

# A Modification of the Lernmatrix for Real Valued Data Processing

José Juan Carbajal-Hernández<sup>1,2</sup>, Luis P. Sánchez-Fernández<sup>2</sup>,  
Luis A. Sánchez-Pérez<sup>2</sup>, Jesús Ariel Carrasco-Ochoa<sup>1</sup>,  
and José Francisco Martínez-Trinidad<sup>1</sup>

<sup>1</sup> Computer Science Department, National Institute of Astrophysics, Optics and Electronics,  
Luis Enrique Erro #1, Santa Maria Tonantzintla, Puebla, México

<sup>2</sup> Center of Computer Research – National Polytechnic Institute. Av. Juan de Dios Bátiz s/n,  
Nueva. Industrial Vallejo, Gustavo A. Madero, México D.F., México  
juancarvajal@sagitario.cic.ipn.mx

**Abstract.** An associative memory is a binary relationship between inputs and outputs, which is stored in an  $M$  matrix. In this paper, we propose a modification of the Steinbuch Lernmatrix model in order to process real-valued patterns, avoiding binarization processes and reducing computational burden. The proposed model is used in experiments with noisy environments, where the performance and efficiency of the memory is proven. A comparison between the proposed and the original model shows a good response and efficiency in the classification process of the new Lernmatrix.

**Keywords:** Associative memories, artificial intelligence, neurocomputing, pattern processing, classifier.

## 1 Introduction

An associative memory can be seen as a particular kind of neural network specially designed to recall output patterns in terms of input patterns that might appear altered by some kind of noise. Associative memories have demonstrated usefulness in the pattern processing field and considerable importance in the developed activities of numerous researchers, mainly in theory, recognition and pattern classification applications [2]. Karl Steinbuch developed the first associative memory in 1961, which works as a binary patterns classifier: the Lernmatrix [19, 20]. Researchers have tackled the problem of generating models of associative memories, achieving important results [5], [9], [13], [22] – [25]. In 1982, the John Hopfield work was a very important contribution of research in this field [8], due to Hopfield model works as an associative memory and as a neural network, unifying both research fields [1]. The Lernmatrix is a crucial precedent in the development of current associative memories models and is one of the first successful attempts to encode information in an arrangement, known as crossbar grids [16 – 18].

A particular limitation with data information is that associative memories in general, are based on processing binary patterns; real values ( $\Re$ ) must be binarized,

which implies huge information loss and high computational burden. In order to resolve this gap, different works implement algorithms for building associative memories with real valued data, using new or classical models as [13] and Vázquez and Sossa [21]. In this sense, the aim of this work is to introduce a new algorithm for storing and recovering real valued patterns in the Lernmatrix.

## 2 Associative Memories

An associative memory (**M**) is a system of inputs and outputs that relates as follows:  $x \rightarrow [\mathbf{M}] \rightarrow y$ . The input pattern is represented by a column vector denoted as  $x$  and the output pattern by a column vector  $y$ . The goal of an associative memory is to restore full patterns from input patterns that can be altered [12], [14], [15]. Each input pattern forms an association with the corresponding output pattern as  $(x, y)$ .

An associative memory **M** is represented by a matrix, whose  $ij^{th}$  component is  $m_{ij}$ . The **M** matrix is generated by a finite set of associations known as a fundamental set [12]. The set cardinality is denoted as  $p$ . For a positive integer  $\mu$ , the corresponding association will be denoted as follows:

$$\left\{ (x^\mu, y^\mu) \mid \mu = 1, 2, \dots, p \right\} . \tag{1}$$

The patterns that build the fundamental set associations are called fundamental patterns. If it holds that  $x^\mu = y^\mu \forall \mu = 1, 2, \dots, p$ , then the memory is autoassociative; otherwise it is heteroassociative. Each column vector that represents an input pattern has  $n$  components that fall within the set  $A$ , where  $A = \{0, 1\}$  and each column vector that represents an output pattern has  $m$  components that fall within the set  $A$  as follows:

$$x^\mu \in A^n \text{ and } y^\mu \in A^m \forall \mu = 1, 2, \dots, p . \tag{2}$$

An associative memory works in two clearly established phases:

1. Learning phase (creation of the associative memory **M**)
2. Recalling phase (operation of the associative memory **M**)

## 3 The Steinbuch’s Lernmatrix

The Lernmatrix is a heteroassociative memory that works as a classifier of binary patterns if output patterns are properly chosen. It is a system of input-output that accepts as input binary patterns  $x^\mu \in A^n$ , and produces output binary patterns such like  $y \in A^m$  [17], [19], [20]. It should be noted that there are  $m$  different classes, each one coded with a single rule: class  $k \in \{1, 2, \dots, m\}$  will be represented by a column vector whose components will be assigned by  $y_k^\mu = 1, y_j^\mu = 0$  for  $j = 1, 2, \dots, k - 1, k + 1, \dots, m$ .

### Learning Phase

The learning phase consists in finding the way to generate a matrix **M** that will store the information of the  $p$  associations of the fundamental set  $\{(x^1, y^1), \dots, (x^p, y^p)\}$  [3],

[4]. The process for determining each of the  $m_{ij}$  components can be described following two steps: 1) each of the  $m_{ij}$  components of  $\mathbf{M}$  is initialized with zero; 2) each of the  $m_{ij}$  components must be updated according the following rule:

$$m_{ij} = m_{ij} + \Delta m_{ij}. \tag{3}$$

where  $\Delta m_{ij}$  can be computed using the following expression:

$$\Delta m_{ij} = \begin{cases} \varepsilon & \text{if } x_j^\mu = 1 = y_i^\mu \\ -\varepsilon & \text{if } x_j^\mu = 0 \text{ and } y_i^\mu = 1 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where  $\varepsilon$  is a positive constant previously chosen.

**Recalling Phase**

The recalling phase consists in determining the corresponding class of an input pattern  $x^\omega \in A^n$ . Finding the class means to obtain  $y^\omega \in A^m$  that corresponds to  $x^\omega \in A^n$ ; according to the methodology for constructing the  $y^\mu$  patterns, the class should be obtained without ambiguity [10], [15].

The  $i^{th}$  component of  $y^\omega$  can be determined using the following expression, where  $\vee$  is the maximum operator.

$$y_i^\omega = \begin{cases} 1 & \text{if } \sum_{j=1}^n m_{ij} x_j^\omega = \bigvee_{h=1}^m \left[ \sum_{j=1}^n m_{hj} x_j^\omega \right] \\ 0 & \text{otherwise} \end{cases} . \tag{5}$$

**4 Our Proposed Modification in the Lernmatrix Model**

Our proposed algorithm is an addendum to the Lernmatrix learning and recalling phase, since the new algorithm is applied for real valued patterns that will be stored in the associative memory, avoiding the binarization process and reducing computational burden.

**4.1 New Learning Phase**

The new learning phase consists on storing the original input vector in the associative memory  $M$  in order to compute the complete information of real-valued patterns. The process for determining each of the  $m_{ij}$  components can be described following the next steps: a) inicialization process can be determined using steps 1 and 2 of section 3; b)  $\Delta m_{ij}$  can be determined using the following expression:

$$\Delta m_{ij} = \begin{cases} \varepsilon & \text{if } x_i^\mu = 1 = y_i^\mu \\ 0 & \text{otherwise} \end{cases} . \tag{6}$$

where  $\varepsilon$  is a positive constant previously chosen.

### 4.2 New Recalling Phase

The following concepts define the steps to follows in order to recover real valued patterns; those concepts are well documented and represent a novelty for the actual associative memory.

**Definition 1.** Let  $A=\{0,1\}$  and  $x^\omega \in A^n$  be a pattern, then the adjusted vector  $a_i$  is defined as follows:

$$a_i = \begin{cases} x_i^\omega & \text{if } x_i^\omega \neq 0 \\ \varepsilon & \text{if } x_i^\omega = 0 \end{cases} \tag{7}$$

**Definition 2.** Let  $a_i$  be the adjusted vector, then the inverse vector  $z_i^\omega$  is defined as follows:

$$z_i = \frac{1}{a_i} . \tag{8}$$

**Definition 3.** Let  $z_i$  be the inverse vector and  $m_{ij}$  be the  $ij^{th}$  component of an associative memory M, then the multiplicative matrix  $q_{ij}$  can be defined as follows:

$$q_{ij} = m_{ij} \cdot z_j . \tag{9}$$

**Definition 4.** Let  $q_{ij}$  be the multiplicative matrix, then the asymptotic matrix  $s_{ij}$  can be defined as follows:

$$s_{ij} = \tanh(q_{ij} - 1) . \tag{10}$$

**Definition 5.** Let  $s_{ij}$  be the asymptotic matrix, then the class vector  $s_i^\omega$  is defined as follows:

$$C_i^\omega = \sum_{j=1}^n |s_{ij}| . \tag{11}$$

Using the previous definitions, it is possible to enunciate the new Lernmatrix recalling rule. The  $i^{th}$  component of  $y^\omega$  is computed according to the following rule:

$$y_i^\omega = \begin{cases} 1 & \text{if } C_i^\omega = \bigwedge_{h=1}^m [C_h^\omega] \\ 0 & \text{otherwise} \end{cases} . \tag{12}$$

where  $\bigwedge$  is the minimum operator.

## 5 Numerical Examples

For a better understanding of the proposed algorithm, this section exemplifies the operation of the new Lernmatrix. Due to space limitations, one pattern is recovered in order to show the real-valued Lernmatrix operation; however, the reader can easily recover the rest of the proposed input patterns. Suppose a fundamental set with  $p = 3$ ,  $n = 5$  and  $m = 3$  and the fundamental associations expressed as ordered pairs:  $\{(x^1, y^1), (x^2, y^2), (x^3, y^3)\}$  as follows:

$$x^1 = \begin{pmatrix} 5.3 \\ 7.2 \\ 1 \\ -4 \\ 1.2 \end{pmatrix} \quad y^1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad x^2 = \begin{pmatrix} 3.1 \\ -3.2 \\ 0 \\ -5 \\ 2 \end{pmatrix} \quad y^2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad x^3 = \begin{pmatrix} 4.2 \\ -6.4 \\ -1.3 \\ 2.1 \\ 0 \end{pmatrix} \quad y^3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

*Learning phase*

To create the matrix **M** using the *p* associations of the fundamental set, it is necessary to follow steps 1 and 2 of section 4.1:

$$\begin{array}{cc} \text{Step 1} & \text{Step 2} \\ M = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} & M = \begin{pmatrix} 5.3 & 7.2 & 1 & -4 & 1.2 \\ 3.1 & -3.2 & \varepsilon & -5 & 2 \\ 4.2 & -6.4 & -1.3 & 2.1 & \varepsilon \end{pmatrix} \end{array}$$

In this case, we have chosen  $\varepsilon = 0.001$ , due to it provides a good performance of our algorithm. Once the learning phase is computed and the associative memory **M** is determined, the next step is to compute the recalling phase.

*Recalling phase*

The recalling phase for the proposed method using the input pattern  $x^l$  is computed using the matrix **M**. According to Definition 1 and 2, the inverse vectors  $z_i$  is calculated using Eq. (7) and (8) as follows:

$$z^1 = \begin{pmatrix} 0.188 \\ 0.138 \\ 1 \\ -0.25 \\ 0.833 \end{pmatrix}$$

According to Definition 3, the multiplicative matrices  $q_{ij}$  can be computed using expression (9) as follows:

$$q_{ij}^1 = \begin{pmatrix} 5.3 & 7.2 & 1 & -4 & 1.2 \\ 3.1 & -3.2 & \varepsilon & -5 & 2 \\ 4.2 & -6.4 & -1.3 & 2.1 & \varepsilon \end{pmatrix} \begin{pmatrix} 0.188 \\ 0.138 \\ 1 \\ -0.25 \\ 0.833 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 5.84 & -0.44 & \varepsilon & 1.25 & 1.66 \\ 0.79 & -0.88 & -1.3 & -0.52 & 0.8\varepsilon \end{pmatrix}$$

According to Definition 4, the asymptotic matrices  $s_{ij}$  can be calculated using (10):

$$s_{ij}^1 = \tanh(q_{ij}^1 - 1) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -0.392 & -0.894 & -0.761 & 0.244 & 0.582 \\ -0.204 & -0.955 & -0.980 & -0.909 & -0.761 \end{pmatrix}$$

Using Definition 5, the subclass vector  $C_i^0$  is obtained using the expression (11) as follows:

$$C_{ij}^1 = \sum_{j=1}^n |s_{ij}^1| = \begin{pmatrix} 0 + 0 + 0 + 0 + 0 \\ 0.392 + 0.894 + 0.761 + 0.244 + 0.582 \\ 0.204 + 0.955 + 0.980 + 0.909 + 0.761 \end{pmatrix} = \begin{pmatrix} 0 \\ 2.876 \\ 3.810 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \rightarrow y^1$$

Finally, the output class  $y^0$  is obtained according to expression (12).

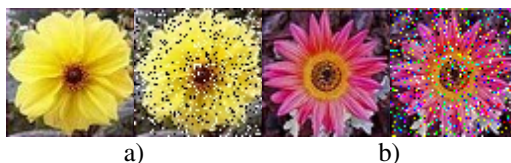
## 6 Experimental Results

Noise can be defined as the presence of an undesirable signal, since it degrades the accuracy and precision of analysis [11]. In binary codes, noise can appear in different forms. When one or several zeros are changed to one, it is known as additive noise. When one or several ones are changed to zero, it is known as subtractive noise. The combination between additive and subtractive noise is known as combined noise [6], [7]. In this case, true color images (24 bits image depth) of flowers (Fig. 1) with a  $65 \times 65$  pixel resolution were used as fundamental set in order to estimate the performance of the proposed algorithm with real data information. A true color pixel has a valued range from 0 to  $0x\text{FFFFFF}$ . Therefore, additive noise changes the value of selected pixels to the maximum value allowed for RGB pixels ( $0 \times \text{FFFFFF}$ ). Subtractive noise changes the value of selected pixels to the minimum value allowed for RGB pixels ( $0 \times \text{000000}$ ). Mixed noise changes values of selected pixels to the minimum value or maximum value allowed for RGB pixels randomly. Gaussian noise changes the value of selected pixels between the minimum and maximum value allowed for RGB using a Gaussian distribution function.

A comparison between models (original and proposed) was performed using a database of 20 patterns. Those patterns were binarized for making experiments with the original associative memory. Different levels of mixed and Gaussian noises were introduced in the fundamental pattern set in order to estimate the behavior of the proposed model with a density from 5% to 40% respectively.



**Fig. 1.** 20 flower images compound the fundamental set of patterns



**Fig. 2.** Different levels of a) mixed and b) Gaussian noises alter the input patterns (0, 5, 10, 15, 20, 25, 30, 35 and 40% of density)

**Table 1.** Comparison between the Steinbuch Lernmatrix and the Real-valued Lernmatrix

Noise	Steinbuch Lernmatrix		<i>Real Lernmatrix</i>	
	Combined	Gaussian	<i>Combined</i>	<i>Gaussian</i>
Density	Recalled patterns	Recalled patterns	<i>Recalled patterns</i>	<i>Recalled patterns</i>
0%	20	20	20	20
5%	18	18	20	20
10%	16	17	20	20
15%	15	16	20	20
20%	14	16	20	20
25%	14	15	20	20
30%	12	13	20	20
35%	12	12	19	20
40%	11	11	18	19

## 7 Conclusions

In this paper a modification for building a real valued data classifier was proposed using the Steinbuch's Lernmatrix. This model is one-shot trained and also is capable for storing and recalling real-valued patterns. Algorithms based on the original Lernmatrix work their computational processes only with binary patterns; this behavior increases the computational burden and presents information loss when real-valued patterns must be binarized. The proposed modification to the Lernmatrix worked perfectly with real patterns and increased the effectiveness in the pattern recalling phase against the original associative memory (Table 1).

Experimental results using true-color patterns and with different kinds of noise levels showed good performance on the recalling process of the proposed model and in this case, a better recovering of the input patterns. This improvement to the original Lernmatrix provides an alternative solution for the pattern processing field.

## References

1. Abu, M., Jacques, J.: Information capacity of the Hopfield model. *IEEE Transactions on Information Theory*, IT-31(4), 461–464 (1985)
2. Acevedo, M.: *Memorias asociativas bidireccionales alfa – beta*. Ph.D. Thesis, Centre of Computer Research, Institute Polytechnic National, Mexico (2006)
3. Amari, S.: Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Transaction on Computers* C-21(11), 1197–1206 (1972)
4. Anderson, J., Bower, G.: *Memoria Asociativa*, México, Limusa (1977)
5. Austin, J.: ADAM: a distributed associative memory for scene analysis. In: *Proceedings of First International Conference on Neural Networks*, pp. 285–295 (1987)
6. Flores, R., Yáñez, C.: *Memorias asociativas alfa beta basadas en código Johnson Möbius modificado*. Ms.C. Thesis. Center of Computer Research, Institute Polytechnic National, Mexico (2006)
7. Hassoun, M.: *Associative Neural Memories*. Oxford University Press, New York (1993)

8. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences* 79, 2554–2558 (1982)
9. Kohonen, T.: Correlation matrix memories. *IEEE Transactions on Computers* C-21(4), 353–359 (1972)
10. Kosko, B.: Bidirectional associative memories. *IEEE Transactions on Systems, Man and Cybernetics* 18(1), 49–60 (1998)
11. Lidner, D.: *Introducción a las señales y los sistemas*. McGraw Hill (1997)
12. Palm, G., Schwenker, F., Sommer, F.T., Strey, A.: Neural associative memories. In: Krikkelis, A., Weems, C.C. (eds.) *Associative Processing and Processors*, pp. 307–326. IEEE Computer Society, Los Alamitos (1997)
13. Ritter, G., Sussner, P., Días de León, J.: Morphological associative memories. *IEEE Transactions on Neural Networks* 9, 281–293 (1998)
14. Román, G., López, I., Yáñez, M.: A new classifier based on Associative Memories. In: *Proc. 15th International Conference on Computing CIC*, pp. 55–59. IEEE Computer Society, Los Alamitos (2006)
15. Sánchez, F., Días de León, J., Yáñez, C.: Lernmatrix de Steinbuch: condiciones necesarias y suficientes para recuperación perfecta de patrones. *Research on Computing Science* 10, 91–102 (2002)
16. Sánchez, A., Días de León, S., Yáñez, C.: New results on the Lernmatrix properties. *Research in Computing Science* 10, 91–102 (2004a)
17. Sánchez, A., Días de León, S., Yáñez, C.: Lernmatrix de Steinbuch: Avances Teóricos. *Computación y Sistemas* 7(3), 175–189 (2004b)
18. Simpson, P.K.: *Artificial Neural Systems*. Pergamon Press, New York (1990)
19. Steinbuch, K.: Die Lernmatrix. *Kybernetik* 1(1), 36–45 (1961)
20. Steinbuch, K., Frank, H.: Nichtdigitale Lernmatrizen als perzeptoren. *Kybernetik* 1(3), 117–124 (1961)
21. Vázquez, H., Sossa, H.: A bidirectional hetero-associative memory for true – color patterns. *Neural Processing Letters* 28, 131–153 (2008)
22. Willshaw, D., Buneman, O., Longuet-Higgins, H.: Non-holographic associative memory. *Nature* 222, 960–962 (1969)
23. Yáñez, C.: *Memorias Asociativas basadas en relaciones de orden y operadores binarios*. Ph.D. Thesis. Center of Computer Research, Institute Polytechnic National, Mexico (2002)
24. Yáñez, C., Días de León, J.: 2003a. *Memorias Autoasociativas Morfológicas max: condiciones suficientes para convergencia, aprendizaje y recuperación de patrones*. IT-175, Serie Azul, CIC-IPN, Mexico (2003) ISBN 970-36-0035-2
25. Yáñez, C., Días de León, J.: *Memorias Autoasociativas Morfológicas min: condiciones suficientes para convergencia, aprendizaje y recuperación de patrones*. IT-177, Serie Azul, CIC-IPN, Mexico (2003b) ISBN 970-36-0036-0