

HOJAS DE CÁLCULO PARA LA SIMULACIÓN DE REDES DE NEURONAS ARTIFICIALES (RNA)

J. GARCÍA¹, A. M. LÓPEZ^{2,6}
J. E. ROMERO³, A. R. GARCÍA⁴
C. CAMACHO², J. L. CANTERO⁵
M. ATIENZA⁵ y R. SALAS⁵

La utilización de Redes de Neuronas Artificiales (RNA) en problemas de predicción de series de tiempo, clasificación y reconocimiento de patrones ha aumentado considerablemente en los últimos años. Programas informáticos de matemáticas de propósito general tales como MATLAB, MATHCAD y aplicaciones estadísticas como SPSS y S-PLUS incorporan herramientas que permiten implementar RNAs. A esta oferta de software hay que añadir programas específicos como NeuralWare, EasyNN o Neuron. Desde un punto de vista educativo, el acceso de los estudiantes a estos programas puede ser difícil dado que no están pensadas como herramientas didácticas. Por otro lado, las hojas de cálculo como Excel y Gnumeric incorporan utilidades que permiten implementar RNAs y son de fácil acceso para los estudiantes. El objetivo de este trabajo es proporcionar un pequeño tutorial sobre la utilización de Excel para implementar una RNA que nos permita ajustar los valores de una serie de tiempo correspondiente a actividad cerebral alfa y que permita al alumno entender el funcionamiento de estos dispositivos de cálculo.

Spreadsmeet for the simulation of artificial neural networks (ANNs)

Palabras clave: Redes de neuronas artificiales, hoja de cálculo, aprendizaje supervisado

Clasificación AMS (MSC 2000): 97U70

¹ Departamento de Ingeniería del Diseño. Universidad de Sevilla

² Departamento de Psicología Experimental. Universidad de Sevilla

³ Departamento de Economía Aplicada I. Universidad de Sevilla

⁴ I.E.S. Los Viveros. Sevilla

⁵ Laboratorio de Sueño y Cognición

⁶ La correspondencia debe dirigirse a Ana María López Jiménez, Departamento de Psicología Experimental, Universidad de Sevilla, Avda. Camilo José Cela s/n. 41005 Sevilla, España. Telf: (34) 954557812. Fax: (34) 954551784. E-mail: analopez@us.es

–Recibido en marzo de 2001.

–Aceptado en diciembre de 2001.

1. INTRODUCCIÓN

El uso cada vez más frecuente de redes de neuronas artificiales (en adelante RNA) en distintos campos es la razón de que nos planteemos la necesidad de incluir esta temática en asignaturas optativas de análisis de datos, de estadística y/o en los programas de doctorado de nuestros departamentos. Sin pretender ser exhaustivos, pensamos que las razones de este incremento hay que buscarlas en que

- son procedimientos muy flexibles de optimización no lineal,
- son, o pueden ser, una alternativa no paramétrica a las técnicas de análisis multivariante, y
- el número de aplicaciones informáticas que permiten implementarlas es cada vez mayor.

No obstante en el contexto de aprendizaje en el que insertamos este trabajo nos encontramos con que tanto los programas informáticos de matemáticas de propósito general (MATLAB, MATHEMATICA, IMSL, etc.) como los estadísticos (SPSS, S-PLUS, SAS, etc.) que tienen implementados módulos de RNA, así como las aplicaciones específicas (EasyNN, Neuron, NeuroShell, NeuralWare, etc.), son, a veces, de difícil acceso para los alumnos y que, además, funcionan como una «caja negra» lo que no facilita la comprensión de las RNA. Este trabajo pretende solventar, en parte, esos problemas mostrando la implementación de una RNA en una aplicación informática muy difundida como es Excel. El uso de hojas de cálculo va a permitir al alumno entender mejor el funcionamiento de una RNA y le va a facilitar el acceso a una metodología cada vez más extendida.

En los apartados que siguen desarrollaremos los conceptos básicos de las RNA y un pequeño tutorial para mostrar la implementación de una RNA en Excel.

2. REDES DE NEURONAS ARTIFICIALES (RNA)

Las RNA son dispositivos de cálculo inspirados en las redes de neuronas biológicas. Como estas últimas, están constituidas por elementos simples denominados *nodos* o *neuronas* organizados en capas y altamente interconectados. A la forma particular de organizarse y conectarse las neuronas se le denomina *arquitectura* o *topología de red*. A cada conexión se le asigna un peso numérico que va a constituir el principal recurso de memoria a largo plazo de las RNA. El aprendizaje se realiza, usualmente, con la actualización de los pesos mediante una determinada regla de aprendizaje.

Vamos a describir, brevemente, las características más importantes de las RNA. Para mayor información sobre RNA pueden consultarse: Bishop, 1995; Cheng y Tittering-

ton, 1994; Hilera y Martínez, 1995; Kohonen, 1989; Michie, Spiegelhalter y Taylor, 1994; Ripley, 1996; Smith, 1993; van der Smagt, 1994.

2.1. Neuronas artificiales: funcionamiento y tipos

Una neurona es la unidad básica de procesamiento. Recibe y emite información de otras neuronas y del/hacia el mundo exterior. En cada neurona se realiza un cálculo local y sencillo con las entradas que le proporcionan sus vecinas sin que sea necesario un control global en el conjunto de unidades. La neurona artificial procesa la información que le llega mediante la obtención de dos componentes. El primero es un componente lineal denominado *función de entrada* (in_i), que calcula la suma ponderada de los valores de entrada a la neurona ($in_i = \sum w_{ij} a_j$). El segundo es un componente, generalmente, no lineal conocido como *función de activación* o *función de transferencia*, f , que transforma la suma ponderada en el valor de salida de la neurona ($a_i = f(in_i)$). La Figura 1 es un esquema del funcionamiento de una neurona artificial. En dicho esquema a_j representan las entradas a la neurona y w_{ij} los pesos de cada conexión.

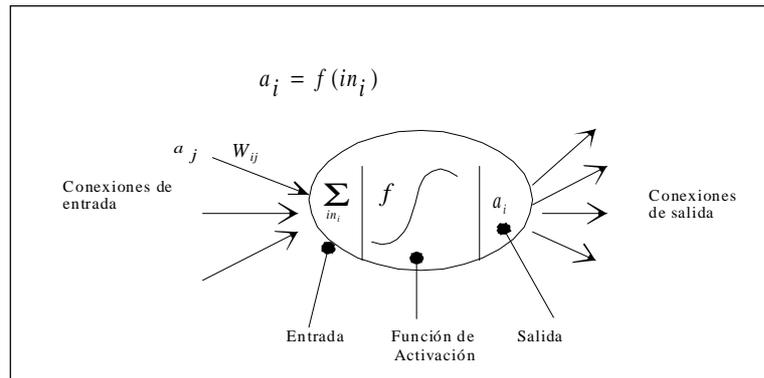


Figura 1. Esquema de una neurona artificial.

La utilización de diferentes funciones matemáticas para f da lugar a distintos tipos de neuronas. Cuatro de las funciones de activación más comunes son: escalón, signo, sigmoidal o logística y lineal.

La *función escalón* o *umbral* tiene un límite, u , de manera que produce un 1 cuando la entrada es mayor que u y en el caso contrario produce 0. La *función signo* produce +1 si la entrada es mayor que u y -1 en caso contrario. La *función lineal* o *identidad* deja invariante la entrada. La *función sigmoidal* o *logística* dada por

$$(1) \quad f(in_i) = \frac{1}{1 + e^{-in_i}}$$

es la función de activación más usada en las aplicaciones estadísticas de las RNA.

2.2. Arquitecturas de Red

Para caracterizar a una RNA, además del funcionamiento de los elementos que la constituyen, es necesario describir como se organizan y conectan dichos elementos.

Las neuronas se suelen organizar en columnas paralelas denominadas *capas*. Es útil distinguir entre tres tipos de capas: entrada, salida y ocultas. Las neuronas de la *capa de entrada* reciben señales del entorno y no realizan más operación que la de distribuir estas señales a las neuronas de la capa siguiente. Las neuronas de la *capa de salida* procesan la información y envían la señal fuera de la red. Las *unidades ocultas* son aquellas cuyas entradas y salidas se encuentran dentro de la red. Las neuronas de capas distintas pueden estar conectadas unidireccionalmente o bidireccionalmente. Las redes que sólo contienen relaciones unidireccionales se denominan *estáticas* o *redes acíclicas*. Las redes con relaciones bidireccionales se denominan *redes dinámicas* o *cíclicas*. En las redes estáticas la información fluye en un solo sentido (ni lateralmente ni hacia atrás). Entre las redes dinámicas se pueden distinguir las que contienen relaciones bidireccionales entre los nodos de la misma capa, denominadas *redes competitivas*, y redes con relaciones bidireccionales entre neuronas de distintas capas, denominadas *redes recurrentes*. A su vez, las neuronas de distintas capas pueden estar total o parcialmente conectadas. A las redes con conexiones unidireccionales y totalmente conectadas se les denomina *perceptrones multicapa* o *redes feedforward* y son, junto a las redes competitivas, las más utilizadas en análisis de datos. La Figura 2 muestra tres redes con distintos tipos de conexiones.

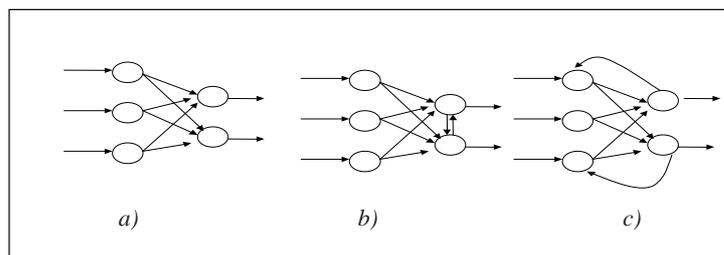


Figura 2. Tres arquitecturas de redes de neuronas artificiales: a) Red feedforward, b) red competitiva y c) red recurrente.

2.3. Aprendizaje de una RNA: tipos y reglas

La función que realiza una RNA depende de la función de activación y de las conexiones entre nodos. Al procedimiento mediante el que se modifica el valor de los pesos de conexión para obtener la salida deseada se le denomina *aprendizaje* o *entrenamiento*. Desde un punto de vista muy global se distinguen dos tipos de aprendizaje: *supervisado* y *no supervisado*. El aprendizaje supervisado (o aprendizaje con maestro) trata de conseguir que la red sea capaz de predecir, a partir de un conjunto de características suministradas como entradas, el valor que tomarán otras características, llamadas *objetivo*, habiendo sido observadas ambos tipos de características en cada uno de los patrones de entrenamiento. El aprendizaje se lleva a cabo modificando los pesos de la red según alguna regla que trata de minimizar la discrepancia entre la salida de la red y la salida correcta proporcionada por los objetivos. El proceso de aprendizaje comienza una vez definida la arquitectura de la red. A los datos utilizados en el proceso de aprendizaje se les denomina *conjunto de entrenamiento*. Durante el aprendizaje se procesan varias veces los pares de vectores del conjunto de entrenamiento y en cada paso del aprendizaje se modifican los pesos de las conexiones. Terminado el proceso de aprendizaje, la red se evalúa sobre un conjunto de datos no utilizados en el entrenamiento y denominado *conjunto de validación*.

En el aprendizaje no supervisado existen características de entrada pero no objetivo (aprendizaje sin maestro). No existe información que indique si la salida que produce la red es o no correcta. Las técnicas de aprendizaje no supervisado se utilizan sobre todo para obtener estructuras, relaciones o clasificaciones (*cluster analysis*).

Existen muchos algoritmos de aprendizaje pero los más extendidos son la regla delta generalizada (*backpropagation*) para el aprendizaje supervisado y la regla estándar del aprendizaje competitivo en el caso no supervisado.

3. USANDO EXCEL PARA IMPLEMENTAR UNA RNA

3.1. Planteamiento del problema

Las posibilidades de aplicación de las RNA a problemas de investigación abordados tradicionalmente desde las técnicas estadísticas clásicas son múltiples. El problema que pretendemos abordar, en este trabajo, con la implementación de una RNA en Excel es el ajuste a una serie de tiempo empírica.

La serie seleccionada corresponde a ritmo cerebral alfa (7.5-12.8 Hz) extraída del EGG de un estudiante de psicología. Se utilizó una frecuencia de muestreo de 200 Hz (período de muestreo de 5 milisegundos) y la señal original ($N = 1024$) se multiplicó por 20000 (ganancia) para que fuera observada sin problemas.

3.2. Topología de la RNA utilizada

Para ajustar los valores de la serie hemos utilizado un perceptron multicapa con tres capas (entrada-oculta-salida). La capa de entrada está constituida por tres neuronas correspondientes a los valores de la serie y sus retardos de lag 1 y 2. La capa oculta está constituida por dos neuronas. Cada neurona de esta capa recibe el vector $\mathbf{y} = (y_{t-2}, y_{t-1}, y_t)$ lo procesa y el resultado lo envía a la neurona que constituye la capa de salida. En la Figura 3 hemos representado la arquitectura de la red utilizada para el ajuste.

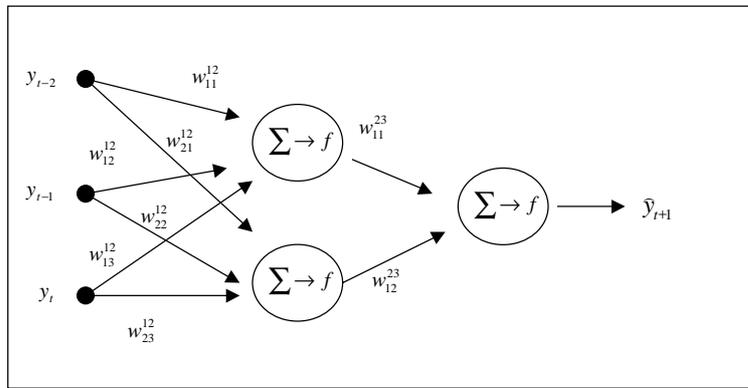


Figura 3. Perceptron multicapa.

El número de capas de la RNA utilizada es el mínimo compatible con la propiedad del aproximador universal.

Kolmogorov en 1957, demostró que toda función continua de varias variables definida sobre un compacto (cerrado y acotado) puede ser representada como la superposición de funciones de una variable. Este teorema, en términos de RNA, se traduce en que toda función $f: C \subseteq \mathbb{R}^p \rightarrow \mathbb{R}$, C compacto, puede ser representada exactamente por una RNA tipo perceptron multicapa con dos capas ocultas, la primera constituida por $p(2p + 1)$ nodos y la segunda por $2p + 1$ nodos. Posteriormente, se ha demostrado que los perceptrones multicapa con una sola capa oculta verifican interesantes propiedades teóricas, como la del aproximador universal, según la cual toda función $f: C \subseteq \mathbb{R}^p \rightarrow \mathbb{R}^q$, C compacto, puede ser aproximada uniformemente mediante un perceptron multinivel con una capa oculta, con función de activación logística en dicha capa y función de activación identidad en la capa de salida (ver Cybenko, 1989). Esta última propiedad, junto con su simplicidad, justifica que el perceptron multinivel con una sola capa oculta y función de transferencia logística sea el modelo utilizado para el desarrollo de nuestra red.

La composición de la capa de entrada obedece a resultados de estudios exploratorios previos acerca del alcance de la dependencia temporal en la serie.

3.3. Implementación de la RNA en Excel

En este apartado describiremos la secuencia de operaciones realizadas en Excel para implementar la RNA de la Figura 3. Partiremos de la Tabla 1 que contiene una porción de la serie empírica y el resultado de la secuencia de operaciones necesarias para resolver mediante la RNA el problema planteado.

Tabla 1. Una porción de Excel mostrando las operaciones necesarias para implementar la red de la Figura 3.

	A	B	C	D	E	F	G	H	I	J
1	ALFA	PREP	IN1A	IN2A	ON1A	ON2A	OR	DIF	DIFCUA	POSP
2	516		-0,0565855	-0,308024						
3	508		0,24836293	3,7925034						
4	478		0,02437906	-5,179935						
5	446		2,62826963	-2,61636						
6	416	-1,35								
7	400	-1,12								
8	386	-1,06	-0,2280374	1,6489059	0,4432364	0,8387431	-1,029509	-0,154135	0,0237576	387,10539
9	384	-0,88	-0,2209722	0,8622244	0,4449806	0,7031252	-0,670099	0,0441864	0,0019524	389,25279
10	384	-0,71	-0,1748784	0,7064098	0,4563915	0,6696074	-0,552414	-0,492653	0,2427065	386,2662
11	388	-0,06	-0,1293256	-2,129734	0,4677136	0,1062402	0,9513148	1,6221352	2,6313227	394,76742
12	384	-0,67	0,00922171	3,468178	0,5023054	0,9697686	-1,21707	0,4702534	0,2211382	383,02284
13	374	-1,69	-0,2043606	6,2145431	0,4490869	0,9980039	-1,430816	0,2087081	0,0435591	375,33778
14	356	-1,64	-0,4210798	2,3000779	0,3962584	0,9088835	-1,336492	0,1292379	0,0167024	359,91837
15	336	-1,47	-0,3474521	1,8942239	0,4140004	0,8692364	-1,18613	0,2798095	0,0782933	342,02795
16	304	-1,47	-0,306998	2,5396997	0,4238477	0,9268785	-1,311062	0,113659	0,0129184	308,94447
17	266	-1,42	-0,3158794	2,2718589	0,4216803	0,9065194	-1,263491	0,0856594	0,0073375	272,92573
18	228	-1,35	-0,3037879	2,0368013	0,4246318	0,8846072	-1,198404	0,1121967	0,0125881	235,82143
19	186	-1,31	-0,2864118	2,1110148	0,4288826	0,8919692	-1,206493	0,1068235	0,0114113	192,19591
20	140	-1,31	-0,2811796	2,24801	0,4301646	0,9044787	-1,235853	0,0162587	0,0002643	145,00177
21	102	-1,25	-0,2825435	1,9087969	0,4298303	0,8708839	-1,148836	0,0376325	0,0014162	108,79648
22	70	-1,19	-0,2655885	1,8017232	0,4339904	0,8583586	-1,105131	0,0711351	0,0050602	75,15527
23	40	-1,18	-0,2524996	1,9789778	0,4372084	0,8785722	-1,149559	0,1904805	0,0362828	41,53483
24	-6	-1,34	-0,2576729	2,8457877	0,4359358	0,9451005	-1,326966	0,0614478	0,0037758	-5,266328
25	-58	-1,39	-0,3001049	2,4721063	0,4255318	0,9221631	-1,294298	0,0414517	0,0017182	-52,06189
26	-110	-1,34	-0,301568	2,0662974	0,4251742	0,8875841	-1,204767	0,0230725	0,0005323	-100,4686
27	-152	-1,23	-0,2831202	1,7219572	0,429689	0,8483808	-1,090331	0,0567748	0,0032234	-141,3831
28	-184	-1,15	-0,257331	1,6967916	0,4360199	0,8451152	-1,065148	0,0645138	0,004162	-178,1413
29	-212	-1,13	-0,2429607	1,8793754	0,4395569	0,8675394	-1,114521	0,2100056	0,0441023	-211,0779
30	-258	-1,32	-0,2479472	2,9300539	0,4383288	0,9493123	-1,331696	0,1101061	0,0121233	-258,4049

En la primera fila de la Tabla 1, hemos escrito etiquetas representativas del contenido y de las operaciones realizadas en cada una de las columnas.

La columna A de la Tabla 1 contiene los valores de la serie (celdas A2 a A1025). Previo al ajuste con la RNA, la serie fue filtrada y estandarizada (Columna B: PREP). El filtrado obedece a la necesidad de eliminar ruido y destacar la posible señal; y, dado el recorrido de la función de activación logística empleada, es también necesario, para no perjudicar la capacidad de aprendizaje de la red, que los valores a estudiar no sean muy elevados por lo que estandarizamos la serie.

La columna B muestra los valores suavizados de la serie obtenidos mediante la expresión (2).

$$(2) \quad PREP = \frac{y_{t+5} - \frac{\sum_{i=1}^5 y_{t+i}}{5}}{S}$$

En (2), S es la desviación típica para los cinco valores promediados.

Para obtener los valores de la columna B colocamos el cursor en la celda B6 e introducimos la fórmula (2) con la siguiente sintaxis:

$$= (\$A6-PROMEDIO(\$A2:\$A6))/DESVEST(\$A2:\$A6)$$

Copiando esta fórmula en las celdas B7 a B1025 obtenemos los valores de la serie suavizada (Columna B de la Tabla 1).

Las celdas C2 a C5 y D2 a D5 (Fila 5 de la Tabla 1) contienen los valores de los pesos de la RNA de la Figura 3 tras el entrenamiento de la RNA. Concretamente, en las celdas C2 a C4 (Filas 2 a 4 de la Tabla 1), hemos definido los pesos de las conexiones de los nodos de la capa de entrada con el primer nodo (IN1A) de la capa oculta. Los valores de las celdas D2 a D4 (Filas 2 a 4 de la Tabla 1) corresponden a los pesos de las conexiones de los nodos de la capa de entrada con el segundo nodo (IN2A) de la capa oculta. Las celdas C5 y D5 contienen los pesos de conexión de los nodos de la capa oculta (IN1A e IN2A) con el nodo de la capa de salida. Los pesos iniciales de la red pueden fijarse en cualquier valor. En nuestro caso se fijaron en 0.5.

En las celdas C8 a C1024 y D8 a D1024 hemos calculado el input total a las neuronas de la capa oculta. El input total (in_i) es la suma ponderada de los valores de la serie. En nuestro ejemplo hemos utilizado la serie suavizada. Para calcular la entrada a la primera neurona de la capa oculta (Columna C de la Tabla 1: IN1A), nos situamos en la celda C8 y escribimos la siguiente expresión:

$$(3) \quad = \$C\$2*\$B6+\$C\$3*\$B7+\$C\$4*\$B8$$

Copiando (3) en las celdas C9 a C1024 obtenemos los valores de la Columna C correspondientes a la entrada neta a la neurona 1 (IN1A) para el conjunto de patrones de entrenamiento.

De manera análoga, para calcular la entrada neta a la segunda neurona de la capa oculta (IN2A), nos situamos en la celda D8 y escribimos

$$(4) \quad = \$D\$2*\$B6+\$D\$3*\$B7+\$D\$4*\$B8$$

Copiando (4) en las celdas D9 a D1024, obtenemos los valores de la columna D de la Tabla 1.

Las salidas de las neuronas IN1A e IN2A se obtienen aplicando la función de activación seleccionada a los valores de las celdas C8 a C1024 y D8 a D1024. En nuestro ejemplo hemos utilizado la función logística (1) por las razones comentadas en el apartado 3b.

Los valores de las columnas E y F, etiquetadas en la Tabla 1 como ON1A y ON2A respectivamente, corresponden a las salidas de las neuronas de la capa oculta para el conjunto de patrones de entrenamiento. Para obtener los valores de la columna E nos situamos en la celda E8 e insertamos la fórmula:

$$(5) \quad = 1 / (1 + \text{EXP}(-\$C8))$$

Copiando (5) en las celdas E9 a E1024 se obtienen el resto de los valores.

Para los valores de la columna F se procede de la misma manera. Situándonos en la celda F8 escribimos la fórmula

$$(6) \quad = 1 / (1 + \text{EXP}(-\$D8))$$

y copiamos (6) en las celdas F9 a F1024.

La columna G (OR) de la Tabla 1 contiene la salida de la red. Este valor es la suma ponderada de las entradas al nodo de la capa de salida. Para obtener los valores de la columna G multiplicamos las columnas E y F de la Tabla 1 por los pesos de las celdas C5 y D5. Situándonos en la celda G8 escribimos la expresión:

$$(7) \quad = \$C\$5*\$E8+\$D\$5*\$F8$$

Copiando (7) en las celdas G9 a G1024 obtenemos la salida de la RNA para el conjunto de patrones de entrenamiento. El nodo de la capa de salida, en esta red, sólo calcula la suma ponderada de las entradas, y el valor resultante es la salida de la red (OR).

Los valores de la columna H (DIF) en la Tabla 1 corresponden al error de predicción esto es, a la diferencia entre la serie empírica suavizada (Columna B: PREP) y la salida

de la RNA (Columna G: OR). Para calcular dichos errores, situándonos en la celda H8, restamos las columnas G (OR) y B(PREP) utilizando la expresión:

$$(8) \quad = \$G8-\$B9$$

Copiando (8) en las celdas G9 a G1024 se obtienen el resto de los valores de la columna H.

En la columna I (DIFCUA) hemos calculado el cuadrado de los errores de predicción. Para ello en la celda I8 hemos escrito la siguiente fórmula:

$$(9) \quad = \$H8*\$H8$$

Copiando (9) en las celdas I9 a I1024 se obtienen los cuadrados de los errores de predicción para el conjunto de patrones de entrenamiento.

En la celda J1025 hemos calculado la suma de cuadrados de los errores ($\sum e^2$). La fórmula para calcular esta suma es:

$$(10) \quad \text{SUMA}(\$I8:\$I1024).$$

Los valores de los pesos de la red se modificarán en el sentido de hacer mínimo el valor de la celda J1025.

3.4. Cálculo de los valores óptimos de los pesos de la RNA

El problema de aprendizaje de la RNA se formula en términos de la minimización de la función de error (10). Este error depende de los parámetros adaptativos de la red. El proceso de aprendizaje se lleva a cabo modificando los pesos de manera que la suma de cuadrados de los errores sea mínima para el conjunto de patrones de entrenamiento. Para resolver este problema Excel dispone de una macro denominada «Solver» que permite obtener el valor máximo o mínimo de una celda modificando los valores de otras celdas.

Para ejecutar «Solver» seleccionamos, en el menú «Herramientas», dicha opción. Cliqueando sobre «Solver» abrimos el cuadro de diálogo: «Parámetros de Solver». En este cuadro, insertamos la celda objetivo ($\$J\1025) y el rango de celdas a modificar ($\$C\$2:\$D\5). Podemos elegir, también, el método de resolución para el problema de mínimos planteado. Cliqueando en «Opciones» del cuadro de diálogo «Parámetros de Solver» abrimos el cuadro «Opciones de Solver» en este cuadro podemos modificar las opciones por defecto de los siguiente parámetros:

Tiempo: Esta opción establece un límite temporal para encontrar la solución al problema. En nuestro ejemplo hemos elegido la opción por defecto: 100 segundos.

Iteraciones: Esta opción establece un límite al número de cálculos provisionales. Este valor es el número de ciclos de aprendizaje (epochs) de la red. El valor por defecto es 100.

Precisión: Esta opción controla la precisión de la solución. La precisión debe indicarse como una fracción entre 0 y 1.

Tolerancia: Es el porcentaje máximo de discrepancia entre el valor de la celda objetivo y las restricciones externas si las hubiese. Valores grandes de tolerancia aceleran el proceso de solución. Nosotros hemos utilizados el valor por defecto: 50.

Convergencia: Si el valor del cambio relativo en la celda objetivo es más pequeño que el valor introducido en esta opción en las cinco últimas iteraciones «Solver» se detendrá. Esta opción sólo se aplica a problemas no lineales y debe indicarse mediante una fracción entre 0 y 1.

Además de las opciones anteriores, hemos seleccionado el método de Newton para la aproximación de raíces, derivadas progresivas y estimación cuadrática. La selección de este conjunto de opciones obedece a que suelen requerir menos tiempo de cálculo. No obstante, en el caso de la selección del método de aproximación de raíces, la elección del método de Newton o del método del gradiente conjugado no es crucial ya que «Solver» es capaz de cambiar automáticamente entre ambos métodos en función de las necesidades de almacenamiento que requiera el problema que se esté resolviendo.

Una vez que hemos dado valores a las opciones anteriores pulsamos en «Aceptar» y volvemos al cuadro «Parámetros de Solver». En este cuadro, pulsamos en «Resolver» y obtenemos la solución al problema planteado.

Los valores finales de los pesos después de 100 iteraciones son los correspondientes a las celdas C2 a C5 y D2 a D5 de la Tabla 1.

Finalmente, en la columna J de la Tabla 1, efectuamos la operación inversa a la realizada para obtener los valores de la serie suavizada (columna B:PREP). Para realizar esta operación, nos situamos en la celda J8 y escribimos la siguiente expresión:

$$(11) \quad = \$G8*DESVEST(\$A4:\$A8)+PROMEDIO(\$A4:\$A8)$$

La expresión (11) la copiamos en las celdas J9 a J1024 y obtenemos los valores de la columna J.

3.5. Valoración del ajuste obtenido

Para evaluar el grado de ajuste de la red hemos realizado un gráfico de líneas utilizando el asistente para gráficos de Excel. Hemos representado la serie de la columna A (ALPHA) para el rango de valores A9 a A1025 y la serie de la columna J (POSP) para el

rango de valores de J8 a J1024 (ver Figura 4). Como puede observarse en la Figura 4, las dos series (ALFA y POSP) están superpuestas por lo que el ajuste es prácticamente perfecto.

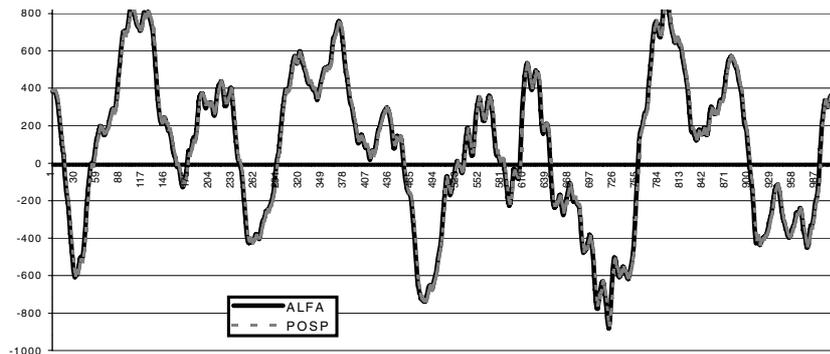


Figura 4. Gráfico de líneas para evaluar el ajuste a la serie empírica.

Hemos calculado también el coeficiente de correlación de Pearson entre las dos series obteniendo un valor de 0.99644869. Para calcular este coeficiente hemos utilizado la siguiente expresión

$$(12) \quad = \text{COEF.DE.CORREL}(A9:A1025;J8:J1024)$$

3.6. Alternativas

En este trabajo hemos utilizado todos los valores de la serie para entrenar a la RNA. Otra posibilidad, de cara a validar la ejecución de la RNA y a compararla con otros procedimientos de ajuste y predicción de series de tiempo, consiste en utilizar parte de la serie, por ejemplo el rango B8 a B1000 para entrenar a la RNA y los valores restantes utilizarlos como conjunto de validación. La implementación es la expuesta en este trabajo salvo que, una vez terminado el proceso de adaptación de los pesos, las fórmulas correspondientes a las distintas columna de la Tabla 1 hay que copiarlas en las celdas del conjunto de validación.

Para el ajuste y predicción de series de tiempo pueden utilizarse técnicas estadísticas clásicas como modelos ARIMA, análisis espectral, métodos exponenciales, etc. No es el objetivo de este trabajo comparar la ejecución de estos métodos con la ejecución de la metodología basada en redes de neuronas.

4. CONCLUSIONES

En este trabajo hemos pretendido mostrar las posibilidades didácticas del uso de Excel para implementar redes de neuronas artificiales. Hemos elegido un problema de ajuste a una serie empírica; pero, de la misma manera, pueden implementarse redes para resolver problemas de reducción de datos (análisis de componentes principales), clasificación y discriminación. La utilización de hojas de cálculo como herramientas didácticas para la implementación de RNA tiene, frente a programas específicos, una serie de ventajas que podríamos resumir en: 1º permiten el acceso a una metodología cada vez más utilizada, las RNA, a través de aplicaciones de uso frecuente y de fácil acceso (hojas de cálculo) y 2º aumentan las posibilidades de aprender y entender estos dispositivos de cálculo.

5. REFERENCIAS

- Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford: Oxford University Press.
- Chen, B. y Titterton, D. M. (1994). «Neural networks: a review from a statistical perspective». *Statistical Science*, 9, 1, 2-59.
- Cybenko, G. (1989). «Approximation by superposition of a sigmoidal function». *Mathematics of Control, Signal and Systems*, 2, 303-314.
- Hilera, J. R. y Martínez, V. J. (1995). *Redes Neuronales Artificiales. Fundamentos, Modelos y Aplicaciones*. Madrid: Ra-Ma.
- Kohonen, T. (1989). *Self-organization and associative memory*. Berlín: Springer-Verlag.
- Michie, D., Spiegelhalter, D. J. y Taylor, C. C. (1994). *Machine Learning, Neural and Statistical Classification*. Disponible en: <http://www.amsta.leeds.ac.uk/charles/statlog/>
- Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press.
- Smith, M. (1995). *Neural networks for statistical modeling*. New York: Van Nostrand Reinhold.
- Van der Smagt, P. P. (1994). «Minimization methods for training feedforward neural networks». *Neural Networks*, 7(1), 1-11.

Los archivos de este trabajo están disponibles en analopez@us.es

Queremos agradecer al profesor Antoni Solanas sus acertados comentarios a este manuscrito.

ENGLISH SUMMARY

SPREADSHEET FOR THE SIMULATION OF ARTIFICIAL NEURAL NETWORKS (ANNs)

J. GARCÍA¹, A. M. LÓPEZ^{2,6}
J. E. ROMERO³, A. R. GARCÍA⁴
C. CAMACHO², J. L. CANTERO⁵
M. ATIENZA⁵ and R. SALAS⁵

In recent years, the use of Artificial Neural Networks or ANNs has increased considerably to solve prediction problems in time series, classification and recognition of patterns. General-purpose mathematical programs such as MATLAB, MATHCAD and mathematical and statistical programs such as SPSS and S-PLUS incorporate tools that allow the implementation of ANNs. In addition to these, specific programs such as NeuralWare, EasyNN, or Neuron, complete the software offer using ANNs.

From an educational point of view, an aspect that concerns the authors of this work, student access to these programs can be expensive or, in some case, inadvisable given the few possibilities they provide as didactic instruments. These programs are usually easy to use but do not facilitate the understanding of the technique used. On the other hand, spreadsheets like Excel or Gnumeric incorporate tools that allow all of the necessary calculations to implement an ANN. These programs are user-friendly to the point that they are used by university laboratories, as well as psychology, economic science, and engineering students, to mention a few. This paper provides a small tutorial on the use of a spreadsheet, specifically Excel, to implement an ANN to adjust the values of a time series corresponding to cerebral alpha activity.

Keywords: Artificial neural network, spreadsheet, supervised learning

AMS Classification (MSC 2000): 97U70

¹ Departamento de Ingeniería del Diseño. Universidad de Sevilla

² Departamento de Psicología Experimental. Universidad de Sevilla

³ Departamento de Economía Aplicada I. Universidad de Sevilla

⁴ I.E.S. Los Viveros. Sevilla

⁵ Laboratorio de Sueño y Cognición

⁶ La correspondencia debe dirigirse a Ana María López Jiménez, Departamento de Psicología Experimental, Universidad de Sevilla, Avda. Camilo José Cela s/n. 41005 Sevilla, España. Telf: (34) 954557812. Fax: (34) 954551784. E-mail: analopez@us.es

–Received March 2001.

–Accepted December 2001.

In recent years, the use of Artificial Neural Networks or ANNs has increased considerably in order to solve prediction problems in time series, classification and recognition of patterns. We believe that this increase is due to the difficulty of solving this type of problem when the behavior of the series is nonlinear or when the classes are not lineally separable. Another reason is the access to numerous relatively user-friendly software applications allowing the implementation of frequent use networks.

General-purpose mathematical programs such as MATLAB, MATHCAD and mathematical and statistical programs such as SPSS and S-PLUS incorporate tools that allow the implementation of ANNs. In addition to these, specific programs such as NeuralWare, EasyNN, or Neuron, complete the software offer using ANNs.

From an educational point of view —an aspect that concerns the authors of this work— student access to these programs can be expensive or, in some case, inadvisable given the few possibilities they provide as teaching instruments. These programs are usually easy to use but do not facilitate the understanding of the technique. On the other hand, spreadsheets like Excel, or Gnumeric incorporate tools that allow all of the necessary calculations to implement an ANN. These programs are user-friendly to the point that they are used by University laboratories, as well as psychology, economic science, and engineering students, to mention a only few.

In the first part of this work, we introduce the basics aspects of ANNs: elements, architectures, and types and rules of learning. In the second part, we describe the necessary operations to implement in Excel an ANN to adjust the values of a time series corresponding to cerebral alpha activity.

The ANN to implement consists of three layers. The input layer is composed of three nodes corresponding to the values of the series (y_{t-2}, y_{t-1}, y_t) . The hidden layer is composed of two nodes, each of which applies the logistic function as the activation function and, finally, the output layer with only one node and linear activation function.

To perform the task, the ANN modifies the synaptic weights (w_{ij}) adjust so that the value of the function

$$\sum (y_{t+1} - \hat{y}_{t+1})^2$$

is minimal.

Table 1 denotes a portion of Excel to implement the net.

The column A of Table 1 contains the values of the series (cells A2 to A1025) and the column B shows the softened values of the series.

Columns C and D of Table 1 contain the values of the weights of the net to implement. Specifically in cells C2 to C4, we have defined the weights of the connections of the nodes of the input layer with the first node (IN1A) of the hidden layer. The values of

cells D2 to D4 correspond to the weights of the connections of the nodes of the input layer with the second node (IN2A) of the hidden layer. Cells C5 and D5 contain the values of the weights of the connection of node IN1A and IN2A of the hidden layer with the neuron of the output layer. Initially we can fix the weights at any value. In cells C8 to C1024 and D8 to D1024 we have calculated the total input to nodes 1 and 2 of the hidden layer. The total input is the weight sums of the values of the series.

Table 1. A portion of Excel.

	A	B	C	D	E	F	G	H	I	J
1	ALFA	PREP	IN1A	IN2A	ON1A	ON2A	OR	DIF	DIFCUA	POSP
2	516		-0,0565855	-0,308024						
3	508		0,24836293	3,7925034						
4	478		0,02437906	-5,179935						
5	446		2,62826963	-2,61636						
6	416	-1,35								
7	400	-1,12								
8	386	-1,06	-0,2280374	1,6489059	0,4432364	0,8387431	-1,029509	-0,154135	0,0237576	387,10539
9	384	-0,88	-0,2209722	0,8622244	0,4449806	0,7031252	-0,670099	0,0441864	0,0019524	389,25279
10	384	-0,71	-0,1748784	0,7064098	0,4563915	0,6696074	-0,552414	-0,492653	0,2427065	386,2662
11	388	-0,06	-0,1293256	-2,129734	0,4677136	0,1062402	0,9513148	1,6221352	2,6313227	394,76742
12	384	-0,67	0,00922171	3,468178	0,5023054	0,9697686	-1,21707	0,4702534	0,2211382	383,02284
13	374	-1,69	-0,2043606	6,2145431	0,4490869	0,9980039	-1,430816	0,2087081	0,0435591	375,33778
14	356	-1,64	-0,4210798	2,3000779	0,3962584	0,9088835	-1,336492	0,1292379	0,0167024	359,91837
15	336	-1,47	-0,3474521	1,8942239	0,4140004	0,8692364	-1,18613	0,2798095	0,0782933	342,02795
16	304	-1,47	-0,306998	2,5396997	0,4238477	0,9268785	-1,311062	0,113659	0,0129184	308,94447
17	266	-1,42	-0,3158794	2,2718589	0,4216803	0,9065194	-1,263491	0,0856594	0,0073375	272,92573
18	228	-1,35	-0,3037879	2,0368013	0,4246318	0,8846072	-1,198404	0,1121967	0,0125881	235,82143
19	186	-1,31	-0,2864118	2,1110148	0,4288826	0,8919692	-1,206493	0,1068235	0,0114113	192,19591
20	140	-1,31	-0,2811796	2,24801	0,4301646	0,9044787	-1,235853	0,0162587	0,0002643	145,00177
21	102	-1,25	-0,2825435	1,9087969	0,4298303	0,8708839	-1,148836	0,0376325	0,0014162	108,79648
22	70	-1,19	-0,2655885	1,8017232	0,4339904	0,8583586	-1,105131	0,0711351	0,0050602	75,15527
23	40	-1,18	-0,2524996	1,9789778	0,4372084	0,8785722	-1,149559	0,1904805	0,0362828	41,53483
24	-6	-1,34	-0,2576729	2,8457877	0,4359358	0,9451005	-1,326966	0,0614478	0,0037758	-5,266328
25	-58	-1,39	-0,3001049	2,4721063	0,4255318	0,9221631	-1,294298	0,0414517	0,0017182	-52,06189
26	-110	-1,34	-0,301568	2,0662974	0,4251742	0,8875841	-1,204767	0,0230725	0,0005323	-100,4686
27	-152	-1,23	-0,2831202	1,7219572	0,429689	0,8483808	-1,090331	0,0567748	0,0032234	-141,3831
28	-184	-1,15	-0,257331	1,6967916	0,4360199	0,8451152	-1,065148	0,0645138	0,004162	-178,1413
29	-212	-1,13	-0,2429607	1,8793754	0,4395569	0,8675394	-1,114521	0,2100056	0,0441023	-211,0779
30	-258	-1,32	-0,2479472	2,9300539	0,4383288	0,9493123	-1,331696	0,1101061	0,0121233	-258,4049

The outputs of neurons IN1A and IN2A are obtained by applying the activation function selected for the values of columns C and D. Columns E and F are labeled as ON1A and ON2A and correspond to neuron outputs of the hidden layer.

The column G of Table 1 contains the network output. It corresponds to the weighted sum from the inputs to the node of the output layer. To obtain the values of column G, the values of columns E and F are multiplied times the weights of cells C5 and D5. The output layer of the node only calculates the weighted sum of the input and this is the output of the net (OR).

The values of column H in Table 1 correspond to prediction errors (DIF). To calculate these errors we subtract columns G (OR) and B (PREP).

The column I contains the errors to the square (DIFCUA). They are obtained by multiplying the values of column H with the same values. Cell J1025 has calculated the sum of squares of the errors. The formula to calculate this sum is: SUMA (\$I8:\$I1024). The values of the network weights will be modified in the sense of making the value of this cell minimum.

To solve this problem Excel has a macro denominated: «Solver» that allows the maximum value or minimum of a cell modifying other cells to be determined. To execute this macro in the Tools menu one must click on «Solver». Clicking on «Solver» opens the dialog box: «Parameters to Solver2». In this box, the cell objective (\$J\$1025) must necessarily be fixed and the range of the cells they modify (\$C\$2:\$D\$5). We can also choose the desired resolution method to solve the problem. By clicking on «Options» of the dialog box «Parameters to Solver» we can choose the values to the next options: Maximum time, Iterations, Precision, Tolerance, Convergence. In addition to the previous options, we can choose the method for the approach of roots.

Finally, in column J of Table 1, we have defined the inverse operation of the column B and to evaluate the adjustment of the implemented net, we have carried out a line graph with the Assistant of Excel graphics. We have represented the series of the column A (ALPHA) and the series of the column J (POSP).