

# Memorias asociativas en la agricultura: un caso práctico

Mario Aldape Pérez, José Antonio Estrada Pavía, Joel Omar Juarez Gambino

Centro de Innovación y Desarrollo Tecnológico en Cómputo (CIDETEC),  
Instituto Politécnico Nacional (IPN),  
Ciudad de México, México

mario@aldape.org.mx; jaestrada@gmail.com; omarjg82@gmail.com

**Resumen.** Las memorias asociativas tienen una serie de características, incluyendo un rápido y eficiente método de clasificación, así como una tolerancia intrínseca al ruido que las hace ideales para gran variedad de aplicaciones. En este artículo se utilizarán las memorias alfa-beta autoasociativas con el propósito de recomendar un herbicida en base a diferentes parámetros. El objetivo de este trabajo es presentar las memorias alfa-beta como una posibilidad para resolver problemas reales en la agronomía.

**Palabras clave:** Memorias asociativas Alfa-Beta, herbicidas, reconocimiento de patrones.

## 1. Introducción

Los sistemas agrícolas relacionados con herbicidas son complejos, en algunas ocasiones pueden ser considerados como sistemas mal definidos ya que es difícil, en este tipo de sistemas, cuantificar las relaciones entre la entrada y la salida, esta basta cantidad de relaciones existentes se debe a que la mayor parte de sistemas agrícolas involucran gran cantidad de variables diferentes. Desde 1997 los sistemas de control inteligente han sido de las tecnologías más prosperas en el campo de los sistemas complejos [1]. A continuación se muestra una pequeña reseña de la evolución de métodos de Inteligencia Artificial (IA) usados en aplicaciones agronómicas.

En 1997 las redes neuronales artificiales (ANN) demostraron ser una alternativa eficaz para su utilización en los sistemas complejos, sus principales características son su alta capacidad de aprendizaje y su capacidad de identificar y modelar una compleja relación no lineal entre la entrada y la salida del sistema [1].

Por otro lado en 2001, un nuevo método de clasificación de redes neuronales fue presentado en este campo Self-Organizing Map (SOM), este método logra una convergencia rápida y buena generalización. La clasificación del método propuesto se probó superior, en aplicaciones agrícolas, comparado con otros clasificadores estadísticos y neurológicos [2].

La ciencia sigue cambiando y avanzando día con día, también nuevos métodos se desarrollan en el campo de la inteligencia artificial aplicada en la agricultura,

uno de estos métodos fue una red neuronal auto organizada, este método presentado en 2003 probó que las redes neuronales pueden discriminar entre diferentes especies de plantas con una precisión superior al 75 %, sin que se halla definido anteriormente las características de la planta.[3].

No sólo las ANN se han utilizado en los sistemas agrícolas, también las Máquinas de Soporte Vectorial (SVM) han demostrado ser un método útil en la IA, en 2006 las SVM demostraron que pueden ser usadas como una herramienta para la clasificación de imágenes hiperespectrales. Este método logró más de un 93 % de eficiencia clasificando datos de prueba que no habían sido enseñados [4].

Más recientemente, en 2011 el Generalized Softmax Perceptron (GSP) una arquitectura de red neuronal, se usó en conjunto con el Posterior Probability Model Selection (PPMS) en una selección compleja para clasificar las imágenes en girasoles o no girasoles [5].

Algunos autores describen las memorias asociativas como redes neuronales sin pesos, pero las memorias asociativas tienen otras cualidades que las hacen perfectas para una gran cantidad de problemas. El propósito fundamental de una memoria asociativa es recuperar patrones completos a partir de patrones de entrada, que pueden contener ruido aditivo, sustractivo o mixto [6]. Las memorias asociativas tiene dos fases funcionales, la fase de aprendizaje y la fase de recuperación, en la fase de aprendizaje la memoria asociativa es construida con un conjunto de patrones clasificados un patrón de entrada y otro un patrón de salida.

Una de las referencias más antigua de memorias asociativas data de 1961 cuando Karl Steinbuch desarrolló el modelo de memoria asociativa Lernmatrix [7] una memoria heteroasociativa capaz de trabajar como un clasificador de patrones binarios. Luego, en 1972, el modelo Linear Associator se presentó de forma independiente por Anderson [8] y Kohonen [9]. Pero el trabajo más relevante de la memoria asociativa fue desarrollado por Hopfield en 1982 [10] su modelo demuestra la interacción de elementos simples de procesamiento, similares a las neuronas, dando lugar a la aparición de propiedades computacionales colectivas, como la estabilidad de las memorias [6] sin embargo, este modelo tuvo inconvenientes uno de ellos es la limitada capacidad de recuperación siendo esta de sólo  $0.015 n$ , donde  $n$  es la dimensión de los patrones almacenados [6].

El modelo de memoria asociativa usado en este documento se llama memoria alfa-beta, la mayor ventaja de esta memoria es la recuperación completa del conjunto fundamental.

## 2. Memorias asociativas Alfa-Beta

El propósito de la memoria asociativa es la correcta recuperación de patrones de salida relacionados con patrones de entrada, que pueden ser alterados con ruido aditivo, sustractivo o mezclado. Los conceptos que se utilizan en esta sección se presentan en [6].

**Tabla 1.** Operadores Alfa Beta.

$\alpha : A \times A \rightarrow B$	
x\y	$\alpha(x,y)$
0\0	1
0\1	0
1\0	2
1\1	1

$\beta : B \times A \rightarrow A$	
x\y	$\beta(x,y)$
0\0	0
0\1	0
1\0	0
1\1	1
2\0	1
2\1	1

Una memoria asociativa  $M$  es un sistema que relaciona un patrones de entrada y salida de la siguiente forma:  $x \rightarrow \boxed{\mathbf{M}} \rightarrow y$  con  $x$  e  $y$ , respectivamente, como los vectores-patrones de entrada y salida. Cada vector entrada esta relacionado con su correspondiente vector salida. Para cada  $k$  entero positivo, la asociación será denotada como:  $(x^k, y^k)$ . La memoria asociativa  $M$  es representada por una matriz de la cual su  $ij$ -ésima componente es  $m_{ij}$  [9]. La memoria  $M$  es generada *anticipadamente* de un conjunto finito de asociaciones conocidas, llamado conjunto fundamental de asociaciones. Si  $\mu$  es un índice, el conjunto fundamental es representado como:  $\{(x^\mu, y^\mu) \mid \mu = 1, 2, \dots, p\}$  con  $p$  como la cardinalidad del conjunto. Los patrones del conjunto fundamental se llaman patrones fundamentales. Si definimos  $x^\mu = y^\mu \forall \mu \in \{1, 2, \dots, p\}$   $M$  es autoasociativa, de otra forma esta es heteroasociativa; en este caso, es posible establecer que  $\exists \mu \in \{1, 2, \dots, p\}$  para cada  $x^\mu \neq y^\mu$ . Si consideramos el conjunto fundamental de patrones  $\{(x^\mu, y^\mu) \mid \mu = 1, 2, \dots, p\}$  donde  $n$  y  $m$  son la dimensión de los patrones de entrada y salida, respectivamente, decimos que  $x^\mu \in A^n$ ,  $A = \{0, 1\}$  y  $y^\mu \in A^m$ . Entonces la  $j$ -ésima componente del patrón de entrada es  $x_j^\mu \in A$ . De la misma forma, la  $j$ -ésima componente de una patrón de salida es representada como  $y_j^\mu \in A$ . Una versión distorsionada de un patrón  $x^k$  para ser recuperado será denotado como  $\tilde{x}^k$ . Si se presenta un patrón de entrada desconocido  $x^\omega$  con  $\omega \in \{1, 2, \dots, k, \dots, p\}$  a una memoria asociativa  $M$ , de tal forma que la salida corresponde exactamente con el patrón asociado  $y^\omega$ , se dice que la recuperación es perfecta.

La memoria asociativa Alfa-Beta matemáticamente esta basada en dos operadores binarios:  $\alpha$  y  $\beta$ . El operador Alfa es usado en la fase de aprendizaje mientras que el operador Beta es usado en la fase de recuperación. Las propiedades matemáticas de estos operadores, permiten que las memorias asociativas  $\alpha\beta$  poseer características similares a la versión binaria de las memorias asociativas morfológicas, en el sentido de: la capacidad de aprendizaje, la memoria es robusta para soportar cierto tipo y cantidad de ruido y posee las características necesarias para una recuperación perfecta [11]. Primero, definimos un conjunto  $A = \{0, 1\}$  y un conjunto  $B = \{00, 01, 10\}$ , entonces los operadores  $\alpha$  y  $\beta$  pueden ser definidos en la Tabla 1.

Estos dos operadores binarios junto con el operador máximo ( $\vee$ ) y mínimo ( $\wedge$ ) establecen las herramientas matemáticas que le conciernen al modelo Alfa-Beta.

Las definiciones de  $\alpha$  y  $\beta$  presentadas en la Tabla 1, implican que:  $\alpha$  incrementa a la izquierda y decrementa a la derecha,  $\beta$  incrementa a la izquierda y a la derecha,  $\beta$  es el inverso izquierdo de  $\alpha$ . De acuerdo al tipo de operador que es usado en la fase de aprendizaje, dos tipos de memorias asociativas Alfa-Beta pueden ser obtenidas. Si el operador máximo ( $\vee$ ) es usado, una Memoria Asociativa Alfa-Beta de tipo *MAX* será obtenida, llamemosla *M*; de manera análoga, si el operador mínimo ( $\wedge$ ) es usado, la Memoria Asociativa Alfa-Beta tipo *min* será obtenida, será llamada *W* [6]. En cualquier caso, los patrones fundamentales de entrada y salida son representados de la siguiente forma:

$$x^\mu = \begin{pmatrix} x_1^\mu \\ x_2^\mu \\ \vdots \\ x_n^\mu \end{pmatrix} \in A^n \quad y^\mu = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \in A^m$$

Para entender como se llevan a cabo las fases de aprendizaje y recuperación, serán definidas algunas operaciones matriciales.

$$\begin{aligned} \alpha \text{ máx Operación: } P_{m \times r} \nabla_\alpha Q_{r \times n} &= [f_{ij}^\alpha]_{m \times n}, \text{ donde } f_{ij}^\alpha = \vee_{k=1}^r \alpha(p_{ik}, q_{kj}) \\ \beta \text{ máx Operación: } P_{m \times r} \nabla_\beta Q_{r \times n} &= [f_{ij}^\beta]_{m \times n}, \text{ donde } f_{ij}^\beta = \vee_{k=1}^r \beta(p_{ik}, q_{kj}) \\ \alpha \text{ mín Operación: } P_{m \times r} \Delta_\alpha Q_{r \times n} &= [f_{ij}^\alpha]_{m \times n}, \text{ donde } f_{ij}^\alpha = \wedge_{k=1}^r \alpha(p_{ik}, q_{kj}) \\ \beta \text{ mín Operación: } P_{m \times r} \Delta_\beta Q_{r \times n} &= [f_{ij}^\beta]_{m \times n}, \text{ donde } f_{ij}^\beta = \wedge_{k=1}^r \beta(p_{ik}, q_{kj}) \end{aligned}$$

Siempre que un vector columna de dimensión  $m$  es operado con un vector fila de dimensión  $n$ , ambas operaciones  $\nabla_\alpha$  y  $\Delta_\alpha$ , son representadas por  $\oplus$ ; consecuentemente, la siguiente expresión es valida:

$$y \nabla_\alpha x^t = y \oplus x^t = y \Delta_\alpha x^t.$$

Si se considera el conjunto fundamental de patrones  $\{(x^\mu, y^\mu) \mid \mu = 1, 2, \dots, p\}$  entonces la  $ij$ -ésima entrada de la matriz  $y^\mu \oplus (x^\mu)^t$  es expresada como:

$$\left[ y^\mu \oplus (x^\mu)^t \right]_{ij} = \alpha(y_i^\mu, x_j^\mu).$$

## 2.1. Fase de aprendizaje

Encontrar los operadores adecuados y una manera de generar la matriz *M* esta guardará las  $p$  asociaciones del conjunto fundamental  $\{(x^1, y^1), (x^2, y^2), \dots, (x^p, y^p)\}$ , donde  $x^\mu \in A^n$  y  $y^\mu \in A^m \forall \mu \in \{1, 2, \dots, p\}$ .

**Paso 1.** Para cada asociación de patrones fundamentales  $\{(x^\mu, y^\mu) \mid \mu = 1, 2, \dots, p\}$ , generar  $p$  matrices de acuerdo a la siguiente regla:

$$\left[ y^\mu \oplus (x^\mu)^t \right]_{m \times n}$$

**Paso 2.** Para obtener la Memoria Alfa-Beta tipo *MAX*, aplicar el operador binario *MAX* ( $\vee$ ) de acuerdo a la siguiente regla:

$$M = \vee_{\mu=1}^p \left[ y^\mu \oplus (x^\mu)^t \right]$$

**Paso 3.** Para obtener un Memoria Alfa-Beta tipo *min* aplicar el operador binario *min* ( $\wedge$ ) de acuerdo a la siguiente regla:

$$W = \wedge_{\mu=1}^p \left[ y^\mu \oplus (x^\mu)^t \right]$$

En consecuencia, la *ij*-ésima entrada de la Memoria Alfa-Beta de tipo *MAX* esta dada por la siguiente expresión:

$$\nu_{ij} = \vee_{\mu=1}^p \alpha(y_i^\mu, x_j^\mu)$$

Análogamente, la *ij*-ésima entrada de una Memoria Asociativa Alfa-Beta de tipo *min* esta dada por la siguiente expresión:

$$\psi_{ij} = \wedge_{\mu=1}^p \alpha(y_i^\mu, x_j^\mu).$$

## 2.2. Fase de recuperación

Encontrar los operadores adecuados y las condiciones suficientes para obtener un patrón fundamental de salida  $y^\mu$  cuando una memoria  $M$  o una memoria  $W$  es operada con un patrón fundamental de entrada  $x^\mu$ .

**Paso 1.** Un patrón  $x^\omega$ , con  $\omega \in \{1, 2, \dots, p\}$ , es presentado a la Memoria Asociativa Alfa-Beta, entonces,  $x^\omega$  es recuperado de acuerdo a una de las siguiente reglas.

Memoria Asociativa Alfa-Beta de tipo *MAX*:

$$M \Delta_\beta x^\omega = \wedge_{j=1}^n \beta(\nu_{ij}, x_j^\omega) = \wedge_{j=1}^n \left\{ \left[ \vee_{\mu=1}^p \alpha(y_i^\mu, x_j^\mu) \right], x_j^\omega \right\}$$

Memoria Asociativa Alfa-Beta de tipo *min*:

$$W \nabla_\beta x^\omega = \vee_{j=1}^n \beta(\psi_{ij}, x_j^\omega) = \vee_{j=1}^n \left\{ \left[ \wedge_{\mu=1}^p \alpha(y_i^\mu, x_j^\mu) \right], x_j^\omega \right\}$$

Sin importar el tipo de Memoria Asociativa Alfa-Beta usado la fase de recuperación, un vector columna de dimensión  $n$  será obtenido.

## 3. DEAQ

El diccionario de especialidades agro-químicas es un libro editado año con año por PLM desde 1989, este libro contiene información de más de 1800 productos agro-químicos, este libro puede ser consultado por:

- Productos
- Usos
- Cultivos
- Ingredientes activos

En este libro, la información proporcionada por la empresa que fabrica el producto es recopilada y clasificada, para proporcionar un gran conjunto de datos. La página principal del DEAQ puede ser accesada en [12].

#### 4. Conjuntos de datos de herbicidas

La base de datos relacional, se generó como una abstracción del DEAQ este puede ser consultado en [13]. Se tomaron las características principales de 40 herbicidas, creando 3473 instancias. Las características de estas instancias son toxicidad, método de acción, ingrediente activo, periodo de reentrada, incompatibilidad, contraindicaciones y primeros auxilios.

El conjunto de datos tiene la estructura siguiente:

{ "maleza", cultivo, "toxicidad", "periodo de reentrada", "herbicida" }

Donde "maleza", cultivo, "toxicidad" "periodo de reentrada" son los principales atributos a ser analizados y "herbicida" es la mejor opción por elegir.

Se tienen 68 malezas, 50 cultivos, 2 tipos de toxicidad y 6 periodos de reentrada diferentes.

El conjunto de datos esta mapeado a la base de datos relacional por los índices, cada número representa un elemento diferente ya sea maleza, cultivo, toxicidad o herbicida.

#### 5. Detalles de implementación

El presente trabajo fue realizado haciendo uso de las Memorias Asociativas Alfa-Beta tipo *min*, autoasociativas, estas memorias fueron combinadas con el código Johnson-Möbius [14].

Esto debido a que las Memorias Asociativas Alfa-Beta clasifican patrones de entrada con ruido aditivo o sustractivo, pero estas no son tan eficientes con ruido combinado, pero si incluimos la codificación de Johnson-Möbius, las Memorias Asociativas Alfa-Beta serán buenas clasificando patrones con ruido combinado.

Este código tiene una propiedad importante, cuando un valor es codificado en Johnson-Möbius este código preserva el ruido aditivo o sustractivo pero no mixto.

Para generar el código Johnson-Möbius de un número de cero a  $n - 1$ , se requieren  $\frac{n}{2}$  bits. Si modificamos el código Johnson-Möbius para solo usar los códigos entre cero y  $\frac{n}{2}$ , donde  $\frac{n}{2}$  es el número máximo representado (denotado como  $n_{max}$ ) e invertimos el orden de los bits de la forma siguiente. El bit más significante se convierte en el bit menos significante, y el menos significante se convierte en el más significante. Por lo tanto, para representar un número que va de cero a  $n_{max}$ , se necesitan  $n_{max}$  bits.

El algoritmo para generar un número codificado en Johnson-Möbius es:

- 1.- Seleccionar el mayor de los valores dados
- 2.- Escribir el número como una concatenación de bits con el valor 1
- 3.- Completar con bits de valor 0 de izquierda a derecha hasta alcanzar la longitud  $n$ .

Ejemplo:

Valores: 3,5,10,6,13

- 1.- Seleccionamos el mayor de los valores dados, en este caso es 13,  $n=13$ .

2.- Escribir el número como una concatenación de bits con el valor 1

3: 111  
5: 11111  
10: 1111111111  
6: 111111  
13: 1111111111111

3.- Completar con bits de valor 0 de izquierda a derecha hasta alcanzar la longitud  $n$ .

3: 000000000111  
5: 000000011111  
10: 0001111111111  
6: 000000111111  
13: 1111111111111

Debido a características del desarrollo se diseñaron dos Memorias Asociativas Alfa-Beta tipo min autoasociativas, una estricta y otra flexible, estas dos memorias proporcionan un mejor rango de soluciones, las dos memorias fueron entrenadas usando el mismo conjunto fundamental, la diferencia entre estas dos memorias es que la Memoria Asociativa Alfa-Beta tipo min autoasociativa estricta utiliza el conjunto completo de características en la fase de aprendizaje y recuperación al contrario de la Memoria Asociativa Alfa-Beta tipo min autoasociativa flexible que utiliza un conjunto reducido de características en la fase de aprendizaje y recuperación. Usando la característica de una completa recuperación de los patrones de salida la Memoria Asociativa Alfa-Beta tipo min estricta nos puede dar un excelente resultado si el usuario selecciona un patrón del conjunto fundamental, pero cuando un patrón entrante no pertenece al conjunto fundamental, se utiliza la Memoria Asociativa Alfa-Beta tipo min flexible. La memoria asociativa flexible se realizó utilizando la técnica de selección de características que nos ofrece las memorias Alfa-Beta, con esto seleccionamos las características principales, que son maleza y cultivo.

Después de codificar el patrón de entrada con el código de Johnson-Möbius se utiliza la Memoria Asociativa Alfa-Beta tipo min estricta para buscar un buen resultado si se encuentra, se muestra la información del producto, si no, el patrón de entrada va a la Memoria Asociativa Alfa-Beta tipo min flexible donde se busca un mejor resultado.

## 6. Resultados numéricos

**Ejemplo 4.1.** Sea  $p = 5$ ,  $n = 4$ ,  $m = 4$ . Dados los patrones fundamentales  $\{(x^\mu, y^\mu) \mid \mu = 1, 2, \dots, p\}$ , se obtiene una Memoria Asociativa Alfa-Beta. Las asociaciones fundamental serán escritas como:  $\{(x^1, y^1), (x^2, y^2), \dots, (x^5, y^5)\}$ .

$$\begin{array}{ccccc}
 x^1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} & x^2 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} & x^3 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} & x^4 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} & x^5 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \\
 y^1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} & y^2 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} & y^3 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} & y^4 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} & y^5 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}
 \end{array}$$

**Fase de aprendizaje.** Obtenemos las matrices correspondientes  $M_1, M_2, \dots, M_5$ , de acuerdo al paso 1, indicado en la sección 2.1.

$$\begin{array}{l}
 y^1 \oplus (x^1)^t = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} \oplus (1\ 1\ 0\ 1) = \begin{pmatrix} 1\ 1\ 2\ 1 \\ 1\ 1\ 2\ 1 \\ 0\ 0\ 1\ 0 \\ 1\ 1\ 2\ 1 \end{pmatrix} \\
 y^2 \oplus (x^2)^t = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \oplus (1\ 0\ 0\ 1) = \begin{pmatrix} 1\ 2\ 2\ 1 \\ 0\ 1\ 1\ 0 \\ 0\ 1\ 1\ 0 \\ 1\ 2\ 2\ 1 \end{pmatrix} \\
 \vdots \\
 y^5 \oplus (x^5)^t = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \oplus (1\ 0\ 1\ 1) = \begin{pmatrix} 1\ 2\ 1\ 1 \\ 0\ 1\ 0\ 0 \\ 1\ 2\ 1\ 1 \\ 1\ 2\ 1\ 1 \end{pmatrix}
 \end{array}$$

De acuerdo al paso 2, una Memoria Asociativa Alfa-Beta de tipo *MAX* representada por  $M$ , es obtenida, análogamente de acuerdo al paso 3 una Memoria Asociativa Alfa-Beta de tipo mín representada por  $W$ , es obtenida

$$M = \begin{pmatrix} 1\ 2\ 2\ 1 \\ 2\ 1\ 2\ 2 \\ 2\ 2\ 1\ 1 \\ 2\ 2\ 2\ 1 \end{pmatrix} \quad ; \quad W = \begin{pmatrix} 1\ 0\ 0\ 0 \\ 0\ 1\ 0\ 0 \\ 0\ 0\ 1\ 0 \\ 1\ 0\ 1\ 1 \end{pmatrix}$$

**Fase de recuperación.** Obtener el correspondiente patrón de salida, realizando las operaciones  $M \triangle_{\beta} x^{\mu}$ ,  $\forall \mu \in \{1, 2, \dots, p\}$  como fue declarado en la sección 2.2. Debido a las limitaciones de espacio, solo la fase de recuperación de las Memorias Alfa-Beta tipo *MAX* es mostrada.

$$M \triangle_{\beta} x^1 = \begin{pmatrix} 1\ 2\ 2\ 1 \\ 2\ 1\ 2\ 2 \\ 2\ 2\ 1\ 1 \\ 2\ 2\ 2\ 1 \end{pmatrix} \triangle_{\beta} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = y^1$$

$$\begin{aligned}
 M \triangle_{\beta} x^2 &= \begin{pmatrix} 1 & 2 & 2 & 1 \\ 2 & 1 & 2 & 2 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 2 & 1 \end{pmatrix} \triangle_{\beta} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = y^2 \\
 M \triangle_{\beta} x^3 &= \begin{pmatrix} 1 & 2 & 2 & 1 \\ 2 & 1 & 2 & 2 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 2 & 1 \end{pmatrix} \triangle_{\beta} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} = y^3 \\
 M \triangle_{\beta} x^4 &= \begin{pmatrix} 1 & 2 & 2 & 1 \\ 2 & 1 & 2 & 2 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 2 & 1 \end{pmatrix} \triangle_{\beta} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = y^4 \\
 M \triangle_{\beta} x^5 &= \begin{pmatrix} 1 & 2 & 2 & 1 \\ 2 & 1 & 2 & 2 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 2 & 1 \end{pmatrix} \triangle_{\beta} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = y^5
 \end{aligned}$$

El lector fácilmente puede verificar que la fase de recuperación de Alfa-Beta de tipo *min* también puede recuperar perfectamente y por completo el conjunto fundamental de patrones.

## 7. Resultados experimentales

El conjunto de datos utilizado para las pruebas, contenía 5 razgos, 40 clases y 3473 instancias, fue probado en "WEKA 3: Data Mining Software in Java" [15]. WEKA es un programa open source bajo la licencia "GNU General Public License", disponible de forma gratuita en internet [16]. Todos los experimentos fueron realizados usando una computadora personal con un procesador AMD Phenom II X4 955 a 3.2 GHz, corriendo sobre un Windows 8 Profesional con 8GB de RAM. En todas las pruebas se aplicaron las mismas condiciones y esquemas de validación para cada experimento. La condición para la experimentación fue enseñar al algoritmo con un conjunto de patrones y después recuperar todos los patrones. En la Tabla 2, se compara la eficiencia que tuvieron diversos algoritmos recuperando la totalidad de patrones enseñados, cabe resaltar, que las Memorias Asociativas Alfa-Beta tuvieron una recuperación de patrones del 100 % en la etapa de validación.

## 8. Conclusiones e investigación en progreso

Usando Memorias Asociativas Alfa-Beta, se puede recuperar completamente el conjunto fundamental, esto nos brinda una gran oportunidad para desarrollar un conjunto de aplicaciones enfocadas a la agronomía. En este artículo se

**Tabla 2.** Eficiencia de recuperacin probado con weka.

Algoritmo	Instancias clasificadas correctamente	Porcentaje
IB1	2510	72.27 %
RandomForest	2505	72.13 %
BayesNet	2484	71.52 %
Bagging	2444	70.37 %
AtributedSelectedClassifier	2361	67.98 %
J48	2361	67.98 %
END	2342	67.43 %
REPTree	2322	66.86 %
RandomSubSpace	2299	66.20 %
BFTree	2296	66.11 %
DataNearBalancedND	2275	65.51 %
ClassBalancedND	2269	65.33 %
REPTree	2200	63.35 %
SimpleCart	2189	63.03 %
Memorias Asociativas Alfa-Beta	3473	100 %

mostró que las Memorias Asociativas pueden facilmente ser usadas en cualquier tipo de desarrollo, en este caso particular una aplicación agronómica.

Particularmente sobre los sistemas que emiten recomendaciones, estos son muy útiles para el campo mexicano, ya que un ingeniero agrónomo, difícilmente, domina un catálogo extenso de productos agro-químicos y posee información sobre las condiciones exactas de aplicación, el ingeniero agrónomo puede ayudarse de este tipo de sistemas, cuando desconozca en cierta medida, el mejor producto para una aplicación en determinado cultivo y maleza.

Un sistema basado en Memorias Asociativas Alfa-Beta también puede recomendar el mejor momento para la aplicación de un producto, considerando variables como la velocidad del viento, la altura de la maleza y la posibilidad de lluvia, modelando todas estas características en diferentes ejemplos y usando las Memorias Asociativas Alfa-Beta como clasificador.

Actualmente investigamos sobre como mejorar las memorias asociativas a través de la selección de características, esto para implementar las memorias asociativas en otros campos de la ciencia.

**Agradecimientos.** Los autores del presente artículo quisieran dar las gracias a las siguientes instituciones por su apoyo: Instituto Politécnico Nacional, México (CIDETEC, ESCOM, CGPI, PIFI, COFAA) , CONACyT y SNI.

## Referencias

1. Y. Hashimoto: Applications of artificial neural networks and genetic algorithms to agricultural systems. Computers and electronics in agriculture, 18, 71-72 (1997)

2. Dimitrios Moshou, Els Vrindts, Bart De Ketelaere, Josse De Baerdemaeker, Herman Ramon: A neural network based plant classifier, *Computers and electronics in agriculture*, 31, 5–16 (2001)
3. M.J. Aikenhead, I.A. Dalgetty, C.E. Mullins, A.J.S McDonald, N.J.C. Strachan: Weed and crop discrimination using image analysis and artificial intelligence methods, *Computers and electronics in agriculture*, *Computers and electronics in agriculture*, 39, 157–171 (2003)
4. T. Karimi, S.O. Prasher, R.M. Patel, S.H. Kim: Application of support vector machine technology for weed and nitrogen stress detection in corn, *Computers and electronics in agriculture*, 51, 99–109 (2006)
5. Juan Ignacio Arribas, Gonzalo V. Sánchez-Ferrero, Gonzalo Ruiz-Ruiz, Jaime Gómez-Gil: Leaf classification in sunflower crops by computer vision and neural networks, 78, 9–18 (2011)
6. Acevedo-Mosqueda, M. E., Yáñez-Márquez, C., & López-Yáñez, I.: Alpha–Beta bidirectional associative memories: theory and applications. *Neural Processing Letters*, 26 (2007)
7. Steinbuch, K.: Die Lernmatrix. *Kybernetik* 1(1):36–45 (1961)
8. Anderson, J.A.: A simple neural network generating an interactive memory. *Math Biosci* 14:197–220 (1972)
9. Kohonen, T.: Correlation Matrix Memories. *IEEE Transactions on Computers*, 21, 353–359 (1972)
10. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. In: *Proc Nat AcadSci* 79:2554–2558 (1982)
11. Acevedo-Mosqueda, M. E., Yáñez-Márquez, C., & López-Yáñez, I.: A New Model of BAM: Alpha-Beta Bidirectional Associative Memories. *Lecture Notes in Computer Science (LNCS)*, 4263, 286–295 (2006)
12. PLM. DEAQ Diccionario de especialidades Agroquímicas. (2014) [Online]. <http://www.agroquimicos-organicosplm.com/inicio>
13. PLM. DEAQ Diccionario de especialidades Agroquímicas. (2014) [Online]. <http://www.elcamporadio.com/source/>
14. C. Yáñez, E.M.F. Riverón, I. López-Yáñez, R. Flores-Carapia: A novel approach to automatic color matching. In: *CIARP*, pp. 529–538 (2006)
15. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten: The WEKA data mining software: an update, *SIGKDD Explorations* 11 (1), pp. 10–18 (2009)
16. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten: *WEKA 3: Data mining software in java* (2010). URL <http://www.cs.waikato.ac.nz/ml/weka/>