

# Autómatas celulares alfa-beta

Benjamín Luna Benoso y Cornelio Yáñez Márquez

Centro de Investigación en Computación, Instituto Politécnico Nacional, D.F.,  
México

mobius\_95@hotmail.com, coryanez@gmail.com

**Resumen.** Los autómatas celulares Alfa-Beta constituyen un puente conceptual entre dos áreas de investigación que han sido ajenas hasta el día de hoy: las memorias asociativas Alfa-Beta, por un lado, y los autómatas celulares, por el otro. Con los resultados de este artículo es posible aplicar las herramientas propias de las memorias asociativas Alfa-Beta en el ámbito de los autómatas celulares y, viceversa, se puede aplicar el bagaje teórico de los autómatas celulares en los algoritmos propios de las memorias asociativas Alfa-Beta. Al aplicar los autómatas celulares Alfa-Beta en la clasificación de dígitos escritos a mano tomados de la base de datos MNIST del NIST (National Institute of Standards and Technology), los resultados son competitivos al compararlos con otros algoritmos.

**Palabras Clave.** Autómatas celulares, reconocimiento de patrones, memorias asociativas Alfa-Beta.

## Alpha-Beta Cellular Automata

**Abstract.** Alpha-Beta Cellular Automata arise as a conceptual bridge between two research areas which have remained disjoint to this day: Alpha-Beta associative memories on one hand and cellular automata on the other hand. The results presented in this work make it possible to apply tools developed on the basis of the Alpha-Beta associative models to the field of cellular automata and vice versa: one can use the theoretical body of cellular automata in applications of the Alpha-Beta associative models. Specifically, the proposed model is applied to the task of classification on the MNIST database of handwritten digits as published by the US National Institute of Standards and Technology (NIST). The results are competitive when compared to those given by other algorithms applied to the same database.

**Keywords.** Cellular automata, pattern recognition, Alpha-Beta associative memories.

## 1 Introducción

El concepto de autómatas celulares fue introducido en 1951 por John von Neumann [5, 17], quien lo define como un espacio capaz de autoreproducirse. Los autómatas celulares son modelos matemáticos en los que el comportamiento de cada uno de los elementos que componen el sistema depende de la interacción local entre todos los elementos. Un autómata celular de dimensión  $d$  consiste de un enrejado (o lattice)  $d$ -dimensional que se extiende infinitamente y que representa el espacio, donde cada sitio de la lattice es llamada célula y tiene asociado una variable de estado que fluctúa sobre un conjunto finito, el cual es denominado conjunto de estados. En un autómata celular se considera que el tiempo avanza en etapas discretas y la dinámica es dada por una regla explícita llamada función local, la cual es usada en cada etapa de tiempo por cada célula para determinar su nuevo estado a partir del estado actual de ciertas células en su vecindad. Las células alteran sus estados sincrónicamente en etapas de tiempo discreto de acuerdo con la función local. La lattice es homogénea por lo que todas las células operan bajo la misma función local. La asignación de estados a todas las células en la lattice es llamada una configuración, la cual es considerada como el estado de la lattice total. Es preciso hacer notar que, desde su creación, los autómatas celulares han tenido una variedad de aplicaciones en diferentes disciplinas de las ciencias y la ingeniería [11, 16, 19, 20, 26].

Ajeno al campo de los autómatas celulares, se tiene el desarrollo y estudio del reconocimiento de patrones, y una problemática de esta área se refiere a la clasificación de patrones [6, 22, 28]. El objetivo de la clasificación consiste en

particionar el espacio de características para formar regiones, a las cuales se les asigna una categoría o clase. Los diferentes patrones deben de ser asignados en alguna de las regiones creadas en el espacio de características y, en general, la descripción completa de las clases es desconocida; no obstante ello, se tiene un conjunto finito y reducido de patrones que proveen información parcial sobre un problema específico.

En la actualidad hay varios algoritmos con los que se intenta resolver el problema de la clasificación de patrones; entre los más utilizados se pueden mencionar: el clasificador  $k$ -NN [3, 15, 29], el clasificador euclidiano [13, 23], el clasificador lineal (1-layer NN) [4, 10], las redes neuronales, las máquinas de soporte vectorial (SVM) y los árboles de decisión [6].

Por otra parte, se tiene el desarrollo de las memorias asociativas, modelos que han estado vigentes desde la década de los 60 [24]. El propósito fundamental de una memoria asociativa es recuperar correctamente patrones completos a partir de patrones de entrada, los cuales pueden estar alterados con ruido aditivo, sustractivo o combinado [1]. Recientemente, algunos modelos de memorias asociativas se han aplicado en la clasificación de patrones [21, 27].

Varios investigadores han abordado el problema de desarrollar modelos de memorias asociativas [2, 8, 9, 21, 24, 25] y han logrado resultados de suma importancia para dicho campo de investigación.

En este artículo se presenta, por primera vez, un modelo asociativo Alfa-Beta implementado mediante la teoría de los autómatas celulares. Este modelo celular Alfa-Beta establece un puente conceptual entre los modelos asociativos Alfa-Beta y los autómatas celulares, con lo cual es posible aplicar las herramientas propias de los modelos asociativos Alfa-Beta en el ámbito de los autómatas celulares y, viceversa, aplicar el bagaje teórico de los autómatas celulares en aplicaciones de los modelos asociativos Alfa-Beta. Específicamente, en este artículo se aplica el modelo propuesto a la tarea de clasificación de dígitos escritos a mano tomados de la base de datos MNIST del NIST (National Institute of Standards and Technology).

El resto del artículo está organizado como sigue: la sección 2 incluye los conceptos básicos de memorias asociativas y de los autómatas celulares, cuyas definiciones y conceptos sirven de base para exponer, en la sección 3, el modelo original propuesto en este artículo. En la sección 4 se describen los experimentos y se discuten los resultados, mientras que en la sección 5 se presentan las conclusiones de este trabajo.

Finalmente, se incluyen los agradecimientos y las referencias.

## 2 Materiales y método

### 2.1 Memorias asociativas

Una memoria asociativa puede formularse como un sistema de entrada y salida el cual se divide en dos fases [1, 27]:

Fase de aprendizaje:  $x \rightarrow [M] \leftarrow y$   
(generación de la memoria asociativa)

Fase de recuperación:  $x \rightarrow [M] \rightarrow y$   
(operación de la memoria asociativa)

El patrón de entrada está representado por un vector columna denotado por  $x$  y el patrón de salida, por un vector columna denotado por  $y$ . Cada uno de los patrones de entrada forma una asociación con el correspondiente patrón de salida. La notación para una asociación es similar a la de una pareja ordenada; por ejemplo, los patrones  $x$  y  $y$  del esquema forman la asociación  $(x, y)$ .

La memoria asociativa  $M$  se representa mediante una matriz cuya  $ij$ -ésima componente es  $m_{ij}$  [18]; y la matriz  $M$  se genera a partir de un conjunto finito de asociaciones conocidas de antemano, llamado *conjunto fundamental*.

Considérese  $p \in \mathbb{N}$ . El conjunto fundamental se representa de la siguiente manera:

$$\{(x^\mu, y^\mu) \mid \mu = 1, 2, \dots, p\} \quad (1)$$

A los patrones que conforman las asociaciones del conjunto fundamental, se les llama patrones fundamentales.

Si bien el objetivo de las memorias asociativas es el reconocimiento de patrones, éstas se pueden utilizar como un clasificador de patrones de la siguiente manera: Sea  $N \in \mathbb{N}$ , si  $D = \{(x^i, c^i) \mid i = 1, 2, \dots, N\}$  es el conjunto de aprendizaje con  $x^i$  que pertenece a la clase  $c^i$ , entonces el conjunto fundamental se define como

$$\{(x^\mu, x^\mu) \mid \mu = 1, 2, \dots, p\} \quad (2)$$

Las memorias que cumplen con la condición 2 son llamadas memorias autoasociativas; en caso contrario, se denominan memorias heteroasociativas. Si  $\tilde{x}$  es un patrón de prueba que se le presenta a la memoria asociativa y ésta recupera el patrón  $x^i$  para algún  $i = 1, 2, \dots, p$ , entonces a  $\tilde{x}$  se le asigna la clase  $c^i$ .

## 2.2 Autómatas celulares

Las siguientes definiciones son tomadas de [14].

Sea  $I$  un conjunto de índices, y sea  $A = \{[a_i, b_i]\}_{i \in I}$  una familia contable de intervalos cerrados en  $\mathbb{R}$  tales que cumplen las siguientes condiciones:

$$\bigcup_{X \in A} X = [a, b] \text{ para algún } a, b \in \mathbb{R} \text{ o bien}$$

$$\bigcup_{X \in A} X = \mathbb{R}.$$

Si  $[a_i, b_i] \in A$  entonces  $b_i - a_i > 0$ .

Si  $[a_i, b_i]$  y  $[c_j, d_j]$  están en  $A$  con  $b_i \leq c_j$ , entonces ocurre una de estas dos situaciones: o sucede que  $[a_i, b_i] \cap [c_j, d_j] = \emptyset$  o bien  $[a_i, b_i] \cap [c_j, d_j] = b_i = c_j$ .

**Definición 2.1.** Sea  $[a, b]$  un intervalo en  $\mathbb{R}$  con  $a \neq b$  y sea  $A$  una familia de intervalos cerrados que cumplen las condiciones 1, 2 y 3 mencionadas en el párrafo anterior. Una *lattice* de dimensión 1, también llamada *lattice 1-dimensional* es el conjunto  $\mathcal{L}$

$= \{x_i \times [a, b] \mid x_i \in A\}$ . Si  $A_1, A_2, \dots, A_n$  son familias de intervalos que cumplen las condiciones 1, 2 y 3 mencionadas en el párrafo anterior, entonces una *lattice* de dimensión  $n > 1$  es el conjunto  $\mathcal{L} = \{x_1 \times x_2 \times \dots \times x_n \mid x_i \in A_i\}$ .

**Definición 2.2.** Sea  $r \in \mathbb{R}$ . Una *lattice 1-dimensional* es *regular* si  $b_i - a_i = r$  para cada  $[a_i, b_i] \in A$ . Una *lattice  $n$ -dimensional* es *regular* si  $b_{i_k} - a_{i_k} = r$  para cada  $[a_{i_k}, b_{i_k}] \in A_i$  con  $i = 1, \dots, n$ .

**Definición 2.3.** Sea  $\mathcal{L}$  una *lattice  $n$ -dimensional*. Una *célula, celda o sitio* es un elemento de  $\mathcal{L}$ . Esto es, una célula es un elemento de la forma  $[a_{1_k}, b_{1_k}] \times \dots \times [a_{n_k}, b_{n_k}]$  con  $[a_{i_k}, b_{i_k}] \in A_i$  para  $i = 1, \dots, n$ .

**Definición 2.4.** Sea  $\mathcal{L}$  una *lattice*, y sea  $c$  una célula de  $\mathcal{L}$ . Una *vecindad* de tamaño  $n \in \mathbb{N}$  para  $c$ , es el conjunto  $v(c) = \{k_1, k_2, \dots, k_n \mid k_j \text{ es célula de } \mathcal{L} \text{ con } d(c, k_j) \leq 2 \text{ para cada } j = 1, \dots, n\}$

**Definición 2.5.** Sea  $n \in \mathbb{N}$ . Un *Autómata Celular (AC)* es una tupla  $(\mathcal{L}, S, \mathcal{N}, f)$  tal que:

$\mathcal{L}$  es una *lattice regular*.

$S$  es un conjunto finito de estados.

$\mathcal{N}$  es un conjunto de *vecindades* definido de la siguiente manera:

$\mathcal{N} = \{v(c) \mid c \text{ es una célula y } v(c) \text{ es una vecindad de } c \text{ de tamaño } n\}$ .

$f : \mathcal{N} \rightarrow S$  es una función llamada *función de transición*.

**Definición 2.6.** Una *configuración* del AC  $\mathcal{Q} = (\mathcal{L}, S, \mathcal{N}, f)$  es una función  $C_t : \mathcal{L} \rightarrow S$  que asocia a cada célula de la *lattice*  $\mathcal{L}$  (al tiempo  $t$ ) un estado de  $S$ .

Si  $Q=(\mathcal{L},S,\mathcal{N},f)$  es un AC y  $c \in \mathcal{L}$  es una célula, entonces la configuración  $C_t$  está relacionada con  $f$  mediante:

$$C_{t+1}(c) = f(\{C_t(i) \mid i \in \mathcal{N}(c)\}).$$

**Definición 2.7.** Sean  $Q=(\mathcal{L},S,\mathcal{N},f)$  y  $W=(\mathcal{L},S,\mathcal{N}',g)$  dos AC. Se define el AC composición de los AC  $Q$  y  $W$  en el tiempo  $t = t_0$  denotada como  $W*Q$  mediante el AC  $W*Q=(\mathcal{L},S,\mathcal{N},h)$  donde  $h, f$  y  $g$  están relacionadas de la siguiente manera:

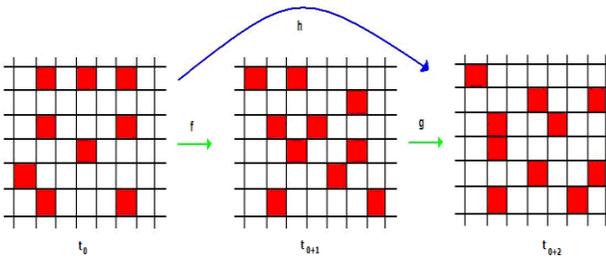


Fig. 1. Ejemplo de un AC composición

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

$$C_{t_0+2}(c) = g(\{C_{t_0+1}(i) \mid i \in \mathcal{N}'(c)\})$$

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

### 3 Modelo propuesto

En esta sección se construirá la memoria asociativa Alfa-Beta mediante Autómatas Celulares: lo llamaremos modelo celular Alfa-Beta.

En lo que sigue, consideremos dos conjuntos:

$A = \{0,1\}$  y el conjunto fundamental

$CF = \{(x^\mu, y^\mu) \mid \mu = 1, 2, \dots, p\}$  con  $x^\mu \in A^n$  y  $y^\mu \in A^m$ , donde  $n, m \in \mathbb{N}$ .

La lattice  $\mathcal{L}$  para el autómata celular estará compuesta por la matriz de tamaño  $2m \times 2n$  cuyo primer índice es la pareja  $(0,0)$ .

El conjunto  $S = \{0,1\}$  es el conjunto finito de estados.

Sean  $I = \{i \in \mathbb{Z} \mid i = 2k \text{ para algún } k = 0, 1, 2, \dots, n-1\} = \{0, 2, 4, \dots, 2(n)-2\}$  y  $J = \{j \in \mathbb{Z} \mid j = 2k+1 \text{ para algún } k = 0, 1, 2, \dots, m-1\} = \{1, 3, 5, \dots, 2m-1\}$ .

Consideremos la partición de  $\mathcal{L}$  formada por la familia de subconjuntos

$$IJ = \{v_{(i,j)} \mid (i,j) \in I \times J\} \text{ con}$$

$$v_{(i,j)} = \{(i,j), (i,j-1), (i+1,j), (i+1,j-1)\}.$$

Puesto que  $IJ$  es una partición de  $\mathcal{L}$ , dado  $l \in \mathcal{L}$  existe un único  $(i,j) \in I \times J$  tal que  $l \in v_{(i,j)}$ .

Denotaremos mediante  $v^l$  a este elemento único de  $IJ$ , esto es,  $v^l = v_{(i,j)}$ . Por ejemplo, si

$l = (3,0)$ , entonces

$$l \in v^{(3,0)} = v_{(2,1)} = \{(2,1), (2,0), (3,1), (3,0)\}.$$

A partir del hecho anterior se define el conjunto de vecindades

$$\mathcal{N} = \{v^l \mid l \in \mathcal{L}\} \tag{3}$$

**Definición 3.1.** Consideremos el conjunto  $A^k$ . Se define la función de proyección de la  $i$ -ésima componente ( $1 \leq i \leq k$ )  $\text{Pr}_i : A^k \rightarrow A$  como

$$\text{Pr}_i(z) = z_i, \text{ con } z = (z_1, z_2, \dots, z_k) \tag{4}$$

#### Teorema 3.1.

Si

$$(y_i, x_j) \in \text{Pr}_{yx} = \{(y_i, x_j) \mid y_i = \text{Pr}_i(y) \wedge x_j = \text{Pr}_j(x)\}$$

entonces

$$(2j-2+y_i, 2i-2+x_j) \in v_{(2j-2, 2i-1)}.$$

Demostración.- Se tiene que

$$v_{(2j-2, 2i-1)} = \{(2j-2, 2i-1), (2j-2, 2i-2),$$

$$(2j-1, 2i-1), (2j-1, 2i-2)\}.$$

Puesto que  $(y_i, x_j) \in \text{Pr}_{yx}$ , entonces  $y_i = \text{Pr}_i(y)$  y

$x_j = \text{Pr}_j(x)$ , y puesto que  $x \in A^n$  y  $y \in A^m$ , entonces  $y_i, x_j \in \{0,1\}$ , luego

Si  $y_i = x_j = 0$ , entonces

$$(2j-2 + y_i, 2i-2 + x_j) = (2j-2, 2i-2) \in v_{(2j-2, 2i-1)}$$

2. Si  $y_i = 0$  y  $x_j = 1$ , entonces

$$(2j-2 + y_i, 2i-2 + x_j) = (2j-2, 2i-1) \in v_{(2j-2, 2i-1)}$$

3. Si  $y_i = 1$  y  $x_j = 0$ , entonces

$$(2j-2 + y_i, 2i-2 + x_j) = (2j-1, 2i-2) \in v_{(2j-2, 2i-1)}$$

4. Si  $y_i = x_j = 1$ , entonces

$$(2j-2 + y_i, 2i-2 + x_j) = (2j-1, 2i-1) \in v_{(2j-2, 2i-1)}$$

Se define el conjunto

$$L_{CF} = \{(2j-2 + y_i^\mu, 2i-2 + x_j^\mu) \mid 1 \leq \mu \leq p, 1 \leq i \leq m \wedge 1 \leq j \leq n\} \subseteq \mathcal{L}.$$

A continuación se plantea el modelo de la memoria asociativa en su fase de aprendizaje por medio de autómatas celulares.

Consideremos los autómatas celulares

$$Q = (\mathcal{L}, S, \mathcal{N}, f_Q) \text{ y } W = (\mathcal{L}, S, \mathcal{N}', f_W) \text{ con } \mathcal{N}' = IJ, \text{ y}$$

$f_Q: \mathcal{N} \rightarrow S$ ,  $f_W: \mathcal{N}' \rightarrow S$  definidas como sigue:

$$f_Q(v^{(i,j)}) = \begin{cases} 1 & \text{si } (i, j) \in L_{CF} \\ 0 & \text{si } (i, j) \notin L_{CF} \end{cases}$$

$$f_W(v^{(i,j)}) = \begin{cases} 1 & \text{en la posición } (i+1, j) \text{ si } (i, j-1) = 1 \\ 1 & \text{en la posición } (i, j-1) \text{ si } (i+1, j) = 1 \end{cases}$$

Se define el Autómata Celular Asociativo en su fase de aprendizaje como

$$W^* Q = (\mathcal{L}, S, \mathcal{N}, f_A) \quad (5)$$

Para la fase de recuperación, se tiene el siguiente algoritmo.

#### Algoritmo AC asociativo para recuperación de patrones

Entrada: Patrón a reconocer  $\tilde{x} = \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{pmatrix}$

Salida: Patrón recuperado  $\tilde{y} = \begin{pmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_m \end{pmatrix}$

Proceso:

```

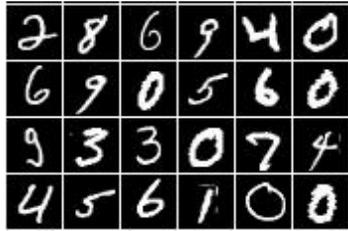
for i = 1, 2, ..., m
   $\tilde{y}_i = 1$ 
  for j = 1, 2, ..., n
    if
       $\tilde{x}_j = 0 \wedge (2j-1, 2i-2) = 1$ 
        continue
    if
       $\tilde{x}_j = 1 \wedge ((2j-2, 2i-2) = 1 \vee (2j-1, 2i-2) = 1)$ 
        continue
    else
       $\tilde{y}_i = 0$ 
      break
  fin segundo for
fin primer for

```

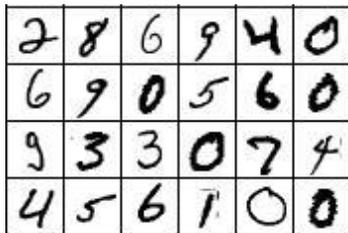
## 4 Experimentos y resultados

La base de datos MNIST de dígitos escritos a mano es un subconjunto de imágenes tomadas del NIST (National Institute of Standards and Technology) [12]. Así, la base de datos MNIST fue construida a partir de las bases de datos especiales 1 y 3 del NIST, que contiene imágenes de dígitos escritos a mano. Varios grupos de investigación han utilizado la base de datos MNIST para evaluar el desempeño de sus algoritmos de aprendizaje [12], permitiendo hacer una comparación con los resultados presentados en este trabajo.

La base de datos MNIST consiste de un conjunto de 70,000 imágenes agrupadas en 10 clases diferentes, de las cuales 60,000 instancias



**Fig. 2.** Base de datos MNIST de dígitos escritos a mano



**Fig. 3.** Negado de las imágenes de la base de datos MNIST de dígitos escritos a mano

son utilizadas para el entrenamiento del algoritmo y las 10,000 restantes para prueba. Cada una de las instancias de la base de datos se encuentra normalizada y centrada dentro de una imagen en tono de gris de  $28 \times 28$  píxeles (figura 2).

No obstante que en la base de datos original los tonos más oscuros se acercan al cero y los tonos más claros al valor 255, en los experimentos de este trabajo se consideraron las imágenes negadas de las originales de modo que el fondo quedó blanco y el carácter quedó en un tono cercano al negro (figura 3).

En virtud de que el modelo propuesto requiere patrones binarios para su funcionamiento tanto en la fase de aprendizaje como de recuperación, y dado que los patrones de la base de datos MNIST son imágenes en escala de grises, se requiere realizar una codificación de los valores de cada píxel, para obtener las cadenas binarias correspondientes.

Los experimentos se realizaron haciendo uso del código Johnson-Möbius modificado [7], ya que con este código se preserva el tipo de ruido que exhiben los patrones de prueba respecto de los patrones del conjunto fundamental.

No obstante que existen otras opciones para codificar en binario los patrones, como por ejemplo el código binario usual y los códigos Gray, estas codificaciones alteran el tipo de ruido en los patrones, situación que no es deseable en las pruebas de rendimiento del modelo, porque al alterarse el tipo de ruido original, se alteran los resultados de la operación del modelo.

La tabla 1 muestra los resultados obtenidos de un estudio comparativo al considerar el rendimiento que exhibió el modelo celular Alfa-Beta, en comparación con los rendimientos que obtuvieron otros modelos cuyos resultados experimentales están publicados en [12].

Los protocolos experimentales son similares a los reportados en los trabajos de investigación que prueban sus modelos teóricos en esta misma base de datos MNIST. De las 70,000 imágenes disponibles, se utilizan 60,000 en la fase de aprendizaje y, al entrar en la fase de clasificación de patrones, el algoritmo se prueba con las 10,000 imágenes restantes. Se cuantifica entonces el número de patrones clasificados correctamente por el modelo, y se calcula el rendimiento del modelo como el porcentaje de patrones clasificados correctamente respecto de 10,000 que es la cardinalidad del conjunto de prueba. El error reportado es simplemente la diferencia entre 100% y el rendimiento correspondiente: los mejores modelos serán los que exhiban errores menores. Nuestro modelo celular Alfa-Beta exhibió un porcentaje de rendimiento del 99.48%; es decir, un error de 0.52%.

En general, dado un modelo clasificador de patrones, existen criterios para evaluar su rendimiento, como por ejemplo el leave-one-out, el llamado hold-out y el más utilizado en la literatura científica, el k-fold cross validation. Sin embargo, esta base de datos del MNIST es muy específica en su estructura, dado que de las 70,000 imágenes, 60,000 de ellas, fijas, son para la fase de aprendizaje y las restantes 10,000, también fijas, son reservadas para la prueba en la fase de clasificación, y de ahí surgen los porcentajes de rendimiento y error de modelo.

Decidimos usar el criterio estándar para validación del modelo, con objeto de que el estudio comparativo sea significativo. Para poder comparar con los rendimientos y errores

Tabla 1. Tabla comparativa

Algoritmo	Preprocesamiento	Error	Año
linear classifier (1-layer NN)	Ninguno	12.0	1998
2-layer NN, 300 HU, MSE, [distortions]	Ninguno	3.6	1998
linear classifier (1-layer NN)	Deskewing	8.4	1998
k-NN, Euclidean (L2)	Ninguno	3.09	2007
unsupervised sparse features + SVM, (no distortions)	Ninguno	0.59	2008
large conv. net, unsuppretraining (no distortions)	Ninguno	0.53	2009
<b>AC Alfa-Beta</b>	<b>Ninguno</b>	<b>0.52</b>	<b>2010</b>

publicados en [12] para otros modelos teóricos, es muy importante usar el mismo criterio de validación, razón por la cual escogimos este, y no otro.

Vale la pena enfatizar el mayor rendimiento (y por tanto menor error) exhibido por el modelo celular Alfa-Beta, en relación con otros modelos aplicados en la clasificación de dígitos escritos a mano de la base de datos MNIST. Esto, a pesar de que entre los modelos incluidos en la tabla aparecen algoritmos muy populares y de gran uso entre los investigadores dedicados a clasificación de patrones. La razón es simple: mientras que algunos conocidos modelos como las redes neuronales (NN) o las máquinas de soporte vectorial (SVM) incluyen complejas operaciones en ambas fases (aprendizaje y clasificación de patrones), el modelo celular Alfa-Beta tiende a lo simple, y por ello su operación es más eficaz, al tiempo que es más eficiente.

## 5 Conclusiones

Se ha presentado un nuevo modelo de memoria asociativa Alfa-Beta implementado con autómatas celulares al que hemos llamado modelo celular Alfa-Beta. Este modelo propuesto

se aplicó al problema de clasificación de caracteres de la base de datos MNIST obtenido en [12]. Se mostró una tabla comparativa de los resultados obtenidos (error de clasificación) en relación con los resultados exhibidos por algunos de los modelos de clasificación que aparecen en [12].

Se observa en la tabla 1 que al utilizar el código Johnson-Möbius modificado para binarizar los patrones, el modelo celular Alfa-Beta arrojó un error del 0.52%, superando así a otros conocidos modelos en eficacia y eficiencia.

El modelo celular Alfa-Beta hereda algunos de los problemas exhibidos por las memorias asociativas Alfa-Beta, como lo son la sensibilidad a cierto tipo de ruido, problema que se ataca con el uso del código Johnson-Möbius modificado, dado que esta codificación conserva el tipo de ruido original, sin alterarlo, como ocurre con otros esquemas de clasificación como los códigos Gray, por ejemplo.

Hasta la fecha, no se han observado limitaciones del modelo celular Alfa-Beta, relacionadas con el tipo de patrones con los que se prueba; es decir, es posible usar este modelo en cualquier banco de datos que incluya patrones que se puedan binarizar con el código Johnson-Möbius modificado.

Un posible trabajo a futuro consiste en buscar un código de binarización alternativo al código Johnson-Möbius modificado, ya que una de sus desventajas es que cuando los datos dentro de un patrón son grandes, el número de dígitos binarios que se requieren crece demasiado, lo cual demanda recursos computacionales de almacenamiento.

## Agradecimientos

Los autores agradecen al Instituto Politécnico Nacional (Secretaría Académica, COFAA, SIP, ESCOM y CIC), al CONACYT, y SNI por su apoyo económico para el desarrollo de este trabajo. Específicamente al Proyecto SIP-20110661.

## Referencias

1. **Acevedo-Mosqueda, M.E., Yáñez-Márquez, C., & López-Yáñez, I. (2007).** Alpha-Beta Bidirectional Associative Memories: Theory and Applications. *Neural Processing Letters*, 26(1), 1–40.
2. **Amari, S.I. (1972).** Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Transactions on Computers*, C-21(11), 197–206.
3. **Baffetta, F., Fattorini, L., Franceschi, S., & Corona, P. (2009).** Design-based approach to k-nearest neighbours technique for coupling field and remotely sensed data in forest surveys. *Remote Sensing of Environment*, 113(3), 463–475.
4. **Chang, Y.C., Chen, S.M., & Liaw, C.J. (2008).** Multilabel text categorization based on a new linear classifier learning method and a category-sensitive refinement method. *Expert Systems with Applications: An International Journal*, 34 (3), 1948–1953.
5. **Chen, F.F. & Chen, F.Y. (2009).** Complex dynamics of cellular automata rule 119. *Physica A: Statistical Mechanics and its Applications*, 388(6), 984–990.
6. **Duda, R.O., Hart P.E., & Stork, D.G. (2001).** *Pattern Classification* (2<sup>nd</sup> Ed.). New York: Wiley.
7. **Flores, R. (2006).** *Memorias Asociativas alfa-beta basadas en el código Johnson-Mobiüs Modificado*. Tesis de maestría, Centro de Investigación en Computación del Instituto Politécnico Nacional. México, D.F.
8. **Hopfield, J.J. (1982).** Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8), 2554–2558.
9. **Kosko, B. (1988).** Bidirectional associative memories. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1), 49–60.
10. **Kostin, A. (2006).** A simple and fast multi-class piecewise linear pattern classifier. *Pattern Recognition*, 39(11), 1949–1962.
11. **Krane, M.J., Johnson, D.R., & Raghavan, S. (2009).** The development of a cellular automaton-finite volume model for dendritic growth. *Applied Mathematical Modelling*, 33(5), 2234–2247.
12. **LeCun, Y. & Cortes, C. (s.f.).** The MNIST database of handwritten digits. Retrieved from <http://yann.lecun.com/exdb/mnist/>.
13. **Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998).** Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.
14. **Luna-Benoso, B., Yáñez-Márquez, C., Figueroa-Nazuno, J., & López-Yáñez, I. (2007).** Cellular Mathematical Morphology. Sixth Mexican International Conference on Artificial Intelligence-Special Session (MICAI 2007), Aguascalientes, México, 105–112.
15. **Magnussen, S., McRoberts, R.E., & Tomppo, E.O. (2009).** Model-based mean square error estimators for k-nearest neighbour predictions and applications using remotely sensed data for forest inventories. *Remote Sensing of Environment*, 113(3), 476–488.
16. **Nagatani, T. (2009).** Traffic states and fundamental diagram in cellular automaton model of vehicular traffic controlled by signals. *Physica A: Statistical Mechanics and its Applications*, 388(8), 1673–1681.
17. **Neumann, J.V. (1966).** *Theory of self-reproducing automata*. Urbana: University of Illinois Press.
18. **Markert, H., Kaufmann, U., Kayikci, Z.K., & Palm, G., (2009).** Neural associative memories for the integration of language, vision and action in an autonomous agent. *Neural Networks*, 22(2), 134–143.
19. **Reis, E.A, Santos, L.B.L., & Pinho, S.T.R. (2009).** A cellular automata model for avascular solid tumor growth under the effect of therapy. *Physica A: Statistical Mechanics and its Applications*, 388(7), 1303–1314.
20. **Richard, M., Kirkby, K.J., Webb, R.P., & Kirkby, N.F. (2009).** Cellular automaton model of cell response to targeted radiation. *Applied Radiation and Isotopes*, 67(3), 443–446.
21. **Ritter, G.X., Sussner, P., & Díaz-de-León, J.L. (1998).** Morphological associative memories. *IEEE Transactions on Neural Networks*, 9(2), 281–293.
22. **Ryabko, D. (2006).** Pattern Recognition for Conditionally Independent Data. *The Journal of Machine Learning Research*, 7(Apr), 645–664.
23. **Sohn, S.Y. & Shin, H.W. (2007).** Experimental study for the comparison of classifier combination methods. *Pattern Recognition*, 40(1), 33–40.
24. **Steinbuch, K. (1961).** Die Lernmatrix. *Biological Cybernetics*, 1(1), 36–45.
25. **Willshaw, D.J., Buneman, O.P., & Longuet-Higgins, H.C. (1969).** Non-holographic associative memory. *Nature*, 222, 960–962.
26. **Wolfram, S. (1983).** Statistical mechanics of previous term cellular automata. *Reviews of Modern Physics*, 55(3), 601–644.
27. **Yáñez-Márquez, C., Sánchez-Fernández, L.P., & López-Yáñez, I. (2006).** Alpha-Beta Associative Memories for Gray Level Patterns.

*Third International Symposium on Neural Networks 2006, Lecture Notes in Computer Science, 3971, 818–823.*

28. **Zhang, B., Zhang, Y., & Begg, R.K. (2009).** Gait classification in children with cerebral palsy by Bayesian approach. *Pattern Recognition*, 42(4), 581–586.
29. **Zhao, Y. & Teng, S. (2009).** Combinatorial and spectral aspects of nearest neighbor graphs in doubling dimensional and nearly-Euclidean spaces. *Theoretical Computer Science*, 410(11), 1081–1092.



**M. en C. Benjamín Luna Benoso.**

Nacionalidad Mexicana. Lic. en Física y Matemáticas con especialidad en Matemáticas Puras (ESFM-IPN). Maestría en Ciencias de la Computación (CIC-IPN).  
Temas de interés: Memorias

Asociativas, Reconocimiento de Patrones, Morfología Matemática, Autómatas Celulares.



**Dr. Cornelio Yáñez Márquez.**

Nacionalidad Mexicana. Licenciado en Física y Matemáticas (1989) por la ESFM-IPN. Grados de maestro en ciencias (1995) en ingeniería de cómputo y de doctor en ciencias de la

computación (2002), obtenidos en el CIC-IPN. Profesor investigador titular C del CIC-IPN. Presea Lázaro Cárdenas 2002, recibida de manos del C. Presidente de la República. Miembro del Sistema Nacional de Investigadores. Áreas de Interés: Memorias Asociativas, Redes Neuronales, Morfología Matemática e Ingeniería de Software.

*Artículo recibido el 04/10/2011; aceptado el 07/11/2011.*