



Asepelt  
España

# Comunicaciones XIV Reunión

REDES NEURONALES ARTIFICIALES Y MÉTODOS  
DE ESTIMACIÓN DE FUNCIONES

*Federico Palacios González* - [fpalacio@platon.ugr.es](mailto:fpalacio@platon.ugr.es)  
Universidad de Granada

## Anales de Economía Aplicada

Oviedo 2<sup>3</sup>  
Junio 2000 4



Reservados todos los derechos.

Este documento ha sido extraído del CD Rom "Anales de Economía Aplicada. XIV Reunión ASEPELT-España. Oviedo, 22 y 23 de Junio de 2000".

ISBN: 84-699-2357-9

# REDES NEURONALES ARTIFICIALES Y MÉTODOS DE ESTIMACIÓN DE FUNCIONES

**Federico Palacios González**  
*Economía Aplicada - Universidad de Granada*  
*fpalacio@platon.ugr.es*

*Código UNESCO: 530204*

*Palabras clave: Redes neuronales artificiales, estimación de funciones, sistemas lineales y no lineales*

*Area temática: G2*

## Resumen

En este trabajo se muestra que una red neuronal artificial de arquitectura "feedforward" puede interpretarse como un mecanismo muy versátil de construcción de funciones de  $\mathbb{R}^1$  en  $\mathbb{R}^m$ . Se analiza, a la luz de la metodología de estimación de funciones, aquello que en el ámbito de las redes neuronales se llama "aprendizaje de la red". Se muestra que una red con función de transferencia identidad equivale a un sistema lineal. Se explica el mecanismo por el cual, para una información dada y fija, el aprendizaje de una red conduce a múltiples soluciones todas ellas con la misma respuesta frente a cada entrada. Se comprueba, en definitiva que hay multitud de soluciones para una red neuronal (con función de transferencia identidad) que definen el mismo sistema lineal.

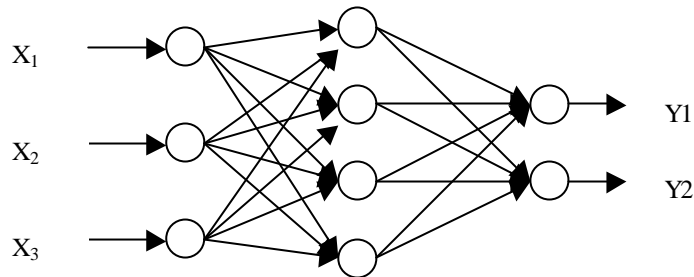
En lo que respecta a las Redes de Expansión se comprueba que son equivalentes a un modelo lineal en los parámetros, uni o multiecuacional dependiendo del número de neuronas en la capa de salida.

Se plantea el problema de la amplitud de la clase de funciones que pueden generar este tipo de redes neuronales. Es decir su capacidad para recoger las dispares relaciones entre variables de entrada y salida propias de las distintas aplicaciones.

## 1.- Introducción.

El tipo de red objeto de este trabajo tiene una capa de entrada o conjunto de neuronas que reciben valores exteriores a la red, una capa de salida con neuronas que emiten al exterior la respuesta elaborada por la red y una o más capas ocultas de neuronas que se interconectan entre sí. Es una red de tipo "feedforward" donde las conexiones son unidireccionales desde una capa a la siguiente, comenzando por la de entrada y terminando por la de salida. Un esquema de red con una sola capa oculta puede verse en la figura 1

**Figura 1: Esquema de red neuronal de tipo feedforward con una capa oculta**



Cada conexión entre neuronas tiene asociado un coeficiente  $w$  positivo o negativo. De este modo, la red correspondiente a la figura 1 ha de poseer dos matrices de coeficientes:  $W_1$  de dimensión  $4 \times 3$  que contienen todas las ponderaciones relativas a las interconexiones entre las tres neuronas de la capa de entrada y las cuatro neuronas de la capa oculta y la matriz  $W^2$  de dimensión  $2 \times 4$  que contiene las ponderaciones correspondientes a las interconexiones entre las dos neuronas de la capa de salida y las cuatro de la capa oculta. Cualquier coeficiente de estas dos matrices se asocia con una conexión entre dos neuronas de la red. Por ejemplo el coeficiente  $w_{2,1}^1$  que ocupa la segunda fila y primera columna de la matriz  $W^1$  corresponde a la conexión que se establece entre la neurona 2 de la capa de entrada y la neurona 1 de la capa oculta. Éste, es el coeficiente de ponderación de un impulso que sale desde la citada neurona 2 de la capa de entrada y llega a la también citada neurona 1 de la capa oculta. El valor cero de un coeficiente puede interpretarse como ausencia de conexión real entre las dos neuronas correspondientes. Obsérvese que las dos matrices  $W^1$  y  $W^2$  recogen y definen la estructura de la red neuronal propuesta en la figura 1.

Sea  $\tau$  la función de transferencia de la red y sea la función vectorial (considérese también su análoga para un vector columna)

$$\mathbf{t}(z_1, z_2, \dots, z_n) = (\mathbf{t}(z_1), \mathbf{t}(z_2), \dots, \mathbf{t}(z_n)) \quad (1)$$

Si  $\mathbf{X}$  representa un vector columna que contiene tres valores cualesquiera de las variables de entrada en la red de la figura 1, e  $\mathbf{Y}$  es el vector columna que contiene los valores de salida, entonces podemos escribir que dicha red es una función definida de  $\mathbb{R}^3$  en  $\mathbb{R}^2$  mediante la expresión analítica

$$Y = \mathbf{t}(W^2 \times \mathbf{t}(W^1 \times X)) \quad (2)$$

Si la función de transferencia es la función identidad se obtiene

$$Y = W^2 \times W^1 \times X \quad (3)$$

Es evidente que  $\mathbf{b}^T = W^2 \times W^1$  es una matriz de dimensión  $2 \times 4$  y por tanto podemos escribir la red neuronal de la siguiente forma:

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = Y = \mathbf{b}^T \times X = \begin{pmatrix} \sum_{j=1}^3 \mathbf{b}_{1,j} \times X_j \\ \sum_{j=1}^3 \mathbf{b}_{2,j} \times X_j \end{pmatrix} \quad (4)$$

La red neuronal se convierte, por tanto, en un simple sistema de dos ecuaciones lineales. Si una de las entradas a la red es constantemente igual a 1, cada ecuación tiene un término constante.

Si la función de transferencia tiene la forma  $\mathbf{t}(x_i) = ax_i + b$ , entonces la red neuronal (2) puede escribirse de la siguiente forma

$$Y = HX + C \quad (5)$$

donde  $H = a^2 W^2 W^1$ ,  $C = a.b.W^1.1_3 + b.1_2$  y donde  $1_s$  representa un vector columna con  $s$  componentes idénticas al valor 1. La red neuronal es un sistema de dos ecuaciones lineales con un término constante en cada ecuación.

## 2.- Análisis formal de los modelos resultantes de una red neuronal de tipo "feedforward".

En general, una red de tipo "feedforward" con  $k-1$  capas ocultas,  $m$  neuronas en la capa de entrada, y  $n$  neuronas en la de salida, se identifica con la función definida de  $R^m$  en  $R^n$  mediante la expresión analítica siguiente

$$Y = T(X) = \mathbf{t}(W^k \times \mathbf{t}(\dots \mathbf{t}(W^1 \times X))) \quad (6)$$

donde necesariamente,  $W^1$  ha de ser una matriz con  $m$  columnas y  $W^k$  ha de ser una matriz con  $n$  filas. Para que la expresión (6) sea coherente, las matrices han de tener dimensiones que permitan efectuar los productos matriciales en ella expresados. Es decir las dimensiones de  $W^k, W^{k-1}, \dots, W^2, W^1$ , han de ser las siguientes:  $n \times n_k, n_k \times n_{k-1}, \dots, n_3 \times n_2, n_2 \times n_1, n_1 \times m$ . Evidentemente esto significa que la  $i$ -ésima capa oculta (para todo  $i$  desde 1 a  $k-1$ ), contiene  $n_i$  neuronas igual al número de filas de la matriz  $W^i$ .

Si utilizamos la función identidad como función de transferencia entonces la expresión (5) queda reducida a la siguiente

$$Y = T(X) = H \times X \quad \text{donde } H = W^k \times \dots \times W^2 \times W^1 \quad (7)$$

Si la función de transferencia es lineal, la red adquiere la forma expresada en (5) aunque el proceso de obtención de  $H$  y  $C$  se prolonga  $k$  etapas y las expresiones de ambos elementos son más complejas.

Las redes neuronales con una función de transferencia lineal quedan reducidas a sistemas lineales. Además, hay multitud de redes neuronales de estructura diferente pero de comportamiento idéntico al único sistema lineal que todas ellas definen (cuando la función de transferencia es la identidad). Esto se debe a que toda matriz rectangular se puede descomponer en producto de dos o más matrices, en un número infinito no numerable de formas

diferentes. Por ejemplo, las dos redes siguientes se pueden considerar idénticas

$$W^2 = (1 \ 2) \quad W^1 = \begin{pmatrix} 1 & 2 & 5 \\ 4 & 3 & 1 \end{pmatrix} \quad (8)$$

$$W^2 = (1 \ 2 \ 1) \quad W^1 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 1 \\ 2 & 4 & 2 \end{pmatrix} \quad (9)$$

ya que ambas estructuras originan el mismo modelo lineal

$$Y = 9X_1 + 8X_2 + 7X_3 \quad (11)$$

¿Tiene alguna ventaja utilizar ocho o doce parámetros para obtener idéntico resultado que con solo tres?

Análoga conclusión se obtiene cuando la función de transferencia es lineal. Aunque las expresiones sean más complejas, todas ellas continúan siendo lineales y los argumentos anteriores son válidos.

Como consecuencia de lo expresado hasta el momento, resulta evidente que los modelos conocidos como redes de expansión, García G. (1998), son equivalentes a modelos lineales en los parámetros. La metodología del modelo lineal general y sus extensiones multiecuacionales les son aplicables. Desde esta perspectiva, dichos modelos de red neuronal no hacen nuevas aportaciones. Por otra parte, el fundamento matemático de esta técnica está perfectamente desarrollado. Es la clásica aproximación de funciones mediante series. En un comienzo polinómicas, posteriormente de Fourier, y de forma más general y abstracta mediante un sistema ortonormal que constituye una base del espacio vectorial de dimensión infinita que contiene a una clase de funciones suficientemente amplia y bien comportada

Cuando la función de transferencia no es lineal, resulta difícil comprobar si hay una situación análoga a la descrita mediante (8) y (9). Es difícil comprobar si existe una función no lineal con un número inferior de parámetros capaz de producir exactamente los mismos resultados que toda una clase no numerable de redes equivalentes (definidas por distintas colecciones de valores en las matrices  $W$  que explicitan su estructura). Si en algún caso llegara a suceder será detectado inmediatamente por que al someter a la red al proceso de aprendizaje con unos mismos datos en reiteradas ocasiones, se observarán soluciones muy dispares pero que producen respuestas prácticamente coincidentes.

## **2.1 Transformación de Variables Aleatorias y Modelos de Regresión.**

En este apartado se muestra cómo una red neuronal puede definir transformaciones de variables aleatorias y modelos de regresión. Se quiere resaltar que en la mayoría de las aplicaciones que se hacen de este tipo de red, de una manera más o menos explícita, se está definiendo y utilizando un modelo de regresión. Se presentan conjuntamente para resaltar algunas diferencias clarificadoras de ideas sobre qué se está haciendo cuando se aplica una red neuronal a problemas concretos.

### **a.- Transformaciones de Variables Aleatorias**

Dado un espacio de probabilidad  $(\Omega, A, P)$ , correspondiente a un experimento aleatorio, una variable aleatoria  $m$ -dimensional es una función medible  $X$  de  $\Omega$  en  $\mathbb{R}^m$ . Por otra parte, una red neuronal con  $m$  neuronas en la capa de entrada y  $n$  en la de salida es una función  $Y=T(X)$  de  $\mathbb{R}^m$  en  $\mathbb{R}^n$ . Dicha función es Borel-medible, entre otros casos, cuando la función de transferencia es continua y acotada. La composición de ambas funciones (variable aleatoria y red neuronal) define una función medible

$$Y = T \circ X : \Omega \rightarrow \mathfrak{R}^n \quad (12)$$

Por tanto,  $Y$  es una variable aleatoria  $n$ -dimensional transformada de  $X$  mediante  $T$ . Para cada  $\mathbf{w} \in \Omega$ , se obtiene un punto  $x = X(\mathbf{w}) \in \mathfrak{R}^m$  y un punto

$$y = T(x) = T(X(\mathbf{w})) \quad (13)$$

Los puntos  $(x, y)$  pertenecen a una Variedad de ecuación (13) contenida en  $\mathfrak{R}^{m+n}$ . Cada realización del experimento aleatorio conduce a un resultado del espacio muestral al que las variables aleatorias  $X$  y su transformada  $Y$ , le hacen corresponder un punto sobre la citada Variedad.

Por ejemplo, consideremos el modelo de red neuronal lineal propuesto en (8) o en (9). Ambos definen la función lineal (11). Aplicado a una variable bidimensional  $(X_1, X_2)$  y a  $X_3 = 1$  constante, da lugar a una variable aleatoria  $Y$  que es combinación lineal de las anteriores. La variable aleatoria  $(X_1, X_2, Y)$  es tal que para cada  $\mathbf{w} \in \Omega$  define un punto que se sitúa sobre el plano de ecuación

$$Y = 9x_1 + 8x_2 + 7 \quad (14)$$

inmerso en  $\mathfrak{R}^3$ . Las realizaciones de la variable  $(X_1, X_2, Y)$  son puntos que se distribuyen aleatoriamente sobre el plano de ecuación (14).

### **b.- Modelos de Regresión.**

Una forma alternativa de relacionar variables aleatorias es definir la función de regresión entre dos vectores aleatorios mediante una red neuronal, de manera que para cada realización  $x$  de la variable  $m$ -dimensional  $X$  se defina la esperanza del vector  $n$ -dimensional  $Y$  condicionada a  $x$  como

$$E[Y / x] = T(x) \quad (15)$$

donde  $T(x)$  es una expresión como la definida en (6). De esta forma, puede afirmarse que para cada  $\mathbf{w} \in \Omega$

$$Y(\mathbf{w}) = T(X(\mathbf{w})) + u(\mathbf{w}) \quad (16)$$

donde  $u$  es un vector aleatorio  $n$ -dimensional cuyo valor esperado es  $(0) \in \mathfrak{R}^n$ .

Ahora las realizaciones de  $(X, Y)$  no se sitúan exactamente sobre la variedad de ecuación (13) sino relativamente próximos (alrededor de ella) dependiendo de las varianzas asociadas con los distintos elementos que componen el vector aleatorio  $u$ . En particular, utilizando la misma red lineal que condujo a (14), ahora, los puntos  $(X_1, X_2, Y)$  no son realizaciones aleatorias sobre el plano sino en regiones, (alrededor de él), más o menos próximas dependiendo de la varianza de  $u$ .

Obsérvese que a pesar de tratarse de relaciones entre variables aleatorias, (13) establece una relación funcional o determinista entre los vectores  $X$  e  $Y$ , mientras que (16) establece una relación de tipo estocástico entre dichos vectores aleatorios. En (13), dada una realización (aleatoria) de  $X$  automáticamente queda determinado la realización de  $Y$  (que también es aleatoria pero perfectamente conocida una vez que se conoce la de  $X$ ). En (16), dada una realización de  $X$ , queda determinado el valor esperado de la correspondiente realización de  $Y$  pero no es posible determinar el valor concreto de  $Y$  a partir del de  $X$ .

## **2.2 Capacidad de construir modelos de una red neuronal tipo feedforward.**

Ya se ha visto que una red de este tipo es una relación funcional de  $R^m$  en  $R^n$  con  $m$  y  $n$  determinados por la estructura de la red. La pregunta es si la metodología es suficientemente flexible. Es decir si dada una función de  $R^m$  en  $R^n$  perteneciente a una cierta clase de funciones medianamente bien comportadas (con ciertas condiciones de continuidad y poca rugosidad) siempre existe una red neuronal tan próxima ella como se quiera. En el argot del Análisis Matemático el problema consiste en demostrar que la clase  $C_n$  de redes neuronales es densa en una clase  $C$  más amplia de funciones. Desde el punto de vista aplicado habría que añadir que, además,  $C$  es suficientemente amplia para proporcionar el modelo necesario en una gran mayoría de aplicaciones concretas. En cualquiera de estos casos, una red neuronal de tipo feedforward sería una buena solución.

En el trabajo de Weaver, S. y otros (1988), se afirma que cualquier función continua puede ser perfectamente aproximada sobre un dominio compacto, con una red neuronal con función de transferencia sigmoïdal y con el número suficiente de neuronas en sus capas ocultas. El artículo de Ritter, G. (1999) hace referencia a la propiedad de densidad de este tipo de redes en la clase de las funciones continuas sobre un conjunto compacto.

Por otra parte, los conceptos acuñados en el ámbito de las redes neuronales artificiales sobre “aprendizaje” y “memoria” corroboran este hecho. En efecto; en la mayor parte de las aplicaciones de este tipo de redes se tiene una respuesta  $Y$   $n$ -dimensional, no determinista, a valores observados de un vector aleatorio  $m$ -dimensional que actúa como estímulo o entrada de la red. Se plantea un sistema de regresión (lineal o no lineal) de  $Y$  sobre otro  $X$ . Es generalmente aceptado que no se debe exagerar la cantidad de neuronas en las capas ocultas de la red, porque la red memoriza las cantidades de entrada y salida en lugar de aprender la relación estructural existente entre las variables, de forma que responde perfectamente cuando se introducen datos utilizados para su entrenamiento, y sin embargo es incapaz de aproximar una respuesta coherente cuando se le suministran nuevos datos de entrada, aún cuando pertenezcan al cierre convexo engendrado por los valores de  $X$  utilizados en el aprendizaje supervisado. En el ámbito de la regresión no paramétrica se afirmaría que se ha construido una función demasiado flexible y rugosa, cuya gráfica pasa por todos y cada uno de los puntos del diagrama de dispersión. No se ha producido ningún tipo de suavizamiento, el modelo ha recogido toda la componente perturbadora y su capacidad de predicción es mala. Hay que aumentar el parámetro de suavizamiento para que la función resultante sea menos rugosa y evitar que atraviese cada uno de los puntos del diagrama de dispersión. Obsérvese que el dilema rugosidad, suma de cuadrados residual que se está planteando resulta muy similar al dilema “memoria”, “aprendizaje” y sobre todo, parece poner de manifiesto la enorme flexibilidad de una estructura de red neuronal que ocasiona respuestas rugosas y que contienen todos los puntos del diagrama de dispersión. Esto, básicamente es contestar afirmativamente al problema planteado. Respuesta

comprobada de forma empírica por los múltiples practicantes de aplicaciones de esta metodología, los cuales incluso han propuesto una regla para el número óptimo de neuronas a utilizar: Una o dos capas internas con un número total de neuronas cercano al logaritmo en base 2 del número de vectores de entrenamiento disponibles. Podría decirse que en el ámbito de las redes neuronales con arquitectura feedforward, el equivalente al parámetro de suavizamiento (bandwidth) es el número de neuronas en las capas internas y el propio número de capas internas de la red (en relación inversa).

### **3.- Reflexiones sobre los algoritmos de aprendizaje supervisado.**

El algoritmo inicial para el aprendizaje de este tipo de redes es conocido bajo el nombre de propagación hacia atrás (Back-Propagation). Dicho algoritmo tiene su origen en el año 1986 y es descubierto por tres grupos de investigación simultáneamente Omlsted (1998). En años recientes se han publicado multitud de versiones de dicho algoritmo para mejorar, acelerar e interpretar el proceso de aprendizaje de la red. En los tres últimos años, el número de trabajos que pretenden mejorarlo, es tan elevado que solamente los localizados hasta el momento de redactar esta comunicación obligan a una referencia colectiva, como la presente, remitiendo al lector al listado que se facilita en el apartado 5.1

En general, el aprendizaje supervisado se basa en la presentación sucesiva de los datos de entrada  $X$  y comparación de la respuesta de la red con los valores  $Y$  reales observados. Se mide la distancia entre respuesta y observaciones y se modifican los parámetros de la red hasta que la distancia se hace mínima. Cada vector de aprendizaje  $(X_i, Y_i)$  (cada dato) es sometido a la red de forma secuencial. La secuencia se repite multitud de veces hasta que la red se estabiliza y deja de sufrir modificaciones. Supongamos que se dispone de  $k$  vectores de entrenamiento. Para cada  $i=1,2, \dots, k$  se minimiza la distancia (euclídea generalmente)

$$d(Y_i, \hat{Y}_i) \quad (17)$$

siendo  $\hat{Y}_i = T(X_i)$  la respuesta de la red al estímulo representado por  $X_i$ . La secuencia se repite hasta que se estabilizan las modificaciones de los parámetros de la red. El nombre del algoritmo se debe a que el proceso de modificación de dichos parámetros, no se realiza en una etapa, sino en varias yendo desde la capa de salida hacia atrás: Se modifican los parámetros de la matriz  $W^k$  hasta alcanzar el mínimo de (17), a continuación los de  $W^{k-1}$ , y así sucesivamente hasta  $W^1$ . El mínimo de (17) se suele obtener mediante métodos de aproximación numérica como el de Newton o el del gradiente mínimo o máximo descenso Burden y Faires (1985). El primero exige el cálculo del Jacobiano de la función a minimizar, y el segundo necesita del vector gradiente. En ambos casos manejar simultáneamente todas las capas de la red puede convertir el proceso analítico de derivación en un problema muy voluminoso y de difícil tratamiento. Es por ello que se obtienen mínimos parciales por capas. No obstante, si se utiliza el método de Broyden para aproximar una derivada parcial mediante el correspondiente cociente incremental, (Burden y Fayres (1985)), para obtener el gradiente o el jacobiano en un punto concreto, sólo es necesario estar capacitado para valorar con eficacia la función (la red neuronal). En este caso puede encontrarse el conjunto de matrices  $W_i$   $i=1,2,\dots,K$  de parámetros que minimizan (17), utilizando un algoritmo de optimización que trabaja simultáneamente con todas las capas de la red en lugar de hacerlo capa a capa. Actualmente existen programas para PC que son muy eficaces y capaces de encontrar el óptimo de una función de muchas variables sin más que proporcionarle un método para valorar la función en cualquier punto. Por ejemplo existe una aplicación para la hoja de cálculo Excel, conocida con el nombre de "Solver", desarrollada por las Universidades Leon Landon de Austin (Texas) y Allan Waren (Cleveland).

Si se dispone de todos los vectores de entrenamiento en el momento de someter la red a un proceso de aprendizaje puede presentársele toda la información simultáneamente y en una sola ocasión en lugar de hacerlo secuencialmente y de forma repetida. En efecto, puesto que (17) siempre tomará valores superiores o iguales a cero es equivalente encontrar los

parámetros que minimizan simultáneamente cada sumando de (18) que encontrar aquellos que minimizan la suma en su conjunto

$$Q = \sum_{i=1}^k d(Y_i, \hat{Y}_i) \quad (18)$$

Parece más eficaz minimizar (18), en su conjunto y presentando toda la información solamente una vez, que buscar mínimos de cada sumando y reiterando el proceso cuantas veces sea necesario hasta alcanzar un punto fijo. Frecuentemente el método tradicional conduce un mínimo local, según se reitera de forma continua en las publicaciones recogidas en 5.1.

Obsérvese que si en (18) se utiliza la distancia euclídea, definida mediante la matriz identidad, estamos planteando el problema de ajustar un sistema multiecuacional no lineal por el método de mínimos cuadrados ordinarios. Si la métrica se define mediante una matriz definida positiva distinta de la matriz identidad el problema será de mínimos cuadrados generalizados. Si se utiliza otra métrica como la suma de los valores absolutos de las diferencias de las componentes, se está planteando un problema de regresión no lineal robusta asociados con una función de regresión definida con la mediana de las distribuciones condicionadas. Cualquier otra métrica de las utilizadas en el ámbito de la regresión robusta es también factible.

#### **4.- Conclusiones.**

La mayoría de las aplicaciones de redes neuronales de tipo feed forward, en los diferentes campos científicos, consiste en definir un sistema no lineal de regresión que relaciona un vector de variables de entrada  $X$  con otro de variables de salida  $Y$ . La metodología de este tipo de redes podría por tanto clasificarse como un método de regresión no lineal.

Las redes con transferencia lineal son equivalentes a un sistema de regresión lineal cuyo número de parámetros es manifiestamente inferior a los de dicha red y cuya metodología de estimación es bastante más rápida y perfecta. El problema clásico de la multicolinealidad aparece también en este ámbito García G. (1998)

La gran ventaja que ofrecen las redes de tipo "feedforward" es su capacidad para construir modelos. Con el auxilio de k matrices rectangulares (generalmente 2 ó 3 matrices) y una función real de variable real continua (también con funciones impulso bivaluadas), se puede construir el modelo adecuado para estudiar casi cualquier sistema no lineal resultantes en los diversos campos científicos.

Las redes neuronales de tipo feedforward no hacen una aportación importante a los métodos de estimación de funciones, ni desde el punto de vista conceptual ni desde el de la estimación de sus parámetros. Pero si proporcionan un mecanismo muy sencillo para la construcción de sistemas no lineales.

Por el contrario, los algoritmos de aprendizaje utilizados para este tipo de redes pueden completarse con los bien estudiados y abundantes métodos de análisis numérico para la optimización de funciones, desarrollados durante los últimos 30 años. Muchos de ellos se han empleado en el ámbito de la regresión no lineal desde antes de los años 80 (por ejemplo véase el magnífico libro sobre regresión no lineal de Siebel y Wild (1989)). Trabajos como los de Derks y Buydens (1998), Zhou-G Si-J (1998), Yam- JYF y Chow-TWS (1997), Hertz y otros (1997), parecen apuntar en esta dirección.

## 5.- Referencias bibliográficas

- .- **Burden R.L. y Faires J.D (1985)**; *Análisis Numérico*; Grupo Editorial Iberoamericana México 1985.
- .- **García Gëumes A. (1998)**; Una Red Neuronal de Expansión Multi-Output; XII Reunión Anual de ASEPELT España Córdoba 1997 CD-Actas
- .- **García Gëumes A. (1998)**; El problema de la relación entre variables de entrada en las Redes Neuronales; XII Reunión Anual de ASEPELT España Córdoba 1997 CD-Actas
- .- **Seber G.A.F. y Wild C. (1989)**; *Nonlinear Regression*; John Wiley & Sons Inc.; New York (1998)
- .-**Weaver-S Baird-L Polycarpou-MM (1998)**; An Analytical Framework for Local Feedforward Networks; IEEE TRANSACTIONS ON NEURAL NETWORKS 1998, Vol 9, Iss 3, pp 473-482.
- .-**Ritter G.(1999)**; Efficient Estimation of Neural Weights by Polynomial-Approximation; IEEE TRANSACTIONS ON INFORMATION THEORY 1999, Vol 45, Iss 5, pp. 1541-1550.
- .- **Omlsted D.D. (1998)**;History and Principles of Neural Networks [<http://neurocomputing.org/history.htm>].

## 5.1 Listado de trabajos cuyo objetivo es mejorar el algoritmo de backpropagation.

### Año 1999

- .-**Campolucci-P Uncini-A Piazza-F Rao-BD (1999)**; Online Learning Algorithms for Locally Recurrent Neural Networks; IEEE TRANSACTIONS ON NEURAL NETWORKS 1999, Vol 10, Iss 2, pp 253-271.
- .-**Cavalieri-S Mirabella-O (1999)**; A Novel Learning Algorithm Which Improves the Partial Fault-Tolerance of Multilayer Neural Networks; NEURAL NETWORKS 1999, Vol 12, Iss 1, pp 91-106.
- .-**Gustafson-K (1999)**; Internal Sigmoid Dynamics in Feedforward Neural Networks; CONNECTION SCIENCE 1998, Vol 10, Iss 1, pp 43-73.
- .-**Holm-JEW Botha-EC (1999)**; Leap-Frog Is a Robust Algorithm for Training Neural Networks; NETWORK-COMPUTATION IN NEURAL SYSTEMS 1999, Vol 10, Iss 1, pp 1-13.
- .-**Jin-LA Gupta-MM (1999)**; Stable Dynamic Backpropagation Learning in Recurrent Neural Networks; IEEE TRANSACTIONS ON NEURAL NETWORKS 1999, Vol 10, Iss 6, pp 1321-1334
- .-**Kamarthi-SV Pittner-S (1999)**; Accelerating Neural-Network Training Using Weight Extrapolations; NEURAL NETWORKS 1999, Vol 12, Iss 9, pp 1285-1299.
- .-**Kim-J Kasabov-N (1999)**; Hyfis - Adaptive Neuro-Fuzzy Inference Systems and Their Application to Nonlinear Dynamical-Systems; NEURAL NETWORKS 1999, Vol 12, Iss 9, pp 1301-1319.
- .-**Magoulas-GD Vrahatis-MN Androulakis-GS (1999)**; Improving the Convergence of the Backpropagation Algorithm Using Learning Rate Adaptation Methods; NEURAL COMPUTATION 1999, Vol 11, Iss 7, pp 1769-1796.
- .-**Mandic-DP Chambers-JA (1999)**; Exploiting Inherent Relationships in Rnn Architectures; NEURAL NETWORKS 1999, Vol 12, Iss 10, pp 1341-1345.
- .-**Mandic-DP Chambers-JA (1999)**; Relating the Slope of the Activation Function and the Learning Rate Within a Recurrent Neural-Network; NEURAL COMPUTATION 1999, Vol 11, Iss 5, pp 1069-1077.
- .-**Oh-SH Lee-SY (1999)**; A New Error Function at Hidden Layers for Fast Training of Multilayer Perceptrons; IEEE TRANSACTIONS ON NEURAL NETWORKS 1999, Vol 10, Iss 4, pp 960-964.
- .-**Ponnappalli-PVS Ho-KC Thomson-M (1999)**; A Formal Selection and Pruning Algorithm for Feedforward Artificial Neural-Network Optimization; IEEE TRANSACTIONS ON NEURAL NETWORKS 1999, Vol 10, Iss 4, pp 964-968.
- .-**Schmitz-GPJ Aldrich-C (1999)**; Combinatorial Evolution of Regression Nodes in Feedforward Neural Networks; NEURAL NETWORKS 1999, Vol 12, Iss 1, pp 175-189.
- .-**Treadgold-NK Gedeon-TD (1999)**; Exploring Constructive Cascade Networks; IEEE TRANSACTIONS ON NEURAL NETWORKS 1999, Vol 10, Iss 6, pp 1335-1350
- .-**Walczak-S Sincich-T (1999)**; A Comparative-Analysis of Regression and Neural Networks for University Admissions; INFORMATION SCIENCES 1999, Vol 119, Iss 1-2, pp 1-20
- .-**Wang-CA Principe-JC (1999)**; Training Neural Networks with Additive Noise in the Desired Signal; IEEE TRANSACTIONS ON NEURAL NETWORKS 1999, Vol 10, Iss 6, pp 1511-1517.
- .-**Zhang-YM Li-XR (1999)**; A Fast U-D Factorization-Based Learning Algorithm with Applications to Nonlinear-System Modeling and Identification; IEEE TRANSACTIONS ON NEURAL NETWORKS 1999, Vol 10, Iss 4, pp 930-938.
- .-**Zhou-WY (1999)**; Verification of the Nonparametric Characteristics of Backpropagation Neural Networks for Image Classification; IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING 1999, Vol 37, Iss 2, pp 771-779.

## Año 1998

- .-Bilski-J Rutkowski-L (1998)**; A Fast Training Algorithm for Neural Networks; IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II-ANALOG AND DIGITAL SIGNAL PROCESSING 1998, Vol 45, Iss 6, pp 749-753.
- .-Boudreau-R Darenfed-S Gosselin-CM (1998)**; On the Computation of the Direct Kinematics of Parallel Manipulators Using Polynomial Networks; IEEE TRANSACTIONS ON SYSTEMS MAN AND CYBERNETICS PART A-SYSTEMS AND HUMANS 1998, Vol 28, Iss 2, pp 213-220.
- .-Breda-M (1998)**; Appendix 1 - Convergence Aspects on Back-Propagation Neural Networks; SUBSTANCE USE & MISUSE 1998, Vol 33, Iss 2, pp 503+
- .-Derks-EPPA Buydens-LMC (1998)**; Aspects of Network Training and Validation on Noisy Data - Part 1 - Training Aspects; CHEMOMETRICS AND INTELLIGENT LABORATORY SYSTEMS 1998, Vol 41, Iss 2, pp 171-184.
- .-Fukuoka-Y Matsuki-H Minamitani-H Ishida-A (1998)**; A Modified Backpropagation Method to Avoid False Local Minima; NEURAL NETWORKS 1998, Vol 11, Iss 6, pp 1059-1072.
- .-Gori-M Tsoi-AC (1998)**; Comments on Local Minima Free Conditions in Multilayer Perceptrons; IEEE TRANSACTIONS ON NEURAL NETWORKS 1998, Vol 9, Iss 5, pp 1051-1053;
- .-Hahnloser-R (1998)**; Learning Algorithms Based on Linearization; NETWORK-COMPUTATION IN NEURAL SYSTEMS 1998, Vol 9, Iss 3, pp 363-380.
- .-Hamey-LGC (1998)**; Xor Has No Local Minima - A Case-Study in Neural-Network Error Surface-Analysis; NEURAL NETWORKS 1998, Vol 11, Iss 4, pp 669-681.
- .-Huang-L Zhang-BL Huang-Q (1998)**; Robust Interval Regression-Analysis Using Neural Networks; FUZZY SETS AND SYSTEMS 1998, Vol 97, Iss 3, pp 337-347;
- .-Hush-DR Horne-B (1998)**; Efficient Algorithms for Function Approximation with Piecewise-Linear Sigmoidal Networks; IEEE TRANSACTIONS ON NEURAL NETWORKS 1998, Vol 9, Iss 6, pp 1129-1141.
- .-Lu-SW Basar-T (1998)**; Robust Nonlinear-System Identification Using Neural-Network Models; IEEE TRANSACTIONS ON NEURAL NETWORKS 1998, Vol 9, Iss 3, pp 407-429.
- .-Magoulas-GD Vrahatis-MN Androulakis-GS (1998)**; On the Alleviation of the Problem of Local Minima in Backpropagation; NONLINEAR ANALYSIS-THEORY METHODS & APPLICATIONS 1997, Vol 30, Iss 7, pp 4545-4550.
- .-Ratray-M Saad-D (1998)**; Analysis of Online Training with Optimal Learning Rates; PHYSICAL REVIEW E 1998, Vol 58, Iss 5, pp 6379-6391
- .-Redei-L Fried-M Barsony-I Wallinga-H (1998)**; A Modified Learning-Strategy for Neural Networks to Support Spectroscopic Ellipsometric Data Evaluation; THIN SOLID FILMS 1998, Vol 313, Iss FEB, pp 149-155.
- .-Sundareshan-MK Condarcure-TA (1998)**; Recurrent Neural-Network Training by a Learning Automaton Approach for Trajectory Learning and Control-System Design; IEEE TRANSACTIONS ON NEURAL NETWORKS 1998, Vol 9, Iss 3, pp 354-368.
- .-Treadgold-NK Gedeon-TD (1998)**; Title: Simulated Annealing and Weight Decay in Adaptive Learning - The Sarprop Algorithm; IEEE TRANSACTIONS ON NEURAL NETWORKS 1998, Vol 9, Iss 4, pp 662-668.
- .-Tseng-P (1998)**; An Incremental Gradient(-Projection) Method with Momentum Term and Adaptive Stepsize Rule; SIAM JOURNAL ON OPTIMIZATION 1998, Vol 8, Iss 2, pp 506-531.
- .-West-AHL Saad-D (1998)**; Role of Biases in Online Learning of 2-Layer Networks; PHYSICAL REVIEW E 1998, Vol 57, Iss 3, pp 3265-3291.
- .-Zhou-G Si-J (1998)**; Advanced Neural-Network Training Algorithm with Reduced Complexity Based on Jacobian Deficiency; IEEE TRANSACTIONS ON NEURAL NETWORKS 1998, Vol 9, Iss 3, pp 448-453.

## Año 1997

- .-**Arad-BS Elamawy-A (1997)**; On Fault-Tolerant Training of Feedforward Neural Networks; NEURAL NETWORKS 1997, Vol 10, Iss 3, pp 539-553.
- .-**Chow-TWS Cho-SY (1997)**; An Accelerated Recurrent Network Training Algorithm Using Iir Filter Model and Recursive Least-Squares Method; IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I-FUNDAMENTAL THEORY AND APPLICATIONS 1997, Vol 44, Iss 11, pp 1082-1086.
- .-**Dai-HC Macbeth-C (1997)**; Effects of Learning Parameters on Learning Procedure and Performance of a Bpnn; NEURAL NETWORKS 1997, Vol 10, Iss 8, pp 1505-1521.
- .-**Freeman-JAS Saad-D (1997)**; Online Learning in Radial Basis Function Networks; NEURAL COMPUTATION 1997, Vol 9, Iss 7, pp 1601-1622.
- .-**Hertz-J Krogh-A Lautrup-B Lehmann-T (1997)**; Nonlinear Backpropagation - Doing Backpropagation Without Derivatives of the Activation Function; IEEE TRANSACTIONS ON NEURAL NETWORKS 1997, Vol 8, Iss 6, pp 1321-1327.
- .-**Heskes-T Coolen-J**; Learning in 2-Layered Networks with Correlated Examples; JOURNAL OF PHYSICS A-MATHEMATICAL AND GENERAL 1997, Vol 30, Iss 14, pp 4983-4992.
- .-**Hochreiter-S Schmidhuber-J (1997)** Long Short-Term-Memory; NEURAL COMPUTATION 1997, Vol 9, Iss 8, pp 1735-1780.
- .-**Kogiantis-AG Papantonikakos-T (1997)**; Operations and Learning in Neural Networks for Robust Prediction; IEEE TRANSACTIONS ON SYSTEMS MAN AND CYBERNETICS PART B-CYBERNETICS 1997, Vol 27, Iss 3, pp 402-411.
- .-**Kwok-TY Yeung-DY (1997)**; Objective Functions for Training New Hidden Units in Constructive Neural Networks; IEEE TRANSACTIONS ON NEURAL NETWORKS 1997, Vol 8, Iss 5, pp 1131-1148.
- .-**Lee-CW (1997)**; Training Feedforward Neural Networks - An Algorithm Giving Improved Generalization; NEURAL NETWORKS 1997, Vol 10, Iss 1, pp 61-68
- .-**Lee-JW Oh-JH (1997)**; Hybrid Learning of Mapping and Its Jacobian in Multilayer Neural Networks; NEURAL COMPUTATION 1997, Vol 9, Iss 5, pp 937-958.
- .-**Lightbody-G Irwin-GW (1997)**; Nonlinear Control-Structures Based on Embedded Neural System Models; IEEE TRANSACTIONS ON NEURAL NETWORKS 1997, Vol 8, Iss 3, pp 553-567.
- .-**Magoulas-GD Vrahatis-MN Androulakis-GS (1997)**; Effective Backpropagation Training with Variable Stepsize; NEURAL NETWORKS 1997, Vol 10, Iss 1, pp 69-82.
- .- **Pacheco Bonrostro J.A. y Delgado Serna C.R. (1997)**; Modificación del aprendizaje de la red de propagación hacia atrás; XI Reunión ASEPELT Bilabao (1997); CD Actas
- .-**Stager-F Agarwal-M (1997)**; Methods to Speed-Up the Training of Feedforward and Feedback Perceptrons; NEURAL NETWORKS 1997, Vol 10, Iss 8, pp 1435-1443.
- .-**Verma-B**; Fast Training of Multilayer Perceptrons; IEEE TRANSACTIONS ON NEURAL NETWORKS 1997, Vol 8, Iss 6, pp 1314-1320.
- .-**Vitela-JE Reifman-J (1997)**; Premature Saturation in Backpropagation Networks - Mechanism and Necessary Conditions; NEURAL NETWORKS 1997, Vol 10, Iss 4, pp 721-735.
- .-**West-AHL Saad-D (1997)**; Online Learning with Adaptive Backpropagation in 2-Layer Networks; PHYSICAL REVIEW E 1997, Vol 56, Iss 3, pp 3426-3445.
- .-**Yam-JYF Chow-TWS (1997)**; Extended Least-Squares Based Algorithm for Training Feedforward Networks; IEEE TRANSACTIONS ON NEURAL NETWORKS 1997, Vol 8, Iss 3, pp 806-810.
- .-**Yu-XH Chen-GA (1997)**; Efficient Backpropagation Learning Using Optimal Learning Rate and Momentum; NEURAL NETWORKS 1997, Vol 10, Iss 3, pp 517-527.