Lattice computing for Artificial Intelligence applications

Manuel Graña Romay

Grupo de Inteligencia Computacional

UPV/EHU

www.ehu.es/ccwintco

Contents

- Introductory ideas and history
- Review of early models
  - Fuzzy ART
  - Max-min classifiers
- Recent approaches
  - Associative Morphological Memories
  - Fuzzy Lattice Neurocomputing
  - Fuzzy Mathematical Morphology
- Conclusions and the future

Introduction

- Lattice computing assumes that the basic computing structure is a lattice.
- A lattice $(L, \vee, \wedge)$ is a Poset $(L, \leq)$ any two of whose elements have
  - a supremum, denoted by $x \vee y$
  - an infimum, denoted by $x \wedge y$

Introduction

- Poset

A *partially-ordered set*, briefly **poset** $(\mathcal{P}, \leq)$, is a set $\mathcal{P}$ in which a binary relation $\leq$ is defined that is a *partial ordering*, i.e., satisfies the following three properties for all $X, Y, Z \in \mathcal{P}$:

(P1). $X \leq X$ (reflexive)
(P2). $X \leq Y$ and $Y \leq X$ imply $X = Y$ (antisymmetric)
(P3). $X \leq Y$ and $Y \leq Z$ imply $X \leq Z$ (transitive)

Introduction

- Computational paradigm shift (Ritter)
  - Traditional Artificial Neural Networks are defined on the ring $(\mathbb{R}, +, \times)$

$$\tau_j(\mathbf{x}) = \sum_{i=1} x_i w_{ij} - \theta_j$$

  - Lattice ANN work on the semi-rings

$$(\mathbb{R}_{-\infty}, \vee, +) \text{ or } (\mathbb{R}_{\infty}, \wedge, +)$$

$$\tau_j(\mathbf{x}) = p_j \bigvee_{i=1}^{n} r_{ij}(x_i + w_{ij}) \qquad \tau_j(\mathbf{x}) = p_j \bigwedge_{i=1}^{n} r_{ij}(x_i + w_{ij})$$

Introduction

- Biological justification (Ritter)
  - Dendrites account for 50% of brain mass
  - Dendrite computation is more akin to AND, XOR, NOT logical operations

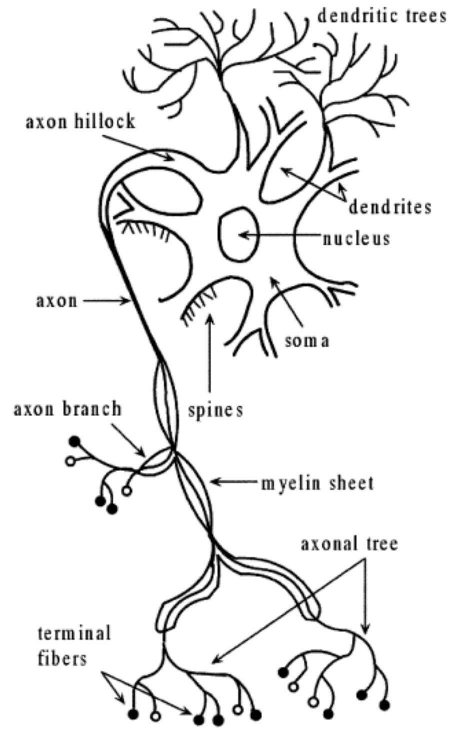


Fig. 1. Diagram of a neuron cell showing dendrites, dendritic trees, axon branches, and terminal branches. Excitatory and inhibitory inputs are indicated, respectively, by black small disks (•) and small circles (○).

Introduction

- Mathematical morphology for image processing is also a lattice paradigm shift from linear processing (Maragos)
  - Linear translation-invariant (LTI) operators are uniquely represented by linear convolution with the impulse response
  - Erosion (Dilation) translation invariant (ETI(DTI)) operators are uniquely represented by inf-(sup) convolution with the impulse response

Introduction

$$\psi \text{ is LTI} \Leftrightarrow \psi(F)(x) = (F * H)(x) = \sum_{y} F(y) H(x-y)$$

DTI $$(F \circledast H)(x) \triangleq \bigvee_{y \in \mathbb{E}} F(y) \star H(x-y)$$

ETI $$(F \circledast' H')(x) \triangleq \bigwedge_{y \in \mathbb{E}} F(y) \star' H'(x-y)$$

Introduction

- Kinds of processes in Artificial Intelligence
  - Filtering

$$\psi : R^N \rightarrow R^N$$

  - Dimension reduction

$$\psi : R^N \rightarrow R^d ; d << N$$

  - Classification (supervised, unsupervised)

$$\psi : R^N \rightarrow \Omega; \quad \Omega = \{\omega_1, \cdots, \omega_c\}$$

Introduction

- Kinds of lattice computing
  - Filtering: Mathematical Morphology
  - Dimension reduction: ? ? ? ? ? ? ?
  - Classification- recognition
    - Fuzzy systems
    - Artificial Neural Networks
  - Specific processes
    - Target Localization in images
    - Endmember induction in hyperspectral images

Introduction

- The learning problem
  - Gradient descent schemas need to compute derivatives of sup, inf functions.
  - Heuristic growing produces overfitting (category explosion) and there is no proof of convergence.
  - Random search algorithms are computationally expensive.

Some historical landmarks

- 1979
  - R. Cuninghame-Green: Minimax Algebra
- 1982
  - J. Serra: Image Analysis and Mathematical Morphology
- 1991
  - Carpenter, Grossberg: Fuzzy-ART
- 1992
  - Simpson: Min-max Neural Networks
  - Pedrycz: Relational System Learning
- 1995
  - Yang, Maragos: Min-max Classifiers
- 1998
  - Ritter, Sussner: Morphological Associative Memories
  - Gader: Shared-weight Morphological Neural Networks
- 2000
  - Kaburlassos, Petridis: Fuzzy Lattice Neurocomputing
- 2003
  - Ritter: Dendritic Computing
- 2005
  - Kaburlassos: Towards a unified modeling and knowledge representation based on Lattice Therory
  - Maragos: Lattice image processing: a unification of morphological and Fuzzy algebraic systems
- 2007
  - Kaburlassos, Ritter: Computational Intelligence based on Lattice Theory

Fuzzy ART

Carpenter, Grossberg

Starting point

- It is an extensión of binary input Adaptive Resonance Theory (ART) to continuous variables in [0,1]:
  - Logical AND, intersection --> inf operator
- Coding:
  - appending the complementary (1-x_i) to each input variable x_i.
- Category == Cluster

ART 1 (BINARY)	FUZZY ART (ANALOG)

CATEGORY CHOICE

$$T_j = \frac{|\mathbf{I} \cap \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|} \qquad\qquad T_j = \frac{|\mathbf{I} \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|}$$

MATCH CRITERION

$$\frac{|\mathbf{I} \cap \mathbf{w}|}{|\mathbf{I}|} \geq \rho \qquad\qquad \frac{|\mathbf{I} \wedge \mathbf{w}|}{|\mathbf{I}|} \geq \rho$$

FAST LEARNING

$$\mathbf{w}_j^{(\text{new})} = \mathbf{I} \cap \mathbf{w}_j^{(\text{old})} \qquad\qquad \mathbf{w}_j^{(\text{new})} = \mathbf{I} \wedge \mathbf{w}_j^{(\text{old})}$$

$\cap$ = logical AND intersection $\qquad$ $\wedge$ = fuzzy AND minimum

Fig. 2. Comparison of ART 1 and fuzzy ART.

Algorithm Elements

- Category selection based on T_j
  - It is a measure of inclusion of the input in the category

$$T_J = \max\{T_j : j = 1 \cdots N\}. \qquad (\boldsymbol{p} \wedge \boldsymbol{q})_i \equiv \min(p_i, q_i)$$

$$T_j(\boldsymbol{I}) = \frac{|\boldsymbol{I} \wedge \boldsymbol{w}_j|}{\alpha + |\boldsymbol{w}_j|}, \qquad |\boldsymbol{p}| \equiv \sum_{i=1}^{M} |p_i|$$

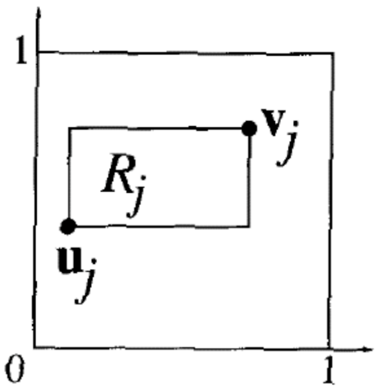
- Resonance: Vigilance parameter ρ
  - Decision about the creation of a new category
  - Measure of category compactness: inclusion of the weight $\boldsymbol{w}_J$ in the input $\boldsymbol{I}$

$$\frac{|\boldsymbol{I} \wedge \boldsymbol{w}_J|}{|\boldsymbol{I}|} \geq \rho;$$

Input accepted in the winning category

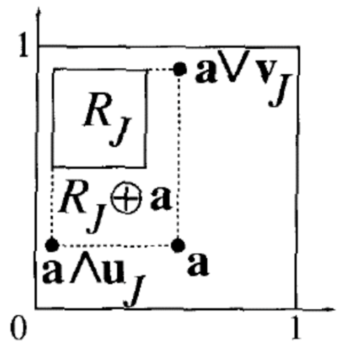
- Learning
  - Enlarging the category enclosing the new data

$$\boldsymbol{w}_J^{(\mathrm{new})} = \beta\left(\boldsymbol{I} \wedge \boldsymbol{w}_J^{(\mathrm{old})}\right) + (1-\beta)\boldsymbol{w}_J^{(\mathrm{old})}.$$



(a)

After presentation of **a** (β=1)- - ->



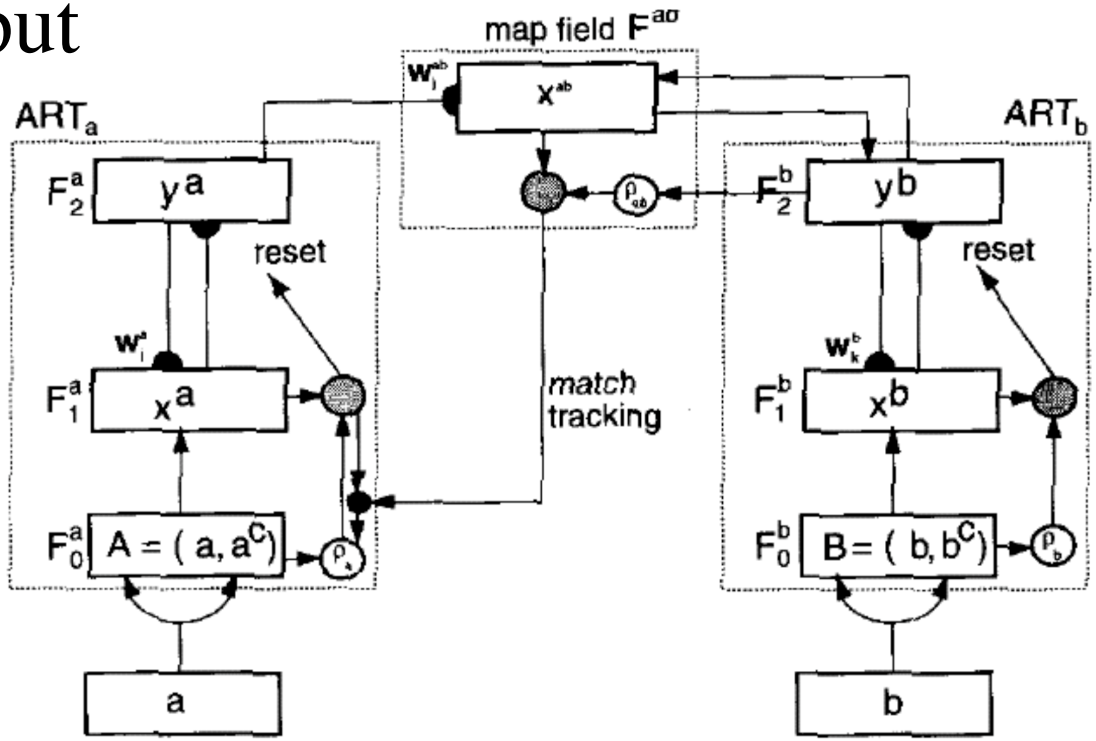
(b)

Fuzzy-ART properties

- Forms hyper-rectangular categories covering the data
- Hyper-rectangles grow monotonically in all dimensions during training
- The size of a category equals $|R_j| = M - |\mathbf{w}_j|$
- It is bounded by $|R_j| < M(1-\rho)$
- If $0 \leq \rho < 1$ the number of categories is bounded (but most times grows big!)

Supervised learning ARTMAP

- Encodes and categorizes both input and output



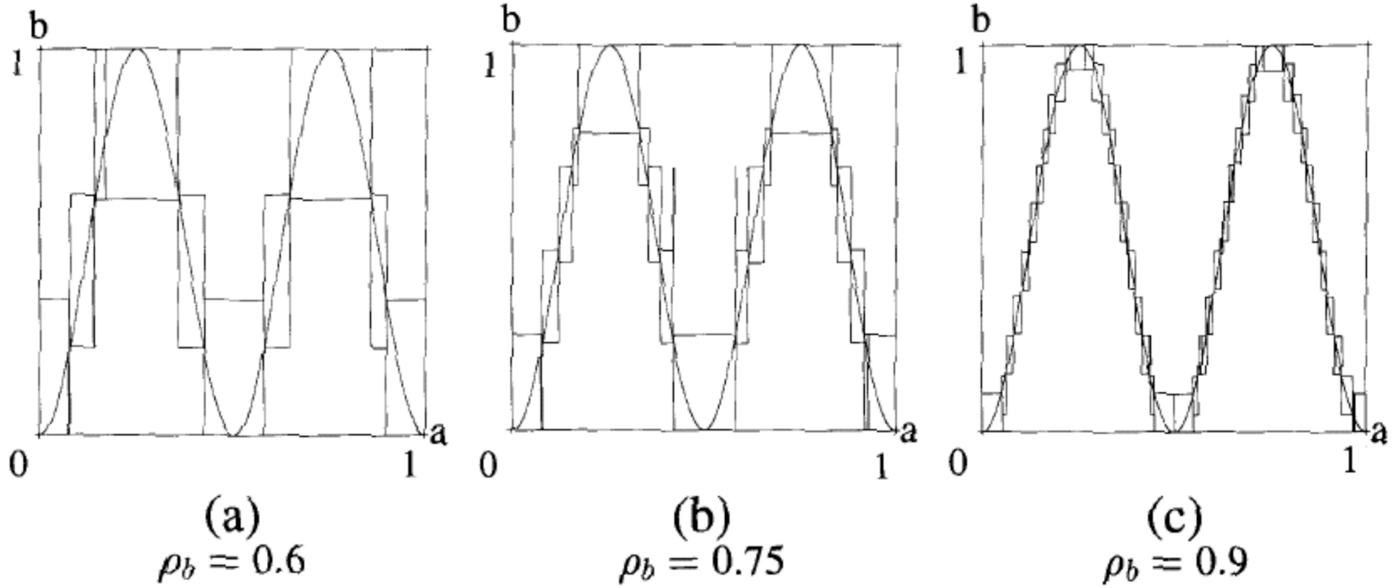


Fig. 12. Incremental approximation of a sinusoidal function for ART_b vigilance parameters, with ρ_b equal to (a) 0.6, (b) 0.75, and (c) 0.9. In each simulation the fuzzy ARTMAP system was trained on 1000 randomly chosen points $a \in [0, 1]$. *Each graph shows the test set confidence intervals* R_K^b selected by the test set points. The maximum lengths of these intervals are (a) 0.4, (b) 0.25, and (c) 0.1. Graph (c), with $\rho_b = 0.9$, is close to the asymptotic state of the three graphs in Fig. 11. See Table III.

Fuzzy-ARTMAP applications

- Control
- Classification and pattern recognition
- Data mining

Yang, Maragos 1995

Min-Max classifiers

Starting point

- Boolean functions in DNF

$$B(\vec{b}),\ \vec{b} = (b_1, \ldots, b_d) \in \{0, 1\}^d; \quad b_i \in \{0, 1\}$$

- Min-max functions are obtained replacing Boolean literals by real-valued variables

$$f : [0, 1]^d \to [0, 1] \qquad x_i \in [0, 1]$$

$$f(x_1, x_2, \ldots, x_d) = \bigvee_j \bigwedge_{i \in I_j} l_i, \quad l_i \in \{x_i, 1 - x_i\}$$

- For classification a thresholding step is added

$$\theta \in [0, 1]$$

$$f_\theta(\vec{x}) = P[f(\vec{x}) \geq \theta] = \begin{cases} 1 & \text{if } f(\vec{x}) \geq \theta, \\ 0 & \text{otherwise.} \end{cases}$$

Learning

- Minimization of the Mean Square Error (MSE)

$$\mathscr{E}(t) = E[(z(t) - d(t))^2].$$

- Gradient descent on the function parameters

$$\vec{p}(t+1) = \vec{p}(t) - \mu \nabla_{\vec{p}} \mathscr{E}(t).$$

- Instantaneous error

$$\vec{p}(t+1) = \vec{p}(t) - 2\mu(z(t) - d(t))\mu\nabla_{\vec{p}} z(t)$$

- Trick
  - Assume no input variable is complemented
  - Extend the input space to 2d including the complements … Fuzzy-ART?
- Problems
  - Define parameters to allow differentiability
  - Approximate gradient of min, max, threshold

Functional form

$$h_j = \bigwedge_{i \in I_j} x_i, \quad j = 1, 2, \ldots, k$$ clause

$$y = \bigvee_{j=1}^{k} h_j$$ expression

$$z = \begin{cases} 1 & y \geq \theta, \\ 0 & y < \theta. \end{cases}$$ Decision through threshold

- How to model continuosly the conjunctive expression structure: I_j ?
  - Continuous variables m_{ij} such that
    - x_i is included in I_j if $m_{ij} \geq 0$,
    - x_i is excluded from I_j if $m_{ij} < 0$.
  - The parameters to be learnt

$$\vec{p}(t) = (\theta(t), m_{11}(t), \ldots, m_{d1}(t), \ldots, m_{dk}(t)).$$

- Derivative with respect to the threshold

$$\frac{\partial z}{\partial \theta} = \begin{cases} -\frac{1}{2\beta} & \text{if } |y-\theta| \le \beta, \\ 0 & \text{otherwise.} \end{cases}$$

- Where β is the width of a pulse approximating the derivative of the step function

- Derivative with respect to the structure parameters

$$\frac{\partial z}{\partial m_{ij}} = \frac{\partial z}{\partial y}\frac{\partial y}{\partial h_j}\frac{\partial h_j}{\partial m_{ij}}$$

- Implies the derivative of maximum and minimum functions.

Derivative of maximum

- Implicit formulation of maximum

$$G(y, h_1, \ldots, h_k) = \sum_{j=1}^{k} \{U_3(y - h_j) - 1\} + \frac{G_e}{2} = 0$$

$$U_3(x) = \begin{cases} 1 & \text{if } x > 0, \\ \frac{1}{2} & \text{if } x = 0, \\ 0 & \text{if } x < 0. \end{cases}$$

- Leads to the following expression

$$\frac{\partial y}{\partial h_j} \approx \begin{cases} \dfrac{1}{N_{max}} & \text{if} \quad 0 \leq y - h_j \leq \beta \\ 0 & \text{otherwise.} \end{cases}$$

$$N_{max} \overset{\Delta}{=} \text{number of } h_j\text{'s such that } y - h_j \leq \beta$$
$$= \sum_{j=1}^{k} U_2(\beta - (y - h_j)).$$

Results on handwritten digit recognition

Table 1. Results for 0–1 classification problem employing *both* shape-size histograms and Fourier descriptors

Distinguishing 0's and 1's Normalized radial size histograms and Fourier descriptors					
	Min–max			Neural network	
No. of minima	% error (train)	% error (test)	Network	% error (train)	% error (test)
1	0.083	0.25	1, 1	0.083	0
3	0.083	0.25	3, 1	0.083	0
5	0.1	0.25	5, 1	0.083	0
7	0.083	0.25	7, 1	0.083	0
Normalized shape-size histograms with 2 × 2 square and Fourier descriptors					
1	3.867	2.6	1, 1	0.633	1.2
3	1.9	2.8	3, 1	0.633	0.85
5	1.083	3	5, 1	0.567	0.8
7	1.733	3	7, 1	0.533	0.55

The top two tables are generated using normalized radial histograms and Fourier descriptors, while the lower two using normalized shape-size histogram with 2 × 2 square and Fourier descriptors.

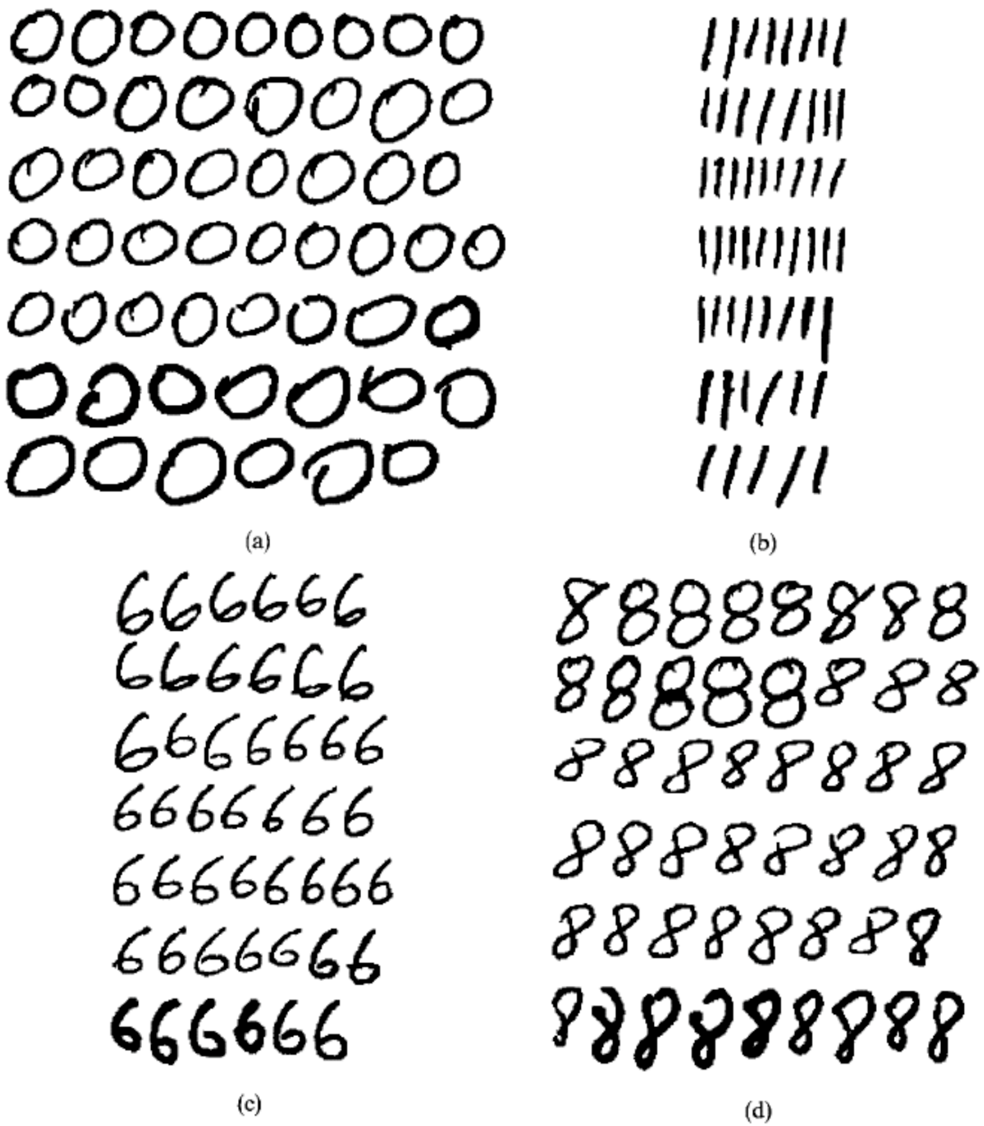


Fig. 4. Sample data from the handwritten database. (a) A collection of 0's. (b) A collection of 1's. (c) A collection of 6's. (d) A collection of 8's.

Associative Morphological Memories

Ritter, Sussner

Starting point

- Linear neuron

$$\tau_i(t+1) = \sum_{j=1}^{n} a_j(t) \cdot w_{ij} \qquad a_i(t+1) = f(\tau_i(t+1) - \theta_i)$$

- Matrix notation

$$T(t+1) = W \cdot \mathbf{a}(t)$$

$$\mathbf{a}(t) = (a_1(t), \cdots, a_n(t))',$$

$$T(t+1) = (\tau_1(t+1), \cdots, \tau_n(t+1))',$$

- Morphological dilative neuron:

$$\tau_i(t+1) = \bigvee_{j=1}^{n} a_j(t) + w_{ij}$$

- Matrix notation: max product $T(t+1) = W \boxtimes \mathbf{a}(t)$

$$C = A \boxtimes B$$

$$c_{ij} = \bigvee_{k=1}^{p} a_{ik} + b_{kj} = (a_{i1} + b_{1j}) \vee (a_{i2} + b_{2j}) \vee \cdots \vee (a_{ip} + b_{pj}).$$

- Morphological erosive neuron:

$$\tau_i(t+1) = \bigwedge_{j=1}^{n} a_j(t) + w_{ij}$$

- Matrix notation: min-product $T(t+1) = W \boxdot \mathbf{a}(t)$

$$C = A \boxdot B$$

$$c_{ij} = \bigwedge_{k=1}^{p} a_{ik} + b_{kj} = (a_{i1} + b_{1j}) \wedge (a_{i2} + b_{2j}) \wedge \cdots \wedge (a_{ip} + b_{pj}).$$

Morphological associative memories

- Hopfield associative memory: given an input **x** recalls response **y** as

$$\mathbf{y} = W \cdot \mathbf{x}.$$

- To store *k* vector pairs

$$(\mathbf{x}^1, \mathbf{y}^1), \cdots, (\mathbf{x}^k, \mathbf{y}^k), \text{ where } \mathbf{x}^\xi \in R^n \text{ and } \mathbf{y}^\xi \in R^m$$

$$W = \sum_{\xi=1}^{k} \mathbf{y}^\xi \cdot (\mathbf{x}^\xi)'.$$

- The Hopfield associative memory provides **perfect recall** if the input patterns are orthogonal
- If they are not orthogonal, the recall is corrupted by crosstalk noise.

- Morphological Associative Memories
- Construction with a single pair:

$$W = \mathbf{y} \boxtimes (-\mathbf{x})'$$

- Recall (perfect):

$$W \boxtimes \mathbf{x} = \mathbf{y}$$

- Given a set of input-output patterns

$$\{(\mathbf{x}^\xi, \mathbf{y}^\xi) \ : \ \xi = 1, \cdots, k\}$$

- Define: (X, Y),

$$X = (\mathbf{x}^1, \cdots, \mathbf{x}^k) \qquad Y = (\mathbf{y}^1, \cdots, \mathbf{y}^k).$$

- Two natural morphological memories

$$W_{XY} = \bigwedge_{\xi=1}^{k} [\mathbf{y}^\xi \times (-\mathbf{x}^\xi)'] \quad \text{and} \quad M_{XY} = \bigvee_{\xi=1}^{k} [\mathbf{y}^\xi \times (-\mathbf{x}^\xi)'].$$

- Basic recall property:
  - the erosive and dilative memory recalls bound the exact response

$$W_{XY} \leq \mathbf{y}^{\xi} \times (-\mathbf{x}^{\xi})' \leq M_{XY}$$

$$W_{XY} \boxed{\vee}\, \mathbf{x}^{\xi} \leq \mathbf{y}^{\xi} \leq M_{XY} \boxed{\wedge}\, \mathbf{x}^{\xi}$$

$$W_{XY} \boxed{\vee}\, X \leq Y \leq M_{XY} \boxed{\wedge}\, X.$$

- Conditions for perfect recall

Theorem 2: W_{XY} is $\boxtimes$-perfect for (X, Y) if and only if for each $\xi = 1, \cdots, k$, each row of the matrix $[\mathbf{y}^\xi \times (-\mathbf{x}^\xi)'] - W_{XY}$ contains a zero entry. Similarly M_{XY} is $\boxtimes$-perfect for (X, Y) if and only if for each $\xi = 1, \cdots, k$, each row of the matrix $M_{XY} - [\mathbf{y}^\xi \times (-\mathbf{x}^\xi)']$ contains a zero entry.

Autoassociative memories

- When *X*=*Y*, memories W_{XX} and M_{XX} are called autoassociative.
- They have perfect recall and unlimited capacity

$$W_{XX} \boxed{\vee} X = X \text{ and } M_{XX} \boxed{\wedge} X = X.$$

- Recalling converges in one step

Noise

- Memory W_{XX} is robust to erosive noise and sensitive to dilative noise
- Memory M_{XX} is robust to dilative noise and sensitive to erosive noise

$\tilde{\mathbf{x}}^{\gamma} \leq \mathbf{x}^{\gamma}$ Erosive noise

$\tilde{\mathbf{x}}^{\gamma} \geq \mathbf{x}^{\gamma}$ Dilative noise

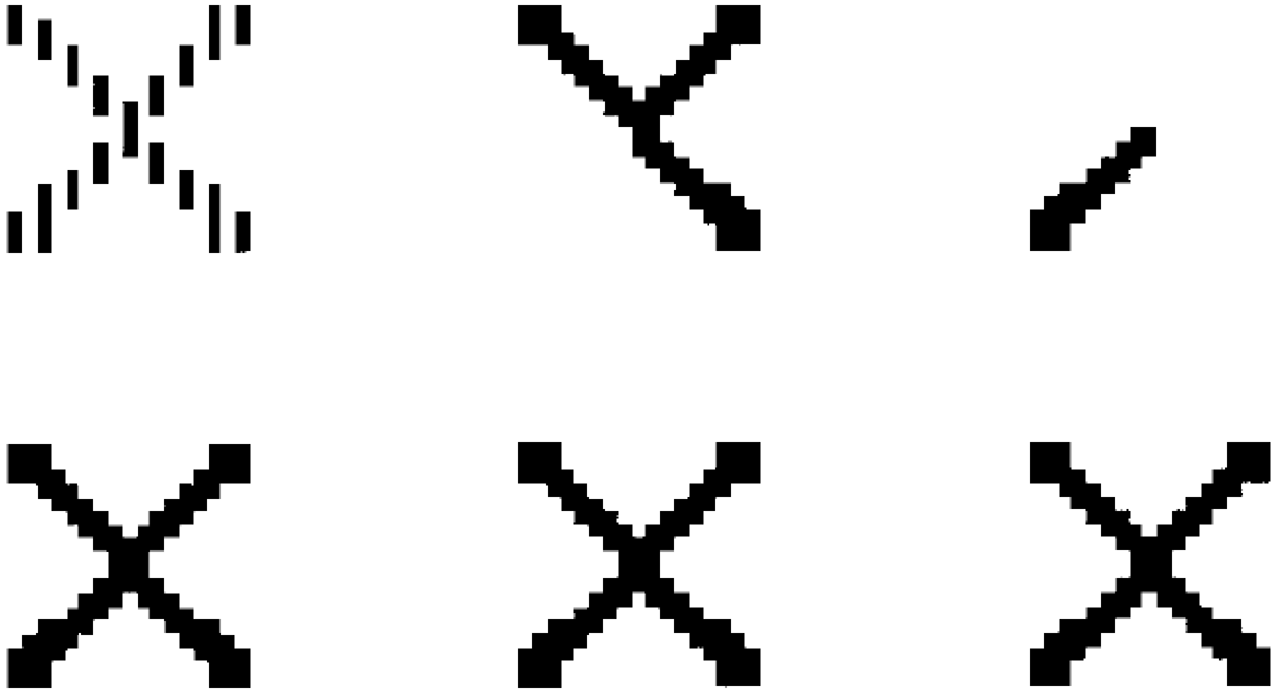


Fig. 4. The top row shows the corrupted input patterns and the bottom ro
the corresponding output patterns of the morphological memory W_{XX}.

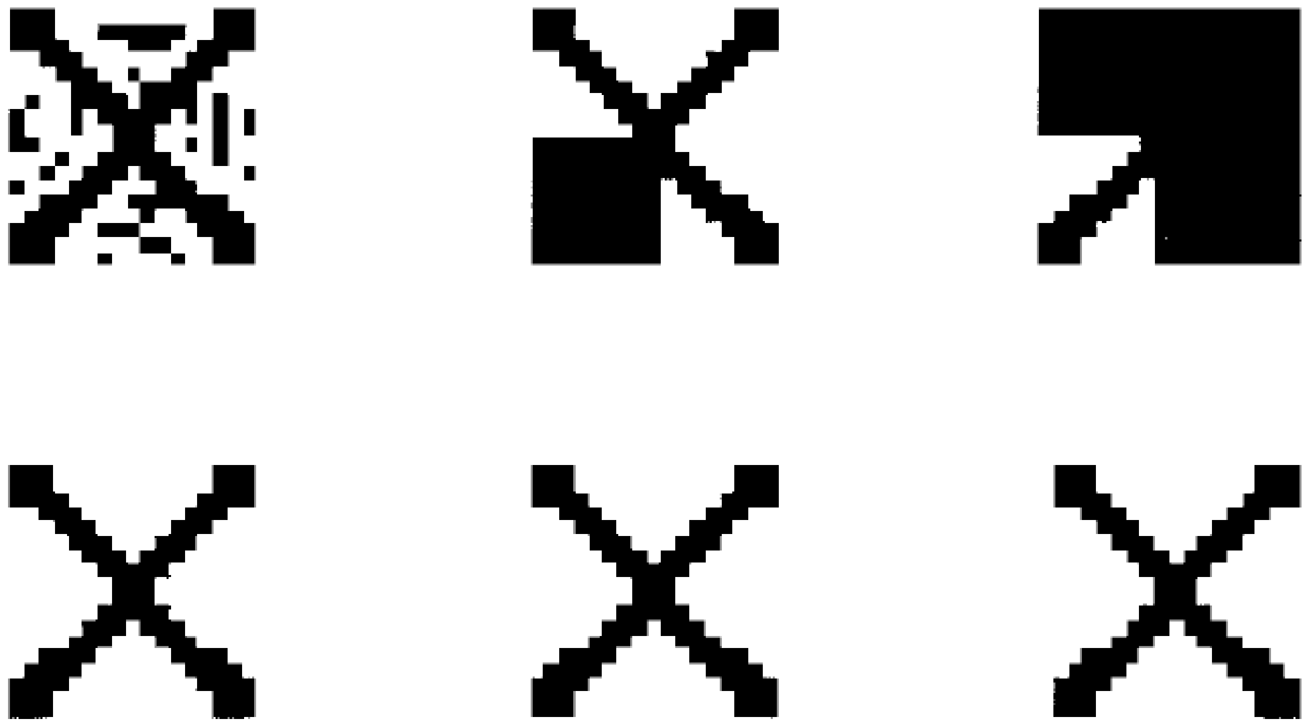


Fig. 5. The top row shows the corrupted input patterns and the bottom row the corresponding output patterns of the morphological memory M_{XX}.

Approaches to solve the noise problem

- Definition of kernels

Definition 2: Let $Z = (\mathbf{z}^1, \mathbf{z}^2, \cdots, \mathbf{z}^k)$ be an $n \times k$ matrix. We say that Z is a *kernel* for (X, Y) if and only if the following two conditions are satisfied:

1. $M_{ZZ} \boxed{\vartriangle} X = Z$;
2. $W_{ZY} \boxed{\triangledown} Z = Y$.

It follows that if Z is a kernel for (X, Y), then

$$W_{ZY} \boxed{\triangledown} (M_{ZZ} \boxed{\vartriangle} X) = W_{ZY} \boxed{\triangledown} Z = Y.$$

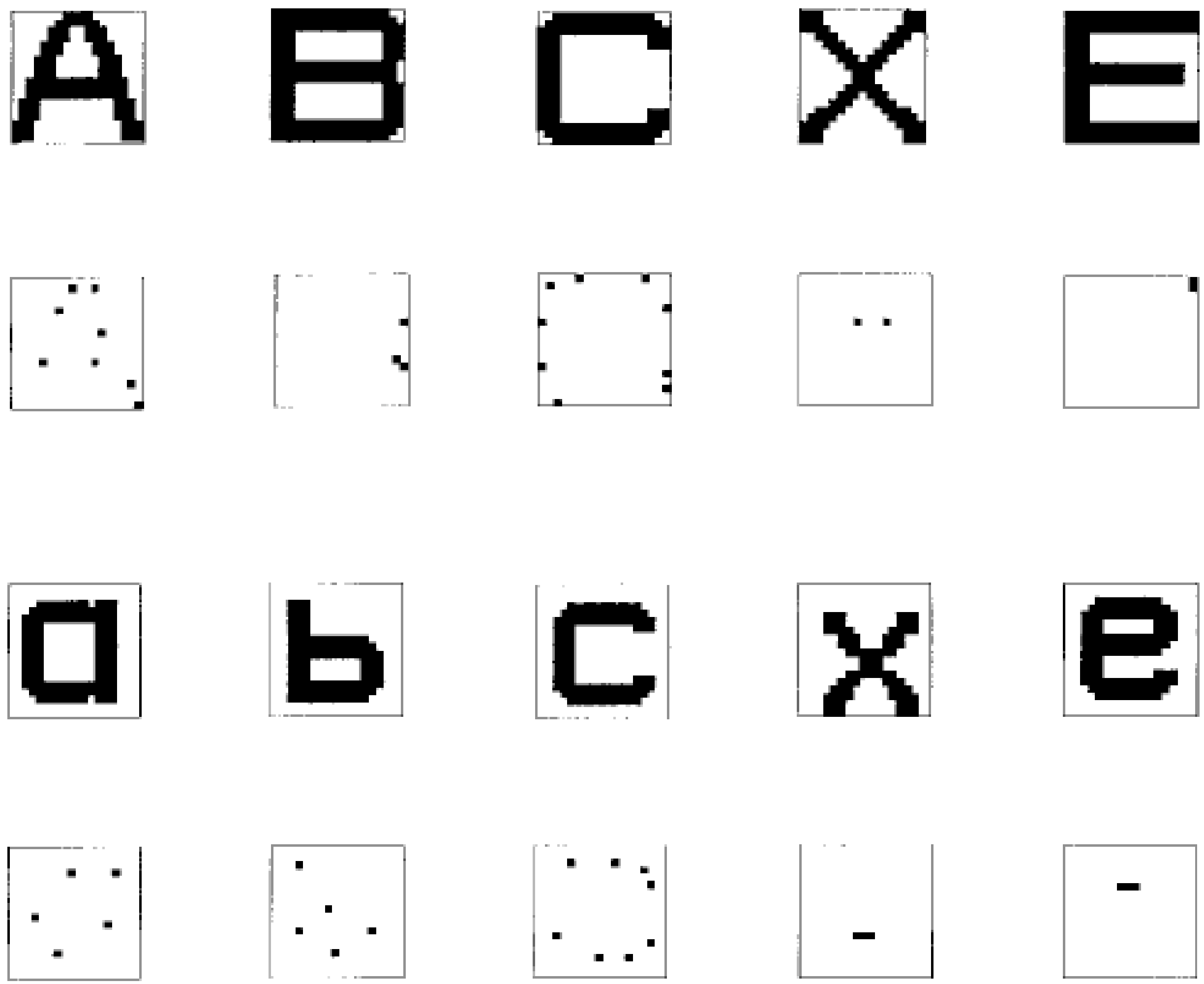


Fig. 6. An example of kernel images. The kernel image corresponding toa particular letter image is the image directly below the letter image.

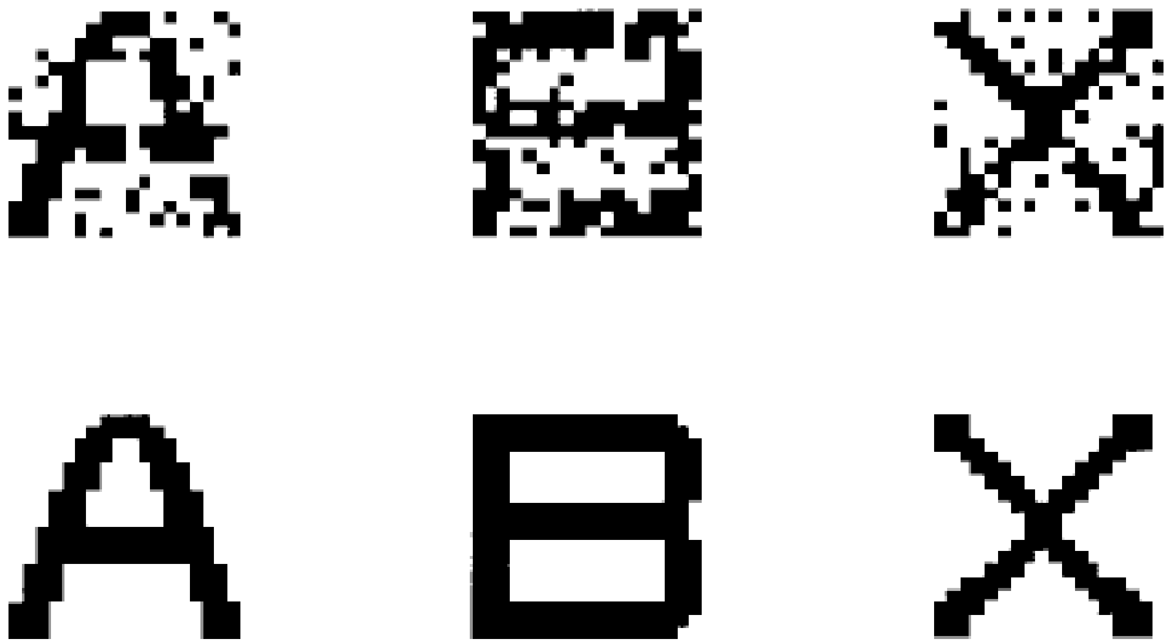


Fig. 7. An example of the behavior of the memory $\{input \rightarrow M_{ZZ} \rightarrow W_{ZY} \rightarrow output\}$. The memory was trained using the ten exemplars shown in Fig. 2. Presenting the memory with the corrupted patterns of the letters A, B, and X resulted in perfect recall (lower row). Each letter was corrupted by randomly reversing each bit with a probability of 0.15.

Lattice Image Processing: A Unification of Morphological and Fuzzy Algebraic Systems

P. Maragos

Starting point

- Design of new filters: generalized opening and closing
- Works on the lattice of functions

$$\mathcal{S} = \mathbb{V}^E \qquad F : E \to \mathbb{V}$$

$$F \le G \Leftrightarrow F(x) \le G(x) \;\; \forall x \in E$$

Inherited partial order

$$\Big(\bigvee_{i \in J} F_i\Big)(x) \triangleq \bigvee_{i \in J} F_i(x), \quad x \in E$$

$$\Big(\bigwedge_{i \in J} F_i\Big)(x) \triangleq \bigwedge_{i \in J} F_i(x), \quad x \in E$$

Inherited supremum and infimum

Increasing operators

δ is **dilation** iff $\delta(\bigvee_{i\in J} X_i) = \bigvee_{i\in J} \delta(X_i)$

ε is **erosion** iff $\varepsilon(\bigwedge_{i\in J} X_i) = \bigwedge_{\mathrm{i}\in \mathrm{J}} \varepsilon(X_i)$

α is **opening** iff α is increasing, idempotent & anti-extensive

β is **closing** iff β is increasing, idempotent & extensive

Adjunction

- The operator pair (ε, δ) is an **adjunction** if

$$\delta(X) \leq Y \Leftrightarrow X \leq \varepsilon(Y) \quad \forall X, Y \in \mathcal{L}$$

- An adjunction defines a pair of morphological filters

$\delta\varepsilon$ *is an opening, and* $\varepsilon\delta$ *is a closing.*

Signal processing

- Algebraic structure of the scalars:

$$(\mathbb{V}, \vee, \wedge, \star, \star')$$

- Complete lattice-ordered double monoid
  - Addition $\vee$
  - Dual addition $\wedge$
  - Multiplication $\star$
  - Dual multiplication $\star'$

Signal processing

- The space of signals is a function lattice

$$\mathcal{S} = \mathrm{Fun}(E, \mathbb{V})$$

- It inherits the clodum structure of the scalars, with appropriate natural definitions of addition and multiplication

Parallelism to linear processing

- Representation of a signal as a supremum (infimum) of translated impulses

$$F(x) = \bigvee_{y \in E} F(y) \star q_y(x) = \bigwedge_{y \in E} F(y) \star' q'_y(x)$$

- Linear superposition principle

$$\psi\left(\sum_{i\in J} a_i \cdot F_i\right) = \sum_{i\in J} a_i \cdot \psi(F_i)$$

- Nonlinear superposition principle

$$\delta\left(\bigvee_{i\in J} c_i \star F_i\right) = \bigvee_{i\in J} c_i \star \delta(F_i),$$

- Translation invariant operator: commutes with all translations

$$\tau \in \mathbb{T}; \text{ i.e. } \psi\tau = \tau\psi .$$

- Nonlinear convolutions define the effect of Erosion and Dilation translation invariant systems

$$\psi \text{ is LTI} \Leftrightarrow \psi(F)(x) = (F * H)(x)$$
$$= \sum_{y} F(y) H(x-y)$$

DTI
$$(F \circledast H)(x) \triangleq \bigvee_{y \in \mathbb{E}} F(y) \star H(x-y)$$

ETI
$$(F \circledast' H')(x) \triangleq \bigwedge_{y \in \mathbb{E}} F(y) \star' H'(x-y)$$

Generalized convolution adjunctions

- using scalar adjunctions $(\overleftarrow{\lambda}_{H(x-y)}, \lambda_{H(x-y)})$
- It is possible to obtain the adjoint operator

$$\Delta_H(F)(x) = \bigvee_{y \in \mathbb{E}} F(y) \star H(x-y) \quad = \bigvee_{y \in \mathbb{E}} \lambda_{H(x-y)}(F(y))$$

- Which looks like a correlation

$$\mathcal{E}_H(G)(x) = \bigwedge_{y \in \mathbb{E}} G(y) \star [H(y-x)]^*$$

Lattice operators using fuzzy norms

- Fuzzy intersection norm --> scalar dilation

$$T: [0, 1]^2 \rightarrow [0, 1]$$

F1. $T(a, 1) = a$ and $T(a, 0) = 0$

F2. $T(a, T(b, c)) = T(T(a, b), c)$ (associativity).

F3. $T(a, b) = T(b, a)$ (commutativity).

F4. $b \leq c \Rightarrow T(a, b) \leq T(a, c)$ (increasing).

F5. T is a continuous function.

- Fuzzy union norm --> scalar erosion

$$U: [0, 1]^2 \to [0, 1]$$

$$\text{F1}'.\ U(a, 0) = a \text{ and } U(a, 1) = 1.$$

- Translations under the fuzzy framework

$$\mathcal{S} = \mathrm{Fun}(\mathbb{E}, [0, 1])$$

$$\tau_{h,v}(f)(x) = T(f(x - y), v)$$

$$\tau'_{h,v}(f)(x) = U(f(x - y), v)$$

$$(h, v) \in \mathbb{E} \times [0, 1]$$

- Signal representation with fuzzy translations

$$f(x) = \bigvee_{y} T[q(x-y), f(y)]$$
$$= \bigwedge_{y} U[q'(x-y), f(y)]$$

$$q(x) \triangleq \begin{cases} 1, & x = \vec{0} \\ 0, & x \neq \vec{0} \end{cases}, \quad q'(x) \triangleq \begin{cases} 0, & x = \vec{0} \\ 1, & x \neq \vec{0} \end{cases}$$

- Translation invariant signal fuzzy **dilations** and **erosions** with sup-*T* and inf-*U* convolutions

$$(f \bigcirc_T g)(x) \triangleq \bigvee_y T[g(x-y), f(y)],$$

$$(f \bigcirc'_U g)(x) \triangleq \bigwedge_y U[g(x-y), f(y)]$$

- Fuzzy dilation adjoint $\Delta_{H,T}(F)(x) \triangleq (F \bigcirc_T H)(x)$

$$\mathcal{E}_{H,\Omega}(G)(y) \triangleq \bigwedge_{x \in \mathbb{E}} \Omega[H(x-y), G(x)]$$

where $\Omega[H(x-y), G(x)]$ is actually the adjoint of the fuzzy T-norm:

$$T(a, v) \leq w \Leftrightarrow v \leq \Omega(a, w)$$

$$\Omega(a, w) = \sup\{v \in [0, 1] : T(a, v) \leq w\}$$

Example norms

Fuzzy intersection norm

$$\text{Min} : T_1(a, v) = \min(a, v)$$

$$\text{Product} : \quad T_2(a, v) = a \cdot v$$

$$\text{Yager} : T_3(a, v) = 1 - \Big(1 \wedge [(1 - v)^p + (1 - a)^p]^{1/p}\Big), \quad p > 0.$$

Adjoint norm

$$\Omega_1(a, w) = \begin{cases} w, & w < a \\ 1, & w \geq a \end{cases}$$

$$\Omega_2(a, w) = \begin{cases} \min(w/a, 1), & a > 0 \\ 1, & a = 0 \end{cases}$$

$$\Omega_3(a, w) = \begin{cases} 1 - [(1 - w)^p - (1 - a)^p]^{1/p}, & w < a \\ 1, & w \geq a \end{cases}$$

Results

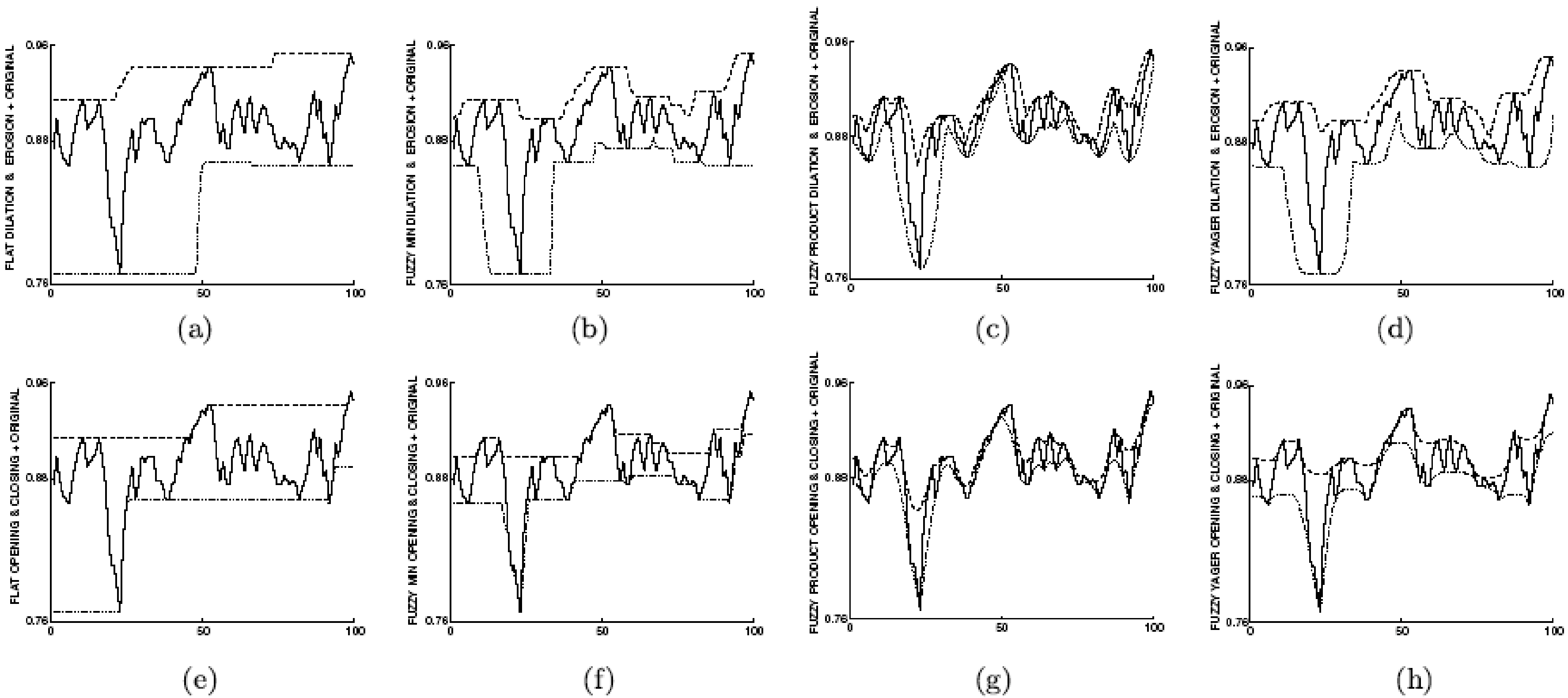


Figure 1. Comparison of 1D basic morphological and lattice-fuzzy signal operators. Rows 1 and 2, left to right: flat, minimum, product, Yager. Row 1: original signal (solid line), dilation (dashed line), erosion (dotted line). Row 2: closing (dashed line), opening (dotted line). Courtesy of [27].

Modelling and Knowledge representation based in Lattice Theory

V. G. Kaburlasos

Starting point

- Generalizes the Fuzzy-ART and Fuzzy-ARTMAP architectures
- The Fuzzy Lattice Neurocomputing
  - Proposes an abstract representation (FIN) based on generalized interval (GI).
  - Is defined based on inclusion measures and distances on the FINs

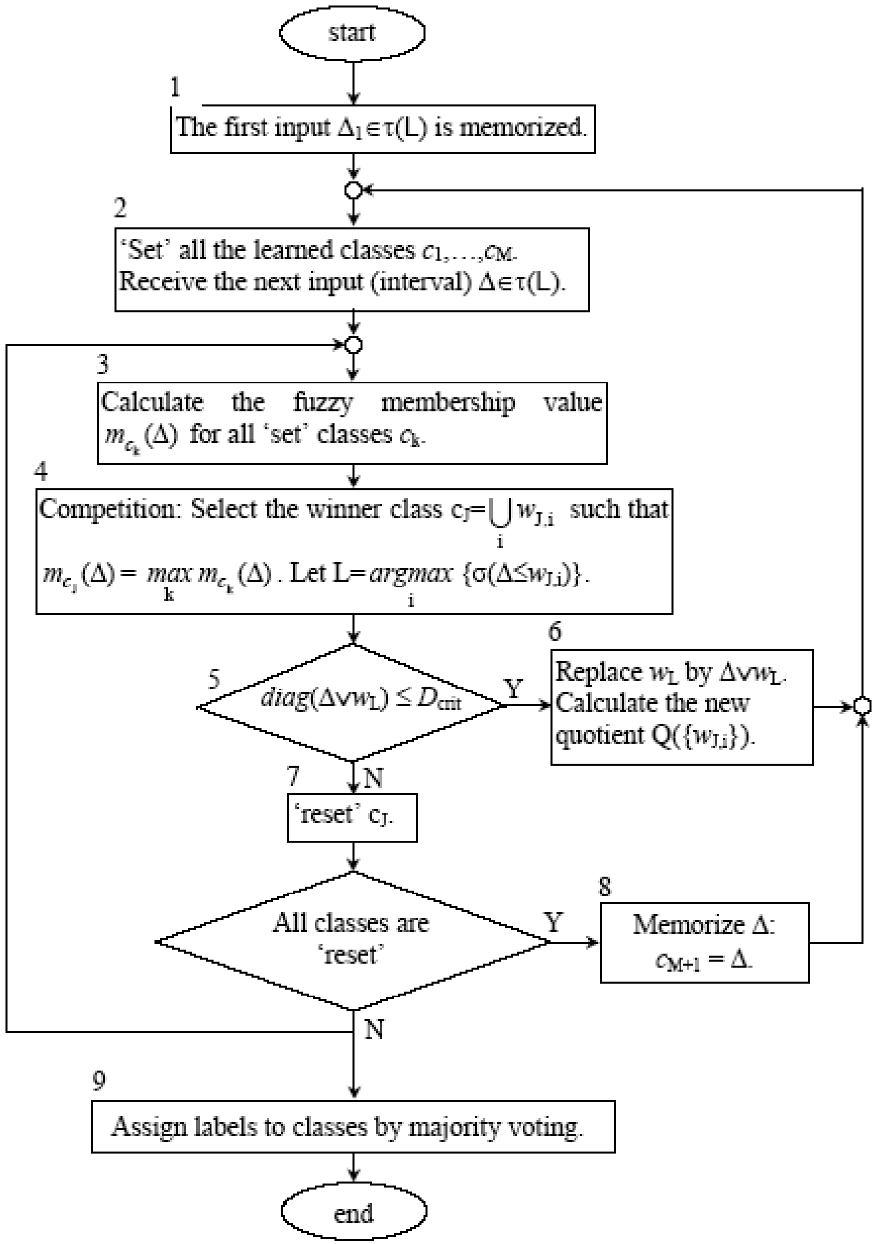


Fig.7-1 Flowchart of algorithm σ-FLN for learning (training).

Inclusion measure

$\sigma(\Delta \leq w_{J,i})$

Vigilance parameter

D_{crit}.

learning

$\Delta \vee w_L$.

Advantages of σ-FLN

- Deals with data uncertainty
- Different positive valuation functions
- Deals with disparate (lattice) data types
- *Missing* and *don't care* values are treated naturally: least and greatest lattice elements.
- Learning in one step, presentation order dependent

Intervals in the unit hypercube

- Lattice interval corresponds to a hyperbox

$$\Delta = [a,b] = [(a_1,\ldots,a_N),(b_1,\ldots,b_N)] = [a_1,b_1,\ldots,a_N,b_N],$$

- Positive valuation function

$$v(w) = v(\theta(p)) + v(q) = N + \sum_{i=1}^{N}(q_i - p_i)$$

- Lattice join

$$\Delta \vee w = [a_1,b_1,\ldots,a_N,b_N] \vee [p_1,q_1,\ldots,p_N,q_N] =$$

$$[a_1 \wedge p_1,\ b_1 \wedge q_1,\ldots,a_N \vee p_N,\ b_N \vee q_N].$$

- Degree of inclusion

$$\sigma(\Delta \leq w) = \frac{v(\theta(p)) + v(q)}{v(\theta(a \vee p) + v(b \vee q)} = \frac{N + \sum_{i=1}^{N} (v_i(q_i) - v_i(p_i))}{N + \sum_{i=1}^{N} [v_i(b_i \vee q_i) - v_i(a_i \wedge p_i)]}.$$

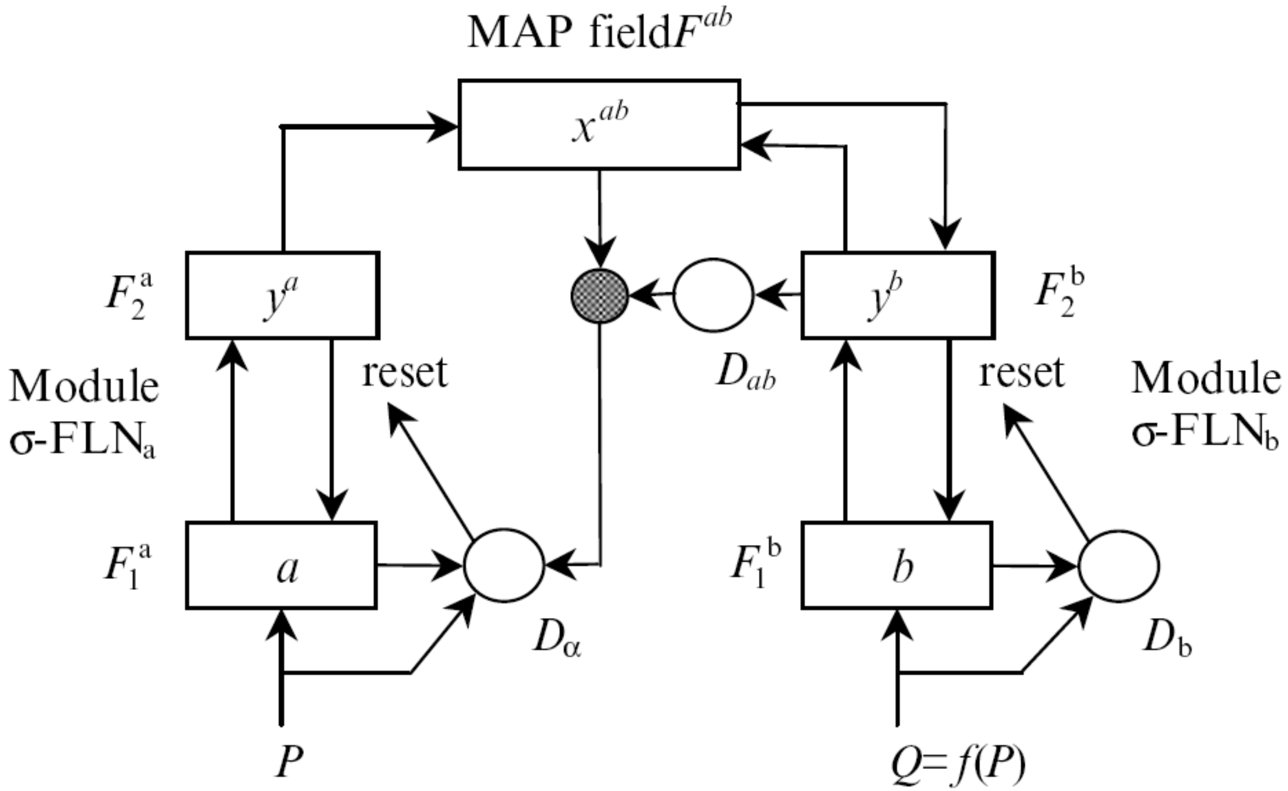


Fig.7-4 The σ-FLNMAP neural network for inducing a function f: τ(L)→τ(K), where both L and K are mathematical lattices.

Generalization

- Positive Valuation function on a lattice (L,≤) satisfies

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

$$x < y \Rightarrow v(x) < v(y)$$

- A positive valuation in a lattice (L,≤) induces a metric (distance) $d : L \times L \to R_0^+$

$$d(x, y) = v(x \vee y) - v(x \wedge y)$$

- An inclusion measure is a function $\sigma : L \times L \to [0,1]$ satisfying

(IM1) $\sigma(x,x) = 1, \forall x \in L$

(IM2) $x \wedge y < x \Rightarrow \sigma(x,y) < 1$

(IM3) $u \leq w \Rightarrow \sigma(x,u) \leq \sigma(x,w)$

- If *v* is a positive valuation in lattice (L,≤) then both expressions are inclusion measures

(a) $k(x,u) = \dfrac{v(u)}{v(x \vee u)}$ (b) $k(x,u) = \dfrac{v(x \wedge u)}{v(x)}$

Fuzzy Interval Numbers (FIN)

- A **FIN** is a function $F:(0,1]\rightarrow M$ such that

  (1) $F(h)\in M^h$

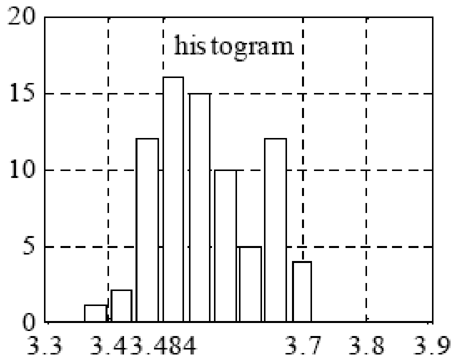
  (2) either $F(h)\in M_+^h$ or $F(h)\in M_-^h$

  (3) $h_1 \leq h_2 \Rightarrow \{x : F(h_1)\neq 0\} \supseteq \{x : F(h_2)\neq 0\}$

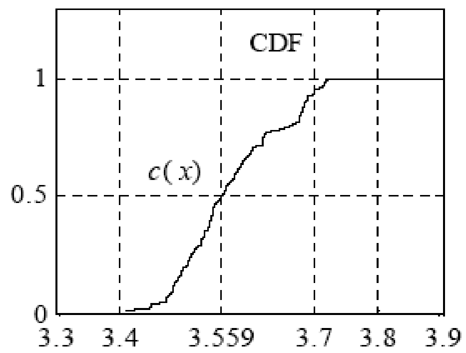
- where M^h denotes the set of generalized intervals of heigh h. It is a **lattice ordered linear space**.

- FINs can be models of
  - Real numbers
  - Intervals
  - Fuzzy numbers
  - Probability distributions
- FINs inherit valuation, inclusion, metric functions from the set of generalized intervals

Probability distribution FIN

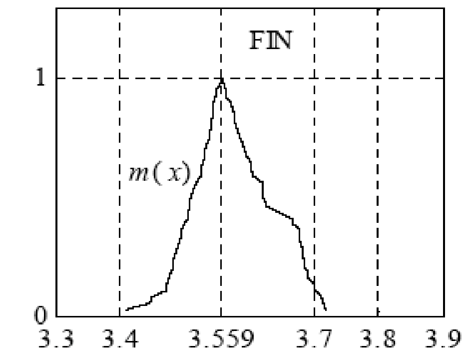


(a)



(b)

$$m(x) = \begin{cases} 0.5c(x), & x \leq 3.559 \\ 1\text{-}0.5c(x), & x > 3.559 \end{cases}$$



(c)

Applications

- Classification and clustering
  - Benchmark problems
  - Epidural surgery planification
  - Orthopedics bone drilling
  - Ambient ozone estimation
  - Prediction of industrial sugar production

Table 8-8 Recognition results for the Sonar benchmark data set.

Method	% Right on Testing
σ-FLNMAP for classification	96.15
σ-FLN for clustering	94.23
Fuzzy ARAM	94.20
K-Nearest Neighbor	91.60
BackProp: Angle-Dep. (1)	90.40
Nearest Neighbor	82.70
BackProp: Angle-Ind. (2)	77.10

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

Table 9-3 Performance of various methods on the Forest Covertype benchmark data set. The last column shows the number of induced rules.

Method	% Correct	no. of rules
Backpropagation	70	(6600)
FLNff	68.25	654
FLNmtf	66.27	1566
FLNotf	66.13	516
FLNsf	66.10	503
FLNtf	62.58	3684
Linear discriminant analysis	58	-

Conclusions

Conclusions

- Lattice computing defined as computing on the lattice algebra $(R, \wedge, \vee, +)$ has been maintaining its appeal in the last fifteen years.

Conclusions

- Application of lattice theory leads to new computational paradigms arising from
  - Fusion of established paradigms
    - Mathematical morphology and fuzzy systems
    - Neural networks and fuzzy systems
  - Generalization of approaches
    - Fuzzy Lattice Neurocomputing

Future

- Lattice theory may be the formal framework for the development of new approaches:
  - Feature extraction based on linear unmixing based on the identification of endmembers in the data set.
  - Fusion of stochastic models (Random Markov Fields, Hidden Markov Models) and Fuzzy Systems.

Thank you for your attention