Associative Processing Applied to Word Reconstruction in the Presence of Letter Scrambling

Humberto Sossa, Ricardo Barrón and Benjamín Torres
Centro de Investigación en Computación-IPN
Av. Juan de Dios Bátiz, esquina con Miguel Othón de Mendizábal
Mexico City, 07738. MEXICO
E-mails: hsossa@cic.ipn.mx, rbarron@cic.ipn.mx, benjamincruz@sagitario.cic.ipn.mx

Abstract. In this note we describe how an associative memory can be applied to restore a word to its original "position" given a permutation of its letters. The idea is to first memorize a set of original words with different number of letters. Then the issue is to find the correct word given a permutation of its letters. We provide the formal conditions under which the proposal can be used to perfectly restore the desired word. We also give several examples to show the effectiveness of the proposal.

1 Introduction

A very well known word game is the following: Suppose we are given a set S_w of p different words of different cardinality $C(w_i), i = 1,2,\ldots,p$ (by cardinality we mean the number of letters of word w_i). From this set, at random, we have a word w_i but with its letters scrambled. After scrambling, some of the letters of the words will remain in their original positions, but some others not. The issue is to find the corresponding original word.

We humans do have a notable capacity to solve problems like this by iteratively exchanging the positions of the scrambled letters and making guesses. An exhaustive rearranging of all possible combinations would allow us to sooner or later find the searched word. By taking into account the orthographical rules of forming words would reduce the searching space. When the number of words, p, grows, the complexity of searching also grows.

Associative memories have been used for years to recover patterns from the unaltered or altered patterns keys. See for example [1-9]. In this work we propose to use an associative memory to restore a given word given a permutation of its letters.

2 Basics About Associative Memories

As defined by several researchers, an associative memory, denoted as **M** is a device with the capacity to relate input patterns and output patterns: $\mathbf{x} \rightarrow \mathbf{M} \rightarrow \mathbf{y}$, with **x** and **y**, respectively the input and output patterns vectors. Each input vector forms an association with a corresponding output vector.

© A. Gelbukh, R. Monroy. (Eds.)
Advances in Artificial Intelligence Theory
Research on Computing Science 16, 2005, pp. 141-150

An associative memory **M** is represented by a matrix whose *ij*-th component is m_{ij}. Matrix **M** is generated from a finite a priori set of known associations, known as the *fundamental set of associations*, or simply the *fundamental set* (FS). If ξ is an index, the fundamental set is represented as: $\{(\mathbf{x}^{\xi},\mathbf{y}^{\xi}) \mid \xi = 1,2,\ldots,p\}$ with p the cardinality of the set. Patterns that form the fundamental set are called *fundamental patterns*.

If it holds that $\mathbf{x}^{\xi} = \mathbf{y}^{\xi}\ \forall\ \xi \in \{1,2,\ldots p\}$, then **M** is auto-associative, otherwise it is hetero-associative. A distorted version of a pattern **x** to be recalled will be denoted as $\tilde{\mathbf{x}}$. If when presenting a distorted version of $\mathbf{x}^{w}$ with $w \in \{1,2,\ldots,p\}$ to an associative memory **M**, then it happens that the output corresponds exactly to the associated pattern $\mathbf{y}^{w}$, we say that recalling is robust.

3 Basics of Median Associative Memories

Median associative memories (MEDMEMs) first proposed in [9], have proven to be very powerful tools to recover patterns from distorted versions of their corresponding keys. Two associative memories are fully described in [9]. Due to space limitations, only hetero-associative memories are described. Auto-associative memories can be obtained simply by doing $\mathbf{x}^{\xi} = \mathbf{y}^{\xi}\ \forall\ \xi \in \{1,2,\ldots p\}$. Let us designate Hetero-Associative Median Memories as HAM-memories.

3.1 Memory construction

Two steps are required to build the HAM-memory. Let $\mathbf{x} \in \mathbf{Z}^{n}$ and $\mathbf{y} \in \mathbf{Z}^{m}$ two vectors:

Step 1: For each $\xi = 1,2,\cdots,p$, from each couple $(\mathbf{x}^{\xi},\mathbf{y}^{\xi})$ build matrix: $\mathbf{M}^{\xi}$ as:

$$\mathbf{M}^{\xi} = \begin{pmatrix} \mathrm{A}(y_1,x_1) & \mathrm{A}(y_1,x_2) & \cdots & \mathrm{A}(y_1,x_n) \\ \mathrm{A}(y_2,x_1) & \mathrm{A}(y_2,x_2) & \cdots & \mathrm{A}(y_2,x_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{A}(y_m,x_1) & \mathrm{A}(y_m,x_2) & \cdots & \mathrm{A}(y_m,x_n) \end{pmatrix}_{m\times n} \quad \textbf{(1)}$$

Step 2: Apply the median operator to the matrices obtained in Step 1 to get matrix **M** as follows:

$$\mathbf{M} = \operatorname*{\mathbf{med}}_{\xi=1}^{p}\left[\mathbf{M}^{\xi}\right]. \quad \textbf{(2)}$$

The *ij*-th component **M** is given as follows:

$$m_{ij} = \underset{\xi=1}{\overset{p}{\mathbf{med}}} \mathrm{A}\left(y_i^{\xi}, x_j^{\xi}\right). \quad \textbf{(3)}$$

3.2 Pattern recall

We have two cases:

Case 1: Recalling of a fundamental pattern. A pattern $\mathbf{x}^w$, with $w \in \{1,2,\cdots,p\}$ is presented to the memory **M** and the following operation is done:

$$\mathbf{M}\Diamond_{\mathrm{B}}\mathbf{x}^{w}. \quad \textbf{(4)}$$

The result is a column vector of dimension *n*, with *i*-th component given as:

$$\left(\mathbf{M}\Diamond_{\mathrm{B}}x^{w}\right)_i = \underset{j=1}{\overset{n}{\mathbf{med}}} \mathrm{B}\left(m_{ij}, x_j^{w}\right). \quad \textbf{(5)}$$

Case 2: Recalling of a pattern from an altered version of it. A pattern $\tilde{\mathbf{x}}$ (altered version of a pattern $\mathbf{x}^w$ is presented to the hetero-associative memory **M** and the following operation is done:

$$\mathbf{M}\Diamond_{\mathrm{B}}\tilde{\mathbf{x}}. \quad \textbf{(6)}$$

Again, the result is a column vector of dimension *n*, with *i*-th component given as:

$$\left(\mathbf{M}\Diamond_{\mathrm{B}}\tilde{\mathbf{x}}\right)_i = \underset{j=1}{\overset{n}{\mathbf{med}}} \mathrm{B}\left(m_{ij}, \tilde{x}_j\right). \quad \textbf{(7)}$$

Operators A and B might be chosen among those already proposed in the literature. In this paper we adopt operators A and B used in [5]. Operators A and B are defined as follows:

$$\mathrm{A}(x,y) = x - y \quad \textbf{(8.a)}$$

$$\mathrm{B}(x,y) = x + y \quad \textbf{(8.b)}$$

Sufficient conditions, for perfect recall of a pattern of the FS or from an altered version of them, according to [9] follow:

Proposition 1 [9]. Let $\{(\mathbf{x}^{\alpha}, \mathbf{y}^{\alpha}) \mid \alpha = 1,2,\ldots,p\}$ with $\mathbf{x}^{\alpha} \in \mathbf{R}^n$, $\mathbf{y}^{\alpha} \in \mathbf{R}^m$ the fundamental set of an HAM-memory **M** and let $(\mathbf{x}^{\gamma}, \mathbf{y}^{\gamma})$ an arbitrary fundamental couple with $\gamma \in \{1,\cdots,p\}$. If $\underset{j=1}{\overset{n}{\mathbf{med}}}\, \varepsilon_{ij} = 0$, $i = 1,\cdots,m$, $\varepsilon_{ij} = m_{ij} - \mathrm{A}\left(y_i^{\gamma}, x_j^{\gamma}\right)$ then $\left(\mathbf{M}\Diamond_{\mathrm{B}}\mathbf{x}^{\gamma}\right)_i = \mathbf{y}_i^{\gamma}, i = 1\ldots m$.

Corollary 1 [9]. Let $\{(\mathbf{x}^{\alpha}, \mathbf{y}^{\alpha}) \mid \alpha = 1,2,\ldots,p\}$, $\mathbf{x}^{\alpha} \in \mathbf{R}^n$, $\mathbf{y}^{\alpha} \in \mathbf{R}^m$. A HAM-median memory **M** has perfect recall if for all $\alpha = 1,\cdots,p$, $\mathbf{M}^{\alpha} = \mathbf{M}$ where

$\mathbf{M} = \mathbf{y}^{\xi} \Diamond_{\mathrm{A}} \left(\mathbf{x}^{\xi}\right)^{t}$ is the associated partial matrix to the fundamental couple $\left(\mathbf{x}^{\alpha}, \mathbf{y}^{\alpha}\right)$ and p is the number of couples.

Proposition 2 [9]. Let $\left\{\left(\mathbf{x}^{\alpha}, \mathbf{y}^{\alpha}\right) \mid \alpha = 1,2,\ldots,p\right\}$, $\mathbf{x}^{\alpha} \in \mathbf{R}^{n}$, $\mathbf{y}^{\alpha} \in \mathbf{R}^{m}$ a FS with perfect recall. Let $\eta^{\alpha} \in \mathbf{R}^{n}$ a pattern of mixed noise. A HAM-median memory **M** has perfect recall in the presence of mixed noise if this noise is of median zero, this is if $\operatorname*{\mathbf{med}}_{j=1}^{n} \eta_{j}^{\alpha} = 0, \forall \alpha$.

3.3 Case of a general fundamental set

In [10] was shown that due to in general a fundamental set (FS) does not satisfy the restricted conditions imposed by Proposition 1 and its Corollary, in [10] it is proposed the following procedure to transform a general FS into an auxiliary FS' satisfying the desired conditions:

TRAINING PHASE:

Step 1. Transform the FS into an auxiliary fundamental set (FS') satisfying Theorem 1:

1) Make $D = cont$, a vector.
2) Make $\left(\bar{\mathbf{x}}^{1}, \bar{\mathbf{y}}^{1}\right) = \left(\mathbf{x}^{1}, \mathbf{y}^{1}\right)$.
3) For the remaining couples do {
   For $\xi = 2$ to p {
   $\bar{\mathbf{x}}^{\xi} = \bar{\mathbf{x}}^{\xi-1} + D$; $\hat{\mathbf{x}}^{\xi} = \bar{\mathbf{x}}^{\xi} - \mathbf{x}^{\xi}$; $\bar{\mathbf{y}}^{\xi} = \bar{\mathbf{y}}^{\xi-1} + D$; $\hat{\mathbf{y}}^{\xi} = \bar{\mathbf{y}}^{\xi} - \mathbf{y}^{\xi}$ }

Step 2. Build matrix **M** in terms of set FS': Apply to FS' steps 1 and 2 of the training procedure described at the beginning of this section.

RECALLING PHASE:

We have also two cases, i.e.:

Case 1: Recalling of a fundamental pattern of FS:

1) Transform $\mathbf{x}^{\xi}$ to $\bar{\mathbf{x}}^{\xi}$ by applying the following transformation: $\bar{\mathbf{x}}^{\xi} = \mathbf{x}^{\xi} + \hat{\mathbf{x}}^{\xi}$.
2) Apply equations (**4**) and (**5**) to each $\bar{\mathbf{x}}^{\xi}$ of FS' to recall $\bar{\mathbf{y}}^{\xi}$.
3) Recall each $\mathbf{y}^{\xi}$ by applying the following inverse transformation: $\mathbf{y}^{\xi} = \bar{\mathbf{y}}^{\xi} - \hat{\mathbf{y}}^{\xi}$.

Case 2: Recalling of a pattern $\mathbf{y}^{\xi}$ from an altered version of its key: $\tilde{\mathbf{x}}^{\xi}$:

1) Transform $\tilde{\mathbf{x}}^{\xi}$ to $\bar{\mathbf{x}}^{\xi}$ by applying the following transformation: $\bar{\mathbf{x}}^{\xi} = \tilde{\mathbf{x}}^{\xi} + \hat{\mathbf{x}}^{\xi}$.
2) Apply equations (**6**) and (**7**) to $\bar{\mathbf{x}}^{\xi}$ to get $\bar{\mathbf{y}}^{\xi}$, and
3) Anti-transform $\bar{\mathbf{y}}^{\xi}$ as $\mathbf{y}^{\xi} = \bar{\mathbf{y}}^{\xi} - \hat{\mathbf{y}}^{\xi}$ to get $\mathbf{y}^{\xi}$.

In general, the noise added to a pattern does not satisfy the conditions imposed by Proposition 2. The following result (in the transformed domain) state the conditions under which MEDMEMs present perfect recall under general mixed noise [11]:

Proposition 3 [11]. Let $\{(\mathbf{x}^{\alpha}, \mathbf{y}^{\alpha}) \mid \alpha = 1,2,\ldots,p\}$, $\mathbf{x}^{\alpha} \in \mathbf{R}^{n}$, $\mathbf{y}^{\alpha} \in \mathbf{R}^{m}$ a fundamental set $\mathbf{x}^{\xi+1} = \mathbf{x}^{\xi} + D$, $\mathbf{y}^{\xi+1} = \mathbf{y}^{\xi} + D$, $\xi = 1,2,\ldots,p$, $D = (d,\cdots,d)^{T}$, $d = Const$. Without lost of generality suppose that is p odd. Thus the associative memory $\mathbf{M} = \mathbf{y}^{\xi} \Diamond_{\mathrm{A}} (\mathbf{x}^{\xi})^{T}$ has perfect recall in the presence of noise if less than $(n+1)/2-1$ of the elements of any of the input patterns are distorted by mixed noise.

4 The Proposal

The proposal to solve the problem described in section 1 is composed of two phases: Construction of the banks of memories and restoration of the word. The steps of each of these two phases are next explained. Also, in which follows, letters of words are represented in decimal ASCII code before further processing. This way letter “A” is represented thus as 65 in decimal ASCII code, letter “B” as 66, and so on.

4.1 Phase 1: Construction of the bank of memories

This phase has two steps as follows. Given a set S_w of p different words with different cardinality $C(w_i), i = 1,2,\ldots,p$:

Step 1: Group words according to their cardinality.

Step 2: For lowest cardinality C_{lowest} to biggest cardinality $C_{biggest}$:

1. Codify each word as explained.
2. Due to each FS does not satisfy conditions stated by Theorem 1 and Corollary 1, transform corresponding FS to auxiliary fundamental set FS’.
3. Built corresponding memory **M**.

4.3 Phase 2: Word restoration

This phase follows four steps. Given a scrambled word:

Step 1: Codify word in decimal ASCII code as explained.
Step 2: Transform codified version as explained in step 1 of case 2 of recalling phase (Section 3.3).
Step 3: Apply equations (**6**) and (**7**) to transformed version.
Step 4: Anti-transform recalled pattern to get desired pattern (step 3 of case 2 of recalling phase (Section 3.3)).

5 Numerical Example

To better understand the functioning of the proposal, let us suppose that we are given the following two sets of Spanish words grouped by cardinality 4 and 5 as follows:

$$\{\text{Gato, Sebo, Trío}\} \text{ and } \{\text{Félix, Lanar, Opino}\}.$$

Represented in decimal ASCII code, these two sets are as follows:

$$\{(71,97,116,111),(83,101,98,111),(84,114,161,111)\}$$
and
$$\{(70,130,108,105,120),(76,97,110,97,114),(79,112,105,110,111)\}.$$

Phase 1: Memory construction:

Step 1: Transformation of FS to auxiliary FS: Suppose that $d = 10$:

First FS. Words Gato, Sebo and Trío		Second FS. Words: Félix, Lanar and Opino	
Transformed vector	Difference	Transformed vector	Difference
(71,97,116,111)	(0,0,0,0)	(70,130,108,105,120)	(0,0,0,0,0)
(81,107,126,121)	(-2,6,28,10)	(80,140,118,115,130)	(4,43,8,18,16)
(91,117,136,131)	(-7,3,-25,20)	(90,150,128,125,140)	(11,38,23,15,29)

Step 2: Construction of memories:

According to the material exposed in Section 3.1 we have to memories, one for words of cardinality 3 and one for words of cardinality 4:

$$\mathbf{M}^1 = \begin{pmatrix} 0 & -26 & -45 & -40 \\ 26 & 0 & -19 & -14 \\ 45 & 19 & 0 & 5 \\ 40 & 14 & -5 & 0 \end{pmatrix} \text{ and } \mathbf{M}^2 = \begin{pmatrix} 0 & -60 & -38 & -35 & -50 \\ 60 & 0 & 22 & 25 & 10 \\ 38 & -22 & 0 & 3 & -12 \\ 35 & -25 & -3 & 0 & -15 \\ 50 & -10 & 12 & 15 & 0 \end{pmatrix}.$$

Phase 2: Word restoration:

Example 1: Given altered version Lanra, reconstruct corresponding word (Lanar):

Solution:

Step 1: Codification of word: Decimal ASCII code for scrambled version Lanra is: (76,97,110,114,97).

Step 2: Transformation of word: By adding difference vector (4,43,8,18,16) to altered version we get transformed vector (step 1 of case 2: Recalling of a pattern from an altered version of its key):

$$(76,97,110,114,97) + (4,43,8,18,16) = (80,140,118,132,116)$$

Step 3: Application of corresponding memory transformed version: In this case we apply matrix $\mathbf{M}^2$ and equations (6) and (7) to transformed version. We get:

$$(80,140,118,115,130)$$

Step 4: Anti-transformation of recalled pattern to get desired pattern. As explained in section 4.3 this is done by subtracting from recalled pattern corresponding difference vector. In this case vector (4,43,8,18,16). We get:

$$(80,140,118,115,130) - (4,43,8,18,16) = (80,140,118,97,114),$$

which corresponds as you can appreciate to word: Lanar.

6 Experimental Results

In this section we show how the proposal described in section 4 can be used to recover a given word from a scrambled version of its letters. For this the set Spanish words shown in Table 1 is used:

Number of letters per word and words used in the experiments		
5	7	9
ABRIR	DECIMAL	CASADEROS
FÉLIX	FLUVIAL	COLIBRÍES
IBIZA	FORMADO	INCOMODEN
LANAR	IDIOTEZ	POPULARES
OPINO	LINCHAR	VIOLENCIA
PANAL	OSTENDE	-
RUEDA	SEMANAL	-
RUGBY	-	-
TRINA	-	-

Table 1. List of words used in the experiments.

6.1 Memory construction

Each word is first codified in decimal ASCII as specified. Each set of codified words, beginning by words cardinality 5 and ending with words of cardinality 9, is then transformed to its corresponding auxiliary fundamental set. First codified word of each set is used to build corresponding associative memory. At the end of the process we end with six matrices: $\mathbf{M}^1$, $\mathbf{M}^2$ and $\mathbf{M}^3$. Matrix $\mathbf{M}^1$ codifies the information of words with 5 letters. Matrix $\mathbf{M}^2$ codifies the information of words with 7 letters, while matrix $\mathbf{M}^3$ codifies the information of words with nine letters.

6.2 Recalling of each fundamental set

Each word of each set was transformed and presented to its corresponding memory. Of course, due to Theorem 1 and its Corollary all words were perfectly recalled.

6.3 Recalling of a word from a distorted version of it (first experiment)

In this experiment less than 50% of the letters of each word of Table 1 were exchanged. In the case of words of five letters two letters were chosen, in the case of words of seven letters three letters were chosen, and in the case of the words of nine letters four letters were exchanged. One scrambled version of each word was generated. Each scrambled version was processed as described and the results were summarized in Table 2. As can be seen from this table, in all cases the desired word was correctly recalled. This of course is an expected result due to the noise added to the patterns satisfies the conditions for recalling specified by Proposition 3.

6.4 Recalling of a word from a distorted version of it (second experiment)

In this experiment more than 50% of the letters of each word of Table 1 were exchanged. In the case of words of four letters two letters were chosen, in the case of words of seven letters six letters were chosen, and in the case of the words of eight letters four letters were exchanged. One scrambled version of each word was generated. Each scrambled version was processed as described and the results were summarized in Table 3. As can be seen from this table, in some cases the desired word was not correctly recalled, in other cases it was. One can ask why of this fact. In the one hand it was simply because the percentage of 50% given by Proposition 3 was surpassed. In the other hand we have to remember that if the noise added to a pattern is median zero, it does not matter how the pattern is altered it should be correctly recalled. This is exactly what is happening in this case. Let us take for example altered version FDOAMOR (decimal ASCII code: 70 68 79 65 77 79 82) of word FORMADO. You can easily verify that the noise added to word DECIMAL (decimal ASCII code: 70 79 82 77 65 68 79) is:

70	68	79	65	77	79	82
70	79	82	77	65	68	79
0	-11	-3	-12	12	11	3

By arranging the component of last row and by taking the median we have that the median of the noise added to word in ASCII code equals median (-12,-11,-3,0,3,11,12)=0. Thus despite more than 50% of the components of the word are modified correct recalled is obtained. In the remaining cases the word was not correctly recalled because more than 50% of its components were altered and because the noise added to the word has no median equal to 0. From this we can conclude that Proposition 2 is a stronger than Proposition 3.

Word	Generated word	Recalled word
ABRIR	AIRBR	ABRIR
FELIX	XELIF	FELIX
IBIZA	IAIZB	IBIZA
LANAR	LANRA	LANAR
OPINO	OIPNO	OPINO
PANAL	PALAN	PANAL
RUEDA	RAEDU	RUEDA
RUGBY	RUGYB	RUGBY
TRINA	RTINA	TRINA

(a)

Word	Generated word	Recalled word
DECIMAL	DECLMAI	DECIMAL
FLUVIAL	ALUVIFL	FLUVIAL
FORMADO	FOOMADR	FORMADO
IDIOTEZ	ZDIOTEI	IDIOTEZ
LINCHAR	LIACHNR	LINCHAR
OSTENDE	ESTONDE	OSTENDE
SEMANAL	SAMANEL	SEMANAL
-	-	-
-	-	-

(b)

Word	Generated word	Recalled word
CASADEROS	CESADAROS	CASADEROS
COLIBRÍES	COLIBIRES	COLIBRÍES
INCOMODEN	INCOMODEN	INCOMODEN
POPULARES	PUPOLARES	POPULARES
VIOLENCIA	VIOCENLIA	VIOLENCIA

(c)

Table 2. Recalling results. (a) Words of five letters. (b) For words of 7 letters. (c) For words of 9 letters. In all cases the desired word was correctly recalled.

Word	Generated word	Recalled word
ABRIR	ARIRB	ABRIR
FELIX	FXLEI	FELIX
IBIZA	IZAIB	IBIZA
LANAR	LNARA	LANAR
OPINO	ONOPI	OPINO
PANAL	PLANA	PANAL
RUEDA	REAUD	RUEDA
RUGBY	RYBUG	RUGBY
TRINA	TANIR	TRINA

(a)

Word	Generated word	Recalled word
DECIMAL	LACEDMI	**ABAFJAI**
FLUVIAL	UILLVAF	**CIRSFAI**
FORMADO	FDOAMOR	FORMADO
IDIOTEZ	TEIIZDO	IDIOTEZ
LINCHAR	NCLRHIA	LINCHAR
OSTENDE	DTNSOEE	**PTUFOEF**
SEMANAL	MSEALNA	**QCKALAJ**
-	-	-
-	-	-

(b)

Word	Generated word	Recalled word
CASADEROS	SACORADSE	CASADEROS
COLIBRÍES	RIESBCLIO	COLIBRÍES
INCOMODEN	MEDONCNOI	**JODPNPEFO**
POPULARES	LAROPUSEP	POPULARES
VIOLENCIA	VAICNELOI	VIOLENCIA

(c)

Table 3. Recalling results. (a) Words of five letters. (b) For words of 7 letters. (c) For words of 9 letters.

It is worth to mention than during recall if the value of a recalled letter goes under the value 65 (‘A’) or above the value 90 (‘Z’), this value is to 65 and 90. This way in a recalled word we avoid having symbols different from letters.

7 Conclusions and Present Research

In this brief note we have shown how an associative memory can be used to find (recover) a desired word given a scrambled version of it. The scrambled version is first taken to a transformed domain were it can be operated by the corresponding memory. This operation automatically reorders the letters of the scrambled word.

In the general case we do not know from which word a distorted version was obtained. We are actually working through an efficient method that allows to recognize a given from a distorted version of it without having to compare it with all possible words. We are also looking for more real situations were the proposal could find applicability.

Acknowledgements. This work was economically supported by CGPI-IPN under grants 20050156 and CONACYT by means of grant 46805. H. Sossa specially thanks COTEPABE-IPN, CONACYT (Dirección de Asuntos Internacionales) and DAAD (Deutscher Akademischer Austauschdienst) for the economical support granted during research stay at Friedrich-Schiller University, Jena, Germany.

References

1. K. Steinbuch (1961). die Lernmatrix, Kyberneitk C-1,1,26-45.
2. J. Anderson (1972). A simple neural network generating an interactive memory, Mathematical Biosciences C-14, 197-220.
3. T. Kohonen (1972). Correlation matrix memories, IEEE Transactions Computers C-21, 4, 444-445.
4. J. Hopfield (1982). Neural Network and Physicals systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences, C-79, 2554-2558.
5. G. X. Ritter qt al. (1998). Morphological associative memories, IEEE Transactions on Neural Networks, C-9,281-293.
6. G. X. Ritter et al. (1999). Morphological bi-directional associative memories, Neural Networks, 12:851-867.
7. H. Sossa et al. (2005). Associative gray-level pattern processing using binary decomposition and a-b memories. To appear in Neural Processing Letters.
8. C. Yáñez (2002). Associative Memories based on Order Relations and Binary Operators (In Spanish), PhD Thesis, Center for Computing Research-IPN.
9. H. Sossa, R. Barrón and R. A. Vázquez (2004). New Associative Memories to Recall Real-Valued Patterns. LNCS 3287. Springer Verlag. Pp. 195-202.
10. H. Sossa and R. Barrón (2004). Transforming Fundamental Set of Patterns to a Canonical Form to Improve Pattern Recall. LNAI 3315. Springer Verlag. Pp. 687-696.
11. H. Sossa and R. Barrón (2005). Median Associatives Memories: New Results. Submmited to CIARP 2005.