

Redes Neuronales Artificiales: más allá del Modelo de Regresión Lineal

Artificial Neural Networks: Beyond the Linear Regression Model

Manuel García-Ramos*

22

Palabras clave

Redes neuronales artificiales

Perceptrón multicapa

Algoritmo backpropagation

Modelación econométrica

Keywords

Artificial neural networks

Multilayer perceptrón

Backpropagation algorithm

Econometric modeling

Jel: C45, C63, C65

Resumen

En el presente trabajo se presentan el Perceptrón Multicapab y el Algoritmo Backpropagation, la Red Neuronal Artificial y la regla de aprendizaje más utilizadas en los últimos años para la modelación económica univariada y multivariada, debido a su gran capacidad para la resolución de problemas tanto lineales como no lineales. La orientación es teórica, de tal forma que resulte una guía básica, pudiendo ser entendida por quien cuente con una formación matemática y económica básica.

Abstract

This paper introduces the Multilayer Perceptron and the Backpropagation algorithm, the artificial neural network architecture and learning rule that have become predominant in recent years for univariate and multivariate economic modeling. Their appeal lies in their capacity to address both linear and nonlinear problems with remarkable flexibility. The focus is theoretical: the discussion is intended as an introductory guide accessible to readers with basic mathematical and economic training.

Introducción

Las Redes Neuronales Artificiales (RNA), inspiradas en la existencia de una organización celular horizontal en capas en el *córtex cerebral*,¹ son modelos matemáticos que mimetizan las habilidades básicas de las Redes Neuronales Biológicas para procesar grandes cantidades de información. Las RNA intentan simular dos de las características más importantes del cerebro: 1) capacidad de *aprendizaje* o generalización y 2) procesamiento de información *incompleta* y/o no lineal.

1 El *córtex* o corteza cerebral contiene cerca de cien mil millones de neuronas, las cuales son aproximadamente el número de estrellas en la Vía Láctea. Brunak, S. y B. Lautrup (1990)

Gracias a su flexibilidad, las RNA son aplicadas en diferentes campos de conocimiento, como complementos de los métodos existentes o como metodologías alternativas. Las RNA se han vuelto sumamente populares en economía y finanzas (Trippi y Turban, 1993), ya que estos modelos pueden utilizarse como aproximadores universales de funciones para el reconocimiento de patrones, la clasificación y el pronóstico.

Sin embargo, al igual que los métodos econométricos, las RNA son un conjunto de técnicas difícil de abordar en su totalidad; por esta razón, en este documento nos enfocaremos en la metodología básica más utilizada en economía: el *Perceptrón Multicapa* —Rumelhart *et al.*, 1986— y el *Algoritmo Backpropagation* —Levenberg (1944) y Marquardt (1963)—. El planteamiento teórico pertenece a diversos autores que se señalarán debidamente, la notación y el desarrollo matemático son propios.²

1. RNA para el modelado econométrico

Bajo los supuestos de que los agentes son racionales y que toman sus decisiones de forma óptima (es decir, los agentes conocen el modelo *real* de la economía), los analistas construyen modelos que muestren *regularidades* en los hechos económicos. La técnica más usada para ello es el Modelo de Regresión Lineal (MRL), basado en el método de Mínimos Cuadrados Ordinarios. Su popularidad radica en que su aplicación es sencilla y los resultados son fáciles de interpretar, en términos de *propensiones marginales* o de *elasticidades*; por estas razones, el MRL constituye la base de otras metodologías más avanzadas, como los modelos de Vectores Autoregresivos (VAR).

2 Para un desarrollo teórico y empírico más extenso consúltese García-Ramos, M (2009).

Sin embargo, estas técnicas se vuelven poco prácticas cuando el economista se enfrenta a problemas demasiado complejos para ser simplificados. Algunos estudios (Granger, 1991) muestran que un número considerable de relaciones económicas son no lineales o, incluso, caóticos. Por esta razón, en los últimos años se han desarrollado técnicas estadísticas y matemáticas que permiten la modelación y pronóstico de problemas complejos e imprecisos. Muchas de estos métodos alternativos tienen su origen en la Inteligencia Artificial, como los Algoritmos Genéticos, los Sistemas de Clasificación Inteligentes y las RNA.

El uso de RNA para el modelado y pronóstico econométrico permite relajar el supuesto de que los agentes conocen el modelo real de la economía. Además, gracias al desarrollo computacional de los últimos años, las RNA encarnan una atractiva herramienta para los investigadores, puesto que el costo de oportunidad de uso y aplicación es muy bajo, y, a diferencia de los modelos basados en técnicas econométricas tradicionales, las RNA son sistemas adaptativos; es decir, “aprenden” las formas funcionales a través de los datos incluso si las relaciones son desconocidas o son difíciles de describir. Gracias a su flexibilidad, las RNA pueden ser aplicadas en diferentes campos de conocimiento, como complementos de los métodos lineales existentes o como metodologías alternativas. Los problemas básicos que se abordan a través de ellas son los siguientes:

- *Clasificación de patrones*: el objetivo es asignarle a un patrón de entrada una clase pre-especificada. Las aplicaciones mejor conocidas son el reconocimiento del habla, la clasificación de cardiogramas y la clasificación de tipos sanguíneos.
- *Clustering*: también conocido como *clasificación no supervisada de patrones*, explora

la semejanza entre patrones en un cluster. Las aplicaciones mejor conocidas incluyen el *data mining*, la compresión de datos y el análisis exploratorio de datos.

- *Aproximación de funciones*: supóngase un conjunto de n patrones $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ generados a partir de una función desconocida $\mu(x)$ con ruido. El objetivo de la aproximación de funciones es encontrar un estimado, $\hat{\mu}$, de la función desconocida μ .
- *Pronóstico*: dado un conjunto de m muestras $\{y_1, y_2, \dots, y_m\}$ donde $t=1, \dots, m$; el objetivo es calcular y_{t+1} . El pronóstico tiene impacto significativo en las decisiones de negocios, ingeniería, etc.; la predicción económica y climática son aplicaciones típicas.
- *Optimización*: una gran variedad de problemas en matemáticas, estadística, ingeniería, medicina y economía son de optimización. El objetivo es encontrar una solución que satisfaga un conjunto de restricciones.

En el área de las ciencias económicas las RNA son una eficaz herramienta para detectar y reproducir relaciones lineales y no lineales entre un conjunto de variables (Zhang *et al.*, 1997). En general, cualquier tipo de regresión o modelo multivariante puede reescribirse como una RNA; algunos ejemplos son: la Regresión Lineal, la Regresión Polinómica, la Regresión Logística, el Análisis Factorial y el Análisis de Componentes Principales.

Algunas de las ventajas de la implementación de las RNA en el modelado y pronóstico de series económicas son: 1) no es necesario elaborar modelos *a priori*, 2) no es necesario especificar funciones de distribución de probabilidad, 3) generalización de procesos, 4) tolerancia a información incompleta o no lineal, 5) operación de datos en tiempo real.

2. RNA: un ingrediente extra

Una pregunta fundamental en el análisis económico contemporáneo es: ¿qué pasa si los agentes tienen que “aprender” la naturaleza de los shocks, los objetivos de política y retroalimentar las reglas de las instituciones a sus propias decisiones?; en otras palabras, ¿qué pasa si los agentes no conocen el modelo real de la economía y tienen que hacer “aproximaciones” para capturar y aprender la forma en que las variables clave de la economía se comportan?

Este tipo de interrogantes, de gran trascendencia teórica y empírica, no son triviales, pero tampoco son recientes. El objetivo de R. E. Lucas (1976) al publicar el artículo *Econometric Policy Evaluation: A Critique* era indicar que, bajo la hipótesis de expectativas racionales, los modelos econométricos no representaban adecuadamente la realidad económica debido a su incapacidad para 1) incorporar en sus modelos choques económicos aleatorios y 2) los ajustes de los agentes económicos ante cambios de política económica.³ Por lo tanto, la econometría (clásica) no podría usarse para formulación de políticas económicas.

La crítica argumentaba que los parámetros de los modelos tienen una tendencia a ser inestables en el tiempo, debido a que los cambios en las políticas económicas van a llevar a los agentes económicos a revisar estos parámetros; de tal manera que, si se tenía un conjunto de parámetros estimados y de pronto se produce un cambio, esos parámetros (considerados invariables) no serían más aplicables.

En una primera lectura, la *crítica de Lucas* parece invalidar todo esfuerzo por modelar la economía, sea a nivel macro o microeconómico; sin embargo, no es así. Los contraar-

³ Lo cual significa que entre los agentes existe 1) información asimétrica y 2) relaciones no lineales.

gumentos asociados al trabajo de Lucas han mejorado las técnicas econométricas y matemáticas de los últimos años. Por ejemplo, Sims (1980) desarrolló la metodología VAR para demostrar que la *crítica* sólo opera en casos excepcionales; por otro lado, algunos teóricos (en su mayoría matemáticos y estadistas), bajo el hecho de que la realidad económica es compleja y caótica y no simple y ordenada, han desarrollado diversas técnicas para plantear y evaluar modelos con parámetros no lineales o que varían en el tiempo, como los Modelos Dinámicos Lineales y las RNA.

Sin embargo, el uso de este tipo de modelos, clasificados como sistemas de *caja negra*,⁴ también ha recibido numerosas críticas por parte de los investigadores más ortodoxos, debido a que muchos de los parámetros estimados no pueden ser justificados o interpretados económicamente. Y es aquí donde radica su desaprobación: en el deseo de que toda estimación empírica tenga una lectura económica directa o esté respaldada por la teoría económica. Pero muchas veces lo que dictan las teorías es imposible de verificar en la práctica, ya que hay una brecha enorme entre lo que *es* y lo que *debería*; el mismo Lucas indica: “la construcción de modelos econométricos está sujeta a una gran cantidad de críticas mal informadas y casuales. Así, los modelos están condenados por ser *demasiado grandes* (con igual introspección, supongo que uno podría culpar a modelos con menor número de ecuaciones por ser *demasiado pequeños*), demasiado complicados, demasiado simplistas (es decir, no demasiado complicados), y, el golpe final, inferiores a los modelos ingenuos”.

⁴ El término *caja negra* se refiere a un sistema del que se desconocen (o no son de interés de análisis) los elementos o mecanismos internos de solución, pero se sabe que a determinadas entradas (inputs) corresponden salidas bien delimitadas (outputs).

Al parecer, la regla más sensata es: “anteponer la nitidez de los datos a la divinidad econométrica” (Kennedy, 2002). De esta forma, si bien la compleja forma funcional de un sistema *caja negra* como las RNA hace difícil (pero no imposible) interpretar económicamente los parámetros estimados, su potencia no consiste necesariamente en la *elegancia* de una solución simple, sino en la *generalización* de una respuesta concreta a partir de información de la que se desconoce un modelo *a priori*; además, en vez de considerar que los agentes actúan de forma racional y conocen a la perfección los mecanismos de la economía, el modelado a través de RNA supone que esto no se cumple; es más, se permite que los agentes aprendan a través del tiempo y reajusten sus errores (expectativas racionales) de forma dinámica.

Por otro lado, también es cierto que el uso de RNA y otras técnicas de *caja negra* transgrede la regla “modelos simples deben ser preferidos a modelos complicados” (basada en la *navaja de Ockham* o *ley de la Parsimonia*).⁵ Sin embargo, Menger (1960), fundador de la Escuela Austríaca de Economía, al saber que los economistas son demasiado parsimoniosos (y renuentes) en lo que respecta al uso de nuevos modelos (y técnicas) matemáticos y/o estadísticos, formuló una *anti-navaja de Ockham*.⁶ “No tiene caso tratar de hacer con menos lo que requiere más”. El objetivo del economista no es simplificar hasta sus últimas consecuencias la realidad, sino proporcionar información

de mayor calidad, así como reducir la incertidumbre de los tomadores de decisiones; además, debido a la creciente complejidad de los problemas a los que se enfrentan los investigadores, es preciso enriquecer los métodos y técnicas de análisis para el procesamiento de información y, de esta forma, obtener soluciones más rápidas y eficaces. Un modelo econométrico, estadístico o matemático debe ser simple, pero no más simple.

Aprovechar las sinergias entre las técnicas convencionales y las nuevas metodologías para la modelación y pronóstico permitirá, entre otras cosas, evitar los Errores Tipo III,⁷ ya que es mejor una respuesta *aproximada* a la pregunta “correcta”, la cual es a menudo imprecisa, que una respuesta *exacta* a la pregunta “incorrecta”, la cual con mucha frecuencia puede formularse con exactitud (Tukey, 1962). Las RNA son un ingrediente extra en el análisis de datos, en la búsqueda de evidencia adicional.

3. Breve historia de las RNA

El cerebro humano está formado por millones de neuronas conectadas entre sí que transmiten y procesan la información percibida a través de los sentidos (vista, tacto, oído, gusto, olfato), para dar una respuesta adecuada a cada uno de los diferentes estímulos. Cada neurona se conforma fundamentalmente de cuatro partes básicas (gráfica 1):

1. *Dendritas*: ramas fibrosas que actúan como conexiones de entrada de la *soma*. Las dendritas conducen los impulsos nerviosos hacia el cuerpo de la célula.

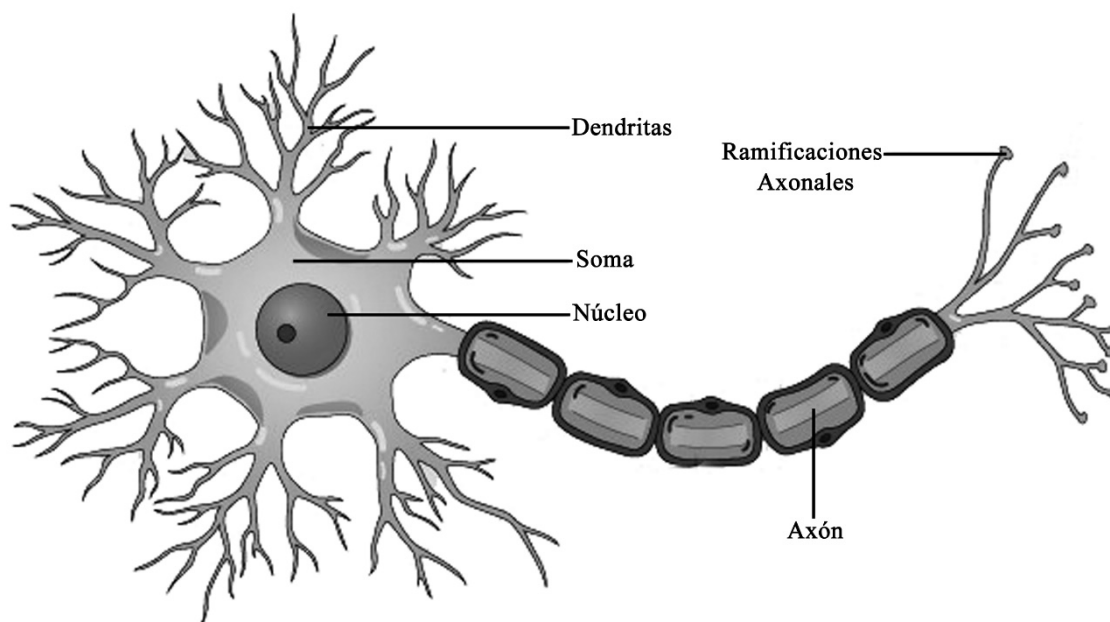
⁵ *Frustra fit per plura quod potest fieri per pauciora* o “No tiene caso hacer con más lo que se puede hacer con menos”. Frase atribuida a Guillermo de Ockham, fraile inglés del siglo XIV. Para una discusión más amplia, véase Rodríguez-Fernández (1999).

⁶ Existen otras *anti-navajas de Ockham*; por ejemplo: el *Principio de Plenitud* de Gottfried Wilhelm von Leibniz, la *Crítica de la Razón Pura* de Immanuel Kant, la *Regla Zellner* de Arnold Zellner.

⁷ Un Error Tipo III ocurre cuando el investigador presenta la respuesta correcta a la pregunta incorrecta. Véase Kimball (1957).

2. *Soma*: cuerpo celular o estructura de entrada.
3. *Axón*: canal transmisor de los impulsos generados por la soma. En el extremo, la fibra principal se ramifica para conectarse con dendritas de otras neuronas o directamente a los músculos del cuerpo.
4. *Núcleo*: es el centro de síntesis de información de la célula; en él se procesan las señales que llegan de otras neuronas, en forma de impulsos, para generar un nuevo impulso si se cumplen ciertas condiciones.

Gráfica 1 Esquema básico de una neurona biológica



La conexión entre neuronas se realiza por medio de uniones especiales llamadas *sinapsis*, las cuales determinan la fuerza y tipo de relación entre ellas. A la intensidad de una sinapsis se le llama *peso sináptico* y se determina biológicamente por el proceso de *aprendizaje*. El aprendizaje de las Redes Neuronales Biológicas (RNB) se produce mediante la variación de las sinapsis; es decir, en el cambio de la influencia o *peso* que unas neuronas ejercen sobre otras; de esta forma, con el paso del tiempo, el sistema neuronal se modifica para establecer nue-

vas conexiones, romper otras, alterar las intensidades sinápticas, etc. Estas transformaciones en la intensidad de las conexiones constituyen la idea de trasfondo de las RNA.

Históricamente, el trabajo del neurofisiólogo Warren McCulloch y del matemático Walter Pitts es el primero en proponer un modelo matemático que imita el comportamiento básico simplificado de las RNB (McCulloch y Pitts, 1943). Los años, autores y aportaciones más importantes en la historia de las RNA puede consultarse en el siguiente cuadro:

Cuadro 1.1 Breve historia de las RNA

Año	Autor	Aportación
1943	McCulloch y Pitts	Desarrollan el primer modelo de Neurona Artificial.
1949	Hebb	Desarrolla la regla de Hebb, antecesora de las técnicas modernas de entrenamiento de las RNA.
1958	Rosenblatt	A partir de un trabajo de Minsky (1954), desarrolla el Perceptrón, un modelo simple que aprende a conectar y asociar datos de entrada dados con una unidad de salida aleatoria.
1960	Widrow y Hoff	Crean una variación de la regla de Hebb. La denominada ley de Widrow-Hoff da lugar a las primeras RNA aplicadas a problemas reales: ADALINE (ADaptive LINear Elements) y MADALINE (Multiple ADALINE).
1969	Minsky y Papert	Demuestran las limitaciones teóricas en el aprendizaje del Perceptrón, incapaz de aprender funciones no linealmente separables.
1974	Werbos	Desarrolla el método de aprendizaje conocido como Backpropagation, el algoritmo más utilizado en las aplicaciones de RNA.
1977	Anderson et al.	Desarrolla el Asociador Lineal y su extensión Brain-State-in-a-Box, que modela funciones complejas.
1980	Fukushima	Desarrolla diversas redes autoasociativas: la red Neocognitron se especializa en la detección de clusters.
1982	Hopfield	Desarrolla una variante del Asociador Lineal conocido como red de Hopfield, la cual se utiliza en la reconstrucción de patrones y optimización.
1984	Hinton et al.	Desarrollan la Máquina de Boltzman, que reconoce patrones y optimiza funciones simples.
1985	Instituto Americano de Física (IEEE)	Primera reunión anual del Neural Networks for Computing.
1986	Carpenter y Grossberg	Desarrollan el modelo ART (Adaptative Resonance Theory), el cual da respuesta a dos interrogantes fundamentales: cómo puede aprender una RNA nuevos patrones y cómo puede retener los patrones previamente aprendidos.
	Rumelhart et al.	Perfeccionan el algoritmo Backpropagation, ofreciendo una poderosa solución para la construcción de RNA más complejas.

Cuadro 1.1, continuación

1987	Grossberg et al.	Creación de la primera conferencia internacional sobre RNA: International Neural Network Society (INNS).
	Varios	Creación de la reunión anual del Neural Information Processing Systems (NIPS).
1988	IEEE e INNS	Creación del International Joint Conference on Neural Networks (IJCNN).
1990	Kohonen	Formaliza el algoritmo de aprendizaje denominado Self-Organizing Map (SOM), base de las RNA autoorganizadas de tipo competitivo.
1991	Sociedad Europea de Redes Neuronales	Creación del International Conference on Artificial Neural Networks (ICANN).
1997	Kohonen	Formaliza el algoritmo de aprendizaje denominado Self-Organizing Feature Map (SOFM), el cual permite la formación de mapas topológicos multidimensionales.

4. La Neurona Artificial

Una Neurona Artificial es la unidad mínima de procesamiento de información en la operación de una RNA. El modelo, propuesto por Mc Culloch y Pitts y desarrollado por Rosenblatt (1958), se conoce como *Perceptrón Simple* (PS); sin embargo, lo más frecuente es darle el nombre de nodo o elemento de procesamiento (Freeman y Skapura, 1993).⁸

Gráficamente, un nodo puede representarse por la gráfica 1.2 y sus elementos constitutivos son: 1) *señales de entrada* o *inputs*, X_1, \dots, X_n ; 2) magnitudes asociadas a las señales de entrada llamadas *pesos sinápticos* o *intensidades de conexión*, w_1, \dots, w_n ; 3) un término de *sesgo* o *umbral*, α , el cual tiene el efecto de incrementar o disminuir la entrada neta dependiendo si éste es positivo o negativo; 4) una *función de propagación* o *entrada neta*, NET , que es la suma ponderada de las señales de entrada con sus respectivos pesos sinápticos más un umbral; 5) una *función de activación* o *transferencia*, γ , que combina las señales de entrada con el estado de activación inicial para producir un nuevo valor de activación; 6) una *función de salida*, $\varphi(\bullet)$, que transforma el estado final de activación de la neurona en las señales de salida, las cuales son enviadas a los nodos conectados a ella; 7) una *señal de salida* o *output*, Y ; 8) una *regla* o *algoritmo* de aprendizaje.

Una unidad de procesamiento funciona como un sistema input-output, ya que usa los valores de los nodos de entrada para calcular el valor de la neurona de salida. Si definimos las señales de entrada, X_1, \dots, X_n , y la señal de salida, Y , como vectores m -dimensionales; es decir, $X_n = [x_{n1} \dots x_{nm}]$ y $Y = [y_1 \dots y_m]$,⁹ podemos especificar matricialmente un PS a través de las siguientes expresiones:

$$(1.1) \quad Net = \mathbf{W}\mathbf{X} + \alpha$$

$$(1.2) \quad Act_t = \gamma(Act_{t-1}, Net_t)$$

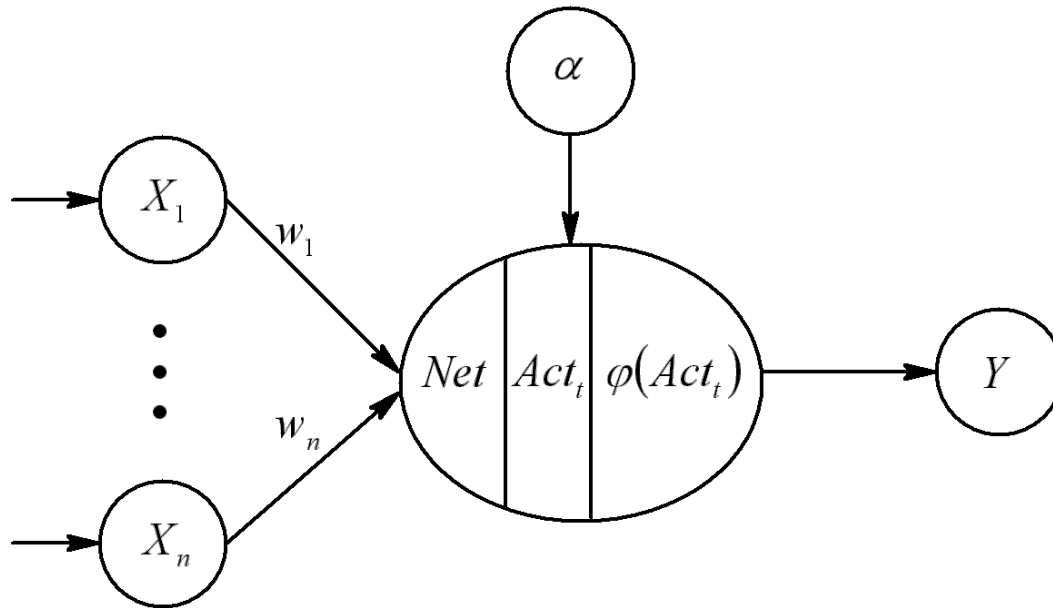
$$(1.3) \quad \mathbf{Y} = \varphi(Act_t)$$

Donde: \mathbf{X} y \mathbf{W} son matrices $n \times 1$ y $1 \times n$, respectivamente; $\mathbf{Y} = [Y]$.

⁸ Términos que se emplearán de manera intercambiable a lo largo del documento.

⁹ Donde m es el número de elementos u observaciones. Es importante mencionar que no es necesario que la señal de salida contenga los mismos elementos que las señales de entrada.

Gráfica 2 Esquema básico de un Perceptrón Simple



La ecuación (1.1) representa la forma en que se ponderarán las señales de entrada y los pesos sinápticos; en este caso, se trata de una combinación lineal entre los inputs y sus respectivas intensidades de conexión más un umbral. En esta expresión si w_n es positivo la señal de entrada tendrá un efecto *excitador*, es decir, la neurona se activará; por el contrario, si w_n es negativo la sinapsis será *inhibidora*, es decir, la neurona se desactivará; por último, si w_n es igual a cero, significa que no existe conexión entre los nodos. Además, es importante advertir que el efecto de un cambio en el nodo sobre será siempre, independientemente de que sea positivo o negativo; en otras palabras, los efectos son simétricos. Ahora bien, el uso de α tiene el propósito de buscar una transformación afín para WX porque dependiendo si éste es positivo o negativo, la relación Net es modificada (gráfica 1.3).¹⁰ Nótese que si $WX=0$, entonces $Net=a$.

¹⁰ El parámetro a puede pensarse como un peso w_0 asociado a una señal de entrada $X_0=1$. Haykin, 1999.

Por otro lado, la ecuación (1.2) combina el potencial post-sináptico, con el estado inicial de la neurona, Act_{t-1} , para producir un nuevo estado de activación acorde con la información recibida;¹¹ sin embargo, en la mayoría de las arquitecturas de RNA aplicadas se considera que el estado actual de la neurona no depende de su estado previo. De esta forma, podemos expresar la función de transferencia como:

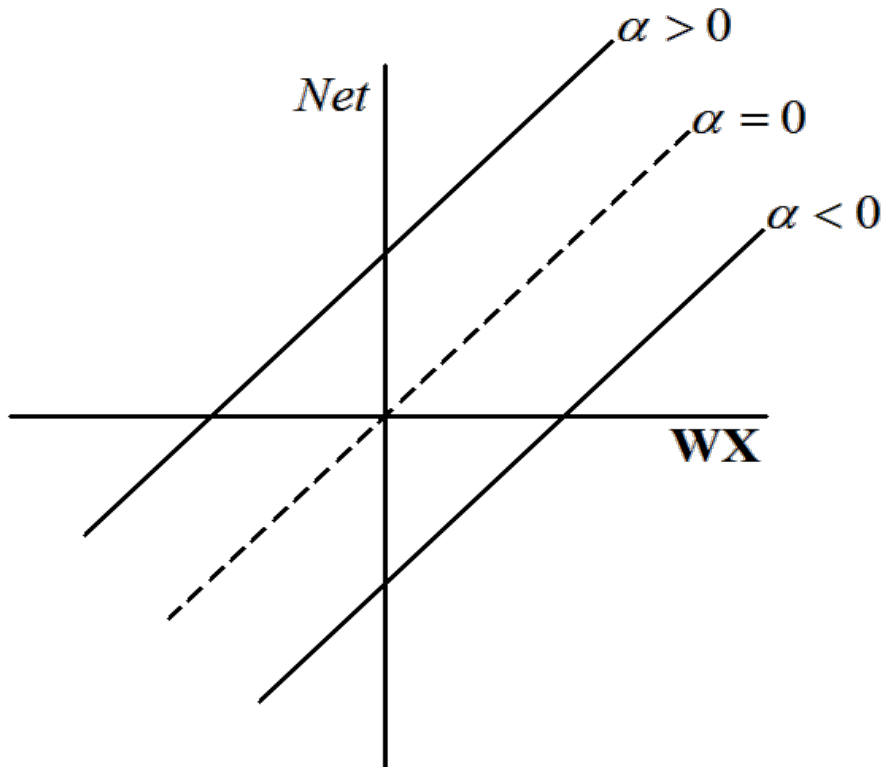
$$(1.4) \quad Act_t = \gamma(Net_t)$$

Además, si consideramos que $\gamma(\bullet)$ tiene carácter determinista, es decir, que se trata de una función identidad, entonces podemos simplificar la expresión anterior de la siguiente forma:

$$(1.5) \quad Act = Net$$

¹¹ Toda neurona presenta cierto valor o *estado* inicial, de reposo o excitación que puede ser continuo, discreto, limitado o ilimitado.

Gráfica 3 Transformaciones afines causadas por α



La ecuación (1.3) es la señal de salida; es decir, la transformación de (1.2) o (1.5) a través de la función $\varphi(\bullet)$, que permite capturar las posibles relaciones no lineales entre *Net* y el output.

Finalmente, la regla de aprendizaje es la manera en que se calcula y actualiza la matriz \mathbf{W} , con el objetivo de minimizar el error entre la señal de salida calculada y la señal de salida observada.

Al simplificar el sistema de ecuaciones (1.1) a (1.3) podemos reescribirlo como:

$$(1.6) \text{ NET} = \mathbf{VW} + \alpha$$

$$(1.7) Y = \varphi(\text{Net})$$

Resulta interesante notar que la ecuación (1.6) es idéntica al *modelo de regresión lineal*: los nodos de entrada y salida son equivalentes a las variables independientes y dependiente, respectivamente; mientras que los pesos sinápticos son semejantes a los coeficientes estimados y el sesgo, al intercepto. Si el sistema (1.6) a (1.7) tuviera más de un nodo de salida, $\mathbf{Y}^T = [Y_{1,t} \dots Y_{k,t}]$, y el conjunto de inputs fueran los p rezagos de las señales de salida, $\mathbf{X}^T = [Y_{1,t-p} \dots Y_{k,t-p}]$, obtendríamos una RNA similar a un modelo de *Vectores Autorregresivos* (González, 2000). Más aún: al introducir una conexión entre los outputs Y_1, \dots, Y_k definiríamos un *sistema de ecuaciones simultáneas*.

5. La función de activación-salida

La función de activación-salida transforma el estado final de activación de la neurona, Net , en las señales de salida. Por lo general, $\varphi(\bullet)$ es de tipo *sigmoidea* como la logística o la tangente hiperbólica —ecuación (1.8) y (1.9), respectivamente—. Este tipo de funciones, no lineales, crecientes y restringidas en un intervalo monótonico con límites superiores e inferiores, se caracterizan por presentar una derivada definida en todos sus puntos.

$$(1.8) \quad \varphi(Net) = \frac{1}{1 + e^{-Net}}$$

$$(1.9) \quad \varphi(Net) = \frac{e^{Net} - e^{-Net}}{e^{Net} + e^{-Net}}$$

La función logística tiene un rango de 0 a 1 (gráfica 1.4); en consecuencia, cuando $Net = 0$ la salida del nodo es 0.5, es decir, la neurona tiene cierta actividad aún en ausencia de estimulación; es importante notar que al aumentar Net la unidad aumenta su activación y al disminuir, se inhibe. Por otro lado, la función tangente hiperbólica tiene un rango de -1 a 1 (gráfica 1.5); de esta forma, si $Net = 0$ la salida es 0, lo cual permite resolver problemas continuos acotados o no. El objetivo esencial para utilizar estas funciones es capturar las posibles relaciones no lineales entre Net y el output; además, su empleo es apropiado en la modelación económica, debido a que las formas funcionales *a priori* entre variables son, frecuentemente, desconocidas.¹²

12 Una forma funcional incorrecta lleva a una deficiente estimación de los parámetros.

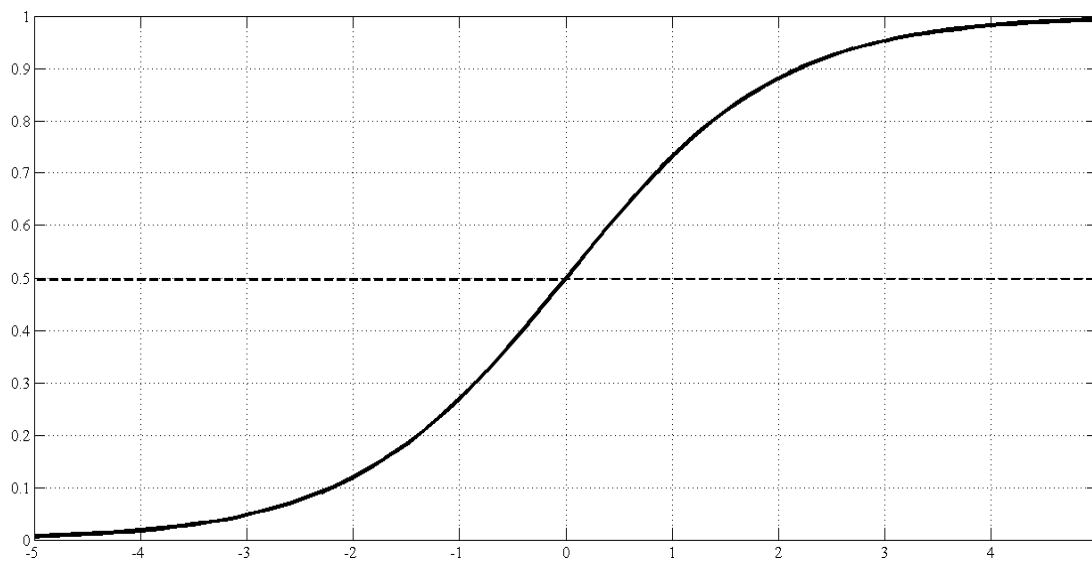
6. La regla de aprendizaje

El *conocimiento* de una neurona se encuentra representado en los pesos sinápticos entre los nodos; así, el proceso de aprendizaje o *entrenamiento* implica cierto número de cambios en la intensidad de las conexiones; es decir, el Perceptrón *aprende* al modificar de forma óptima los valores de la matriz \mathbf{W} .

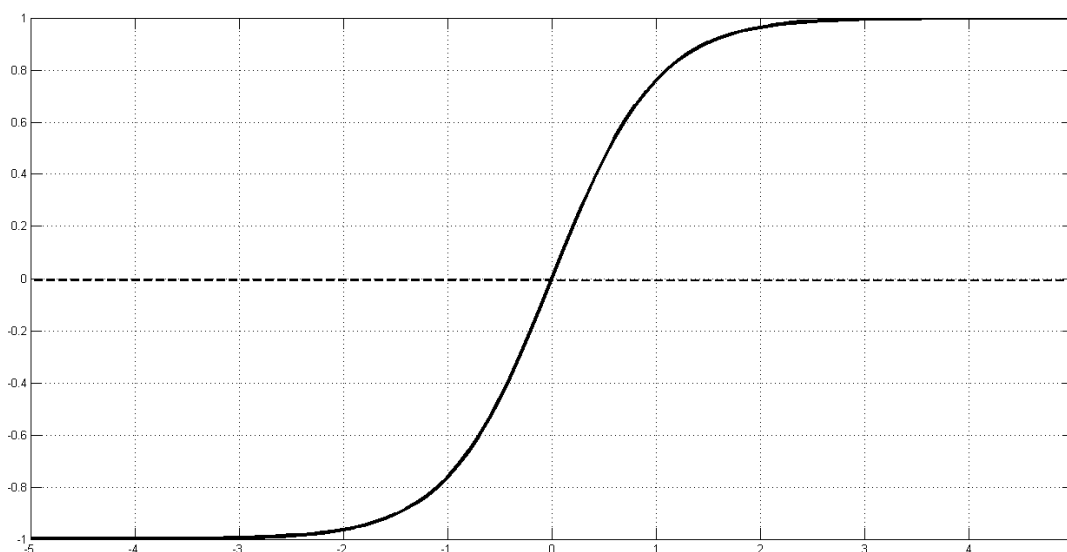
Básicamente, existen dos clases de algoritmos para la actualización de los pesos: 1) supervisados y 2) no supervisados.¹³ Dentro del conjunto de reglas de aprendizaje supervisado pueden distinguirse tres tipos: 1) por corrección de error, 2) por refuerzo y 3) estocásticas. En la práctica, los *algoritmos por corrección de error* son los más utilizados. Este tipo de reglas se caracterizan por establecer un *agente* que compara la señal de salida calculada con la señal de salida deseada y, si éstas son diferentes, ajusta la intensidad de las conexiones a través de minimizar el error cometido hasta que la señal de salida sea igual a la deseada; en otras palabras, el aprendizaje por corrección de error confronta el output estimado, $\hat{\mathbf{Y}}$, con el output observado, \mathbf{Y} , y ajusta \mathbf{W} iterativamente hasta que $\hat{\mathbf{Y}} = \mathbf{Y}$.

13 Su implementación dependerá de la *arquitectura* neuronal utilizada. Para mayor detalle, consúltese Flórez y Fernández (2008).

Gráfica 4 Función de activación-transferencia logística



Gráfica 5 Función de activación-transferencia tangente hiperbólica



A continuación, se presenta el algoritmo básico de aprendizaje del PS:

1. Inicio. Se establecen los pesos sinápticos de \mathbf{W} con números aleatorios.¹⁴
2. Iteración i -ésima. Se calcula:

$$\begin{aligned} Net_i &= \mathbf{W}_i \mathbf{X} + \mathbf{a}_i \\ \hat{Y}_i &= \varphi(Net_i) \end{aligned}$$

$i = 1, \dots$

Donde: i indica la iteración tal que

3. Se calcula el error de salida:

$$E_i = \mathbf{Y} - \hat{Y}_i \quad (1.10)$$

4. Donde: \hat{Y} es la señal de salida calculada.
5. Corrección de los pesos sinápticos. Si $\hat{Y}_i \neq Y$, entonces \mathbf{W} y \mathbf{a} se modifican a través de las siguientes expresiones:

$$\left. \begin{aligned} \mathbf{W}_{i+1} &= \mathbf{W}_i + \eta E_i \mathbf{X}^T \\ \alpha_{i+1} &= \alpha_i + \eta E_i \end{aligned} \right\} \quad (1.11)$$

6. Donde: η es una constante positiva (conocida como *tasa de aprendizaje*) tal que $0 < \eta < 1$; si ésta es pequeña, la neurona aprenderá con lentitud y, si es grande, lo hará con mayor rapidez (comúnmente, $\eta = 0.5$); T indica transposición.
7. Parada. Si no se han modificado las intensidades de las conexiones en las últimas r

¹⁴ Los cuales pueden estar en los intervalos (0,1) o (-1,1), dependiendo si se utiliza la función (1.8) o la (1.9).

iteraciones —ecuación (1.12)— el algoritmo se detiene, ya que el PS se ha estabilizado. En otro caso, se va al paso dos con $i = i + 1$.

$$\left. \begin{aligned} \mathbf{W}_s &= \mathbf{W}_i \\ \alpha_s &= \alpha_i \end{aligned} \right\} \quad (1.12)$$

Donde: $s = i + 1, \dots, i + r$.

Aunque eficiente, el algoritmo anterior presenta una limitación importante: si las señales de entrada no son linealmente separables,¹⁵ el aprendizaje del PS no converge después de un número finito de iteraciones (Minsky y Papert, 1969). Por esta razón, Werbos (1974) y, a la postre, Rumelhart *et al.*, Levenberg y Marquardt plantean, desarrollan y perfeccionan —respectivamente— el algoritmo *Backpropagation*, una regla de aprendizaje más poderosa y eficiente.

7. El Perceptrón Multicapa

La palabra *arquitectura* está reservada para la descripción formal de los elementos topológicos de una RNA: 1) número de capas, 2) número de nodos por capa, 3) tipo de conexión entre nodos y 4) algoritmo de aprendizaje. Las arquitecturas neuronales pueden clasificarse según la estructura de las capas, el flujo de datos, el tipo de respuesta de la red, etc.; sin embargo, debido a la enorme riqueza

¹⁵ Por ejemplo: se dice que dos conjuntos de puntos A y B son linealmente separables en un espacio n -dimensional si existen números reales, $w_1, \dots, w_n, \mathbf{a}$, tal que cada punto satisface y que cada punto no satisface. Véase gráfica 1.7.

za de arquitecturas y de reglas de aprendizaje existentes, en esta sección se desarrollarán únicamente las características fundamentales del *Perceptrón Multicapa* (PM) y del algoritmo *Backpropagation* (BP).

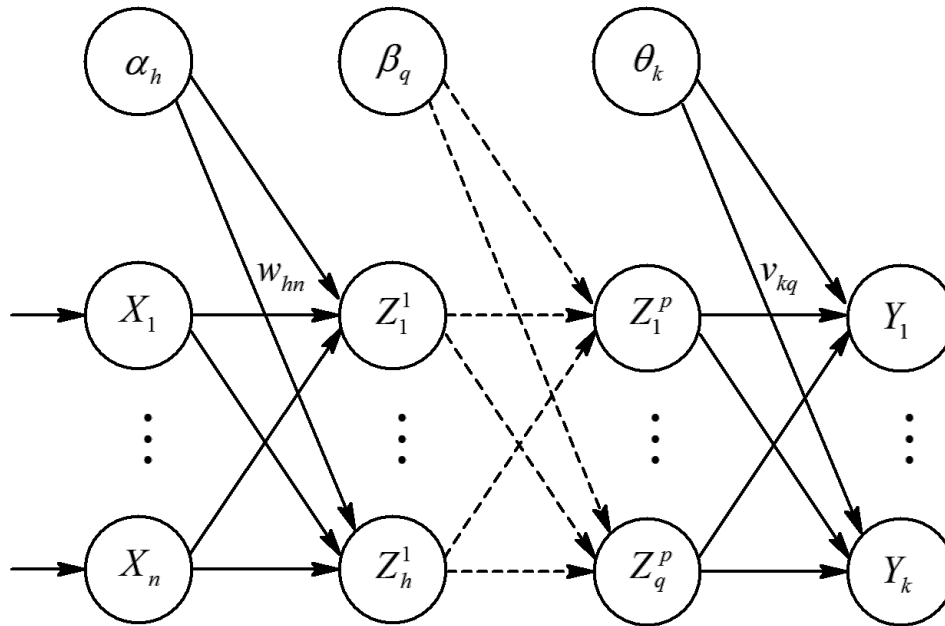
En esencia, el PM es una extensión y generalización del PS. Se trata de una arquitectura formada por diversas capas de neuronas interconectadas *hacia adelante* (figura 1.6),¹⁶ las cuales pueden clasificarse en tres tipos:

1. Capa de entrada: es la que recibe y procesa directamente los inputs, X_1, \dots, X_n .
2. Capas ocultas: son internas a la RNA y su objetivo es capturar todas las relaciones no directas entre las señales de entrada y de salida, Z_h^1, \dots, Z_q^p .
3. Capa de salida: es la que transfiere la información procesada al exterior de la red, Y_1, \dots, Y_k .

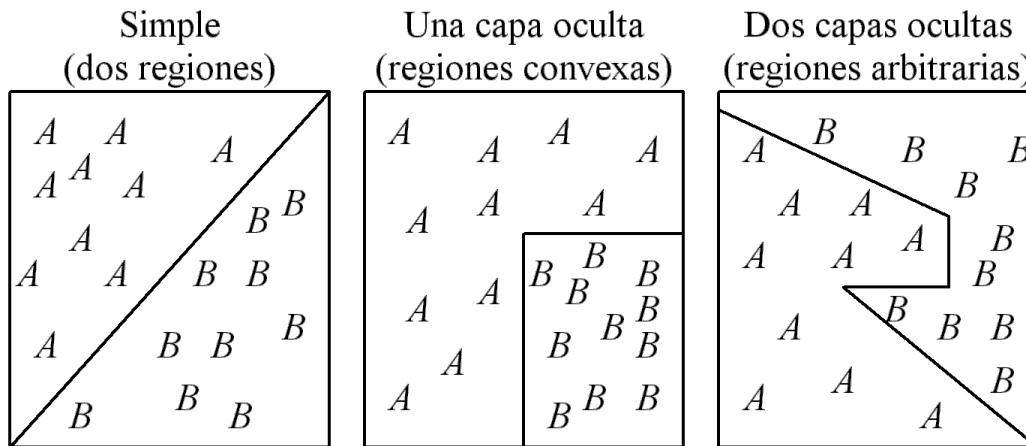
Si bien es posible construir arquitecturas con un número indeterminado de capas ocultas, Funahashi (1989) demostró que un PM con una capa oculta es un aproximador universal de funciones y puede representar adecuadamente regiones polinomiales convexas, tanto cerradas como abiertas; por otro lado, Lapedes y Faber (1987) comprobaron que un PM con dos capas ocultas puede representar regiones arbitrariamente complejas (gráfica 1.7). En general, una arquitectura con al menos una capa oculta y con suficientes nodos puede aprender y reproducir cualquier tipo de función o relación continua entre un grupo de señales de entrada y de salida.

¹⁶ Es decir, una capa recibe señales de entrada desde una capa anterior (la cual está más cerca de la capa de entrada) y envía señales de salida a una capa posterior (la cual está más próxima a la capa de salida). Este tipo de RNA se conocen como arquitecturas *feedforward*.

Gráfica 6 Esquema básico de un Perceptrón Multicapa



Gráfica 7 Tipo de Perceptrón y de región clasificable ¹⁷



¹⁷ Cuando la función de activación-salida es no lineal —como la ecuación (1.8) o (1.9)—, entonces las regiones no están limitadas por rectas, sino por curvas.

Si definimos las señales de entrada, x_1, \dots, x_n , y de salida, Y_1, \dots, Y_k , como vectores m -dimensionales; es decir, $X_n = [x_{n1} \dots x_{nm}]$ y $Y_k = [y_{k1} \dots y_{km}]$,¹⁸ podemos especificar matricialmente un PM $[n-h-k]$ ¹⁹ a través de las siguientes expresiones:

$$Net = \mathbf{W}\mathbf{X} + \boldsymbol{\alpha} \quad (1.13)$$

$$\mathbf{Z} = \varphi(Net) \quad (1.14)$$

$$\mathbf{Y} = \delta(\mathbf{V}\mathbf{Z} + \boldsymbol{\beta}) \quad (1.15)$$

Donde: \mathbf{X} , \mathbf{Z} y \mathbf{Y} son matrices de señales $n \times 1$, $h \times 1$ y $k \times 1$, respectivamente; \mathbf{W} es una matriz $h \times n$ con los pesos sinápticos de la capa de entrada; \mathbf{V} es una matriz $k \times h$ con los pesos sinápticos de la capa oculta; $\boldsymbol{\alpha}$ es una matriz $h \times 1$ con los umbrales de la capa de entrada y $\boldsymbol{\beta}$ es una matriz $k \times 1$ con los umbrales de la capa oculta.

Al igual que la ecuación (1.6), la expresión (1.13) establece una combinación lineal entre los inputs y sus respectivas intensidades de conexión más un umbral de activación.

La ecuación (1.14) transforma el potencial post-sináptico, Net , a través de la función $\varphi(\bullet)$ y envía la nueva señal a la capa oculta, \mathbf{Z} , con el objetivo de capturar patrones complejos entre los inputs y el output. Como señala González, es importante aclarar que los h nodos de la capa oculta no representan ningún concepto real; es decir, no tienen interpretación o significado económico directo.²⁰ Sin embargo, esto

no representa una limitación; por el contrario, como se indicó en párrafos anteriores, una arquitectura con una capa oculta y suficientes neuronas en ella puede aproximar cualquier función lineal o no lineal, lo cual sugiere que una RNA correctamente especificada puede utilizarse como una poderosa herramienta en la identificación y reproducción de Procesos Generadores de Información no conocidos *a priori* (ver gráfico 8).

Por otro lado, aunque las funciones $\varphi(\bullet)$ y $\delta(\bullet)$ pueden ser distintas para cada una de las capas de la red —inclusive para cada una de las neuronas—, por lo general, en la modelación de variables económicas $\varphi(\bullet)$ puede ser cualquiera de las ecuaciones (1.8) o (1.9) mientras que $\delta(\bullet)$ se determina como lineal. La ecuación (1.15), al igual que la ecuación (1.7), es la señal de salida calculada. Puede observarse que si el conjunto de inputs del sistema (1.13) a (1.15) fueran los p rezagos de las señales de salida, $\mathbf{x}^t = [Y_{1,t-p} \dots Y_{k,t-p}]$, obtendríamos una RNA similar a un modelo de *Vectores Autorregresivos*. Como dato adicional para futuras investigaciones, resulta interesante conocer y comparar el costo de formulación (en términos de parámetros) de estas metodologías: $q = (2p + 1) \times 1$ para un VAR irrestricto y $g = [(n+1) \times h] + [(h+1) \times k]$ para una RNA Multicapa; donde q y g son los grados de libertad de un VAR irrestricto y una RNA Multicapa, respectivamente; p es el número de variables; l es el número de rezagos; n , h y k son el número de nodos en la capa de entrada, oculta y de salida, respectivamente.

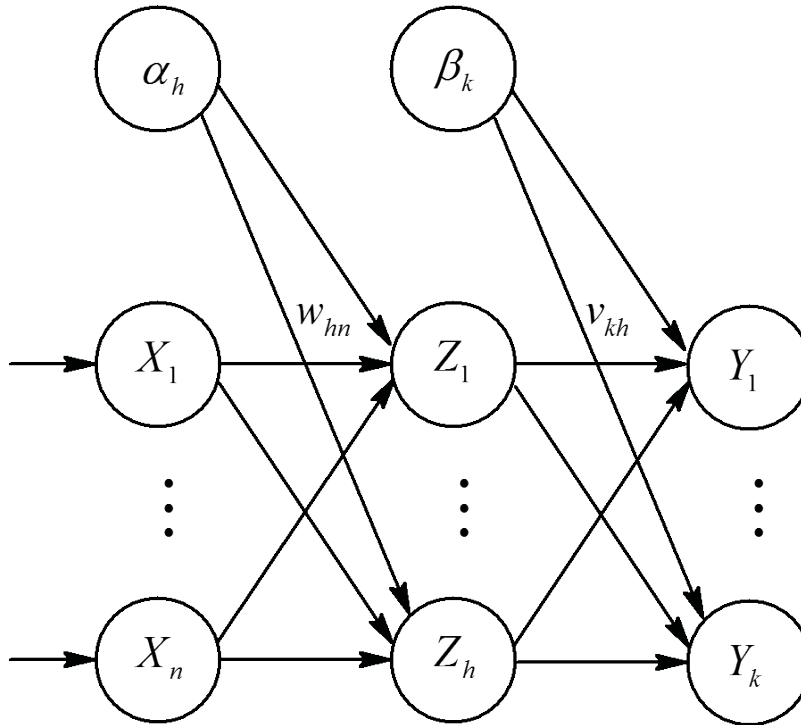
Finalmente, para calcular y actualizar las matrices de pesos sinápticos, \mathbf{W} y \mathbf{V} , se utiliza el algoritmo BP, con el objetivo de minimizar de forma óptima el error de estimación.

oculta como variables *latentes*; es decir, expectativas subjetivas o fuerzas conductoras del mercado: los “animal spirits” de Keynes.

¹⁸ Donde m es el número de elementos u observaciones.

¹⁹ Donde n , h y k hacen referencia al número de nodos en la capa de entrada, oculta y de salida, respectivamente. Véase gráfica 1.8.

²⁰ McNelis (2005) interpreta las neuronas de la capa

Gráfica 8**Esquema básico de un PM [n - h - k]**

38

8. El algoritmo Backpropagation

Se dice que una red aprende cuando ésta modifica sus pesos sinápticos una y otra vez hasta encontrar un conjunto que minimice el error de salida cometido. En el caso del PM, el aprendizaje se lleva a cabo a través del algoritmo *Backpropagation* (BP), el cual fue propuesto por Werbos y, posteriormente, desarrollado por Rumelhart *et al.*

El objetivo del algoritmo BP es estimar las matrices \mathbf{W} y \mathbf{V} minimizando una función de costo —ecuación (1.21)— a través de su gradiente; así, una vez que las señales de entrada se aplican a la primera capa de la red y éstas se transforman y propagan a través de la capa

oculta hasta generar un conjunto de outputs estimados,²¹ se calcula una señal de error para cada unidad de salida —ecuación (1.16)—.

En seguida, las señales de error se transmiten hacia atrás, partiendo de la capa de salida, hacia todos los nodos de la capa oculta —ecuaciones (1.17) a (1.18)—.²² Es importante notar que las unidades ocultas sólo reciben una fracción de la señal total del error, basándose aproximadamente en la contribución relativa

²¹ Proceso conocido como *Propagación hacia delante*.

²² Proceso conocido como *Propagación hacia atrás*.

que haya aportado la unidad a la salida original. Este proceso se repite hasta que todos los nodos de la red hayan recibido una señal de error que describa su contribución relativa del error total; de esta forma, con base en la señal de error percibida, se actualizan los pesos sinápticos de cada unidad para hacer que la red converja asintóticamente hacia una solución estable —ecuaciones (1.19) a (1.20)—.

A continuación, se presenta el algoritmo básico BP:

1. Inicio. Se establecen los pesos sinápticos de \mathbf{W} y \mathbf{V} con números aleatorios.²³
2. Iteración i -ésima. Se calcula:

$$Net_i = \mathbf{W}_i \mathbf{X} + \mathbf{a}_i$$

$$\mathbf{Z}_i = \varphi(Net_i)$$

$$\hat{\mathbf{Y}}_i = \delta(\mathbf{V}_i \mathbf{Z}_i + \mathbf{b}_i)$$

Donde: i indica la iteración tal que $i = 1, \dots$

3. Se calcula el error para los nodos de la capa de salida:

$$\mathbf{E}_i = \mathbf{Y} - \hat{\mathbf{Y}}_i \quad (1.16)$$

Donde: \mathbf{E} es una matriz $k \times 1$; $\hat{\mathbf{Y}}$ es la señal de salida calculada.

4. Si $\hat{\mathbf{Y}}_i \neq \mathbf{Y}$, entonces se propaga el error hacia los nodos de la capa oculta y de entrada:

$$\boldsymbol{\zeta}_i = \boldsymbol{\Lambda}_i \mathbf{E}_i \quad (1.17)$$

$$\boldsymbol{\xi}_i = \boldsymbol{\Omega}_i \mathbf{V}_i^T \boldsymbol{\zeta}_i \quad (1.18)$$

²³ Los cuales pueden estar en los intervalos (0,1) o (-1,1), dependiendo si se utiliza la función (1.8) o la (1.9).

Donde: $\boldsymbol{\Lambda} = \text{diag}(\delta'(\bullet)_1, \dots, \delta'(\bullet)_k)$; $\boldsymbol{\Omega} = \text{diag}(\varphi'(\bullet)_1, \dots, \varphi'(\bullet)_h)$; $\boldsymbol{\zeta}$ y $\boldsymbol{\xi}$ son matrices $k \times 1$ y $h \times 1$, respectivamente.

5. Se actualizan los pesos de la capa de entrada y de la capa oculta:

$$\left. \begin{aligned} \mathbf{W}_{i+1} &= \mathbf{W}_i - \eta \boldsymbol{\zeta}_i \mathbf{X}^T \\ \boldsymbol{\alpha}_{i+1} &= \boldsymbol{\alpha}_i - \eta \boldsymbol{\zeta}_i \end{aligned} \right\} \quad (1.19)$$

$$\left. \begin{aligned} \mathbf{V}_{i+1} &= \mathbf{V}_i - \eta \boldsymbol{\xi}_i \mathbf{Z}_i^T \\ \boldsymbol{\beta}_{i+1} &= \boldsymbol{\beta}_i - \eta \boldsymbol{\xi}_i \end{aligned} \right\} \quad (1.20)$$

Donde: η es una constante positiva (conocida como *tasa de aprendizaje*) tal que $0 < \eta < 1$; si ésta es pequeña, la neurona aprenderá con lentitud y, si es grande, lo hará con mayor rapidez (comúnmente, $\eta = 0.5$); T indica transposición.

6. Parada. Se busca minimizar el índice de rendimiento de la red —ecuación (1.21)—; en este caso, el Error Cuadrático Medio. Si no se cumple con el error objetivo, se va al paso dos con $i = i + 1$.

$$\mathbf{e} = \frac{1}{M} \mathbf{E}^T \mathbf{E} \quad (1.21)$$

Donde M es el número de patrones de entrenamiento.

La virtud de este proceso consiste en que, a medida que se entrena la red, los nodos de la capa oculta se organizan a sí mismos de tal modo que las distintas neuronas aprenden a reconocer distintas características de las señales de entrada. Así, cuando se les presente un conjunto arbitrario de señales de entrada que contenga ruido o que esté incompleta, las unidades de la capa oculta responderán con una salida activa si las nuevas entradas contienen información semejante a aquella característica

que las unidades individuales hayan aprendido a reconocer durante su entrenamiento; y a la inversa, las unidades de las capas ocultas tienen una tendencia a inhibir sus salidas si la información de entrada no contiene las características apropiadas. De esta forma, el algoritmo Backpropagation es a un Perceptrón Multicapa lo que el Método de Mínimos Cuadrados Ordinarios es al Modelo de Regresión Lineal.

En la actualidad, existen diferentes variantes del algoritmo BP como el *Levenberg-Marquardt* (BP-LM), el cual consiste en calcular y actualizar los pesos de una matriz Ψ que contiene los pesos de toda la red,²⁴ a través de la siguiente expresión:²⁵

$$\Psi_{i+1} = \Psi_i - \left(\mathbf{J}_i^T \mathbf{J}_i + \lambda_i \mathbf{I} \right)^{-1} \mathbf{J}_i^T \mathbf{E}_i \quad (1.22)$$

Donde: $\mathbf{E}_i = \mathbf{Y} - \hat{\mathbf{Y}}_i$; $\hat{\mathbf{Y}}_i$ es la señal de salida calculada; \mathbf{J} es la matriz Jacobiana; λ es un parámetro de ajuste que se modifica en cada iteración usando la tasa de aprendizaje η tal que $0 < \eta < 1$; T indica transposición.

Como señalan Abolfazl *et al.* (2005), el algoritmo BP-LM tiene un excelente desempeño en el entrenamiento de un PM donde su rendimiento esté determinado por el Error Cuadrático Medio, ya que esta regla de aprendizaje converge a una solución estable (mínimo local o global) en menos iteraciones; sin embargo, a diferencia del algoritmo BP, el BP-LM requiere más cálculos por iteración.

9. Estadísticos de bondad de ajuste y correcta especificación

Como señalan McNelis y McMenamin (1996), el complejo sistema no lineal que representa una RNA necesita que se tomen en cuenta los siguientes criterios para su correcta especificación:

1. Número de neuronas en la capa oculta: determinar la dimensión de la capa oculta, es la parte más complicada en la construcción de una arquitectura neuronal; gran parte de la eficiencia de una RNA recaerá en la elección del número de neuronas ocultas. Aunque se trata (en esencia) de un proceso *ensayo-error*, existen diversas propuestas que pueden ayudar a fijarlas:

$$h = \sqrt{n \times k} \quad (1.23)$$

$$h = 2k \pm 1 \quad (1.24)$$

$$h = \frac{M/10}{n(k+1)} - 1 \quad (1.25)$$

Donde: n , h y k son el número de nodos en la capa de entrada, oculta y de salida, respectivamente; M es el número de patrones de entrenamiento.

Cabe mencionar que un número muy grande de nodos ocultos hará que el sistema se ajuste demasiado a las singularidades de los inputs durante el entrenamiento, aprendiendo incluso el ruido presente en ellos, por lo que perderá capacidad de generalización.

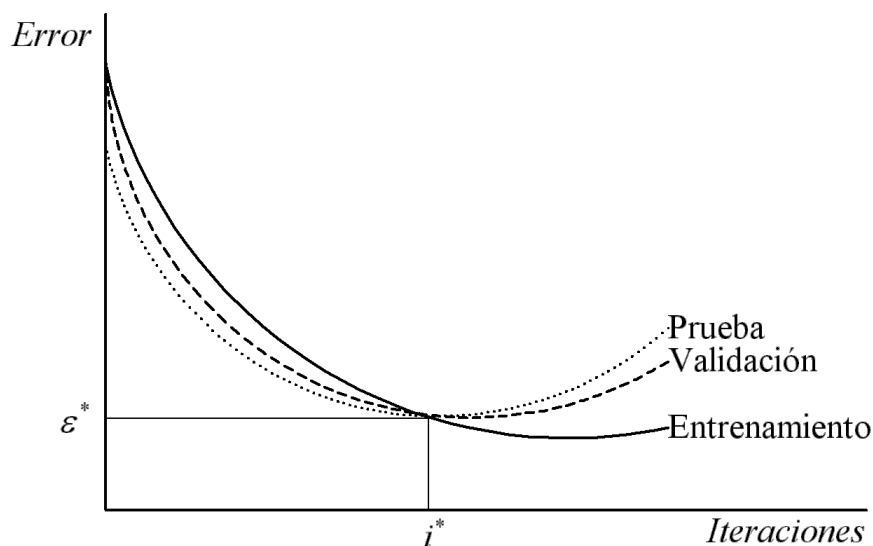
²⁴ Es decir, $\psi = \{W, V\}$..

²⁵ Para consultar un desarrollo más extenso y elegante, véase Hagan y Menhaj (1994).

2. Validación cruzada: el excesivo ajuste de la red neuronal a las particularidades de los patrones empleados en la fase de entrenamiento es un fenómeno conocido como *sobreaprendizaje*. Una RNA con *sobreaprendizaje* o *sobreajuste* no puede generalizar satisfactoriamente las relaciones subyacentes en los datos; por esta razón, el algoritmo BP necesita que las señales de entrada se dividan en tres conjuntos: 1) de entrenamiento, 2) de validación y 3) de prueba. De esta forma, cuando la red actualice los pesos de la capa de entrada

y de la capa oculta —ecuaciones (1.19) a (1.20)— se podrá observar el proceso de cómo el error de entrenamiento y el de validación disminuyen (a medida que aumenta el número de iteraciones) hasta alcanzar un mínimo en la superficie del error; mientras que, al mismo tiempo, los datos de prueba proporcionan una estimación insesgada del error de generalización. Este proceso se conoce como *validación cruzada* (gráfica 1.9) y sirve para detectar el punto óptimo de aprendizaje.

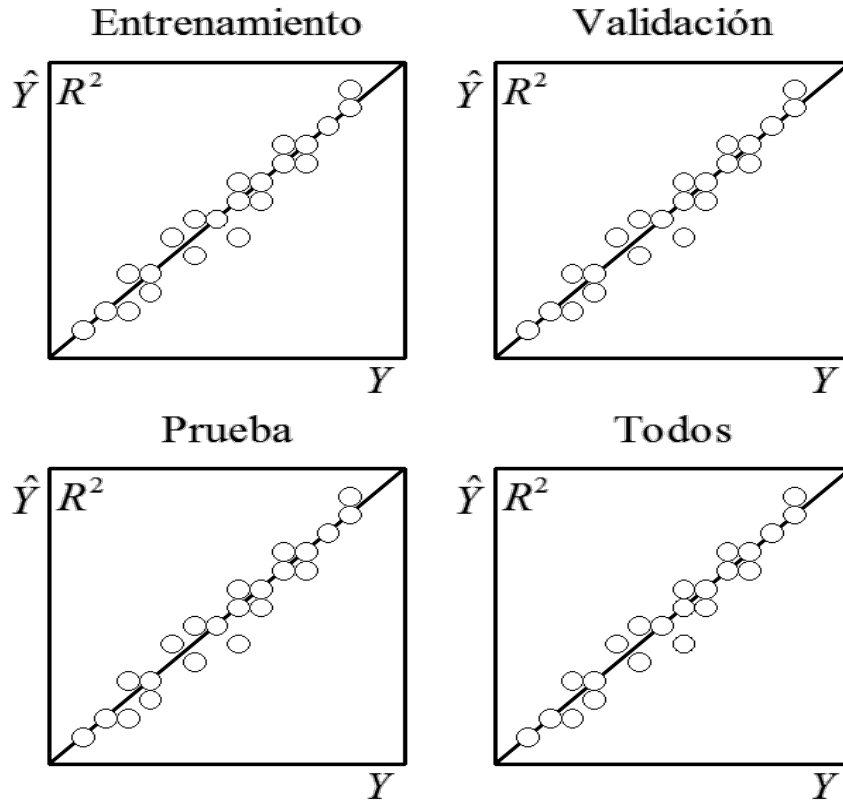
Gráfica 9 Validación cruzada



Además de lo anterior, podemos realizar un análisis de regresión lineal entre las señales de salida observadas y las calculadas a través de los

conjuntos de datos de entrenamiento, de validación y de prueba (gráfica 10.), para conocer la bondad de ajuste de la arquitectura neuronal.

Gráfica 10 / Análisis de regresión



42

3. Criterios dentro de la muestra:²⁶ si se desea realizar un análisis de inferencia a través de una RNA, entonces es necesario un apropiado ajuste entre la señal de salida calculada y la observada, así como verificar que se satisfagan las propiedades estadísticas de los errores de estimación, esto es: normalidad, homoscedasticidad y ausencia de autocorrelación serial. A continuación, se presentan los estadísticos básicos dentro de la muestra:

3.1 Bondad de ajuste. Indica la proximidad existente entre las señales de salida observadas y las calculadas a través de los conjuntos de datos de entrenamiento, de validación y de prueba.

$$R^2 = 1 - \frac{\mathbf{E}^T \mathbf{E}}{(\mathbf{Y} - \bar{\mathbf{Y}})^T (\mathbf{Y} - \bar{\mathbf{Y}})} \quad (1.26)$$

$$\bar{R}^2 = 1 - (1 - R^2) \frac{M-1}{M-g} \quad (1.27)$$

²⁶ Los datos dentro de la muestra se conforman por los conjuntos de señales de entrenamiento y de validación.

Donde: $g = (n+1)h + (h+1)k$ es el número de parámetros estimados y M el número total de observaciones.

3.2 Error Estándar. Es una medida basada en la varianza estimada de los residuales.

$$s = \sqrt{\frac{\mathbf{E}^T \mathbf{E}}{M - g}} \quad (1.28)$$

3.3 Normalidad. El estadístico Jarque-Bera permite contrastar la hipótesis de normalidad en los residuos:

$$JB = \frac{M}{6} \left(S^2 + \frac{(K-3)^2}{4} \right) \quad (1.29)$$

Donde: K es la kurtosis y S , el sesgo.

3.4 Homoscedasticidad. El estadístico de Engle sirve para comprobar la ausencia de Heteroscedasticidad Condicional Autorregresiva; en una arquitectura bien especificada la varianza debe ser constante:

$$ARCH = \mu_0 + \sum_{\tau=1}^g \mu_{\tau} \mathbf{E}_{t-\tau}^2 + v_t \quad (1.30)$$

Donde: $v \sim N(0, \sigma^2 \mathbf{I})$.

3.5 No autocorrelación serial. Para evidenciar autocorrelación serial de cualquier orden se utiliza el estadístico Ljung-Box —ecuación (1.31)—, el cual examina la significancia conjunta de las primeras l autocorrelaciones con una distribución X^2 con L grados de libertad:

$$Q(L) = (M)(M+2) \sum_{l=1}^L \frac{\rho_l^2(\mathbf{E})}{(M-l)} \quad (1.31)$$

Donde: $\rho_l(\mathbf{E})$ es la función de autocorrelación parcial.

3.6 Criterios de información. Los estadísticos de Akaike y de Schwarz —ecuaciones (1.32) y (1.33), respectivamente— resultan herramientas muy útiles cuando se determina el número óptimo de nodos en la capa oculta o para comparar modelos:

$$AIC = -\frac{2LV}{M} + \frac{2g}{M} \quad (1.32)$$

$$SC = -\frac{2LV}{M} + \frac{g \log M}{M} \quad (1.33)$$

Donde: $LV = \frac{M}{2} (1 + \log(2\pi)) + \log \left(\frac{\mathbf{E}^T \mathbf{E}}{M} \right)$

4. Criterios fuera de la muestra: 27 finalmente, si se desea construir una RNA únicamente para fines de pronóstico, entonces se debe contrastar su desempeño predictivo o de generalización a través de dos métodos complementarios: 1) el gráfico y 2) el estadístico. El primero consiste en analizar visualmente la simulación histórica de $\hat{\mathbf{Y}}$; el segundo, en evaluar los siguientes criterios:

4.1 Error Medio.

$$EM = \frac{1}{M} \sum_{m=1}^M \mathbf{E}_m \quad (1.34)$$

4.2 Error Medio Absoluto:

$$EMA = \frac{1}{M} \sum_{m=1}^M |\mathbf{E}_m| \quad (1.35)$$

27 Los datos fuera de la muestra se conforman por el conjunto de señales de prueba.

Error Cuadrático Medio:

$$ECM = \frac{1}{M} \mathbf{E}^T \mathbf{E} \quad (1.36)$$

Raíz del Error Cuadrático Medio:

$$RECM = \sqrt{\frac{1}{M} \mathbf{E}^T \mathbf{E}} \quad (1.37)$$

Coefficiente de *Theil - U*:

$$Theil - U = \frac{RECM}{\sqrt{\frac{1}{M} \mathbf{E}^{*T} \mathbf{E}^*}} \quad (1.38)$$

Donde: E^* es el error de pronóstico ingenuo.

Conclusiones

En el presente documento se definieron los conceptos básicos y la técnica fundamental de un Perceptrón Multicapa, así como las características principales de la regla de aprendizaje para la construcción de este tipo de arquitecturas: el algoritmo Backpropagation, con el objetivo de proporcionar una guía teórica que pudiese ser entendida por quien cuente con una formación matemática y económica básica.

Bajo el hecho de que la realidad económica es compleja y caótica y no simple y ordenada, el uso de una RNA para el modelado económico permite relajar el supuesto de que los agentes conocen el modelo “real” de la economía: los factores económicos y extraeconómicos que gobiernan el comportamiento de las variables económicas.

Si bien es cierto que la compleja forma funcional de un sistema *caja negra* como las RNA hace difícil (pero no imposible) interpretar económicamente los parámetros estimados, su potencia no consiste necesariamente

en la *elegancia* de una solución simple, sino en la *generalización* de una respuesta concreta a partir de información de la que se desconoce un modelo *a priori*. El objetivo del economista no es simplificar hasta sus últimas consecuencias la realidad, sino proporcionar información de mayor calidad, así como reducir la incertidumbre de los tomadores de decisiones, especialmente en el área de los pronósticos. Sin embargo, es importante tomar en cuenta las palabras de Jain y Mao (1996): “no hay evidencia conclusiva de su superioridad sobre las técnicas convencionales”. Por tanto, el uso de esta técnica reside en el criterio del economista y de la naturaleza de la aplicación. De esta forma, como señala González: “la metodología RNA debe ser vista como un complemento natural al análisis de regresión lineal”. Debido a la creciente complejidad de los problemas a los que se enfrentan los investigadores, es preciso enriquecer los métodos y técnicas de análisis para el procesamiento de información y, de esta forma, obtener soluciones más rápidas y eficaces. 🌐

Bibliografía

- Abolfazl, A., M. Bagher y A. Hoseinabadi (2005) “Modified Levenberg-Marquardt Method for Neural Networks Training”, *Proceedings of World Academy of Science, Engineering and Technology*, Vo. 6.
- Brunak, S. y B. Lautrup (1990). *Neural Networks*, World Scientific, Singapore.
- Flórez, R. y J. M. Fernández (2008). *Las Redes Neuronales Artificiales. Fundamentos teóricos y aplicaciones prácticas*. Serie Metodología y Análisis de Datos en Ciencias Sociales, Netbiblo.
- Freeman, J. A. y D. M. Skapura (1993). *Redes neuronales. Algoritmos, aplicaciones y técnicas de programación*, Addison-Wesley/Diaz de Santos.
- Funahashi, K. I. (1989). “On the approximate realization of continuous mappings by neural Networks”, *Neural Networks*, 2.

- García-Ramos, M. (2009). *Redes Neuronales Artificiales para el pronóstico del PIB: México, 1980.1-2008.3*, UNAM.
- Granger, C. W. J. (1991). "Developments in the Non-linear Analysis of Economic Series", *Scandinavian Journal of Economics*, 93, 2.
- González, S. (2000). "Neural Networks for Macroeconomic Forecasting: A Complementary Approach to Linear Regression Models", *Finance Canada Working Paper*.
- Hagan, M. T. y M. B. Menhaj (1994). "Training Feed-forward Networks with Marquardt Algorithm", *IEEE Transactions on Neural Networks*, Vol. 5, No. 6.
- Haykin, S. (1999). *Neural Networks. A comprehensive foundation*. Second Edition, Prentice Hall.
- Jain, A. K. y J. Mao (1996). "Artificial Neural Networks: A Tutorial", *IEEE*.
- Kimball, A. W. (1957). "Errors of the third kind in statistical consulting", *Journal of the American Statistical Association*, 52.
- Kennedy, P. E. (2002), "Sinning in the Basement: What are the Rules? The Ten Commandments of Applied Econometrics," *Journal of Economic Surveys*, Vol. 16
- Lapedes, A. S. y R. M. Faber (1987). "Nonlinear signal processing using neural networks: Prediction and system modeling", *Technical Report LA-UR-87-2662*.
- Levenberg, K. (1944). "A Method for the Solution of Certain Non-linear Problems in Least Squares". *Quarterly of Applied Mathematics*, 2(2):164-168, Jul.
- Lucas, R. E. (1976). "Econometric Policy Evaluation: A Critique", *Carnegie Rochester Conference Series on Public Policy*, 1.
- Marquardt, D.W. (1963). "An Algorithm for the Least-Squares Estimation of Nonlinear Parameters". *SIAM Journal of Applied Mathematics*, 11(2):431-441, Jun.
- McCulloch, W. S. y W. Pitts (1943). "A Logical Calculus of Ideas Immanent in Nervous Activity", *Bulletin of Mathematical Biophysics*, Vol. 5.
- McMenamin, J. S. (1996). "Why Not Pi? A Primer on Neural Networks for Forecasting", *Technical White Papers*, Itron.
- McNelis, P. D. (2005). *Neural Networks in Finance. Gaining Predictive Edge in the Market*. Academic Press Advanced Finance Series, Elsevier Academic Press.
- Menger, K. (1960). "A counterpart of Occam's razor in pure and applied mathematics ontological uses", *Synthese*, Volume 12, Number 4
- Minsky, M., (1954). *Neural Nets and the Brain-Model Problem*, PhD Dissertation, University of Princeton.
- Minsky, M. y S. Papert (1969). *Perceptrons: An Introduction to Computational Geometry*, MIT Press, Cambridge.
- Rodríguez-Fernández, J. L. (1999). "Ockham's Razor", *Endeavour*, 23.
- Rosenblatt, R (1958). "The Perceptron. A Probabilistic Model for Information Storage and Organization in the Brain", *Psychological Review*, 65.
- Rumelhart, D. E., G. E. Hinton y R. K. Williams (1986). "Learning Representations by Backpropagation Errors", *Nature*, 323.
- Sims, C. A. (1980). "Macroeconomics and reality", *Econometrica*, Vol. 48, 1.
- Trippi, R. R. y E. Turban (1993). *Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real-World Performance*, Probus.
- Tukey, J. W. (1962). "Statistical methods in psychology journals", *American Psychologist*, 54.
- Werbos, P. (1974). "Beyond Regression: New Tools for Prediction and Analysis in the Behavior Sciences", PhD thesis, Dept. of Applied Mathematics, Harvard University, Cambridge.
- Zhang, G., B. E. Patuwo y M. Y. Hu (1997). "Forecasting with Artificial Neural Networks: the State of the Art", *International Journal of Forecasting*, 14.