

# APRENDIZAJE CON REDES NEURONALES ARTIFICIALES

Francisco Javier Gómez Quesada  
Miguel Ángel Fernández Graciani  
María Teresa López Bonal  
María Alonso Díaz-Marta

*Francisco Javier Gómez Quesada, Miguel Ángel Fernández Graciani y, María Teresa López Bonal, María Alonso Díaz-Marta son profesores del Departamento de Informática, Escuela Universitaria Politécnica de Albacete, Universidad de Castilla - La Mancha.*

*Francisco Javier Gómez Quesada es Licenciado en Informática por la Universidad de Granada.*

*Miguel A. Fernández Graciani es Licenciado en Ciencias Físicas por la Universidad de Granada.*

*María Teresa López Bonal es Licenciada en Ciencias Físicas por la Universidad de Valencia.*

*María Alonso Díaz-Marta es Licenciada en Informática por la Universidad Politécnica de Valencia*

## RESUMEN

La gran eficacia de la computación biológica, dado que el cerebro es excepcional en lo referente a capacidad de resolución de problemas aleatorios y en el procesamiento de información incompleta, estocástica o incluso contradictoria, induce a pensar en la posibilidad de lograr capacidades similares en dispositivos artificiales basados en los mismos principios de diseño que los sistemas neuronales. La idea de poder construir algún día una máquina que imite el cerebro humano es el gran sueño. Sin embargo, tanto limitaciones teóricas como de hardware han dado hasta ahora relativamente pocos progresos en la materia. Las redes neuronales se presentan como un sistema ideal para resolver computacionalmente este tipo de problemas.

## 1. LIMITACIONES DE LOS COMPUTADORES CONVENCIONALES

LOS ordenadores digitales modernos son novicios en el mundo de la computación. Desde hace millones de años existen ordenadores biológicos, como el cerebro y el sistema nervioso de humanos y animales, que muestran una maravillosa eficacia en el procesamiento de la información sensorial y en el control de las interacciones con el en-

torno. Echar mano a un bocadillo, reconocer un rostro o recordar cosas relacionadas con el sabor de las magdalenas son tareas computacionales, lo mismo que lo son el tratamiento matemático de grandes cantidades de números o el desarrollo de un videojuego. Pero, ¿por qué un niño de tres años es capaz de señalar con suma facilidad un árbol en una fotografía y, por contra, programas muy complejos y perfeccionados, instalados en los más poderosos superordenadores, apenas si logran resultados mediocres al acometer una tarea esencialmente equivalente: el reconocimiento de figuras? Lo paradójico de esta situación es que los ordenadores pueden obtener rápidamente la solución de problemas que desbordarían la capacidad del cerebro humano. Más aún, una simple calculadora de bolsillo supera al cerebro humano en tareas como la de efectuar el producto de dos números de diez cifras. ¿En qué reside la diferencia entre la multiplicación de números y el reconocimiento de objetos, para que cueste tanto lograr la segunda en los ordenadores? Con otras palabras, ¿por qué es tan difícil hacer que un ordenador reconozca un árbol?

En última instancia, la respuesta a este tipo de cuestiones radica en que los problemas de identificación de figuras no admiten una definición formal y compacta. Para reconocer árboles se requiere una definición muy amplia de la naturaleza del árbol, que equivaldría a una descripción de todas las variantes concebibles. Los problemas de la clase de los planteados por el reconocimiento de figuras constituyen un subconjunto de los llamados problemas aleatorios, es decir, problemas cuya resolución exige, en esencia, un conocimiento de todos los posibles estados del sistema. Por consiguiente, la resolución de un problema aleatorio comporta tener registrados en la memoria la totalidad de posibles estados, más la facultad de seleccionar rápidamente la más idónea de las soluciones entre el conjunto de las registradas, habida cuenta de los datos disponibles. Por el contrario, la solución de una tarea computacional típica (multiplicación de números) puede quedar sucintamente expresada mediante un algoritmo, o sea, una secuencia de instrucciones perfectamente especificadas, que definan de qué forma han de manipularse los datos para llegar a una solución.

Inicialmente, los computadores fueron construidos para ejecutar operaciones en serie. Esta aproximación «Von Neumann» se extendió con las técnicas de pipeline, arrays de procesadores y multiprocesamiento. Todas estas propuestas están basadas en lo que podríamos denominar «Arquitecturas Algorítmicas», o sea, los problemas se resuelven mediante algoritmos que son directamente transformados en instrucciones máquina. Los pasos que se siguen son siempre los mismos: extracción, decodificación y ejecución de la instrucción. Los resultados son precisos y previsibles.

En general, estas arquitecturas son muy competentes y adecuadas para ejecutar de manera mecánica las instrucciones de un algoritmo

(operaciones aritmético-lógicas), pero no pueden rivalizar con la capacidad de memorización y recuperación de recuerdos del cerebro humano, que con regularidad y sin esfuerzo va resolviendo y superando los problemas que se le plantean. Los procesadores en paralelo tienen todavía reducida su capacidad para resolver importantes problemas de tiempo real en áreas como visión, reconocimiento de patrones, reconocimiento del habla, optimización, memoria asociativa, etc.

Algo pues, debe estar equivocado en la forma en que estos problemas son enfocados. Esta opinión se justifica en algunos ejemplos, concretamente en la naturaleza de los sistemas biológicos que son capaces de dar soluciones en tiempo real a problemas, más allá de la capacidad del supercomputador más grande. Y este proceso ocurre por el uso de elementos biológicos que son un millón de veces más lentos que los dispositivos semiconductores.

## 2. EL PARADIGMA DE LA RED NEURONAL

Las redes neuronales artificiales representan un cambio fundamental en el funcionamiento y la arquitectura del computador. Después de muchos años en los que el interés por las redes neuronales ha estado un poco apagado, varios factores se han combinado recientemente para reanudar y avivar dicho interés:

1. Los progresos en el hardware permiten ahora el desarrollo económico de grandes computadores en paralelo.
2. Las teorías sobre el cálculo colectivo en las redes neuronales han madurado gracias a la investigación continuada de un número de consagrados investigadores.
3. Después de muchos años de investigación en Inteligencia Artificial, están claros qué tipos de problemas se adaptan mejor a estos planteamientos simbólicos. Las redes neuronales aparecen para complementar los métodos convencionales usados en Inteligencia Artificial, dando soluciones a problemas que requieren búsquedas asociativas, reconocimiento de patrones, clasificación y optimización de datos incompletos o imprevisibles, etc.

Algunas de las principales características que distinguen a las redes neuronales de la informática convencional son:

- Fuerte paralelismo: en gran escala las redes neuronales pueden tener cientos de miles de celdas (células) individuales de proceso.
- Elevada intercomunicación: algunos modelos requieren interconexiones específicas entre cada procesador. Otros modelos pueden organizarse en capas o niveles que requieren menos interconexiones. El número de conexiones depende de la aplicación, del modelo teórico y del resultado deseado.

- Representación distribuida: la fortaleza de las conexiones se altera para añadir nueva información. Los datos se almacenan en un patrón de conexiones que se extiende por toda la red.
- Tolerancia a fallos: si una neurona falla, el sistema sigue funcionando. La precisión de la red vendrá dada por el número de neuronas que fallen; cuantas más fallen menos precisión tendrá.
- Cálculo colectivo: una red neuronal no ejecuta instrucciones particulares, es decir, todas las celdas de la red están preparadas para resolver colectivamente un problema particular.
- Auto-Organización: una red neuronal puede adaptar su estructura para descubrir nuevos patrones.

### 3. PERO, ¿QUÉ ES UNA RED NEURONAL ARTIFICIAL?

Una Neurona Artificial es un procesador elemental (PE) que recibe una serie de entradas (bien del exterior, bien de otras neuronas) con pesos diferentes, las procesa y proporciona una salida única. A cada neurona llegan muchas señales de otras (sinapsis) y producen una única salida (axon). Una sinapsis que comunica dos neuronas puede ser de naturaleza excitadora o inhibidora. En el primer caso, la neurona emisora tenderá a activar a la neurona receptora, y en el segundo caso la neurona emisora tenderá a inhibir la actividad de la neurona receptora. Cada sinapsis se caracteriza además por la eficacia con la que se establece la conexión. En definitiva, aunque la neurona toma decisiones en función de la suma de informaciones que recibe, la contribución de cada una de estas informaciones es ponderada por la eficacia de la sinapsis correspondiente.

La primera neurona artificial fue concebida por W. McCulloch y W. Pitts, de la Universidad de Chicago en 1943. Se trata de un modelo binario cuyo estado es 1 (activo) ó 0 (inactivo). Periódicamente actualiza su estado de la siguiente forma: calcula la suma de sus entradas (que son las salidas de las neuronas artificiales a las que está conectada) con el valor de cada entrada modulado por la eficacia sináptica correspondiente, y toma una decisión comparando esta suma con un cierto nivel o *umbral* ( $U$ ) ya prefijado. Si la suma es superior al umbral, la neurona se activa (1); en caso contrario se mantiene inactiva (0). Por tanto, todas las neuronas toman sus decisiones simultáneamente teniendo en cuenta la evolución del estado global de la red. En general se utiliza una *Función de Activación*  $Y = F(S)$  para activar una neurona artificial. Esta función de activación puede tener distintas formas y características. El utilizar funciones de activación no lineales tiene la ventaja de poder *normalizar* el valor de  $S$  para dar una salida  $Y$  dentro de un rango prefijado e independiente del valor de  $S$ . Normalmente

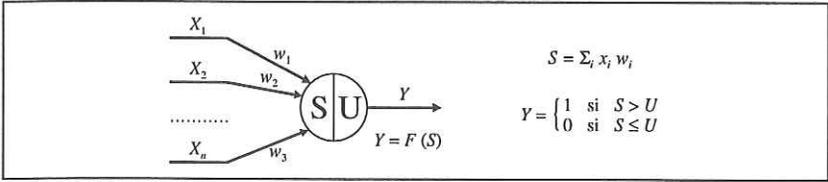


FIGURA 1

[0, 1] como hace la función Sigmoide ó [-1, 1] para la función Tangente Hiperbólica.

La sinapsis, el punto de conexión entre neuronas, gradúa la *intensidad* con que la señal pasa de una neurona a otra (Fig. 1).

Una Red Neuronal Artificial es un sistema o conjunto de procesadores elementales interconectados, no lineal ni estacionario, que realiza al menos alguna de las siguientes funciones: Aprendizaje, Memorización, Generalización o Abstracción de características esenciales.

La red neuronal más simple consta de una única *capa o nivel*, pudiendo estar formada por tantas como se desee y en donde las salidas de una capa están conectadas a las entradas de la siguiente. Los pesos son propios de cada neurona, y no se está obligado a tener todas las conexiones (alguna puede tener peso 0) (Fig. 2).

En un sentido amplio, una red neuronal consta de tres elementos principales:

1. Topología: cómo está una red neuronal organizada en capas y cómo se conectan esta capas.
2. Aprendizaje: cómo se almacena la información en la red.
3. Recuperación: cómo esta información almacenada en la red puede recuperarse.

El objeto de este trabajo es el proceso de aprendizaje, por lo que nos centraremos en él.

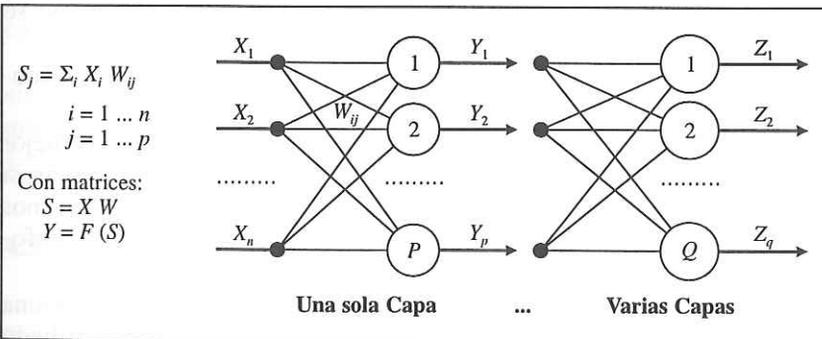


FIGURA 2

## 4. APRENDIZAJE DE UNA RED NEURONAL

Es con toda seguridad la característica más importante de una red neuronal. Durante este aprendizaje o entrenamiento de la red y por aplicación de un conjunto de entradas, se van ajustando adecuada e internamente todos y cada uno de los pesos asociados a cada rama para obtener la salida deseada (o al menos una salida consistente), de forma que la red pueda responder después por sí sola a situaciones diferentes a las *aprendidas*. Para casos sencillos, los pesos pueden asignarse de forma manual, pero lo usual es utilizar algún algoritmo para llevar a cabo este proceso de entrenamiento.

Este aprendizaje que tiene lugar en las redes neuronales es el factor que determina las ventajas (y también los inconvenientes) de estos sistemas. Si la red está bien ajustada, y gracias a la masiva operación en paralelo de cada nodo, será capaz de trabajar con información incompleta o difícil de predecir, teniendo un cierto grado de *memoria asociativa* que le permite generalizar su comportamiento ante una cierta entrada si esta entrada es razonablemente parecida a aquéllas para las que ha sido entrenada.

### 4.1. Aprendizaje Supervisado y Aprendizaje No Supervisado

Todos los métodos de aprendizaje pueden ser englobados en dos categorías:

1. **Aprendizaje Supervisado:** es un caso de entrenamiento Con Profesor y utiliza información global. Se presentan dos vectores (entrada y salida deseada). La salida computada por la red se compara con la salida deseada, y los pesos de la red se modifican en el sentido de reducir el error cometido. Se repite iterativamente, hasta que la diferencia entre salida computada y deseada sea aceptablemente pequeña. Con  $n$  parejas de este tipo se forma un *Conjunto de Entrenamiento*.

El aprendizaje Supervisado se suele dividir a su vez en dos sub-categorías:

- *Aprendizaje Estructural:* se refiere a la búsqueda de la mejor conexión o afinidad posible entrada/salida para cada pareja de patrones individuales. La mayor parte de los algoritmos de aprendizaje que veremos a continuación tienen un enfoque estructural.
- *Aprendizaje Temporal:* hace referencia a la captura de una serie de patrones necesarios para conseguir algún resultado final. En el aprendizaje temporal la respuesta actual de la red

depende de las entradas y respuestas previas. En el aprendizaje estructural no existe tal dependencia.

2. **Aprendizaje No Supervisado:** es un caso de entrenamiento Sin Profesor y sólo usa información local durante todo el proceso de aprendizaje. Es un modelo más cercano al sistema biológico, no se utiliza el vector de salida esperada, y sólo hay vectores de entrada en el conjunto de entrenamiento. El algoritmo modifica los pesos de forma que las salidas sean consistentes, es decir, que a entradas muy parecidas, la red compute la misma salida. Las salidas se asocian a las entradas de acuerdo con el proceso de entrenamiento. El proceso extrae características, abstractando las propiedades colectivas subyacentes del conjunto de entrenamiento, y agrupa por clases de similitudes.

Vamos ahora, a ver algunos de los tipos de aprendizaje más importantes:

## 4.2. Aprendizaje Hebbiano

Este tipo de aprendizaje propone que los pesos (fuerzas sinápticas) se incrementan si las neuronas origen y destino están activadas, es decir, refuerza los caminos usados frecuentemente en la red, lo que explicaría por ejemplo, los hábitos y el aprendizaje por repetición.

Los valores de los procesadores elementales y pesos son ilimitados. El aprendizaje hebbiano está matemáticamente caracterizado por la ecuación

$$w_{ij}^{\text{nuevo}} = w_{ij}^{\text{viejo}} + a_{ki} b_{kj} \quad (1)$$

donde  $i = 1, 2, \dots, n$ ;  $j = 1, 2, \dots, p$ ;  $w_{ij}$  es el peso de la conexión entre los dos PEs.

Redes neuronales como la memoria asociativa lineal emplea este tipo de aprendizaje. El número de patrones que una red adiestrada usando (1) con conexiones y pesos ilimitados puede producir está limitado por la dimensión de los patrones de entrada.

Si los valores de los Pes están limitados y los pesos ilimitados nos encontramos con las denominadas redes de Hopfield (Hopfield, 1982) que restringen el valor de los Pes a un valor binario  $\{0, 1\}$  o bipolar  $\{-1, +1\}$ . La ecuación (1) es utilizada para estos tipos de correlaciones.

## 4.3. Aprendizaje Competitivo

En su forma más simple, el aprendizaje competitivo usa inhibición lateral para activar una sola neurona (el ganador se lo lleva todo). Algunas redes neuronales que emplean aprendizaje competitivo son los

Mapas Auto-Organizativos (Kohonen, 1984) y ART (*Adaptive Resonance Theory*, Carpenter y Grossberg, 1987).

#### 4.4. Aprendizaje Min-Max

Un clasificador min-max usa un par de vectores para cada clase. La clase  $j$  está representada por el PE  $y_j$  y está definida por los vectores  $V_j$  (el vector min) y  $W_j$  (el vector max). El aprendizaje min-max en un sistema neuronal viene dado por la ecuación

$$v_{ij}^{\text{nuevo}} = \min (a_{ki}, v_{ij}^{\text{viejo}}) \quad (2)$$

para el vector min y

$$w_{ij}^{\text{nuevo}} = \max (a_{ki}, w_{ij}^{\text{viejo}}) \quad (3)$$

para el vector max.

#### 4.5. Aprendizaje de Corrección de Error

Este tipo de aprendizaje ajusta los pesos de conexión entre PEs en proporción a la diferencia entre los valores deseados y computados para cada PE de la capa de salida. Dependiendo del número de capas de la red podemos distinguir dos casos:

- a) **Red de dos capas:** puede capturar mapeos lineales entre las entradas y las salidas. Dos redes neuronales que utilizan este tipo de aprendizaje son el Perceptrón (Rosenblatt, 1962) y ADALINE (Widrow y Hoff, 1960).
- b) **Red multicapa:** puede capturar mapeos no lineales entre las entradas y las salidas. La versión multinivel de este algoritmo es comúnmente denominada Regla de Aprendizaje de Retropropagación de Errores o simplemente Retropropagación (Backpropagation). Utilizando la regla encadenada, podemos calcular los cambios de pesos para un número arbitrario de capas. El número de iteraciones que deben ser realizadas para cada patrón del conjunto de datos es grande, haciendo que este algoritmo de aprendizaje sea muy lento de entrenar. El algoritmo de Retropropagación que ha habido sido estudiado por Werbos (1974) y Parker (1982) fue introducido por Rumerlhart, Hinton y Williams (1986).

#### 4.6. Aprendizaje por Reforzamiento

Fue ideado por Widrow, Gupta y Maitra (1973) y desarrollado por Williams (1983). Este tipo de aprendizaje es similar al anterior en que

los pesos son fortalecidos en las acciones desarrolladas correctamente y penalizados en aquéllas mal realizadas. La diferencia entre estas dos técnicas de aprendizaje supervisado estriba en que el aprendizaje por corrección de error utiliza información de error más específica reuniendo los valores de error de cada PE de la capa de salida, mientras que el aprendizaje por reforzamiento utiliza información de error no específica para determinar el desarrollo de la red. Mientras que el primero tiene un vector completo de valores que utiliza para la corrección del error, sólo un valor es usado para describir la ejecución de la capa de salida durante el aprendizaje por reforzamiento. Esta forma de aprendizaje es ideal en situaciones donde no está disponible información específica sobre el error, pero sí información global de la ejecución, tal como predicción y control.

Redes neuronales que emplean aprendizaje por reforzamiento son AHC (*Adaptive Heuristic Critic*, Barto, Sutton y Anderson, 1983) y ARP (*Associative Reward-Penalty*, Barto, 1985).

#### 4.7. Aprendizaje Estocástico

Utiliza procesos aleatorios, probabilidades y una relación de energía para seleccionar los pesos a cambiar y su tamaño en una red neuronal multicapa. Aceptan los que mejoran una función objetivo y los cambios malos se rechazan o aceptan según una distribución probabilística.

La primera red que empleo aprendizaje estocástico fue la Máquina de Boltzmann (Ackley, Hinton y Sejnowski, 1985). Szu (1986) mejoró el procedimiento utilizando la función de distribución de Cauchy en lugar de la función de distribución Gaussiana, resultando una red que converge a una solución más rápidamente (Tabla 1).

TABLA 1.

Algoritmo de Entrenamiento	Tiempo de Entrenamiento	Supervisado / No Supervisado	Lineal / No Lineal	Estructural / Temporal	Capacidad Almacenam.
Aprendizaje Hebbiano	Rápido	No Supervisado	Lineal	Estructural	Pobre
Aprendizaje Competitivo	Lento	No supervisado	Lineal	Estructural	Buena
Aprendizaje Min-Max	Rápido	No Supervisado	No lineal	Estructural	Buena
Aprendizaje Corrección Error Dos Niveles	Lento	Supervisado	Lineal	Ambos	Buena
Aprendizaje Corrección Error Multinivel	Muy lento	Supervisado	No lineal	Ambos	Muy buena
Aprendizaje por Reforzamiento	Super lento	Supervisado	No lineal	Ambos	Buena
Aprendizaje Estocástico	Super lento	Supervisado	No lineal	Estructural	Buena

**TABLA 2.**  
**Algunas de las redes neuronales mejor conocidas.**

Nombre de la red	Inventores y Desarrolladores	Año	Aplicaciones Principales	Limitaciones	Comentarios
ART	Gail, Carpenter, Univ. Northeastern; Stephen Grossberg, Univ. Boston	1978-86	Reconocimiento patrones, especialmente cuando el patrón es complicado o poco familiar para los humanos (sonidos, lecturas de radar o sonar)	Sensible a distorsiones y cambios en la escala	Muy sofisticado, no aplicado todavía a demasiados problemas
Madaline	Bernard Widrow, Univ. Stanford	1960-62	Modems adaptativos; ecualizadores adaptativos (canceladores de eco) en líneas telefónicas	Supone una relación lineal entre entrada y salida	Cuenta con aplicaciones comerciales desde hace más de 20 años
Neocognitron	Kunihilo Fukushima, Laboratorios NHK	1978-84	Reconocimiento de caracteres manuscritos	Necesita de un número muy grande de elemento de proceso y conexión	Una de las redes más complicadas que se han desarrollado; insensible a las diferencias de escala, traslación, rotación
Perceptrón	Frank Rosenblatt, Univ. Cornell	1957	Reconocimiento de caracteres	No puede reconocer caracteres complejos (como los chinos)	Es la red neuronal más antigua; poco utilizada hoy en día
Retropropagación	Paul Werbos, Univ. Harvard; David Rumelhart, Univ. Stanford	1974-85	Síntesis fonética de textos; control adaptativo de brazos de robots	Sólo entrenamientos supervisado, los ejemplos de entrada-salida correctos deben ser abundantes	Es la red neuronal más popular hoy en día
Hopfield	John Hopfield, California, Laboratorio AT&T Bell	1982	Recuperación de datos o imágenes completos de fragmentos	No aprende, los pesos deben estar compuestos de fragmentos	Puede ser implementada en gran escala
Memoria, asociativa bidireccional	Bark Kosko, Univ. de Southern California	1985	Memoria asociativa direccionable por contenidos	Baja densidad de almacenamiento; los datos deben estar convenientemente codificados	La red que más fácilmente aprende

## 5. APLICACIONES

Como ya vimos anteriormente, por sus propias características de funcionamiento, las redes neuronales no son apropiadas para la resolución de problemas numéricos como los habitualmente programados en los ordenadores. Su investigación, no obstante ha expandido los horizontes de los ordenadores como herramientas no sólo capaces de calcular con extrema rapidez, sino también como elementos capaces de comportarse de forma adecuada en el mundo real, donde la información es enormemente amplia e impredecible.

En el amplio abanico de aplicaciones que hoy en día pueden tener las redes neuronales cabe destacar:

- Procesamiento del habla
  - Reconocimiento del habla
  - Pronunciación fonética
- Reconocimiento de caracteres y palabras

- Visión
- Reconocimiento de patrones
- Problemas de optimización
- Comunicación de ruta de tráfico
- Robótica
- Sistemas de control
- Aplicaciones médicas (diagnóstico, ultrasonidos, etc)
- Detección de explosivos camuflados en equipajes y mercancías
- Análisis financieros
- Procesamiento de la señal (radar, sonar, etc)
- ... (Tabla 2).

## REFERENCIAS

- CARPENTER, G.; GROSSBERG (1987): *A massively parallel architecture for a self-organizing neural pattern recognition machine*. Computer Vision, Graphics, and Image Understanding, vol. 37, pp. 54-115.
- FUKUSHIMA, K. (1980): *Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position*. Biological Cybernetics.
- HOPFIELD, J. J. (1982): *Neural Networks and Physical System with Emergent Collective Computational Abilities*. Proceedings of the National Academy of Sciences.
- KOHONEN, T. (1984): *Self-organization and Associative Memory*. Berlín: Springer-Verlag.
- KOHONEN, T. (Diciembre 1987): *Adaptive, Associative, and Self-Organizing Functions in Neural Computing*. Applied Optics.
- KOSKO, B. (Enero/Febrero 1988): *Bidirectional Associative Memories*. IEEE Transactions on Systems, Man, and Cybernetics.
- POGGIO, T.; GIROSI, F. (Septiembre 1990): *Networks for Approximation and Learning*. Proceedings of the IEEE.
- SEJNOWSKI, T. J.; ROSENBERG, C. R. (1987): *Parallel Networks that Learn to Pronounce English Test*. Complex Systems.
- WIDROW, B.; LEHR, M. A. (Septiembre 1990): *30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation*. Proceedings of the IEEE.