1

Granular Enhancement of Fuzzy-ART/SOM Neural Classifiers Based on Lattice Theory

Vassilis G. Kaburlasos

Technological Educational Institution of Kavala, Dept. of Industrial Informatics
65404 Kavala, Greece, EU
vgkabs@teikav.edu.gr

Summary. Fuzzy adaptive resonance theory (fuzzy-ART) and self-organizing map (SOM) are two popular neural paradigms, which compute lattice-ordered granules. Hence, lattice theory emerges as a basis for unified analysis and design. We present both an enhancement of fuzzy-ART, namely fuzzy lattice reasoning (FLR), and an enhancement of SOM, namely granular SOM (grSOM). FLR as well as grSOM can rigorously deal with (fuzzy) numbers as well as with intervals. We introduce inspiring novel interpretations. In particular, the FLR is interpreted as a reasoning scheme, whereas the grSOM is interpreted as an energy function minimizer. Moreover, we can introduce tunable nonlinearities. The interest here is in classification applications. We cite evidence that the proposed techniques can clearly improve performance.

1.1 Introduction

Two novel approaches to neural computing were proposed lately by different authors based on lattice theory [22, 23, 35]. More specifically, one approach is based on the "order" definition for a lattice [22, 23], whereas the other one is based on the "algebra" definition for a lattice [35] as explained later. This chapter builds on the former definition; furthermore, it proposes granular enhancements of two popular neural paradigms, namely adaptive resonance theory (ART) and self-organizing map (SOM).

Stephen Grossberg, the founder of ART, points out occassionally an inherent affinity of the biologically-motivated clustering mechanisms of ART with Kohonen's SOM [26]. This work shows yet another aspect of the aforementioned affinity since both ART and SOM can be studied analytically (and also be further improved) based on mathematical lattice theory. Here we consider the fuzzy enhancement of ART known as fuzzy-ART [5].

The operation of both fuzzy-ART and SOM is based on the computation of clusters in R^N. In particular, a cluster for fuzzy-ART corresponds to a hyperbox, whereas a cluster for SOM corresponds to a Voronoi region in a metric space. Another term for cluster is *granule*. Lately, there is a growing interest in *granular computing* as explained next.

Granular computing is a category of theories, methodologies, techniques and tools that make use of *information granules* in the process of problem solving. Where, an (information) granule can be conceived as a collection of entities grouped together by similarity, functional adjacency, indistinguishability, coherency, etc. The basic notions and principles of granular computing have appeared under different names in many related fields such as information hiding in programming, granularity in artificial intelligence, divide and conquer in theoretical computer science, interval computing, cluster analysis, fuzzy and rough set theories, and many other. Granular computing is an emerging computational paradigm [27, 30].

Granules in R^N are partially-ordered, in particular they are lattice-ordered. Hence, lattice theory emerges as a basis for analysis and design in granular computing. This work shows granular enhancements of the popular fuzzy-ART and SOM classifiers. Moreover, novel interpretations are introduced. In particular, fuzzy-ART is interpreted as an interactive reasoning scheme, whereas SOM is interpreted as an energy minimizer. Note also that the techniques presented here could be useful elsewhere in granular computing.

This chapter is organized as follows. Section 1.2 summarizes, in context, the learning mechanisms of fuzzy-ART and SOM. Section 1.3 covers the mathematics required for describing the enhancements proposed later. Section 1.4 describes enhancements of both fuzzy-ART and SOM. Section 1.5 summarizes, in perspective, the contribution of this work.

1.2 Fuzzy-ART and SOM

This section summarizes the operation of both *fuzzy Adaptive Resonance Theory (fuzzy-ART)* and *Self-Organizing Map (SOM)* for unsupervised learning, i.e. clustering. It also presents interesting extensions by different authors.

1.2.1 Fuzzy-ART operation

The original fuzzy-ART neural network regards a two-layer architecture [5]. Layer F1 of fuzzy-ART fans out an input vector to the fully-interconnected, competitive neurons in layer F2. A layer F2 neuron filters an input vector $\mathbf{x}$ by computing vector $\mathbf{x} \wedge \mathbf{w}$, where $\mathbf{w}$ is the *code* (vector) stored on interlayer links. More specifically, an entry of vector $\mathbf{x} \wedge \mathbf{w}$ equals the minimum of the corresponding (positive real number) entries of vectors $\mathbf{x}$ and $\mathbf{w}$. Algorithm fuzzy-ART for training (learning) by clustering is briefly described next.

Algorithm fuzzy-ART for training

ART-1: Do while there are more inputs.
Apply the *complement coding* technique in order to represent input $[x_{i,1}, ..., x_{i,N}] \in [0, 1]^N$ by $\mathbf{x}_i = [x_{i,1}, ..., x_{i,N}, 1 - x_{i,1}, ..., 1 - x_{i,N}] \in \mathsf{R}^{2N}$, $i = 1, ..., n$. Then, present $\mathbf{x}_i$ to the (initially) "set" neurons in layer F2.

ART-2: Each layer F2 neuron with code $\mathbf{w}_j \in \mathsf{R}^{2N}$ computes its *choice* (*Weber*) function $T_j = |\mathbf{x}_i \wedge \mathbf{w}_j|/(\alpha + |\mathbf{w}_j|)$.

ART-3: If there are no "set" neurons in layer F2 then memorize input $\mathbf{x}_i$.
Else, competition among the "set" neurons in layer F2: Winner is neuron J such that $T_J \doteq arg\max_j T_j$.

ART-4: *Similarity Test*: $(|\mathbf{x}_i \wedge \mathbf{w}_J|/|\mathbf{x}_i|) \geq \rho$, where $|\mathbf{x}_i \wedge \mathbf{w}_J|/|\mathbf{x}_i|$ is the *match* function and $\rho \in (0, 1]$ is the user-defined *vigilance parameter*.

ART-5: If the *Similarity Test* is not satisfied then "reset" the winner neuron; goto step ART-3 to search for another winner.
Else, replace the winner neuron code $\mathbf{w}_J$ by $\mathbf{x}_i \wedge \mathbf{w}_J$; goto step ART-1.

We remark that $|\mathbf{x}|$ above equals, by definition, the sum of vector $\mathbf{x}$ (positive) entries. Parameter "α" in the choice (Weber) function T_j is a very small positive number whose role is to break ties in case of multiple winners [16].

As soon as training (learning) completes, each neuron defines a cluster by a hyperbox. It follows algorithm fuzzy-ART for testing (generalization).

Algorithm fuzzy-ART for testing

art-1: Feed an input vector $\mathbf{x}_0 = [x_{0,1}, ..., x_{0,N}, 1 - x_{0,1}, ..., 1 - x_{0,N}] \in \mathsf{R}^{2N}$.

art-2: A layer F2 neuron with code $\mathbf{w}_j \in \mathsf{R}^{2N}$ computes the choice (Weber) function $|\mathbf{x}_0 \wedge \mathbf{w}_j|/(\alpha + |\mathbf{w}_j|)$.

art-3: Competition among the neurons in layer F2: Winner is neuron J such that $T_J \doteq arg\max_j T_j$. Assign input $\mathbf{x}_0$ to the cluster represented by neuron J.

1.2.2 SOM operation

Kohonen's self-organizing map (SOM) architecture [26] includes a two dimensional $L \times L$ *grid* (or, *map*) of *neurons* (or, *cells*). Each cell $C_{i,j}$ stores a vector $\mathbf{m}_{i,j} = [m_{i,j,1}, ..., m_{i,j,N}]^T \in \mathsf{R}^N$, $i = 1, ..., L$, $j = 1, ..., L$. Vectors $\mathbf{m}_{i,j}$ are called *code vectors* and they are initialized randomly. A version of algorithm SOM for training (learning) by clustering is briefly described next.

Algorithm SOM for training

SOM-1: Memorize the first input datum $\mathbf{x}_1 \in \mathsf{R}^N$ by committing, randomly, a neuron on the $L \times L$ grid.
Repeat the following steps a user-defined number N_{epochs} of epochs, $t = 1, ..., N_{epochs}$.

SOM-2: For each training datum $\mathbf{x}_k \in \mathsf{R}^N$, $k = 1, ..., n$ carry out the following computations.

SOM-3: Calculate the Euclidean distance $d(\mathbf{m}_{i,j}, \mathbf{x}_k)$, $i, j \in \{1, ..., L\}$.

SOM-4: Competition among the neurons on the $L \times L$ grid: Winner is neuron $(I, J) \doteq arg \min_{i,j \in \{1,...,L\}} d_1(\mathbf{m}_{i,j}, \mathbf{x}_k)$.

SOM-5: *Assimilation Condition*: Vector $\mathbf{m}_{i,j}$ is in the neighborhood of vector $\mathbf{m}_{I,J}$ on the $L \times L$ grid.

SOM-6: If the *Assimilation Condition* is satisfied then compute a new value $\mathbf{m}'_{i,j}$:

$$\mathbf{m}'_{i,j} = \mathbf{m}_{i,j} + a(t)(\mathbf{x}_k - \mathbf{m}_{i,j}) = [1 - a(t)]\mathbf{m}_{i,j} + a(t)\mathbf{x}_k, \qquad (1.1)$$

where $a(t) \in (0, 1)$ is a decreasing function in time (t).

As soon as training (learning) completes, each cell $C_{i,j}$ defines a cluster by a code vector $\mathbf{m}_{i,j}$. It follows algorithm SOM for testing (generalization).

Algorithm SOM for testing

som-1: Present an input $\mathbf{x}_0 \in \mathsf{R}^N$, $k = 1, ..., n$ to the neurons of the $L \times L$ grid.

som-2: Calculate the Euclidean distance $d(\mathbf{m}_{i,j}, \mathbf{x}_0)$, $i, j \in \{1, ..., L\}$.

som-3: Competition among the neurons on the $L \times L$ grid: Winner is neuron $(I, J) \doteq arg \min_{i,j \in \{1,...,L\}} d_1(\mathbf{m}_{i,j}, \mathbf{x}_k)$. Assign input $\mathbf{x}_0$ to the cluster represented by neuron J.

Note that the set of clusters computed during training by both fuzzy-ART [10, 16] and SOM depends on the order of data presentation.

1.2.3 Extensions by different authors in context

Both fuzzy-ART and SOM compute information granules. In particular, fuzzy-ART computes fuzzy-sets with hyperbox cores, whereas SOM partitions its data domain in Voronoi-regions.

Both fuzzy-ART and SOM for clustering have been extended by a number of authors. More specifically, on the one hand, fuzzy-ARTMAP has been proposed for supervised learning [6]. Improvements of fuzzy-ARTMAP were proposed in various learning applications [7, 8]. Furthermore, fuzzy-ART(MAP) has inspired various min-max neural networks [2, 18]. An interesting probabilistic analysis of fuzzy-ARTMAP was proposed lately using martingales [1]. An extension of the fuzzy-ART(MAP) algorithm to a mathematical lattice data domain is the FLR algorithm [19]. Note that the FLR algorithm can be implemented as a neural network towards fuzzy lattice neurocomputing, or FLN for short [18, 23, 32]. On the other hand, SOM has been popular in signal- and other information- processing applications [26]. Various SOM extensions have been proposed including nonEuclidean metrics [31], weighting factors [13], etc. A different SOM extension is the generative topographic mapping (GTM) based on a constrained mixture of Gaussians [4]. However, conventional SOM as well as its extensions cannot cope with ambiguity. In response, SOM-based fuzzy c-means algorithms have been proposed [24]. Note that an early employment of SOM in fuzzy inference systems (FISs) appeared [37]. Lately, SOM extensions were presented for FIS analysis and design based on positive FINs [21, 29], the latter (FINs) are presented below.

1.3 Mathematical Background

This section covers lattice theory mathematics necessary for introducing novel enhancements later.

1.3.1 Crisp lattices

We present two equivalent definitions for a mathematical *lattice* [3], namely *order-based*- and *algebra-based*- definition, respectively. The former is based on the notion *partially ordered set*, or *poset* for short, defined in the Appendix.

By "a covers b" in a poset $(P, \leq)$ it is meant that $b < a$ but $b < x < a$ for no $x \in P$. Let $(P, \leq)$ be a poset with least element O. Every $x \in P$ which covers O, if such x exists, is called *atom*.

Let $(P, \leq)$ be a poset and $X \subseteq P$. An *upper bound* of X is a $a \in P$ with $x \leq a$, $\forall x \in X$. The *least upper bound* (l.u.b.), if it exists, is the unique upper bound contained in every upper bound. The l.u.b. is also called *supremum* or *lattice join* of X and denoted by $\texttt{sup}X$ or $\vee X$. The notions *lower bound* of X and *greatest lower bound* (g.l.b.) of X are defined dually. The g.l.b. is also called *infimum* or *lattice meet* of X and denoted by $\texttt{inf}X$ or $\wedge X$.

The *order-based* definition for a lattice follows.

Definition 1.1 *A* lattice *is a poset* $(\mathsf{L}, \leq)$ *any two of whose elements have both a greatest lower bound (g.l.b.), denoted by* $x \wedge y$*, and a least upper bound (l.u.b.), denoted by* $x \vee y$*. A lattice* $(\mathsf{L}, \leq)$*, or equivalently* crisp lattice*, is called* complete *when each of its subsets* X *has a l.u.b. and a g.l.b. in* L*.*

Setting $X = \mathsf{L}$ in definition 1.1 it follows that a nonvoid complete lattice contains both a *least* element and a *greatest* element denoted, respectively, by O and I. By definition, an *atomic lattice* $(\mathsf{L}, \leq)$ is a complete lattice in which every element is a joint of atoms.

The *algebra-based* definition for a lattice follows based on the notion *algebra* defined in the Appendix to this chapter.

Definition 1.2 *An algebra with two binary operations which satisfy L1-L4 is a lattice, and conversely.*

$(L1)$	$x \wedge x = x$	$x \vee x = x$	(Idempotent)
$(L2)$	$x \wedge y = y \wedge x$	$x \vee y = y \vee x$	(Commutative)
$(L3)$	$x \wedge (y \wedge z) = (x \wedge y) \wedge z$		(Associative)
	$x \vee (y \vee z) = (x \vee y) \vee z$		
$(L4)$	$x \wedge (x \vee y) = x$	$x \vee (x \wedge y) = x$	(Absorption)

We remark that definition 1.2 is popular in applications of mathematical morphology [12, 34, 36]. This work employs mainly definition 1.1.

Both definitions 1.1 and 1.2 regard a *crisp* lattice, where the binary relation $x \leq y$ is either true or false. In particular, if $x \leq y$ (or, $y \leq x$) then x and y are called *comparable*; otherwise, x and y are called *incomparable* or *parallel*, symbolically $x||y$. A simple crisp lattice example is shown next.

Example 1.1 *Let $\mathcal{P}(A) = \{\{\}, \{a\}, \{b\}, \{c\}, \{a,b\}, \{b,c\}, \{a,c\}, \{a,b,c\}\}$ be the power set of set $A = \{a,b,c\}$. It turns out that $(\mathcal{P}(\mathcal{A}), \subseteq)$ is a complete lattice, ordered by set-inclusion, with* least *and* greatest *elements $O = \{\}$ and $I = \{a,b,c\}$, respectively. Fig. 1.1 shows a* Hasse *diagram of lattice $(\mathcal{P}(A), \subseteq)$ such that a line segment connects two sets X (below) and Y (above) if and only if Y covers X. A* Hasse- *or, equivalently,* line- *diagram can be drawn only for a finite lattice.*

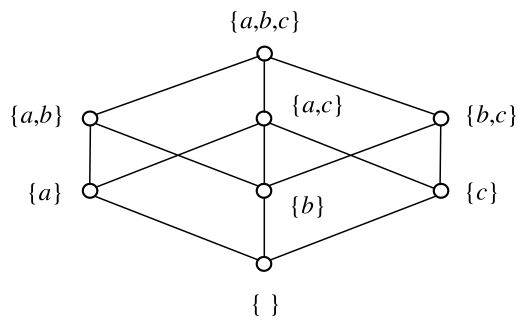

Fig. 1.1. Hasse diagram of the partially-ordered, complete lattice $(\mathcal{P}(A), \subseteq)$, where "$\mathcal{P}(A)$" is the powerset of set $A = \{a,b,c\}$ and "$\subseteq$" is the set-inclusion relation.

The inverse $\geq$ of an order relation $\leq$ is itself an order relation. More specifically, the order $\geq$ is called *dual* (order) of $\leq$, symbolically also $\leq^{\partial}$, or $\leq^{-1}$. Furthermore, the Cartesian product $(\mathsf{L}, \leq) = (\mathsf{L}_1, \leq_1) \times ... \times (\mathsf{L}_N, \leq_N)$ of N *constituent* lattices $(\mathsf{L}_1, \leq_1), ..., (\mathsf{L}_N, \leq_N)$ is a lattice [3]. In particular, if both $(a_1, ..., a_N)$ and $(b_1, ..., b_N)$ are in L then $(a_1, ..., a_N) \leq (b_1, ..., b_N)$ if and only if $a_i \leq b_i$, $i = 1, ..., N$.

Of particular interest here is lattice $(\tau(\mathsf{L}), \leq)$, where $\tau(\mathsf{L})$ denotes the set of intervals in L (including also the empty set) partially-ordered by set-inclusion — For a definition of an interval see in the Appendix. One way of dealing with lattice $(\tau(\mathsf{L}), \leq)$ is based on the product lattice $(\mathsf{L}^{\partial} \times \mathsf{L}, \leq^{\partial} \times \leq)$ [23].

1.3.2 Fuzzy lattices

The binary relation "$\leq$" in a crisp lattice can be fuzzified resulting in a *fuzzy lattice* as explained next — Note that a *fuzzy set* is denoted here by a pair (U, μ), where U is the *universe of discourse* and μ is a function $\mu : U \to [0,1]$, namely *membership* function.

Definition 1.3 *A* fuzzy lattice *is a triple $(\mathsf{L}, \leq, \mu)$, where $(\mathsf{L}, \leq)$ is a crisp lattice and $(\mathsf{L} \times \mathsf{L}, \mu)$ is a fuzzy set such that $\mu(x,y) = 1$ if and only if $x \leq y$.*

Function μ in definition 1.3 is a *weak (fuzzy) partial order* relation in the sense that both $\mu(x,y) = 1$ and $\mu(y,z) = 1$ imply $\mu(x,z) = 1$, whereas if either $\mu(x,y) \neq 1$ or $\mu(y,z) \neq 1$ then $\mu(x,z)$ could be any number in $[0,1]$. Fuzzification of a lattice can be pursued using either a *generalized zeta function* [25] or an *inclusion measure* function. The latter is defined next [19].

Definition 1.4 *Let $(\mathsf{L}, \leq)$ be a complete lattice with least element O. An inclusion measure is a map $\sigma : \mathsf{L} \times \mathsf{L} \to [0,1]$, which satisfies the following conditions.*
$(IM0)$ $\sigma(x, O) = 0, x \neq O$
$(IM1)$ $\sigma(x, x) = 1, \forall x \in \mathsf{L}$
$(IM2)$ $x \wedge y < x \Rightarrow \sigma(x, y) < 1$
$(IM3)$ $u \leq w \Rightarrow \sigma(x, u) \leq \sigma(x, w)$ (Consistency Property)

For noncomplete lattices condition $(IM0)$ drops.

We remark that $\sigma(x,y)$ may be interpreted as a (fuzzy) degree of inclusion of x in y. Therefore, notations $\sigma(x,y)$ and $\sigma(x \leq y)$ are used interchangably. Alternative inclusion measure function definitions have been proposed by different authors [9]. If $\sigma : \mathsf{L} \times \mathsf{L} \to [0,1]$ is an inclusion measure, in the sense of definition 1.4, then $(\mathsf{L}, \leq, \sigma)$ is a fuzzy lattice [18, 19].

1.3.3 Useful functions in a lattice

An inclusion measure can be defined in a crisp lattice $(\mathsf{L}, \leq)$ based on a *positive valuation* function (the latter is defined in the Appendix) as shown next.

Theorem 1.1 *If $v : \mathsf{L} \to \mathsf{R}$ is a positive valuation function in a crisp lattice $(\mathsf{L}, \leq)$ then both functions (a) $k(x,u) = v(u)/v(x \vee u)$, and (b) $s(x,u) = v(x \wedge u)/v(x)$ are inclusion measures.*

We point out that a positive valuation in a crisp lattice $(\mathsf{L}, \leq)$ also defines a *metric* function $d : \mathsf{L} \times \mathsf{L} \to \mathsf{R}_0^+$ given by $d(x,y) = v(x \vee y) - v(x \wedge y)$ — For a definition of a metric see in the Appendix to this chapter.

Given (i) a product lattice $(\mathsf{L}, \leq) = (\mathsf{L}_1, \leq_1) \times ... \times (\mathsf{L}_N, \leq_N)$, and (ii) both a positive valuation $v : \mathsf{L}_i \to \mathsf{R}$ and an *isomorphic* function $\theta_i : \mathsf{L}_i^\partial \to \mathsf{L}_i$ in a constituent lattice $(\mathsf{L}_i, \leq_i)$, $i = 1, ..., N$ — for a definition of an isomorphic function see in the Appendix to this chapter — then: (1) A positive valuation $v : \mathsf{L} \to \mathsf{R}$ is given by $v(x_1, ..., x_N) = v_1(x_1) + ... + v_N(x_N)$, (2) an isomorphic function $\theta : \mathsf{L}^\partial \to \mathsf{L}$ is given by $\theta(x_1, ..., x_N) = (\theta_1(x_1), ..., \theta_N(x_N))$, and (3) countably infinite *Minkowski metrics* d_p are given in L by

$$d_p(\mathbf{x}, \mathbf{y}) = [d_1^p(x_1, y_1) + ... + d_N^p(x_N, y_N)]^{1/p}, \tag{1.2}$$

where $p = 1, 2, ...$ and $d_i(x_i, y_i) = v_i(x_i \vee y_i) - v_i(x_i \wedge y_i)$, $x_i, y_i \in \mathsf{L}_i$, $i = 1, ..., N$. In the following, interest focuses on lattices stemming from the set R of real numbers.

1.3.4 Lattices stemming from R

Three different lattice examples are shown in Examples 1.2, 1.3, and 1.4 next including geometric interpretations on the plane.

Example 1.2 *Consider the set R of real numbers represented by a line (Fig. 1.2). It turns out that $(\mathsf{R}, \leq)$ is a noncomplete lattice including only comparable elements. Hence, lattice $(\mathsf{R}, \leq)$ is called* totally-ordered *or, equivalently,* chain. *Of particular interest is the complete sublattice $(\mathsf{I} = [0, 1], \leq)$.*

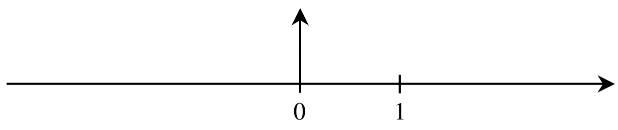

Fig. 1.2. The totally-ordered, noncomplete lattice $(\mathsf{R}, \leq)$ of real numbers. Note that lattice $(\mathsf{I}, \leq)$, where $\mathsf{I} = [0, 1]$, is a complete one.

Example 1.3 *Lattices of interest are both $(\tau(\mathsf{R}), \leq)$ and $(\tau(\mathsf{I}), \leq)$, where $\tau(\mathsf{R})$ and $\tau(\mathsf{I})$ denote the set of (closed) intervals in R and I, respectively. Consider the set of* hyperrectangles *or, equivalently,* hyperboxes, *in the partially-ordered lattice R^N. It turns out that a hyperbox is a (lattice) interval in R^N. Moreover, $(\tau(\mathsf{R}^N), \leq)$ denotes the noncomplete lattice of hyperboxes in R^N. Note that $\tau(\mathsf{R}^N) = [\tau(\mathsf{R})]^N$. Of particular interest is complete lattice $(\mathsf{I}^N, \leq)$, namely* unit-hypercube*. The corresponding complete lattice of hyperboxes is denoted by $(\tau(\mathsf{I}^N), \leq) \equiv ([\tau(\mathsf{I})]^N, \leq)$. Fig. 1.3 shows (hyper)boxes in lattice $(\tau(\mathsf{R}^N), \leq)$ for $N = 2$ (i.e. the plane). The unit-square is also shown.*

The *diagonal* of a hyperbox in R^N is defined as follows.

Definition 1.5 *The* diagonal *of a hyperbox $[a, b]$ in R^N, where $a, b \in \mathsf{R}^N$ with $a \leq b$, is defined as a nonnegative real function* $\text{diag}_p : \tau(\mathsf{R}^N) \rightarrow \mathsf{R}_0^+$ *given by $diag_p([a, b]) = d_p(a, b)$, $p = 1, 2, ...$*

1.3.5 Lattices of generalized intervals

Definition 1.6 *(a) A* positive generalized interval *of height h is a map $\mu^h_{a,b}$: $\mathsf{R} \rightarrow \{0, h\}$ given by* $\mu^h_{a,b}(x) = \begin{cases} h, \; a \leq x \leq b \\ 0, \; otherwise \end{cases}$ *, where $h \in (0, 1]$. (b) A* negative generalized interval *of height h is a map $\mu^h_{a,b} : \mathsf{R} \rightarrow \{0, -h\}$ given by* $\mu^h_{a,b}(x) = \begin{cases} -h, \; a \geq x \geq b \\ 0, \; otherwise \end{cases}$ *, where $a > b$ and $h \in (0, 1]$.*

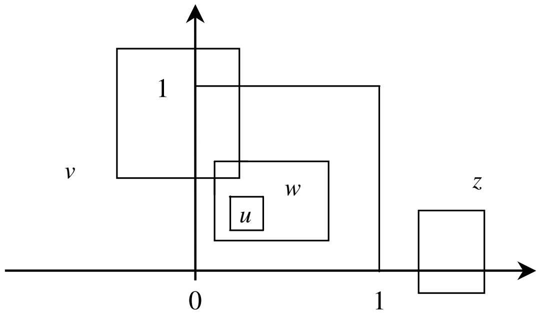

Fig. 1.3. Partially-ordered hyperboxes in the noncomplete lattice $(\mathsf{R}^N, \leq)$ are shown for $N = 2$ (i.e. the plane). The complete lattice unit-square is also shown. Box u is included in box w, i.e. $u \leq w$; all the other boxes are *incomparable*, e.g. $w||z$, etc.

Note that a generalized interval is a "box" function, either positive or negative. In the interest of simplicity a generalized interval will be denoted as $[a, b]^h$, where $a \leq b$ $(a > b)$ for a positive (negative) generalized interval.

The set of positive (negative) generalized intervals of height h is denoted by $\mathsf{M}_+^h(\mathsf{M}_-^h)$. The set of generalized intervals of height h is denoted by M^h, i.e. $\mathsf{M}^h = \mathsf{M}_-^h \cup \mathsf{M}_+^h$. It turns out that the set M^h of generalized intervals is *partially ordered*; more specifically, M^h is a *mathematical lattice* [17, 18] with *lattice meet* and *lattice join* given, respectively, by $[a, b]^h \wedge [c, d]^h = [a \vee c, b \wedge d]^h$ and $[a, b]^h \vee [c, d]^h = [a \wedge c, b \vee d]^h$. Moreover, the corresponding lattice order relation $[a, b]^h \leq [c, d]^h$ in M^h is equivalent to "$c \leq a$".AND."$b \leq d$".

Example 1.4 *Fig. 1.4 shows elements of lattice M^h. In particular, Fig. 1.4 shows all combinations for generalized intervals of height h as detailed in [17, 18, 20].*

In the totally-ordered lattice R of real numbers any strictly increasing function $f_h : \mathsf{R} \to \mathsf{R}$ is a *positive valuation*, whereas any strictly decreasing function $\theta_h : \mathsf{R} \to \mathsf{R}$ is an *isomorphic* function. Given both f_h and θ_h, a positive valuation in lattice $(\mathsf{M}^h, \leq)$ is given by $v([a, b]^h) = f_h(\theta_h(a)) + f_h(b)$. Therefore, a metric between two generalized intervals is given by

$$d_h([a, b]^h, [c, d]^h) = [f_h(\theta_h(a \wedge c)) - f_h(\theta_h(a \vee c))] + [f_h(b \vee d) - f_h(b \wedge d)] \quad (1.3)$$

Choosing $\theta_h(x) = -x$ and f_h such that $f_h(x) = -f_h(-x)$ it follows $d_h([a, b]^h, [c, d]^h) = [f_h(a \vee c) - f_h(a \wedge c)] + [f_h(b \vee d) - f_h(b \wedge d)]$.

The set-union of all M^hs is the set M of *generalized intervals*, i.e. $\mathsf{M} = \bigcup_{h \in (0,1]} \mathsf{M}^h$. Our interest is in generalized intervals $[a, b]^h$ with $h \in (0, 1]$ because the latter emerge from α-cuts of fuzzy numbers [18, 22]. It is interesting that

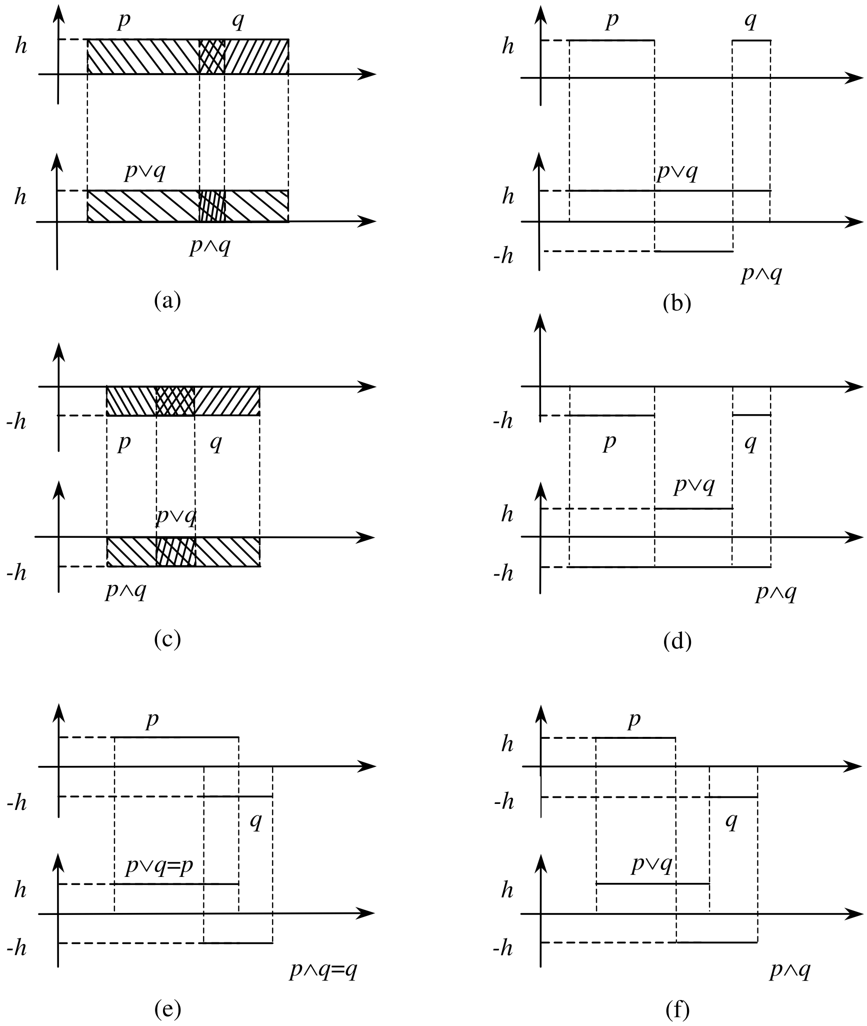

Fig. 1.4. Demonstrating the lattice- *join* $(p \vee q)$ and *meet* $(p \wedge q)$ for all different pairs (p, q) of generalized intervals of height h. Different fill-in patterns are used for partially overlapped generalized intervals. (a) "Intersecting" positive generalized intervals. (b) "Nonintersecting" positive generalized intervals. (c) "Intersecting" negative generalized intervals. (d) "Nonintersecting" negative generalized intervals. (e) "Intersecting" positive and negative generalized intervals. (f) "Nonintersecting" positive and negative generalized intervals.

different authors lately have considered the notion "α-fuzzy set" [33], the latter is identical to the notion "positive generalized interval" here.

The significance of a positive valuation function is demonstrated next.

Example 1.5 *Consider the positive generalized intervals* $[-1,0]^1$ *and* $[3,4]^1$*. Let* $f_1(x) = x^3$ *and* $f_2(x) = (1-e^{-x})/(1+e^{-x})$ *be the two strictly increasing functions shown in Fig. 1.5(a) and Fig. 1.5(b), respectively. Note that function* $f_1(x)$ *is steeply increasing, whereas function* $f_2(x)$ *is saturated. The computation of the (metric) distance* $d_1([-1,0]^1,[3,4]^1)$ *using* $f_1(x)$ *equals* $d_1([-1,0]^1,[3,4]^1) = [f_1(3)-f_1(-1)]+[f_1(4)-f_1(-0)] = 65+27 = 92$*. Whereas, the computation of the (metric) distance* $d_1([-1,0]^1,[3,4]^1)$ *using* $f_2(x)$ *equals* $d_1([-1,0]^1,[3,4]^1) = [f_2(3)-f_2(-1)]+[f_2(4)-f_2(-0)] = 1.3672+0.9640 = 2.3312$*. This example was meant to demonstrate that different positive valuation functions can drastically change the distance between two intervals. In practice, we often employ parametric positive valuations in order to introduce tunable nonlinearities by optimal parameter estimation.*

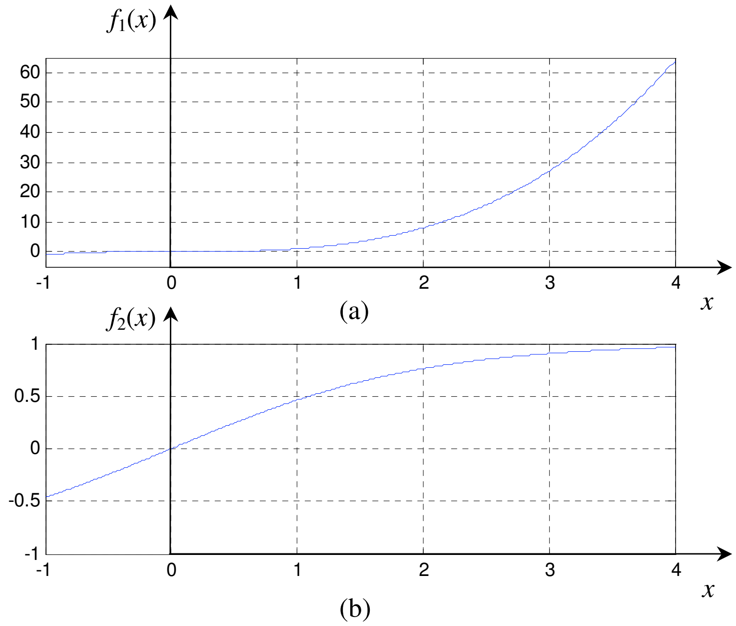

Fig. 1.5. Two positive valuation functions are shown on the domain $[-1,4]$ including (a) The steeply increasing *cubic* function $f_1(x) = x^3$, and (b) The saturated *logistic* function $f_2(x) = (1-e^{-x})/(1+e^{-x})$.

The space M^h of generalized intervals is a real *linear space* [18, 22] with

- *addition* defined as $[a,b]^h + [c,d]^h = [a+c, b+d]^h$.
- *multiplication* (by $k \in \mathsf{R}$) defined as $k[a,b]^h = [ka, kb]^h$.

A subset C of a linear space is called *cone* if for all $x \in C$ and a real number $\lambda > 0$ we have $\lambda x \in C$. It turns out that both M^h_+ and M^h_- are cones.

1.3.6 The lattice of Fuzzy Interval Numbers (FINs)

Consider the following definition.

Definition 1.7 *A* Fuzzy Interval Number*, or* FIN *for short, is a function* $F : (0,1] \to \mathsf{M}$ *such that (1)* $F(h) \in \mathsf{M}^h$*, (2) either* $F(h) \in \mathsf{M}^h_+$ *(positive FIN), or* $F(h) \in \mathsf{M}^h_-$ *(negative FIN) for all* $h \in (0,1]$*, and (3)* $h_1 \leq h_2 \Rightarrow \{x : F(h_1) \neq 0\} \supseteq \{x : F(h_2) \neq 0\}$.

A FIN F can be written as the set union of generalized intervals; in particular, $F = \bigcup_{h\in(0,1]} \{[a(h), b(h)]^h\}$, where both interval-ends $a(h)$ and $b(h)$ are functions of $h \in (0,1]$. The set of FINs is denoted by F. More specifically, the set of positive (negative) FINs is denoted by F_+ (F_-).

Example 1.6 *Fig. 1.6 shows a positive FIN. The only restriction is that a FIN's membership function needs to be "convex".*

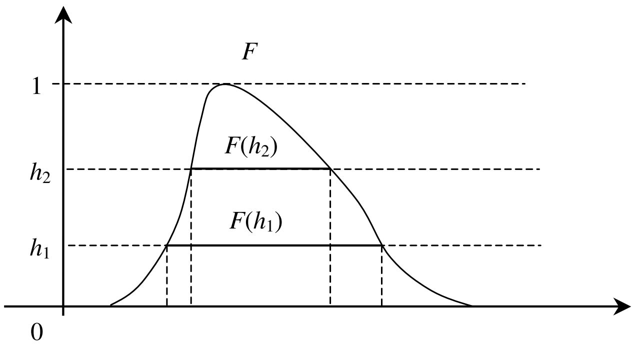

Fig. 1.6. A positive FIN $F = \bigcup_{h\in(0,1]} \{F(h)\}$ is the set-union of positive generalized intervals $F(h)$, $h \in (0,1]$.

We define an *interval-FIN* as $F = \bigcup_{h\in(0,1]} \{[a(h), b(h)]^h\}$, where both $a(h)$ and $b(h)$ are constant, i.e. $a(h) = a$ and $b(h) = b$. In particular, for $a = b$ an interval-FIN is called *trivial-FIN*. In the aforementioned sense F_+ includes both (fuzzy) numbers and intervals.

We remark that a FIN is a mathematical object, which can be interpreted either as a possibility distribution (i.e. a fuzzy number) or as a probability distribution, etc. [17, 18, 22]. An ordering relation has been introduced in F as follows: $F_1 \leq F_2 \Leftrightarrow F_1(h) \leq F_2(h), \forall h \in (0,1]$. It turns out that F is a mathematical lattice. The following proposition introduces a metric in F.

Proposition 1.1 *Let F_1 and F_2 be FINs in the lattice F of FINs. A metric function $d_K : \mathsf{F} \times \mathsf{F} \to \mathsf{R}_0^+$ is given by*

$$d_K(F_1, F_2) = \int_0^1 d_h(F_1(h), F_2(h))dh \tag{1.4}$$

Based on d_K, a metric $D : \mathsf{F}^N \times \mathsf{F}^N \to \mathsf{R}_0^+$ can be defined between two N-dimensional FINs $\mathbf{F}_1 = [F_{1,1}, ..., F_{1,N}]^T$ and $\mathbf{F}_2 = [F_{2,1}, ..., F_{2,N}]^T$ as follows.

$$D(\mathbf{F}_1, \mathbf{F}_2) = \sqrt{\sum_{i=1}^{N} d_K^2(F_{1,i}, F_{2,i})} \tag{1.5}$$

We remark that formula (1.5) may involve a vector $\mathbf{x} = [x_1, ..., x_N]^T \in \mathsf{R}^N$ under the assumption that a vector entry x_i (number) is represented by the trivial-FIN $x_i = \bigcup_{h\in(0,1]} \{[x_i, x_i]^h\}$, $i = 1, ..., N$.

Addition and multiplication are extended from M^h to F as follows.

- The *product* kF_1, where $k \in \mathsf{R}$ and $F_1 \in \mathsf{F}$, is defined as $F_p : F_p(h) = kF_1(h)$, $h \in (0, 1]$.
- The *sum* F_1+F_2, where $F_1, F_2 \in \mathsf{F}$ is defined as $F_s : F_s(h) = (F_1+F_2)(h) = F_1(h) + F_2(h)$, $h \in (0, 1]$.

We remark that, on the one hand, the product kF_1 is always a FIN. On the other hand, when both F_1 and F_2 are in cone F_+ (F_-) then the sum $F_1 + F_2$ is in cone F_+ (F_-). However, if $F_1 \in \mathsf{F}_+$ and $F_2 \in \mathsf{F}_-$ then $F_1 + F_2$ might not be a FIN. The interest of this work is in positive FINs.

1.3.7 Practical FIN representation

From a practical viewpoint a FIN F is represented in the computer memory by a $L \times 2$ matrix $\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ \vdots & \vdots \\ a_L & b_L \end{bmatrix}$ of real numbers, where L is a user-defined number of levels $h_1, h_2, ..., h_L$ such that $0 < h_1 \leq h_2 \leq ... \leq h_L = 1$; that is, FIN F equals $F = \bigcup_{i\in\{1,...,L\}} \{[a_i, b_i]^{h_i}\}$. In our experiments we usually use either $L = 16$ or $L = 32$ levels, spaced equally in the interval $[0, 1]$.

1.4 Enhancement of Both Fuzzy-ART and SOM

Based on the lattice-theoretic notions and tools presented previously, this section describes enhancements of both fuzzy-ART and SOM, namely *fuzzy lattice reasoning* (FLR) and *granular SOM* (grSOM), respectively, for supervised granular learning. We point out that the FLR is based on an *inclusion measure* function, whereas the grSOM is based on a *metric* function. Both aforementioned functions are used here in the lattice F_+ of positive FINs.

1.4.1 FLR: An enhancement of fuzzy-ART for classification

Algorithm FLR for training is presented next followed by algorithm FLR for testing. Both algorithms are applied on interval-FINs.

Algorithm FLR for training

FLR-0: A rule-base $RB = \{(u_1, C_1), ..., (u_L, C_L)\}$ is given, where $u_i \in \mathsf{F}_+^N$ is a hyperbox and $C_i \in \mathsf{C}$, $i = 1, ..., L$ is a class label — Note that C is a finite set of class labels.

FLR-1: Present the next input pair $(x_i, c_i) \in \mathsf{F}_+^N \times \mathsf{C}, i = 1, ..., n$ to the initially "set" RB.

FLR-2: If no more pairs are "set" in RB then store input pair (x_i, c_i) in RB; $L \leftarrow L + 1$; goto step FLR-1.
Else, compute the fuzzy degree of inclusion $k(x_i \leq u_l), l \in \{1, ..., L\}$ of input hyperbox x_i to all "set" hyperboxes $u_i, i = 1, ..., L$ in RB.

FLR-3: Competition among the "set" pairs in the RB: Winner is pair (u_J, C_J) such that $J \doteq arg \max\limits_{l \in \{1,...,L\}} k(x_i \leq u_l)$. In case of multiple winners, choose the one with the smallest diagonal size.

FLR-4: The *Assimilation Condition*: Both (1) $diag(x_i \vee u_J)$ is less than a maximum user-defined threshold size D_{crit}, and (2) $c_i = C_J$.

FLR-5: If the *Assimilation Condition* is not satisfied then "reset" the winner pair (u_J, C_J); goto step FLR-2.
Else, replace the winner hyperbox u_J by the join-interval $x_i \vee u_J$; goto step FLR-1.

Algorithm FLR for testing

flr-0: Consider a rule-base $RB = \{(u_1, C_1), ..., (u_L, C_L)\}$.

flr-1: Present a hyperbox $x_0 \in \mathsf{F}_+^N$ to the rule base RB.

flr-2: Compute the fuzzy degree of inclusion $k(x_0 \leq u_l), l \in \{1, ..., L\}$ of hyperbox x_0 in all hyperboxes $u_i, i = 1, ..., L$ in the RB.

flr-3: Competition among the hyperboxes in RB: Winner is pair (u_J, C_J) such that $J \doteq arg \max\limits_{l \in \{1,...,L\}} k(x_i \leq u_l)$.

flr-4: Hyperbox x_0 is classified to the class with label C_J.

By "hyperbox" above we mean "interval-FIN". We remark that the FLR has been described as a rule-based classifier [19], where a hyperbox h is assigned a class label thus corresponding to the following rule: If a point p is inside hyperbox h (let the latter by labeled by c) then p is in class c. For points outside all hyperboxes, as well as for points inside overlapping hyperboxes, inclusion measure k is used to assign a class. Note also that FLR has been implemented on a neural network architecture [18, 23].

There are inherent similarities as well as substantial differences between fuzzy-ART and FLR. In particular, both fuzzy-ART and FLR carry out learning rapidly in a single pass through the training data by computing hyperboxes in their data domain. Note that a computed hyperbox corresponds to the core of a fuzzy set and it can be interpreted as an information granule (cluster).

Advantages of FLR over fuzzy-ART include (1) *comprehensiveness*, (2) *flexibility*, and (3) *versatility* as summarized next [23]. (1) The FLR can handle intervals (granules), whereas fuzzy-ART deals solely with trivial intervals the latter are points in the unit-hypercube. (2) It is possible to optimize FLR's behavior by tuning an underlying positive valuation function v as well as an isomorphic function θ, whereas fuzzy-ART implicitly uses, quite restrictively, only $v(x) = x$ and $\theta(x) = 1 - x$. (3) The FLR can handle general lattice elements including points in the unit-hypercube, the latter is fuzzy-ART's sole application domain.

In addition, the FLR can deal with "missing" data as well as with "don't care" data in a constituent complete-lattice by replacing the aforementioned data by the *least* and the *greatest* element, respectively, in the corresponding constituent lattice [23].

Both of fuzzy-ART's *choice (Weber)* function and *match* function correspond to FLR's inclusion measure function "k". Moreover, fuzzy-ART's *complement coding* technique corresponds to a specific isomorphic function, namely $\theta(x) = 1 - x$. Apparently, choosing a different isomorphic function than $\theta(x) = 1 - x$ results in a different "coding" technique [18, 23].

Let f_i and θ_i be strictly-increasing and strictly-decreasing functions, respectively, in a constituent lattice R in R^N. A typical assumption for both fuzzy-ART and FLR is to select f_i and θ_i such that equation $v([a, b]) = 1 + diag_1([a, b])$ is satisfied [19]. On the one hand, two popular functions f_i and θ_i in the complete lattice unit-interval $[0, 1]$ are $f_i(x) = x$ and $\theta_i(x) = 1 - x$. On the other hand, two popular functions f_i and θ_i in the noncomplete lattice R are $f_i(x) = 1/(1 + e^{-\lambda(x-x_0)})$ and $\theta_i(x) = 2x_0 - x$.

Inclusion measure "k" above retains an *Occam razor* semantic interpretation as detailed in [19]. In particular, winner of the competition in steps FLR-3/flr-3 above is the hyperbox whose diagonal needs to be modified *the least* so as to "barely" include an input datum/hyperbox.

The FLR was interpreted lately as a reasoning scheme, which supports two different modes of reasoning, namely *Generalized Modus Ponens* and *Reasoning by Analogy* [19]. A novel interpretation is presented here as follows. Inclusion measure $k(p \leq q)$ is interpreted as the degree of truth of implication

"$p \Rightarrow q$" involving the truth values p and q, respectively, of two propositions. Note that various mechanisms have been proposed in the literature for calculating the degree of truth of an implication "$p \Rightarrow q$" given the truth values p and q [14]. The basic difference here is that the truth values p and q of the two propositions involved in implication "$p \Rightarrow q$" take on values in a general complete lattice [11] rather than taking on values solely in the unit-interval $[0, 1]$. However, the truth of implication "$p \Rightarrow q$" takes on values in the unit-interval $[0, 1]$. More specifically, the truth of implication "$p \Rightarrow q$" is calculated as $k(p \Rightarrow q) = v(q)/v(p \vee q)$. In conclusion, the FLR carries out interactively tunable inferences. A couple of FLR drawbacks are described next.

Fuzzy-ART's *proliferation problem*, that is the proliferation of hyperboxes/clusters, is inherited to FLR. However, FLR is equipped with tools such as an inclusion measure as well as a metric function to reduce "in principle" the number of hyperboxes.

Another drawback of FLR, also inherited from fuzzy-ART, is that the learned clusters (in particular their total number, size, and location) depend on the order of presenting the training data. A potential solution is to employ an ensemble of FLR classifiers in order to boost performance stably [18].

1.4.2 grSOM: An enhancement of SOM for classification

Algorithm FLR is applicable in the space F_+ of (positive) FINs. Algorithm grSOM for learning (training) is presented next followed by algorithm grSOM for generalization (testing).

Algorithm grSOM for training

GR-0: Define the size L of a $L \times L$ grid of neurons. Each neuron can store both a N-dimensional FIN $W_{i,j} \in \mathsf{F}_+^N$, $i, j \in 1, ..., L$ and a class label $C_{i,j} \in \mathsf{C}$, where C is a finite set. Initially all neurons are *uncommitted*.

GR-1: Memorize the first training data pair $(x_1, C_1) \in \mathsf{F}_+^N \times \mathsf{C}$ by committing, randomly, a *neuron* in the $L \times L$ grid.
Repeat the following steps a user-defined number N_{epochs} of epochs, $p = 1, ..., N_{epochs}$.

GR-2: For each training datum $(x_k, C_k) \in \mathsf{F}_+^N \times \mathsf{C}$, $k = 1, ..., n$ "reset" all $L \times L$ grid neurons. Then carry out the following computations.

GR-3: Calculate the Minkowski metric $d_1(x_k, W_{i,j})$ between x_k and *committed* neurons $W_{i,j}$ $i, j \in \{1, ..., L\}$.

GR-4: Competition among the "set" (and, *committed*) neurons in the $L \times L$ grid: Winner is neuron (I, J) whose weight $W_{I,J}$ is the nearest to x_k, i.e. $(I, J) \doteq arg \min_{i,j \in \{1,...,L\}} d_1(x_k, W_{i,j})$.

GR-5: *Assimilation Condition*: Both (1) Vector $W_{i,j}$ is in the neighborhood of vector $W_{I,J}$ on the $L \times L$ grid, and (2) $C_{I,J} = C_k$.

GR-6: If the *Assimilation Condition* is satisfied then compute a new value $W'_{i,j}$ as follows:

$$W'_{i,j} = \left[1 - \frac{h(k)}{1+d_K(W_{I,J},W_{i,j})}\right] W_{i,j} + \frac{h(k)}{1+d_K(W_{I,J},W_{i,j})} x_k.$$

Else, if the *Assimilation Condition* is not satisfied, "reset" the winner (I, J); goto GR-4.

GR-7: If all the $L \times L$ neurons are "reset" then commit an *uncommitted* neuron from the grid to memorize the current training datum (x_k, C_k).
If there are no more *uncommitted* neurons then increase L by one.

Algorithm grSOM for testing

gr-0: Present $x_0 \in \mathsf{F}_+^N$ to a trained grSOM.
gr-1: Calculate the Minkowski metric $d_1(x_0, W_{i,j})$ for *committed* neurons $W_{i,j}$, $i, j \in \{1, ..., L\}$.
gr-2: Competition among the *committed* neurons in the $L \times L$ grid: Winner is neuron (I, J) such that $(I, J) \doteq arg \min_{i,j \in \{1,...,L\}} d_1(x_0, W_{i,j})$.
gr-3: The class C_0 of x_0 equals $C_0 \doteq C_{I,J}$.

Function "$h(k)$", in the training phase above, reduces smoothly from 1 down to 0 with the epoch number k. The above algorithm is called *incremental-grSOM* [20]. It differs from another grSOM algorithm, namely *greedy-grSOM* [22], in that only the incremental-grSOM employs convex combinations of (positive) FINs. Both grSOM and SOM partition the data domain in Voronoi-regions, and each one of the aforementioned regions can also be interpreted as an information granule.

A fundamental improvement of grSOM over SOM is the sound capacity of grSOM to rigorously deal with nonnumeric data including both fuzzy numbers and intervals represented by FINs. However, the decision-making function of grSOM (as well as the corresponding function of SOM) does not admit a logical/linguistic interpretation. Rather, since the aforementioned function is an energy-type objective function, that is a *metric*, optimization is pursued during learning using *energy minimization* techniques.

1.5 Conclusion

This chapter was meant as a reference towards proliferating the employment of both fuzzy-ART and SOM in granular classification applications. Enhancements of fuzzy-ART as well as of SOM were presented, namely FLR and grSOM, respectively. FLR/grSOM is applicable in the lattice of fuzzy interval numbers, or FINs for short, including both (fuzzy) numbers and intervals. The FLR was interpreted as a reasoning scheme, whereas the grSOM was interpreted as an energy minimizer. The employment of mathematical lattice theory was instrumental for introducing useful tools.

Ample experimental evidence suggests that FLR and/or grSOM can comparatively improve classification performance [18, 19, 20, 21, 22, 23].

Future work will consider alternative granular inputs to modified FLR /grSOM classifiers including type-2 fuzzy set inputs, rough set inputs, etc. In addition, fuzzy logic reasoning applications [15, 28] will be pursued.

References

1. Andonie R, Sasu L (2006) Fuzzy ARTMAP with input relevances. IEEE Trans Neural Networks 17(4):929–941
2. Bargiela A, Pedrycz W, Hirota K (2004) Granular prototyping in fuzzy clustering. IEEE Trans Fuzzy Systems 12(5):697–709
3. Birkhoff G (1967) Lattice Theory. American Math Society, Colloquium Publications XXV
4. Bishop CM, Svensén M, Williams CKI (1998) GTM: The generative topographic mapping. Neural Computation 10(1):215–234
5. Carpenter GA, Grossberg S, Rosen DB (1991) Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Networks 4(6):759–771
6. Carpenter GA, Grossberg S, Markuzon N, Reynolds JH, Rosen DB (1992) Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Trans Neural Networks 3(5):698–713
7. Castro J, Georgiopoulos M, DeMara R, Gonzalez A (2005) Data-partitioning using the Hilbert space filling curves: Effect on the speed of convergence of Fuzzy ARTMAP for large database problems. Neural Networks 18(7):967–984
8. Castro J, Secretan J, Georgiopoulos M, DeMara RF, Anagnostopoulos G, Gonzalez AJ (2007) Pipelining of fuzzy-ARTMAP without match-tracking: Correctness, performance bound, and beowulf evaluation. Neural Networks 20(1):109–128
9. Cripps A, Nguyen N (2007) Fuzzy lattice reasoning (FLR) classification using similarity measures. This volume, chapter 13
10. Georgiopoulos M, Fernlund H, Bebis G, Heileman GL (1996) Order of search in fuzzy ART and fuzzy ARTMAP: Effect of the choice parameter. Neural Networks 9(9):1541–1559
11. Goguen JA (1967) L-fuzzy sets. J Math Analysis and Applications 18:145–174
12. Graña M, Sussner P, Ritter G (2003) Innovative applications of associative morphological memories for image processing and pattern recognition. Mathware & Soft Computing 10(2-3):155–168
13. Hammer B, Villmann T (2002) Generalized relevance learning vector quantization. Neural Networks 15(8-9):1059–1068
14. Hatzimichailidis A, Kaburlasos VG, Papadopoulos B (2006) An implication in fuzzy sets. In: Proc World Congress Computational Intelligence (WCCI) FUZZ-IEEE Program pp 203–208
15. Igel C, Temme KH (2004) The chaining syllogism in fuzzy logic. IEEE Trans Fuzzy Systems 12(6):849–853
16. Kaburlasos VG (1992) Adaptive Resonance Theory with Supervised Learning and Large Database Applications. PhD Thesis, Univ Nevada, Reno

17. Kaburlasos VG (2004) FINs: Lattice theoretic tools for improving prediction of sugar production from populations of measurements. IEEE Trans Systems, Man and Cybernetics - B, Cybernetics 34(2):1017–1030
18. Kaburlasos VG (2006) Towards a Unified Modeling and Knowledge-Representation Based on Lattice Theory — Computational Intelligence and Soft Computing Applications, ser Studies in Computational Intelligence 27. Springer, Heidelberg, Germany
19. Kaburlasos VG, Athanasiadis IN, Mitkas PA (2007) Fuzzy lattice reasoning (FLR) classifier and its application for ambient ozone estimation. Intl J Approximate Reasoning (in press)
20. Kaburlasos VG, Chatzis V, Tsiantos V, Theodorides M (2005) granular Self-Organizing Map (grSOM) neural network for industrial quality control. In: Astola JT, Tabus I, Barrera J (eds) Proc SPIE Math Methods in Pattern and Image Analysis 5916 pp 59160J:1–10
21. Kaburlasos VG, and Kehagias A (2007) Novel fuzzy inference system (FIS) analysis and design based on lattice theory. IEEE Trans Fuzzy Systems (in press)
22. Kaburlasos VG, Papadakis SE (2006) Granular self-organizing map (grSOM) for structure identification. Neural Networks 19(5):623–643
23. Kaburlasos VG, Petridis V (2000) Fuzzy lattice neurocomputing (FLN) models. Neural Networks 13(10):1145–1170
24. Karayiannis NB, Bezdek JC (1997) An integrated approach to fuzzy learning vector quantization and fuzzy c-means clustering. IEEE Trans Fuzzy Systems 5(4):622–628
25. Knuth KH (2007) Valuations on lattices: Fuzzification and its implications. This volume, chapter 15
26. Kohonen T (1995) Self-Organizing Maps, ser Information Sciences 30. Springer.
27. Lin TY (2005) Introduction to special issues on data mining and granular computing (editorial). Intl J Approximate Reasoning 40(1-2):1–2
28. Liu J, Ruan D, Xu Y, Song Z (2003) A resolution-like strategy based on a lattice-valued logic. IEEE Trans Fuzzy Systems 11(4):560–567
29. Papadakis SE, Kaburlasos VG (submitted) Efficient structure identification in TSK models based on a novel self-organizing map (SOM)
30. Pedrycz W (2005) Knowledge-Based Clustering — From Data to Information Granules. John Wiley & Sons, Hoboken, NJ
31. Peltonen J, Klami A, Kaski S (2004) Improved learning of Riemannian metrics for exploratory analysis. Neural Networks 17(8-9):1087–1100
32. Piedra-Fernández JA, Cantón-Garbín M, Guindos-Rojas F (2007) Application of fuzzy lattice neurocomputing (FLN) in ocean satellite images for pattern recognition. This volume, chapter 11
33. Popescu M, Gader P, Keller JM (2006) Fuzzy spatial pattern processing using linguistic hidden Markov models. IEEE Trans Fuzzy Systems 14(1):81–92
34. Ritter GX, Gader PD (2006) Fixed points of lattice transforms and lattice associative memories. In: Hawkes P (ed) Advances in Imaging and Electron Physics 144:165–242. Elsevier, Amsterdam, The Netherlands
35. Ritter GX, Urcid G (2003) Lattice algebra approach to single-neuron computation. IEEE Trans Neural Networks 14(2):282–295
36. Sussner P, Valle ME (2006) Gray-scale morphological associative memories. IEEE Trans Neural Networks 17(3):559–570
37. Vuorimaa P (1994) Fuzzy self-organizing map. Fuzzy Sets Syst 66(2):223–231

Chapter 1 Appendix

A *poset* is a pair $(P, \leq)$, where P is a set and $\leq$ is a binary *partial order relation* defined next.

Definition 1.8 *A* partial order *relation satisfies the following laws.*
$(PO1)$ $x \leq x$ (Reflexivity)
$(PO2)$ $x \leq y$ *and* $y \leq x \Rightarrow x = y$ (Antisymmetry)
$(PO3)$ $x \leq y$ *and* $y \leq z \Rightarrow x \leq z$ (Transitivity)

We remark that relation $<$ means both $\leq$ and $\neq$.

Definition 1.9 *An* algebra *is a pair (S, F), where S is a non-empty set, and F is a set of operations f_a, each mapping a power $S^{n(a)}$ of S into S for some non-negative finite integer $n(a)$.*

We remark that each operation f_a assigns to every $n(a)$-ple $(x_1, \ldots, x_{n(a)})$ of elements of S, an element $f_a(x_1, \ldots, x_{n(a)})$ in S, the result of performing the operation f_a on the sequence $x_1, \ldots, x_{n(a)}$. In particular, if $n(a) = 1$, the operation f_a is called *unary*; if $n(a) = 2$, it is called *binary*, etc.

Definition 1.10 *An* interval $[a, b]$*, with $a \leq b$ in a poset $(P, \leq)$, is defined as the set $[a, b] = \{x \in P : a \leq x \leq b\}$.*

Definition 1.11 *A* positive valuation *in a crisp lattice $(\mathsf{L}, \leq)$ is a real function $v : \mathsf{L} \to \mathsf{R}$, which satisfies both*
(PV1) $v(x) + v(y) = v(x \wedge y) + v(x \vee y)$*, and*
(PV2) $x < y \Rightarrow v(x) < v(y)$.

Definition 1.12 *A* metric *in a set A is a nonnegative real function $d : A \times A \to \mathsf{R}_0^+$, which satisfies*
$(D0)$ $d(x, y) = 0 \Rightarrow x = y$
$(D1)$ $d(x, x) = 0$
$(D2)$ $d(x, y) = d(y, x)$
$(D3)$ $d(x, z) \leq d(x, y) + d(y, z)$ (Triangle Inequality)

If only conditions $D1$, $D2$, and $D3$ are satisfied then function d is called *pseudo-metric*. We remark that a *metric space* is a pair (A, d) including both a set A and a metric $d : A \times A \to \mathsf{R}_0^+$.

Definition 1.13 *Let P and Q be posets. A map $\psi : P \to Q$ is called*
(i) Order-perserving *(or, alternatively,* monotone*), if $x \leq y$ in P implies $\psi(x) \leq \psi(y)$ in Q.*
(ii) Order-isomorphism *(or, simply,* isomorphism*), if both $x \leq y$ in $P \Leftrightarrow \psi(x) \leq \psi(y)$ in Q and "ψ is onto Q".*

We remark that when there is an isomorphism from P to Q, then P and Q are called *isomorphic*, symbolically $P \cong Q$; moreover, the corresponding function ψ is called *isomorphic* (function).