge 0$.

It is further known [1] that if v(.) is an isotone valuation in $\mathbb{L}$ then $d(x\vee u,x\vee w) + d(x\wedge u,x\wedge w) \leq d(u,w)$. Recall that a positive valuation is an isotone valuation. Hence, $d(x\vee u,x\vee w) \leq d(x\vee u,x\vee w)+d(x\wedge u,x\wedge w) \leq d(u,w) \Rightarrow v(x\vee w)-v(x\vee u) \leq v(w)-v(u) \Rightarrow v(x\vee w) \leq v(w) - v(u) + v(x\vee u)$. Assuming $w\neq O$ it follows that

$$\frac{v(u)}{v(w)}v(x\vee w) \leq \frac{v(u)}{v(w)}[v(w) - v(u) + v(x\vee u)] = \frac{v(w)-v(u)}{v(w)}v(u) + \frac{v(u)}{v(w)}v(x\vee u) \leq \frac{v(w)-v(u)}{v(w)}v(x\vee u) + \frac{v(u)}{v(w)}v(x\vee u) = v(x\vee u) \Rightarrow \frac{v(u)}{v(x\vee u)} \leq \frac{v(w)}{v(x\vee w)} \Rightarrow$$

$k(x\leq u) \leq k(x\leq w)$.

Remark. The existence of a positive valuation v(.) is far too restrictive a necessary condition for the function $k(x\leq u)= \frac{v(u)}{v(x\vee u)}$ to be an inclusion measure in a complete lattice $\mathbb{L}$. In effect, a positive valuation can be replaced by another real-valued function $\tau(.)$ that satisfies the following three more relaxed conditions,

(R1) $\tau(O)= 0$,

(R2) $x<y \Rightarrow \tau(x)<\tau(y)$, $x,y\in\mathbb{L}$, and

(R3) $u\leq w \Rightarrow \tau(x\vee w)-\tau(x\vee u)\leq\tau(w)-\tau(u)$

It can be easily shown by following the previous three steps in the proof of Theorem 2 that when $\tau(.)$ satisfies (R1) through (R3), then the function $k(x\leq u)= \frac{t(u)}{t(x\vee u)}$ satisfies all the conditions (C1)-(C3) of definition 3, therefore $k(x\leq u)$ is an inclusion measure and the FLN is applicable. We make use of an inclusion measure of the latter form in the last example of section 4.

Regarding a positive valuation in a lattice $\mathbb{L}$, note that a significant implication of its existence is the existence of a metric d(.,.) in $\mathbb{L}$ which could be used for learning and decision making. Because no metric is required by the FLN, a positive valuation in a lattice is too restrictive a necessary condition for the existence of an inclusion measure in the lattice in question.

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 set F of families such that $\{w_i\}<\{q_i\}$. We will delineate a method for constructing an ever (strictly) larger maximal expansion of a family. This construction is shown to terminate in a finite number of steps and a global maximum will have been reached, that is the quotient $Q(F) = Q(\{w_i\})$.

The truth of Lemma 2 will be shown in case the family $\{w_i\}$ contains exactly two connected constituent intervals, say w_1 and w_2. To construct maximal expansions of $\{w_1,w_2\}$, 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,…,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 specifying the maximum interval “$\max(w_{11}\wedge w_{21})$” which contains $w_{11}\wedge w_{21}$ and consists of elements of w_{11} or w_{21}. Note that the latter is a trivial problem framed within the first constituent lattice. Hence, the corresponding maximal expansion implies the interval [max($w_{11} \wedge w_{21}$),$w_{12} \wedge w_{22}$, …,$w_{1L} \wedge w_{2L}$]. In the following 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 all the maximal expansions "along" all the constituent lattices has to be the maximum element in the set F, that is, the quotient Q(F) or me(w_1,w_2)= Q(F).

To show the truth of 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,…,u_L] contains exclusive elements of w_1=[w_{11},…,w_{1L}] AND exclusive elements of w_2=[w_{21},…,w_{2L}]. This implies that at least for one constituent lattice interval u_i, i=1,…,L $w_{1i} \wedge w_{2i} < u_i$. 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 either w_1 or w_2. Because of the way the set me(w_1,w_2) was constructed, it can be inferred 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).

Now consider a third interval w_3 such that {w_3}$\cup$Q({w_1,w_2}) is connected. Assume the maximal expansions me(w_1,w_3) and me(w_2,w_3). 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 {w_1,w_2,w_3}$\cup$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_3, me(w_1,w_2)), me(w_1, me(w_2,w_3)), and me(w_2, me(w_3,w_1)). Simplifications result in 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. Note 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}) in finite time and it was proven helpful in the classification examples presented in this paper, where families of relatively small cardinality (a few tens of intervals) 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.

TABLE 1 The second level clustering is accomplished by determining, as shown in the non-symmetric square matrix below, the value of the inclusion measure of each class into all the others. The largest such number(s) for each class, which is not on the main diagonal, is underlined in each row and may associate one class with another class.

cluster	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	1.0	.67	.42	.64	.66	.38	.59	.54	.45	.36	.30	.43	.25	.27	.29
1	.67	1.0	.52	.61	.61	.45	.65	.68	.57	.43	.35	.54	.28	.29	.32
2	.54	.68	1.0	.58	.51	.71	.57	.82	.72	.66	.58	.81	.45	.40	.44
3	.64	.61	.44	1.0	.58	.40	.65	.56	.43	.36	.31	.42	.26	.24	.26
4	.93	.86	.55	.82	1.0	.50	.82	.74	.60	.48	.40	.57	.34	.36	.39
5	.53	.63	.76	.56	.50	1.0	.55	.83	.74	.74	.67	.80	.51	.46	.49
6	.85	.93	.62	.93	.83	.56	1.0	.81	.65	.53	.44	.63	.37	.36	.39
7	.76	.93	.89	.75	.76	.81	.78	1.0	.85	.72	.58	.83	.46	.44	.48
8	.65	.81	.79	.62	.61	.76	.65	.90	1.0	.71	.55	.83	.44	.44	.49
9	.50	.61	.71	.50	.48	.74	.52	.74	.70	1.0	.71	.90	.54	.52	.59
10	.40	.46	.59	.42	.38	.63	.41	.56	.51	.67	1.0	.74	.65	.55	.60
11	.61	.78	.87	.60	.58	.80	.63	.87	.85	.92	.80	1.0	.59	.53	.60
12	.25	.28	.34	.26	.24	.37	.26	.33	.31	.38	.49	.41	1.0	.68	.64
13	.27	.29	.31	.24	.26	.33	.25	.32	.31	.37	.42	.37	.68	1.0	.72
14	.41	.44	.47	.37	.39	.49	.38	.48	.47	.56	.61	.56	.90	.95	1.0

TABLE 2 FLN Processing of the IRIS benchmark.

Number of Clustering Stages = 2
Number of Classes = 3

class	1	2	3
1	100 %		
2		94 %	6 %
3			100 %

TABLE 3 Min-max neural processing of the IRIS benchm ark

Number of Clustering Stages = 1
Number of Classes = 14

class	1	2	3
1	100 %		
2		88 %	12 %
3		10 %	90 %

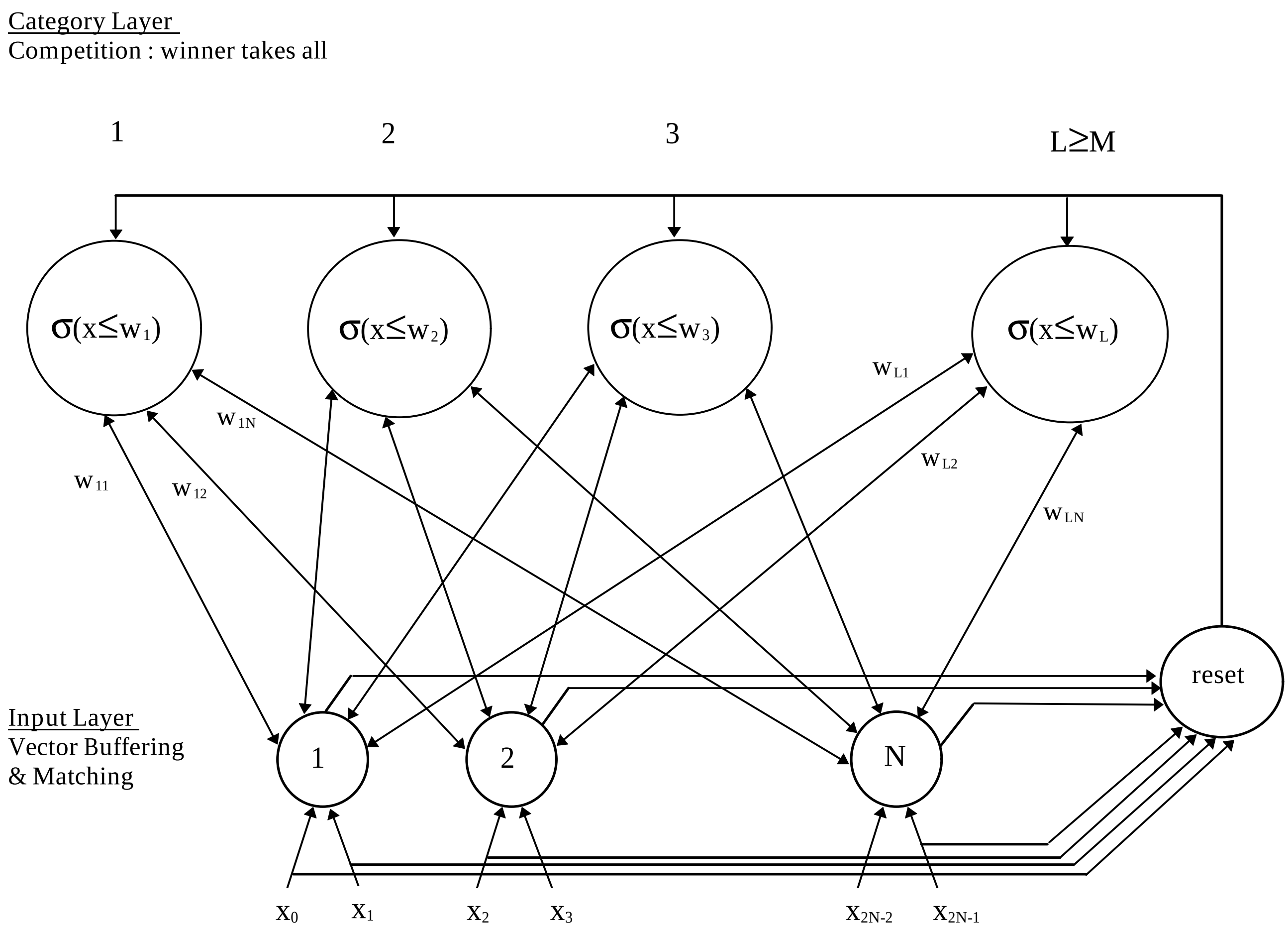


Figure 1 The two layer FLN architecture for clustering in the lattice $\mathbb{V}_L$. M is the number of learned classes. L is the number of upper layer neurons which coincides with the total number of intervals used to define the M classes. N is the number of lower layer neurons which coincides with the total number of intervals that input the FLN, where the i^{th} interval is specified by $[x_{2i-2}, x_{2i-1}]$, i=1, …,N.

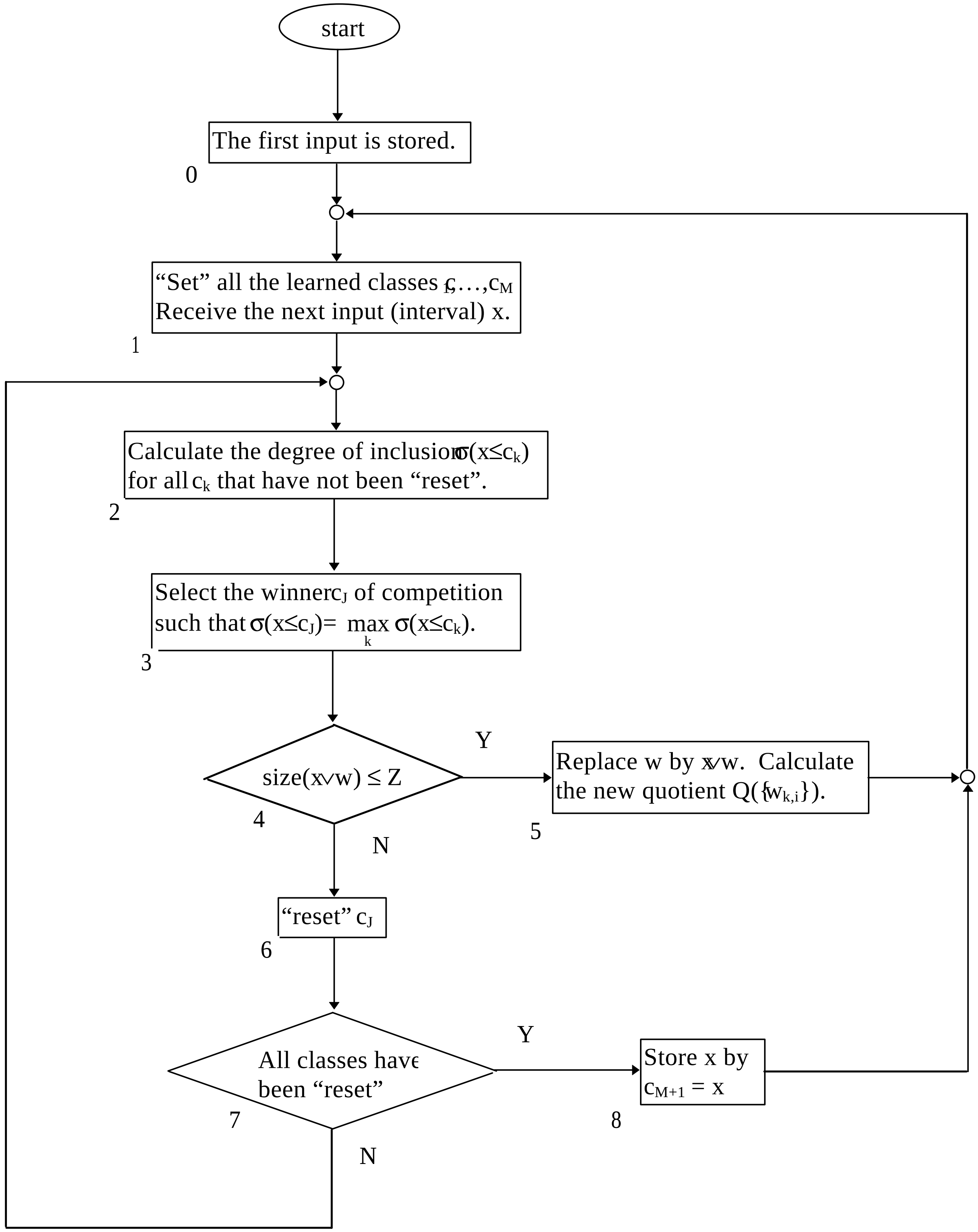
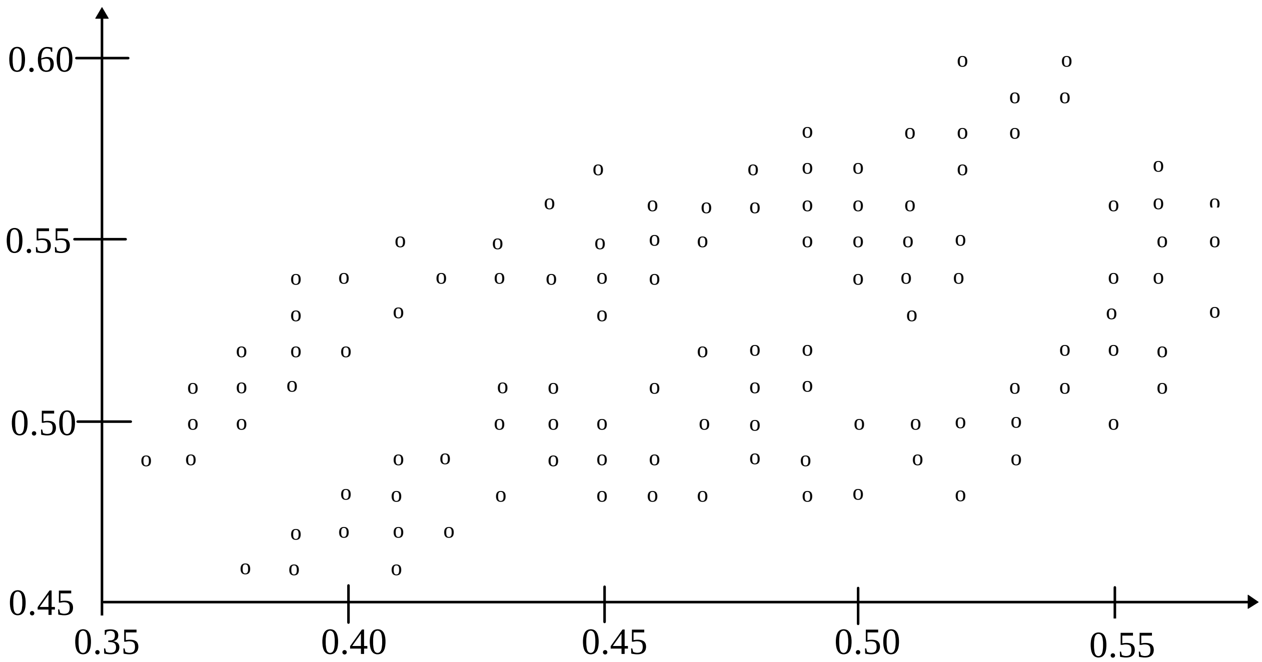
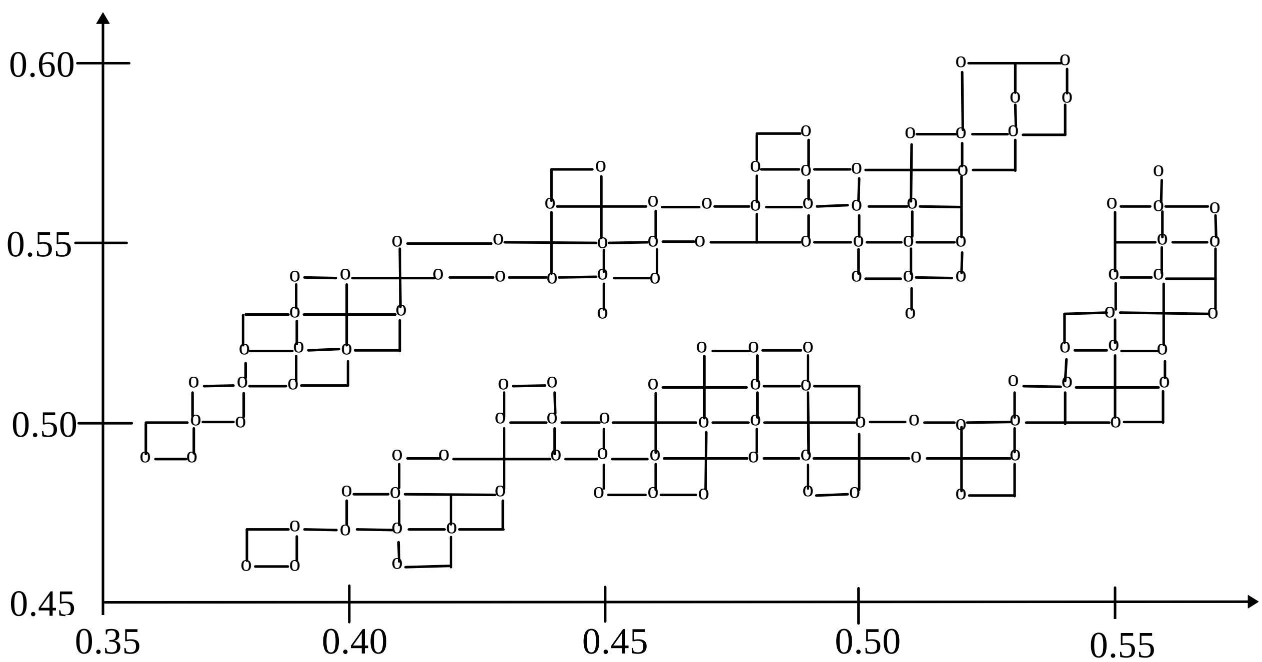


Figure 2 The fuzzy lattice neurocomputing (FLN) scheme's processing cycle.

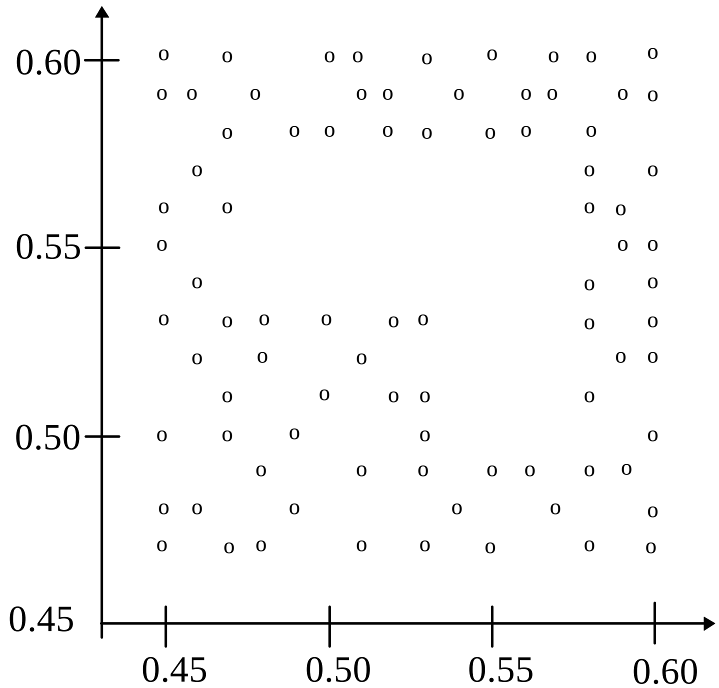


(a)

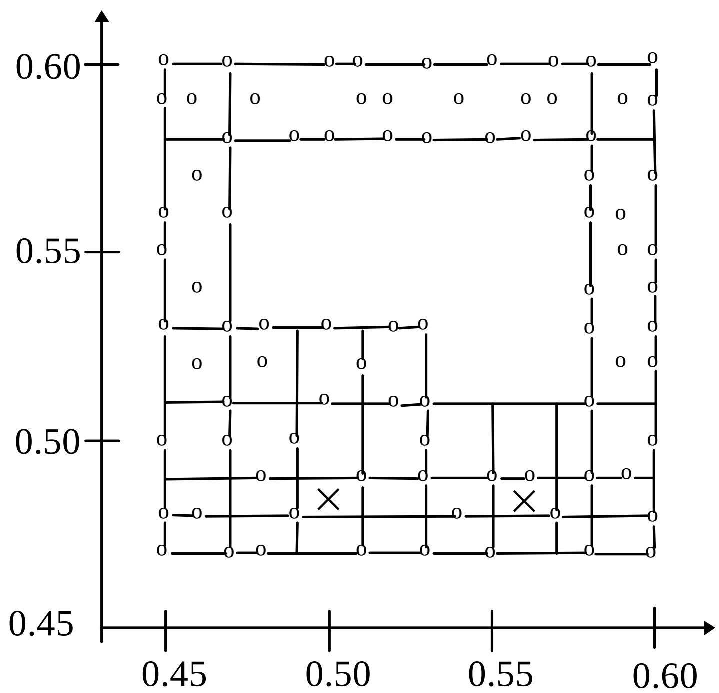


(b)

Figure 3 (a) Non-linearly separable two-dimensional categories.
(b) Separation was possible by FLN clustering. 39 overlapping boxes were used to identify each one of the two shown categories.



(a)



(b)

Figure 4 (a) A non-convex (hollow) two-dimensional category.
(b) The identification of this non-convex category was made possible by the FLN using 10 overlapping boxes. Boxes marked by an × are not inside any of the above “10 boxes”.

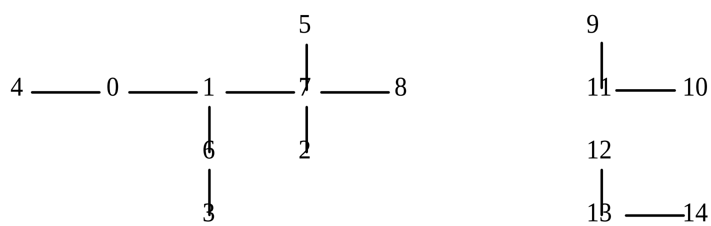


Figure 5 The second level of FLN clustering, partitions the first level's classes.

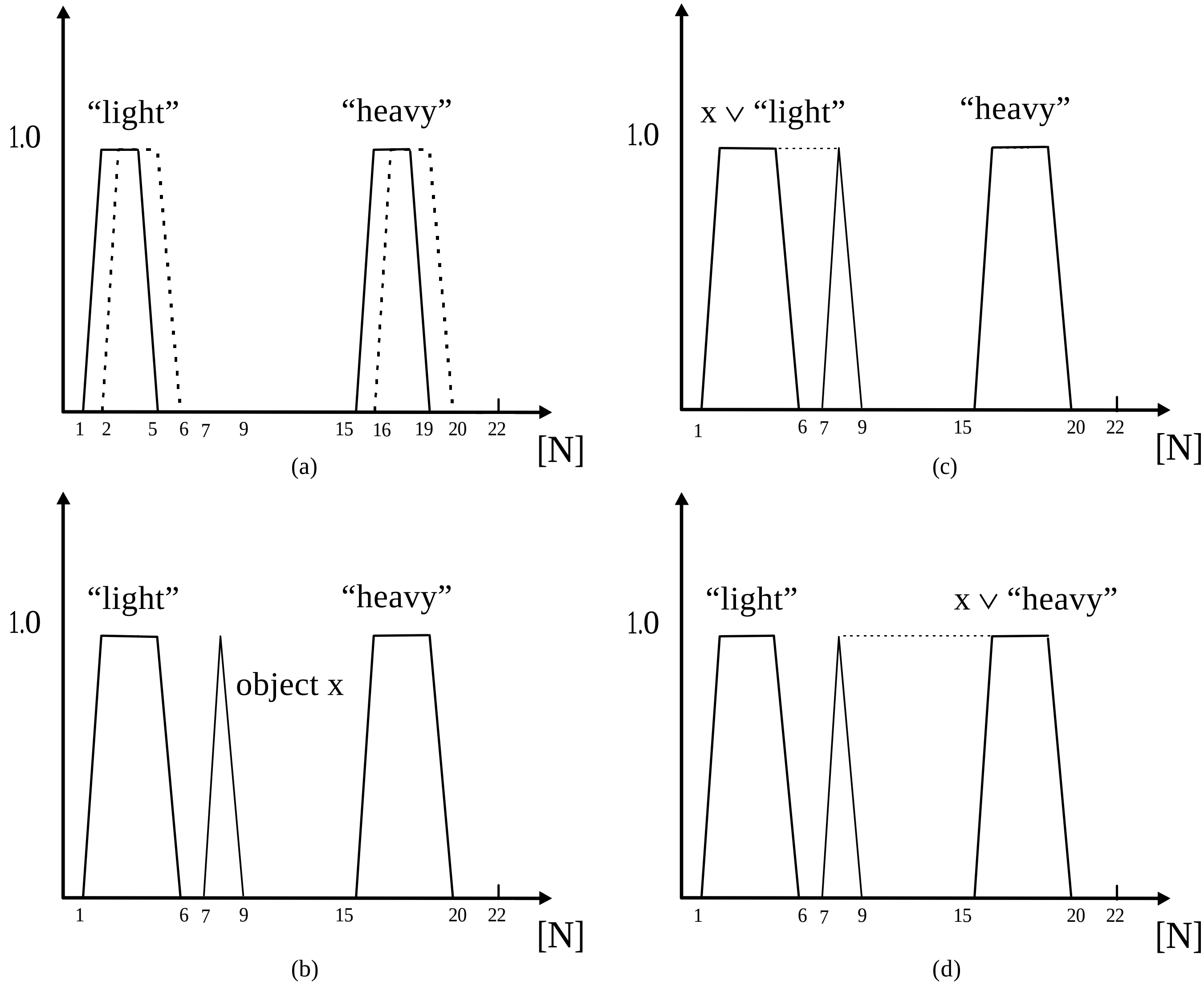


Figure 6 (a) Fuzzy definitions for “light” and “heavy” by two experts in the fuzzy “lattice of weights”. The definitions of expert #1 are by the solid lines, whereas those of expert #2 are by the dashed lines.

(b) The FLN combined the definitions of the two experts by the lattice-join (∨) operation to form a single “light” and a single “heavy” class. An object x, characterized by the shown triangular membership function, is to be assigned to one of the classes “light” or “heavy”.

(c) The degree of inclusion of x in the class “light” was calculated, via the lattice-join x∨“light”, to be σ(x≤“light”)≅0.57.

(d) The degree of inclusion of x in the class “heavy” was calculated, via the lattice-join x∨“heavy”, to be σ(x≤“heavy”)≅0.33.