Generation of Associative Memories Using Cellular Automata

B. Luna-Benoso  
Instituto Politécnico Nacional  
Escuela Superior de Cómputo  
Av. Juan de Dios Bátiz, esq. Miguel Othón de Mendizábal, Ciudad de México 07320, México

J. C. Martínez Perales  
Instituto Politécnico Nacional  
Escuela Superior de Cómputo  
Av. Juan de Dios Bátiz, esq. Miguel Othón de Mendizábal, Ciudad de México 07320, México

R. Flores-Carapia  
Instituto Politécnico Nacional  
Centro de Innovación y Desarrollo Tecnológico en Cómputo  
Av. Juan de Dios Bátiz, esq. Miguel Othón de Mendizábal, Ciudad de México 07700, México
Abstract

This paper presents the proposal of an associative memory implemented model with cellular automata. The model was applied to the iris plant database of the repertoire bases available by the UCI Machine Learning Repository. The model was compared with others by the reported performance making use of the k-fold cross validation.

Keywords: cellular automaton, associative memories, patterns classification.

1. Introduction

The concept of a cellular automaton (CA) was introduced in 1951 by John Von Neumann [1]. Von Neumann defines a cellular automaton as a space able to reproduce itself [2]. The cellular Automaton are mathematical models where the behavior of each one of the elements in the system depends of the local interaction with each other. A CA d-dimensional consist in a lattice or lattice d-dimensional extended infinitely that represents the "space", where each site of the lattice is called cell and have asociated a state variable, called the cell state that fluctuates on an infinite set, called state set. The time advances in discrete stages and the dynamic is given by an explicit rule called local function; the local function is used in each time stage for each cell to determine its new state from the current state of certain cells in its neighborhood. The cells alter their states synchronously in discrete time stages according to the local function. The Lattice is homogeneous so that all cells operate under the same local function. The state assignation to all the cells in the lattice is called a configuration, which is considered as the state of the total lattice. The cellular automatons have had a variety of applications in various science disciplines [3,4,5,6,7].

Oblivious to the field of cellular automata, there is the development and study of pattern recognition, and a problem of this area refers to the patterns classifications. The objective in the classification consists in partition the characteristic space to generate regions, which will be assigned to a category or a class. Different patterns must be assigned in some of the created regions in the characteristics space. In general, the full description of the classes is unknown. instead of this, there is a finite and reduced set of patterns that provides partial information about a specific problem.

Moreover, there is the development of associative memories, which have been in force since the early 60's. The fundamental proposal of an associative memory is recover correctly full patterns from input patterns, which can be altered with additive noise, subtractive or combined. The patterns classification is one of the applications that are given to the associative memories.

Several researchers have addressed the problem of developing models of associative memories [8,9,10,11,12,13] and have achieved important results for the field of research.

2. Associative Memories.

An associative memory can be formulated as a system input and output which is divided into two phases:

Learning phase: \( x \rightarrow [M] \leftarrow y \) (associative memory generation).

Recovering phase: \( x \rightarrow [M] \rightarrow y \) (associative memory operation).

The input pattern is represented by a column vector denoted by \( x \) and the output pattern for a column vector denoted by \( y \). Each one of the input patterns generate an association with the corresponding output pattern. The notation for an association is similar to an ordered pair \((x, y)\).

The associative memory \( M \) is represented by a matrix whose component \( ij \)-th is \( m_{ij} \) [14]; the matrix \( M \) is generated from a finite set of associations previously known, called fundamental set. We denote by \( p \) the cardinality of the fundamental set.

The fundamental set is represented as follows:

\[ \{(x_\mu, y_\mu) \mid \mu = 1,2,\ldots, p\} \]

The patterns that form the fundamental set associations are called fundamental patterns.
3. Cellular Automata

Be $A_\alpha$ a countable family of closed intervals in $\mathbb{R}$ such that meet the following conditions:

1. $\bigcup_{X \in A_\alpha} X = [a, b]$ for some $a, b \in \mathbb{R}$ or

$\bigcup_{X \in A_\alpha} X = \mathbb{R}$.

2. $[a_i, b_i] \in A_\alpha \Rightarrow b_i - a_i > 0$.

3. $[a, b_i], [c_j, d_j] \in A_\alpha \Rightarrow [a, b_i] \cap [c_j, d_j] = \emptyset \vee [a, b_i] \cap [c_j, d_j] = b_i - c_j$

Definition 3.1 Be $[a, b]$ an interval of $\mathbb{R}$ with $a \neq b$ and $A_\alpha$ a closed intervals family that satisfy 1,2 and 3. A lattice of dimensions 1 or 1-dimensional is the set $L = \{x_i \times [a_i, b_i] | x_i \in A_\alpha\}$. If $A_{\alpha_1}, A_{\alpha_2}, ..., A_{\alpha_m}$ are intervals families that satisfy 1, 2 and 3, so a lattice of dimension $n > 1$ is the set $L = \{x_{a_1} \times x_{a_2} \times \cdots \times x_{a_n} | x_{a_i} \in A_{\alpha_i}\}$.

Definition 3.2 Be $r \in \mathbb{R}$ a lattice 1-dimensional is regular if $[a_i, b_i] = r$ for each $[a_i, b_i] \in A_\alpha$. A lattice n-dimensional is regular if $[a_{a_i}, b_{a_i}] = r$ for each $[a_{a_i}, b_{a_i}] \in A_{\alpha_i}$ for $i = 1, ..., n$.

Definition 3.3 Be $L$ a lattice. A cellule, cell or site is an element of $L$. This is, a cell is an element of the form $[a_{a_1, \ldots, a_{a_n}}, b_{a_1, \ldots, a_{a_n}}] \times \cdots \times [a_{a_n, \ldots, a_n}, b_{a_n, \ldots, a_n}]$ with $[a_{a_1, \ldots, a_{a_n}}, b_{a_1, \ldots, a_{a_n}}] \in A_{\alpha_i}$ for $i = 1, ..., n$.

Definition 3.4 Be $L$ a lattice, and is $r$ a cell of $L$. A neighbourhood of size $n \in \mathcal{N}$ to $r$, is the set $v(r) := \{k_1, k_2, ..., k_n\} | k_j$ is a cell of $L$ for each $j$.

Definition 3.5 Be $n \in \mathcal{N}$. A cellular automata, is a tuple $(L, S, \mathcal{N}, f)$ such that:

1. $L$ is a regular lattice.

2. $S$ is a finite set of states

3. $\mathcal{N}$ is a defined neighborhoods set as follows.

$\mathcal{N} = \{\mathcal{N}(r) | r \text{ is a cell and } \mathcal{N}(r) \text{ is a neighborhood of } r \text{ of size } n\}$

4. $f : \mathcal{N} \rightarrow S$ is a function called transition function.

Definition 3.6 Is $Q = (L, S, \mathcal{N}, f)$ and $W = (L, S, \mathcal{N}', g)$ two CA. Is defined the CA composition of the CA $Q$ y $W$ in the time $t = t_0$ denoted as $W \ast Q$ by the CA $W \ast Q = (L, S, \mathcal{N}, f)$ where $h, f, y$ and $g$ are related as follows:

$C_{t_0 + 1}(r) = f(\{C_{t_0}(i) : i \in \mathcal{N}(r)\})$

$C_{t_0 + 2}(r) = g(\{C_{t_0}(i) : i \in \mathcal{N}'(r)\})$

$C_{t_0 + 2}(r) = h(\{C_{t_0}(i) : i \in \mathcal{N}(r)\})$

Fig. 2. Example of a CA composition.

4. Proposed model

This section will build the associative memory by Cellular Automata.

In what follows, consider the set $A = \{0, 1\}$ an the fundamental set $CF = \{(x^\mu, y^\mu) | \mu = 1, 2, ..., p\}$ with $x^\mu \in A^n$ and $y^\mu \in A^m$.

The lattice $L$ for the CA will be composed by the matrix of size $2m \times 2n$ with the first index the couple $(0,0)$. 
The set \( S = \{0,1\} \) is the finite states set

\[
I = \{i \in \mathbb{Z} | i = 2k \text{ for some } k = 0,1,2,\ldots, n - 1\} = \{0,2,4,\ldots,2(n-2)\} \text{ and } J = \{j \in \mathbb{Z} | j = 2k + 1 \text{ for some } k = 0,1,2,\ldots, m - 1\} = \{1,3,5,\ldots,2m-1\}.
\]

Consider the partition of \( \mathcal{L} \) formed by the subsets family \( \mathcal{L} = \{v(i,j) | (i, j) \in I \times J\} \) with \( v(i,j) = (i, j), (i, j-1), (i+1, j), (i+1, j-1)\}. Inasmuch as \( \mathcal{L} \) is a partition of \( \mathcal{L} \), given \( v^l \) exist an unique \( \mathcal{L} \) such that \( v^l = v(i,j) \). For example, if \( l = (3,0) \), so \( l \in v^l(3,0) = v(3,0) = \{(2,1), (2,0), (3,1), (3,0)\} \).

From the previous fact is defined the neighbourhood set

\[
N = \{v' | l \in L\}
\]

**Definition 4.1** Consider the set \( A^k \). Is defined the projected function of the \( i \)-th component \((1 \leq i \leq k) Pr_i : A^k \rightarrow A \) as

\[
Pr_i(z) = z_i, \text{ con } z = (z_1, z_2, \ldots, z_k)
\]

**Proposition 4.2** If \((y_i, x_j) \in Pr_{yx} = \{(y_i, x_j) | y_i = Pr_y(y) \text{ and } x_j = Pr_x(x)\}, \) so

\[
(2j - 2 + y_i, 2i - 2 + x_j) \in v(2j-2,2i-1,).
\]

**Demonstration** Must be

\[
v(2j-2,2i-1) = \{(2j - 2, 2i - 1), (2j - 2, 2i - 2)\},
\]

Inasmuch as \((y_i, x_j) \in Pr_{yx}, \text{ so } y_i = Pr_y(y) \) \( y_j = Pr_{j}(x) \) and in as much as \( x \in A^n \) and \( \in A^n \), so \( y_i, x_j \in \{0,1\} \), then

1. if \( y_i = x_j = 0 \), then

\[
(2j - 2 + y_i, 2i - 2 + x_j) = (2j - 2, 2i - 2) \in v(2j-2,2i-1,).
\]

2. if \( y_i = 0 \) \( x_j = 1 \), then

\[
(2j - 2 + y_i, 2i - 2 + x_j) = (2j - 2, 2i - 1) \in v(2j-2,2i-1,).
\]

3. if \( y_i = 1 \) \( x_j = 0 \), then

\[
(2j - 2 + y_i, 2i - 2 + x_j) = (2j - 1, 2i - 2) \in v(2j-2,2i-1,).
\]

4. if \( y_i = x_j = 1 \), then

\[
(2j - 2 + y_i, 2i - 2 + x_j) = (2j - 1, 2i - 1) \in v(2j-2,2i-1,).
\]

Is defined the set \( \mathcal{L}_{CF} = \{(2j - 2 + y_i, 2i - 2 + y_j^a) | 1 \leq \mu \leq p, 1 \leq i \leq m, 1 \leq j \leq n\} \subseteq l. \)

Consider the cellular automata \( Q = (\mathcal{L}, S, N, f_Q) \) and \( W = (\mathcal{L}, S, N', f_w) \) with

\[
N' = I, \text{ y}
\]

\[
f_Q : N \rightarrow S, f_w : N' \rightarrow S \text{ defined as follows:}
\]

\[
f_Q(v^{(i,j)}) = \begin{cases} 
1 & \text{if (i,j) } \in \mathcal{L}_{CF} \\
1 & \text{if (i,j) } \notin \mathcal{L}_{CF} 
\end{cases}
\]

It defines the CA associative (CAA) in its learning phase as \( W*Q = (\mathcal{L}, S, N, f_{A^*}) \).

For the learning phase, is represented two
different algorithms, aim is recover an associative pattern associated to an input pattern. First is represented the max algorithm to recover the patterns and follows the min algorithm to recover the patterns.

Algorithm to recovering patterns max

INPUT : Pattern that recognized \( \bar{x} = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_n \end{pmatrix} \)

OUTPUT: Recovered pattern \( \bar{y} = \begin{pmatrix} \bar{y}_1 \\ \bar{y}_2 \\ \vdots \\ \bar{y}_n \end{pmatrix} \)

PROCESS:

\[ \text{for } i = 1,2,\ldots,m \]
\[ \bar{y}_i = 0 \]

\[ \text{for } j = 1,2,\ldots,n \]
\[ \text{if } \bar{x}_j = 1 \& \&(2j-2,2i-1) = 1 \]
\[ \text{continue} \]
\[ \text{if } \bar{x}_j = 0 \& \&(2j-2,2i-1) = 1 \]
\[ \text{continue} \]
\[ \text{else} \]
\[ \bar{y}_i = 1 \]
\[ \text{break} \]

end second for

end first for

Algorithm to recovering patterns min

INPUT : Pattern to recognized \( \bar{x} = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_n \end{pmatrix} \)

OUTPUT: Recovered Pattern \( \bar{y} = \begin{pmatrix} \bar{y}_1 \\ \bar{y}_2 \\ \vdots \\ \bar{y}_n \end{pmatrix} \)

\[ \text{for } i = 1,2,\ldots,m \]
\[ \bar{y}_i = 0 \]

\[ \text{for } j = 1,2,\ldots,n \]
\[ \text{if } \bar{x}_j = 1 \& \&(2j-2,2i-1) = 1 \]
\[ \text{continue} \]
\[ \text{if } \bar{x}_j = 0 \& \&(2j-2,2i-1) = 1 \]
\[ \text{continue} \]
\[ \text{else} \]
\[ \bar{y}_i = 0 \]
\[ \text{break} \]

end second for

end first for

Example 4.3. Be \( m = 4, n = 3 \) and \( p = 3 \). The fundamental set \( CF = \{ (x^1, y^1), (x^2, y^2), (x^3, y^3) \} \) is given by:

\[
\begin{align*}
\text{for } i = 1,2,\ldots, m \quad & \quad x^1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\
\text{for } j = 1,2,\ldots, n \quad & \quad y^1 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \\
\text{for } i = 1,2,\ldots, m \quad & \quad x^2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \\
\text{for } j = 1,2,\ldots, n \quad & \quad y^2 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\
\text{for } i = 1,2,\ldots, m \quad & \quad x^3 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \\
\end{align*}
\]

- The lattice \( L \) is composed by the matrix of size \( 2m \times 2n = 8 \times 6 \).
- The states set \( S = \{0,1\} \).
- The Neighbourhood set is given by \( N = \{v^i, 1 \in L\} \).
- The next set \( L_{CE} \) is one in which \( f_Q \) take the value of 1 and 0 in its complement. (Figure 3a)

\[
L_{CE} = \{(2j-2+ y^m, 2i-2+x^i) | 1 \leq \mu \leq 3,1 \leq i \leq 4, 1 \leq j \leq 3\} = \{(0,0), (1,0), (2,0), (4,0), (5,0), (0,1), (3,1), (4,1), (0,2), (1,2), (2,2), (4,2), (0,3), (2,3), (5,3), (0,4), (2,4), (4,4), (0,5), (2,5), (4,5), (0,6), (2,6), (3,6), (4,6), (5,6), (1,7), (2,7), (4,7)\}
\]
Applying $f_w$ to the previous CA, is obtained the CAA that it shows in the figure 3b.

Now apply the algorithm to recovering patterns max and min from the CAA:

![Configuration of the CA of the example 5.3, in a) after to apply $f_Q$ and in b) after to apply $f_w$.](image)

**Figure 3.** Configuration of the CA of the example 5.3, in a) after to apply $f_Q$ and in b) after to apply $f_w$.

The table 1 shows the initial an the final value for each component of the output vector $y^l_1$ when is applied the max algorithm for pattern recovery. The first column shows the value por the variable $i$ considered the first for of the algorithm. the second column is the value of $y^l_1$ for default, which is 1. The third column are the different values for the cycle of the variable $j$, the fourth column is the respective value that has $x^l_j$, the fifth column is the condition that must comply in the algorithm depends if the value of $x^l_j$ is 0 or 1.

Finally the sixth column shows the final value of the $y^l_1$ component.

**Table 1.** Configuration of the CA of the example 5.3 in a) after to apply $f_Q$ and in b) after to apply $f_w$.

<table>
<thead>
<tr>
<th>(i)</th>
<th>(y^l_1)</th>
<th>(j)</th>
<th>(x^l_j)</th>
<th>condition</th>
<th>(\rightarrow y^l_1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>((2j - 2i - 2) = 1)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>0</td>
<td>((2j - 2i - 2)! = 1)</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>((2j - 2i - 2) = 1)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>0</td>
<td>((2j - 2i - 2)! = 1)</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>((2j - 2i - 2) = 1)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>0</td>
<td>((2j - 2i - 2)! = 1)</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>((2j - 2i - 2)! = 1)</td>
<td>1</td>
</tr>
</tbody>
</table>

From the table 1 we have $y^l_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$. So recover the pattern $y^l_1$ when is presented the input pattern $x^l$ using the max algorithm for pattern recovery.

Similarly to the above table, Table 2 shows the initial and the final value for each component of the output vector $y^l_1$ when is applied the min algorithm for pattern recovery.

**Table 2.** Configuration of the CA of the example 5.3 in a) after to apply $f_Q$ and in b) after to apply $f_w$.

<table>
<thead>
<tr>
<th>(i)</th>
<th>(y^l_1)</th>
<th>(j)</th>
<th>(x^l_j)</th>
<th>condition</th>
<th>(\rightarrow y^l_1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>((2j - 2i - 1) = 1)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>0</td>
<td>((2j - 2i - 1)! = 1)</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>((2j - 2i - 1) = 1)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>0</td>
<td>((2j - 2i - 1)! = 1)</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>((2j - 2i - 1) = 1)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>0</td>
<td>((2j - 2i - 1)! = 1)</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>((2j - 2i - 1)! = 1)</td>
<td>1</td>
</tr>
</tbody>
</table>
From the table 2 we have \( y^t = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \). So, recover the pattern \( y^t \) when is presented the pattern \( x^t \) using the \( \min \) algorithm for pattern recovery.

5. Experiments and results

For the experimental part was used the Iris database provided by the University of California Irvine Machine Learning Repository available in http://www.ics.uci.edu/~mlearn/mlrepository.html. The Iris database count with 150 instances, each instance with 4 real attributes without information loss, divided in 3 classes: Iris Setosa, Iris Versicolour and Iris Virgínica. To validate the test, it was considered the \( k \)-fold cross validation method with \( k = 10 \). The CAA was applied in its learning phase. For the recovering phase it was applied the \( \max \) algorithm and the \( \min \) algorithm. The figure 4 shows the CAA configuration in its learning phase, and the table 3 shows the result of the model compared with another results applied at the Iris Plant database.

![Figure 4. CAA configuration in its learning phase for the Iris Plant database.](image)

Table 3. Comparison of proposed model with other models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Iris Plant (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayesian Network (K2) [15]</td>
<td>93.20</td>
</tr>
<tr>
<td>Adaboost NB [15]</td>
<td>94.80</td>
</tr>
<tr>
<td>Bagging NB [15]</td>
<td>95.53</td>
</tr>
<tr>
<td>NBTree [15]</td>
<td>93.53</td>
</tr>
<tr>
<td>LogitBoostDS [15]</td>
<td>94.93</td>
</tr>
<tr>
<td>K-Means [16]</td>
<td>89</td>
</tr>
<tr>
<td>Neural Gas [16]</td>
<td>91.7</td>
</tr>
<tr>
<td>MLP [17]</td>
<td>95.99</td>
</tr>
<tr>
<td>NBT [17]</td>
<td>93.99</td>
</tr>
<tr>
<td>PART [17]</td>
<td>94.66</td>
</tr>
<tr>
<td>ACA with ( \max ) recuperação</td>
<td>99.33</td>
</tr>
<tr>
<td>ACA with ( \min ) recuperação</td>
<td>99.33</td>
</tr>
</tbody>
</table>

6. Conclusions

It has been presented a model of associative memory based on cellular automata that we call CAA. For the learning phase the CAA is builded from a fundamental set. For the recovering there is two algorithms: the \( \max \) and the \( \min \) recovering algorithms. The model was applied to the database of Iris Plant from the databases available in the repertory by the UCI Machine Learning Repository. The model was compared with other models by their showed yield.

7. References


