Int. J. Appl. Math. Comput. Sci., *2023*, *Vol. 33*, *No. 2*, *249–265*
DOI: 10.34768/amcs-2023-0019

BINARY ASSOCIATIVE MEMORIES WITH COMPLEMENTED OPERATIONS

ARTURO GAMINO-CARRANZA [a]

[a] Academic Subdirectorate
Merida Technological Institute/National Technological Institute of Mexico
Av. Tecnológico S/N Km. 4.5, C.P. 97118, Mérida, Yucatán, Mexico
e-mail: arturo.gc@merida.tecnm.mx

Associative memories based on lattice algebra are of great interest in pattern recognition applications due to their excellent storage and recall properties. In this paper, a class of binary associative memory derived from lattice memories is presented, which is based on the definition of new complemented binary operations and threshold unary operations. The new learning method generates memories $\mathbf{M}$ and $\mathbf{W}$; the former is robust to additive noise and the latter is robust to subtractive noise. In the recall step, the memories converge in a single step and use the same operation as the learning method. The storage capacity is unlimited, and in autoassociative mode there is perfect recall for the training set. Simulation results suggest that the proposed memories have better performance compared to other models.

Keywords: associative memory, neural network, morphological associative memory, perfect recall.

1. Introduction

One of the most interesting primary functions of the human brain is the ability to associate information. This function, known as *associative memory*, allows us to store, maintain and recall information. For example, we can learn the relationship between a person's face and their name, memorize this face–name association for years, and remember the person's name by recognizing their face, even if they have aged. Associative memory is a special type of one-layer artificial neural network that stores and recalls patterns. This memory is modeled and represented as a system that stores input–output pattern associations $(\mathbf{x}, \mathbf{y})$ and recalling patterns based on complete inputs or distorted versions of them (Hassoun, 1993; Ritter and Urcid, 2021). If the vectors $\mathbf{x}$ and $\mathbf{y}$ are equal, then the memory is said to operate in autoassociative mode; if they are different, then it operates in heteroassociative mode. The association feature is used in the field of artificial intelligence and image recalling, and is of interest in various associative memory models for the study of learning, recalling and storage capacity, noise sensitivity, hardware design for implementation and processing time.

In this article, an associative memory is presented with a new method of computing the learning and recalling phases for binary patterns; for this purpose, two new operations are defined, a binary $\circ$ and a unary $^{\downarrow}$. Two learning algorithms are generated to calculate memories $\mathbf{M}$ and $\mathbf{W}$. The first one is the result of the superposition of maxima of the partial calculations of the operation $\circ$ between each pair of input–output vectors, while the second one uses the superposition of maxima of the partial computations of the complement of the operation $\circ$. In the recalling phase, the output is obtained from the result of the superposition of the maxima of the calculation of the operation $\circ$ (or its complement) between memory $\mathbf{M}$ (or $\mathbf{W}$) and the input presented to the memory. The unary operation $^{\downarrow}$ is used as a threshold in the internal association of input (learning phase) and in the recalling of the input–output association (recalling phase). The associative memories $\mathbf{M}$ and $\mathbf{W}$ are robust to additive and subtractive noise, respectively, and their convergence is in a single step in both the heteroassociative and autoassociative modes. In the latter mode, there is unlimited pattern storage capacity and perfect recall for the training set. For example, an associative memory (content-addressable memory) can be used to significantly reduce the time to find an object stored in a memory buffer by the content of its data; this idea can be employed in more complex queuing systems such as those described by Tikhonenko *et al*. (2021).

The rest of the article is organized as follows. Works related to associative memory models are briefly presented in Section 2. Associative memory concepts are introduced in Section 3. Theoretical foundations and learning-recalling methods of the proposed associative memories are defined in Section 4. Numerical and computer simulations results of memory performance in comparison with other models are explained in Section 5. A discussion and some conclusions are presented in Section 6. Finally, the proofs of the theorems are shown in Appendix.

2. Related works

Hopfield (1982) proposed an autoassociative model of bipolar patterns. This network learns through Hebb's rule considering that, on the main diagonal of the weight matrix, its entries are equal to zero. The network recursively recalls patterns until it converges to a local minimum. It has been shown that the number of perfectly stored patterns that are recalling without error by the Hopfield memory of n neurons is asymptotically less than or equal to $n/(4\log n)$ as $n \to \infty$ (McEliece *et al.*, 1987). By modifying the Hopfield network with a hysteresis activation function, it converges much faster and shows better performance (Xia *et al.*, 2004).

In a chaotic element network model each element is updated in time according to a function called a logistic map, and its elements are coupled with each other, either globally or by the nearest neighbor. Based on this type of model, Ishi *et al.* (1996) introduced one with a cubic function that generates a symmetric map (S-GCM), which implies that the map has at most two periodic orbits that attract each other and the bistability of the function is controlled by the evolution of a parameter α. The learning method uses the conventional covariance matrix and the recovery uses the global coupling of the symmetric function, which recursively converges to a local minimum. Other models derived from the S-GCM model are the new parameter control method for S-GCM proposed by Zheng and Tang (2005), which shows that the introduction of a new control parameter α increases the speed of the model; the globally coupled map using the sine map presented by Wang *et al.* (2012) replaces the cubic function of the S-GCM by a sinusoidal function, which introduces a control parameter μ; the globally coupled map using cubic logistic map exposed by Wang and Jia (2017) exhibits a new cubic function different from that of S-GCM, which controls its bistability through a parameter μ. Lee and Farhat (2001) presented two chaotic neural network models named parametrically coupled sine map networks, which use the bistability of the sinusoidal function map to encode binary patterns. Both models in their learning phase use Hebb's rule to determine the weight matrix and for its recovery the coupling of the maps is employed. The first one uses the amplitude of the sinusoidal function as a bifurcation parameter to control bistability, while the other uses an offset parameter of the sinusoidal function. These models operate recursively until convergence to either of the two attracting points on the map to determine the output.

Based on fuzzy set theory, Kosko (1991) introduced one of the first models of fuzzy associative memories, which stores in its learning matrix the fuzzy rules that associate patterns; however, this model has storage limitations because it uses a matrix to store each fuzzy rule, which means that all fuzzy outputs generated by each matrix are combined for pattern recalling. To reduce this limitation, Zhang *et al.* (1993) proposed an improved model of optical fuzzy associative memory through the computation of a single weight matrix combining all fuzzy rules, for which he designed a learning algorithm based on the adoption of Gödel's fuzzy implication operators and the superposition of minima. Similarly, the fuzzy relational memories proposed by Chung and Lee (1994) combine all fuzzy rules in a single weight matrix; however, unlike in Zhang's model, a max-t composition is used for pattern encoding, where t is a triangular norm. Xiao *et al.* (1997) proposed a max-min encoding learning algorithm for a fuzzy max-multiplication model, which uses a max-min implication operator in the learning algorithm to determine the weight matrix, while the max-multiplication operator is used in pattern recalling. The max-min fuzzy neural network with a threshold proposed by Liu (1999) increases the storage capacity by adding a threshold to each network node, i.e., one threshold for input layer and another for output.

Another type of associative memory includes those that base their framework on lattice algebra. Ritter *et al.* (1998) introduced morphological associative memories, which replace the operations of the field $(\mathbb{R}, +, \times)$ of the classical models by semirings $(\mathbb{R}_{\pm\infty}, \vee, \wedge, +, +')$. These memories exhibit better capabilities, such as convergence in a single step, greater tolerance to noisy inputs, better storage capacity and, in the autoassociative mode, unlimited capacity and perfect recall for the training set. Wang and Lu (2004) proposed a new fuzzy morphological associative memory which replaces the operations used in the morphological memories of $+$ by $\times$ and $+'$ by $\div$, while Feng *et al.* (2015) proposed a logarithmic and exponential morphological associative memory which similarly replaces the operations of $+$ and $+'$ for the logarithm and exponential, respectively; both memories retain the main characteristics of morphological memories. Feng *et al.* (2009) presented a new method to obtain morphological associative memories, which consists in exchanging the order of the operations used in the learning and recalling phases of morphological memories, while Feng and Yao (2016) substituted the operations used in the new method to obtain morphological associative memories by the operations of division and multiplication, naming this model no rounding reverse fuzzy morphological associative memory; these two memory models show in some cases better performance than morphological memories in the heteroassociative mode. Sussner and Valle (2006) developed implicative fuzzy associative memories; this model uses the Gödel, Goguen, and Lukasiewicz R-implications for learning and their dual operations in pattern recall.

3. Associative memory concepts

An associative memory $\mathbf{M}$ expresses an input–output patterns relations represented symbolically by $\mathbf{x} \to \mathbf{M} \to \mathbf{y}$. The purpose of the memory is to recall a pattern from a complete input or its distorted version (Hassoun, 1993). Considering that the patterns are binary, an input vector is represented by $\mathbf{x}^{\xi} \in A^n$ and an output vector by $\mathbf{y}^{\xi} \in A^m$, where $A = \{0, 1\}$, $\xi \in \{1, \ldots, k\}$ is an index and k is the number of pairs of binary input–output vectors that are stored in the associative memory, n and m are the dimensions of the input and output vectors, respectively, and the j-th element of the vectors $\mathbf{x}^{\xi}$ and $\mathbf{y}^{\xi}$, is represented by x_j^{ξ} and y_j^{ξ}. Let $\mathbf{X} = \left(\mathbf{x}^1, \ldots, \mathbf{x}^k\right)$ be a matrix of dimension $n \times k$ with its (i, j)-th element denoted by x_{ij} or x_i^j, and $\mathbf{Y} = \left(\mathbf{y}^1, \ldots, \mathbf{y}^k\right)$ be a matrix of dimension $m \times k$ with its (i, j)-th element denoted by y_{ij} or y_i^j; then, $\mathbf{M}$ is the associative memory that stores the k binary input–output vectors $(\mathbf{X}, \mathbf{Y})$. If $\mathbf{x}^{\xi} = \mathbf{y}^{\xi} \quad \forall \xi = 1, \ldots, k$, then $\mathbf{M}$ is said to be an *autoassociative memory*; otherwise, it is a *heteroassociative memory*.

A memory provides *perfect recall* when it is presented with an input vector $\mathbf{x}^{\xi}$ or a distorted version of it, $\tilde{\mathbf{x}}^{\xi}$, and the result is equal to the corresponding output vector $\mathbf{y}^{\xi}$. Let $I = \{1, \ldots, n\}$; then, the distorted version $\tilde{\mathbf{x}}^{\xi}$ of the vector $\mathbf{x}^{\xi} \in A^n$ is caused by three types of noise (Urcid and Ritter, 2007): *additive* if $\tilde{\mathbf{x}}^{\xi} \geq \mathbf{x}^{\xi}$, i.e., $\tilde{x}_i^{\xi} \geq x_i^{\xi} \quad \forall i \in I$; *subtractive* if $\tilde{\mathbf{x}}^{\xi} \leq \mathbf{x}^{\xi}$, i.e., $\tilde{x}_i^{\xi} \leq x_i^{\xi} \quad \forall i \in I$; and *mixed*, composed of additive and subtractive noise at the same time, i.e., $\tilde{x}_i^{\xi} > x_i^{\xi} \quad \forall i \in G$ and $\tilde{x}_i^{\xi} < x_i^{\xi} \quad \forall i \in L$, where $L, G \subset I$ are two nonempty subsets of indices and disjoint from each other.

The *similarity* between two arbitrary vectors of equal size n, $\mathbf{a} = (a_1, \ldots, a_n)$ and $\mathbf{b} = (b_1, \ldots, b_n)$, in this paper will be determined by the *Gamma binary similarity distance* (Mustafa, 2018)

$$\gamma(\mathbf{a}, \mathbf{b}) = \left|1 - \frac{2}{n} \sum_{i=1}^{n} |a_i - b_i|\right|, \tag{1}$$

where the vectors are said to be distinct if $0 \leq \gamma \leq 0.01$, or to have minimal similarity if $0.01 < \gamma \leq 0.2$, low similarity if $0.2 < \gamma \leq 0.4$, medium similarity if $0.4 < \gamma < 0.6$, good similarity if $0.6 \leq \gamma < 0.8$, high similarity if $0.8 \leq \gamma < 1$, and similar if $\gamma = 1$. This last value represents perfect recall, and memory performance can be evaluated with the *perfect recall rate*, which represents the number of training patterns perfectly stored in the memory.

Let the maximum and minimum operations be represented by the symbols $\vee$ and $\wedge$, respectively. We introduce the *max matrix product* $\mathbf{C} \boxed{\vee} \mathbf{D}$ and the *min matrix product* $\mathbf{C} \boxed{\wedge} \mathbf{D}$ of two matrices $\mathbf{C}$ and $\mathbf{D}$, of dimensions $m \times p$ and $p \times n$, respectively, given by

$$(\mathbf{C} \boxed{\vee} \mathbf{D})_{ij} = \bigvee_{k=1}^{p} (c_{ik} \cdot d_{kj}) \qquad \forall i \in \{1, \ldots, m\} \text{ and } \forall j \in \{1, \ldots, n\}, \tag{2a}$$

$$(\mathbf{C} \boxed{\wedge} \mathbf{D})_{ij} = \bigwedge_{k=1}^{p} (c_{ik} \cdot d_{kj}) \qquad \forall i \in \{1, \ldots, m\} \text{ and } \forall j \in \{1, \ldots, n\}, \tag{2b}$$

where $\cdot$ is a binary operation, for example, addition.

The design of associative memory consists of *learning* and *recalling* phases (Rani *et al.*, 2018). The former presents the operations and conditions to generate an algorithm that allows the storage of input–output matrices $(\mathbf{X}, \mathbf{Y})$ in memory $\mathbf{M}$, while the latter contains the operations and conditions to obtain an algorithm that allows perfect recall, i.e., when an input vector $\mathbf{x}^{\xi}$ or a distorted version of it is presented to memory, the obtained result corresponds to its associated output vector $\mathbf{y}^{\xi}$.

4. Method for obtaining associative memories with complemented operations

In this section the proposed model is presented, which operates in heteroassociative and autoassociative modes. New operations and their properties are defined, which support the framework of the construction methods of the memory and recalling of a vector, and the type of noise supported by the memories is characterized.

4.1. Theoretical basis. In this paper, $A = \{0, 1\} \subset \mathbb{N}$ is a binary subset, $B = \{00, 01, 10\} \subset \mathbb{N}$, $C = \{01, 10, 11\} \subset \mathbb{N}$ and $D = \{00, 01, 10, 11\} \subset \mathbb{N}$ are nonnegative subsets of integers expressed in the two-bit binary number system and $(A, \leq)$ is the well-ordered set. The variables $x, y, z \in A$ and, depending on the associative memory used, $u, v \in B$ or $u, v \in C$.

Definition 1. Let $A = \{0, 1\}$, $B = \{00, 01, 10\} \subset \mathbb{N}$ and $x \in A$; then, the binary operator $\circ : A \times A \to B$ is defined in Table 1.

The *complement of an element* $x \in A$, represented by $\overline{x}$, is defined by $\overline{0} = 1$ and $\overline{1} = 0$. This concept is extended to elements that are expressed by more than one bit, replacing zeros by ones and vice versa; for example, $\overline{01} = 10$. Applying the elemental complement to the operation in Table 1, its complement is defined.

Table 1. Binary operator $\circ$.

x	y	$x \circ y$
0	0	01
0	1	00
1	0	10
1	1	01

Table 2. Complemented binary operator $\overline{\circ}$.

x	y	$x \,\overline{\circ}\, y$
0	0	10
0	1	11
1	0	01
1	1	10

Table 3. Projection operator $^{\downarrow}$.

u	$u^{\downarrow}$
00	0
01	0
10	1
11	1

Definition 2. Let $A = \{0,1\}$, $C = \{01, 10, 11\}$ and $x \in A$; then, the binary operator $\overline{\circ} : A \times A \to C$ is defined in Table 2.

With the operation given in Definition 1, the method to build robust memories to additive noise in binary images will be developed, while using its complement (Definition 2), memories that have better tolerance to subtractive noise will be obtained.

Definition 3. Let $A = \{0,1\}$, $D = \{00, 01, 10, 11\}$, $x \in A$ and $u \in D$; then, the unary *projection operator* $^{\downarrow} : B \to A$ is defined in Table 3.

The form in which the operations $\circ, \overline{\circ}$ are constructed together with the operator $^{\downarrow}$ allows sufficient properties to be fulfilled to characterize the results concerning the associative memory model proposed for recalling binary images, namely, increasing in relation to $\leq$, distributive with respect to the maximum or minimum, and inverse to each other. The operator $^{\downarrow}$ is applied to the result obtained by the associative memory to ensure that the values are congruent to binary elements. Let $x, y, z \in A$, $^{\downarrow}$ be the projection operator and $\vee, \wedge$ be the maximum and minimum operations, respectively; then, the operator $\circ$ and its complement exhibit the properties of Table 4.

Given $u, v \in B$ in the case of the operation $\circ$ or $u, v \in C$ for its complement, the projection operator $^{\downarrow}$ and the maximum and minimum operations $\vee, \wedge$, respectively, the following relations hold:

$$\begin{aligned} 0 \leq u^{\downarrow} \wedge v^{\downarrow} \leq (u \wedge v)^{\downarrow} \leq u^{\downarrow} \vee v^{\downarrow} \leq 1, \\ 0 \leq u^{\downarrow} \wedge v^{\downarrow} \leq (u \vee v)^{\downarrow} \leq u^{\downarrow} \vee v^{\downarrow} \leq 1. \end{aligned} \tag{3}$$

In addition to the properties shown in Table 4, from Eqn. (2), new matrix operations $\boxed{\vee}$ and $\boxed{\wedge}$ are introduced by replacing the binary operation $\cdot$ by one of the new operations $\circ$ or $\overline{\circ}$. In this article, to refer to a matrix operation using the operator $\circ$ or $\overline{\circ}$, the terminology $\overset{\circ}{\boxed{\vee}}, \overset{\circ}{\boxed{\wedge}}$ or $\overset{\overline{\circ}}{\boxed{\vee}}, \overset{\overline{\circ}}{\boxed{\wedge}}$, respectively, will be used.

4.2. Heteroassociative memory.

Assume that $\mathbf{x} = (x_1, \ldots, x_n)^T$ and $\mathbf{y} = (y_1, \ldots, y_m)^T$ are binary patterns of dimensions n and m, respectively, where ^{T} represents the transposed vector; then, the memories $\mathbf{M}$ and $\mathbf{W}$ that associate the pair of binary input–output vectors $(\mathbf{x}, \mathbf{y})$ are obtained by

$$\mathbf{M} = \mathbf{y} \overset{\circ}{\boxed{\vee}} (\mathbf{x})^T, \quad \mathbf{W} = \mathbf{y} \overset{\overline{\circ}}{\boxed{\vee}} (\mathbf{x})^T. \tag{4}$$

Therefore, the memories are determined by

$$\mathbf{M} = \begin{pmatrix} y_1 \circ x_1 & \cdots & y_1 \circ x_n \\ \vdots & \ddots & \vdots \\ y_m \circ x_1 & \cdots & y_m \circ x_n \end{pmatrix}, \tag{5}$$

$$\mathbf{W} = \begin{pmatrix} y_1 \,\overline{\circ}\, x_1 & \cdots & y_1 \,\overline{\circ}\, x_n \\ \vdots & \ddots & \vdots \\ y_m \,\overline{\circ}\, x_1 & \cdots & y_m \,\overline{\circ}\, x_n \end{pmatrix}.$$

These memories for recalling a binary vector satisfy the conditions

$$\left(\mathbf{M}^{\downarrow} \overset{\circ}{\boxed{\vee}} \mathbf{x}\right)^{\downarrow} = \begin{pmatrix} \bigvee_{j=1}^{n} \left((y_1 \circ x_j)^{\downarrow} \circ x_j\right) \\ \vdots \\ \bigvee_{j=1}^{n} \left((y_m \circ x_j)^{\downarrow} \circ x_j\right) \end{pmatrix}^{\downarrow} = \mathbf{y},$$

$$\left(\mathbf{W}^{\downarrow} \overset{\overline{\circ}}{\boxed{\vee}} \mathbf{x}\right)^{\downarrow} = \begin{pmatrix} \bigvee_{j=1}^{n} \left((y_1 \,\overline{\circ}\, x_j)^{\downarrow} \,\overline{\circ}\, x_j\right) \\ \vdots \\ \bigvee_{j=1}^{n} \left((y_m \,\overline{\circ}\, x_j)^{\downarrow} \,\overline{\circ}\, x_j\right) \end{pmatrix}^{\downarrow} = \mathbf{y}. \tag{6}$$

From Eqn. (6), it is observed that the projection operation is applied to the resulting vector as a threshold that allows obtaining binary vectors in the associative memory output.

4.2.1. Learning and recalling.

Let $A = \{0,1\}$ and $(\mathbf{X}, \mathbf{Y})$ be the matrices with k pairs of binary input–output vectors; then, the learning phase is determined by

Table 4. Properties of complemented operations.

Property	Operator $\circ$	Operator $\bar{\circ}$
1. Incresing	$x \le y \leftrightarrow x \circ z \le y \circ z$ $x \le y \leftrightarrow z \circ y \le z \circ x$	$x \le y \leftrightarrow y \,\bar{\circ}\, z \le x \,\bar{\circ}\, z$ $x \le y \leftrightarrow z \,\bar{\circ}\, x \le z \,\bar{\circ}\, y$
2. Distributive over $\vee$	$(x \vee y) \circ z = (x \circ z) \vee (y \circ z)$	$z \,\bar{\circ}\, (x \vee y) = (z \,\bar{\circ}\, x) \vee (z \,\bar{\circ}\, y)$
3. Distributive over $\wedge$	$(x \wedge y) \circ z = (x \circ z) \wedge (y \circ z)$	$z \,\bar{\circ}\, (x \wedge y) = (z \,\bar{\circ}\, x) \wedge (z \,\bar{\circ}\, y)$
4. Associative	$[(x \circ y)^{\downarrow} \circ z]^{\downarrow} = [x \circ (y \circ z)^{\downarrow}]^{\downarrow}$	$[(x \,\bar{\circ}\, y)^{\downarrow} \,\bar{\circ}\, z]^{\downarrow} = [x \,\bar{\circ}\, (y \,\bar{\circ}\, z)^{\downarrow}]^{\downarrow}$
5. Commutative	$(x \circ y)^{\downarrow} = (y \circ x)^{\downarrow}$	$(x \,\bar{\circ}\, y)^{\downarrow} = (y \,\bar{\circ}\, x)^{\downarrow}$
6. Inverse	$[(x \circ y)^{\downarrow} \circ y]^{\downarrow} = x$ $[(x \circ y)^{\downarrow} \circ x]^{\downarrow} = y$	$[(x \,\bar{\circ}\, y)^{\downarrow} \,\bar{\circ}\, y]^{\downarrow} = x$ $[(x \,\bar{\circ}\, y)^{\downarrow} \,\bar{\circ}\, x]^{\downarrow} = y$
7. Equalities	$z \circ (x \vee y) = (z \circ x) \wedge (z \circ y)$ $z \circ (x \wedge y) = (z \circ x) \vee (z \circ y)$	$(x \vee y) \,\bar{\circ}\, z = (x \,\bar{\circ}\, z) \wedge (y \,\bar{\circ}\, z)$ $(x \wedge y) \,\bar{\circ}\, z = (x \,\bar{\circ}\, z) \vee (y \,\bar{\circ}\, z)$

memories $\mathbf{M_{XY}}$ and $\mathbf{W_{XY}}$ defined in

$$\begin{aligned} \mathbf{M_{XY}} &= \bigvee_{\xi=1}^{k} \left(\mathbf{y}^{\xi} \overset{\circ}{\boxtimes} (\mathbf{x}^{\xi})^{T} \right), \\ \mathbf{W_{XY}} &= \bigvee_{\xi=1}^{k} \left(\mathbf{y}^{\xi} \overset{\bar{\circ}}{\boxtimes} (\mathbf{x}^{\xi})^{T} \right), \end{aligned} \tag{7}$$

and their (i, j)-th elements are given by

$$\begin{aligned} (\mathbf{M_{XY}})_{ij} = m_{ij} &= \bigvee_{\xi=1}^{k} \left(y_i^{\xi} \circ x_j^{\xi} \right), \\ (\mathbf{W_{XY}})_{ij} = w_{ij} &= \bigvee_{\xi=1}^{k} \left(y_i^{\xi} \,\bar{\circ}\, x_j^{\xi} \right), \end{aligned} \tag{8}$$

respectively.

The recalling phase, when presented with an input vector $\mathbf{x}^{\lambda}$, is determined by

$$\left((\mathbf{M_{XY}})^{\downarrow} \overset{\circ}{\boxtimes} \mathbf{x}^{\lambda} \right)^{\downarrow} \quad \text{and} \quad \left((\mathbf{W_{XY}})^{\downarrow} \overset{\bar{\circ}}{\boxtimes} \mathbf{x}^{\lambda} \right)^{\downarrow}, \tag{9}$$

where the i-th elements of the resulting vectors are obtained by

$$\begin{aligned} \left((\mathbf{M_{XY}})^{\downarrow} \overset{\circ}{\boxtimes} \mathbf{x}^{\lambda} \right)_i^{\downarrow} &= \left(\bigvee_{j=1}^{n} \left(\left(m_{ij} \right)^{\downarrow} \circ x_j^{\lambda} \right) \right)^{\downarrow} \\ &= \left(\bigvee_{j=1}^{n} \left(\left(\bigvee_{\xi=1}^{k} \left(y_i^{\xi} \circ x_j^{\xi} \right) \right)^{\downarrow} \circ x_j^{\lambda} \right) \right)^{\downarrow}, \\ \left((\mathbf{W_{XY}})^{\downarrow} \overset{\bar{\circ}}{\boxtimes} \mathbf{x}^{\lambda} \right)_i^{\downarrow} &= \left(\bigvee_{j=1}^{n} \left((w_{ij})^{\downarrow} \,\bar{\circ}\, x_j^{\lambda} \right) \right)^{\downarrow} \\ &= \left(\bigvee_{j=1}^{n} \left(\left(\bigvee_{\xi=1}^{k} \left(y_i^{\xi} \,\bar{\circ}\, x_j^{\xi} \right) \right)^{\downarrow} \,\bar{\circ}\, x_j^{\lambda} \right) \right)^{\downarrow}. \end{aligned} \tag{10}$$

Memories $\mathbf{M_{XY}}$ are robust to the presence of input vectors distorted by additive noise and $\mathbf{W_{XY}}$ memories by subtractive noise. It is observed that the learning and recalling of memories $\mathbf{M_{XY}}$ are determined by the operation $\circ$, while in the case of memories $\mathbf{W_{XY}}$ by their complemented operation $\bar{\circ}$. For this reason, it has been decided to name them *associative memories with complemented operations*.

Definition 4. A matrix $\mathbf{C}$ is said to be a $\overset{\circ}{\boxtimes}$-perfect recalling memory for $(\mathbf{X}, \mathbf{Y})$ if and only if $(\mathbf{C}^{\downarrow} \overset{\circ}{\boxtimes} \mathbf{X})^{\downarrow} = \mathbf{Y}$. The matrix $\mathbf{C}$ is said to be a $\overset{\bar{\circ}}{\boxtimes}$-perfect recall memory for $(\mathbf{X}, \mathbf{Y})$ if and only if $(\mathbf{C}^{\downarrow} \overset{\bar{\circ}}{\boxtimes} \mathbf{X})^{\downarrow} = \mathbf{Y}$.

Definition 4 implies that $(\mathbf{C}^{\downarrow} \overset{\circ}{\boxtimes} \mathbf{x}^{\xi})^{\downarrow} = \mathbf{y}^{\xi}$ and

$$(\mathbf{C}^{\downarrow} \overset{\bar{\circ}}{\boxtimes} \mathbf{x}^{\xi})^{\downarrow} = \mathbf{y}^{\xi} \quad \forall \xi = 1, \dots, k.$$

Theorem 1 shows the conditions that the memories must satisfy to exhibit perfect recovery, while Theorem 2 is the matrix representation of these conditions.

Theorem 1. *$\mathbf{M_{XY}}$ is a $\overset{\circ}{\boxtimes}$-perfect recall memory for $(\mathbf{X}, \mathbf{Y})$ if and only if for each row index $i = 1, \dots, m$ and each $\xi \in \{1, \dots, k\}$ there exist column indices $j_0 \in \{1, \dots, n\}$, which depend on both ξ and i, such that*

$$m_{ij_0} = y_i^{\xi} \circ x_{j_0}^{\xi} \quad \forall \xi = 1, \dots, k. \tag{11}$$

Similarly, $\mathbf{W_{XY}}$ is a $\overset{\bar{\circ}}{\boxtimes}$-perfect recall memory for $(\mathbf{X}, \mathbf{Y})$ if and only if for each row index $i = 1 \dots, m$ and each $\xi \in \{1, \dots, k\}$ there exist column indices $j_0 \in \{1, \dots, n\}$, which depend on both ξ and i, such that

$$w_{ij_0} = y_i^{\xi} \,\bar{\circ}\, x_{j_0}^{\xi} \quad \forall \xi = 1, \dots, k. \tag{12}$$

Theorem 2. *$\mathbf{M_{XY}}$ is a $\overset{\circ}{\boxtimes}$-perfect recall memory for $(\mathbf{X}, \mathbf{Y})$ if and only if, for each $\xi = 1, \dots, k$, each row of matrix $\mathbf{M_{XY}} - \mathbf{y}^{\xi} \overset{\circ}{\boxtimes} (\mathbf{x}^{\xi})^{T}$ contains a zero element. Similarly, $\mathbf{W_{XY}}$ is a $\overset{\bar{\circ}}{\boxtimes}$-perfect recall memory for $(\mathbf{X}, \mathbf{Y})$ if, and only if for each $\xi = 1, \dots, k$, each row of matrix $\mathbf{y}^{\xi} \overset{\bar{\circ}}{\boxtimes} (\mathbf{x}^{\xi})^{T} - \mathbf{W_{XY}}$ contains a zero element.*

From Theorem 2, it follows that $(\mathbf{M}^{\downarrow}_{\mathbf{XY}} \overset{\circ}{\boxslash} \mathbf{x}^{\xi})^{\downarrow} = \mathbf{y}^{\xi} \quad \forall \xi = 1, \dots, k$ if and only if, for each index ξ and each row index i, there exists a column index j (which depends on ξ and i) such that $m_{ij} = \left(\mathbf{y}^{\xi} \overset{\circ}{\boxslash} (\mathbf{x}^{\xi})^T\right)_{ij}$. Similarly, that is expressed for $\mathbf{W_{XY}}$.

Corollary 1. *$((\mathbf{M_{XY}})^{\downarrow} \overset{\circ}{\boxslash} \mathbf{X})^{\downarrow} = \mathbf{Y}$ if and only if, for each row index $i = 1, \dots, m$ and each $\lambda \in \{1, \dots, k\}$, there exists a column index $j \in \{1, \dots, n\}$ (as a function of i and λ) such that*

$$x_j^{\lambda} = \Big(\Big(\bigvee_{\xi=1}^{k} (y_i^{\xi} \circ x_j^{\xi}) \Big)^{\downarrow} \circ y_i^{\lambda} \Big)^{\downarrow}. \tag{13}$$

Similarly, $((\mathbf{W_{XY}})^{\downarrow} \overset{\bar{\circ}}{\boxslash} \mathbf{X})^{\downarrow} = \mathbf{Y}$ if, and only if for each row index $i = 1, \dots, m$ and each $\lambda \in \{1, \dots, k\}$, there exists a column index $j \in \{1, \dots, n\}$ (as a function of i and λ) such that

$$x_j^{\lambda} = \Big(\Big(\bigvee_{\xi=1}^{k} (y_i^{\xi} \mathbin{\bar{\circ}} x_j^{\xi}) \Big)^{\downarrow} \mathbin{\bar{\circ}} y_i^{\lambda} \Big)^{\downarrow}. \tag{14}$$

The proposed associative memory model is able to provide perfect recall from noise-distorted versions. Theorem 3 shows the conditions under which memories $\mathbf{M_{XY}}$ tolerate additive noise and memories $\mathbf{W_{XY}}$ subtractive noise.

Theorem 3. *Let $\tilde{\mathbf{x}}^{\lambda}$ be a distorted version with additive noise of $\mathbf{x}^{\lambda}$; then, $((\mathbf{M_{XY}})^{\downarrow} \overset{\circ}{\boxslash} \tilde{\mathbf{x}}^{\lambda})^{\downarrow} = \mathbf{y}^{\lambda}$ if and only if for each row index $i = 1, \dots, m$ there exists a column index $j_0 \in \{1, \dots, n\}$ which depends on both λ and i, such that*

$$m_{ij_0} = y_i^{\lambda} \circ \tilde{x}_{j_0}^{\lambda}. \tag{15}$$

Similarly, let $\tilde{\mathbf{x}}^{\lambda}$ be a distorted version with subtractive noise of $\mathbf{x}^{\lambda}$; then, $((\mathbf{W_{XY}})^{\downarrow} \overset{\bar{\circ}}{\boxslash} \tilde{\mathbf{x}}^{\lambda})^{\downarrow} = \mathbf{y}^{\lambda}$ if and only if for each row index $i = 1, \dots, m$ there exists a column index $j_0 \in \{1, \dots, n\}$ which depends on both λ and i, such that

$$m_{ij_0} = y_i^{\lambda} \mathbin{\bar{\circ}} \tilde{x}_{j_0}^{\lambda}. \tag{16}$$

4.3. Autoassociative memory. A reference of autoassociative memories is the Hopfield model, whose storage capacity in a network of n neurons is no greater than $n/(4 \log n)$. The storage capacity is infinite in the proposed autoassociative model, i.e., there are no restrictions on the number of input–output patterns pairs to be stored, and in the absence of noise the model provides perfect recall for all patterns stored in memory. This is proven in Theorem 4.

Theorem 4. *We have $\left((\mathbf{M_{XX}})^{\downarrow} \overset{\circ}{\boxslash} \mathbf{X}\right)^{\downarrow} = \mathbf{X}$ and $\left((\mathbf{W_{XX}})^{\downarrow} \overset{\bar{\circ}}{\boxslash} \mathbf{X}\right)^{\downarrow} = \mathbf{X}$.*

Another characteristic of the proposed model is shown in Theorem 5, in which it is observed that the memory does not present convergence problems in the recalling phase; therefore, the processing time is reduced because it is not necessary to iterate the output operations until a stability point is reached.

Theorem 5. *If $\left((\mathbf{M_{XX}})^{\downarrow} \overset{\circ}{\boxslash} \mathbf{z}\right)^{\downarrow} = \mathbf{u}$, then we have $\left((\mathbf{M_{XX}})^{\downarrow} \overset{\circ}{\boxslash} \mathbf{u}\right)^{\downarrow} = \mathbf{u}$. Similarly, if $\left((\mathbf{W_{XX}})^{\downarrow} \overset{\bar{\circ}}{\boxslash} \mathbf{z}\right)^{\downarrow} = \mathbf{u}$, then $\left((\mathbf{W_{XX}})^{\downarrow} \overset{\bar{\circ}}{\boxslash} \mathbf{u}\right)^{\downarrow} = \mathbf{u}$.*

5. Experiments

In this section the performance report of the proposed memories is presented in comparison with other models.

5.1. Numerical results. Numerical examples show how the proposed memories perfectly recall both the training set and a distorted version.

Example 1. Let

$$\begin{aligned}
\mathbf{x}^1 &= \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}, \quad \mathbf{y}^1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \\
\mathbf{x}^2 &= \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad \mathbf{y}^2 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}, \\
\mathbf{x}^3 &= \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad \mathbf{y}^3 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \\
\mathbf{x}^4 &= \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{y}^4 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \\
\mathbf{x}^5 &= \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{y}^5 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}.
\end{aligned} \tag{17}$$

Then $\mathbf{W_{XY}}$ memories $\mathbf{M_{XY}}$ are given by Eqns. (18) and (19), respectively.

It can be verified that $\mathbf{M_{XY}}$ and $\mathbf{W_{XY}}$ exhibit perfect recall for $\mathbf{X}$. This is shown in Eqns. (20) and (21).

$$\mathbf{M_{XY}} = \bigvee_{\xi=1}^{5}\left(\mathbf{y}^{\xi}\,\overset{\circ}{\boxtimes}\,(\mathbf{x}^{\xi})^{T}\right) = \begin{pmatrix} 01 & 01 & 00 & 00 & 01 \\ 10 & 10 & 01 & 01 & 10 \\ 10 & 10 & 01 & 01 & 10 \\ 01 & 01 & 00 & 00 & 01 \end{pmatrix} \vee \begin{pmatrix} 01 & 01 & 10 & 10 & 01 \\ 00 & 00 & 01 & 01 & 00 \\ 01 & 01 & 10 & 10 & 01 \\ 01 & 01 & 10 & 10 & 01 \end{pmatrix}$$
$$\vee \begin{pmatrix} 01 & 10 & 10 & 10 & 01 \\ 00 & 01 & 01 & 01 & 00 \\ 01 & 10 & 10 & 10 & 01 \\ 00 & 01 & 01 & 01 & 00 \end{pmatrix} \vee \begin{pmatrix} 01 & 00 & 01 & 00 & 01 \\ 10 & 01 & 10 & 01 & 10 \\ 10 & 01 & 10 & 01 & 10 \\ 10 & 01 & 10 & 01 & 10 \end{pmatrix} \vee \begin{pmatrix} 01 & 01 & 00 & 00 & 01 \\ 10 & 10 & 01 & 01 & 10 \\ 10 & 10 & 01 & 01 & 10 \\ 01 & 01 & 00 & 00 & 01 \end{pmatrix} \tag{18}$$
$$= \begin{pmatrix} 01 & 10 & 10 & 10 & 01 \\ 10 & 10 & 10 & 01 & 10 \\ 10 & 10 & 10 & 10 & 10 \\ 10 & 01 & 10 & 10 & 10 \end{pmatrix},$$

$$\mathbf{W_{XY}} = \bigvee_{\xi=1}^{5}\left(\mathbf{y}^{\xi}\,\overset{\overline{\circ}}{\boxtimes}\,(\mathbf{x}^{\xi})^{T}\right) = \begin{pmatrix} 10 & 10 & 11 & 11 & 10 \\ 01 & 01 & 10 & 10 & 01 \\ 01 & 01 & 10 & 10 & 01 \\ 10 & 10 & 11 & 11 & 10 \end{pmatrix} \vee \begin{pmatrix} 10 & 10 & 01 & 01 & 10 \\ 11 & 11 & 10 & 10 & 11 \\ 10 & 10 & 01 & 01 & 10 \\ 10 & 10 & 01 & 01 & 10 \end{pmatrix}$$
$$\vee \begin{pmatrix} 10 & 10 & 01 & 01 & 10 \\ 11 & 10 & 10 & 10 & 11 \\ 10 & 01 & 01 & 01 & 10 \\ 11 & 10 & 10 & 10 & 11 \end{pmatrix} \vee \begin{pmatrix} 10 & 11 & 10 & 11 & 10 \\ 01 & 10 & 01 & 10 & 01 \\ 01 & 10 & 01 & 10 & 01 \\ 01 & 10 & 01 & 10 & 10 \end{pmatrix} \vee \begin{pmatrix} 10 & 10 & 11 & 11 & 10 \\ 01 & 01 & 10 & 10 & 01 \\ 01 & 01 & 10 & 10 & 01 \\ 10 & 10 & 11 & 11 & 10 \end{pmatrix} \tag{19}$$
$$= \begin{pmatrix} 10 & 11 & 11 & 11 & 10 \\ 11 & 11 & 10 & 10 & 11 \\ 10 & 10 & 10 & 10 & 10 \\ 11 & 10 & 11 & 11 & 11 \end{pmatrix}.$$

The memories satisfy the conditions of Theorem 1 for perfect recall. For example, for $\xi = 1$,

$$\begin{aligned} m_{11} &= y_1^1 \circ x_1^1 = 1 \circ 1 = 01, \\ w_{15} &= y_1^1 \,\overline{\circ}\, x_5^1 = 1 \,\overline{\circ}\, 1 = 10, \\ m_{23} &= y_2^1 \circ x_3^1 = 1 \circ 0 = 10, \\ w_{24} &= y_2^1 \,\overline{\circ}\, x_4^1 = 1 \,\overline{\circ}\, 1 = 10, \\ m_{33} &= y_3^1 \circ x_3^1 = 1 \circ 0 = 10, \\ w_{31} &= y_3^1 \,\overline{\circ}\, x_1^1 = 1 \,\overline{\circ}\, 1 = 10, \\ m_{42} &= y_4^1 \circ x_2^1 = 0 \circ 0 = 01, \\ w_{45} &= y_4^1 \,\overline{\circ}\, x_5^1 = 0 \,\overline{\circ}\, 1 = 11. \end{aligned} \tag{22}$$

♦

Example 2. Suppose that $\widetilde{\mathbf{X}}_A$ and $\widetilde{\mathbf{X}}_S$ represent a distorted version of $\mathbf{X}$ with additive and subtractive noise, respectively:

$$\widetilde{\mathbf{X}}_A = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix},$$

$$\widetilde{\mathbf{X}}_S = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}. \tag{23}$$

Then memories $\mathbf{M_{XY}}$ and $\mathbf{W_{XY}}$ are both capable of providing perfect recall when presented with inputs $\widetilde{\mathbf{X}}_A$ and $\widetilde{\mathbf{X}}_S$, i.e., they are robust to additive and subtractive noise, respectively; cf. (24) and (25). ♦

5.2. Simulation results. Experiments with binary images were performed to test noise tolerance of associative memory models. This set of 7×7 pixels corresponds to the uppercase letters of the alphabet (Fig. 1), where a pixel in black represents the value of 1 and that in white represents the value of 0. Simulations were implemented using the Python programming language on a computer with 8 GB of installed RAM memory and an Intel® Core™ i5-7200U processor with a 64-bit Kaby Lake architecture and a base frequency speed of 2.5 GHz. The source code of the simulation is published in the public repository of GitHub (Gamino-Carranza, 2022).

5.2.1. Experiment 1: Pairs of letters (I, T), (F, E) and (C, G). Consider heteroassociative mode with pairs of

$$
\begin{aligned}
\left((\mathbf{M_{XY}})^{\downarrow}\overset{\circ}{\boxbslash}\mathbf{X}\right)^{\downarrow} &= \left(\begin{pmatrix} 01 & 10 & 10 & 10 & 01 \\ 10 & 10 & 10 & 01 & 10 \\ 10 & 10 & 10 & 10 & 10 \\ 10 & 01 & 10 & 10 & 10 \end{pmatrix}^{\downarrow} \overset{\circ}{\boxbslash} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}\right)^{\downarrow} \\
&= \left(\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \overset{\circ}{\boxbslash} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}\right)^{\downarrow} \\
&= \begin{pmatrix} 01 & 01 & 01 & 10 & 10 \\ 01 & 10 & 10 & 01 & 01 \\ 01 & 01 & 01 & 01 & 01 \\ 10 & 01 & 10 & 01 & 10 \end{pmatrix}^{\downarrow} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} = \mathbf{Y},
\end{aligned} \tag{20}
$$

$$
\begin{aligned}
\left((\mathbf{W_{XY}})^{\downarrow}\overset{\bar{\circ}}{\boxbslash}\mathbf{X}\right)^{\downarrow} &= \left(\begin{pmatrix} 10 & 11 & 11 & 11 & 10 \\ 11 & 11 & 10 & 10 & 11 \\ 10 & 10 & 10 & 10 & 10 \\ 11 & 10 & 11 & 11 & 11 \end{pmatrix}^{\downarrow} \overset{\bar{\circ}}{\boxbslash} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}\right)^{\downarrow} \\
&= \left(\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \end{pmatrix} \overset{\bar{\circ}}{\boxbslash} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}\right)^{\downarrow} \\
&= \begin{pmatrix} 11 & 11 & 11 & 10 & 10 \\ 11 & 10 & 10 & 11 & 11 \\ 11 & 11 & 11 & 11 & 11 \\ 10 & 11 & 10 & 11 & 10 \end{pmatrix}^{\downarrow} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} = \mathbf{Y}.
\end{aligned} \tag{21}
$$

letters (I, T), (F, E) and (C, G) from Fig. 1. Note that for each pair the first one is a subset of the other. Therefore, the second one can be treated as an additively noised first one or the first one as a subtractively noised second one.

Figure 2 shows the recall for the training set and distorted patterns with additive and subtractive noise (in both cases, 10 and 17 noisy pixels). Although in each pair the first one is a subset of the other, we observe that $\mathbf{M_{XY}}$ is robust to additive noise but sensitive to subtractive noise, while $\mathbf{W_{XY}}$ is robust to subtractive noise and additive noise destroys the recall patterns.

5.2.2. Experiment 2: Uppercase letters of the alphabet. Three simulations were carried out with additive, subtractive and mixed noise at the input (Fig. 3). In simulations, autoassociative operating mode of the memories was used, and the results presented were calculated from the average of 500 trials. The proposed memories in simulations were compared with the following models:

- based on lattice algebra: morphological associative memories (MAMs), logarithmic and exponential morphological associative memory (LEMAM), new fuzzy morphological associative memory

Algorithm 1. Learning phase.

Require: $\{\mathbf{p}^{\xi} | \xi = 1, \ldots, k\}$ {training set}
Ensure: $\mathbf{M_{XX}} \in B, \mathbf{W_{XX}} \in C$ {associative memory}
$\mathbf{M_{XX}} := 00$
$\mathbf{W_{XX}} := 01$ {memories initialization}
for $\xi = 1$ to k **do**
$x^{\xi}_{7(i-1)+j} := \begin{cases} 1 \text{ if } \mathbf{p}^{\xi}(i,j) \text{ is black} \\ 0 \text{ if } \mathbf{p}^{\xi}(i,j) \text{ is white} \end{cases}$ {image to vector}
$\mathbf{M_{XX}} := \mathbf{M_{XX}} \vee \left(\mathbf{x}^{\xi}\overset{\circ}{\boxbslash}(\mathbf{x}^{\xi})^{T}\right)$
$\mathbf{W_{XX}} := \mathbf{W_{XX}} \vee \left(\mathbf{x}^{\xi}\overset{\bar{\circ}}{\boxbslash}(\mathbf{x}^{\xi})^{T}\right)$ {adjust the weights}
end for
return $\mathbf{M_{XX}}, \mathbf{W_{XX}}$ {associative memories}

$$
\begin{aligned}
\left((\mathbf{M}_{\mathbf{XY}})^{\downarrow}\,\overset{\circ}{\boxtimes}\,\widetilde{\mathbf{X}}_A\right)^{\downarrow} &= \left(\begin{pmatrix} 01 & 10 & 10 & 10 & 01 \\ 10 & 10 & 10 & 01 & 10 \\ 10 & 10 & 10 & 10 & 10 \\ 10 & 01 & 10 & 10 & 10 \end{pmatrix}^{\downarrow} \overset{\circ}{\boxtimes} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}\right)^{\downarrow} \\
&= \left(\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \overset{\circ}{\boxtimes} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}\right)^{\downarrow} \\
&= \begin{pmatrix} 01 & 01 & 01 & 10 & 10 \\ 01 & 10 & 10 & 01 & 01 \\ 01 & 01 & 01 & 01 & 01 \\ 10 & 01 & 10 & 01 & 10 \end{pmatrix}^{\downarrow} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} = \mathbf{Y},
\end{aligned}
\tag{24}
$$

$$
\begin{aligned}
\left((\mathbf{W}_{\mathbf{XY}})^{\downarrow}\,\overset{\bar{\circ}}{\boxtimes}\,\widetilde{\mathbf{X}}_S\right)^{\downarrow} &= \left(\begin{pmatrix} 10 & 11 & 11 & 11 & 10 \\ 11 & 11 & 10 & 10 & 11 \\ 10 & 10 & 10 & 10 & 10 \\ 11 & 10 & 11 & 11 & 11 \end{pmatrix}^{\downarrow} \overset{\bar{\circ}}{\boxtimes} \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}\right)^{\downarrow} \\
&= \left(\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \end{pmatrix} \overset{\bar{\circ}}{\boxtimes} \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}\right)^{\downarrow} \\
&= \begin{pmatrix} 11 & 11 & 11 & 10 & 10 \\ 11 & 10 & 10 & 11 & 11 \\ 11 & 11 & 11 & 11 & 11 \\ 10 & 11 & 10 & 11 & 10 \end{pmatrix}^{\downarrow} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} = \mathbf{Y}.
\end{aligned}
\tag{25}
$$

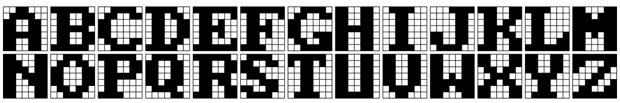


Fig. 1. Training set $\{\mathbf{p}^{\xi}|\xi = 1, \ldots, 26\}$. Numbering goes from left to right and from top to bottom.

(NFMAM), implicative fuzzy associative memory (IFAM), a new method of morphological associative memory (NMMAM) and no rounding reverse fuzzy morphological associative memory (NR^{2}FMAM);

- based on fuzzy logic: fuzzy relational memory (FRM), an improved model of optical fuzzy associative memory (IMOFAM), a max-min encoding learning algorithm for fuzzy max-multiplication (max-min ELAFMM) and a max-min fuzzy neural network with threshold (max-min FNNT);
- based on chaotic: a globally coupled map using the symmetric map (S-GCM), the new parameter control method for S-GCM (new S-GCM), a globally coupled map using the cubic logistic map (CL-GCM), globally coupled map using the sine map (SI-GCM), a parametrically coupled sine map network 1 (PCSMN 1) and a parametrically coupled sine map network 2 (PCSMN 2);
- based on the Hopfield network: the Hopfield network (Hopfield) and Hopfield with hysteresis (Hopfield hysteresis).

In the case of the proposed memories and those based on lattice algebra, max type memories $\mathbf{M}$ and min type memories $\mathbf{W}$ were used for additive and subtractive noise, respectively. In the third simulation, the comparison was made with the kernel method for morphological associative memory.

The code used for programming the learning and recalling phases of the proposed associative memories is shown in Algorithms 1 and 2. The first one describes the procedure to obtain memories $\mathbf{M}$ and $\mathbf{W}$, while the other details the calculation of memory performance

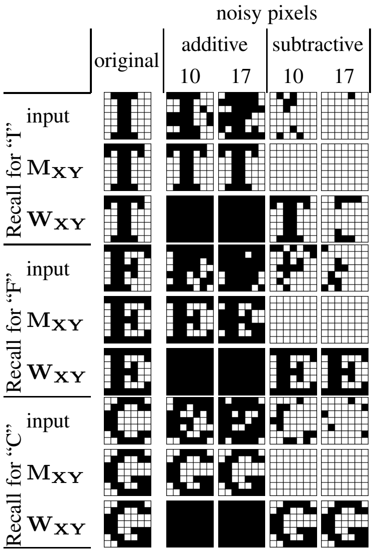
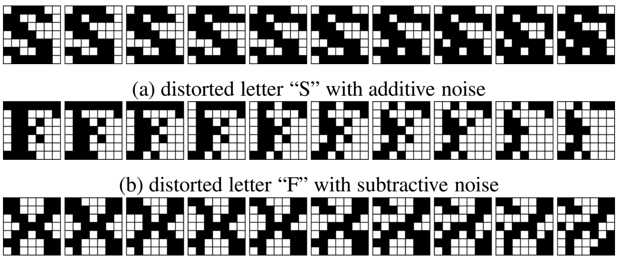


Fig. 2. Recall of the proposed memories for pairs of letters: (I, T), (F, E) and (C, G).



(a) distorted letter "S" with additive noise

(b) distorted letter "F" with subtractive noise

(c) distorted letter "X" with mixed noise

Fig. 3. Examples of different versions of letters distorted by noise. From left to right in each subfigure the number of noisy pixels varies from 1 to 10.

in recalling binary input patterns, which are distorted with additive and subtractive noise. By modifying the operations and rules used to determine memories and the recall of output vectors in the two algorithms, the other associative memories models used in the simulations can be determined.

Simulation with additive noise in Fig. 4 shows that the proposed associative memories, morphological and their variant exhibit the same performance. As the number of distorted pixels in the binary image increases, the memories exhibit higher tolerance than other associative memory models. In the absence of noise, in addition to these memories, fuzzy memories also present perfect recall of the training set patterns, i.e., for the 26 binary images, there is a similarity $\gamma = 1$. Table 5 shows similar results with the number of training patterns perfectly stored by associative memories (perfect recall ratio), i.e., the proposed associative memories, morphological and their variant, continue exhibiting better performance compared to the other associative memories.

Algorithm 2. Recalling phase.

Require: $\{\mathbf{p}^\xi | \xi = 1, \ldots, k\}$ {training set}
$\{\mathbf{x}^\xi \in A | \xi = 1, \ldots, k\}$ {training vector set}
σ {number of noisy pixel}
$varNoise$ {type of noise}
$f_A(r, \mathbf{x})$ where r is number of pixels to be altered and $\mathbf{x}$ is a binary vector {additive noise function}
$f_S(r, \mathbf{x})$ where r is number of pixels to be altered and $\mathbf{x}$ is a binary vector {subtractive noise function}
Ensure: $\mathbf{q}_r^\xi(i, j), d(\xi, r)$ {recalled pattern data}
$\mathbf{R} := \{0, 0\}$ {results array initialization}
for $r = 1$ to σ **do**
  for $\xi = 1$ to k **do**
    if $varNoise == additive$ **then**
      $\tilde{\mathbf{x}}^\xi := f_A(r, \mathbf{x}^\xi)$ {additive noise vector}
      $\mathbf{y} := \left((\mathbf{M_{XX}})^{\downarrow} \overset{\circ}{\boxtimes} \tilde{\mathbf{x}}^\xi\right)^{\downarrow}$ {recall vector}
    else if $varNoise == sutractive$ **then**
      $\tilde{\mathbf{x}}^\xi := f_S(r, \mathbf{x}^\xi)$ {subtractive noise vector}
      $\mathbf{y} := \left((\mathbf{W_{XX}})^{\downarrow} \overset{\bar{\circ}}{\boxtimes} \tilde{\mathbf{x}}^\xi\right)^{\downarrow}$ {recall vector}
    end if
    $d(\xi, r) := \gamma\left(\mathbf{x}^\xi, \mathbf{y}\right)$ {similarity}
    $\mathbf{q}_r^\xi(i, j) := \begin{cases} \text{black if } y_{7(i-1)+j} = 1 \\ \text{white if } y_{7(i-1)+j} = 0 \end{cases}$ {vector to image}
    $\mathbf{R}_r^\xi := \{\mathbf{q}_r^\xi(i, j), d(\xi, r)\}$ {save results}
  end for
end for
return $\mathbf{R}$ {recalled pattern data}

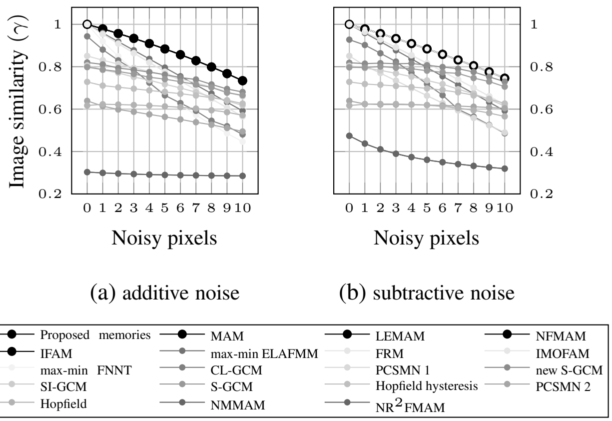
For simulation with subtractive noise, similar performance and noise tolerance comments are provided for the proposed associative memories, morphological and their variant; however, according to Fig. 4 and Table 6, the max-min fuzzy neural network with a threshold also exhibits similar results as these memories.

Although the outcomes suggest that the proposed associative memories, morphological and their variant, exhibit superior performance with respect to most memories, it is pertinent to consider the following aspects:

- According to Theorem 5 the proposed associative memories converges in a single step, while the Hopfield network, chaotic memories based on the S-GCM model and the parametrically coupled sine map networks require several iterations to converge to a local minimum; therefore, the processing time required by the proposed associative memories is smaller with respect to the memories mentioned.
- Operations used for logarithmic and exponential morphological associatives require more processing time than the proposed associative memories, and the same occurs for the new fuzzy morphological

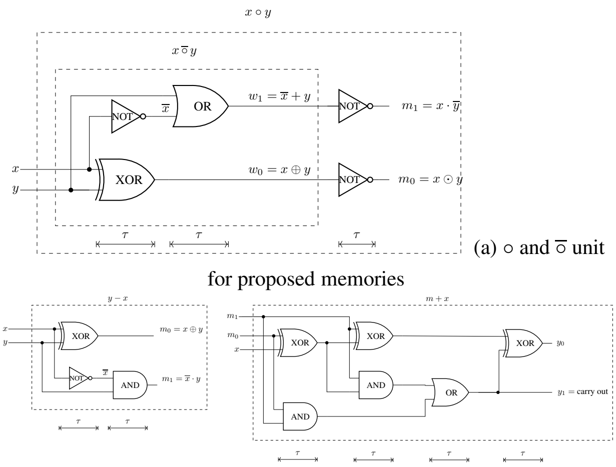
Table 5. Perfect recall rate [%] of associative memories with pixels inputs training patterns distorted by additive noise. The number of perfectly recalled patterns divided by that of stored training patterns.

Associative memory	Additive noisy pixels										
	0	1	2	3	4	5	6	7	8	9	10
Proposed memories, MAM, LEMAM, NFMAM, IFAM	100	47.6	26.1	15.5	9.5	6.1	4	2.6	1.8	1.2	0.7
max-min FNNT, FRM, IMOFAM, max-min ELAFMM	100	0	0	0	0	0	0	0	0	0	0
CL-GCM	90.5	0	0	0	0	0	0	0	0	0	0
PCSMN 1	23.1	14.1	8.8	6.1	4.5	3.2	2.4	1.5	1.2	0.7	0.4
Hopfield hysteresis	7.7	4.7	4.7	4.6	4.5	4.3	4.1	4.1	3.9	3.7	3.5
new S-GCM	7.7	7	6.5	6.6	6.5	6.2	5.8	5.4	4.4	3.7	3
S-GCM	7.5	7	6.3	6.1	5.9	5.7	5.4	5.1	4.7	4.2	3.5
SI-GCM	5.6	3.4	3	2.1	1.9	1.5	1.4	1.2	1	1	0.8
PCSMN 2	3.8	2.1	1.2	0.9	0.6	0.3	0.2	0	0.1	0.1	0
Hopfield, NMMAM, NR^2FMAM	0	0	0	0	0	0	0	0	0	0	0



(a) additive noise (b) subtractive noise

Fig. 4. Performance of associative memories using gamma binary distance averaging when pixels training input patterns are distorted by noise.



(a) $\circ$ and $\bar{\circ}$ unit for proposed memories

(b) $-$ and $+$ unit for morphological memories

Fig. 5. Digital logic gates implementation of associative memories operations (τ represents propagation delay time in the logic gate).

associative memory that uses the division and multiplication operations.

- Figure 5 shows that the implementation of morphological associative memories operations requires a higher number of digital logic gates than the proposed memories. In addition, morphological memories implement two different circuits, one for learning and another for recalling, while the proposed memories implement the same digital logic circuit in both phases. Finally, if we suppose the delay time through the digital gate is τ, then the measure of how long it takes for the digital logic circuit to shift to the final state is less in the proposed memories than in morphological memories. A similar remark is applicable to implicative fuzzy associative memories.

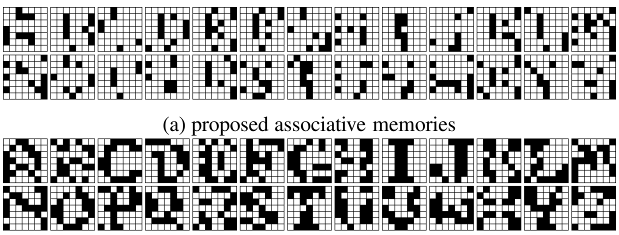
In the last simulation, input patterns were distorted with mixed noise, and the *kernel* method proposed by Ritter *et al.* (1998) was used, which consists of defining a matrix $\mathbf{Z} = (\mathbf{z}^1, \ldots, \mathbf{z}^k)$ such that

$$\left(\left(\mathbf{W}_{\mathbf{ZX}}\right)^{\downarrow} \bar{\overset{\circ}{\boxtimes}} \left(\left(\mathbf{M}_{\mathbf{ZZ}}\right)^{\downarrow} \overset{\circ}{\boxtimes} \mathbf{X}\right)^{\downarrow}\right)^{\downarrow} = \mathbf{X},$$

and it follows that $\mathbf{z}^\lambda \ll \mathbf{x}^\xi$, $\mathbf{z}^\lambda \wedge \mathbf{x}^\xi = \mathbf{0}\ \forall \lambda \neq \xi$, where $\ll$ denotes that the vector $\mathbf{z}^\lambda$ contains more entries with zero values than $\mathbf{x}^\xi$ and $\mathbf{0}$ is the vector whose all entries are zero (Sussner, 2000). To determine kernel vectors of proposed associative memories, an algorithm developed by Hattori *et al.* (2002) was adapted. The kernel vectors for morphological associative memories were determined by the conventional trial-and-error method. Figure 6 shows the kernel vectors used for the simulation. It is observed that vectors $\mathbf{z}$ for morphological associative memories do not contain a majority of inputs with zero values; however, this set of kernel vectors in the absence

Table 6. Perfect recall rate [%] of associative memories with pixels inputs training patterns distorted by subtractive noise. The number of perfectly recalled patterns divided by that of stored training patterns.

Associative memory	Subtractive noisy pixels										
	0	1	2	3	4	5	6	7	8	9	10
Proposed memories, MAM, LEMAM, NFMAM, IFAM, max-min FNNT	100	47.2	22.2	10.8	5.5	2.8	1.5	0.8	0.4	0.2	0.1
FRM, IMOFAM	100	12.3	8.8	5.7	3.2	1.6	0.7	0.2	0.1	0	0
max-min ELAFMM	100	0	0	0	0	0	0	0	0	0	0
CL-GCM	88.1	0	0	0	0	0	0	0	0	0	0
PCSMN 1	23.1	0	0	0	0	0	0	0	0	0	0
Hopfield hysteresis	7.7	5.4	4.2	3.5	3.3	2.9	2.7	2.5	2.4	2.1	1.9
new S-GCM	7.5	6.9	7	6.7	6.4	6	5.1	4.6	4.2	3.5	3.1
S-GCM	7.5	6.7	6.3	5.8	5.4	5	4.4	3.9	3.5	2.8	2.3
SI-GCM	5.3	3.7	2.6	1.9	1.6	1.2	0.9	0.8	0.8	0.7	0.5
PCSMN 2	3.8	2.9	2.3	2.4	2.6	2.6	2.5	2.6	2.3	2.3	2
Hopfield, NMMAM, NR^{2}FMAM	0	0	0	0	0	0	0	0	0	0	0



(a) proposed associative memories

(b) morphological associative memories

Fig. 6. Kernel patterns $\{\mathbf{z}^{\xi} | \xi = 1, \dots, 26\}$. In each subfigure the numbering starts from left to right and from top to bottom.

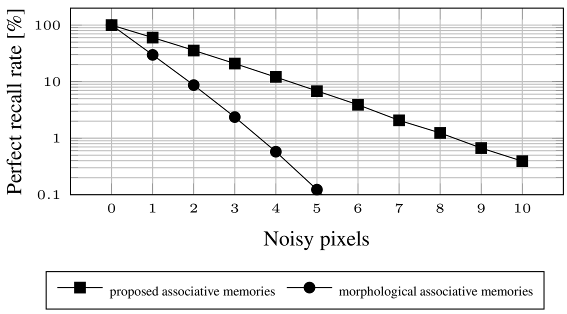


Fig. 7. Number of perfectly recalled patterns divided by that of stored training patterns when pixel input training patterns are distorted by mixed noise.

of noise in the input guarantees a perfect recall from the training set.

The perfect recall rate indicator (Fig. 7) shows that the proposed associative memories exhibit better performance than morphological memories using their associated kernel vectors from Fig. 6. Because the performance curve of morphological associative memories decreases very fast, the representation on a graph is semilogarithmic.

The numerical and computer simulation results indicate that the proposed associative memories are a competent model in efficient recall of binary images and, in general, in applications of artificial intelligence; therefore, these memories are an alternative for obtaining an associative memory model.

6. Conclusions

In this paper, a new associative memory model derived from lattice algebra-based memories was proposed. The new learning and recalling methods are based on the design of a binary operation $\circ$, its complement $\overline{\circ}$ and a unary threshold operator $^{\downarrow}$ named projection. The proposed model has the following characteristics:

- an associative memory $\mathbf{M}$, whose learning method is based on the superposition of maxima of the partial results of the operation $\circ$ between the input and output vectors, while in the recalling phase, the memory uses the superposition of maxima of the results of the same operation $\circ$ between $\mathbf{M}$ and the input vector presented to memory;
- an associative memory $\mathbf{W}$, which is obtained by changing the operation $\circ$ by its complement;
- $\mathbf{M}$ and $\mathbf{W}$ implement the same digital circuit for learning and recalling phases;
- the convergence of the associative memories is performed in a single step;
- memories $\mathbf{M}$ and $\mathbf{W}$ operate in heteroassociative and autoassociative modes; in the last mode, the memory storage capacity is unlimited, and there is perfect recall for the training set;
- memory $\mathbf{M}$ exhibits better tolerance to the presence of significant amounts of additive noise, while

memory $\mathbf{W}$ exhibits better tolerance to subtractive noise; in the case of mixed noise, the kernel method was used to combine memories $\mathbf{M}$ and $\mathbf{W}$.

In future works, we intend to implement these associative memories in specialized hardware embedded systems such as FPGAs and to explore some structural decomposition method that reduces the chip area occupied by the logic circuits that define associative memory (Barkalov *et al.*, 2022). Another important feature is the application of new probabilistic approaches that improve the performance of the proposed memory in the presence of distorted inputs with mixed noise (Salgado-Ramírez *et al.*, 2022). Also, we intend to investigate grayscale image decomposition techniques to apply the proposed associative memories on each binary layer resulting from the decomposition.

References

Barkalov, A., Titarenko, L. and Mazurkiewicz, M. (2022). Improving the LUT count for Mealy FSMs with transformation of output collections, *International Journal of Applied Mathematics and Computer Science* **32**(3): 479–494, DOI: 10.34768/amcs-2022-0035.

Chung, F.-L. and Lee, T. (1994). Towards a high capacity fuzzy associative memory model, *Proceedings of the 1994 IEEE International Conference on Neural Networks (ICNN'94), Florida, USA*, Vol. 3, pp. 1595–1599, DOI: 10.1109/ICNN.1994.374394.

Feng, N., Cao, X., Li, S., Ao, L. and Wang, S. (2009). A new method of morphological associative memories, *Emerging Intelligent Computing Technology and Applications, with Aspects of Artificial Intelligence, ICIC 2009, Ulsan, South Korea,* pp. 407–416, DOI: 10.1007/978-3-642-04020-7_43.

Feng, N.-Q., Tian, Y., Wang, X.-F., Song, L.-M., Fan, H.-J. and Shuang-Xi, W. (2015). Logarithmic and exponential morphological associative memories, *Journal of Software* **26**(7): 1662–1674, DOI: 10.13328/j.cnki.jos.004620.

Feng, N. and Yao, Y. (2016). No rounding reverse fuzzy morphological associative memories, *Neural Network World* **26**(6): 571–587, DOI: 10.14311/NNW.2016.26.033.

Gamino-Carranza, A. (2022). Binary associative memories, `https://github.com/arturogam/Binary-Associative-Memories`, (programming code).

Hassoun, M.H. (1993). *Associative Neural Memories: Theory and Implementation*, Oxford University Press, Inc., New York.

Hattori, M., Fukui, A. and Ito, H. (2002). A fast method of constructing kernel patterns for morphological associative memory, *9th International Conference on Neural Information Processing, ICONI 02, Singapore*, pp. 1058–1063, DOI: 10.1109/ICONIP.2002.1198222.

Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities, *Proceedings of the National Academy of Sciences of the United States of America* **79**(8): 2554–2558, DOI: 10.1073/pnas.79.8.2554.

Ishi, S., Fukumizu, K. and Watanabe, S. (1996). A network of chaotic elements for information processing, *Neural Networks* **9**(1): 25–40, DOI: 10.1016/0893-6080(95)00100-X.

Kosko, B. (1991). Fuzzy associative memories, *Proceedings of the 2nd Joint Technology Workshop on Neural Networks and Fuzzy Logic, Houston, USA*, pp. 3–58.

Lee, G. and Farhat, N.H. (2001). Parametrically coupled sine map networks, *International Journal of Bifurcation and Chaos* **11**(07): 1815–1834, DOI: 10.1142/S0218127401003048.

Liu, P. (1999). The fuzzy associative memory of max-min fuzzy neural network with threshold, *Fuzzy Sets and Systems* **107**(2): 147–157, DOI: 10.1016/S0165-0114(97)00352-7.

McEliece, R., Posner, E., Rodemich, E. and Venkatesh, S. (1987). The capacity of the Hopfield associative memory, *IEEE Transactions on Information Theory* **33**(4): 461–482, DOI: 10.1109/TIT.1987.1057328.

Mustafa, A.A. (2018). Probabilistic binary similarity distance for quick binary image matching, *IET Image Processing* **12**(10): 1844–1856, DOI: 10.1049/iet-ipr.2017.1333.

Rani, S.S., Rao, N. and Vatsal, S. (2018). Review on neural networks associative memory models, *International Journal of Pure and Applied Mathematics* **120**(6): 3143–3154.

Ritter, G.X., Sussner, P. and Díaz de León, J.L. (1998). Morphological associative memories, *IEEE Transactions on Neural Networks* **2**(9): 281–293, DOI: 10.1109/72.661123.

Ritter, G.X. and Urcid, G. (2021). *Introduction to Lattice Algebra. With Applications in AI, Pattern Recognition, Image Analysis, and Biomimetic Neural Networks*, Chapman and Hall/CRC, Boca Raton.

Salgado-Ramírez, J.C., Vianney Kinani, J.M., Cendejas-Castro, E.A., Rosales-Silva, A.J., Ramos-Díaz, E. and Díaz-de Léon-Santiago, J.L. (2022). New model of heteroasociative min memory robust to acquisition noise, *Mathematics* **10**(148): 2–35, DOI: 10.3390/math10010148.

Sussner, P. (2000). Observations on morphological associative memories and the kernel method, *Neurocomputing* **31**(1–4): 167–183, DOI: 10.1016/S0925-2312(99)00176-9.

Sussner, P. and Valle, M.E. (2006). Implicative fuzzy associative memories, *IEEE Transactions on Fuzzy Systems* **14**(6): 793–807, DOI: 10.1109/TFUZZ.2006.879968.

Tikhonenko, O., Ziółkowski, M. and Kempa, W.M. (2021). Queueing systems with random volume customers and a sectorized unlimited memory buffer, *International Journal of Applied Mathematics and Computer Science* **31**(3): 471–486, DOI: 10.34768/amcs-2021-0032.

Urcid, G. and Ritter, G.X. (2007). Noise masking for pattern recall using a single lattice matrix associative memory, *in* V.G. Kaburlasos and G.X. Ritter (Eds), *Computational Intelligence Based on Lattice Theory*, Springer, Berlin/Heidelberg, pp. 81–100, DOI: 10.1007/978-3-540-72687-6_5.

Wang, S. and Lu, H. (2004). On new fuzzy morphological associative memories, *IEEE Transactions on Fuzzy Systems* **12**(3): 316–323, DOI: 10.1109/TFUZZ.2004.825977.

Wang, T. and Jia, N. (2017). A GCM neural network using cubic logistic map for information processing, *Neural Computing and Applications* **28**(7): 1891–1903, DOI: 10.1007/s00521-016-2407-4.

Wang, T., Jia, N. and Wang, K. (2012). A novel GCM chaotic neural network for information processing, *Communications in Nonlinear Science and Numerical Simulation* **17**(12): 4846–4855, DOI: 10.1016/j.cnsns.2012.05.011.

Xia, G., Tang, Z. and Li, Y. (2004). Hopfield neural network with hysteresis for maximum cut problem, *Neural Information Processing—Letters and Reviews* **4**(2): 19–26.

Xiao, P., Yang, F. and Yu, Y. (1997). Max-min encoding learning algorithm for fuzzy max-multiplication associative memory networks, *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Orlando, USA*, pp. 3674–3679, DOI: 10.1109/ICSMC.1997.633240.

Zhang, S., Lin, S. and Chen, C. (1993). Improved model of optical fuzzy associative memory, *Optics Letters* **18**(21): 1837–1839, DOI: 10.1364/OL.18.001837.

Zheng, L. and Tang, X. (2005). A new parameter control method for S-GCM, *Pattern Recognition Letters* **26**(7): 939–942, DOI: 10.1016/j.patrec.2004.09.041.

Arturo Gamino Carranza received his BS degree in electronic engineering from the Technological Institute of Veracruz, Mexico, in 2002, his MSc degree in automatic control from the Center for Research and Advanced Studies of the National Polytechnic Institute, Mexico, in 2004, and his PhD degree in education science from the Mexiquense College of Psychopedagogical Studies of Zumpango, Mexico, in 2023. Currently, he is a professor at the Merida Technological Institute/National Technological Institute of Mexico. His research interest includes associative memories, mathematical morphology and educational innovation.

Appendix

A1. Proof of Theorem 1

By Definition 4, $\mathbf{M_{XY}}$ is a $\overset{\circ}{\boxslash}$-perfect recalling memory if and only if

$$\left(\left(\mathbf{M_{XY}}\right)^{\downarrow}\overset{\circ}{\boxslash}\mathbf{x}^{\xi}\right)_i^{\downarrow}=\Big(\bigvee_{j=1}^{n}\Big(\left(m_{ij}\right)^{\downarrow}\circ x_j^{\xi}\Big)\Big)_i^{\downarrow}=y_i^{\xi}$$

$$\forall\xi=1,\dots,k \quad \text{and} \quad \forall i=1,\dots,m.$$

Let $i\in\{1,\dots,m\}$ and $\lambda\in\{1,\dots,k\}$ be arbitrarily selected; then

$$\Big((\mathbf{M_{XY}})^{\downarrow}\overset{\circ}{\boxslash}\mathbf{x}^{\lambda}\Big)_i=\bigvee_{j=1}^{n}\Big((m_{ij})^{\downarrow}\circ x_j^{\lambda}\Big). \tag{A1}$$

Selecting an arbitrary $j=j_0$; we get

$$\bigvee_{j=1}^{n}\Big((m_{ij})^{\downarrow}\circ x_j^{\lambda}\Big)\geq(m_{ij_0})^{\downarrow}\circ x_{j_0}^{\lambda}. \tag{A2}$$

By hypothesis $m_{ij_0}=y_i^{\lambda}\circ x_{j_0}^{\lambda}$, which implies

$$\Big(\Big(\mathbf{M_{XY}}\Big)^{\downarrow}\overset{\circ}{\boxslash}\mathbf{x}^{\lambda}\Big)_i\geq\Big(y_i^{\lambda}\circ x_{j_0}^{\lambda}\Big)^{\downarrow}\circ x_{j_0}^{\lambda}. \tag{A3}$$

To prove the converse, let $i\in\{1,\dots,m\}$ and $j\in\{1,\dots,n\}$; then, from Eqn. (8), we have that

$$m_{ij}=\bigvee_{\xi=1}^{k}\Big(y_i^{\xi}\circ x_j^{\xi}\Big). \tag{A4}$$

Therefore,

$$\begin{aligned}(m_{ij})^{\downarrow}&=\Big(\bigvee_{\xi=1}^{k}\Big(y_i^{\xi}\circ x_j^{\xi}\Big)\Big)^{\downarrow}\\ &\Leftrightarrow\Big(\bigvee_{\xi=1}^{k}\Big(y_i^{\xi}\circ x_j^{\xi}\Big)\Big)^{\downarrow}\geq\bigwedge_{\xi=1}^{k}\Big(y_i^{\xi}\circ x_j^{\xi}\Big)^{\downarrow}.\end{aligned} \tag{A5}$$

Let $\lambda\in\{1,\dots,k\}$ such that

$$\bigwedge_{\xi=1}^{k}\Big(y_i^{\xi}\circ x_j^{\xi}\Big)^{\downarrow}=(y_i^{\lambda}\circ x_j^{\lambda})^{\downarrow}.$$

Then,

$$\begin{aligned}&\Big(\bigvee_{\xi=1}^{k}\Big(y_i^{\xi}\circ x_j^{\xi}\Big)\Big)^{\downarrow}\geq(y_i^{\lambda}\circ x_j^{\lambda})^{\downarrow}\\ &\Leftrightarrow\Big(\bigvee_{\xi=1}^{k}\Big(y_i^{\xi}\circ x_j^{\xi}\Big)\Big)^{\downarrow}\circ x_j^{\lambda}\leq(y_i^{\lambda}\circ x_j^{\lambda})^{\downarrow}\circ x_j^{\lambda}\\ &\Leftrightarrow\bigvee_{j=1}^{n}\Big(\Big(\bigvee_{\xi=1}^{k}\Big(y_i^{\xi}\circ x_j^{\xi}\Big)\Big)^{\downarrow}\circ x_j^{\lambda}\Big)\\ &\quad\leq\bigvee_{j=1}^{n}\Big((y_i^{\lambda}\circ x_j^{\lambda})^{\downarrow}\circ x_j^{\lambda}\Big).\end{aligned} \tag{A6}$$

Let $j_0\in\{1,\dots,n\}$ such that

$$\bigvee_{j=1}^{n}\Big((y_i^{\lambda}\circ x_j^{\lambda})^{\downarrow}\circ x_j^{\lambda}\Big)=(y_i^{\lambda}\circ x_{j_0}^{\lambda})^{\downarrow}\circ x_{j_0}^{\lambda}.$$

Then

$$\bigvee_{j=1}^{n}\Big(\Big(\bigvee_{\xi=1}^{k}\Big(y_i^{\xi}\circ x_j^{\xi}\Big)\Big)^{\downarrow}\circ x_j^{\lambda}\Big)\leq\Big(y_i^{\lambda}\circ x_{j_0}^{\lambda}\Big)^{\downarrow}\circ x_{j_0}^{\lambda}. \quad \text{(A7)}$$

This last inequality implies that

$$\Big((\mathbf{M_{XY}})^{\downarrow}\mathring{\boxtimes}\mathbf{x}^{\lambda}\Big)_i\leq\big(y_i^{\lambda}\circ x_{j_0}^{\lambda}\big)^{\downarrow}\circ x_{j_0}^{\lambda}$$

and, according to Eqn. (A3), it is concluded that

$$\begin{aligned}\Big((\mathbf{M_{XY}})^{\downarrow}\mathring{\boxtimes}\mathbf{x}^{\lambda}\Big)_i&=\big(y_i^{\lambda}\circ x_{j_0}^{\lambda}\big)^{\downarrow}\circ x_{j_0}^{\lambda}\\ \Big((\mathbf{M_{XY}})^{\downarrow}\mathring{\boxtimes}\mathbf{x}^{\lambda}\Big)_i^{\downarrow}&=\Big(\big(y_i^{\lambda}\circ x_{j_0}^{\lambda}\big)^{\downarrow}\circ x_{j_0}^{\lambda}\Big)^{\downarrow}=y_i^{\lambda}.\end{aligned} \quad \text{(A8)}$$

In a similar we argue for $\mathbf{W_{XY}}$.

A2. Proof of Theorem 2

According to Theorem 1, if $\mathbf{M_{XY}}$ is a $\mathring{\boxtimes}$-perfect recalling memory for $(\mathbf{X},\mathbf{Y})$, then, for each row index $i=1,\ldots,m$ and each $\lambda\in\{1,\ldots,k\}$, there exist columns indices $j_0\in\{1,\ldots,n\}$ such that $m_{ij_0}=y_i^{\lambda}\circ x_{j_0}^{\lambda}$. From Eqns. (7) and (8) it follows that

$$y_i^{\lambda}\circ x_{j_0}^{\lambda}=(\mathbf{y}^{\lambda}\mathring{\boxtimes}(\mathbf{x}^{\lambda})^T)_{ij_0} \quad \text{(A9)}$$

and

$$m_{ij_0}=(\mathbf{M_{XY}})_{ij_0}.$$

Then $m_{ij_0}=y_i^{\lambda}\circ x_{j_0}^{\lambda}$ is equivalent to

$$\begin{aligned}(\mathbf{M_{XY}})_{ij_0}&=(\mathbf{y}^{\lambda}\mathring{\boxtimes}(\mathbf{x}^{\lambda})^T)_{ij_0},\\ (\mathbf{M_{XY}})_{ij_0}-(\mathbf{y}^{\lambda}\mathring{\boxtimes}(\mathbf{x}^{\lambda})^T)_{ij_0}&=0,\\ \Big(\mathbf{M_{XY}}-(\mathbf{y}^{\lambda}\mathring{\boxtimes}(\mathbf{x}^{\lambda})^T)\Big)_{ij_0}&=0.\end{aligned} \quad \text{(A10)}$$

This last equation shows that each row of matrix $\mathbf{M_{XY}}-(\mathbf{y}^{\lambda}\mathring{\boxtimes}(\mathbf{x}^{\lambda})^T)$ contains an entry with zero element. In a similar way we can argue for $\mathbf{W_{XY}}$.

A3. Proof of Corollary 1

According to Theorem 2,

$$\Big(\Big(\mathbf{M_{XY}}\Big)^{\downarrow}\mathring{\boxtimes}\mathbf{x}^{\xi}\Big)^{\downarrow}=\mathbf{y}^{\xi}\quad\forall\xi=1,\ldots,k$$

if and only if, for each $\lambda\in\{1,\ldots,k\}$ and each row index i, there exists a column index $j\in\{1,\ldots,n\}$ (which depends on both λ and j) such that

$$m_{ij}=(\mathbf{y}^{\lambda}\mathring{\boxtimes}(\mathbf{x}^{\lambda})^T)_{ij}=y_i^{\lambda}\circ x_j^{\lambda}. \quad \text{(A11)}$$

From Eqns. (8) and (A11) it follows that

$$\begin{aligned}y_i^{\lambda}\circ x_j^{\lambda}&=\bigvee_{\xi=1}^{k}\Big(y_i^{\xi}\circ x_j^{\xi}\Big),\\ (y_i^{\lambda}\circ x_j^{\lambda})^{\downarrow}\circ y_i^{\lambda}&=\Big(\bigvee_{\xi=1}^{k}\Big(y_i^{\xi}\circ x_j^{\xi}\Big)\Big)^{\downarrow}\circ y_i^{\lambda},\\ \Big((y_i^{\lambda}\circ x_j^{\lambda})^{\downarrow}\circ y_i^{\lambda}\Big)^{\downarrow}&=\Big(\Big(\bigvee_{\xi=1}^{k}\Big(y_i^{\xi}\circ x_j^{\xi}\Big)\Big)^{\downarrow}\circ y_i^{\lambda}\Big)^{\downarrow}\\ x_j^{\lambda}&=\Big(\Big(\bigvee_{\xi=1}^{k}\Big(y_i^{\xi}\circ x_j^{\xi}\Big)\Big)^{\downarrow}\circ y_i^{\lambda}\Big)^{\downarrow}.\end{aligned} \quad \text{(A12)}$$

In a similar way we can argue for $\mathbf{W_{XY}}$.

A4. Proof of Theorem 3

Let $i\in\{1,\ldots,m\}$ be arbitrarily selected; then,

$$\begin{aligned}\Big((\mathbf{M_{XY}})^{\downarrow}\mathring{\boxtimes}\tilde{\mathbf{x}}^{\lambda}\Big)_i&=\bigvee_{j=1}^{n}\left((m_{ij})^{\downarrow}\circ\tilde{x}_j^{\lambda}\right),\\ \Big((\mathbf{M_{XY}})^{\downarrow}\mathring{\boxtimes}\tilde{\mathbf{x}}^{\lambda}\Big)_i&\geq(m_{ij_0})^{\downarrow}\circ\tilde{x}_{j_0}^{\lambda}.\end{aligned} \quad \text{(A13)}$$

Since $\tilde{\mathbf{x}}^{\lambda}$ is a distorted vector with additive noise of $\mathbf{x}^{\lambda}$, we have

$$\begin{aligned}&\tilde{x}_j^{\lambda}\geq x_j^{\lambda}\quad\forall j=1,\ldots,n\\ \Leftrightarrow&(m_{ij})^{\downarrow}\circ\tilde{x}_j^{\lambda}\leq(m_{ij})^{\downarrow}\circ x_j^{\lambda}\quad\forall j=1,\ldots,n\\ \Leftrightarrow&\bigvee_{j=1}^{n}\left((m_{ij})^{\downarrow}\circ\tilde{x}_j^{\lambda}\right)\leq\bigvee_{j=1}^{n}\left((m_{ij})^{\downarrow}\circ x_j^{\lambda}\right).\end{aligned} \quad \text{(A14)}$$

Let $j_0\in\{1,\ldots,n\}$ such that

$$\bigvee_{j=1}^{n}\left((m_{ij})^{\downarrow}\circ x_j^{\lambda}\right)=(m_{ij_0})^{\downarrow}\circ x_{j_0}^{\lambda}.$$

Therefore,

$$\begin{aligned}\bigvee_{j=1}^{n}\left((m_{ij})^{\downarrow}\circ\tilde{x}_j^{\lambda}\right)&\leq(m_{ij_0})^{\downarrow}\circ x_{j_0}^{\lambda},\\ \Big((\mathbf{M_{XY}})^{\downarrow}\mathring{\boxtimes}\tilde{\mathbf{x}}^{\lambda}\Big)_i&\leq(m_{ij_0})^{\downarrow}\circ x_{j_0}^{\lambda}.\end{aligned} \quad \text{(A15)}$$

Let $\lambda\in\{1,\ldots,k\}$ be an index. According to Eqn. (5) there exists $m_{i_{j_0}}=y_i^{\lambda}\circ x_{j_0}^{\lambda}$, but, by hypothesis, $m_{i_{j_0}}=y_i^{\lambda}\circ\tilde{x}_{j_0}^{\lambda}$, which implies that $x_{j_0}^{\lambda}=\tilde{x}_{j_0}^{\lambda}$. Therefore, based on Eqns. (A13) and (A15), we have that

$$\begin{aligned}\Big((\mathbf{M_{XY}})^{\downarrow}\mathring{\boxtimes}\tilde{\mathbf{x}}^{\lambda}\Big)_i&=(m_{ij_0})^{\downarrow}\circ\tilde{x}_{j_0}^{\lambda},\\ \Big((\mathbf{M_{XY}})^{\downarrow}\mathring{\boxtimes}\tilde{\mathbf{x}}^{\lambda}\Big)_i&=(y_i^{\lambda}\circ\tilde{x}_{j_0}^{\lambda})^{\downarrow}\circ\tilde{x}_{j_0}^{\lambda},\\ \Big((\mathbf{M_{XY}})^{\downarrow}\mathring{\boxtimes}\tilde{\mathbf{x}}^{\lambda}\Big)_i^{\downarrow}&=((y_i^{\lambda}\circ\tilde{x}_{j_0}^{\lambda})^{\downarrow}\circ\tilde{x}_{j_0}^{\lambda})^{\downarrow}=y_i^{\lambda}.\end{aligned} \quad \text{(A16)}$$

In a similar we can argue for $\mathbf{W_{XY}}$.

A5. Proof of Theorem 4

Since $m_{ii} = (\mathbf{x}^\xi \overset{\circ}{\boxtimes} (\mathbf{x}^\xi)^T)_{ii} = (x_i^\xi \circ x_i^\xi) = 01$ for each $i = 1, \dots, n$ and $\forall \xi = 1, \dots, k$; then, each row of $\mathbf{M_{XX}} - \left(\mathbf{x}^\xi \overset{\circ}{\boxtimes} (\mathbf{x}^\xi)^T\right)$ contains a zero element. According to Theorem 2, $\mathbf{M_{XX}}$ is a $\overset{\circ}{\boxtimes}$-perfect recalling memory for $(\mathbf{X}, \mathbf{X})$. A similar argument can be applied to $\mathbf{W_{XX}}$.

A6. Proof of Theorem 5

Suppose that $\left((\mathbf{M_{XX}})^\downarrow \overset{\circ}{\boxtimes} \mathbf{z}\right)^\downarrow = \mathbf{u}$; then, by Theorem 4, $m_{ii} = 01$ for each $i = 1, \dots, n$ and $\forall \xi = 1, \dots, k$. Therefore

$$\left(\left(\mathbf{M_{XX}}\right)^\downarrow \overset{\circ}{\boxtimes} \mathbf{u}\right)_i^\downarrow = \left(\bigvee_{j=1}^{n} \left(\left(m_{ij}\right)^\downarrow \circ u_j\right)\right)^\downarrow,$$

$$\left(\left(\mathbf{M_{XX}}\right)^\downarrow \overset{\circ}{\boxtimes} \mathbf{u}\right)_i^\downarrow = \left(\left(\left(m_{ii}\right)^\downarrow \circ u_i\right) \vee \left(\bigvee_{j \neq i} \left(\left(m_{ij}\right)^\downarrow \circ u_j\right)\right)\right)^\downarrow,$$

$$\left(\left(\mathbf{M_{XX}}\right)^\downarrow \overset{\circ}{\boxtimes} \mathbf{u}\right)_i^\downarrow = \left(\left(1 \circ u_i\right) \vee \left(\bigvee_{j \neq i} \left(\left(m_{ij}\right)^\downarrow \circ u_j\right)\right)\right)^\downarrow. \tag{A17}$$

For the last inequality, if $u_i = 0$, then $1 \circ u_i = 10$ and the result of Eqn. (A17) is equal to $(10)^\downarrow = (1 \circ u_i)^\downarrow$. However, if $u_i = 1$, then $1 \circ u_i = 01$ and the result of Eqn. (A17) is equal to $(01)^\downarrow$ or $(10)^\downarrow$. Both results are less than or equal to $(1 \circ u_i)^\downarrow$. From the above, it follows that

$$\left((\mathbf{M_{XX}})^\downarrow \overset{\circ}{\boxtimes} \mathbf{u}\right)_i^\downarrow \leq (1 \circ u_i)^\downarrow = u_i, \tag{A18}$$

i.e.,

$$\mathbf{u} \geq \left((\mathbf{M_{XX}})^\downarrow \overset{\circ}{\boxtimes} \mathbf{u}\right)^\downarrow. \tag{A19}$$

We now will demonstrate the opposite. Let $i, j, l \in \{1, \dots, n\}$. Then

$$m_{il} = \bigvee_{\xi=1}^{k} \left(x_i^\xi \circ x_l^\xi\right) = \left(x_i^\lambda \circ x_l^\lambda\right)$$

$$\text{for } \lambda \in \{1, \dots, k\} \text{ and } \lambda \text{ equal to maximum,} \tag{A20}$$

$$m_{lj} = \bigvee_{\xi=1}^{k} \left(x_l^\xi \circ x_j^\xi\right) = \left(x_l^\lambda \circ x_j^\lambda\right)$$

for $\lambda \in \{1, \dots, k\}$ and λ equal to maximum, which implies that,

$$(m_{il})^\downarrow = \left(x_i^\lambda \circ x_l^\lambda\right)^\downarrow \tag{A21}$$

for $\lambda \in \{1, \dots, k\}$ and λ equal to maximum,

$$(m_{lj})^\downarrow = \left(x_l^\lambda \circ x_j^\lambda\right)^\downarrow \tag{A22}$$

for $\lambda \in \{1, \dots, k\}$ and λ equal to maximum.

Let

$$\left((m_{il})^\downarrow \circ (m_{lj})^\downarrow\right)^\downarrow = \left(\left(x_i^\lambda \circ x_l^\lambda\right)^\downarrow \circ \left(x_l^\lambda \circ x_j^\lambda\right)^\downarrow\right)^\downarrow$$

for $\lambda \in \{1, \dots, k\}$ and λ equal to maximum,

$$\left(\left(m_{il}\right)^\downarrow \circ \left(m_{lj}\right)^\downarrow\right)^\downarrow = \left(x_i^\lambda \circ \left(x_l^\lambda \circ \left(x_l^\lambda \circ x_j^\lambda\right)^\downarrow\right)^\downarrow\right)^\downarrow$$

for $\lambda \in \{1, \dots, k\}$ and λ equal to maximum,

$$\left(\left(m_{il}\right)^\downarrow \circ \left(m_{lj}\right)^\downarrow\right)^\downarrow = \left(x_i^\lambda \circ x_j^\lambda\right)^\downarrow$$

for $\lambda \in \{1, \dots, k\}$ and λ equal to maximum,

$$\left((m_{il})^\downarrow \circ (m_{lj})^\downarrow\right)^\downarrow = \left(\bigvee_{\xi=1}^{k} \left(x_i^\xi \circ x_j^\xi\right)\right)^\downarrow = \left(m_{ij}\right)^\downarrow. \tag{A23}$$

For this last inequality, we have that for $i = 1, \dots, n$

$$(m_{ij})^\downarrow = \left((m_{il})^\downarrow \circ (m_{lj})^\downarrow\right)^\downarrow \quad \forall l = 1, \dots, n,$$

$$(m_{ij})^\downarrow \circ z_j = \left((m_{il})^\downarrow \circ (m_{lj})^\downarrow\right)^\downarrow \circ z_j \quad \forall l = 1, \dots, n,$$

$$\left((m_{ij})^\downarrow \circ z_j\right)^\downarrow = \left(\left((m_{il})^\downarrow \circ (m_{lj})^\downarrow\right)^\downarrow \circ z_j\right)^\downarrow \quad \forall l = 1, \dots, n,$$

$$\left((m_{ij})^\downarrow \circ z_j\right)^\downarrow = \left((m_{il})^\downarrow \circ \left((m_{lj})^\downarrow \circ z_j\right)^\downarrow\right)^\downarrow \quad \forall l = 1, \dots, n,$$

$$\bigvee_{j=1}^{n} \left(\left((m_{ij})^\downarrow \circ z_j\right)^\downarrow\right) = \bigvee_{j=1}^{n} \left(\left((m_{il})^\downarrow \circ \left(\left(m_{lj}\right)^\downarrow \circ z_j\right)^\downarrow\right)^\downarrow\right) \quad \forall l = 1, \dots, n. \tag{A24}$$

The above holds true if

$$\Big(\bigvee_{j=1}^{n}\big((m_{ij})^{\downarrow}\circ z_j\big)\Big)^{\downarrow}$$
$$=\Big(\bigvee_{j=1}^{n}\big((m_{il})^{\downarrow}\circ\big((m_{lj})^{\downarrow}\circ z_j\big)\big)\Big)^{\downarrow}$$
$$\forall l=1,\dots,n,$$
$$\Big(\bigvee_{j=1}^{n}\big((m_{ij})^{\downarrow}\circ z_j\big)\Big)^{\downarrow}$$
$$=\Big((m_{il})^{\downarrow}\circ\big(\bigwedge_{j=1}^{n}\big((m_{ij})^{\downarrow}\circ z_j\big)^{\downarrow}\big)\Big)^{\downarrow}$$
$$\forall l=1,\dots,n,$$
$$\big(\bigvee_{j=1}^{n}\big((m_{ij})^{\downarrow}\circ z_j\big)\big)^{\downarrow}$$
$$\leq\Big((m_{il})^{\downarrow}\circ\big(\bigvee_{j=1}^{n}\big((m_{ij})^{\downarrow}\circ z_j\big)\big)^{\downarrow}\Big)^{\downarrow}$$
$$\forall l=1,\dots,n,$$

$$\begin{aligned} u_i &\leq \big((m_{il})^{\downarrow}\circ u_l\big)^{\downarrow}\quad \forall l=1,\dots,n,\\ u_i &\leq \bigwedge_{l=1}^{n}\big(\big((m_{il})^{\downarrow}\circ u_l\big)^{\downarrow}\big),\\ u_i &\leq \big(\bigvee_{l=1}^{n}\big((m_{il})^{\downarrow}\circ u_l\big)\big)^{\downarrow}=\big((\mathbf{M}_{\mathbf{XX}})^{\downarrow}\mathbin{\mathring{\boxed{\vee}}}\mathbf{u}\big)_i^{\downarrow}. \end{aligned}\tag{A25}$$

This demonstrates that

$$\mathbf{u}\leq\big((\mathbf{M}_{\mathbf{XX}})^{\downarrow}\mathbin{\mathring{\boxed{\vee}}}\mathbf{u}\big)^{\downarrow}.\tag{A26}$$

By Eqns. (A19) and (A26) $\mathbf{u}=\big((\mathbf{M}_{\mathbf{XX}})^{\downarrow}\mathbin{\mathring{\boxed{\vee}}}\mathbf{u}\big)^{\downarrow}$. In similar way we can argue for $\mathbf{W}_{\mathbf{XX}}$.

Received: 1 August 2022
Revised: 26 December 2022
Accepted: 27 February 2023