

Clasificador Asociativo de Patrones Iterativo (CAPI)

Sergio Valadez¹, Raúl Santiago¹ y Humberto Sossa²

¹ División de Estudios de Posgrado e Investigación-Instituto Tecnológico de León,
Av. Tecnológico s/n, León, Gto., México
svaladezg@gmail.com, rsantiago66@gmail.com

² Centro de Investigación en Computación-IPN,
Av. Juan de Dios Bátiz s/n, México, D.F., 07738. México
hsossa@cic.ipn.mx

Paper received on 21/08/12, Accepted on 07/09/12.

Resumen. En este trabajo se presenta una hibridación del llamado Clasificador Asociativo de Patrones (CAP) y la Memoria Asociativa Bidireccional (BAM). La nueva propuesta (CAPI) permite mejores desempeños de clasificación. No requiere el ajuste de los parámetros manualmente. Es capaz de clasificar correctamente un patrón en muy pocas iteraciones. Además, bajo ciertas condiciones, puede clasificar correctamente todos los patrones de entrenamiento.

Palabras Clave: Clasificación de patrones, Iteración, Memorias Asociativas.

1 Introducción.

Las *clases* son estados asociadas con conceptos [1]. El conjunto de todas las clases es conocido como el *espacio de interpretación* [2]. Los rasgos mediante los cuales se caracteriza a los objetos son conocidos como el *espacio de representación*. Entonces, el problema de la clasificación de patrones es encontrar un algoritmo que permita dividir el espacio de interpretación en diferentes regiones y así separar los patrones conocidos y clasificar los patrones desconocidos [3, 4, 5, 6]. Una manera de hacer esto es mediante el uso de memorias asociativas (MAS), las cuales permiten la clasificación de patrones mediante la asociación de los mismos con una clase [7]. Una memoria asociativa (MA) es una red neuronal artificial (RNA) de una sola capa la cuál mapea un conjunto de patrones de entrada \mathbf{x}^k hacia un conjunto de patrones de salida \mathbf{y}^k de tal manera que cada patrón \mathbf{x}^k es asociado con un patrón \mathbf{y}^k [8]. Una memoria M puede representarse mediante el esquema: $\mathbf{x}^k \rightarrow M \rightarrow \mathbf{y}^k$. En este caso M es una matriz de correlación de p asociaciones [9], cuyo *conjunto fundamental* se representa como:

$$S = \{(\mathbf{x}^k, \mathbf{y}^k) | k = 1, 2, \dots, p\} \quad (1)$$

En el *proceso de construcción* de la memoria M , cada par $(\mathbf{x}^k, \mathbf{y}^k) \in S$ es presentado a la MA. En el *proceso de recuperación* de la memoria M , un patrón de entrada \mathbf{x}^ω se presenta a la entrada de la memoria. Si $m = n$ entonces la memoria es *autoasociativa* y de otro modo es *heteroasociativa* [10].

En [11, 12, 13] se propone un algoritmo llamado *Clasificador Asociativo de Patrones* (CAP) basado en memorias asociativas. Combina la regla de construcción del *Asociador Lineal* (AL) de Anderson [14], Kohonen [10] y Nakano [15] y la regla de recuperación de la *Lernmatrix* (LM) [16, 17]. La nueva propuesta permite la operación con números reales, eliminando la desventaja de la LM de operar solo en el conjunto de los números binarios. También elimina la restricción de ortogonalidad del conjunto de entrenamiento del AL [18] así como la restricción de que el número de los patrones del conjunto de entrenamiento deba ser pequeño con respecto a la dimensión de los patrones de entrada [19, 20]. Además, se mantiene estable en su rendimiento de clasificación al ser entrenado con al menos el 10% del total de patrones de una base de datos dada [21].

La Memoria Asociativa Bidireccional (BAM) es una memoria heteroasociativa de dos capas que procesa información hacia adelante y hacia atrás para producir una búsqueda de los patrones almacenados [22, 23]. Cuando las neuronas de la memoria son activadas, la red evoluciona a un estado estable de dos patrones, un patrón en la salida de cada capa. La evolución de la BAM se detiene cuando ésta llega a un mínimo local de energía. La BAM se diferencia del AL y de la LM por que procesa información a través del tiempo de manera bidireccional, aunque también son utilizadas para almacenar pares de asociación. El funcionamiento de la BAM está basado principalmente sobre el funcionamiento de la memoria de Hopfield [24].

En el presente trabajo se propone una hibridación del CAP con la BAM. Esta hibridación permite obtener mejores resultados de recuperación que con el CAP solo. En forma iterada, los parámetros son ajustados automáticamente. A pesar del proceso iterativo, nuestra propuesta es capaz de clasificar correctamente un patrón en muy pocas iteraciones. Además, bajo ciertas condiciones, el nuevo clasificador puede clasificar correctamente todos los patrones de entrenamiento.

2 Estado del arte.

Son muy pocos los trabajos sobre clasificación de patrones mediante memorias asociativas y procesos de clasificación dinámicos. Esto debe en gran medida a que la meta de las memorias asociativas es el reconocimiento de patrones y no el de la clasificación [25]. Por ejemplo, en [26] se propone el uso de las memorias asociativas alfa-beta [27] combinadas con la BAM (sin el proceso iterativo). Esta combinación permite clasificar pacientes que han presentado una reaparición de cáncer. En [28] la BAM es utilizada para clasificar pacientes con cáncer, al ser probada para diferentes umbrales de la función de activación y después de realizar un proceso de normalización y discretización de los patrones. En [29] la BAM es utilizada para clasificar imágenes binarias de caracteres manuscritos y de diodos encendidos.

3 Diseño del Clasificador Asociativo de Patrones Iterativo.

Dado un conjunto de patrones asociados:

$$S = \{(\mathbf{x}^k, \mathbf{y}^k) | k = 1, 2, \dots, p\} \quad (2)$$

Donde p es la cardinalidad del conjunto fundamental de asociaciones, $\mathbf{x}^k \in \mathbb{R}^n$ es el conjunto de patrones de entrada y $\mathbf{y}^k \in \{-1, 1\}^m$ es el conjunto de clases; n es la dimensión de \mathbf{x}^k y m es la dimensión de \mathbf{y}^k . La clase $c \in \{1, 2, \dots, m\}$ a la que pertenece cada patrón de entrada \mathbf{x}^k se define mediante:

$$y_j^k = \begin{cases} 1 & \text{para } j = c \\ -1 & \text{para } j = 1, 2, \dots, c-1, c+1, \dots, m \end{cases} \forall k \in \{1, 2, \dots, p\} \quad (3)$$

3.1 Mecanismo de hibridación.

Debido a que el CAP funciona sobre los números reales y la BAM sobre los bipolares, se plantea un mecanismo de bipolarización de patrones reales a bipolares con el objetivo de realizar la hibridación. El proceso es como sigue:

1. Se calcula un vector de puntos de corte que servirá para bipolarizar los centroides de cada clase:

$$\theta = \frac{1}{p} \sum_{k=1}^p \mathbf{x}^k \quad (4)$$

2. Se calcula el ξ -ésimo centroide de cada clase para $\xi \in \{1, 2, \dots, m\}$:

$$\bar{\mathbf{x}}^\xi = \frac{1}{p_\xi} \sum_{j=1}^{p_\xi} \mathbf{x}^j \quad (5)$$

Donde $\bar{\mathbf{x}}^\xi$ es el ξ -ésimo centroide, p_ξ es el número de patrones de la ξ -ésima clase y $\mathbf{x}^j \forall j \in \{1, 2, \dots, p_\xi\}$ es el conjunto de patrones que forman la ξ -ésima clase.

3. Los centroides $\bar{\mathbf{x}}^\xi$ se bipolarizan mediante:

$$x_i^b = \begin{cases} -1 & \text{si } \bar{x}_i^\xi < \theta_i \\ 1 & \text{si } \bar{x}_i^\xi \geq \theta_i \end{cases} \forall i \in \{1, 2, \dots, n\} \quad (6)$$

Donde x_i^b es el i -ésimo elemento bipolarizado del centroide, \bar{x}_i^ξ es el i -ésimo componente del centroide y θ_i es el i -ésimo componente del vector de puntos de corte. De esta manera se obtiene un nuevo conjunto de patrones asociados:

$$S = \{(\mathbf{x}^b, \mathbf{y}^b) | b = 1, 2, \dots, p\} \quad (7)$$

donde p es la cardinalidad del conjunto fundamental de asociaciones, $\mathbf{x}^b \in \{-1, 1\}^n$ es el conjunto de centroides bipolarizados y $\mathbf{y}^k \in \{-1, 1\}^m$ es el conjunto de clases; n es la dimensión de \mathbf{x}^k y m es la dimensión de \mathbf{y}^k .

3.2 Fase de construcción.

La fase de construcción de la matriz de entrenamiento M se realiza de la misma manera que en el CAP, pero ahora sobre el conjunto fundamental bipolarizado S como sigue:

$$M = \sum_{b=1}^m \mathbf{y}^b [\mathbf{x}^b]^t \quad (8)$$

3.3 Fase de recuperación.

Sea $\mathbf{x}^\omega \in \mathbb{R}^n$ un patrón de entrada de dimensión n . El patrón de salida $\mathbf{y}^\omega \in \{-1, 1\}^m$ recuperado a partir de \mathbf{x}^ω se determina mediante los siguientes pasos:

1. El patrón de entrada \mathbf{x}^ω se bipolariza mediante la siguiente ecuación para obtener así un nuevo patrón bipolar \mathbf{x}^b :

$$x_i^b = \begin{cases} -1 & \text{si } x_i^\omega < \theta_i \\ 1 & \text{si } x_i^\omega \geq \theta_i \end{cases} \quad \forall i \in \{1, 2, \dots, n\} \quad (9)$$

Donde x_i^b es el i -ésimo elemento bipolarizado del patrón, x_i^ω es el i -ésimo componente del patrón y θ_i es el i -ésimo componente del vector de puntos de corte.

2. Se define un patrón de clase $\mathbf{y}^\omega = \mathbf{0}$.
3. Se define $\mathbf{x}^b(t) = \mathbf{x}^b$ y $\mathbf{y}^b(t) = \mathbf{y}^\omega$ para $t=0$.
4. El patrón $\mathbf{x}^b(t)$ se aplica en la entrada de la primer capa mediante:

$$y_i^b(t+1) = \begin{cases} 1 & > 0 \\ y_i^b(t) & \text{si } \sum_{j=1}^n m_{ij} x_j^b(t) = 0 \\ -1 & < 0 \end{cases} \quad (10)$$

Donde $y_i^b(t+1)$ es el i -ésimo componente del patrón de salida en el instante de tiempo siguiente, $y_i^b(t)$ es el i -ésimo componente del patrón de salida en el instante de tiempo actual, m_{ij} es el ij -ésimo elemento de la matriz de pesos M y $x_j^b(t)$ es el j -ésimo componente del patrón de entrada en el instante de tiempo actual.

5. Suponiendo que las actualizaciones en la ecuacion (10) son síncronas, y el vector \mathbf{y} se propaga hacia la segunda capa, este se procesa usando la matriz transpuesta M^t mediante:

$$x_j^b(t+1) = \begin{cases} 1 & > 0 \\ x_j^b(t) & \text{si } \sum_{i=1}^m m_{ij}y_i^b(t+1) = 0 \\ -1 & < 0 \end{cases} \quad (11)$$

Donde $x_j^b(t+1)$ es el j -ésimo componente del patrón de salida en el instante de tiempo siguiente, $x_j^b(t)$ es el j -ésimo componente del patrón de salida en el instante de tiempo actual, m_{ij} es el ij -ésimo elemento de la matriz de pesos M y $y_i^b(t+1)$ es el i -ésimo componente del patrón de entrada calculado en el paso 4.

La secuencia de recuperación se repite a través de los pasos 4 y 5 hasta que no hay más actualizaciones de los patrones de salida \mathbf{x} y \mathbf{y} .

3.4 Modos de actualización.

Para el caso de una actualización síncrona, todas las neuronas del vector de salida de las ecuaciones (10) y (11) son actualizadas simultáneamente. Para el caso de una actualización asíncrona solo una neurona del vector de salida es actualizada de manera aleatoria. Por lo tanto solo una componente del vector de salida es calculada, según la salida correspondiente.

4 Evaluación del Clasificador Asociativo de Patrones Iterativo.

En esta sección se presentan los resultados obtenidos al comparar los dos algoritmos: el CAP y el CAPI. Este último en los modos de actualización síncrona y asíncrona. Para identificar en que modo de actualización se está probando el algoritmo propuesto, se define como Clasificador Asociativo de Patrones Iterativo Síncrono-CAPIS cuando se utiliza una actualización síncrona y Clasificador Asociativo de Patrones Iterativo Asíncrono-CAPIA cuando se utiliza una actualización asíncrona.

4.1 Descripción de los datos de prueba.

Cada una de las bases de datos que se utilizaron en la experimentación se obtuvo del repositorio de aprendizaje de máquina de la Universidad de California Irvine (UCI) [30]. En la Tabla 1 se presenta un resumen de cada una de ellas: *Wisconsin Breast Cancer Database*, *Pima Indians Diabetes Database*, *Haberman's Survival Data*, *Heart Disease Dataset*, *Hepatitis Domain*, *Johns Hopkins University Ionosphere database*, *Chess End-Game - King+Rook versus King+Pawn on a7*, *BUPA liver disorders* y *Sonar*, *Mines vs. Rocks*.

Tabla 1. Resumen de las bases de datos utilizadas para la experimentación.

No.	Base de datos	Instancias			Atributos	Tipo de los atributos
		Total	Clase 1	Clase 2		
1	Breast	699	458	241	9	Numérico
2	Diabetes	768	500	268	8	Numérico
3	Haberman	306	225	81	3	Numérico y nominal
4	Heart	270	120	150	13	Numérico
5	Hepatitis	155	32	123	19	Numérico y nominal
6	Ionosphere	351	126	225	34	Numérico
7	KingRook	3196	1669	1527	36	Nominal
8	Liver	345	145	200	7	Numérico
9	Sonar	208	111	97	60	Numérico

Los valores nominales de las bases de datos Hepatitis y KingRook fueron pre-procesados. En la base de datos Hepatitis, los valores “yes” fueron asignados numéricamente como 1 y los valores “no” como 2, mientras que “female” como 1 y “male” como 2. En la base de datos KingRook, los valores “f” se reemplazaron por 1, los valores “t” por 2, los valores “l” por 1, los valores “g” por 2, los valores “b” por 1, los valores “n” por 2 y los valores “w” por 3.

4.2 Método de validación de resultados.

Para la obtención de los resultados se utilizó el método k -Folds de validación cruzada para $k = 10$. Cada una de las bases de datos se dividió en diez partes de aproximadamente el mismo tamaño. En cada una de las diez partes, las clases fueron representadas en las mismas proporciones, es decir, se hizo una partición estratificada. Además, las instancias de cada una de las diez partes fueron seleccionadas de forma aleatoria. Cada parte se utilizó como conjunto de prueba exactamente una vez y la unión de las nueve particiones restantes se utilizó como conjunto de entrenamiento. Con esto se obtuvieron diez pares prueba-entrenamiento, los cuales fueron pasados a cada uno de los tres clasificadores. Este experimento se repitió diez veces.

4.3 Análisis del desempeño.

Los resultados de los experimentos realizados se muestran en la Tabla 2. Se pueden observar el rendimiento promedio con intervalo de confianza (μ_1 para el CAP, μ_2 para el CAPIS y μ_3 para el CAPIA,) y la desviación estándar (μ_1 para el CAP, μ_2 para el CAPIS y μ_3 para el CAPIA). En negrita aparecen los mejores rendimientos de clasificación.

Para saber si existe diferencia entre el rendimiento de un algoritmo y otro se utiliza la prueba de ranqueo de signos con suma de Wilcoxon [5]. Esta prueba considera que las distribuciones de los datos no siguen una distribución normal. Primeramente se plantea la hipótesis nula de que el primer algoritmo tiene menor o igual rendimiento que el segundo y una hipótesis alternativa de que el primero tiene mayor rendimiento que el segundo. La hipótesis nula se rechaza cuando $W^- < W^+$. En las Tablas 3-5 se muestra el desarrollo de esta prueba considerando los rendimientos promedio del CAP vs el CAPIS, del CAP vs el CAPIA y de CAPIS vs el CAPIA. En la Tabla 3 se puede observar que $W^- > W^+$ por lo tanto se acepta la hipótesis nula. Entonces el CAP tiene menor o igual rendimiento que el CAPIS. En la Tabla 4 se

puede observar que $W^- < W^+$ por lo tanto se rechaza la hipótesis nula. Entonces el CAP tiene mayor rendimiento que el CAPIA. En la Tabla 5 se puede observar que $W^- > W^+$ por lo tanto se rechaza la hipótesis nula. Entonces el CAPIS tiene mayor rendimiento que el CAPIA.

4.4 Iteraciones promedio.

En la Tabla 7 se muestran las iteraciones promedio obtenidas para el CAPIS y CAPIA al recuperar (clasificar) un patrón dado, considerando las mismas pruebas con la que se obtuvieron los resultados de la Tabla 1.

Tabla 2. Rendimientos de clasificación promedio del CAP, CAPIS y CAPIA.

No.	Base de datos	Clasificadores						Promedio
		CAP		CAPIS		CAPIA		
		μ_1	σ_1	μ_2	σ_2	μ_3	σ_3	
1	Breast	97.15±0.02	0.03	93.25±0.04	0.06	86.23±0.3	0.49	92.21
2	Diabetes	61.78±0.08	0.13	66.91±0.13	0.22	59.09±0.33	0.53	62.59
3	Haberman	66.60±0.49	0.80	59.04±0.52	0.83	63.10±0.78	1.25	62.91
4	Heart	63.54±0.11	0.18	81.75±0.19	0.31	73.08±0.32	0.52	72.79
5	Hepatitis	54.07±0.55	0.88	80.01±0.3	0.48	78.67±0.59	0.94	70.92
6	Ionosphere	68.86±0.08	0.12	80.13±0.24	0.39	77.42±0.34	0.55	75.47
7	KingRook	82.62±0.02	0.04	66.02±0.17	0.27	63.52±0.14	0.22	70.72
8	Liver	54.62±0.16	0.26	51.89±0.18	0.29	51.83±0.31	0.50	52.78
9	Sonar	67.94±0.78	1.25	71.54±0.65	1.05	65.22±0.44	0.71	68.23
	Promedio	68.58		72.28		68.68		

Tabla 3. Prueba de ranqueo de signos con suma de Wilcoxon para el CAP y el CAPIS.

No.	Base de datos	CAP	CAPIS	CAP-CAPIS	CAP-CAPIS	Rango	W^-	W^+
1	Breast	97.15	93.25	3.90	3.90	3	0	3
2	Diabetes	61.78	66.91	-5.13	5.13	4	4	0
3	Haberman	66.60	59.04	7.56	7.56	5	0	5
4	Heart	63.54	81.75	-18.21	18.21	8	8	0
5	Hepatitis	54.07	80.01	-25.94	25.94	9	9	0
6	Ionosphere	68.86	80.13	-11.27	11.27	6	6	0
7	KingRook	82.62	66.02	16.60	16.60	7	0	7
8	Liver	54.62	51.89	2.74	2.74	1	0	1
9	Sonar	67.94	71.54	-3.60	3.60	2	2	0
						Suma	29	16

Tabla 4. Prueba de ranqueo de signos con suma de Wilcoxon para el CAP y el CAPIA.

No.	Base de datos	CAP	CAPIA	CAP-CAPIA	CAP-CAPIA	Rango	W^-	W^+
1	Breast	97.15	86.23	10.92	10.92	7	0	7
2	Diabetes	61.78	59.09	2.69	2.69	1	0	1
3	Haberman	66.60	63.10	3.51	3.51	4	0	4
4	Heart	63.54	73.08	-9.54	9.54	6	6	0
5	Hepatitis	54.07	78.67	-24.60	24.60	9	9	0
6	Ionosphere	68.86	77.42	-8.56	8.56	5	5	0
7	KingRook	82.62	63.52	19.10	19.10	8	0	8
8	Liver	54.62	51.83	2.79	2.79	3	0	3
9	Sonar	67.94	65.22	2.72	2.72	2	0	2
						Suma	20	25

Tabla 5. Prueba de ranqueo de signos con suma de Wilcoxon para el CAPIS y el CAPIA.

No.	Base de datos	CAPIS	CAPIA	CAPIS-CAPIA	CAPIS-CAPIA	Rango	W ⁻	W ⁺
1	Breast	93.25	86.23	7.02	7.02	7	0	7
2	Diabetes	66.91	59.09	7.83	7.83	8	0	8
3	Haberman	59.04	63.10	-4.06	4.06	5	5	0
4	Heart	81.75	73.08	8.67	8.67	9	0	9
5	Hepatitis	80.01	78.67	1.34	1.34	2	0	2
6	Ionosphere	80.13	77.42	2.70	2.70	4	0	4
7	KingRook	66.02	63.52	2.50	2.50	3	0	3
8	Liver	51.89	51.83	0.05	0.05	1	0	1
9	Sonar	71.54	65.22	6.32	6.32	6	0	6
Suma							5	40

Tabla 6. Iteraciones promedio del CAPIS y CAPIA al recuperar un patrón dado.

No.	Base de datos	CAPIS		CAPIA	
		μ	σ	μ	σ
1	Breast	2.00±0.00	0.00	2.61±0.00	0.01
2	Diabetes	1.83±0.00	0.00	2.46±0.01	0.01
3	Haberman	2.00±0.00	0.00	2.62±0.01	0.02
4	Heart	1.99±0.00	0.00	2.86±0.01	0.02
5	Hepatitis	2.00±0.00	0.00	2.91±0.04	0.06
6	Ionosphere	2.00±0.00	0.00	3.10±0.01	0.02
7	KingRook	1.90±0.00	0.00	3.00±0.01	0.01
8	Liver	1.63±0.01	0.01	2.11±0.02	0.03
9	Sonar	1.94±0.00	0.00	3.04±0.02	0.03
Promedio		1.92		2.75	

5 Conclusiones.

El CAPI funciona de manera autónoma y automática. Trabaja en dos fases: de construcción y de recuperación. En la fase de construcción se obtienen los vectores centroides de cada clase y se bipolarizan mediante el método propuesto de discretización de valores continuos. Posteriormente estos centroides son utilizados para construir la memoria del clasificador. En la fase de recuperación, el patrón presentado a la memoria, previamente construida, también se bipolariza. Esta fase funciona en modos de actualización síncrono (CAPIS) y asíncrono (CAPIA). El CAPI es entrenado con los centroides bipolarizados de cada clase de manera supervisada. Es decir, el mapeo que se hace entre el espacio de representación y el espacio de interpretación, se hace de forma inductiva.

El CAP por ser un sistema de recuperación estático realiza una recuperación en una sola iteración. El CAPI, en ambos modos de recuperación, permite obtener una solución en más de una iteración. Para el CAPIS es de 1.92 y para el CAPIA de 2.75 iteraciones promedio.

Las evaluaciones indican que el CAPIS presenta mejor desempeño sobre el CAP y que el CAP sobre el CAPIA. Por lo tanto, el proceso de recuperación iterativo es capaz de incrementar el rendimiento de clasificación.

Como trabajo futuro se sugiere 1) Probar con algún otro método de discretización de valores continuos, 2) Probar la construcción de la memoria del CAPI con el total de patrones de entrenamiento, no solamente centroides. Implementar una extensión al CAPI para procesos multiclase, 3) Evaluar el CAPI con un número mayor de bases de datos, 4) Comparar el CAPI con otros algoritmos clasificadores, entre otros.

Reconocimientos

S. Valadez agradece al CONACYT por la beca recibida, a la DGEST por el apoyo recibido (4574.12-P), al Insitituto Tecnológico de León y al CIC-IPN para realizar los estudios de posgrado. R. Santiago agradece a la DGEST, y al Insitituto Tecnológico de León. H. Sossa agradece al CONACYT y a la SIP-IPN por el apoyo económico en el marco de los proyectos 155014 y 20121311. Agradece también al ICYTDF por el apoyo brindado en el marco del proeycto: Control inteligente de robots humanoides y robots aéreos y aplicaciones.

Referencias

1. J. P. Marqués de Sá, Pattern Recognition, Concepts, Methods, and Applications, Springer-Verlag, 2002.
2. A. R. Webb, Statistical Pattern Recognition, 2nd Edition, 2 ed., {John Wiley \& Sons}, 2002.
3. T. M. Mitchell, Machine Learning, McGraw-Hill, 1997.
4. K. Fukunaga, Introduction to statistical pattern recognition, 2 ed., Academic Press Professional, Inc., 1990
5. E. Alpaydin, Introduction to Machine Learning (Adaptive Computation and Machine Learning), 2 ed., The MIT Press, 2010.
6. M. Friedman and A. Kandel, Introduction to Pattern Recognition : Statistical, Structural, Neural and Fuzzy Logic Approaches (Series in Machine Perception and Artificial Intelligence), World Scientific Publishing Company, 1999.
7. J. M. Zurada, Introduction to Artificial Neural Systems, West Publishing Company, 1992.
8. G. Palm, F. Schwenker, F. T. Sommer and A. Strey, "Neural Associative Memories," Biological Cybernetics, vol. 36, pp. 36-19, 1993
9. B. Prasad, P. E. S, Y. Sagar and P. S. Rama, "A Study on Associative Neural Memories," International Journal of Advanced Computer Sciences and Applications, vol. 1, no. 6, pp. 124-133, 2011.
10. T. Kohonen, "Correlation Matrix Memories," IEEE Transactions on Computers, vol. 21, pp. 353-359, 1972.
11. R. Santiago, C. Yañez and J. L. Diaz-de-Leon, "Clasificador hibrido de patrones basado en la Lernmarix de Steinbuch y el Linear Associator de Anderson-Kohonen," Research on Computing Science. Reconocimiento de patrones, avances y perspectivas. pp. 449-460, 2002.
12. R. Santiago-Montero, "Clasificador hibrido de patrones basado en la Lernmatrix de Steinbuch y el Linear Associator de Anderson-Kohonen," 2003.
13. R. Santiago-Montero, C. Yañez-Marquez and J. L. Diaz-de-Leon, "Clasificador Asociativo de Patrones: Avances Teoricos," Laboratorio de reconocimiento de patrones. Centro de Investigacion en Computacion, IPN, Mexico, 2002.
14. J. A. Anderson, "A simple neural network generating an interactive memory," Mathematical Biosciences, vol. 14, no. 3-4, pp. 197-220, 1972.
15. K. Nakano, "Associatron-A model for associative memory," IEEE Transactions on Systems, Man and Cybernetics, Vols. SMC-2,3, pp. 380-388, 1972.
16. K. Steinbuch, "Die Lernmatrix," Kybernetic, vol. 1, pp. 36-45, 1961.
17. K. Steinbuch and H. Frank, "Nichtdigitale Lernmatrizen als Perzeptoren," Kybernetics, vol. 1, no. 3, pp. 117-124, 1963
18. M. H. Hassoun, Fundamentals of Artificial Neural Networks, 1st ed., MIT Press, 1995.

19. E. Rosenfeld and J. A. Anderson, *Neurocomputing : foundations of research* / edited by James A. Anderson and Edward Rosenfeld, Cambridge, Mass : MIT Press, 1988.
20. M. H. Hassoun, *Associative Neural Memories: Theory and Implementation*, M. H. Hassoun, Ed., New York: Oxford University Press, 1993
21. J. A. Soria-Alcaraz, R. Santiago-Montero and M. Carpio, "One criterion for the selection of the cardinality of learning set used by the Associative Pattern Classifier," *Proceedings of the 2010 IEEE Electronics, Robotics and Automotive Mechanics Conference*, pp. 80-84, 2010
22. B. Kosko, "Adaptive didirectional associative memories," *Appl. Opt.*, vol. 26, no. 23, pp. 4947-4960, 1987
23. B. Kosko, "Bidirectional Associative Memories," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 18, no. 1, pp. 49-60, 1988.
24. J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554-2558, 1982
25. R. Duda, P. Hart and D. Stork, *Pattern classification*, 2nd Edition, Wiley Interscience, New York, 2001
26. M. E. Acevedo, M. A. Acevedo and F. Felipe, "Classification of Cancer Recurrence with Alpha-Beta BAM," *Mathematical Problems in Engineering*, p. 14, 2009.
27. C. Yañez-Márquez, "Memorias asociativas basadas en relaciones de orden y operaciones binarias," *Computación y Sistemas. CIC, IPN.*, vol. 6, no. 4, pp. 300-311, 2003.
28. Z. Zhang, H. Zhuo, S. Liu and P. d. B., "Classification of cancer patients based on elemental contents of serums using bidirectional associative memory networks," *Analytica Chimica Acta*, vol. 436, pp. 281-291, 2001.
29. F. J. López-Aligüé, I. Acevedo-Sotoca, I. Álvarez-Troncoso, C. García-Orellana and H. González-Velasco, "A Neural Associative Pattern Classifier," *Advances in Artificial Intelligence - IBERAMIA 2002*, 8th Ibero-American Conference on AI, pp. 430-439, 2002.
30. A. Frank and A. Asuncion, {UCI} Machine Learning Repository, 2010