Fuzzy Lattice Neurocomputing (FLN) Models[1]

by

Vassilis G. Kaburlasos, Vassilios Petridis

Aristotle University of Thessaloniki
Department of Electrical and Computer Engineering
Robotics and Automation Laboratory
GR-54006 Thessaloniki
Greece

Abstract - In this work it is shown how fuzzy lattice neurocomputing (FLN) emerges as a connectionist paradigm in the framework of fuzzy lattices (F L ¡ framework) whose advantages include the capacity to deal rigorously with: disparate types of data such as numeric and linguistic data, intervals of values, \missing" and \don't care" data. A novel notation for the FL-framework is introduced here in order to simplify mathematical expressions without losing content. Two concrete FLN models are presented, namely \¾ ¡ FLN" for competitive clustering, and \FLN with tightest ¯ts (FLNtf)" for supervised clustering. Learning by the ¾ ¡ FLN, is rapid as it requires a single pass through the data, whereas learning by the FLNtf, is incremental, data order independent, polynomial $O(n^3)$, and it guarantees maximization of the degree of inclusion of an input in a learned class as explained in the text. Convenient geometric interpretations are provided. The ¾ ¡ FLN is presented here as fuzzy-ART's extension in the F L ¡ framework such that ¾ ¡ FLN widens fuzzy-ART's domain of application to (mathematical) lattices by augmenting the scope of both of fuzzy-ART's choice (Weber) and match functions, and by enhancing fuzzy-ART's complement coding technique. The FLNtf neural model is applied to four benchmark data sets of various sizes for pattern recognition and rule extraction. The benchmark data sets in question involve jointly numeric and nominal data with \missing" and/or \don't care" attribute values, whereas the lattices involved include the unit-hypercube, a probability space, and a Boolean algebra. The potential of the F L ¡ framework in computing is also delineated.

Keywords - Lattice theory, fuzzy set theory, self-organization, adaptive resonance theory, fast learning, database mining, hybrid computing, computational learning.

[1]This work was supported in part by the European Brite Euram Project BE7470 (MITOS Project).

LIST OF SYMBOLS

L	A (complete) lattice
$L^{@}$	The dual of a (complete) lattice
O_L (I_L)	Least (greatest) element of a complete lattice L
$_L$	Inclusion relation in L
_$_L$ (^$_L$)	Join (meet) operator in L
k_L	Incomparable relation in a (complete) lattice L
»	An order-isomorphism between two partly ordered sets
(X; ¹)	A fuzzy set ¹ : X ! [0; 1] on the universe of discourse X
< L; ¹ >	A fuzzy lattice, where (L £ L; ¹) is a fuzzy set
¾ : L £ L ! [0; 1]	A lattice inclusion measure
v : L ! R	A positive valuation function
k(x $_L$ u) = $\frac{v(u)}{v(x\__L u)}$	An inclusion measure as a function of a positive valuation v
¿(L)	The lattice of intervals of a lattice L
a(L)	The set of atoms of a lattice L; a(L) ½ ¿(L)
f = fw$_i$g$_{i2I}$	A family (of lattice intervals), where w$_i$ 2 ¿(L) and I is a ¯nite index set
F$_{¿(L)}$	The set of families
c = [$_{i2I}$ w$_i$	De¯nition of a class, where fw$_i$g$_{i2I}$ = f 2 F$_{¿(L)}$
C	The set of classes
Q(c)	The quotient of a class
a$_C$: ¿(L) £ C ! [0; 1]	A category activation function. By de¯nition, a$_C$(x j c) , a$_C$(x j Q(c))
Á : ¿(L) ! L £ L	An injective monotone map
Z : ¿(L) ! R	The size of an interval. By de¯nition, Z([a; b]) = v(b) ¡ v(a), v is a positive val
I	The unit interval of real numbers; I = [0; 1]
U	The unit hypercube; U = I £ : : : £ I
g : a(L) ! D	The category function, where D is a set (of labels) of ¯nite cardinality
(¢; g(¢))	A labelled datum ¢ 2 ¿(L) such that g(x) = g$_0$ for all atoms x in ¢
¾ ¡ FLN	An FLN model for competitive clustering
	(it is fuzzy-ART's extension in the FL-framework)
FLNtf	An FLN model for data-order-indepedent supervised clustering

1 INTRODUCTION

Most connectionist paradigms including the Adaptive Resonance Theory (ART) [8], [14], Radial Basis Function networks [24], n-tuple Classi¯ers [35], [54], k-Means Clustering [1], Probabilistic Neural Networks [61], etc. share a common feature : they are applicable either to a Boolean lattice of zeros and ones, or to the N ¡ dimensional Euclidean space where one dimension corresponds to the totally-ordered line of real numbers. In this work a di®erent approach to neurocomputing is proposed, that is the \fuzzy lattice neurocomputing (FLN) paradigm" whose models are applicable to a fuzzy (mathematical-) lattices [39], [51], [52]. More speci¯cally, an FLN model is either separately or jointly - in any combination - applicable to: the N ¡ dimensional Euclidean space, a Boolean lattice of zeros and ones, a probability space, a collection of fuzzy sets, a set of symbols, a set of propositions, etc. The long-term goal of the \FLN approach to neurocomputing" is the capacity to accommodate jointly and rigorously disparate types of data with a connectionist architecture.

It has been claimed by the authors in [39], [51], [52] that FLN has originated from adaptive resonance theory (ART). This paper focuses, in part, on substantiating the aforementioned claim. In particular it is shown here how a speci¯c FLN model, namely ¾ ¡ FLN, stems from and enhances a speci¯c ART model, the latter is fuzzy-ART. Apart from ¾ ¡ FLN another FLN model, that is the \FLN with tightest ¯ts (FLNtf)", is introduced here and its e±ciency is demonstrated on benchmark problems. In the rest of this section relevant work on both the ART and the FLN is outlined.

The theory of adaptive resonance (ART) began with an analysis of human cognitive information processing and stable coding in a complex input environment [29], [30]. The original work on ART has triggered a lasting research activity and several models have been introduced, studied comparatively, implemented in hardware, and applied to various practical problems. In particular, ART1 was introduced in [8] for self-organizing recognition categories for arbitrary sequences of binary input patterns. The ART2 model [9] pursued the same objective as ART1 but for either binary or analog inputs. In [6] ART2's learning algorithm was presented in analogy with k-means clustering. ART2's principles have been employed in [53] to implement a neural network, whereas an ART2 model was studied in [65] comparatively to a self-organizing neural network namely \Dignet". Precipitation of data processing in the ART2 architecture has been a®ected by the ART 2-A algorithm [15]. The ART3 neural network has implemented both fast and slow learning with either distributed or compressed code representations [10]. The fuzzy-ART architecture has been introduced in [14] for self-organizing analog patterns in the N ¡ dimensional unit-hypercube.

The ARTMAP architecture for supervised learning in [12] has demonstrated a capacity to learn sequences of binary vectors. The fuzzy-ARTMAP [11], [13] extended the applicability domain of ARTMAP to analog patterns. Various learning properties of the fuzzy-ART algorithm have been studied in [34]. The e®ect of the choice parameter on the order of category selection in both fuzzy-ART and fuzzy-ARTMAP has been studied in [26]. In [19] a procedure has been introduced that identi¯es an order of presentation of the training data for fuzzy-ARTMAP in order to improve fuzzy-ARTMAP's generalization performance. The Gaussian-ARTMAP [66] employed ART's choice function as the discriminant function of a Gaussian classi¯er to achieve noise resistant parallel computing. A fairly detailed list of various ART models and their application to an array of learning and pattern recognition tasks has been given in [7] where, in addition, dART with distributed code representations is introduced as a generalization of both fuzzy-ART and fuzzy-ARTMAP. Lately, the ARTMAP-IC model has been introduced [16] for resolving the problem of \con°icting" training data by instance counting.

A hardware implementation of an ART model has been the outcome of various e®orts. For instance, the work in [60] illustrates a hardware implementation of a modi¯ed ART1 algorithm. Older hardware implementations of ART1 have been shown in [36] and [64]. The work in [4] has proposed a more compact and faster hardware implementation for fuzzy-ART.

Furthermore, the basic ART learning algorithm has been employed by researchers in di®erent learning contexts. For instance learning by self-organizing ART models has been presented as a speci¯c learning instance in the Mean Risk Functional framework [44]. In [42] a hybrid network is proposed by integrating a conventional fuzzy ARTMAP neural network and a probabilistic neural network. The FALCON-ART algorithm in [43] employes ART synergistically with backpropagation for structure/parameter learning and automatic control applications. The LAteral Priming ART (LAPART), which is functionally similar to fuzzy-ARTMAP, is combined with the stack interval network in [32] in order to verify and validate the learned rules.

Despite ART's many advantages for learning [28], the underlying applicability domain of an ART model remains invariably the N ¡ dimensional Euclidean space and quite often the N ¡ dimensional unit hypercube, in particular. Note that the unit hypercube includes Boolean vectors of zeros and ones, and moreover, as it has been shown in [14], fuzzy-ART's learning algorithm reduces to ART1's learning algorithm when only binary input vectors are dealt with. It turns out that the learning and decision making capacities of ART can be retained when disparate types of data such as N ¡ dimensional vectors of real numbers, images, fuzzy sets, symbols, propositional statements, etc. are treated either separately or jointly in any combination. To the aforementioned end the theory of (mathematical) lattices provides a suitable framework.

Lattice theory [2], [20], [57] has been employed practically in the past in various contexts. For instance, lattice theory is employed in [47] for describing acquisition of mental models. In [58] lattice theory is employed for rules' learning, while in [18], [22], [25], [27] it is employed for \decision making under ambiguity" in the context of fuzzy logic. Furthermore, it can be argued that ART models deal implicitly with mathematical lattices as detailed in section 5.

The authors of this paper have introduced a novel framework for learning in lattices, that is the framework of fuzzy lattices or FL_framework for short [39], [51], [52] which is delineated in section 2. In section 3 are presented the ¾ ¡ FLN model for competitive clustering as well as the \FLN with tightest ¯ts", or FLNtf for short, for data order independent supervised clustering. Section 4 provides geometric interpretations of the basic tools and techniques of the FL-framework. Section 5 shows an \in depth" comparison of fuzzy-ART and ¾ ¡ FLN in the unit hypercube and it explains how ¾ ¡ FLN has evolved from fuzzy-ART. The learning capacity of the novel FLNtf model is demonstrated on four benchmark data sets in Section 6. Finally, section 7 concludes this work with a discussion of the overall qualities of both FLN and FL-framework as well as of their potential future utility.

2 FUZZY LATTICE FRAMEWORK: A NEW FOUNDATION FOR LEARNING AND DECISION MAKING

The framework of fuzzy lattices, or FL-framework for short, has been introduced elsewhere [39], [51], [52]. In this section the theoretical foundations of the FL-framework are delineated using a novel symbolism which allows for simpler and more elegant mathematical expressions without losing content. It has been recognized lately that the notion dual lattice, which is de¯ned in this section, is central in the FL-framework. Failure to use dual lattices in [39], [51], [52] has made the exposition of useful techniques needlessly complicated. Therefore the notion dual lattice is employed explicitly in this section in order to simplify notation as shown underneath. In addition, an enhanced terminology is employed, for instance the notion \family activation function" in [52] has been replaced here by the more useful notion \category activation function", as well as novel and useful notions have been introduced as explained in this section. Any theoretical results are summarized in this section without mathematical proof, nevertheless the proofs in question can be sought in cited references. Finally, it has to be pointed out that the terminology used in the FL-framework is based on \set theory" rather than on a \domain speci¯c theory" for learning and decision making.

2.1 Fuzzy lattices

An important notion in the FL-framework is the notion partly ordered set which is de¯ned as follows.

De¯nition 1 A **partly ordered set** is a set in which a binary relation x y is de¯ned, which satis¯es the following conditions for all x; y; z

P1. For all x, x x. (Re°exive)
P2. If x y and y x, then x = y. (Antisymmetry)
P3. If x y and y z, then x z. (Transitivity)

Mapping one partly ordered set to another one is useful. The following maps have been considered.

De¯nition 2 Let P and Q be partly ordered sets. A map Ã : P ! Q is said to be
(i) **order-preserving** (or, alternatively, **monotone**), if x y in P implies Ã(x) Ã(y) in Q;
(ii) an **order-isomorphism**, if both \x y in P , Ã(x) Ã(y) in Q" and \Ã is onto Q".

When there is an order-isomorphism from P to Q, then P and Q are called order-isomorphic, symbolically P ≌ Q. The notion conventional lattice (or, alternatively, crisp lattice) is cited next for the reader's convenience.

De¯nition 3 A **conventional lattice**, or alternatively **crisp lattice**, is a partly ordered set any two of whose elements have a greatest lower bound or **meet** denoted by x ^ y and a least upper bound or **join** denoted by x _ y.

The partial ordering relation in a crisp lattice L is denoted in the FL-framework by $_L$, whereas the join and meet operators in a crisp lattice L are denoted, respectively, by $\__L$ and $\hat{}_L$. Instrumental to a simpli¯ed theoretical exposition of the work here is the notion dual crisp lattice. The dual of a crisp lattice L has, by de¯nition, the same underlying set but its partial ordering is the converse of L. In this work the dual of a lattice L is denoted by $L^@$ [20]. For a; b 2 L it follows a $\__L$ b = a $\hat{}_{L^@}$ b and a $\hat{}_L$ b = a $\__{L^@}$ b.

De¯nition 2(ii) regarding \order-isomorphism" applies, as well, to crisp lattices; hence an isomorphic relation between two crisp lattices A and B will be denoted by A ≌ B. The work in this paper deals with complete lattices. A lattice L is complete when each of its subsets has a least upper bound and a greatest lower bound in L. A non-void complete lattice L contains a least and a greatest element denoted, respectively, by O_L and I_L.

Another useful notion is the product lattice of two crisp lattices A and B denoted by A £ B and de¯ned such that $(x_A; x_B)$ $_{A£B}$ $(y_A; y_B)$ if and only if x_A $_A$ y_A and x_B $_B$ y_B [2], [20], [57]. The meet in A £ B is given by $(x_A; x_B)$ $\hat{}_{A£B}$ $(y_A; y_B)$ = $(x_A$ $\hat{}_A$ $y_A; x_B$ $\hat{}_B$ $y_B)$, and the join is given by $(x_A; x_B)$ $\__{A£B}$ $(y_A; y_B)$ = $(x_A$ $\__A$ $y_A; x_B$ $\__B$ $y_B)$ [2], [20]. The product of N lattices ensues likewise. Apparently if L is a lattice then L £ L is a product lattice. Furthermore if L is a complete lattice then so are lattices L £ L, $L^@$, and $L^@$ £ L. The least element in L £ L is $O_{L£L}$ = $(O_A; O_A)$, whereas its greatest element is $I_{L£L}$ = $(I_L; I_L)$.

If x; y 2 L then either \x and y are comparable", that is either x $_L$ y or y $_L$ x, or \x and y are incomparable", that is neither x $_L$ y nor y $_L$ x. Two incomparable elements x; y 2 L are denoted by x k_L y. Note that the subscript in all of $_L$, $\__L$, $\hat{}_L$, and k_L is meant to explicitly identify the underlying lattice. When , _, ^, and k are used in this work without a subscript they refer to the set R of real numbers. Note that R is a lattice such that for x; y 2 R it follows x ^ y = minfx; yg and x _ y = maxfx; yg. Furthermore xky is always false in R hence the elements of R are conventionally called totally ordered and R is called a chain (lattice) [2], [31].

The concept fuzzy lattice has been introduced by the authors of this work in order to extend the lattice ordering relation to all pairs (x; y) 2 L £ L of a crisp lattice L. Such an extended relation may be regarded as a fuzzy set on the universe of discourse $L \times L$ [67]. Note that in this work a fuzzy set is denoted by $(X; \mu)$, where X is the universe of discourse and μ is a membership function $\mu : X \to [0; 1]$. Hence, the aforementioned extended lattice ordering relation implies a fuzzy set $(L \times L; \mu)$, which is defined under condition "$\mu(x; y) = 1$ if and only if $x \le_L y$". The definition of a fuzzy lattice follows.

Definition 4 A **fuzzy lattice** is a pair $\langle L; \mu \rangle$, where L is a crisp lattice and $(L \times L; \mu)$ is a fuzzy set such that $\mu(x; y) = 1$ if and only if $x \le_L y$.

Note that if $\langle L; \mu \rangle$ is a fuzzy lattice then its dual fuzzy lattice is defined as $\langle L; \mu^{\partial} \rangle$ with $\mu^{\partial}(a; b) = \mu(b; a)$. The collection of all fuzzy lattices is referred to by term framework of fuzzy lattices or FL-framework for short. The significance of the above definition is that it allows one to specify a degree of inclusion of a crisp lattice's element to any other element. Note that $\mu(x; y) = 1$ in a fuzzy lattice $\langle L; \mu \rangle$ does not necessarily imply $\mu(y; x) = 0$ and it could well be $\mu(y; x) > 0$. Regarding transitivity in a fuzzy lattice $\langle L; \mu \rangle$ note that the conventional transitivity property holds only in the sense that $\mu(x; y) = 1$ and $\mu(y; z) = 1$ jointly imply $\mu(x; z) = 1$. However, when it is either $\mu(x; y) \neq 1$ or $\mu(y; z) \neq 1$ then $\mu(x; z)$ could be any number in $[0; 1]$. The following definition will eventually enable the fuzzification of a crisp complete lattice.

Definition 5 Let L be a complete crisp lattice with least and greatest elements O_L and I_L, respectively. An **inclusion measure** σ on L is a map $\sigma : L \times L \to [0; 1]$ such that for $u; w; x \in L$ the following conditions are satisfied

(C1) $\sigma(x; O_L) = 0$, $x \neq O_L$,
(C2) $\sigma(x; x) = 1$, $\forall x \in L$, and
(C3) $u \le_L w \Rightarrow \sigma(x; u) \le \sigma(x; w)$ (Consistency Property).

It follows $\sigma(x; I_L) = 1$, $\forall x \in L$ [51]. It can be argued that $\sigma(x; u)$ indicates the degree of inclusion of x in u, therefore notations $\sigma(x; u)$ and $\sigma(x \le_L u)$ will be employed interchangeably. To define an inclusion measure in a crisp lattice L a real number will be attached to each element of L through a valuation function. Recall that a valuation on a crisp lattice L is a real-valued function $v : L \to R$ which satisfies $v(x) + v(y) = v(x \vee_L y) + v(x \wedge_L y)$, $x; y \in L$. A valuation is monotone if and only if $x \le_L y$ implies $v(x) \le v(y)$, and positive if and only if $x <_L y$ implies $v(x) < v(y)$ [2], [57]. Here it has been assumed $v(O_L) = 0$ for a positive valuation because if $v(O_L) \neq 0$ then another positive valuation v^+ with $v^+(O_L) = 0$ can always be defined out of v by simply subtracting $v(O_L)$ from all $v(x); x \in L$. A positive valuation v on a lattice L renders L a metric space with metric (distance): $d(x; y) = v(x \vee_L y) - v(x \wedge_L y)$, $x; y \in L$ [2], [57]. The following result from [52] defines an inclusion measure in L.

Theorem 6 The existence of a positive valuation function v on a (crisp) complete lattice L, with $v(O_L) = 0$, is a sufficient condition for function $k(x; u) = \frac{v(u)}{v(x \vee_L u)}$ to be an inclusion measure in L.

Recall that $k(x; u)$ can be employed for indicating the degree of inclusion of x in u, therefore in the sequel $k(x; u)$ will also be denoted by $k(x \le_L u)$. Note that function $k(x \le_L u)$ equals 1 if and only if $x \le_L u$. Therefore if v is a positive valuation function on a complete lattice L then the pair $\langle L; k \rangle$ implies a fuzzy lattice.

Regarding neurocomputing in fuzzy lattices (FLN) note that "learning" can be attained by a procedure which handles (closed) lattice intervals, where a (closed) non-empty lattice interval $[a; b]$ is defined as a set of lattice elements, in particular $[a; b] = \{x : x \in L \text{ and } a \le_L x \le_L b\}$.

Consider a function τ which maps a lattice L to the collection of non-empty intervals including the empty set. The following result is from [52].

Proposition 7 If L is a complete lattice then ¿(L) is a complete lattice. The least element in ¿(L) is the empty set which is denoted by $O_{¿(L)}$ = [I_L; O_L], whereas the largest element in ¿(L) is $I_{¿(L)}$ = [O_L; I_L]. The implied lattice inclusion relation [a; b] $_{¿(L)}$ [c; d] in ¿(L) is equivalent to \c $_L$ a and b $_L$ d". For two intervals [a; b]; [c; d] 2 ¿(L) their lattice join is given by [a; b]$\__{¿(L)}$[c; d] = [a^$_L$c; b$__L$d], and their lattice meet is given by [a; b] ^$_{¿(L)}$ [c; d] = [a $\__L$ c; b ^$_L$ d] if a $\__L$ c $_L$ b ^$_L$ d, otherwise it is [a; b] ^$_{¿(L)}$ [c; d] = $O_{¿(L)}$.

Of particular interest will be subset a(L) of ¿(L) which includes all trivial intervals (singletons). Note that a(:) can be regarded as a function which maps a lattice L to its collection of singletons [x; x]; x 2 L. An element t_0 2 a(L) is always larger than $O_{¿(L)}$ and since relation \$O_{¿(L)}$ $<_{¿(L)}$ x $<_{¿(L)}$ t_0" holds for no x 2 ¿(L) it is said that an element t_0 of a(L) covers $O_{¿(L)}$, hence the elements of a(L) are called atoms and a(L) is the set of atoms in ¿(L) [2]. In the FL-framework the following term has been coined for referring to some collections of intervals in a lattice.

De¯nition 8 Let L be a lattice. A family (of lattice intervals) denoted by f = fw_ig$_{i2I}$ is a collection of lattice intervals, that is w_i 2 ¿(L) for all i 2 I, where I is an index set of ¯nite cardinality.

If f = fw_ig$_{i2I}$ is a family, the w_i's are called constituent intervals of family f. The collection of families (of lattice intervals) in a lattice L will be denoted by $F_{¿(L)}$. A useful notion is implied from considering the set-union of the members in a family (of intervals). The terms class and category have been reserved and they are used interchangeably in the FL-framework to denote the set-union of the members in a family of intervals [52].

De¯nition 9 Let L be a lattice. A class (or, alternatively, category) c in L is de¯ned by c = $\underset{i2I}{[}$ w_i, where fw_ig$_{i2I}$ is a family in $F_{¿(L)}$, and [is the set-union operator.

The set of classes in a lattice L will be denoted by C. Apparently C is a subset of the power set of L, where by \power set" is meant the set of all subsets of L. Note that one class c 2 C might have many decompositions in ¯nitely many lattice intervals, in other words many families of intervals in $F_{¿(L)}$ may specify the same class. If for two di®erent families of intervals f_1 = fu_jg$_{j2J}$ and f_2 = fw_ig$_{i2I}$, it holds $\underset{j2J}{[}$ u_j = $\underset{i2I}{[}$ w_i then it is said that both families of intervals f_1 and f_2 represent the same class; in other words fu_jg$_{j2J}$ and fw_ig$_{i2I}$ are two distinct decompositions of the same class. It is shown in [51] that for a class c 2 C there exists a unique family, namely quotient of c denoted by Q(c), such that Q(c) includes every other family representing class c. A category activation function with respect to a concrete inclusion measure ¾ in ¿(L) is de¯ned next.

De¯nition 10 A category activation function with respect to an inclusion measure ¾ in ¿(L) is de¯ned as a real valued function a_C : ¿(L) £ C ! [0; 1], such that a_C(x j c) , a_C(x j Q(c)) , $\underset{i2I}{\max}$ ¾(x $_{¿(L)}$ q_i), where Q(c) = fq_ig$_{i2I}$ is the quotient of class c.

Note that a speci¯c class c 2 C implies fuzzy set (¿(L); a_C) on the universe of discourse ¿(L) whose membership function is a_C(x j c). Given a c 2 C, the value of the category activation function a_C(x j c) for a speci¯c x 2 ¿(L), in the context of fuzzy lattice neurocomputing (FLN), can be thought of as the activation of class c at the presence of x, and in this case a_C(x j c) is interpreted as the degree of inclusion of an interval x 2 ¿(L) to class c 2 C.

From the previous analysis it follows that an inclusion measure in lattice ¿(L) will be useful. Nevertheless, the only known way to the authors of this paper for de¯ning an inclusion measure in a crisp lattice is via a positive valuation function. It has been proven in [52] that a positive valuation function in L can not imply a positive valuation in ¿(L). The de¯ciency of a positive valuation function in ¿(L), when there exists one in L, has been mended by employing a monotone map Á from ¿(L) to another lattice where a positive valuation exists. The details are elaborated in the following.

Let v be a positive valuation on a complete lattice L. The following proposition from [52] guarantees the existence of a positive valuation on L £ L.

Proposition 11 If v is a positive valuation in lattice L then function V(a; b) = v(a) + v(b), a; b 2 L de¯nes a positive valuation in lattice L £ L.

Let µ be an isomorphism µ : $L^@$! L. Furthermore consider the **lattice of abstract** (or, alternatively, **generalized**) **intervals** which is de¯ned underneath.

Lemma 12 Let L be a complete crisp lattice. Then $L^@$ £ L is a complete lattice, namely **lattice of abstract intervals** (or, alternatively, **lattice of generalized intervals**).

Obviously, the implied partial ordering relation in $L^@$ £ L is [a; b] $_{L@£L}$ [c; d] if and only if (a $_{L@}$ c , c $_L$ a) and b $_L$ d. Recall that if L is a complete lattice then $L^@$ £ L is a complete lattice as well. The least element in $L^@$ £ L is $O_{L@£L}$ = [I_A; O_A] and its greatest element is $I_{L@£L}$ = [O_A; I_A].

The aforementioned isomorphism µ establishes an isomorphism between lattices L £ L and $L^@$ £ L such that, (1) an element (a; b) of L £ L is mapped to element [µ(a); b] of $L^@$ £ L, and (2) an element [a; b] of $L^@$ £ L is (inverse-) mapped to element (µ(a); b) of L £ L. To distinguish between the elements of the aforementioned lattices, an element of lattice L £ L will be denoted within parentheses as in (a; b), a; b 2 L whereas an element of lattice $L^@$ £ L will be denoted within braces as in [a; b]; a; b 2 L.

At last, consider subset L_t of $L^@$ £ L de¯ned by L_t = f[a; b] : a; b 2 L and a $_L$ bg [f[I_L; O_L]g. It follows that lattice L_t is a sublattice of $L^@$ £ L, where the notion sublattice is de¯ned underneath.

De¯nition 13 A subset S of a crisp lattice L is called **sublattice** of L if a; b 2 S imply a ^$_L$ b 2 S and a _$_L$ b 2 S.

Note that lattice L_t does not specify sets of lattice L elements. Nevertheless it is quite straightforward to establish an isomorphism between lattices L_t and ¿(L), where lattice ¿(L) of intervals speci¯es sets of lattice L elements. The following relation summarizes all previous conclusions

$$¿(L) \cong L_t \; µ \; L^@ £ L \cong L £ L \qquad \text{(ISO)}$$

A map Á : ¿(L) ! L £ L given by Á([a; b]) = (µ(a); b) implies that if [a; b] $_{¿(L)}$ [c; d] in ¿(L) then Á([a; b]) $_{L£L}$ Á([c; d]) in L£L; in other words Á is an injective monotone map [52]. Recall that by virtue of proposition 11 a positive valuation v in L implies a positive valuation V in L £ L, and consequently it implies an inclusion measure k in L £ L by Theorem 6. In other words, based on the aforementioned isomorphic relations (ISO) it follows that k is a valid inclusion measure in ¿(L). That is, an inclusion measure ¾([a; b] $_{¿(L)}$ [c; d]) can be de¯ned in ¿(L) by ¾([a; b] $_{¿(L)}$ [c; d]) = ¾(Á([a; b]) $_{L£L}$ Á([c; d])).

All the analysis in this section has been carried out with regards to lattice L. However, it has to be pointed out that L could itself be the Cartesian product of N lattices, namely constituent lattices, symbolically L = L_1 £ : : : £ L_N. If lattices L_1; : : : ; L_N are all complete with least/greatest elements O_{L_1}=I_{L_1}; : : : ; O_{L_N}=I_{L_N}, respectively, then L is a complete lattice with least element (O_{L_1}; : : : ; O_{L_N}) and greatest element (I_{L_1}; : : : ; I_{L_N}).

Proposition 14 If v_1; : : : ; v_N are valuations on lattices L_1; : : : ; L_N, respectively, then function v = v_1 + : : : + v_N is a valuation on the product lattice L = L_1 £ : : : £ L_N.

Note that it su±ces to be all valuations $v_1; : : : ; v_N$ monotone so as valuation $v = v_1 + : : : + v_N$ to be monotone as well. If at least one of the monotone valuations $v_1; : : : ; v_N$ is, in addition, a positive valuation then v is a positive valuation on the product lattice L [52].

To simplify notation a convention was made regarding the elements of lattices L £ L and $L^@$ £ L when L is a product lattice. Then an element of lattice L £ L will be denoted by $(a_1; b_1; : : : ; a_N; b_N)$, where $(a_1; b_1)$ 2 L_1 £ $L_1; : : : ; (a_N; b_N)$ 2 L_N £ L_N; whereas an element of lattice $L^@$ £ L will be denoted by $[a_1; b_1; : : : ; a_N; b_N]$, where $[a_1; b_1]$ 2 L_1 £ $L_1; : : : ; [a_N; b_N]$ 2 L_N £ L_N. To allow for a quantitative speci¯cation of the magnitude of a lattice interval the following de¯nition is given.

De¯nition 15 The size of a lattice interval [a; b] 2 ¿(L) with respect to a speci¯c positive valuation v on lattice L is de¯ned as a real valued function Z : ¿(L) ! R given by Z([a; b]) = v(b) ¡ v(a).

We remark that the size of a lattice interval can be regarded as some type of a Hamming distance. If t_r = [a; a] is an atom in a(L) ½ ¿(L) then it follows $Z(t_r)$ = Z([a; a]) = v(a) ¡ v(a) = 0, that is the size of an atom is zero. Furthermore note that the size of the least element $[I_A; O_A]$ in ¿(L) is $Z([I_L; O_L]) = v(O_L)$ ¡ $v(I_L)$ = ¡$v(I_L)$ since it has been assumed $v(O_L) = 0$.

2.2 A fuzzy lattice in the unit-hypercube

The results obtained in the FL-framework are valid in a mathematical lattice su±ces the lattice in question is a complete one, and two functions are available namely a positive valuation function v and an isomorphic function µ. In this section a complete lattice is shown equipped with the aforementioned functions. More speci¯cally, a lattice, denoted by U, is de¯ned on the N-dimensional unit-hypercube with partial ordering $(x_1; : : : ; x_N)$ $_U$ $(y_1; : : : ; y_N)$, x_1 $y_1; : : : ; x_N$ y_N; where $(x_1; : : : ; x_N); (y_1; : : : ; y_N)$ 2 U. In the interest of simplicity the dimension of the unit-hypercube will be suppressed therefore the corresponding lattice will be denoted simply by U.

Lattice U is the product of N identical constituent lattices, these are the chains I = [0; 1], where the implied lattice ordering relation is the conventional \less than or equal to" relation between real numbers. Recall that a chain is a lattice characterized by a total ordering of its elements [2], [31]. Moreover each of the N (lattice) chains I, is a complete one with least element O_I = 0:0 and greatest element I_I = 1:0. Fig.1 illustrates some notions of the FL-framework with reference to lattice U in the case N = 2, that is the unit-square on the plane. Note that a rectangle (a box) corresponds to an interval in lattice U. Fig.1 (a) and Fig.1(b) demonstrate the partial ordering relation in U. Fig.1(c) shows how di®erent classes inside the unit-square can be de¯ned by the set-union of families of overlapping boxes.

On the one hand, any monotone increasing function on the unit interval [0; 1] is a positive valuation on chain I. In this paper, linear positive valuations of the form $v_i(x_i) = c_i x_i; i$ 2 f1; : : : ; Ng with $c_i > 0$ have been considered, resulting in positive valuations of the form $v(x_1; : : : ; x_N) = c_1 x_1 + : : : + c_N x_N$, where $c_i > 0; i$ 2 f1; : : : ; Ng in lattice U. Note that when de¯nition 15 is applied in lattice U with positive valuation $v(x_1; : : : ; x_N) = x_1 + : : : + x_N$ then the size Z([a; b]) of a hyperbox $[a; b] = [(a_1; : : : ; a_N); (b_1; : : : ; b_N)]$ is given by Z([a; b]) = v(b) ¡ v(a) = $v(b_1; : : : ; b_N)$ ¡ $v(a_1; : : : ; a_N)$ = $\overset{\mathbf{P}}{_{i=1}}$ $(b_i$ ¡ $a_i)$. In the aforementioned speci¯c case, the size of the least element $[I_U; O_U]$ in lattice U equals $Z([I_U; O_U])$ = ¡$v(I_U)$ = ¡N.

On the other hand, the isomorphic function $µ_I(x)$ = 1 ¡ x; x 2 [0; 1] has been employed in lattice I. Furthermore, $µ_I(x)$ implies an isomorphic function, denoted by $µ_U(:)$, in the product lattice U as follows $µ_U((a_1; : : : ; a_N)) = (µ_I(a_1); : : : ; µ_I(a_N))$ = (1 ¡ $a_1; : : : ;$ 1 ¡ a_N). Note that there exist in¯nitely many isomorphic functions which can be used. For instance any member in the family (1 ¡ $x)^n$ of functions, where n is a positive integer and x 2 [0; 1], is an eligible isomorphic function in I. Apparently, alternative isomorphic functions can be devised, and the authors hold that the choice of a \good" isomorphic function as well as the choice of a \good" positive valuation function is problem dependent.

2.3 A fuzzy lattice in a probability space

Here is shown another example of a fuzzy lattice. Consider the triplet (-; F; q) where - is an abstract space, F is a ¯eld of sets, and q is a measure; if it holds in particular q(-) = 1 then q is a probability measure [21]. Apparently F is a complete crisp lattice, where the underlying lattice ordering relation is the conventional set-inclusion relation (µ). The least element in F is the empty set, denoted by ®, and the greatest element in F is the abstract space - itself, that is O_F = ® and I_F = -. Moreover note that q constitutes a positive valuation function on F, therefore the pair hF; ki de¯nes a fuzzy lattice, where k is de¯ned as usual by k(A; B) = k(A µ B) = $\frac{q(B)}{q(A [B)}$, A; B 2 F.

There exists an isomorphic function µ in F, that is in particular µ(X) = X^c, where X^c denotes the complement of \set X". Note that intervals can be de¯ned in crisp lattice F as [X; Y], where X; Y 2 F and X µ Y. The complete lattice of all intervals will be denoted by ¿(F) whereas the least and the greatest element in ¿(F) are denoted, respectively, by $O_{¿(F)}$ = [-; ®] and $I_{¿(F)}$ = [®; -].

In conclusion, all previous results drawn in the framework of fuzzy lattices are valid in a probability space. The latter underlines the wide scope and utility of the FL-framework.

2.4 Dealing with \missing" and \don't care" attribute values in the data

\Missing" and \don't care" values in the data is a part of reality in applications. Using the terminology of the FL-framework note that in an application there might be \missing" values and/or \don't care" values for some constituent lattices L_1; : : : ; L_N of a product lattice L = L_1 £ : : : £ L_N. A \missing" value may be interpreted as an \unknown" (unavailable) value for an attribute, whereas a \don't care" value may be interpreted as \all possible" attribute values. \Missing" values have been dealt with in the FL-framework by replacing them with the least-element-, and \don't care" values have been dealt with by replacing them with the greatest-element of the corresponding constituent lattice. For instance, had the constituent lattice been the chain of real numbers I = [0; 1] then in place of a \missing" value interval O_I = [1; 0] will be used, whereas in place of a \don't care" value interval I_I = [0; 1] will be used. Furthermore note that for a concrete number, say number x in [0; 1], atom [x; x] will be used.

3 TWO FLN MODELS

Several FLN models have already been introduced and applied for learning and recognition on various data sets including benchmark-, synthetic- [39], [50], [51] and medical data sets [38], [40], [49]. In this section two FLN models are presented, where learning is a®ected by clustering. A function g : a(L) ! D is employed in this section, namely category function g, where a(L) is the set of atoms in lattice L, and D is a set of ¯nite cardinality which contains all possible category labels for an atom in a(L). Recall that an atom has been de¯ned in subsection 2.1 as a trivial interval t_r = [x; x] in the lattice ¿(L) of intervals. Hence, function g assigns to each atom one category label among a ¯nite number of such labels available.

The domain of function g can be extended to the lattice ¿(L) of intervals as follows: if ¢ 2 ¿(L) and g_0 2 D then g(¢) = g_0 if and only if g(t_r) = g_0 for all atoms t_r $_{¿(L)}$ ¢. Likewise the domain of g can be extended to the collection $F_{¿(L)}$ of families as follows: if f = fw_ig$_{i2I}$ 2 $F_{¿(L)}$ and g_0 2 D then g(f) = g_0 if and only if g(w_i) = g_0 for all i 2 I. Finally, if g(f) = g_0 2 D for all families fw_ig$_{i2I}$ 2 $F_{¿(L)}$ which represent a single class c = $\underset{i2I}{[}$ w_i then it is eligible to write g(c) = g_0, in other words, under the aforementioned condition, the domain of function g has been extended to classes in C.

The training data to an FLN model underneath consist of pairs (¢;g(¢)), where ¢ 2 ¿(L) - typically ¢ is an atom in a(L), and g(¢) is the category (label) of ¢.

3.1 The ¾ ¡ FLN model for competitive clustering

The ¾ ¡ FLN model has been inspired from fuzzy-ART [14]. Both learning and testing by the ¾ ¡ FLN are a®ected by employing an inclusion measure ¾ in ¿(L). Note that the ¾ ¡ FLN has been introduced in the past under the names FLN [39], FLNN [51], and ¾ ¡ FLL [52] where the good performance of its application to disparate data sets has been demonstrated. Despite its a±nity with fuzzy-ART and its origins in the biologically motivated adaptive resonance theory (ART) [8], [30], the ¾ ¡ FLN remains, in e®ect, a learning scheme in the framework of fuzzy lattices (FL-framework) [39], [51], [52], where the latter framework has emerged as a cross-fertilization of the theory of lattices [2], [20], [57] with the theory of fuzzy sets [67]. Note that the FL-framework employes a novel-, set-theoretic terminology which \¯ts" its wide applicability domain of mathematical lattices [51], [52]. For the previous reasons, the terminology employed by an \FL-framework based model" may not fully correspond to the terminology employed by another model based on a di®erent theory for learning and decision making such as the adaptive resonance theory (ART). For instance, the terms category and class are used as synonyms in the FL-framework [52] while in ART they have distinctive meaning. Consequently it is eligible to use as synonyms, in the FL-framework, the terms category activation and class activation while in ART only the term category activation is meaningful. The a±nity of ¾ ¡ FLN with fuzzy-ART as well as some important di®erences with fuzzy-ART are summarized in the following.

The architecture of ¾ ¡ FLN is depicted in Figure 2 for the reader's convenience. The names and roles of its various subsystems are analogous to the names and roles of the corresponding subsystems in an ART model and they are explained in [51]. Note that there exist signi¯cant and inherent di®erences with ART. For instance both the inputs and the weights of the ¾ ¡ FLN architecture are speci¯ed by pairs of data indicated by double lines in Fig.2. Regarding the N ¡ dimensional Euclidean space, in particular, an input to the ¾ ¡ FLN is a vector of N intervals, that is an N ¡ dimensional hyperbox. It is clear that an individual N ¡ dimensional point $(x_1; : : : ; x_N)$ corresponds to the trivial N ¡ dimensional vector of intervals $([x_1; x_1] : : : ; [x_N ; x_N])$. Note that the practical advantage of dealing with a vector of intervals is that it may compensate for the uncertainty of measurements since an interval speci¯es a neighborhood of values.

Pairs of weights also appear in the bottom-up and the top-down weights between the category layer and the input layer of the ¾ ¡ FLN architecture (Fig.2). Note that a bottom-up pair of weights connecting an F_1 layer node and an F_2 layer node equals the top-down pair connecting the same nodes. However the ¯rst end, say x, of an interval [x; y] must be encoded by its isomorphic lattice element µ(x) on a pair of ¾ ¡ FLN weights. In particular for the Euclidean space the latter implies that an interval [x; y] can be encoded by storing the numbers \1 ¡ x", \y" in a pair of weights, respectively.

Another signi¯cant di®erence with ART is the type of the data the ¾ ¡ FLN can deal with, these data are elements of a mathematical lattice. For instance, they can be real numbers but they can also be fuzzy sets, events in a probability space, whole images, wave-forms, propositions, symbols, etc. Note that two lattice intervals in two di®erent constituent lattices de¯ne yet another interval in the corresponding product lattice, and the latter accounts for ¾ ¡ FLN's capacity to process jointly disparate types of data [51] including symbols [52].

Learning, by the ¾ ¡ FLN, is carried out rapidly by a single pass through the training data [52]. Nevertheless a potential disadvantage of learning by ¾ ¡ FLN might be that the clusters it learns (in particular their total number, their size, and location) depends on the order of presentation of the training data. The following FLN model remedies the aforementioned potential disadvantage at the \price" of polynomial learning complexity.

3.2 The FLN with tightest ¯ts (FLNtf) model for supervised clustering

3.2.1 Background

The \FLN with tightest ¯ts", or FLNtf for short, has been inspired from the probably approximately correct (PAC) statistical learning model, in particular the Rectangle Learning Game [5], [41]. A basic asumption by the FLNtf is that a training datum belongs to no more than one category. The FLNtf a®ects learning by ¯nding the collection of tightest ¯ts, de¯ned underneath, in the training data, whereas testing is a®ected by employing an inclusion measure ¾ in ¿(L).

De¯nition 16 A ¯t, say ¯t T, over a set f(¢$_i$;g(¢$_i$))g$_{i2f1;:::;ng}$, where ¢$_i$ 2 ¿(L) and g is a category function, is an interval T 2 ¿(L), which satis¯es the following two conditions

(i) T = $\underset{s2S}{\_}$¢$_s$, where S is a subset of f1;:::;ng such that g(¢$_s$) = g$_0$; s 2 S for some constant g$_0$.

(ii) For all ¢$_j$; j 2 f1;:::;ng with g(¢$_j$) 6= g$_0$ it holds ¢$_j$ ^$_{¿(L)}$ T = ®.

A ¯t is called tightest ¯t if it satis¯es the following condition in addition.

(iii) For all ¢$_k$;k 2 f1;:::;ng with g(¢$_k$) = g$_0$ (other than those ¢$_k$ in (i)) it holds (T _$_{¿(L)}$ ¢$_k$) ^$_{¿(L)}$ ¢$_j$ 6= ® for some j 2 f1;:::;ng with g(¢$_j$) 6= g$_0$.

Illustrations of the de¯nition above are provided in the following. Condition (i) requires that a ¯t has to be the lattice join of a number of training data intervals that all belong to the same category g$_0$. Condition (ii) requires that there should be no contradiction between a ¯t and a training datum, where a contradiction between two pairs (T;g(T)) and (¢;g(¢)) with g(T) 6= g(¢) implies that intervals T and ¢ share an atom. Condition (iii) requires that a ¯t T is tightest, in the sense that no other datum of the same category can be accommodated in T without contradicting a training datum. The notion tightest ¯t is explained schematically in Fig.3. Fig.3(a) shows a collection of points (atoms) partitioned in two categories labelled respectively by *" and \o" in the unit square. Fig.3(b) shows the tightest ¯ts which correspond to the aforementioned points. The term rule will also be used alternatively for a tightest ¯t. Note that de¯nition 16 implies that if all the training data are in the same category, that is if g(¢$_1$) = g(¢$_2$) = ::: = g(¢$_n$) = g$_0$, then the corresponding tightest ¯t is given by T = $\underset{s2f1;:::;ng}{\_}$¢$_s$.

The following FLNtf algorithm implies that for a ¯nite training set f(¢$_i$;g(¢$_i$))g$_{i2f1;:::;ng}$ the collection of tightest ¯ts exists and it is unique. Illustrations are provided underneath with reference to Fig.4. In particular, it is shown that the collection of tightest ¯ts is identical to the set of leaves of a tree structure which grows according to the following algorithm.

The FLNtf algorithm

1. Consider the next training pair (¢$_i$;g(¢$_i$)) at level-0, i = 1;:::;n of Fig.4.
2. At level-1 consider all the join-intervals, those are the \lattice joins" ¢$_j$ _$_{¿(L)}$ ¢$_i$ where g(¢$_j$) = g(¢$_i$) and ¾(¢$_j$ $_{¿(L)}$ ¢$_i$) < 1.
3. Delete join-intervals which are contradictory[2] to training data.
4. If all the join-intervals are contradictory then ¢$_i$ is a tree leaf, that is a tightest ¯t.
5. Otherwise, for each (non-contradictory) join-interval ¢ go down to level-2 in Fig.4 and calculate ¢$_j$ _$_{¿(L)}$ ¢ where g(¢$_j$) = g(¢) and ¾(¢$_j$ $_{¿(L)}$ ¢) < 1.
6. Delete join-intervals which are contradictory to training data.

[2]Two intervals ¢$_i$; ¢$_j$ 2 ¿(L) are called contradictory when ¢$_i$ overlaps ¢$_j$ and g(¢$_i$) 6= g(¢$_j$). The test for verifying whether ¢$_i$ and ¢$_j$ contradict each other is called here a comparison.

7. If all the join-intervals of a ¢ are contradictory then ¢ is a tree leaf, that is a tightest ¯t.
8. The previous steps (5 thru 7) are repeated for each node at each level of the tree in Fig.4 for a maximum number of n levels.
9. In conclusion all the tightest ¯ts that include $¢_i$ will be calculated.

Step 8 above guarantees that the FLNtf algorithm will terminate in a ¯nite number of steps since the number of levels is bounded by n that is the total number of training data.

3.2.2 The FLNtf neural architecture

The FLNtf algorithm presented previously can be implemented as shown in Fig.5 on a neural architecture consisting of three layers, namely the Tightest Fits/Rules Layer (F_{tf}), the Hypothesis Testing Layer (F_h), and the Training Data Layer (F_d), plus an Input Bu®er which latches one training datum at a time. The Training Data Layer is employed as a long-term memory for storing permanently the training data one-by-one as they enter. The Tightest Fits Layer is employed as a long-term memory for storing the tightest ¯ts that correspond to the training data presented so far. Note that the contents of the latter layer may change as a result of learning, and the tightest ¯ts it holds may be interpreted as \rules" extracted from the training data. Finally the Hypothesis Testing Layer is employed as the system's short-term memory. In particular in the latter layer it is tested, as it will be explained shortly, whether a lattice interval appearing in a level of the FLNtf algorithm's tree structure (Fig.4) is a tightest ¯t or not. In other words, in the Hypothesis Testing Layer F_h are carried out successively the computations corresponding to successive levels in the conceptual tree structure of Fig.4.

Two consecutive layers of the FLNtf neural architecture (Fig.5) are fully interconnected. There are no interconnections between non-consecutive layers. A node in the FLNtf architecture is a simple ¾ ¡ FLN model with N + 1 input (F_1) nodes and only one category (F_2) node (see Fig.2). The ¯rst N weights of one of the aforementioned ¾ ¡ FLN(s) are employed to store a lattice interval with N constituent lattices, that is a hyperbox in the N ¡dimensional Euclidean space in particular, while the last weight in a ¾ ¡ FLN stores the category of a lattice element.

In Fig.5 the links marked by a small circle and interconnecting two \¾ ¡ FLN nodes" denote a bunch of N + 1 links used for \conducting" the elements of N constituent lattices plus the index of the corresponding category from one ¾ ¡ FLN to another. The unmarked links from the Hypothesis Testing Layer to the Training Data Layer are used for transmitting \binary acknowledgment" signals as explained in the sequel. The nodes in the Hypothesis Testing Layer are ¾ ¡ FLN(s) which appear only during learning and the number of these nodes may vary drastically during learning in the same way as the number of nodes varies in a level of the FLNtf algorithm's tree structure (Fig.4). However, when the learning concludes then the nodes in the Hypothesis Testing Layer cease to exist. That is the reason those nodes are shown by dotted lines in Fig.5.

The FLNtf neural architecture shown in Fig.5 implements a scheme of serial/parallel neurocomputing. In particular, the serial part of the processing carries out successively in the Hypothesis Testing Layer (Fig.5) the computations corresponding to successive \conceptual levels" of Fig.4. That is why the number of ¾ ¡ FLN(s) appearing in the Hypothesis Testing Layer F_h during learning equals successively the number of nodes which appear in a level of the tree structure in Fig.4. Since there exist at most n consecutive such \levels" in Fig.4 it follows that there will be carried out no more than n consecutive serial processing steps in the Hypothesis Testing Layer. The time required for one such serial processing step is a constant and the term processing cycle (PC) has been reserved to denote it. All previous operations are summarized in the following algorithm (At any instant t let there be L(t) tightest ¯ts $¤_j$; j = 1; : : : ; L(t) < 1 and let g($¤_j$) denote the category of tightest ¯t $¤_j$). Note that the training data enter the FLNtf one-by-one, and moreover no initalization of the FLNtf is necessary for learning to commence.

3.2.3 FLNtf for learning

1. The next pair $(\Delta_i; g(\Delta_i))$; $i = 1, \cdots, n$ enters the Input Buffer.
2. Copy permanently a new pair $(\Delta_i; g(\Delta_i))$ to the Training Data Layer F_d.
3. If $g(\Delta_i)$ is a new category then copy $(\Delta_i; g(\Delta_i))$ to the Tightest Fits Layer F_{tf}.
4. Copy temporarily all tightest fits Λ_j; $j = 1, \ldots, L(t) < \infty$ to the Hypothesis Testing Layer F_h.
5. Contradiction Condition : If $\Delta_i \wedge_{\tau(L)} \Lambda_j \neq \alpha$ for a tightest fit Λ_j with $g(\Lambda_j) \neq g(\Delta_i)$ then go to step-11 below to manage contradictions.
6. Store in the Hypothesis Testing Layer F_h only a single datum from the Input Buffer.
7. For all the intervals residing in the Hypothesis Testing Layer F_h, namely roots, calculate their lattice joins, namely potential leaves, with each of the training data of the same category which are not included in the root (while there exist such data).
8. Delete from the Hypothesis Testing Layer all the potential leaves which either contradict a training datum or are inside a tightest fit.
9. If all the potential leaves of one specific root have been deleted, then the root in question is by definition a tightest fit, and therefore it is copied to the Tightest Fits Layer F_{tf}.
10. Go to step 1.
11. (steps 11 thru 14 below manage contradictions) By top-down "binary acknowledgment" signals from the Layer F_h to Layer F_d "mark" all the training data which are inside contradictory tightest fits.
12. Among all "marked" training data (from previous step 11) keep "marked" only the last training datum as well as those "marked" training data which are included solely in contradictory tightest fits.
13. Delete all contradictory tightest fits Λ_j. Replace the contents of the Tightest Fits Layer with the contents of the Hypothesis Testing Layer.
14. Halt any external inputs and go to step-1 to re-feed sequentially all the "marked" data from Layer F_d.
15. After all the training data have been presented, find the quotient of the tightest fits in Layer F_{tf}.

Note that "marking data" in the Training Data Layer F_d in steps 11 and 12 above, can be affected by raising a proper flag. We remark that the above scheme implements the FLNtf algorithm incrementally. That is, when for n training data $\{(\Delta_i; g(\Delta_i))\}_{i\in\{1,\ldots,n\}}$ their collection of tightest fits has been found, then based on that knowledge, the above algorithm calculates the unique collection of tightest fits which corresponds to the data set $\{(\Delta_i; g(\Delta_i))\}_{i\in\{1,\ldots,n+1\}}$.

Any order of presentation of n specific training data $\{(\Delta_i; g(\Delta_i))\}_{i\in\{1,\ldots,n\}}$ will yield the same collection of tightest fits. Nevertheless, from the above algorithm it follows that, depending upon a particular order of presentation, the intermediate collections of tightest fits might be different for different data orders but by the time all the training data $\{(\Delta_i; g(\Delta_i))\}_{i\in\{1,\ldots,n\}}$ will have been presented, the same collection of tightest fits will result in.

In the final step (step-15) of the above algorithm, the collection of tightest fits residing in the Tightest Fit Layer F_{tf} (Fig.5) is replaced by its quotient in order to maximize the degree of inclusion of an interval ¢ 2 ¿(L) in a class c 2 C [51]. Since the quotient of a family is unique it follows that the quotient corresponding to the tightest ¯ts is independent of the order of presentation of the training data.

The previous learning scheme is memory based (step-2) like the n-tuple classi¯er [35], [54], nevertheless the decision-making of F LNtf is not of statistical nature like the one of the n-tuple classi¯er since the F LNtf can generalize using a lattice inclusion measure ¾. Moreover the tightest ¯ts approach to learning by F LNtf implies local learning in a lattice; note that the biological relevance and signi¯cance of local learning has been debated lately [56].

Copying the tightest ¯ts from the Tightest Fits Layer F_{tf} to the Hypothesis Testing Layer F_h (step 4) could be avoided by fully interconnecting the Tightest Fits Layer F_{tf} with the Training Data Layer F_d (Fig.5). Nevertheless it has been decided by the authors to avoid the latter interconnection in order to keep a simple neural architecture.

It is calculated in the following, the \worse case upper bound" for the data processing complexity in terms of the previously de¯ned Processing Cycle (PC). The worse case training scenario for an on-line implementation of the F LNtf occurs when each training datum contradicts all the existing tightest ¯ts causing a re-feeding of all the previous training data. It is known that a \contradiction free" training datum at the Input Bu®er requires O(n) PC(s), hence a contradictory training datum would require $O(n^2)$ PC(s). In the aforementioned \worse case scenario" when all the training data contradict all the existing tightest ¯ts, there would be required $O(n^3)$ PC(s), that is the complexity for the F LNtf model.

3.2.4 F LNtf for testing

The testing phase for the F LNtf is the same as the one employed by the ¾ ¡ F LN [52], that is the activation a_C(¢ j c_k), k = 1; : : : ; M_F of each class c_k is calculated at the presence of ¢ 2 ¿(L), and ¢ is attached the label of the winner class.

In all, (1) the F LNtf calculates intervals/rules in a data order independent fashion, (2) the F LNtf guarantees, by de¯nition, 0% recognition error on the training data, (3) the F LNtf guarantees maximization of the degree of inclusion of the testing data in a class due to the employment of the quotient of a class, and (4) the F LNtf can deal with disparate lattice elements.

3.3 A functional comparison with fuzzy-ARTMAP

At this point it is instructive to delineate a functional comparison of F LNtf with the fuzzy-ARTMAP neural model for supervised learning [11]. The authors of this paper need to point out that F LNtf is functionally di®erent than fuzzy-ARTMAP. On the one hand, recall that fuzzy-ARTMAP consists of two interconnected fuzzy-ART modules and its \learning" depends on the order of data presentation [11], [13]. On the other hand, note that the F LNtf architecture includes a number of simple ¾ ¡ F LN modules in its layers (Fig.5) and, most important, \learning" by the F LNtf is e®ected by calculating the tightest ¯ts in the training data set in a data order independent fashion. A neural architecture for supervised learning corresponding to fuzzy-ARTMAP within the FL-framework would include two ¾ ¡ F LN modules interconnected likewise as the two fuzzy-ART modules of fuzzy-ARTMAP. Note that the name \¾ ¡ F LNMAP" has been reserved by the authors to denote the enhanced extension of fuzzy-ARTMAP in the FL-framework, nevertheless a study of ¾ ¡ F LNMAP as well as its comparison with fuzzy-ARTMAP are topics of future research.

Nevertheless, since both F LNtf and fuzzy-ARTMAP neural networks are meant for supervised learning, it is meaningful to compare their capacity for classi¯cation on the same data set. Such a comparison is shown in subsection 6.1.

4 GEOMETRIC INTERPRETATIONS

Geometric interpretations have been given for various ART models [7], [14]. In this section are illustrated geometrically on the plane tools and techniques employed by an FLN model. Note that functions v(x) = x and µ(x) = 1 ¡ x have been selected in each constituent lattice for a positive valuation function and an isomorphic function, respectively, as it has been explained in subsection 2.2.

4.1 The utility of the inclusion measure

For both Fig.6(a) and Fig.6(b) it holds [0:5; 0:6; 0:3; 0:4] = u $\;_U\;$ w = [0:4; 0:9; 0:2; 0:8]. Therefore for any box x it follows ¾(x $\;_U$ u) ¾(x $\;_U$ w), that is x is included in w more than it is in u. The latter inequality is veri¯ed underneath for both Fig.6(a) and Fig.6(b) when x is a trivial interval (an atom) in the unit square. In Fig.6(a) for x = [0:2; 0:2; 0:2; 0:2] = (0:8; 0:2; 0:8; 0:2) it follows

¾(x $\;_U$ u) = $\frac{v(u)}{v(x\_{U}u)}$ = $\frac{v(0:5;0:6;0:7;0:4)}{v((0:8;0:2;0:8;0:2)\_{U}(0:5;0:6;0:7;0:4))}$ = $\frac{v(0:5;0:6;0:7;0:4)}{v(0:8;0:6;0:8;0:4)}$ = $\frac{2:2}{2:6}$ ¼ 0:846. Likewise,

¾(x $\;_U$ w) = $\frac{v(w)}{v(x\_{U}w)}$ = $\frac{v(0:6;0:9;0:8;0:8)}{v((0:8;0:2;0:8;0:2)\_{U}(0:6;0:9;0:8;0:8))}$ = $\frac{v(0:6;0:9;0:8;0:8)}{v(0:8;0:9;0:8;0:8)}$ = $\frac{3:1}{3:3}$ ¼ 0:939. Hence ¾(x $\;_U$ u) ¾(x $\;_U$ w).

In Fig.3(b) an x inside w but outside u has been selected, in particular x = [0:8; 0:8; 0:7; 0:7] = (0:2; 0:8; 0:3; 0:7). It follows

¾(x $\;_U$ u) = $\frac{v(u)}{v(x\_{U}u)}$ = $\frac{v(0:5;0:6;0:7;0:4)}{v((0:2;0:8;0:3;0:7)\_{U}(0:5;0:6;0:7;0:4))}$ ¼ 0:814, and

¾(x $\;_U$ w) = $\frac{v(w)}{v(x\_{U}w)}$ = $\frac{v(0:6;0:9;0:8;0:8)}{v((0:2;0:8;0:3;0:7)\_{U}(0:6;0:9;0:8;0:8))}$ = 1:0. Hence ¾(x $\;_U$ u) ¾(x $\;_U$ w).

It can be veri¯ed likewise by example that the previous inequality still holds even when x is not an atom but rather x is a rectangle in the unit square. Nevertheless, it might be useful to demonstrate that a similar consistency of inequalities is not retained by the implied lattice metric (distance) function d. For instance, from Fig.6(a), it follows

d(x; u) = v(x $\_{U}$ u) ¡ v(x $^{\wedge}{}_U$ u) = v(0:8; 0:6; 0:8; 0:4) ¡ v(0:5; 0:2; 0:7; 0:2) = 2:6 ¡ 1:6 = 1:0, and

d(x; w) = v(x $\_{U}$ w) ¡ v(x $^{\wedge}{}_U$ w) = v(0:8; 0:9; 0:8; 0:8) ¡ v(0:6; 0:2; 0:8; 0:2) = 3:3 ¡ 1:8 = 1:5, hence d(x; u) d(x; w), whereas from Fig.6(b) it follows

d(x; u) = v(x $\_{U}$ u) ¡ v(x $^{\wedge}{}_U$ u) = v(0:5; 0:8; 0:7; 0:7) ¡ v(0:2; 0:6; 0:3; 0:4) = 2:7 ¡ 1:5 = 1:2, and

d(x; w) = v(x $\_{U}$ u) ¡ v(x $^{\wedge}{}_U$ u) = v(0:6; 0:9; 0:8; 0:8) ¡ v(0:2; 0:8; 0:3; 0:7) = 3:1 ¡ 2:0 = 1:1, hence d(x; u) ¸ d(x; w).

In conclusion, the lattice distance function d is not consistent in the sense of the lattice inclusion measure ¾ (consistency property (C3) of de¯nition 5).

4.2 Illustrating ¾ ¡ FLN's operation and the technique of maximal expansions

Assume that the ¾ ¡ FLN has already stored two distinct classes $c_1 = w_1$ and $c_2 = w_2$ in its category layer as shown in Fig.7(a) and let a new input x, that is a rectangle in general, enter. The two classes c_1 and c_2 compete with one another by comparing their inclusion measures ¾(x $\;_U$ c_1) and ¾(x $\;_U$ c_2), and let c_1 be the winner class. Assume that the \assimilation condition" fails then class c_1 is reset. Searching for a winner class continues, class c_2 is selected next (Fig.7(b)), and let c_2 meet the \assimilation condition". Then w_2 is replaced by w_2^{0} = x $_{U}$$w_2$. Note that rectangles w_1 and w_2^{0} overlap. Consequently, w_1 and w_2^{0} are put in the same family of lattice intervals that de¯nes a single class, namely class-c_1, and hence the technique of maximal expansions is \triggered". The aforementioned technique considers the intersection w_1 $^{\wedge}{}_U$ w_2^{0} and it expands it maximally in both dimensions as illustrated in Fig.7(c). Note that the technique of maximal expansions has been introduced in [51] as an optimization technique, more speci¯cally its goal is to maximize the degree of inclusion of an interval input in class c_1 2 C. The aforementioned optimization/maximization can be met by representing class c_1 2 C by its quotient Q(c_1).

To further the example illustrated in Fig.7, a single class c_1 is speci¯ed consisting of four rectangles c_1 = fw_1; w^0_2; w_3; w_4g, where rectangle w_1 is speci¯ed by its four corners 6-11-3-12, rectangle w^0_2 is speci¯ed by its corners 1-9-8-10, rectangle w_3 is speci¯ed by 5{6-7-8, and rectangle w_4 by 1{2-3-4. The degree of inclusion of a new input y in class-c_1, as shown in Fig.7(d), is given by maxf¾(y ∪ w_1); ¾(y ∪ w^0_2); ¾(y ∪ w_3); ¾(y ∪ w_4)g.

4.3 The utility of the technique of maximal expansions

It has been explained above that the technique of maximal expansions is, in e®ect, an optimization technique for it maximizes the degree of inclusion of an interval in a class c 2 C. The aforementioned goal is attained by representing a class c 2 C by its quotient Q(c). It should be pointed out explicitly that the technique of maximal expansions does not change the de¯nition of class c, but rather the technique in question only changes the actual representation of class c by employing the unique family Q(c) in $F_{¿(L)}$ which includes any other family representing the same class c [51].

A potential quality of the technique of maximal expansions is illustrated in Fig.8, where three rectangles w_1 = [0:15; 0:35; 0:05; 0:3], w_2 = [0:30; 0:40; 0:15; 0:25], w_3 = [0:55; 0:85; 0:15; 0:45], and an input x = [0:47; 0:47; 0:20; 0:20] are shown. Two classes c_1 and c_2 are speci¯ed as c_1 = w_1 [w_2 and c_2 = w_3. Note that the conventional Euclidean distances of x from the nearest edge of the three rectangles are: \distance from x to w_1" =j 0:47 ¡ 0:35 j= 0:12, \distance from x to w_2" =j 0:47 ¡ 0:40 j= 0:07, and \distance from x to w_3" =j 0:47 ¡ 0:55 j= 0:08. Hence it makes sense to classify x in class ¡ c_1. Nevertheless, based on the inclusion measure ¾ in Fig.8(a) and without employing the technique of maximal expansions, input x is classi¯ed in class c_2 = w_3 because in Fig.8(a) it holds

¾(x ∪ w_1) = $\frac{v(w_1)}{v(x\_{\cup}w_1)}$ = $\frac{2:45}{2:57}$ ¼ 0:953,

¾(x ∪ w_2) = $\frac{v(w_2)}{v(x\_{\cup}w_2)}$ = $\frac{2:2}{2:27}$ ¼ 0:969 , and

¾(x ∪ w_3) = $\frac{v(w_3)}{v(x\_{\cup}w_3)}$ = $\frac{2:6}{2:68}$ ¼ 0:970.

The above \counter-intuitive" classi¯cation decision can be corrected using the quotient family representation fw_1; w^0_2g, as shown in Fig.5(b), with w^0_2 = [0:15; 0:40; 0:15; 0:25] produced by the technique of maximal expansions. Winner now will be class c_1 = w_1 [w^0_2 over class c_2 = w_3, because

¾(x ∪ w_1) ¼ 0:953,

¾(x ∪ w^0_2) = $\frac{v(w^0_2)}{v(x\_{\cup}w^0_2)}$ = $\frac{2:35}{2:42}$ ¼ 0:971 , and

¾(x ∪ w_3) ¼ 0:970.

Therefore the technique of maximal expansions could make a positive di®erence in pattern recognition problems since it can imply automatically \common sense" decisions.

4.4 Illustrating the tightest ¯ts calculated by the FLNtf model

The training data shown in Fig.9(a) are fed to the FLNtf model. The data in question are 2-dimensional atoms within the unit square and they belong to two classes which are denoted respectively by an *" (that is class-c_1 with data 1; 2; 3; 4) and by an \o" (that is class-c_2 with data a; b; c; d) in Fig.9(a). Fig.9(b) shows the tightest ¯ts on the aforementioned data. Note that two tightest ¯ts assigned to di®erent classes may overlap, for example the tightest ¯ts a _ b and 2 _ 3 _ 4 overlap each other. That is, even though intervals a _ b and 2 _ 3 _ 4 are in di®erent categories their overlapping is eligible because the overlapping in question does not imply a contradiction with any training data. Fig.9(c) shows the maximal expansions of the tightest ¯ts of Fig.9(b), in particular the tightest ¯t a _ b of Fig.9(b) has been expanded maximally in Fig.9(c).

Fig.9(d) considers two more data in category *", these are data 5 and 6, and Fig.9(d) also shows the corresponding tightest ¯ts. Note that datum 5 in Fig.9(d) has caused the deletion of tightest ¯t a _ b of Fig.9(b) because the latter tightest ¯t (a _ b) now contradicts with training datum 5. Furthermore, note that datum a of class-c_2 now stands \alone" in Fig.9(d). In Fig.9(e) one more datum from category \o" shows up, that is datum e, and the corresponding tightest ¯ts are shown in Fig.9(e). Finally Fig.9(f) shows the maximal expansions of the tightest ¯ts of Fig.9(e), in particular interval e _ d in Fig.9(e) has been expanded maximally in Fig.9(f).

5 FUZZY-ART COMPARED WITH ¾ ¡ FLN LEARNING

The objective of ¾ ¡ FLN is to identify competitively sets of elements of a (complete) lattice by the set-union of crisp lattice intervals. Because the intension in this section has been to compare ¾ ¡ FLN with fuzzy-ART, most of the illustrations will be given in the N ¡ dimensional unit hypercube. Therefore the previously stated objective can be rephrased by saying that the ¾ ¡ FLN aims at identifying sets of points within the unit hypercube by the set-union of hyperboxes, or boxes for short.

Note that fuzzy-ART employes implicitly the notion mathematical lattice. We copy from the caption of Figure 6(b) in [14]: \During fast learning, R_J expands to R_J ©a, the smallest rectangle that includes R_J and a, ...". The authors of this paper note that the smallest rectangle that includes R_J and a is, by de¯nition, the lattice join of R_J and a. Furthermore, an implicit employment of a mathematical lattice is also made by the technique of \rule annihilation" of FALCON-ART [43] in order to delete unnecessary or redundant rules. More speci¯cally, \rule similarity" in [43] is determined on a dimension-by-dimension comparison process in order to calculate heuristically the degree of inclusion of one hyperbox into another and the notion mathematical lattice is employed again implicitly. Note that the FL-framework supplies a sound tool, namely inclusion measure ¾, for de¯ning in principle the degree of inclusion of a hyperbox/rule into another one.

5.1 Interpreting fuzzy-ART's \complement coding" within the FL-framework

Based on the FL-framework, fuzzy-ART's \complement coding" acquires a new interpretation as illustrated in the following. In order to de¯ne an interval on the line of real numbers two numbers are required. Nevertheless, had the two numbers in question been used without any preprocessing the bene¯t stemming from the existence of a positive valuation would be lost. Recall that the aforementioned bene¯t is the existence of an inclusion measure in the lattice ¿(L) of intervals. It has been explained in subsection 2.1 that, an isomorphic function µ : $L^@$! L guarantees the existence of a positive valuation in L £ L and hence it implies an inclusion measure in the lattice ¿(L) of intervals.

It has been further explained in section 2.2 that function $µ_I(x)$ = 1 ¡ x constitutes a valid isomorphic function for ¾ ¡ FLN. Note that isomorphic function $µ_I(x)$ = 1 ¡ x refers directly to fuzzy-ART's complement coding technique, the latter is a preprocessing normalization procedure which replaces a vector (a_1; a_2) by (a_1; a_2; 1 ¡ a_1; 1 ¡ a_2), and likewise in more dimensions, in order to avoid the category proliferation problem [14]. Hence, in the FL-framework fuzzy-ART's complement coding acquires a new-, set-theoretic meaning. Furthermore note that $µ_I(x)$ = 1 ¡ x is only one of the in¯nitely many isomorphic functions which can be used as it has been explained in subsection 2.2. The authors of this paper expect that the choice of a \good" isomorphic function µ is problem dependent for both the ¾ ¡ FLN and fuzzy-ART.

5.2 ART's Choice (Weber) function and Match function versus FLN's Inclusion Measure function.

Both of ART's choice (Weber) function and the match function correspond to ¾ ¡ FLN's inclusion measure function ¾, and the correspondence in question runs quite deep as explained in the following. A couple of \common ground assumptions" need to be made in order to compare meaningfully the abovementioned functions. More speci¯cally the aforementioned \common ground assumptions" include: 1) considering only trivial inputs (atoms), and 2) considering only fuzzy-ART's fast learning mode of operation.

Consider in the N ¡ dimensional Euclidean space, ART's choice (Weber) function and the match function:

Choice (Weber) Function : $\frac{j I \wedge w_j j}{® + j w_j j}$

Match Function : $\frac{j I \wedge wj}{j I j}$ ¸ ½

Note that operator (^) is used by fuzzy-ART as the min operator in the totally ordered set of real numbers [14]. It is known that the role of (the very small positive) parameter \a" in the denominator of the choice function is to break ties in order to select the \F$_2$ node" with the smallest size [37]. The same result can be attained by the following double test : that is (after calculating the numbers $\frac{j I \wedge w_j j}{j w_j j}$ for all nodes in layer F$_2$ - without an a in the denominator), ¯rst, select the node with the largest activation, and second, break ties by selecting the F$_2$ node with the smallest size. Note that a \double test" is employed explicitly by ¾ ¡ FLN in order to select its winner node in layer F$_2$. Moreover ¾ ¡ FLN's inclusion measure ¾ implies a sound set-theoretic interpretation, in particular the ¾ ¡ FLN identi¯es an F$_2$ node whose code \includes the most" the current input x as determined by the lattice inclusion measure ¾(x $_U$ w) = $\frac{v(w)}{v(x \__U w)}$, while ties are broken by selecting the F$_2$ winner code with the smallest size.

Regarding fuzzy-ART's Match Function note that fuzzy-ART's match criterion accepts (rejects) a winner node when the ratio $\frac{j I \wedge wj}{j I j}$ is larger (smaller) than ART's vigilance parameter ½$_{ART}$. It is shown in the following that fuzzy-ART's complement coding implies an implicit comparison of the winner node's size to a threshold size. Following fuzzy-ART's notation for a code norm in the N ¡ dimensional Euclidean space [14] it follows j w j= $\sum_{i=1}$ (w$_i$ + w$_i^c$), in particular for an input I it follows j I j= N. Recall that the size Z of a code in the FL-framework is given by Z(w) = $\sum_{i=1}$ [(1 ¡ w$_i^c$) ¡ w$_i$], and hence it relates to its norm as Z(w) = $\sum_{i=1}$ [1 ¡ (w$_i^c$ + w$_i$)] = N ¡ j w j. In particular, for an input I it follows j I j= N , Z(I) = 0. Therefore fuzzy-ART's match criterion becomes: $\frac{j I \wedge wj}{j I j}$ ¸ ½$_{ART}$) $\frac{N ¡ Z(I \wedge w)}{N}$ ¸ ½$_{ART}$) Z(I ^ w) N(1 ¡ ½$_{ART}$). In words, fuzzy-ART's code I ^ w is accepted when its size is less than or equal to an implicit threshold code size Z$_0$ = N(1 ¡ ½$_{ART}$). Otherwise, if Z(I ^ w) > Z$_0$ = N(1 ¡ ½$_{ART}$), \reset" is triggered and the search for a new winner resumes.

Regarding the ¾ ¡ FLN, its \assimilation condition" suggests an explicit comparison with a user de¯ned size threshold. However, assuming complement coding with µ$_I$(x) = 1 ¡ x and N ¡ dimensional inputs (atoms), an implicit employment of a vigilance parameter, denoted by ½$_{¾FLN}$, is implied by ¾ ¡ FLN as shown in the following. A relation is found in the ¯rst place between a code's positive valuation and its size: Z(w) = $\sum_{i=1}$ [w$_{2i}$ ¡ (1 ¡ w$_{2i¡1}$)] = v(w) ¡ N. Because there have been assumed point inputs (atoms) x with size Z(x) = 0 it follows v(x) = N. The ¾ ¡ FLN re¯nes an existing code w to x _$_U$ w by an input x if and only if the size of x _$_U$ w is less than or equal to a user-de¯ned size threshold Z$_{crit}$, that is Z(x _$_U$ w) Z$_{crit}$) $\frac{N}{Z(x \__U w)+N}$ ¸ $\frac{N}{Z_{crit}+N}$. The latter ratio is ¾ ¡ FLN's implicit vigilance parameter value, that is ½$_{¾FLN}$ = $\frac{N}{Z_{crit}+N}$. Note that since Z$_{crit}$ is in the interval of real numbers [0; N] ¾ ¡ FLN's vigilance parameter ½$_{¾FLN}$ is in the interval of real numbers [0:5; 1]. It is remarkable that the previous result is valid in a general lattice su±ces the inputs to the ¾ ¡ FLN are trivial intervals (atoms). There is much more in inequality $\frac{N}{Z(x \__U w)+N}$ ¸ ½$_{¾FLN}$ = $\frac{N}{Z_{crit}+N}$ than merely showing the existence of an implicit vigilance parameter ½$_{¾FLN}$ for ¾ ¡ FLN. It all stems from noting that ratio $\frac{N}{Z(x \__U w)+N}$ speci¯es, in e®ect, the degree of inclusion ¾(w $_U$ x) of the winner code w to an input atom x.

To recapitulate note that when the inputs to the ¾ ¡ FLN are singletons, these are atoms x = [a; a], then a competition takes place among layer F_2 nodes and the competition in question concludes by calculating the node which <u>includes</u> the current input x more than any other node, that is ¾(x $_U$ w) is the maximum. Consequently the winner is accepted if and only if it is <u>included</u> inside the input code x more than an implicit vigilance parameter, that is ¾(w $_U$ x) ¸ ½$_{¾FLN}$. Such a \double role" of the lattice inclusion measure function ¾ signi¯es a deep set-theoretic interpretation for winner choice and winner match for both the ¾ ¡ FLN and the fuzzy-ART.

It should be noted that the following extended choice (Weber) function: $\frac{v(x\hat{}_L w)}{v(w)}$ has also been considered for employment by ¾ ¡ FLN instead of the inclusion measure k(x $_L$ w) = $\frac{v(w)}{v(x\__L w)}$. Nevertheless, such an \extended choice (Weber) function" su®ers from a potentially serious drawback compared to ¾ ¡ FLN's conventional inclusion measure k(x $_L$ w) = $\frac{v(w)}{v(x\__L w)}$. The aforementioned drawback appears when the lattice meet of two elements is the least lattice element O_L, whose positive valuation has been assumed to be zero ($v(O_L) = 0$) as shown by the following example in a probabilistic context. Consider the intersection of the two sets A = fa; b; cg and B = fd; eg in the power set of X = fa; b; c; d; e; fg, that is their lattice meet A \ B = ®. Then the employment of the extended choice (Weber) function $\frac{v(A \backslash B)}{v(B)}$ will only conclude that the sets A and B are disjoint but it will not quantify the a±nity of the two sets. On the other hand, ¾ ¡ FLN's conventional inclusion measure k(x $_L$ u) indicates by a single number and with respect to a concrete positive valuation \by how much a lattice element u is de¯cient in including another lattice element x". Note that the aforementioned de¯ciency does not arise in the Euclidean space because the lattice meet (^) and the lattice join (_) of two numbers is their minimum and their maximum, respectively, for instance, 0.3^0.8=0.3 and 0.3_0.8=0.8. Hence, due to fuzzy-ART's restriction to the unit hypercube the aforementioned inherent de¯ciency of ART's choice (Weber) function could not be identi¯ed in a conventional ART model.

5.3 Advantages of the ¾ ¡ FLN

The ¾ ¡ FLN appears to be more comprehensive than fuzzy-ART in the sense that ¾ ¡ FLN can handle inputs both atoms and intervals, whereas fuzzy-ART deals solely with atoms. The advantage of dealing with intervals stems from the fact that an interval de¯nes a set of neighboring atoms therefore it might be possible to compensate for the uncertainty of the inputs by feeding to the neural network a whole set, in particular a neighborhood, of measurements instead of feeding it a single point measurement.

The ¾ ¡ FLN appears to be more °exible than fuzzy-ART in the sense that it is possible to calibrate ¾ ¡ FLN's behavior by altering somehow the underlying positive valuation function $v_I(x)$, where I is the chain I = [0; 1] and x 2 [0; 1]. It has been shown in subsection 2.2 that any monotonically increasing function v_I is a valid positive valuation function. Nevertheless, the \best" choice among candidate positive valuations is expected to be problem dependent. On the other hand, fuzzy-ART employs (implicitly and solely) always the same positive valuation function $v_I(x) = x$. The authors of this paper hold that fuzzy-ART's learning and decision making behavior can also be calibrated by adjusting its underlying positive valuation function $v_I(x) = x$. The same arguments extend to isomorphic function $\mu_I(x)$ = 1 ¡ x for both fuzzy-ART and the ¾ ¡ FLN. Regarding fuzzy-ART in particular, an employment of a di®erent isomorphic function than $\mu_I(x)$ = 1 ¡ x will occasion a di®erent \coding technique" than fuzzy-ART's conventional \complement coding". Another aspect of ¾ ¡ FLN's °exibility is that the ¾ ¡ FLN proposes a sensible method for dealing with \missing" and \don't care" attribute values in the data as it has been explained in subsection 2.4. Such a capacity is \vital" for real world applications.

But most of all, note that the ¾ ¡ FLN appears to be overwhelmingly more versatile in principle than fuzzy-ART in the sense that ¾ ¡ FLN can handle lattice elements in addition to handling real numbers. In other words, fuzzy-ART can be applied solely in the unit Euclidean hypercube whereas the ¾ ¡ FLN is applicable to a lattice domain including fuzzy-ART's domain. However, the reader is cautioned that due to the aforementioned modi¯cations of ART's basic equations, such as the replacement of both of ART's choice (Weber) and match functions by inclusion measure ¾, the learning behavior of the ¾ ¡ FLN in the unit hypercube is not expected to be identical to fuzzy-ART's learning behavior.

6 LEARNING EXAMPLES

The capacity for learning by the ¾ ¡ FLN has been demonstrated elsewhere [39], [51], [52]. In the experiments of this work the FLNtf model has been employed solely. The learning experiments presented in this section have been carried out on benchmark data sets from the UCI repository of machine learning data sets [46]. Note that some of the benchmarks treated here by the FLNtf can not be treated, as they are, by the majority of other neural networks including ART models because the benchmarks in question are characterized by combinations of numeric and nominal data.

\Missing" attribute values in a constituent lattice L_i are denoted in a benchmark data set by a question mark \?" and they have been replaced by the least element O_{L_i} of L_i, whereas \don't care" attribute values are denoted by an asterisk *" and they have been replaced by the greatest element I_{L_i} of L_i. Recall that numeric and nominal data are allowed to intermingle freely in the FL-framework as elements of disparate constituent lattices without the need to convert one type of data to another. Note that a rigorous (mathematical) treatment of disparate types of data as suggested here constitutes an innovation, since in practice researchers typically convert one type of data to another. For instance, in the \mushroom" learning example presented with the ARTMAP in [12], nominal data have been converted to numeric data. A like conversion has been practised with both the \animal identi¯cation" data and the \DNA promoter" data presented with the Cascade ARTMAP [63]. On the other hand, for the \cylinder bands" data presented to a decision tree in [23] numeric data have been converted to nominal data.

6.1 Cleveland's HEART disease benchmark

This benchmark consists of 303 data vectors collected by Robert Detrano for the V.A. Medical Center, Long Beach and the Cleveland Clinic Foundation. This benchmark is characterized by \missing" values in a few of its data attributes. The goal is to diagnose the presence of heart disease in a patient from certain vital signs and attributes. The severity of a heart disease is denoted by an integer valued from 0 (no presence) to 4. The distribution of instances into the ¯ve classes is respectively 164, 55, 36, 35, and 13 records of data. Past experiments have only tried to distinguish between absence (value 0) from presence (values 1,2,3,4) of heart disease, the latter is referred to here as \2 categories problem". The \5 categories problem" is the classi¯cation problem which considers all 5 categories in order to diagnose not only the presence of heart disease but also its severity.

Results of processing by various algorithms have been reported for this benchmark's 14-attribute version. Table 1 summarizes the classi¯cation results by di®erent methods reported in the literature for the \2 categories problem". More speci¯cally, results by probability analysis, logistic-regression-derived discriminant function, instance-based prediction (both NTgrowth and C4), and CLASSIT conceptual clustering are reported in the documentation that accompanies this benchmark in the UCI repository [46]. Results by the ARTMAP-IC, the ARTMAP, and the K Nearest Neighbor (KNN) are detailed in [16]. Note that the aforementioned ARTMAP models have been simulated with 10 voters [16].

This benchmark has been processed by the FLNtf using di®erent training and testing sets and the corresponding results are reported in Table 2. In order to provide a good basis for comparison with the results by other methods, including a direct comparison to the ARTMAP results, a series of experiments has been carried out for the \2 categories problem", where 250 data have been randomly \kept in" for training. The previous training case has been dubbed for simplicity \keep-250-in (case)".

For the keep-250-in case, 100 experiments have been carried out and an average classi¯cation accuracy of 77.88% was recorded as shown in Table 2 with corresponding standard deviation 4.58. The average number of rules/boxes was 53.80 with standard deviation 4.58. The \5 categories problem" was treated next. In particular, for the keep-250-in case the recorded average in 100 experiments has been 56.74% (Table 2) with standard deviation 7.23, whereas the average number of rules/boxes has been 86.81 with standard deviation 4.67. In all, the performance for the \2 categories problem" has been superior than the one for the \5 categories problem", as expected, because a larger number of categories to be learned (5 versus 2 categories) increases the di±culty for recognition while it increases the number of rules/boxes. Moreover, the performance by the FLNtf for the \2 categories problem" is comparable to the best of other methods reported in Table 1.

In addition, a 10-fold cross-validation has been carried out by partitioning the ¯rst 300 data of this benchmark into 10 consecutive parts of 30 data each. One part has been employed for testing while the rest of the data in the benchmark have been employed for training. This experiment was repeated 10 times, that is each one of the ten parts has been employed for testing. Due to the odd number of data in this benchmark the last 3 data were always used for training but never for testing. The average classi¯cation accuracy of the testing data in the 10 trials has been 79.34 % with a standard deviation 5.84. The statistics of the corresponding number of rules/boxes in the 10 trials have been an average of 60 with standard deviation 3.52. The 10-fold cross-validation experiment has been repeated also for the \5 categories problem". The average testing classi¯cation accuracy in 10 trials has been 57.34 % with standard deviation 12.15. The average number of rules/boxes in the 10 trials has been 95.90 with standard deviation 3.41. Again, as expected by the FLNtf, a larger number of rules/boxes resulted in the \5 categories problem".

Further, the distribution of the testing data for both the \2 categories-" and the \5 categories-" classi¯cation problem and 10-fold cross-validation has been found in the following. Table 3 shows in a confusion matrix the distribution of the (testing) data for the \2 categories problem", whereas Table 4 shows, likewise, the distribution of the data for the \5 categories problem". By treating categories 1, 2, 3 and 4 as a single category which corresponded to a \heart patient" Table 4 reduced to Table 5, where number \94" within parentheses in Table 5 corresponds to misclassi¯cations among categories 1, 2, 3 and 4. A comparison of Table 3 and Table 5 leads to the interesting conclusion that the total number of misclassi¯cations by the FLNtf remains approximately the same, that is in particular 32+30=62 versus 23+43=66 misclassi¯cations, but the internal distribution of misclassi¯cations between \healthy subjects" and \heart patients" changes substantially from 32:30 to 23:43, respectively, in Table 3 and Table 5.

6.2 SHUTTLE landing control benchmark

The SHUTTLE benchmark is a tiny data set consisting of 15 data vectors only. This benchmark has been donated to the UCI database [46] by Bojan Cestnik of Jozef Stefan Institute of Ljubljana, Slovenia. The goal is to decide whether landing a space shuttle should be an automatic or a non-automatic task. The data consist of 6 nominal attributes plus a class label per data vector. The attributes in a data vector determine 6 conditions for space shuttle landing, and the class label determines whether landing should be automatic or manual. The meaning of the 6 nominal attributes is explained in the documentation that accompanies this benchmark [46]. About 29% of the attribute values are \don't care" and denoted in the data set by an asterisk *". Six and nine data vectors correspond to landing \by manual control" and \by automatic control", respectively. No training set is given explicitly.

In our experiments, one constituent lattice has been considered for each nominal attribute thus implying, in all, 6 constituent lattices. Note that for this particular benchmark the nominal data are totally ordered and they are represented by real numbers in the UCI database [46]; the same real numbers have been employed as positive valuation functions in the learning experiments with the

FLNtf.

The FLNtf has been employed in the leave-1-out mode. That is, one of the 15 data vectors was left out for testing, and the remaining 14 data vectors have been employed for training. The experiment was repeated leaving, in turn, all data out for testing. Several di®erent linear positive valuations have been tried heuristically and the best results have been obtained for the linear coe±cients $(c_1; c_2; c_3; c_4; c_5; c_6) = (1; 2; 1; 1; 1; 1)$; in the latter experiment 1 and 2 data have been misclassi¯ed from the classes \automatic landing" and \manual landing", respectively. Note that for the linear coe±cients $(c_1; c_2; c_3; c_4; c_5; c_6) = (1; 1; 1; 1; 1; 1)$ exactly 4 misclassi¯cations have been recorded. In di®erent experiments, the number of tightest ¯ts (rules) calculated for class \automatic control" has been between 2 and 4, whereas the corresponding number of tightest ¯ts (rules) for class \manual control" has been between 3 and 5.

In conclusion, the FLNtf has demonstrated a capacity to generalize from a limited number of 14 data with nominal attributes and it has been able to decide successfully on 12 out of 15 data vectors, in other words 80% of new and hitherto unknown totally ordered nominal data have been classi¯ed correctly.

6.3 CYLINDER BANDS benchmark

A number of neural networks could have been employed with the previous two examples provided that the neural networks in question can deal with both \missing" and \don't care" attribute values. This is due to the fact that the attribute values of the constituent lattices in the previous examples, be it numeric or nominal, are totally ordered therefore those attribute values could be replaced by real numbers. Nevertheless this is not the case in the following example, where numeric data are intermingled with non-\totally ordered" nominal data.

The CYLINDER BANDS benchmark data set [46] consists of 540 data records and it has been created by Bob Evans for the RR Donnelly & Sons Co., Gallatin, Tennessee. There exist 20 numeric plus 20 nominal attributes per data vector. The \class" attribute is included in the nominal data and it speci¯es one of two classes, these are the \band" and \noband" classes. In all, there exist 228 and 312 data records for the \band" and the \noband" classes, respectively. About 4.75 % of the overall data attributes are \missing" and denoted in the data set by a question-mark \?". No training set is given explicitly.

A qualitative description of the results obtained by processing similar data with a decision tree induction mechanism embedded within a knowledge acquisition system called \Apos" is shown in [23] but without detailing the corresponding performance quantitatively. Only in one instance it is cited in [23]: \Apos rules predicted seven of the eleven banding incidents in September 1993". In addition [23] does not specify whether the UCI benchmark has been used at all in their experiments. Finally note that in [23] the numeric data have been converted to nominal data by interval subdivision.

The CYLINDER BANDS data set of the UCI collection of benchmarks has been processed by treating numeric attribute values as numbers and nominal attribute values as elements of the corresponding abstract space in a probability space. In particular the constituent lattice corresponding to a nominal feature has been de¯ned as a probability space with (1) abstract space, the set of all (¯nitely many) values for a nominal feature, and (2) probability measure (on the power set of the underlying abstract space), de¯ned as the frequency of occurrence of the corresponding attribute values.

The CYLINDER BANDS data have been partitioned into a training-testing pair of data sets in six di®erent ways and the corresponding results are reported in the six lines of Table 6. The ¯rst pair of training/testing data sets has been obtained by selecting the ¯rst datum of every 10 data in the CYLINDER BANDS benchmark for training and leaving the rest nine data out for testing. The next four training/testing pairs of data sets have been obtained by selecting the ¯rst datum of every 5,4,3 and 2 data respectively for training. The sixth training/testing pair of data sets, which correspond to line 6 of Table 6, has been obtained by selecting the ¯rst and second datum of every 3 data for training and leaving the third datum out for testing. Likewise, as in the previous two examples of this section, a tightest ¯t has been interpreted as a \rule". In other words, the FLNtf employs the training data to ¯nd tightest ¯ts, those are lattice intervals [a; b] where a and b consist of both numeric and nominal data. Such a tightest ¯t can then be interpreted as a \rule" in the sense that when a hybrid datum, say datum x, gives the largest value ¾([x; x] $_L$ [a; b]) for tightest ¯t [a; b] then it is concluded that the hybrid datum x is of the same category as the category of all the training data which have given rise to interval/rule R = [a; b].

Table 6 reports experimental results where only a part of the data has been used for training, hence it con¯rms FLNtf's capacity for generalization. That is, a testing datum does not have to be within a tightest ¯t in order for the FLNtf to assigned it to a category, but a testing datum could be outside all tightest ¯ts and still the FLNtf can assign it pointedly to its correct category as it has been veri¯ed experimentally. In all, the FLNtf has demonstrated a very good capacity for generalization which (capacity) was increasing as the number of the data for training was increasing as shown in Table 6. Moreover, Table 6 reports the total number of tightest ¯ts (rules) in each experiment. An increase is noted in the number of tightest ¯ts as the number of training data increased, nevertheless the total number of tightest ¯ts is fairly small compared to the corresponding number of training data in each experiment.

6.4 FOREST COVERTYPE benchmark

The size of the previous data sets has been rather small compared to potential requirements of real world problems. Therefore a much larger data set has been considered in this section in order to address the issue of FLNtf's scalability and applicability to real world problems. The FOREST COVERTYPE data set [46] contains 581,012 data records of whom 11,340 records are given for training, 3,780 are given for validation, and 565,892 data records are given for testing. This data set has been donated by Jock A. Blackard, Colorado State University, and it involves jointly 12 numeric and nominal attributes per data record. A data record has stemmed from a 30£30 meter cell in the Roosevelt National Forest of northern Colorado. The ¯rst 10 attributes are numeric cartographic variables which specify the location of the 30£30 meter cell of forest, whereas the last 2 attributes are Boolean strings of zeros and ones and specify both the corresponding wilderness area and soil type. The \class" of a data record is denoted by an integer between 1 and 7, that is there exist seven underlying classes which specify the forest cover type. There are no missing attribute values.

Previous uses of this benchmark have employed linear discriminant analysis as well as backpropagation [3] and the corresponding results are summarized in Table 7. The FLNtf has been employed for learning, and the numeric attributes have been treated as numbers whereas for each nominal attribute the corresponding Boolean lattice of zeros and ones has been considered. Note that for such a Boolean lattice the positive valuation function employed has been the normalized sum of ones in a string of zeros and ones. Note that only the \training data" have been used for learning by FLNtf, resulting in a classi¯cation accuracy equal to 62.58 % on the testing data as well as 3684 tightest ¯ts (rules), that is a 3.07:1 compression on the training data. Table 8 displays a confusion matrix, where the latter matrix shows the distribution of the 565,892 testing data in the 7 categories learned by the FLNtf.

We remark that no problem of over¯tting the training data has been noted in this work because the results on the testing sets have been very satisfactory in all the experiments.

6.5 A comment on FLN's potential for data discrimination

The FLN models, be it the ¾ ¡ FLN or the FLNtf, have demonstrated a capacity for data discrimination. The latter capacity is attributed to the practical e±ciency of the inclusion measure ¾ and the technique of maximal expansions [51], [52]. Nevertheless another reason should be pointed out as well, that is an FLN model's capacity to deal with mathematical lattices. Note that it has been a practice in the past to introduce extra dimensions to nonlinearly separable classification problems "hoping" to achieve linear separability in more dimensions. For instance Pao's functional link net (unfortunately after having launched the acronym FLN we have discovered that the same acronym had been used for the functional link net, as well) creates automatically new dimensions by introducing products of the feature values (see "tensor models" in [45], p93). Note that the FL-framework suggests an additional "tool" for pattern separability by mapping feature values on the partially ordered constituent lattices instead of mapping features on the totally ordered real line. Such a map of disparate features to different constituent lattices may be regarded as a preprocessing step aiming at a more efficient data separability. The experiments in this work corroborate an enhanced efficiency for data separation.

7 DISCUSSION & CONCLUSION

It has been argued by mathematicians that "lattice theory will play a leading role in the mathematics of the twenty-first century" [55]. This work has demonstrated, by experimenting with a concrete neural model on a number of benchmark data sets, that lattice theory can be useful as well for versatile-automated- learning and decision making.

An assortment of issues have been dealt with in this paper including a novel notation for the framework of fuzzy lattices (FL-framework) in order to simplify mathematical expressions, and hence to enhance their utility, without losing content. Neurocomputing models in the context of FL-framework are labelled by acronym "FLN". In this paper two FLN models have been dealt with, these are the $\sigma - FLN$ model for competitive clustering and the "FLN with tightest fits ($FLNtf$)" for supervised clustering. Convenient geometric interpretations have been given on the plane.

The $\sigma - FLN$, introduced elesewhere [51], [52], has been presented here as an enhanced extension of the fuzzy-ART model in the FL-framework as explained in section 5. Despite $\sigma - FLN$'s rapid learning in "one pass" through the data its performance depends on the order of data presentation. Because some applications might call for a learning capacity independent of the order of data presentation, the $FLNtf$ model has been introduced in this work which, while retaining all of $\sigma - FLN$'s aforementioned advantages, can compute incrementally the same intervals in the training data independent of the order of presentation at the "price" of polynomial complexity $O(n^3)$, where n is the number of training data. Nevertheless note that polynomial $O(n^3)$ complexity is not an actual problem in applications, therefore the $FLNtf$ can have a practical value. The $FLNtf$ has been tested successfully here on four benchmark data sets of various sizes containing either (or both) numeric and nominal values with "missing" and "don't care" values. Moreover, the $FLNtf$ can justify its answers by extracting rules in the data, hence the $FLNtf$ can be useful for data mining and rule extraction involving disparate types of data. Future work on $FLNtf$ includes the study of techniques in order to both improve performance and reduce the number of rules. Another topic for future work includes the study of faster algorithms to calculate the tightest fits in the training data.

Due to its wide scope of applicability, the FL-framework could be employed as a platform for unifying and/or enhancing various learning techniques across different disciplines. For example, the notion "lattice interval" could provide with novel perspectives and even enhance such rule-generation algorithms as R-MINI [33]. Modelling "consciousness" in psychology might be another domain for applications due to FL-framework's capacity to deal with disparate types of data such as propositional statements, real numbers, fuzzy sets, symbols, etc. For instance, in an approach to modelling consciousness [48] the conditional probability $P(\overrightarrow{b}_2/\overrightarrow{b}_1) = \frac{P(\overrightarrow{b}_2 \cap \overrightarrow{b}_1)}{P(\overrightarrow{b}_1)}$ is used for the "binding" of two patterns $\overrightarrow{b}_1$ and $\overrightarrow{b}_2$. It would be interesting to study the efficiency of inclusion measure $k(\overrightarrow{b}_1 \subseteq \overrightarrow{b}_2) = \frac{P(\overrightarrow{b}_2)}{P(\overrightarrow{b}_2 \cup \overrightarrow{b}_1)}$, since, on the one hand, $\overrightarrow{b}_2 \cap \overrightarrow{b}_1 = \emptyset$ implies $P(\overrightarrow{b}_2/\overrightarrow{b}_1) = 0$, on the other hand, $\overrightarrow{b}_2 \cap \overrightarrow{b}_1 = \emptyset$ does not imply $k(\overrightarrow{b}_1\ \mu\ \overrightarrow{b}_2) = 0$ and the latter could make the di®erence in decision making. Regarding another model of consciousness which employs the notion of `bubbles' of activity [62], note that the \lattice domain" applicability of an FLN model might be particularly useful in modelling the higher stages of consciousness after the consciousness' emergence from sensory data transformations in lower stages. By the same token, the FLN could be the platform for launching hybrid neural models whose utility has been argued in [59]. Moreover note that FL-framework's mathematical tools, including its positive-valuation-based metric, might allow for a rigorous mathematical treatment of such learning issues as \convergence in the limit", etc.

Furthermore note that the most widely employed form of today's computing is based on Boolean logic and it is implemented on silicon electronics. Nevertheless \silicon electronics computing" is not unrivaled for all computing tasks. For instance due to its processing speed, optical computing is more suitable for Fast Fourier Transforms. The synergistic coexistence of optical- with electronics- computing has been argued for in [17]. We remark that the collection of images on a plane de¯nes a mathematical lattice. In conclusion, both the Boolean algebra, implemented on silicon electronics, and the images, used in optical computing, imply a mathematical lattice. Alternative mathematical lattices can be considered in applications. Therefore the FL-framework could be considered as a useful platform for computing in practice.

Regarding neurocomputing in fuzzy lattices (FLN) in particular, note that in order to implement an FLN model in hardware the so called \representation problem" must be resolved. That is the problem of how to store and process a lattice element. It is likely that the best implementation for an FLN model is an analog rather than a digital implementation if one is to take full advantage of FLN's capacity for massive parallel processing of disparate types of data. Perhaps the key to \brain-like" behavior by arti¯cial neural networks lies as well in the sound, synergistic combination of disparate types of data as exempli¯ed by FLN models.

8 ACKNOWLEDGEMENTS

The authors wish to acknowledge the kind assistance of Denis J. Dean from Colorado State University for information on processing the FOREST COVERTYPE benchmark by conventional techniques such as Linear Discriminant Analysis and Backpropagation.

References

[1] Anderberg, M. (1973). Cluster analysis for applications. New York: Academic Press.

[2] Birkho®, G. (1967). Lattice Theory. Providence, RI: American Mathematical Society, Colloquium Publications, 25.

[3] Blackard, J.A., & Dean, D.J. (2000). Comparative accuracies of arti¯cial neural networks and discriminant analysis in prediciting forest cover types from cartographic variables. Computers and Electronics in Agriculture. 24(3), 131-151.

[4] Blume, M., & Esener, S.C. (1997). An E±cient Mapping of Fuzzy ART onto a Neural Architecture. Neural Networks, 10(3), 409-411.

[5] Blumer, A., Ehrenfeucht, A. Haussler, D., & Warmuth, M.K. (1989). Learnability and the Vapnik-Chervonenkis dimension. Journal of the ACM, 36(4), 929-965.

[6] Burke, L.I. (1991). Clustering Characterization of Adaptive Resonance. Neural Networks, 4(4), 485-491.

[7] Carpenter, G.A. (1997). Distributed Learning, Recognition, and Prediction by ART and ARTMAP Neural Networks. Neural Networks, 10(8), 1473-1494.

[8] Carpenter, G.A., & Grossberg, S. (1987). A massively parallel architecture for self-organizing neural pattern recognition machine. Computer Vision, Graphics and Image Understanding, 37, 54-115.

[9] Carpenter, G.A., & Grossberg, S. (1987). ART 2: Self-organization of stable category recognition codes for analog input patterns. Applied Optics, 26, 4919-4930.

[10] Carpenter, G.A., & Grossberg, S. (1990). ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. Neural Networks, 3, 129-152.

[11] Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds J.H., & Rosen, D.B. (1992). Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multi-dimensional Maps. IEEE Transactions on Neural Networks, 3(5), 698-713.

[12] Carpenter, G.A., Grossberg, S., & Reynolds, J.H. (1991). ARTMAP: Supervised Real-Time Learning and Classi¯cation of Nonstationary Data by a Self-Organizing Neural Network. Neural Networks, 4(4), 565-588.

[13] Carpenter, G.A., Grossberg, S., & Reynolds, J.H. (1995). A Fuzzy ARTMAP Nonparametric Probability Estimator for Nonstationary Pattern Recognition Problems. IEEE Transactions on Neural Networks, 6(6), 1330-1336.

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

[15] Carpenter, G.A., Grossberg, S., & Rosen D. (1991). ART 2-A: An Adaptive Resonance Algorithm for Rapid Category Learning and Recognition. Neural Networks, 4(4), 493-504.

[16] Carpenter, G.A., & Markuzon, N. (1998). ARTMAP-IC and medical diagnosis : Instance counting and inconsistent cases. Neural Networks, 11(2), 323-336.

[17] Caul¯eld H.J. (1998). Perspectives in Optical Computing. Computer, February 1998, 22-25.

[18] Chen, S.L. (1996). Urysohn-Closedness on Fuzzy Lattices. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 4(1),87-93.

[19] Dagher, I., Georgiopoulos, M., Heileman, G.L., & Bebis, G. (1999). An Ordering Algorithm for Pattern Presentation in Fuzzy ARTMAP That Tends to Improve Generalization Performance. IEEE Transactions on Neural Networks, 10(4), 768-778.

[20] Davey, B.A., & Priestley H.A. (1990). Introduction to Lattices and Order. Cambridge University Press.

[21] Doob, J.L. (1990). Stochastic Processes. John Wiley & Sons, Inc., Copyright 1953.

[22] Edmonds, E.A. (1980). Lattice Fuzzy Logics. International Journal of Man-Machine Studies, 13(4), 455-465.

[23] Evans, B., & Fisher, D. (1994). Overcoming process delays with decision tree induction. IEEE Expert, February 1994, 60-66.

[24] Fung, C.F., Billings S.A, & Luo, W. (1996). On-line Supervised Adaptive Training Using Radial Basis Function Networks. Neural Networks, 9(9),1597-1617.

[25] Gaines, B.R. (1978). Fuzzy and Probability Uncertainty Logics. Information and Control, 38, 154-169.

[26] Georgiopoulos, M., Fernlund, H., Bebis, G., & Heileman G.L. (1996). Order of Search in Fuzzy ART and Fuzzy ARTMAP: E®ect of the Choice Parameter. Neural Networks, 9(9), 1541-1559.

[27] Goguen , J.A. (1967). L-Fuzzy Sets. Journal of Mathematical Analysis and Applications, 18, 145-174.

[28] Grossberg, S. (1998). Birth of A Learning Law. INNS/ENNS/JNNS Newsletter: Appearing with Volume 11(1) of Neural Networks.

[29] Grossberg, S. (1980). How does a brain build a cognitive code ?. Psychological Review, 87, 151.

[30] Grossberg, S. (1976). Adaptive pattern classi¯cation and universal recoding. II: Feedback, expectation, olfaction, and illusions. Biological Cybernetics, 23, 187-202.

[31] Halmos, P.R. (1961). Naive Set Theory.Van Nostrand Co., N.Y.

[32] Healy, M.J., and Caudell, T.P. (1997), Acquiring Rule Sets as a Product of Learning in a Logical Neural Architecture. IEEE Transactions on Neural Networks, 8(3), 461-474.

[33] Hong, S.J. (1997). R-MINI: An Iterative Approach for Generating Minimal Rules from Examples. IEEE Transactions on Knowledge and Data Engineering, 9(5), 709-717.

[34] Huang, J., Georgiopoulos M., & Heileman G.L. (1995). Fuzzy ART Properties. Neural Networks, 8(2), 203-213.

[35] J¿rgensen, T.M., & Linneberg C. (1999). Theoretical Analysis and Improved Decision Criteria for the n-Tuple Classi¯er. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(4), 336-347.

[36] Kaburlasos, V.G., Egbert D.D., Rao, M. (1991). A Hardware Implementation of the Adaptive Resonance Theory Neural Network. In Proceedings of the First Golden West International Conference on Intelligent Systems (pp.21-28). Reno, NV: Computer Science Department.

[37] Kaburlasos, V.G. (1992). Adaptive Resonance Theory with Supervised Learning and Large Database Applications, Ph.D. Dissertation, Dept. Electrical Engineering, University of Nevada, Reno.

[38] Kaburlasos, V., Petridis, V., Allotta, B., & Dario, P. (1997). Automatic Detection of Bone Breakthrough in Orthopedics by Fuzzy Lattice Reasoning (FLR) : The Case of Drilling in the Osteosynthesis of Long Bones. In Proceedings of the Mechatronical Computer Systems for Perception and Action (MCPA97) (pp. 33-40). Pisa Italy.

[39] Kaburlasos, V.G., & Petridis, V. (1997). Fuzzy Lattice Neurocomputing (FLN): A Novel Connectionist Scheme for Versatile Learning and Decision Making by Clustering. International Journal of Computers and Their Applications, 4(3), 31-43.

[40] Kaburlasos, V.G., Petridis, V., Brett, P., & Baker, D. (1998). Learning a Linear Association of Drilling Pro¯les in Stapedotomy Surgery. In Proceedings of the IEEE 1998 International Conference on Robotics & Automation (ICRA98) (pp. 705-710). Leuven, Belgium.

[41] Kearns, M.J., &Vazirani, U.V. (1994). An Introduction to Computational Learning Theory. The MIT Press.

[42] Lim, C.P., & Harrison R.F. (1997). An Incremental Adaptive Network for On-line Supervised Learning and Probability Estimation. Neural Networks, 10(5), 925-939.

[43] Lin, C.-J., & Lin, C.-T. (1997). An ART-Based Fuzzy Adaptive Learning Control Network. IEEE Transactions on Fuzzy Systems, 5(4), 477-496.

[44] Likhovidov, V. (1997). Variational Approach to Unsupervised Learning Algorithms of Neural Networks. Neural Networks, 10(2), 273-289.

[45] Looney, C.G. (1997). Pattern Recognition Using Neural Networks. New York, Oxford: Oxford University Press.

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

[47] Moray, N. (1987). Intelligent aids, mental models, and the theory of machines, International Journal of Man-Machine Studies, 27, 619-629.

[48] Pelaez, J.R. (1997). Plato's Theory of Ideas Revisited. Neural Networks, 10(7), 1269-1288.

[49] Petridis, V., Kaburlasos, V.G., Brett, P., Parker, T., & Day, J.C.C. (1996). Two Level Fuzzy Lattice (2L-FL) Supervised Clustering : A New Method for Soft Tissue Identi¯cation in Surgery. In Proceedings of the CESA'96 IMACS Multiconference (pp. 232-237). Lille France.

[50] Petridis, V., & Kaburlasos, V.G. (1996). FLN: A Fuzzy Lattice Neural Network Scheme for Clustering. In Proceedings of the World Congress on Neural Networks 1996 (WCNN96) (pp. 942-945). San Diego, CA: International Neural Network Society Press.

[51] Petridis, V., & Kaburlasos, V.G. (1998). Fuzzy Lattice Neural Network (FLNN) : A Hybrid Model for Learning. IEEE Transactions on Neural Networks, 9(5), 877-890.

[52] Petridis, V., & Kaburlasos, V.G. (1999). Learning in the Framework of Fuzzy Lattices, IEEE Transactions on Fuzzy Systems, 7(4), 422-440.

[53] Raijmakers, M.E.J., & Molenaar P.C.M. (1997). Exact ART: A Complete Implementation of an ART Network. Neural Networks, 10(4), 649-669.

[54] Rohwer, R., & Morciniec, M. (1998). The Theoretical and Experimental Status of the n-tuple Classi¯er. Neural Networks, 11(1), 1-14.

[55] Rota, G.C. (1997). Memorial Tributes to Garett Birkho®. Notices of the American Mathematical Society, 44(11). (http://www.ams.org/notices/199711/comm-rota.html)

[56] Roy, A. (1998). Personal Communication regarding the panel discussion at the World Congress on Computational Intelligence (WCCI'98) on the theme \could there be real-time, instantaneous learning in the brain ?". Anchorage, Alaska.

[57] Rutherford, D.E. (1965). Introduction to Lattice Theory. Oliver and Boyd Ltd., Edinburgh, Great Britain.

[58] Sahami, M. (1995). Learning Classi¯cation Rules Using Lattices. In Proceedings of the Eightth European Conference on Machine Learning (ECML-95). Heraklion, Crete, Greece.

[59] Sun, R. (1997). Learning, Action and Consciousness: A Hybrid Approach Toward Modelling Consciousness. Neural Networks, 10(7), 1317-1331.

[60] Serrano-Gotarredona, T., & Linares-Barranco, B. (1997). An ART1 Microchip and Its Use in Multi-ART1 Systems. IEEE Transactions on Neural Networks, 8(5), 1184-1194.

[61] Specht, D.F. (1990). Probabilistic neural networks. Neural Networks, 3, 109-118.

[62] Taylor, J.G. (1997). Neural Networks for Consciousness. Neural Networks, 10(7), 1207-1225.

[63] Tan, A.H. (1997). Cascade ARTMAP: Integrating Neural Computation and Symbolic Knowledge Processing. IEEE Transactions on Neural Networks, 8(2), 237-250.

[64] Tsay, S.W., & Newcomb, R.W. (1991). VLSI Implementation of ART1 Memories. IEEE Transactions on Neural Networks, 2(2), 214-221.

[65] Wann C.-D., & Thomopoulos, S.C.A. (1997). A Comparative Study of Self-organizing Clustering Algorithms Dignet and ART2. Neural Networks, 10(4), 737-753.

[66] Williamson, J.R. (1996). Gaussian ARTMAP: A Neural Network for Fast Incremental Learning of Noisy Multidimensional Maps. Neural Networks, 9(5), 881-897.

[67] Zimmermann, H.J. (1991). Fuzzy Set Theory and Its Applications. Kluwer Academic Publishers, Norwell, MA.

TABLE 1
Performance of various methods reported in the literature in classifying the Cleveland's HEART Benchmark. The methods are arranged in a decreasing order of success.

Pattern Classi¯cation Algorithm	% Classi¯cation Accuracy
Probability Analysis	79
Conceptual Clustering (CLASSIT)	78.9
ARTMAP-IC	78
Discriminant Analysis	77
Instance Based Prediction (NTgrowth)	77
Instance Based Prediction (C4)	74.8
Fuzzy ARTMAP	74
KNN	67

TABLE 2
Performance of FLNtf in classifying the Cleveland's HEART Benchmark. The results are arranged in a decreasing order of success.

Pattern Classi¯cation Algorithm	Statistics of % Classi¯cation Accuracy		Statistics of Number of Rules	
	Average	Standard Deviation	Average	Standard Deviation
FLNtf, 2 Categories (10-fold Cross-Validation) (10 experiments)	79.34	5.84	60	3.52
FLNtf, 2 Categories. Keep-250-in (100 experiments with random data orderings)	77.88	4.58	53.80	4.58
FLNtf, 5 Categories (10-fold Cross-Validation) (10 experiments)	57.34	12.15	95.90	3.41
FLNtf, 5 Categories. Keep-250-in (100 experiments with random data orderings)	56.74	7.23	86.81	4.67

TABLE 3
Confusion matrix for the Cleveland's HEART Benchmark in the \2-categories problem" and 10-fold cross-validation.

Number of Data from Category	Number of Data to Category 0	1
0	131	32
1	30	107

TABLE 4

Confusion matrix for the Cleveland's HEART Benchmark in the \5-categories problem" and 10-fold cross-validation.

Number of Data	Number of Data to Category				
from Category	0	1	2	3	4
0	140	15	4	4	0
1	29	12	9	2	2
2	8	11	9	8	0
3	5	10	9	7	3
4	1	1	5	2	4

TABLE 5

Confusion matrix for the Cleveland's HEART Benchmark in the \5-categories problem" and 10-fold cross-validation. Category 0 (healthy subjects) remained intact. Categories 1,2,3,4 (heart patients) were arranged in a single category. In that arrangement the 94 misclassi¯cations among the categories 1,2,3,4 are no longer considered to be misclassi¯cations.

Number of Data	Number of Data to Category	
from Category	0	1
0	140	23
1	43	(94)

TABLE 6
The testing success percentage is shown on 6 di®erent partitions of the CYLINDER BANDS benchmark into a training set of an ever increasing size and a testing set. The last column shows the number of tightest ¯ts (rules) found in the training data.

	# Data (band+noband)		Testing Misclassi¯cations (band+noband)	% Success	# Clusters/Rules
	Training	Testing			
1	23+31= 54	205+281=486	113 + 39	68.72	2
2	46+62=108	182+250=432	63 + 50	73.84	2
3	58+77=135	170+235=405	74 + 27	75.06	3
4	78+102=180	150+210=360	21 + 97	67.22	3
5	114+156=270	114+156=270	46 + 8	80.00	15
6	148+212=360	80+100=180	29 + 9	78.88	10

TABLE 7

Performance of various methods in classifying the FOREST COVERTYPE Benchmark. The methods are arranged is a decreasing order of success.

Pattern Classi¯cation Method	% Classi¯cation Accuracy
Backpropagation	70
F LN tf	62.58
Linear Discriminant Analysis	58

TABLE 8

Confusion matrix for the FOREST COVERTYPE testing data. Only the \training data set" has been employed for learning.

Number of Data	Number of Data to Category						
from Category	1	2	3	4	5	6	7
1	126,541	45,235	150	0	3,766	730	33,258
2	64,403	169,859	8,410	195	22,297	8,104	7,873
3	26	954	24,895	2,023	507	5,189	0
4	0	0	53	524	0	10	0
5	446	1,281	145	0	5,311	148	2
6	111	493	3,765	664	387	9,787	0
7	1,031	103	0	0	1	0	17,215

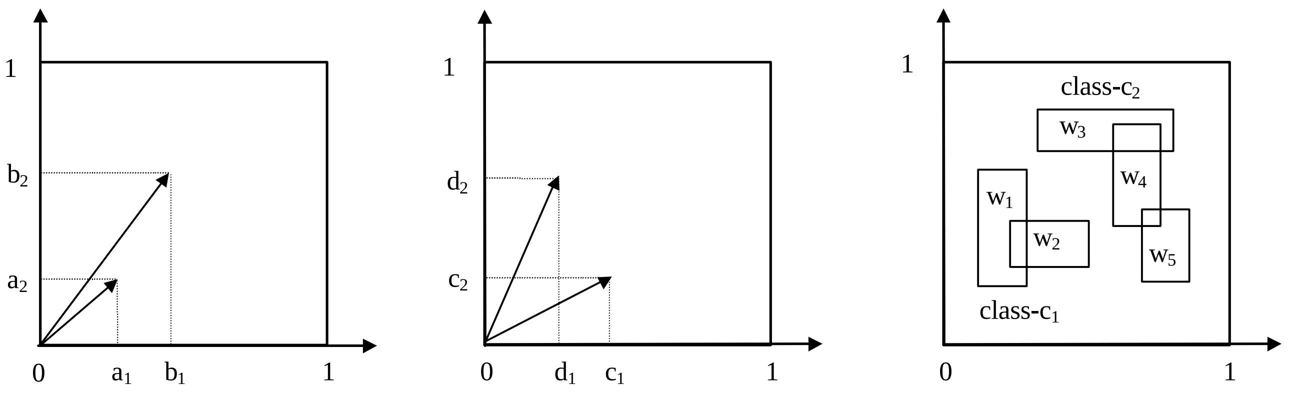


Figure 1: (a) Points (a1,a2) and (b1,b2) are comparable, in particular it is (a1; a2) u (b1; b2). (b) Points (c1,c2) and (d1,d2) are incomparable, that is neither (c1; c2) u (d1; d2) nor (d1; d2) u (c1; c2). (c) Overlapping boxes were put in the same class, but it could be otherwise.

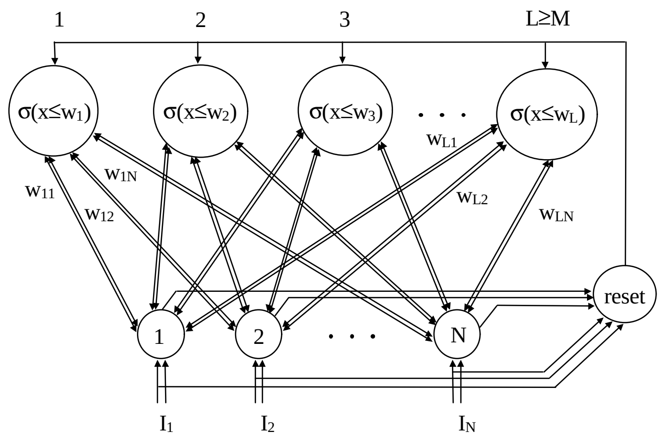
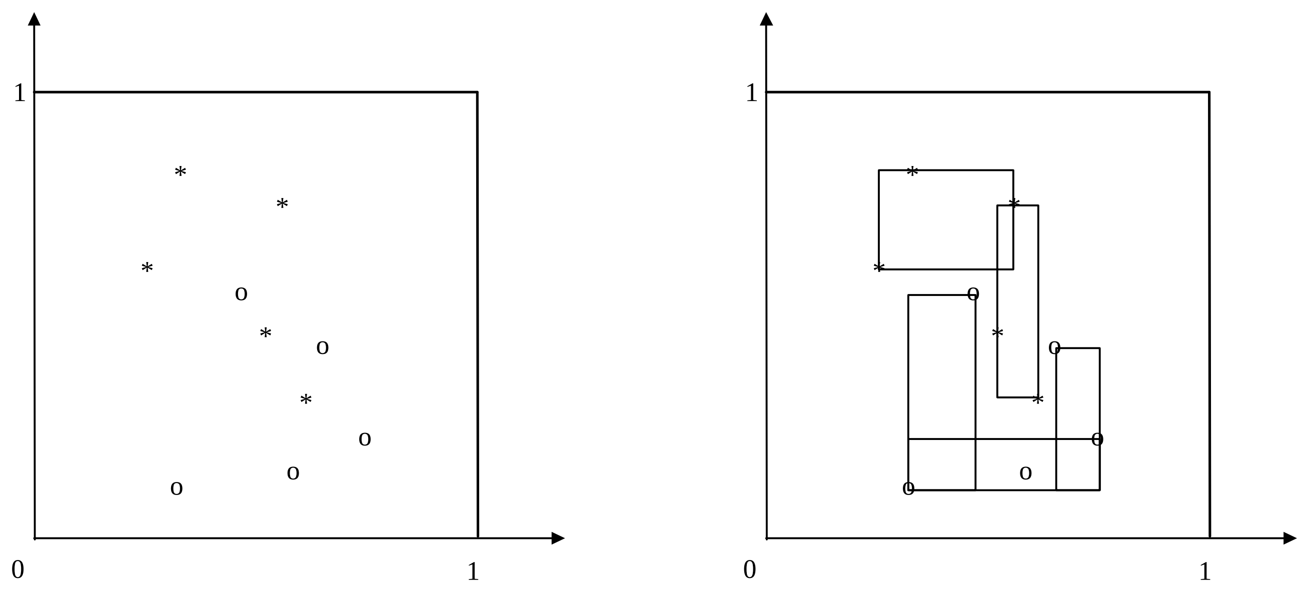


Figure 2: The two layer ¾ ¡ F LN architecture for competitive clustering in the lattice ¿(L) of intervals. L, is the number of Category Layer neurons which equals the total number of intervals used to de¯ne M classes. A Category Layer neuron employs a lattice inclusion measure s as an activation function. N, is the number of the Input Layer neurons. The two layers are fully interconnected by pairs of lattice weighted links that ¯lter -up or -down the activity in a layer. A \reset" node is used for resetting the activity of a node is the Category Layer.



Fi ur 3: a) iv di er nt to s a e g ve fr m e ch ne f t o d st nc ca eg ri s d no ed es ec iv ly y \ " a d \ ". b) he ol ec io of he ig te t ¯ s t at or es on to he iv n a om (t iv al nt rv ls .

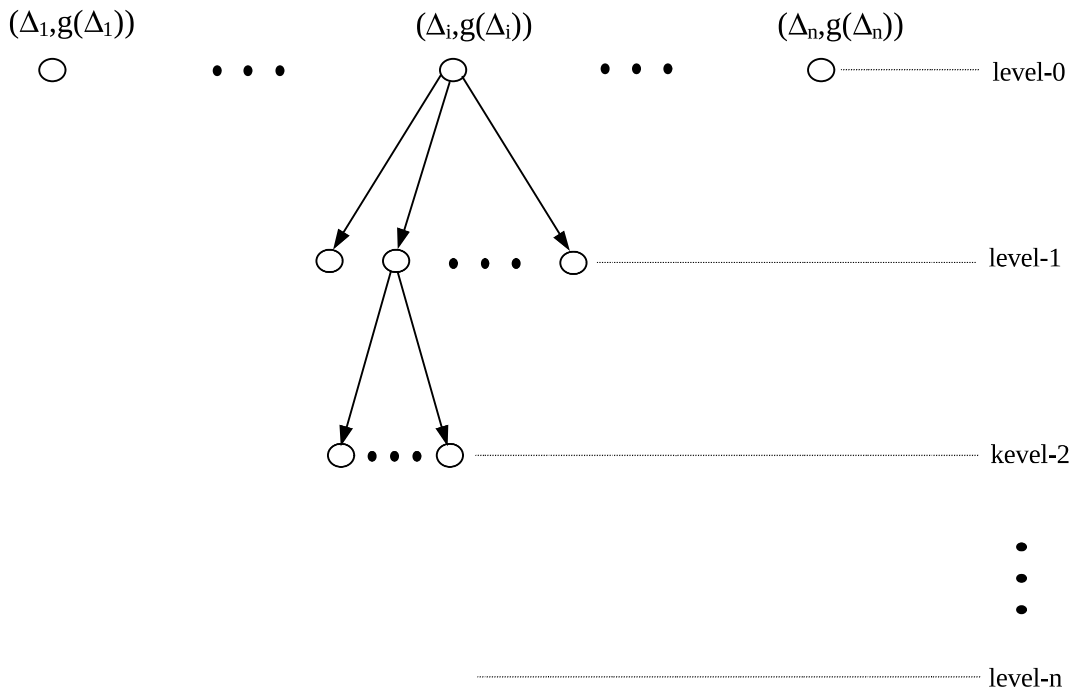
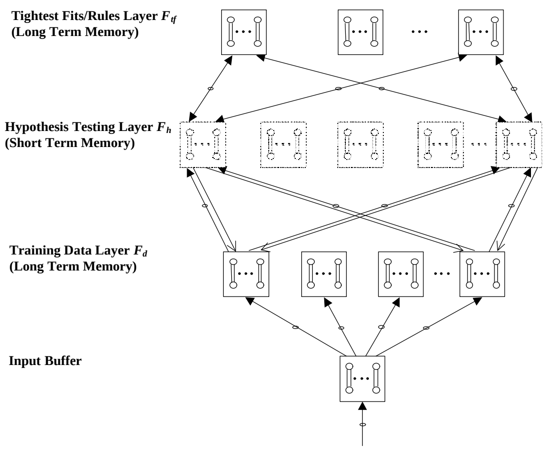
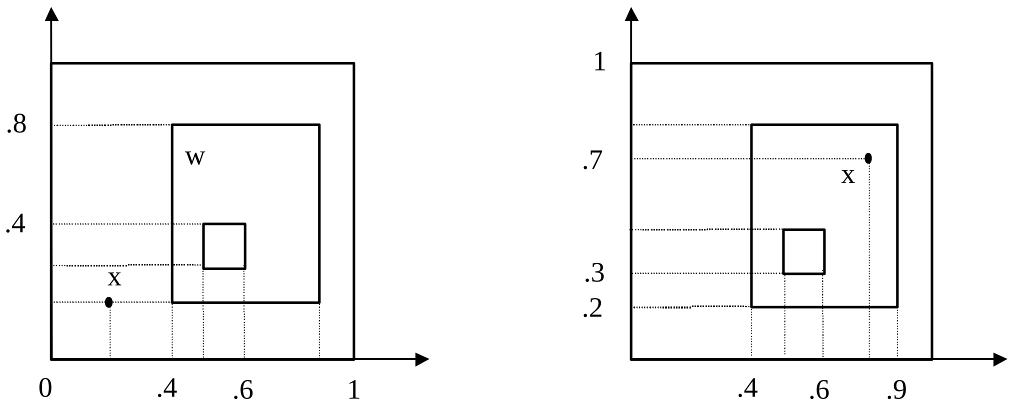


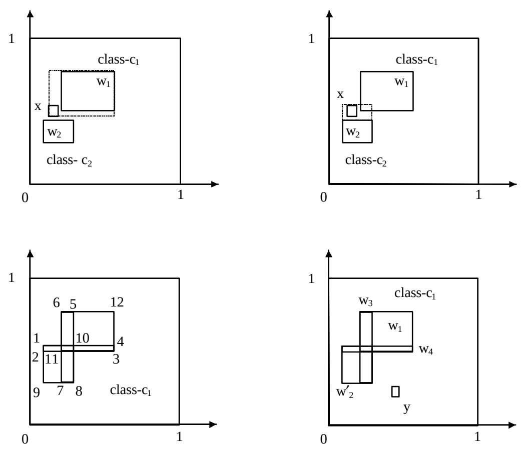
Figure 4: The above tree structure is calculated by the FLNtf algorithm in order to ¯nd the tightest ¯ts on a training data set $f(¢_i; g(¢_i))g_{i2f1;:::;ng}$. A node of the tree corresponds to the lattice join ($\__L$) of a category's subset. Tree-nodes that contradict training data do not grow further. A \leaf" of the tree corresponds to a tightest ¯t. The total number (n) of the training data upper-bounds the size of the tree.


Tightest Fits/Rules Layer F_{tf}
(Long Term Memory)
Hypothesis Testing Layer F_h
(Short Term Memory)
Training Data Layer F_d
(Long Term Memory)
Input Buffer

Fi ur 5: he hr e l ye F L tf eu al rc it ct re A n de f a ay r, en te by re ta gl , i a ¾ F L mo el Th no es n l ye s F an F_t ar dr wn y s li li es nd he co st tu e t e s st m' lo g t rm em ry Th no es n l ye F_h re ra n b do te li es nd he co st tu e t e s st m' sh rt er me or . T e i te co ne ti g l nk ar us d f r p ss ng nf rm ti n a on th ¾ ¡ LN s) Th we gh s o th ¾ ¡ LN s) ee , i e® ct th kn wl dg . T e s or te m m mo y e is s o ly ur ng le rn ng , w er as he on te m m mo y i pe ma en . L ye F_d to es ne by on al th in om ng ra ni g d ta la er $_{tf}$ ee s t e e er pd te ti ht st ts la er $_h$ i ac iv on y d ri g l ar in an it al ul te th no es n a ev l o F L tf lg ri hm s t ee tr ct re Fi .4 .



ig re : A il us ra io of n i cl si n m as re s C ns st nc Pr pe ty u w) (x $_U$ u ¾ x w) wh n x s a at m ( ) x (0 2, .2 , a d ( ) x (0 8, .7 .



F gu e 7 Il us ra in ¾ ¡ LN s o er ti n, s w ll s t e t ch iq e o ma im l e pa si ns (a Cl ss c_1 in th co pe it on ro cl ss c_2 ve th in ut bu cl ss c_1 s \ es t" ec us it ai s t e \ ss mi at on on it on , ( ) C as -c is el ct d a th ne wi ne th t m et th \a si il ti n c nd ti n" (c Cl st r (nt rv l) 0_2 x _ w_2 ep ac s c us er $_2$. he ve la pi g o cl st rs $_1$ a d w tr gg rs he ec ni ue f m xi al xp ns on wh ch ro uc s t o m re lu te s, ho e a e t e w an w_4 an (d A n w i pu in er al tu ns p a d t e l ar in cy le f t e ¾ F L re um s.

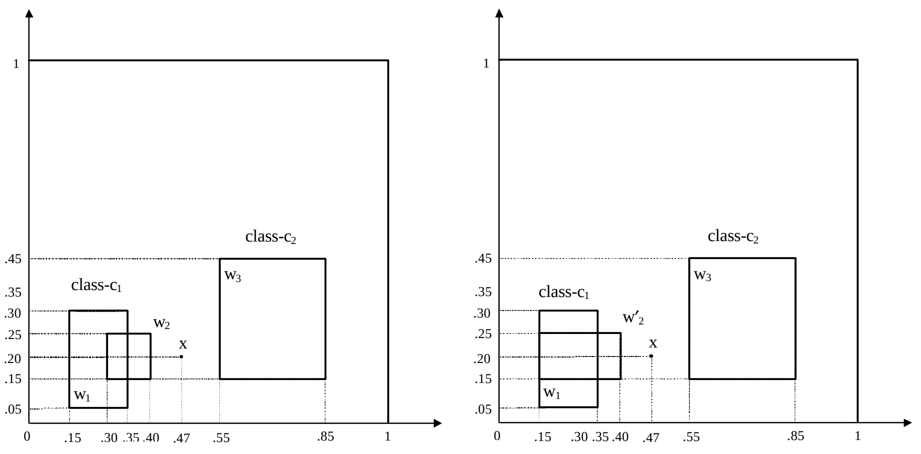


Figure 8: (a) The trivial interval (atom) x is included more in class-c_2 than in class-c_1. (b) The technique of maximal expansions is an optimization technique which replaces the family decomposition fw_1; w_2g of class c_1 = w_1 [w_2 by the quotient family decomposition Q(c) = fw_1; $w0_2$g of class c_1 = w_1 [w^0_2 in order to maximize the degree of inclusion of any interval to class c_1. As a result, in this particular example, x is now included more in class-c_1 than it is in class-c_2.

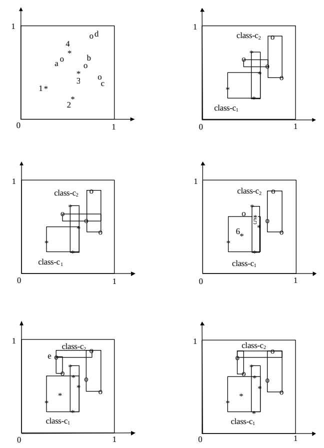


Figure 9: Illustrating "learning" by the FLNtf model. (a) Eight labeled training data partitioned, by four, in two classes are fed to the FLNtf. (b) The classes c_1 and c_2 are \located" in the training data by calculating the tightest ¯ts. Note that overlapping of two tightest ¯ts of di®erent categories is allowed (notice for example the overlapping of a _ b and 2 _ 3 _ 4) as long as no training data are contradicted. (c) Learning concludes by calculating the maximal expansions of the tightest ¯ts of all training data fed so far. (d) Additional data 5, 6 are fed to the FLNtf. The tightest ¯t a _ b is deleted because it contradicts training datum 5. (e) Additional datum e is fed to the FLNtf. The new tightest ¯ts are calculated as usual. (f) Learning concludes by calculating the maximal expansions of the tightest ¯ts on the training data fed so far.