
RED NEURONAL ARTIFICIAL TIPO MEMORIA ASOCIATIVA
BIDIRECCIONAL (MAB)

González Fernández J.J. y Espinosa Espinosa J.I.
Lab. de Cibernética, Depto. de Física, Fac. de Ciencias, U.N.A.M.
550-5215 X 3922, 550-5909, BITNET: ESPIN @ UNAMVM1

RESUMEN

Se describen diferentes algoritmos de aprendizaje para una Red Neuronal Artificial con una arquitectura tipo MAB que permite almacenar varias asociaciones o patrones. Y se presentan ejemplos sobre el reconocimiento de patrones con información barrosa o incompleta, comparando el desempeño de cada algoritmo con base a la reconstrucción a partir del patrón ruidoso. Finalmente se discuten las ventajas y desventajas de los algoritmos.

INTRODUCCION

Un elemento procesador (neurona artificial) está inspirado en una neurona biológica. Este parecido radica en que el procesador recibe múltiples entradas que una a una son ponderadas y luego integradas. El resultado de esta integración es presentado como entrada a una función de activación que produce una respuesta. Hasta aquí el parecido biológico [1]. De esta manera, los componentes de un procesador son el integrador y la función de activación. En la Fig.1, se muestra un procesador donde las entradas u_i son ponderadas por los pesos p_i y la integración de todo ello da el potencial total Suma. Esta Suma es presentada a una función no lineal de activación $f(\text{Suma})$ que produce la Salida (respuesta).

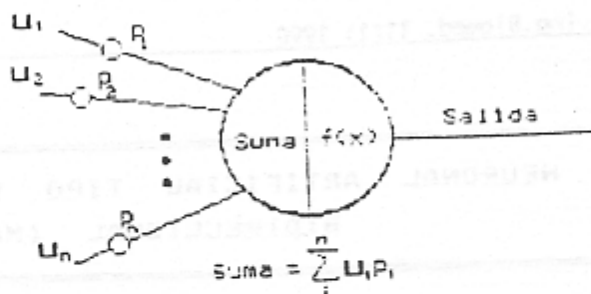


Fig.1 Elemento procesador o neurona artificial.

Sólo una vez que se ha definido al procesador se puede hablar de redes neuronales artificiales. Una red neuronal artificial es un sistema de procesadores masivamente interconectados en arquitecturas específicas que generalmente son en cascada. Tanto en las redes neuronales biológicas, como en las artificiales, su importancia radica en la capacidad para comunicarse unas con otras. Un procesador puede excitar o inhibir a otros muchos y, además, el efecto de un procesador sobre otro está cuantificado o ponderado a la manera que ocurre en las sinapsis neuronales.

Las Memorias Asociativas son sistemas que pueden reconstruir patrones almacenados por especificación de todo o una porción de uno de los patrones previamente "almacenados" por un algoritmo de aprendizaje. El aprendizaje de la red está almacenado en las conexiones en la red neuronal. Una MAB consta de dos capas de procesadores que se interconectan entre sí (sinapsis) y en las que no se permite la inhibición lateral (ver Fig.2). Sean $\{a_1, a_2, \dots, a_n\}$ y $\{b_1, b_2, \dots, b_p\}$ los procesadores de las capas A y B, respectivamente.

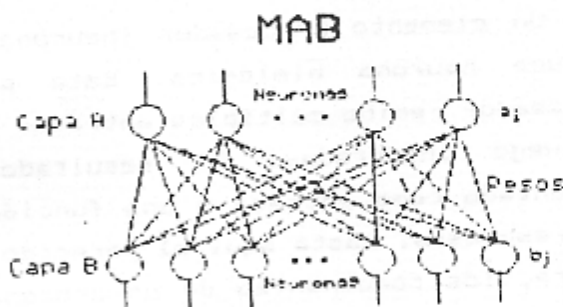


Fig.2 Red Neuronal Artificial tipo MAB.

Entonces una matriz $M_{n \times n}$ puede ser interpretada como una matriz de números reales (pesos de las conexiones) entre las dos capas de procesadores. Así el elemento p_{ij} de la matriz $M_{n \times n}$ representa la conexión sináptica entre a_i y b_j . El signo de p_{ij} determina el tipo de conexión sináptica, será excitatoria si $p_{ij} > 0$ e inhibitoria si $p_{ij} < 0$. La magnitud de p_{ij} representa entonces la intensidad o fuerza sináptica (peso) de la conexión. Así, una asociación consta de un par de patrones que forman un estado de equilibrio del sistema red neuronal tipo MAB. En resumen, el aprendizaje de la red está almacenado en la conectividad entre ambas capas, y esta conectividad es determinada por los algoritmos para calcular los pesos.

ALGORITMOS

La MAB fue primeramente introducida por KOSKO [6],[7]. Este es un tipo de red neuronal con las siguientes operaciones: Existen N asociaciones $\{(A_1, B_1), (A_2, B_2), \dots, (A_N, B_N)\}$ donde $A_i = (a_{i1}, a_{i2}, \dots, a_{in})$, $B_i = (b_{i1}, b_{i2}, \dots, b_{in})$ y a_i, b_j son los estados ON u OFF. En modo binario, ON=1 y OFF=0. Y para modo bipolar, ON=1 y OFF=-1. Forma una matriz de correlación:

$$M = \sum_{i=1}^N X_i^t Y_i \quad (1)$$

donde $X_i(Y_i)$ es el modo bipolar de $A_i(B_i)$. Se desea entonces recuperar una asociación (A_i, B_i) presentando un patrón ruidoso a la red (α, β) . Este último estado determina la secuencia $(\alpha', \beta'), (\alpha'', \beta'') \dots$ hasta un estado de equilibrio (α_r, β_r) , donde:

$$\beta' = g(\alpha M) \quad \alpha' = g(\beta' M^t) \quad (2)$$

$$g(F) = G = (g_1, g_2, \dots, g_n) \quad F = (f_1, f_2, \dots, f_n) \quad (3)$$

$$g_i = 1, f_i > 0$$

$$g_i = \begin{cases} 0 \text{ (binario), } f_i < 0 \\ -1 \text{ (bipolar)} \end{cases} \quad (4)$$

$$g_i = \text{anterior } g_i, f_i = 0.$$

Cualquier estado (α, β) tiene definida una función de energía:

$$E = -\alpha M \beta^t \quad (5)$$

KOSKO ha demostrado que a cada ciclo la energía disminuye. Así el estado final (α_r, β_r) con $E = -\alpha_r M \beta_r^t$ es un mínimo local. El algoritmo de KOSKO calcula los pesos dados por (1). Por otra

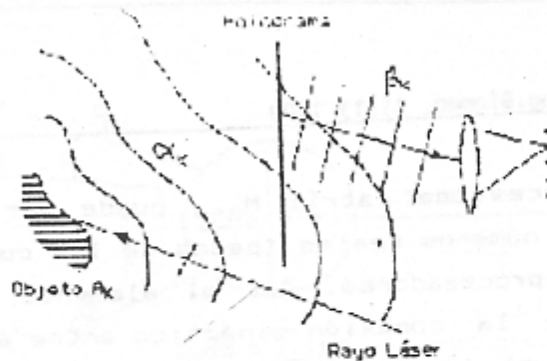


Fig.4 Operación de una Memoria Asociativa Holográfica.

Sea $\{Z_1, Z_2, \dots, Z_N\}$ un conjunto de vectores ortonormales que generan al espacio euclidiano l -dimensional, y sean M_1 y M_2 dos matrices de correlación dadas por:

$$M_1 = \sum_1^N X_i Z_i^t \quad \text{y} \quad M_2 = \sum_2^N Z_j Y_j^t \quad (8)$$

donde $\langle Z_i, Z_j \rangle = 0$ y $\langle Z_i, Z_i \rangle = \langle Z_j, Z_j \rangle = 1$ y como antes $X_i (Y_i)$ es el modo bipolar de $A_i (B_i)$. Comparando con los algoritmos anteriores, aquí existen dos operadores en vez de uno. Supóngase que se le presenta a la red un patrón X (en modo bipolar), la recuperación con el operador M es XM ; aquí es $XM_1 M_2$ ($M_1 M_2$ toman el lugar del operador M y la recuperación inversa es $Y M_2^t M_1^t$). Una opción para la construcción de los vectores ortogonales es el generado por la funciones de Walsh. Brevemente, las funciones de Walsh forman un conjunto de funciones ortogonales en el espacio 2^r . Así el elemento $z_1^{(k)}$ del vector ortogonal Z_k es:

$$z_1^{(k)} = (-1)^s \quad \text{con} \quad s = \sum_1^{r-1} l_{r-1-i} k_i \quad (9)$$

donde $(l)_{\text{decimal}} = (l_{r-1} l_{r-2} \dots l_1 l_0)_{\text{binario}}$

$$(k)_{\text{decimal}} = (k_{r-1} k_{r-2} \dots k_1 k_0)_{\text{binario}} \quad (10)$$

y $r = \log_2 p$ con $l, k = 0, 1, \dots, p-1$. Una vez obtenido el vector ortogonal se normaliza para obtener el vector ortonormal.

A los algoritmos anteriores se les llama algoritmos deterministas. Existen otros algoritmos llamados adaptivos, en esta parte se describen 2 de ellos. Para los algoritmos adaptivos la red aprende las asociaciones que le son presentadas secuencialmente en pares entrada-salida. Se parte de una matriz inicial de pesos con valores aleatorios pequeños. Si la salida no es la deseada, se genera una diferencia o error que se minimiza

parte, KOHONEN [5] ha propuesto utilizar la matriz inversa generalizada de Moore-Penrose (pseudoinversa) [11] como un mecanismo para construir la matriz. De tal forma que la matriz $M_{n \times n}$ está dada por:

$$M = M^* = (YX^*)^e \quad (6)$$

donde $X_{n \times n} = [X_1^e, X_2^e, \dots, X_n^e]$ y $Y_{p \times n} = [Y_1^e, Y_2^e, \dots, Y_p^e]$ son matrices, X^* es la pseudoinversa y $X_i(Y_i)$ es el modo bipolar de $A_i(B_i)$.

PAO proporciona información sobre la construcción de la matriz de memoria $M_{n \times n}$, haciendo una modificación al primer algoritmo [12]. Aquí, la matriz de correlación es:

$$M = \sum_1^N x_i y_i \quad (7)$$

donde $x_i = X_i / |X_i|$ y $y_i = Y_i / |Y_i|$ y de manera que $X_i(Y_i)$ es el modo bipolar de $A_i(B_i)$. Otro algoritmo propuesto por PAO es el de una memoria asociativa holográfica (HOLO) basado en la teoría física de la Holografía. En la holografía ordinaria, se sabe que, si el holograma es iluminado con un rayo de referencia β_k , entonces el campo α_k es generado desde el holograma provocando una imagen virtual, como se ilustra en la Fig.3. El proceso inverso es si el holograma es iluminado con el campo α_k , entonces el campo de referencia β_k es generado. Para éste caso, si los campos de referencia difieren sólo en su ángulo de incidencia, una lente puede ser utilizada para identificar cuál de los campos de referencia ha sido generado por un objeto. Este hecho se muestra en la Fig.4. Así se construye una memoria asociativa holográfica (HOLO) presentando un objeto A_k para formar el campo α_k , entonces se unen los campos α_k y β_k . La intensidad del campo $(\alpha_k + \beta_k)$ es registrado sobre la película en el plano del holograma.

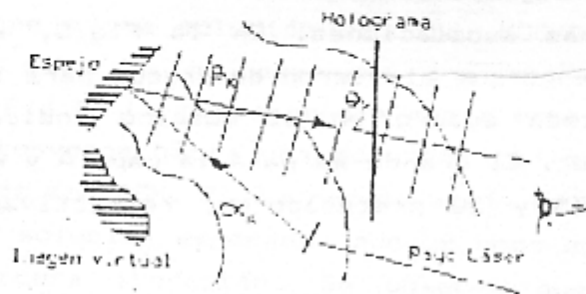


Fig.3 Holografía Ordinaria.

por medio de cambios en los pesos de las conexiones en la red. Esto se hace para cada asociación y, después de un período de entrenamiento, tal error se minimiza y esto da por terminada la etapa de aprendizaje de la red. En términos de los pesos se tiene una nueva matriz de pesos resultado de la evolución dinámica a lo largo del proceso del algoritmo. Estos pesos poseen de manera colectiva el aprendizaje de las asociaciones presentadas [1].

La ley de cambios de pesos para el primer algoritmo adaptivo de KOSKO, Aprendizaje por Señal de Hebb (ASH), es:

$$p_{i,j}' = -p_{i,j} + f(a_i)f(b_j) \quad (11)$$

donde $p_{i,j}' = dp_{i,j}/dt$ y f es la función de activación sigmoideal $f(x) = (1 + e^{-x})^{-1}$. El segundo algoritmo adaptivo es el Aprendizaje Diferencial de Hebb (ADH). Este algoritmo es una modificación al anterior, de tal forma que correlaciona señales y sus variaciones con el tiempo, haciendo:

$$p_{i,j}' = -p_{i,j} + f(a_i)f(b_j) + f'(a_i)f'(b_j) \quad (12)$$

donde $f' = df/dt$ [6].

METODOLOGIA

Con un paquete de computación hecho en TURBO PASCAL Versión 5.0 se programaron los algoritmos. La red neuronal utilizada consta de 55 procesadores, como se muestra en la Fig.5. Se hizo el reconocimiento de patrones con variaciones de ruido. Los mosaicos se dibujaron en PAINTBRUSH Versión 2.0. También se graficó la variación de error a cada cambio en los pesos para los algoritmos adaptivos, utilizando GRAPHPC Versión 2.31. La simulación se hizo en una computadora GAMA BABY 786. Por falta de espacio se muestra sólo una figura en cada caso.

RESULTADOS

Los algoritmos descritos fueron estudiados, uno a uno, con las mismas asociaciones. En la Fig.5, se muestran las cinco asociaciones que sirvieron de prueba para los algoritmos. En esta figura cada cuadro en el mosaico indica la actividad de un procesador. El grande es para la capa A y el pequeño para la capa B, con 35 y 20 procesadores, respectivamente. Los cuadros en

negro son procesadores activados y los cuadros en blanco son procesadores no-activados.

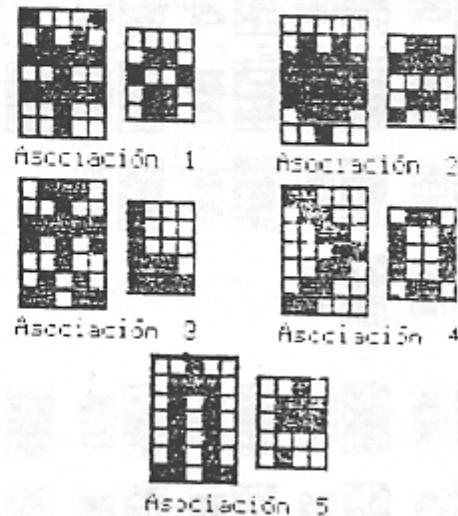


Fig.5 Asociaciones de prueba para los algoritmos. Cada cuadro en el mosaico indica la actividad de un procesador. Los cuadros en negro son procesadores activados y los cuadros en blanco son procesadores no-activados.

En la Fig.6 se muestran las evoluciones de la dinámica de la red para cada uno de los algoritmos. En la parte superior derecha de la figura se muestra la asociación a la que debería llegar la red para el patrón ruidoso dado, que se encuentra a la mitad de la parte superior y del que se muestra la variación del 14.29%. En el ejemplo se presentó el patrón ruidoso en la capa A. En la parte de la derecha se indica el modo de recuperación: binaria o bipolar. La secuencia de iteraciones en la evolución temporal de la red es de izquierda a derecha. El estado final (el mosaico de la derecha) es un estado estable, es decir, a partir de ahí, la red permanece en ese estado. En esta figura para el modo binario sólo HOLA encontró, en la segunda iteración, la solución esperada. KOHONEN cae a una solución espuria. KOSKO, FAO y ADH tienden hacia una solución no esperada que se parece a la asociación 2 de la Fig.5. La convergencia hacia esta solución se debe a que la MAB converge hacia una solución con mayor energía negativa, esto sucede con ASH. En la recuperación en modo bipolar todos encuentran la solución esperada, aunque unos en la segunda y otros en la tercera iteración. Se observa que todas las recuperaciones son diferentes, excepto HOLA y KOHONEN, que además

resultan ser las más rápidas y las únicas que encuentran la solución en los dos modos de recuperación.

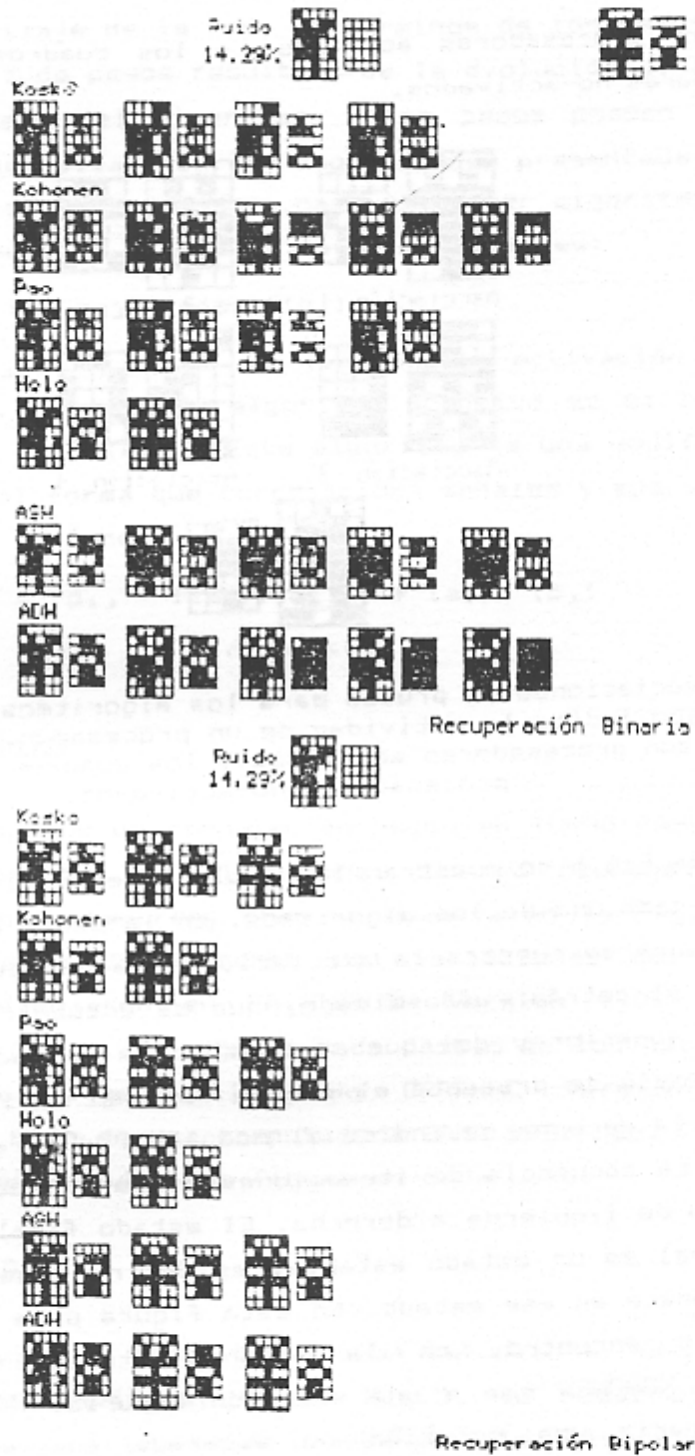


Fig.6 Evolución de la red en modo binario y bipolar.

Una manera de ver la evolución de la etapa de aprendizaje es ver la variación del error a cada cambio en los pesos. En la Fig.7 se muestra la gráfica de error para el algoritmo ADH para la red considerada. ADH hace 49 cambios en los pesos durante 9 minutos y 45 segundos y concluyó con error del 0.68%, en una computadora GAMA BABY con procesador 80286 a 10 MHz.

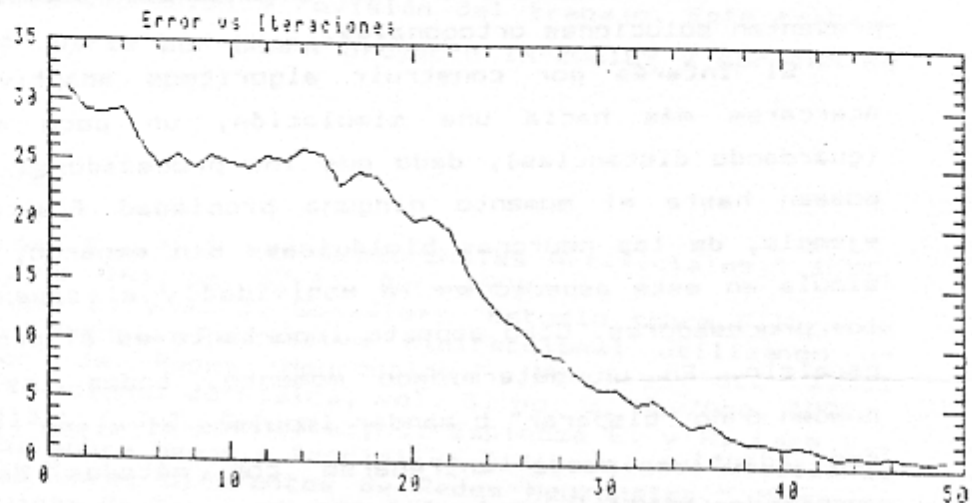


Fig.7 Gráfica de variación del error para la red con 5 asociaciones con el algoritmo ADH.

DISCUSION Y CONCLUSION

Stiles y Denq [9] muestran que la recuperación depende del número de patrones asociados, del número de procesadores por capa y del porcentaje de ruido. En este estudio se verifica lo anterior además que la recuperación también depende del algoritmo de aprendizaje empleado. Se muestra también que la recuperación de patrones en forma bipolar es más precisa que la recuperación en modo binario, tanto en algoritmos deterministas como adaptivos. Se muestra también las ventajas en la recuperación de patrones de un algoritmo con respecto a los demás.

De los algoritmos estudiados, sólo el de KOSKO ha sido estudiado con más detalle en trabajos anteriores [2],[3]. KOHONEN [5] ha discutido con detalle la memoria con matriz inversa generalizada, pero nada dice acerca de la bidireccionalidad ni de la versión discreta. KOHONEN ha demostrado que en el modelo continuo la recuperación será perfecta si los vectores asociados a los patrones son ortogonales. KOSKO afirma lo mismo para su modelo [6]-[7], al igual que PAO [12]. El modelo discreto será

así un caso particular del modelo continuo. El algoritmo para memoria holográfica no presenta este problema, ya que se generaliza para el modelo continuo, es decir, que se sigue el mismo procedimiento en el discreto y el continuo. Por otra parte, en todas las recuperaciones de patrones, el algoritmo para memoria holográfica fue el único que no obtuvo soluciones espurias. Este algoritmo puede entonces tolerar mayor porcentaje de error con respecto a los anteriores, además que no se presentan soluciones ortogonales.

El interés por construir algoritmos adaptivos es el de acercarse más hacia una simulación, un poco más biológica (guardando distancias), dado que los procesadores utilizados no poseen hasta el momento ninguna propiedad fisicoquímica, por ejemplo, de las neuronas biológicas. Sin embargo, lo que si se simula en este aspecto es la masividad y el conexionismo entre los procesadores. Otro aspecto importante es el procesamiento en paralelo. En un determinado momento, todos los procesadores pueden o no "disparar" o mandar impulsos al mismo tiempo. Así una red adaptiva puede entrenarse con métodos de aprendizaje inspirados biológicamente [6]-[7]. Por último está la aplicación tecnológica, se ha utilizado la red tipo MAB para reconocimiento de fotografías y también podría utilizarse para el reconocimiento de firmas.

De los ejemplos y las simulaciones, se observa que el algoritmo determinista HOLD ofrece mayor precisión para la recuperación y reconocimiento de patrones. También se hace evidente la existencia de varios algoritmos de aprendizaje que se valen de diferentes teorías matemáticas. Wang, Cruz y Mulligan [10] ofrecen una corrección al modelo de KOSKO, que es muy similar al de Hopfield [4]. Por otra parte, Murakami y Aibara [8] proponen una mejoría al modelo de KOHONEN. Sin embargo, no existe hasta el momento un algoritmo satisfactorio que llegue a discriminar patrones ruidosos en general.

Entre las muchas limitaciones que presenta el trabajo, una de las más importantes es que se estudiaron las recuperaciones de las asociaciones en forma discreta y no en forma continua. Otra es el número de procesadores que el paquete MAB puede manejar.

AGRADECIMIENTOS

Los autores agradecen a Antonio Ayhlón Meixueiro la ayuda prestada para la impresión y revisión del trabajo. Este trabajo fue apoyado en parte por DGAPA proyecto IN-206189 y por IBM de México.

ARTICULOS

- [1] I. Espinosa E., "Las Redes Neuronales Artificiales," ICYT, vol. 12, no. 163, pp. 37-43, Abr. 1990.
- [2] I. Espinosa E. y J. J. González, "Estudio sobre Filtro de Ruido y Memoria Asociativa Bidireccional utilizando un simulador de Redes Neuronales," Memorias del XXXII Congreso Nacional de Física, vol. 3, no. 2, May/Ago. 1989.
- [3] F. Heredia L.; J.J. González; I. Espinosa E. y R. Lara V., "Estudios sobre Memoria Asociativa Bidireccional utilizando dos simuladores diferentes de Redes Neuronales," Memorias de la Quinta Conferencia Internacional: Las Computadoras en las Instituciones de Educación e Investigación CA-UNAM-UNISYS, Nov. 1989.
- [4] J. J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," Proc. Nat. Acad. Sci., U.S., vol. 79, pp. 2554-2558, Apr. 1982.
- [5] T. Kohonen y M. Ruohonen, "Representation of Associated Data by Matrix Operators", IEEE Trans. Comput., vol. C-22, no. 7, July 1973.
- [6] B. Kosko, "Adaptive Bidirectional Associative Memories," Appl. Opt., vol. 26, no. 23, pp. 4947-4960, Dec. 1, 1987.
- [7] B. Kosko, "Bidirectional Associative Memories," IEEE Trans. Syst., Man, Cybern., vol. SMC-18, no. 1, pp. 49-60, Jan./Feb. 1988.
- [8] K. Murakami y T. Aibara, "An Improvement on the Moore-Penrose Generalized Inverse Associative Memory", IEEE Trans. Syst., Man, Cybern., vol.SMC-17, no. 4, pp. 699-707, July/Aug. 1987.
- [9] G.S. Stiles y D.L. Denq, "A Quantitative Comparison of the Performance of Three Discrete Distributed Associative Memory Models," IEEE Trans. Syst., Man, Cybern., vol SMC-36, no. 3, pp. 257-263, Mar. 1987.
- [10] Y.-F. Wang, J.B. Cruz y J. H. Mulligan, "Two Coding Strategies for Bidirectional Associative Memory," IEEE Trans. Neural Networks, vol. 1, no. 1, pp. 81-92, Mar. 1990.

LIBROS

- [11] A. Albert, Regression and Moore-Penrose Pseudoinverse. New York: Academic, 1972.
- [12] Y.-H. Pao, Adaptive Pattern Recognition and Neural Networks. Addison Wesley, 1989.