UTILIZACIÓN DE REDES NEURONALES EN LA PREDICCIÓN ECONÓMICA

GARCIA MIRANTES, ANDRÉS

andres_g@arrakis.es

SUAREZ RIESTRA, Mª CRISTINA

csr@etsiig.uniovi.es

Departamento de Matemáticas. Universidad de Oviedo

Palabras clave: Series temporales financieras, predicción no lineal, redes neuronales

Resumen

La metodología usual de predicción económica consiste esencialmente en modelos lineales. En esta categoría se incluyen tanto los modelos ARMA habituales como generalizaciones multivariantes del tipo espacio de estado y modelos VARMA. Es bien sabido que el ajuste de modelos lineales a datos reales no es siempre es lo más adecuado. De hecho básicamente las predicciones elaboradas mediante estas parametrizaciones consisten en combinaciones lineales de las realizaciones pasadas de las series en cuestión.

Este problema sugiere utilizar nuevas estrategias de modelización que permitan describir dinámicas no lineales. Para ello, en este trabajo se ha elegido la metodología de redes neuronales, ya que permiten simular casi cualquier tipo de función de predicción, lineal o no, habiéndose mostrado su utilidad empírica en diferentes entornos de aplicación pero cuya utilización en el ámbito económico ha sido escasa.

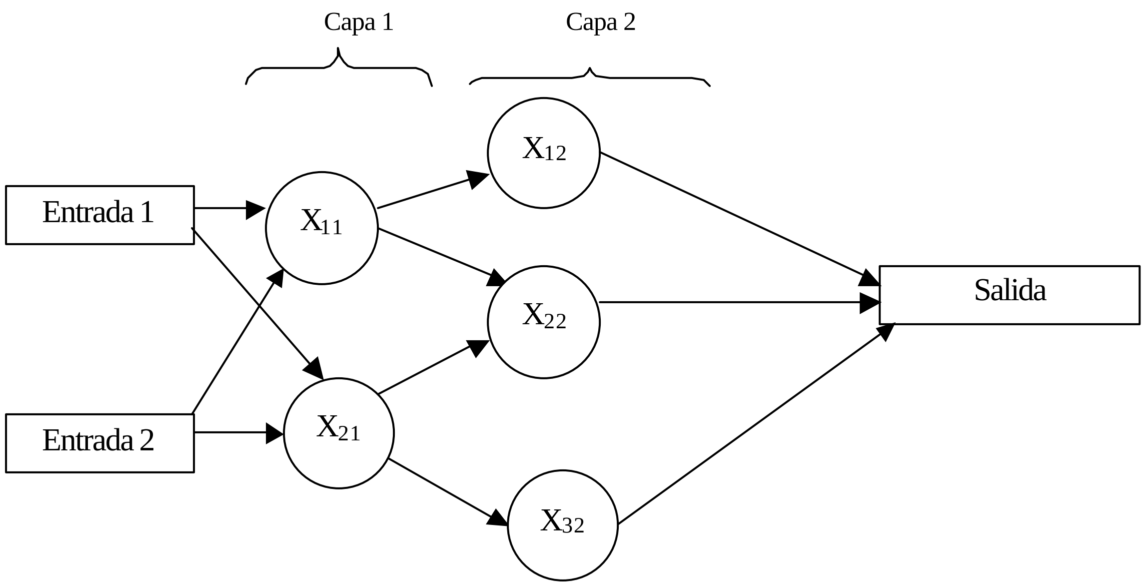
Se presenta un caso empírico de estudio, que emplea la metodología de redes neuronales aplicada a series económicas con muy diferentes problemáticas en su historia para la realización de predicciones. El trabajo se centra en series financieras donde la metodología clásica no se ha demostrado muy eficaz. Las predicciones se evalúan tanto intramuestralmente como extramuestralmente.

1. Estructuras de las redes neuronales

1.1 Conceptos generales

Una red neuronal es un esquema semiparamétrico (esto es, está determinado sólo parcialmente por los parámetros: su especificación completa requiere una estructura) que se usa para una múltiple variedad de problemas, desde predicción de series temporales hasta reconocimiento de notas musicales, pasando por identificación no lineal de sistemas.

Más formalmente, una red neuronal feedforward de L capas es un árbol finito orientado de L capas. Al conjunto de nodos (llamados neuronas) de cada capa se les denota $\{x_{ij}\}_{i=1\ldots N_j}$ donde N_j es el número de neuronas de la capa j. Nos referiremos con las mismas letras a la neurona y a su valor. El valor de una neurona de una capa viene dado por una función (en general no lineal) de una combinación lineal (o afín) de los valores de las neuronas da las capas anteriores. De un modo esquemático:



$$x_{ij} = f_j\left(\sum_{k=1}^{N_{j-1}} w_{ik}^{j} x_{kj-1} + b_j\right)$$

Fig 1. Una red neuronal con dos capas ocultas, dos entradas y una salida

Los valores de las neuronas se van computando secuencialmente mediante la regla anterior, hasta que finalmente se llega a la salida. Las funciones reales de variable real f_j (llamadas **funciones de transferencia**) están fijadas a priori en tanto que los elementos w_{ik}^{j} y b_j son parámetros a estimar. Respecto a las funciones, algunas de las más usuales son:

- Lineal: $f(x) = x$
- Logística: $f(x) = \dfrac{1}{1+e^{x}}$
- Tangente hiperbólica: $f(x) = \dfrac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$

Por supuesto, este tipo de funciones serán en general distintas entre capas. Lo habitual es que estas funciones sean lineales en la primera capa y logísticas o tangentes hiperbólicas en las siguientes.

Es posible demostrar que, mediante una red neuronal con una capa oculta en la cual la función transferencia de la capa oculta a la salida es lineal y la de transferencia de la entrada a la capa oculta es continua y sigmoidal (i.e. acotada tal que sus límites en $+\infty$ y $-\infty$ no coinciden) cualquier función integrable Lebesgue (en particular cualquier función continua) se puede aproximar por redes neuronales.

1.2 Modelo teórico para series temporales

La flexibilidad de las redes neuronales para simular y aproximar cualquier función nos permite postular un modelo general que incluye a los modelos AR como casos especiales. Así asumimos que la serie temporal y_t sigue un proceso estocástico del tipo:

$$y_t = h(y_{t-1}, \ldots, y_{t-k}) + \xi_t \tag{1}$$

Donde h es una función (lineal o no) desconocida y ξ_t es ruido blanco tal que $E[\xi_t / y_{t-1}, y_{t-2} \ldots] = 0$ (independiente en media del conjunto de información pasado). Este modelo resultará familiar a los habituados a trabajar con procesos AR, puesto que estos procesos son un caso especial donde h se restringe a ser lineal[1].

Es bastante común, si bien innecesario para la estimación, suponer que el ruido blanco es gaussiano. Esto suele simplificar los test de hipótesis posteriores sobre la bondad del ajuste (cuya distribución, en otro caso, es sólo cierta asintóticamente).

En el modelo dado por la ecuación (1), el método de estimación consistirá simplemente en minimizar una función de pérdida cuadrática. A diferencia de otros enfoques, en las redes neuronales es absolutamente innecesario especificar el tipo de no linealidad que aparece en la función h. Las decisiones que se deben tomar son las siguientes:

1. Especificar el número de retardos necesario (número de entradas). Es posible introducir retardos alternos que vengan sugeridos por la estacionalidad de la serie o ajustar una red estacional para separar los componentes.
2. Especificar la cantidad de capas, neuronas y estructura de la red. Un exceso de neuronas puede llevar a un sobreajuste ("overfitting") en tanto su defecto puede dejar estructura sin reflejar en nuestro modelo.
3. Es preciso, asimismo, escoger una función de transferencia, si bien cualquier función continua, acotada y sigmoidal tiene garantizada la aproximación de todo tipo de funciones. Tanto en los resultados que se presentan como en otros existentes se refleja que no hay grandes diferencias en el ajuste entre elegir un tipo u otro de función de transferencia.

1.2 Estimación (entrenamiento) de redes neuronales

Una vez que se ha escogido la estructura de red, los retardos y la función de transferencia es preciso estimar los pesos y sesgos que deben utilizarse para la transferencia entre capas. En este caso, se utilizará, como es usual en la literatura, una función de pérdida cuadrática.

[1] Este enfoque puede parecer restrictivo, pues no contempla la parte MA. Si bien no se va a incluir en este trabajo, es preciso resaltar que un tipo especial de redes neuronales (las redes recurrentes o de Elman) permiten que la red tenga un "estado" interno, con lo que el error de predicción (parte MA) puede ser utilizado. En este enfoque los ARIMA son casos especiales.

Se denotará por θ al vector de parámetros de la red y por $Y_t = (y_{t-1},..., y_{t-k})$ a la serie Se debe aproximar $h(Y_t)$ por $R(Y_t;\theta)$ siendo R la red neuronal escogida.

El problema, pues, se reduce al siguiente:

$$\min_{\theta} L(\theta) = \frac{1}{2}\left\{\sum_{t=k+1}^{N}[x_t - R(X_t;\theta)]^2\right\}$$

Esta situación de minimización no lineal no puede, por supuesto, ser resuelta analíticamente salvo casos triviales (por ejemplo, cuando las funciones de transferencia son lineales). Esto lleva a la necesidad de escoger métodos numéricos eficientes para la búsqueda de soluciones. Al proceso de minimización se le suele llamar "entrenamiento" en los trabajos especializados en redes.

Lo primero que se debe hacer notar es que los métodos numéricos más eficientes requieren evaluaciones de la función L y su gradiente (y en ocasiones su Hessiano). Como la forma analítica de L es muy compleja (al serlo R), evaluar estas cantidades directamente es computacionalmente muy costoso. Afortunadamente existen algoritmos que permiten un cálculo rápido de estas magnitudes, principalmente los siguientes:

- **Propagación hacia delante**: Se utiliza para evaluar L. Consiste simplemente en, comenzando por la capa más a la izquierda (la entrada) ir evaluando $x_{11,...}x_{N_1 1}$ lo que es computacionalmente simple. Una vez se tienen esas magnitudes, nos limitamos a calcular $x_{12,...}x_{N_2 2}$ y seguir, capa tras capa, hasta la salida aproximada. Una vez evaluada R, el cálculo de L es directo.

- **Propagación hacia atrás (retropropagación)**[2]: Se trata de un método de cálculo de gradiente de una red. Para ello se emplea la regla de la cadena desde la capa de salida a la de entrada. De un modo más preciso: Sea $s = R(Y_t)$. El gradiente de L es:

[2] Existe una cierta confusión en el uso de este término. En ocasiones se habla de "entrenar una red por propagación hacia atrás". La retropropagación no es más que un método de estimar el gradiente. Esa expresión se refiere simplemente a utilizar retropropagación para el cálculo del gradiente y minmizar por el método de descenso de máxima pendiente.

$$\nabla L(\theta) = \left[\sum_{t=k+1}^{N}(y_t - s)\right]\frac{ds}{d\theta} = \left[\sum_{t=k+1}^{N}(y_t - s)\right]\left\{\frac{\partial s}{\partial \theta} + \begin{pmatrix} \frac{\partial s}{\partial x_{1L-1}} & \frac{\partial s}{\partial x_{2L-1}} & \cdots & \frac{\partial s}{\partial x_{N_L L-1}} \end{pmatrix}\begin{pmatrix} \frac{dx_{1L-1}}{d\theta} \\ \frac{dx_{2L-1}}{d\theta} \\ \cdots \\ \frac{dx_{N_L L-1}}{d\theta} \end{pmatrix}\right\}$$

.

Recordando que $s = x_{1L} = f_L\left(\sum_{k=1}^{N_L} w_{1k}^{L} x_{kL-1} + b_L\right)$ (pues s es la única neurona de la última capa) y que las neuronas de capas anteriores no dependen de los parámetros de las capas siguientes, se pueden calcular las parciales en la última capa simplemente sustituyendo los valores de las neuronas (obtenidos mediante propagación hacia delante).

$$\frac{\partial s}{\partial b_L} = f_L'\left(\sum_{k=1}^{N_L} w_{1k}^{L} x_{kL-1} + b_L\right)$$

$$\frac{\partial s}{\partial x_{kL}} = f_L'\left(\sum_{k=1}^{N_L} w_{1k}^{L} x_{kL-1} + b_L\right) w_{1k}^{L}$$

$$\frac{\partial s}{\partial w_{1k}^{L}} = f_L'\left(\sum_{k=1}^{N_L} w_{1k}^{L} x_{kL-1} + b_L\right) x_{kL-1}$$

Esto permite ir hacia la capa anterior y calcular el gradiente de modo eficiente.

El último paso a elegir consiste, simplemente, en escoger el método numérico que se utilizará para la minimización. Los más populares son: los métodos del gradiente (descenso de máxima pendiente o gradiente conjugado) y los métodos de cuasi-Newton (BFGS y LM). Debido a la complejidad del problema de minimización y al coste computacional de las evaluaciones de la función y el gradiente, incluso utilizando algoritmos eficientes, es preferible el uso de métodos sofisticados. Los mejores resultados se obtienen en general con el algoritmo LM y este será, por tanto, el que se utilizará en todo el análisis empírico.

2. Datos y preproceso

Con el fin de ilustrar el comportamiento de las redes neuronales para el ajuste y la predicción de series temporales, se han escogido dos ejemplos donde el comportamiento de la predicción lineal es muy dispar. A continuación se comentan las series elegidas y el preproceso que se ha efectuado con cada una de ellas.

2.1 Índice IBEX35

La serie elegida en primer lugar, ver Fig. 1, corresponde a mediciones de los datos diarios de cierre del índice IBEX35 (Bolsa de Madrid). Se han recopilado 401 datos, que van desde el 14 de febrero de 1992 hasta el 20 de Septiembre de 1993. Esta serie se ha escogido por ser el ejemplo clásico de serie con la que la modelización y elaboración de predicciones mediante la metodología Box-Jenkins no ha ofrecido resultados satisfactorios.
Para regularizar la serie se han tomado logaritmos y primeras diferencias. Esto ha dado una serie con apariencia de ruido blanco y en una escala del orden de las centésimas. A fin de tener los datos mejor escalados se ha optado por multiplicar por cien, ver Fig. 2.

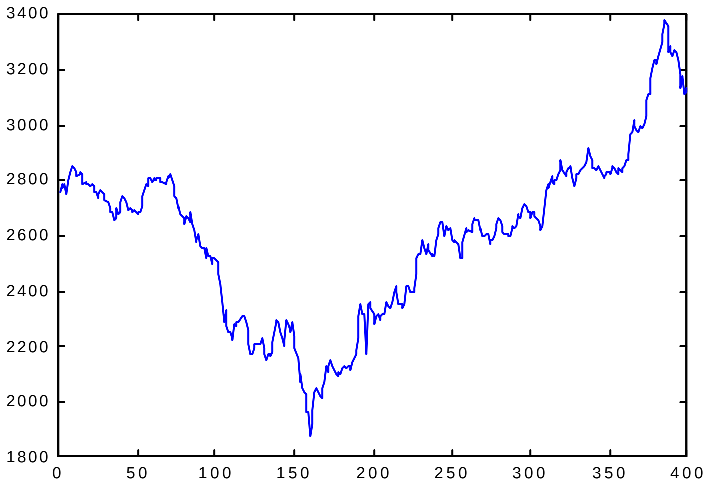


Fig. 1.- Serie original: Índice IBEX35

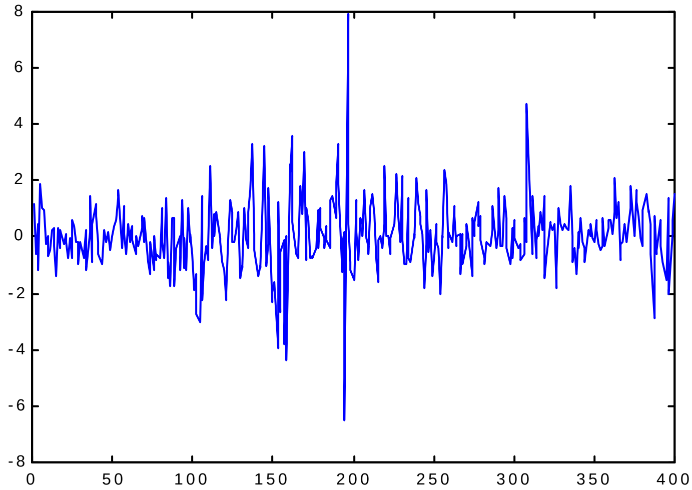


Fig 2.- Serie preprocesada: Índice IBEX35

2.2 Agregado monetario M1

La serie elegida en segundo lugar corresponde al agregado monetario de tipo M1, medido con periodicidad mensual. Se manejan un total de 444 datos, correspondientes a mediciones desde Enero de 1962 hasta Diciembre de 1998. Esta serie se ha escogido por presentar una problemática opuesta a la de la serie anterior, es una serie donde la aproximación lineal es particularmente buena y también, porque representa un modelo estacional, y poder así mostrar la aplicabilidad de las redes neuronales a este tipo de situaciones.

En cuanto al preproceso, lo primero que se ha hecho ha sido, como en el caso anterior y por las mismas razones, tomar logaritmos. La serie muestra una clarísima tendencia lineal y por ello se le han tomado dos diferencias para eliminarla.

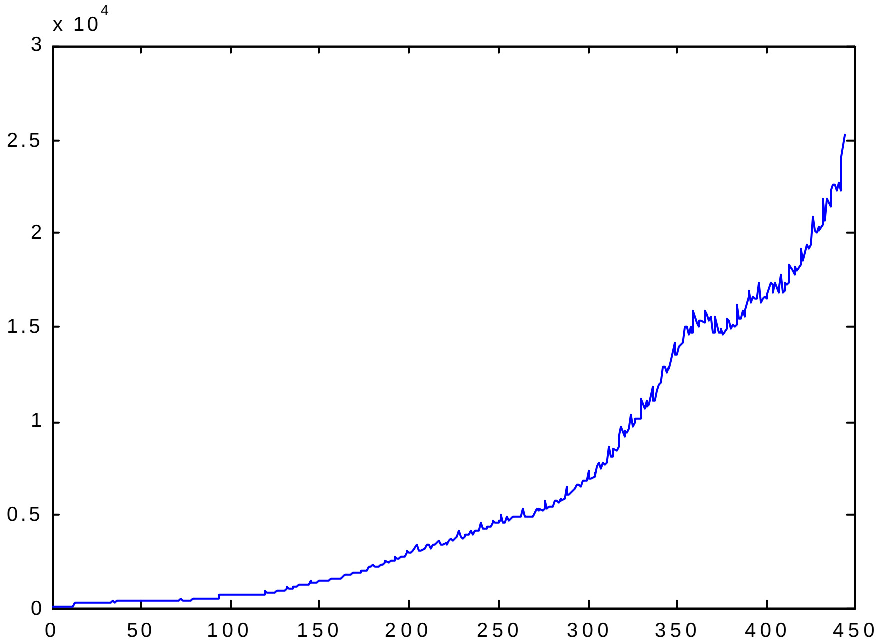


Fig 3.- Serie original: Cantidad de dinero M1

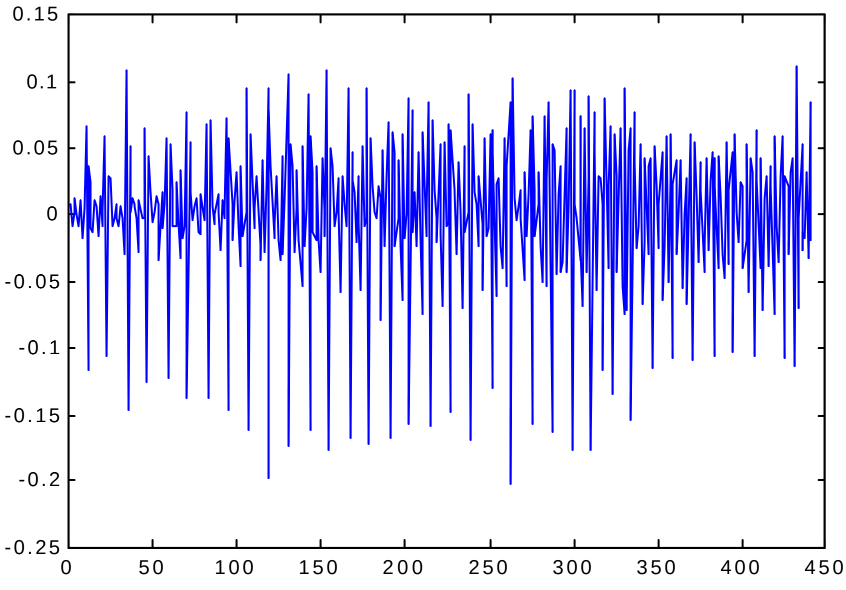


Fig 4.- Serie preprocesada: Cantidad de dinero M1

3. Estimación y resultados

3.1 Serie IBEX35

En el trabajo que se ha realizado y se presenta a continuación, se ha dividido la muestra en dos regiones, la de estimación y la de validación, correspondientes a 300 y 100 valores respectivamente. Con los 300 primeros se han estimado los modelos y efectuado predicciones para los 100 siguientes valores. Para comprobar la bondad predictiva del modelo se han cotejado estos valores con las realizaciones reales de la serie. El criterio que se ha usado para comparar los resultados ha sido el error cuadrático medio[3].

Se ha estimado la serie utilizando una red neuronal con una capa oculta de 10 neuronas del tipo tangente hiperbólica y un proceso AR. Ambos modelos emplean como entrada los 5 valores anteriores de la serie.

A continuación, en la Fig. 5, se presenta un tramo del ajuste intramuestral (valores 101-200) de la serie. En línea continua es la serie original, la predicción mediante redes en "+"y la predicción lineal en "*". Observamos que el ajuste de la red es bastante bueno, en tanto que el ajuste del modelo lineal es escaso.

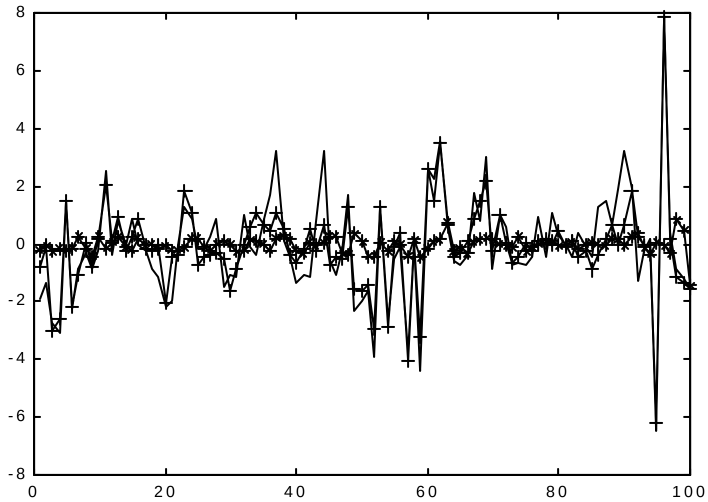


Fig. 5.- Ajuste intramuestral del IBEX35

[3] Calculado con la fórmula habitual: $ECM = \sqrt{\sum (y_t - \hat{y}_t)^2 / N}$

Esta ventaja disminuye en la muestra de validación, donde el ajuste es peor para ambas series, sin embargo, la red continúa captando las dinámicas existentes y los resultados son muy superiores a los obtenidos con el modelo lineal. Nótese asimismo que el modelo lineal predice sólo la media (0) a partir de las primeras iteraciones, en tanto la red neuronal continúa dando valores. A continuación, se presenta en la Fig. 6, un detalle de las aproximaciones extramuestrales (i.e. fuera del periodo de estimación) y los valores reales (como se hizo anteriormente, la línea continua es la muestra real, en "+"la aproximación por redes neuronales y en "*"la aproximación por AR, que no es más que la media).

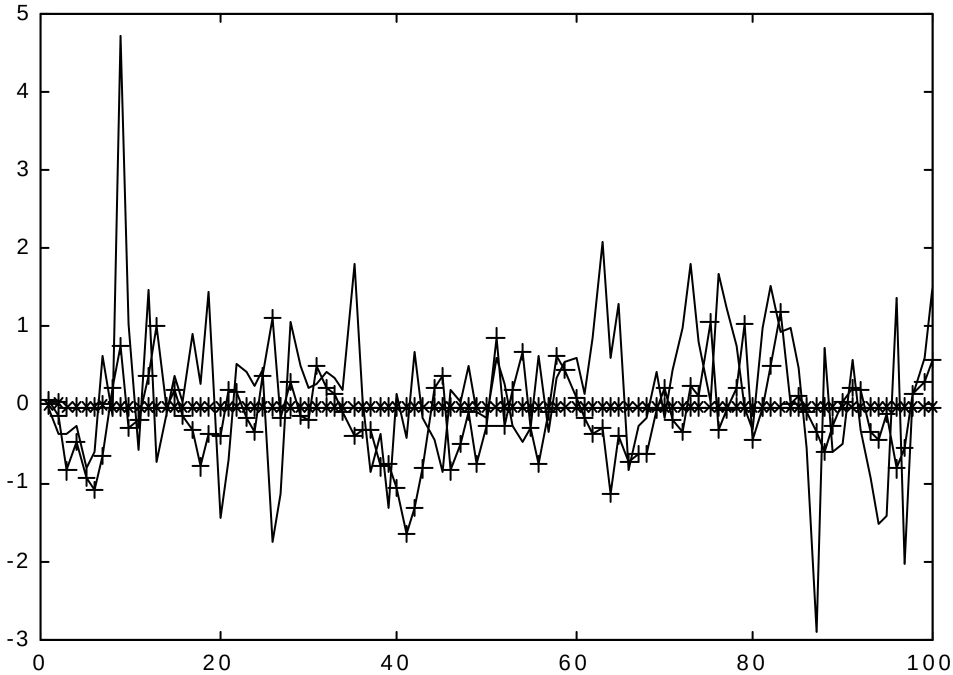


Fig 6.- Serie IBEX35 y predicciones

Se ha incluido esta gráfica sólo con el fin de mostrar cómo se comportan las predicciones a muy largo plazo. Por supuesto, predecir 100 periodos adelante una serie temporal es un objetivo excesivamente ambicioso. Por ello, se mostrará en la Fig. 6, un detalle de las predicciones a 20 periodos, así como sus respectivos errores cuadráticos medios en la tabla adjunta (con la misma notación para cada serie)

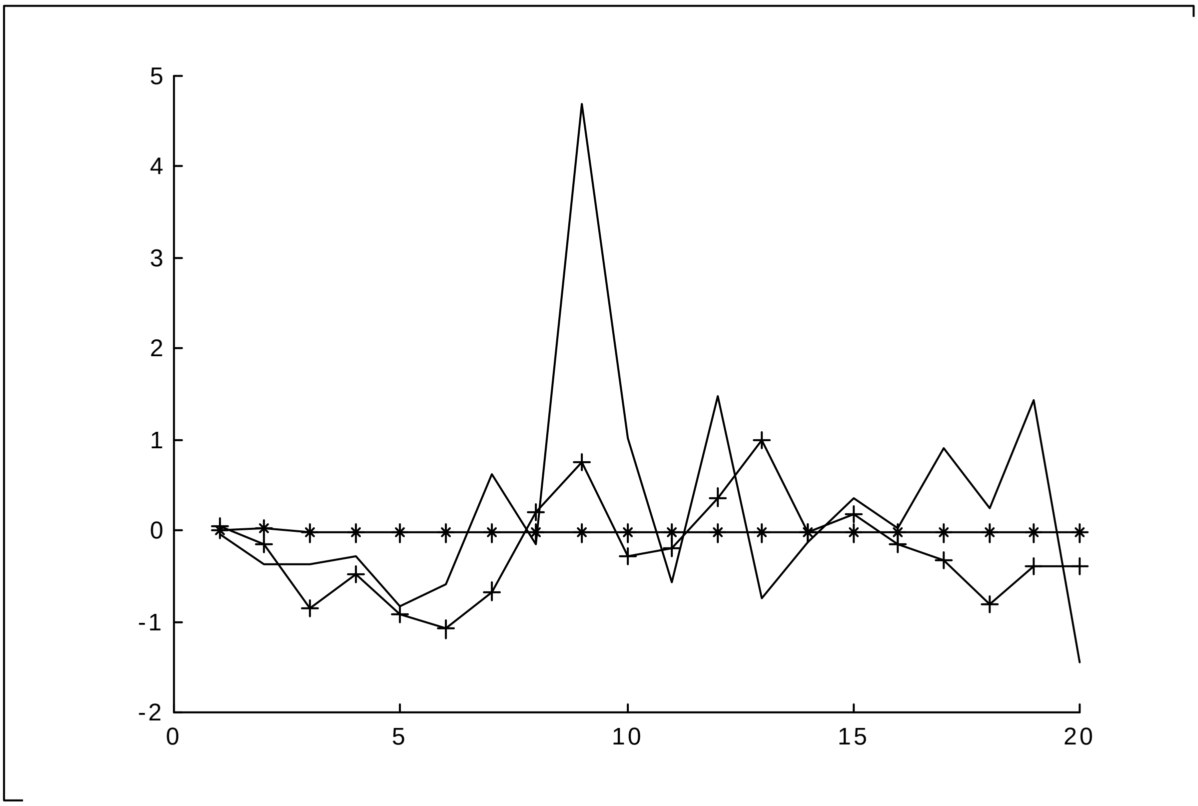


Fig. 7.- Serie IBEX 35 y predicciones a 20 periodos

	ECM Intramuestral	ECM 5 periodos adelante	ECM a 10 periodos adelante	ECM a 15 periodos adelante	ECM a 20 periodos adelante
Red neuronal	0.7449	0.2664	1.4007	1.2647	1.2459
Predicción lineal	1.2563	0.4391	1.5831	1.3722	1.2909

Se observa que la red neuronal es claramente superior intramuestralmente y en los primeros valores de la validación, si bien esta ventaja se va haciendo más pequeña a medida que nos alejamos de la muestra conocida.

3.2 Serie Agregado monetario M1

Para el estudio de esta serie siguiendo la estructura anterior, se ha dividido la muestra en dos regiones, la de estimación y la de validación, correspondientes en este caso a 350 y 92 valores respectivamente. La estimación se ha llevado a cabo con los 350 primeros valores, extendiendo después la serie hacia delante para elaborar predicciones y comparando los resultados con los valores reales. El criterio elegido para evaluar la bondad del modelo ha sido el error cuadrático medio. Para comprobar la calidad predictiva del modelo se han cotejado estos valores con las realizaciones reales de la serie. El criterio que se ha usado para comparar los resultados ha sido de nuevo el error cuadrático medio.

Se ha estimado la serie utilizando una red neuronal con una capa oculta de 10 neuronas del tipo tangente hiperbólica y un proceso AR. Puesto que la serie posee un fuerte componente estacional, ambos modelos emplean como entrada los 3 valores anteriores de la serie y los retardos 12 y 24.

A continuación, en la Fig. 8, se presenta el ajuste intramuestral (valores 151-250) de la serie. En línea continua la serie original, la predicción mediante redes en rayas cortas y la predicción lineal en rayas trazos largos. El ajuste es francamente bueno en ambos casos sin apreciarse grandes diferencias entre las dos aproximaciones .

La prueba de la eficacia de la modelización, sin embargo, se encuentra en los errores que se cometen cuando el modelo trata de predecir fuera de los valores que ya conoce. Por ello, se han realizado predicciones reales y se han comparado los resultados con los valores de la serie original que no se habían considerado. Se muestran en la Fig. 9, los primeros 20 valores de predicción y en la tabla adjunta están los errores cuadráticos medios. En la figura 9, la serie con "*" representa la predicción lineal, en tanto la línea continua es la original y la serie con "+" la predicción con redes neuronales.

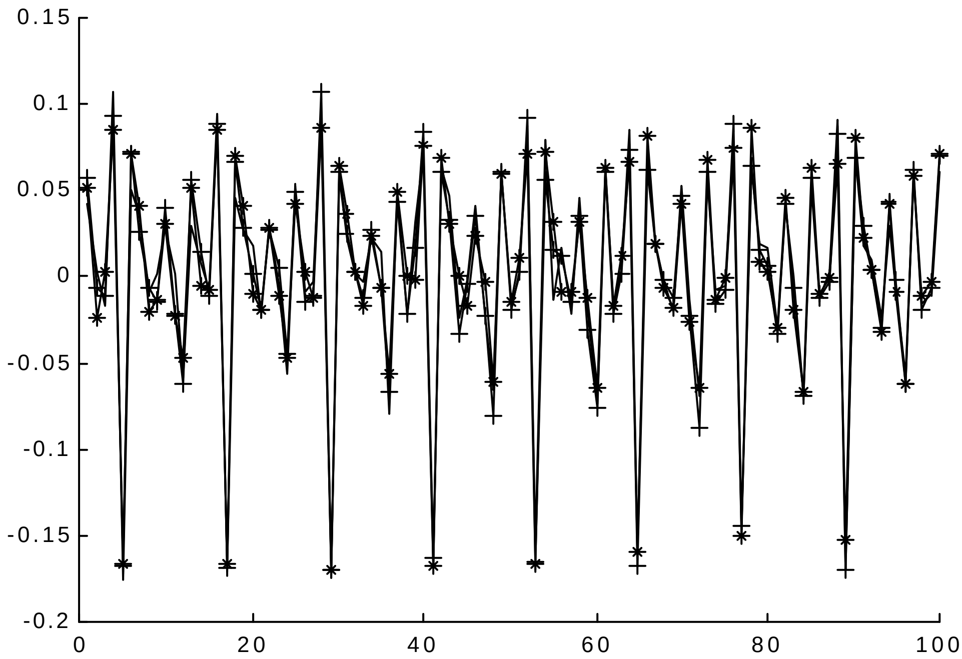


Fig 8..- Serie M1 y aproximaciones

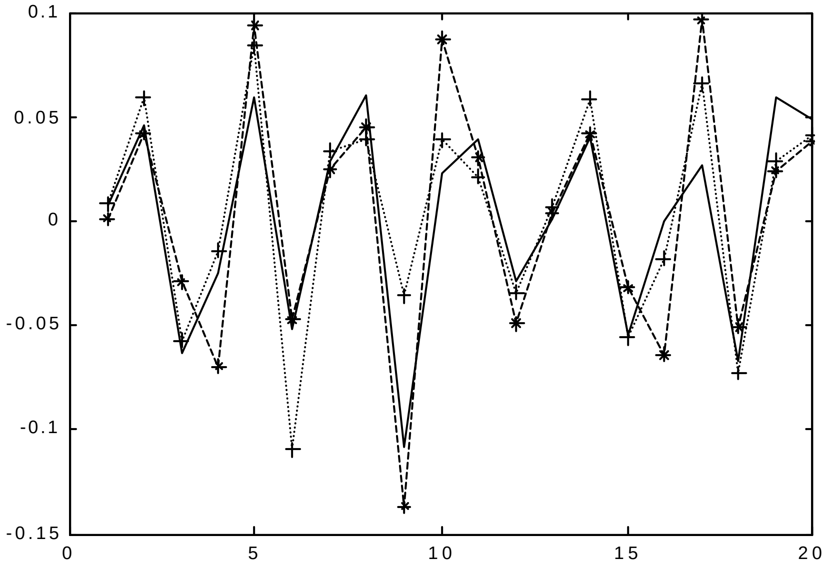


Fig 9.-. Serie original y predicciones de la serie M1

	ECM Intramuestral	ECM 5 periodos adelante	ECM a 10 periodos adelante	ECM a 15 periodos adelante	ECM a 20 periodos adelante
Red neuronal	0.0106	0.0134	0.0319	0.0269	0.0261
Predicción lineal	0.0178	0.0301	0.0312	0.0269	0.0327

Se observa que, en este caso, las dos modelizaciones son muy similares. Aparece cierta ventaja de la red dentro de muestra y en los primeros periodos, pero enseguida disminuye.

4. Conclusiones

En este artículo se ha presentado la metodología de redes neuronales desde un punto de vista básico, contando el tipo de problemas que puede resolver y la modelización que implica para el análisis de series temporales. El punto fuerte de esta metodología sobre el análisis usual es el hecho de permitir el estudio general de series sin restringirnos a modelos lineales ni necesitar especificar el tipo de no-linealidad.

Se han tratado dos series con diferentes problemáticas y con los cuales la metodología usual Box-Jenkins ha arrojado resultados dispares. Asimismo se ha mostrado cómo la estacionalidad no es un problema para la modelización con redes. De hecho este tipo de modelos permite incluir cualquier tipo de modelo ARIMA como caso especial.

Los resultados que se han obtenido con la serie IBEX35 muestran que es posible encontrar cierta estructura en ella y que el hecho de que los modelos ARMA no la encuentren se debe a su restricción a la linealidad más que a que la serie sea impredecible. Es necesario, sin embargo, un análisis más exhaustivo de esta serie, pues los modelos básicos de redes neuronales que hemos planteado no den resultados enteramente satisfactorios, si bien superan ampliamente a la metodología Box-Jenkins.

En cuanto a la serie de agregado monetario, se observa que la modelización ARIMA es más que satisfactoria y las redes neuronales mejoran sólo ligeramente estos resultados. Esto nos indica que las redes sólo dan una ventaja clara cuando realmente la metodología fracasa, pero que, en cualquier caso, su uso no empeora los resultados.

La restricción a modelos AR para el caso lineal puede parecer restrictiva. Los autores han considerado, sin embargo, que en una presentación de la metodología en la que sólo se incluyen modelos sin estado, resultaba más ilustrativo incluir comparaciones con modelos lineales sin estado (modelos AR). En cualquier caso, en las series que se han considerado, los modelos AR y los ARMA generales arrojan resultados harto similares (en el IBEX35 es un paseo aleatorio y en la M1 no hay grandes diferencias por ser muy bueno el ajuste).

Finalmente, sólo queda insistir en la flexibilidad de esta metodología para enfrentarse a gran cantidad de problemas y el hecho de que pocas son las decisiones que deben tomarse para la modelización con redes, similares a la metodología Box-Jenkins. Esperamos haber mostrado satisfactoriamente la aplicabilidad y utilidad de las redes neuronales en los problemas de la economía.

Bibliografía

[1] Demuth, H. and Beale, M. (1998). Neural Network Toolbox (manual de introducción a las redes y su programación en Matlab). Mathworks Inc.

[2] Wei, W. W. S. (1990) Time Series Analysis. Addison-Wesley Publishing Company

[3] Elman, J. L. (1990). Finding Structure in Time. Cognitive Science vol 14, pp 179-211

[4] Battiti, R.(1992) First and Second Order Methods for Learning: Between Steepest Descent and Newton's Method. Neural Computation, vol 4 nº2 pp 141-166

[5] Abraham, B.and Ledolter, J. (1983) Stadistical Methods for Forecasting. John Wiley New York

[6] Box, G. E. .P., Jenkins, G. M. (1976) Time Series Analysis, Forecasting and Control. Holden-Day, San Francisco

[7] Rumelhart, D. E., Hinton, G. E., Williams, R. J.(1986). Learning Internal Representations by Error Propagation. Parallel Data Processing vol 1 Cambridge MA MIT Press pp 318-362

[8] Wasserman, P. D. (1993). Advanced Methods in Neural Computing. Van Nostrand Reinhold New York

[9] Hagan, M. T. Menhaj, M. (1994). Training Feedforward Networks with the Marquadt Algorithm. IEEE Transactions on Neural Networks vol5 nº6 pp 989-993

[10] Caudill, M. (1989) Neural Networks Premier. Miller Freeman Publications San Francisco

[11] Dennis, J. E., Schnabel, R. B.(1983).Numerical Methods for Unconstrained Minimization and Nonlinear Equations. Prentice-Hall, Englewood Clifts

[12] Fletcher, R., Reeves, C. M. (1964) Function Minimization by Conjugate Gradients. Computer Journal, vol 7. pp 149-154