

*Estudios de Economía Aplicada* (1996)  
No. 5, pág. 5-33

# **Análisis comparativo de los métodos de previsión univariante, Box-Jenkins, Redes neuronales artificiales y espacios de Estado**

DAVID DE LA FUENTE GARCÍA  
RAÚL PINO DIEZ  
CRISTINA SUÁREZ RIESTRA  
JOSÉ LUIS MAYO RODRÍGUEZ  
*Universidad de Oviedo*

Esta nueva versión incluye todas las correcciones sugeridas por el censor, las cuales me han parecido oportunas y por las que les quedamos muy agradecidos.

## **RESUMEN**

En el presente trabajo, se presenta un análisis comparativo de los resultados obtenidos al calcular las previsiones de series temporales univariantes mediante las metodologías de Box-Jenkins, Redes Neuronales Artificiales y Espacio de Estados. Para cada uno de los métodos, en primer lugar se hace una somera descripción teórica y se calculan previsiones. Al final del trabajo, se realiza un estudio comparativo de los tres métodos, analizando ventajas e inconvenientes de cada uno de ellos.

**Keywords:** Previsión, Serie temporal univariante, Modelos ARIMA, Espacio de Estados, Redes Neuronales.

A comparative analysis of univariate forecasting methodologies, Box-Jenkins, Neural Networks, and State Space.

## **ABSTRACT**

In this paper, we present a comparative analysis of the results obtained when forecasting univariate time series using Box-Jenkins, Neural Nets and State Space methodologies. For each of these methods, we first describe theoretical aspects and later we make calculations. Finally, we compare all the three methods, analysing advantages and drawbacks of each of them..

## 1. Introducción

El cálculo de adecuadas previsiones, es una ventaja competitiva muy importante en el entorno actual de crisis económica. Por ello, en el presente trabajo hacemos una descripción de diferentes enfoques del cálculo de previsiones; por una parte, realizamos previsiones sobre una serie temporal mediante el enfoque de Box-Jenkins. Por otra parte, utilizamos una de las nuevas herramientas informáticas basadas en la teoría de "Redes Neuronales Artificiales", también para obtener previsiones y, por último, planteamos de una forma teórica y práctica la metodología de espacio de estados, calculando las previsiones basandonos en la teoría de espacio de estado balanceado introducido por Moore y siguiendo la metodología de Mittnik para la identificación y estimación de los modelos

## 2. Procedimiento de Box-Jenkins

Vamos a dividir este apartado en dos partes, una somera introducción teórica de la metodología y una segunda en la que se detallan los resultados obtenidos de su aplicación a varios casos concretos.

### 2.1. Introducción

El procedimiento ARIMA (autorregresivo, integrado, media móvil), identifica, calcula estimaciones y previsiones de modelos, utilizando los métodos descritos por Box-Jenkins (1976). Los procedimientos ARIMA, permiten modelar una serie temporal discreta como una función de términos autorregresivos y términos media móvil. Pueden incluirse en el modelo factores estacionales y no estacionales de cada tipo, llamados habitualmente: MA, AR, SMA y SAR.

El proceso de modelado de series temporales según Box-Jenkins, consta de varias fases:

1. Identificación y propuesta de un modelo para los datos observados.
2. Estimación de los parámetros y contrastes.
3. Crítica y diagnóstico del modelo.
4. Validación del modelo. (Si no es válido se propone otro modelo en el paso 1).
5. Previsiones.

También, se modelan, en su caso, la serie diferenciada o la no diferenciada para calcular tendencias, mediante la utilización de las diferencias estacionales y no estacionales.

La ecuación del modelo que va a ser ajustado, es la siguiente:

$$Y_t = \mu + \frac{\theta(L) \cdot \theta_s(L)}{\phi(L) \cdot \phi_s(L)} \cdot w_t$$

Este modelo, expresa los datos como una combinación de los valores pasados de la serie y los valores pasados de la entrada aleatoria (Box and Jenkins, 1976), donde:

$t$	Índice de tiempo.
$L$	Operador de retardo.
$Y_t$	Datos originales o una diferencia de estos datos.
$\mu$	Término constante.
$\theta(L)$	Operador media móvil no estacional. $1 - \theta_1 L - \dots - \theta_q L^q$
$\phi(L)$	Operador autorregresivo no estacional. $1 - \phi_1 L - \dots - \phi_p L^p$
$\theta_s(L)$	Operador media móvil estacional. $1 - \theta_1 L^s - \dots - \theta_q L^{qs}$
$\phi_s(L)$	Operador autorregresivo estacional. $1 - \phi_1 L^s - \dots - \phi_p L^{ps}$
$w_t$	Error aleatorio.

Este modelo se puede representar con la siguiente notación,  $(p,d,q) \times (P,D,Q)_s$ , donde:

$p$	Orden del término autorregresivo no estacional.
$d$	Orden de la diferencia no estacional.
$q$	Orden del término media móvil no estacional.
$P$	Orden del término autorregresivo estacional.
$D$	Orden de la diferencia estacional.
$Q$	Orden del término media móvil estacional.
$s$	Longitud de la estacionalidad.

La identificación del modelo, se obtiene mediante el cálculo de las funciones de autocorrelación, autocorrelación parcial u otras herramientas, y la validación mediante la obtención de ruido blanco y varianza mínima en los residuos.

Para la estimación de los parámetros del modelo propuesto, se utiliza el algoritmo de Gauss-Marquardt, que es un método de optimización no lineal con restricciones y combina los algoritmos iterativos de Gauss-Newton y el de máxima pendiente. La dificultad de la estimación no lineal con restricciones, es que puede haber más de un mínimo y se puede llegar a situaciones en que encontremos el mínimo local en vez del global. Ello exige, probar combinaciones de parámetros que permitan partir de una buena posición en la superficie de coste y comprobar que tenemos la varianza residual mínima.

Existen en el mercado bastantes paquetes comerciales que utilizan la metodología de Box-Jenkins univariable, STATGRAPHICS, BMDP, RATS, TSP, SPSS etc. Algunos de ellos, utilizan funciones de transferencia, como el BMDP, y muy pocos modelado multivariable (varias entradas y varias salidas), por ejemplo, SCA y RATS (solo modelo AR). Incluso, existen algunos paquetes de cálculo automático de previsiones univariantes como son AUTOBOX, FOCA etc. y últimamente han aparecido sistemas expertos en previsión univariante, como el SCA EXPERT.

## 2.2. Previsión según Box-Jenkins

Vamos a calcular previsiones para cuatro series temporales, dos de ellas recogidas en los trabajos de Box y Jenkins (1976), las series "E" y "G", que a partir de ahora denominaremos "BJ06" y "BJ08" respectivamente, y otras dos series "UN05" y "UN17", obtenidas de la base de datos de Reilly D.P. (1980) (ver anexo). Dos de las series, BJ06 y UN17, (figuras 1 y 2),

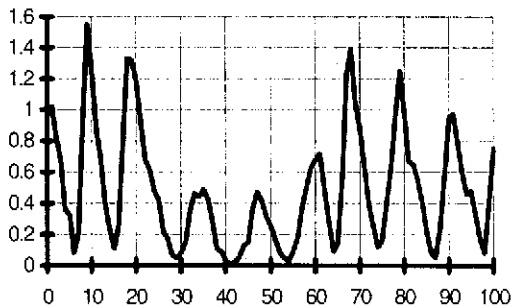


Figura 1.

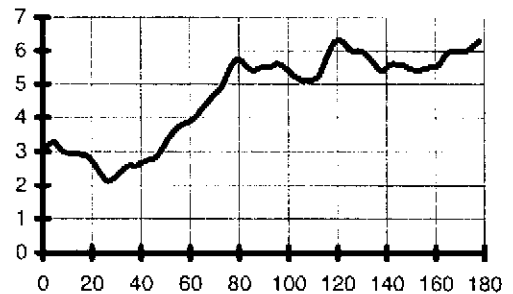


Figura 2.

son no estacionales mientras que las otras dos, UN05 y BJ08 (figuras 3 y 4) sí lo son. De esta manera cubrimos un amplio espectro dentro de las series temporales univariantes.

La serie BJ06.DAT (figura 1), consta de 100 datos, de los cuales se han utilizado los 90 primeros para deducir el modelo, reservándose los 10 últimos para comprobar la bondad de las previsiones calculadas. En el caso de la serie UN17.DAT (figura 2), consta de 178 datos de los cuales se han reservado los últimos 28 para compararlos con las previsiones calculadas.

En el caso de las dos series estacionales, la serie UN05.DAT (figura 3.), consta de 154 datos, de éstos se utilizarán los 134 primeros para determinar el modelo, reservándose los restantes para comparar con las previsiones calculadas. Y, por último, la serie BJ08.DAT (figura 4.), está compuesta por 144 datos, el modelo se deducirá a partir de los 134 primeros, mientras que se reservan los 10 últimos para comprobar la bondad de las previsiones calculadas.

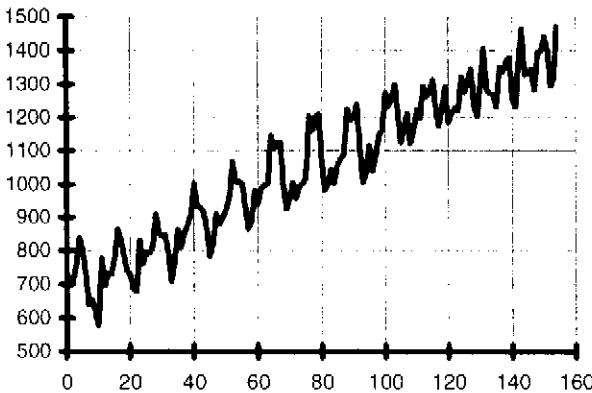


Figura 3.

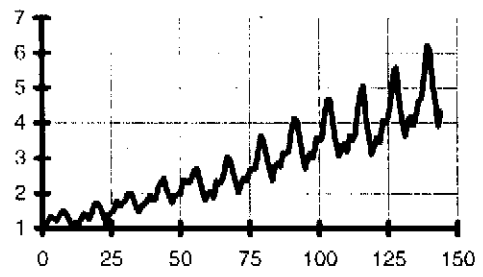


Figura 4.

Los modelos de Box-Jenkins, para las cuatro series de trabajo, han sido calculados utilizando el programa STATGRAPHICS, con los resultados que se recogen en la tabla 1:

Serie Temporal	Modelo de Box-Jenkins
BJ06	$(1-1,3 \cdot B+0,6 \cdot B^2) \cdot Z=CONST+NOISE$
UN17	$(1-1,4 \cdot B+0,5 \cdot B^2) \cdot (1-B) \cdot Z=NOISE$
UN05	$(1-0,6 \cdot B) \cdot (1-B^{12}) \cdot Z=CONST+NOISE$
BJ08	$(1-B) \cdot (1-B^{12}) \cdot LogZ=(1-0,33 \cdot B) \cdot (1-0,57 \cdot B^{12}) \cdot NOISE$

Tabla 1.

Las previsiones de las cuatro series temporales, calculadas aplicando la metodología de Box-Jenkins, aparecen recogidas en las figuras 8 a 11.

### 3. Redes Neuronales Artificiales

Análogamente al apartado anterior, dividiremos éste en dos partes, una primera con una breve introducción teórica y otra de cálculo de previsiones de las series de estudio.

#### 3.1. Introducción a las redes neuronales

Las redes neuronales (o redes de neuronas artificiales), son modelos matemáticos simplificados de las redes de neuronas que constituyen el cerebro humano. Estos modelos están compuestos por un conjunto de "neuronas artificiales" o conjunto de unidades que procesan e intercambian información. Las neuronas de una red, están estructuradas en distintas capas, de forma que una neurona de una capa está conectada con las de la capa siguiente, a las que puede enviar información.

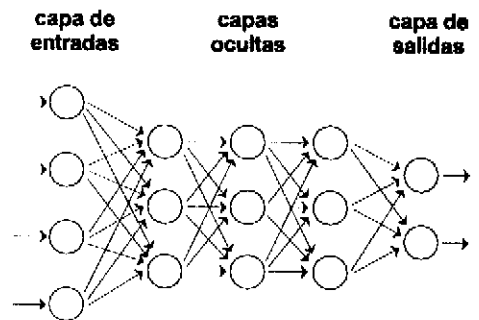


Figura 5.

Tal como se refleja en la figura 5, la arquitectura más habitual consiste en una capa de neuronas de entrada que recibe la información "del exterior", una serie de capas intermedias (u "ocultas") y una capa de salidas, que proporciona "al exterior" el resultado del trabajo de la red.

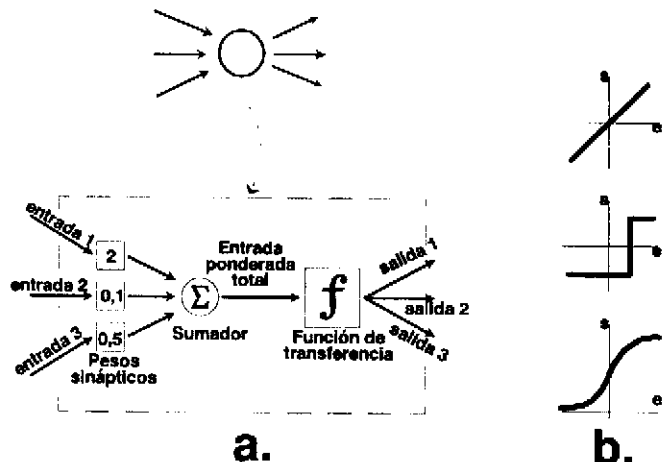


Figura 6.

Cada neurona, tal como se muestra en la figura 6.a, constituye una "unidad de procesamiento" de información, convierte un conjunto de señales de entrada en una salida que es difundida a las neuronas de la capa siguiente. Esta conversión se realiza en dos etapas: primero, cada una de

las señales de entrada es multiplicada por un coeficiente de ponderación ("peso sináptico") atribuido a la conexión; todos los productos son sumados para obtener una cantidad denominada "entrada ponderada total". En una segunda fase, cada unidad utiliza una función de transferencia entrada-salida, o función de activación, que transforma la entrada ponderada total en una señal de salida que es la que se difunde a las neuronas de la capa siguiente. La función de transferencia (figura 6.b.) puede ser de tres tipos, (Lippmann, 1987):

1. *Lineal*. La actividad de salida es proporcional a la entrada ponderada total.
2. *De umbral*. La salida queda fija a uno de dos niveles, dependiendo si la entrada ponderada total es mayor o menor que cierto valor crítico denominado "umbral".
3. *Sigmoide*. La salida varía de forma continua dependiendo de la entrada ponderada total, pero esta dependencia no es lineal.

Habitualmente, se suele utilizar la sigmoide como función de transferencia cuando se trata de aplicar la tecnología de redes neuronales al procesado de señales no-lineales (Lapedes y Farber, 1987), aunque es necesario tener presente que las tres son aproximaciones bastante burdas de la actividad de las neuronas reales.

El proceso de aplicación de la tecnología de redes neuronales artificiales a un problema concreto, consta de tres etapas fundamentales:

1. **Diseño de la Red**. Es necesario decidir la arquitectura que va a tener la red, lo cual implica determinar el número de neuronas de la capa de entradas, el número de capas ocultas y las neuronas que contendrá cada una de ellas, y, por último, el número de neuronas de la capa de salidas. La arquitectura de la red dependerá, como es lógico, del problema concreto que se quiera resolver.

En el caso que nos ocupa, queremos aplicar la tecnología de redes neuronales artificiales a la previsión de series temporales, basándonos en los datos históricos de la serie, por lo tanto, el número de neuronas de la capa de entrada coincidirá con el número de datos anteriores de la serie que son necesarios para calcular un valor concreto. La capa de salida estará compuesta por una sola neurona cuya salida será el valor de la previsión que queremos calcular. En cuanto al número de capas ocultas y las neuronas de estas capas, no existen reglas fijas que determinen los valores óptimos. Un número muy

pequeño de capas ocultas puede hacer que el proceso de entrenamiento se alargue excesivamente, mientras que demasiadas capas ocultas llevan a que se produzca una memorización de los datos, en lugar de la deducción de los patrones que se derivan de los datos, con lo que se falsea la predicción (Richeson y Zimmermann, 1994).

Por otra parte, el número de neuronas en cada capa oculta, dependerá de la complejidad del problema a estudiar, aunque algunos autores dan ciertas recomendaciones que van desde: la utilización de complicadísimas fórmulas para su cálculo (Zurada, 1991), o indicar que el número de neuronas de la capa oculta debe ser como mínimo el 75% del número de neuronas de entrada (Salchenberger, 1992), hasta aplicar la teoría matemática clásica de Kolmogorov para calcular el número de neuronas como  $2k+1$ , siendo "k" el número de neuronas de la capa de entrada (Zaremba, 1990).

Hemos comprobado que el número de neuronas de la capa oculta, siempre que esté entre unos valores mínimo (por ejemplo el 75% de las neuronas de entrada) y máximo (5 veces ese mismo número), sólo influye en la velocidad de entrenamiento de la red, por lo que en nuestro caso hemos partido de un valor inicial ( $2k+1$ ), que posteriormente se ha ajustado, si durante el proceso de entrenamiento de la red se comprueba que mejora los resultados.

- 2. Entrenamiento de la Red Neuronal Artificial.** El entrenamiento de una Red Neuronal consiste en la utilización de un algoritmo (generalmente se usa el algoritmo "Back Propagation") para ajustar los pesos sinápticos de las conexiones entre las neuronas.

El proceso, que aparece esquematizado en la figura 7, consiste en presentar a la red inicial una batería de casos de entrenamiento, que se construyen utilizando los datos reales disponibles (el pasado de la serie temporal), tal como se comenta en el apartado siguiente. Cada uno de estos casos está compuesto por una serie de valores de entrada y el valor de salida correspondiente. Se asignan los valores de entrada a las neuronas de la capa de entradas, y se obtiene al final un valor de salida de la red neuronal. Esta respuesta se compara con la deseada u objetivo, mediante una función de error que da una medida de la eficacia de la configuración actual de pesos sinápticos de la red. El objetivo del aprendizaje es minimizar esta función de error.



Una vez calculado este error, se procede a realizar la fase "hacia atrás" ("backward") variando los pesos sinápticos de los enlaces en función de la magnitud del error cometido y de una constante  $e$  llamada "coeficiente, tasa o ritmo de aprendizaje" que varía entre 0 y 1. La elección de  $e$  es importante, ya que un

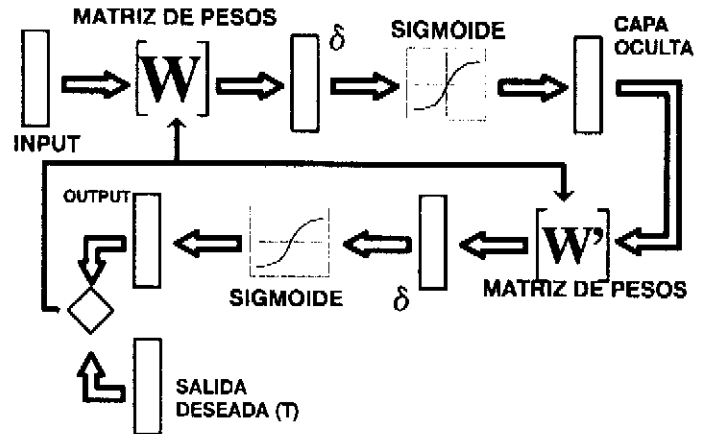


Figura 7.

valor bajo, da como resultado una lenta convergencia pues implicará variar los pesos muy poco, mientras que un valor excesivamente alto, puede conducir a oscilaciones en los pesos de la red, con lo que quizás nunca se alcancen los pesos óptimos. Un valor usual de partida para " $e$ " es 0,3 aunque puede ser ajustado ligeramente durante el proceso de aprendizaje de la red.

Una vez ajustados los pesos sinápticos, se vuelven a presentar los casos a la red y se vuelve a calcular el error, de acuerdo con él, se reajustan los pesos. Este proceso continuará hasta que el error obtenido esté dentro de unos límites previamente fijados como aceptables, en este momento, el entrenamiento de la red habría finalizado. Una forma sencilla de acelerar el entrenamiento de la red, es añadir un término de momento " $m$ " a la hora de ajustar los pesos, que recoja información sobre el último ajuste realizado (Chapman, 1994). Un valor habitual de partida suele ser  $m=0,7$ , aunque también puede ser ajustado a cualquier valor entre 0 y 1 durante el proceso de entrenamiento de la red.

3. **Utilización de la Red Neuronal en "Modo Recuerdo".** Una vez entrenada, la Red Neuronal está en condiciones de ser utilizada. Para ello, no hay más que presentar a la red un caso determinado (por ejemplo los últimos datos disponibles de una serie temporal) para que, utilizando los pesos sinápticos encontrados durante el proceso de entrenamiento, calcule la salida (la previsión del dato siguiente de la serie temporal).

La mayor parte de las aplicaciones de redes neuronales, se realizaron en lenguajes de programación convencionales como FORTRAN ó C. Sin embargo, de forma análoga a lo sucedido con los sistemas expertos, ante la expansión de la

tecnología y las dificultades que presentan algunas de las etapas de diseño y entrenamiento de las redes, diversas instituciones y compañías, han empezado a comercializar entornos de desarrollo o "shells" que facilitan diversos tipos de arquitecturas y algoritmos de entrenamiento. Nosotros, hemos utilizado el paquete comercial denominado "NEUROHELL" en aplicaciones sobre previsión, aunque existen otros como "ANSIM", "BACK PROPAGATION", etc, para ordenadores personales.

### 3.2. Previsión con redes neuronales artificiales.

De la misma manera que con la metodología clásica de Box-Jenkins, hemos calculado las previsiones de las cuatro series temporales utilizando para ello las redes neuronales artificiales. El sistema de trabajo seguido en los cuatro casos es similar por lo que sólo describiremos en detalle el cálculo de previsiones para la primera serie temporal, la BJ06. Como ya se ha comentado, esta serie consta de 100 datos, de los que se utilizan los 90 primeros para "entrenar" a la red neuronal, y se reservan los 10 últimos para compararlos con las previsiones que se realicen.

Siguiendo los pasos indicados anteriormente, la primera fase consiste en el diseño de la red neuronal, es decir, número de neuronas de entrada, número de capas ocultas y número de neuronas en cada una de ellas, así como el número de neuronas de la capa de salidas. Como se ha comentado anteriormente, el número de capas ocultas influye en la velocidad del proceso de entrenamiento, en nuestro caso, teniendo en cuenta el problema que intentamos resolver, es suficiente con una sola capa oculta, ya que las pruebas que se han hecho con dos o tres no mejoraban significativamente los resultados obtenidos con una, pero sí aumentaba de forma considerable el tiempo de entrenamiento. Por lo tanto, trabajaremos con redes neuronales del tipo "d-n-s", llamando "d" al número de neuronas de la capa de entrada, "n" número de neuronas de la capa oculta, y "s" el número de neuronas de la capa de salida.

El valor de "s" es evidente, sólo queremos pronosticar un valor de la serie temporal utilizando para ello una serie de valores anteriores de la misma; por ello sólo habrá una neurona en la capa de salidas, por lo tanto, será una red del tipo "d-n-1". El valor de "d", neuronas de la capa de entrada, dependerá del número de valores anteriores de la serie temporal que son necesarios para que la red deduzca un patrón o un modelo, de forma que pueda calcular el valor de salida correspondiente. El cálculo de "d" se realiza mediante pruebas sucesivas con varios diseños y escogiendo aquel con el que se obtienen los mejores resultados.

Se puede empezar con una red 7-15-1 . Siete neuronas de entrada implica que cada valor de la serie temporal dependerá directamente de los siete valores anteriores, ponemos 15 neuronas (resultado de aplicar la fórmula  $2d+1$ ) en la capa oculta como número inicial, y una de salida.

Para el proceso de entrenamiento de esta red, disponemos de los 90 valores históricos de la serie temporal BJO6. En algunas ocasiones, es útil hacer un procesado previo de los datos de entrada antes de presentarlos a la red para el entrenamiento, por ejemplo, para series que presentan grandes variaciones e irregularidades aleatorias, es conveniente trabajar con la serie en logaritmos de los datos de entrada, o efectuar un alisado exponencial en el caso de variaciones aleatorias menores (Kimoto et al, 1990). Además, es también útil normalizar los datos de entrada antes de presentarlos a la red neuronal, esta normalización se utiliza para reducir el rango del conjunto de valores y que sean los apropiados para la función de transferencia o activación que se está usando. Generalmente se normalizan los datos entre 0 y 1 para prevenir el efecto de la saturación de la función de transferencia utilizada (habitualmente una sigmoide).

Teniendo en cuenta este diseño de la red, se construye el conjunto de casos de entrenamiento; cada uno de ellos se compone de siete valores de entrada que determinan un valor de salida. Si llamamos  $X_1, X_2, \dots, X_{90}$  a los valores históricos (ya normalizados entre 0 y 1) de la serie temporal, se podrían construir 83 casos:

$$\begin{array}{l} X_1 , X_2 , X_3 , X_4 , X_5 , X_6 , X_7 \quad \text{--> } X_8 \\ X_2 , X_3 , X_4 , X_5 , X_6 , X_7 , X_8 \quad \text{--> } X_9 \\ \dots\dots\dots \dots\dots\dots \dots\dots\dots \dots\dots\dots \\ X_{83} , X_{84} , X_{85} , X_{86} , X_{87} , X_{88} , X_{89} \quad \text{--> } X_{90} \end{array}$$

Con estos 83 casos, comienza el proceso de entrenamiento de la red. Mediante las sucesivas pasadas de los casos por la red, y los ajustes que se van haciendo a los correspondientes pesos sinápticos de los enlaces entre las neuronas, se intenta reducir la diferencia entre la salida real y la salida que genera la red. El proceso termina cuando los errores entre la salidas real y generada por la red entran dentro de un intervalo previamente fijado como aceptable, o bien cuando, transcurrido un período de tiempo considerable, no se observan disminuciones en los errores, que quedan estabilizados en valores fuera del intervalo previamente fijado como aceptable.

Cuando el proceso de entrenamiento ha terminado, la red está preparada para ser utilizada en modo recuerdo, los enlaces entre las neuronas de las distintas

capas tendrán unos pesos sinápticos que representan el modelo o patrones de la serie temporal, de forma que se puede pronosticar un valor de salida introduciendo los valores de entrada correspondientes. En nuestro caso, hemos introducido de nuevo los 83 casos para que la red calcule las 83 salidas que componen la serie pronosticada, que podemos comparar con la serie original BJO6. Utilizamos para ello el Error Cuadrático Medio (MSE) calculado según la expresión:

$$\text{MSE} = \frac{\sum_{i=1}^n (x_i - s_i)^2}{n}$$

donde  $x_i$  son los valores reales de la serie BJO6,  $s_i$  los datos pronosticados, y  $n$  el total de casos de entrenamiento (en esta primera red, 83).

Ya hemos comentado que la obtención de la arquitectura óptima de la red en cuanto a número de neuronas de entrada, en la capa oculta y en la salida, en un principio sólo es posible mediante un proceso iterativo, entrenando sucesivamente una red tras otra. En nuestro caso hemos comenzado por la red 7-15-1, la red ha sido entrenada utilizando los 83 casos que se pueden construir con los valores históricos de la serie, y, cuando el entrenamiento ha finalizado (en este caso, después de más de 14.000.000 de pasadas de los casos de entrenamiento por la red), los pesos sinápticos de los enlaces entre las neuronas quedan fijos y la red está lista para su utilización en "modo recuerdo" para hacer previsiones. Se introducen a la red ya entrenada, los mismos 83 casos para que la red deduzca la salida correspondiente para cada uno de ellos, y se calcula el error cuadrático medio (MSE), como método de comprobación de la bondad de la red 7-15-1 para hacer previsiones.

El mismo proceso se ha seguido para entrenar una segunda red, con la arquitectura 6-13-1 (cada valor de la serie BJO6 depende de los seis valores anteriores), 5-11-1, 4-9-1, etc. (el número de neuronas de la capa oculta se ha calculado siempre utilizando la expresión  $2d+1$ , ya comentada), y también aumentando el número de neuronas de la capa de entrada, 8-17-1, 9-19-1, ... hasta la red 15-31-1, que ha sido la última que se ha entrenado. En todas ellas se han utilizado hasta más de 40.000.000 de pasadas con el programa "NeuroShell", hasta que se llegaba a un instante en el que la función de error se estabilizaba o bien bajaba hasta entrar entre los límites considerados como aceptables. En ese momento, se considera la red como suficientemente entrenada y apta para realizar previsiones. Después de calcular las series de previsiones, se calculó el error cuadrático medio (MSE), obteniéndose los valores que se recogen en la tabla 2.

Nombre	Arquitectura	Nº de pasadas	Tiempo	Error Cuadrático Medio (MSE)
BJ06-4	4-9-1	162.536.150	133:52:18	0.00040102
BJ06-5	5-11-1	72.325.400	48:31:12	0.00040021
BJ06-6	6-13-1	44.369.200	22:55:33	0.00038538
BJ06-7	7-15-1	14.646.300	10:04:58	0.00036106
BJ06-8	8-17-1	6.495.250	3:51:40	0.00033758
BJ06-9	9-19-1	5.116.450	3:34:16	0.00036637
BJ06-10	10-21-1	2.152.650	2:46:21	0.00032050
BJ06-11	11-23-1	1.868.650	2:17:23	0.00029489
BJ06-12	12-25-1	1.748.550	2:41:26	0.00034925
BJ06-13	13-27-1	1.441.350	2:39:03	0.00018807
BJ06-14	14-29-1	1.043.900	1:08:17	0.00037014
BJ06-15	15-31-1	729.200	0:59:36	0.00036058

Tabla 2.

Tal como se puede comprobar en la tabla 2, el error cuadrático medio más bajo se obtiene para la red 13-27-1 (cada valor de la serie depende directamente de los trece valores anteriores).

Una vez finalizado el proceso de entrenamiento de la red neuronal, se pasa al cálculo de previsiones, utilizando la red en "modo recuerdo", es decir, se introducen los nuevos casos a la red con los pesos sinápticos ya definitivos que se han ajustado durante el proceso de entrenamiento de la misma. En concreto, se introducen 10 casos para que la red calcule las previsiones  $S_{91}$ ,  $S_{92}$ , ...,  $S_{100}$ , y que representamos de la siguiente forma:

$X_{78}, X_{79}, X_{80}, X_{81}, X_{82}, X_{83}, X_{84}, X_{85}, X_{86}, X_{87}, X_{88}, X_{89}, X_{90} \rightarrow \hat{S}_{91} ?$   
 $X_{79}, X_{80}, X_{81}, X_{82}, X_{83}, X_{84}, X_{85}, X_{86}, X_{87}, X_{88}, X_{89}, X_{90}, X_{91} \rightarrow \hat{S}_{92} ?$   
 $X_{80}, X_{81}, X_{82}, X_{83}, X_{84}, X_{85}, X_{86}, X_{87}, X_{88}, X_{89}, X_{90}, X_{91}, X_{92} \rightarrow \hat{S}_{93} ?$   
 .....  
 $X_{87}, X_{88}, X_{89}, X_{90}, X_{91}, X_{92}, X_{93}, X_{94}, X_{95}, X_{96}, X_{97}, X_{98}, X_{99} \rightarrow \hat{S}_{100} ?$

De esta manera de obtienen los 10 valores que aparecen en la figura 8, junto con los valores reales de la serie BJ06.

Utilizando un sistema de trabajo similar, se pueden calcular las previsiones mediante redes neuronales artificiales para las otras tres series temporales,

obteniéndose los resultados que aparecen en las figuras 9, 10 y 11. En la tabla 3. se pueden ver las arquitecturas de la red neuronal que dió los mejores resultados en cada caso.

Serie Temporal	Estacional	Arquitectura de la Red Neuronal
BJ06	NO	13-26-1
UN17	NO	6-13-1
BJ08	SI	12-25-1
UN05	SI	12-25-1

Tabla 3.

#### 4. Representación en espacio de estado

Para realizar la modelización de procesos temporales, univariantes y multivariantes, también se puede utilizar un tipo de parametrización conocido bajo el nombre de espacio de estado. Los modelos de los sistemas representados en espacio de estado, pueden ser considerados como generalizaciones de los modelos tipo ARMA, VARMA, etc. Han sido utilizados de forma extensa en la teoría de sistemas, las ciencias físicas y la ingeniería; no obstante, pueden ser utilizados en otros campos como por ejemplo el económico. Por ello, la terminología utilizada procede en su mayor parte de esos campos anteriormente citados.

Podemos decir que bajo el punto de vista de la teoría de sistemas, un sistema dinámico admite dos tipos de representaciones, para establecer las posibles relaciones entre las entradas y las salidas al mismo: una externa y otra interna. La representación externa, suministra una relación explícita directa entre las entradas y las salidas; en cambio, la interna que es la que nos interesa en este caso, se basa en el concepto de estado y establece una relación indirecta entre entradas y salidas a través de los estados, por lo que necesita utilizar dos ecuaciones lineales.

Definimos como variables de estado de un sistema, el menor conjunto de variables, no necesariamente identificadas con variables físicas medibles, cuyo valor en un cierto instante, resume el pasado dinámico del mismo y es suficiente para predecir su evolución futura. La idea general que subyace detrás de estos tipos de representaciones para relacionarlos con los procesos temporales, es que una serie o un vector de series de tiempo  $y_1, y_2, \dots, y_T$ , dependerá de un vector

denominado de estado  $z_t$ , posiblemente inobservado, el cual es conducido por un proceso generalmente estocástico, aunque no necesariamente. La relación entre la serie o proceso vectorial que se maneje y el estado, es decir entre  $y_t$  y  $z_t$ , está descrita a través de una ecuación de medida u observación que en algunos casos, tiene incorporado un término de realimentación de las posibles entradas al sistema que representa tal serie y los términos de error o perturbación, y se expresa de la siguiente forma:

$$y_t = Cz_t + Dx_t + w_t \quad (1)$$

donde C y D son matrices características de tal ecuación, y  $w_t$  es el término de error en las medidas de observación, que suele identificarse con un ruido blanco,  $z_t$  representa el vector de estado y  $x_t$  representa las entradas al sistema. El vector de estado, o estado del sistema, tal y como hemos dicho, esta generado por la dinámica del proceso. La relación entre las entradas y las salidas en este tipo de representaciones es indirecta, ya que pasa siempre a través de la denominada ecuación de transición, que describe la evolución de los estados desde un instante de tiempo al siguiente, permitiéndonos conocer el estado en un instante  $t$ , a partir del conocimiento de un estado inicial y de las correspondientes entradas. Dicha ecuación, denominada de transición, como hemos dicho, está simbolizada por:

$$z_{t+1} = Az_t + Bx_t + Kw_t \quad (2)$$

donde las matrices A y B representan al sistema, y K esta relacionada con la ponderación de las perturbaciones o términos de error  $w_t$ .

#### 4.1. Modelos en espacio de estado balanceado.

Anteriormente, se han descrito las representaciones generales en espacio de estado por medio de la utilización de dos ecuaciones, sin hacer referencia para nada a las propiedades que ese tipo de representación podía reflejar, pero, existen representaciones concretas en espacio de estado que, debido a sus propiedades, se denominan representaciones en espacio de estado balanceado. Del mismo modo que un vector, de un determinado espacio vectorial, tiene unas coordenadas en un sistema canónico de referencia y diferentes coordenadas en otro, a pesar de ser el mismo vector, un sistema dinámico puede estar representado en un sistema de referencia 'canónico' o en un sistema 'balanceado', entre otros tipos de representaciones.

Como cabe suponer, un sistema dinámico admitirá infinitos tipos de representaciones de su descripción interna. Se puede pasar de unas a otras, realizando el correspondiente cambio de base, obteniendo un tipo de representación que cumple unas determinadas propiedades, siendo todas ellas representaciones equivalentes.

Si un sistema esta representado en espacio de estado balanceado, podemos afirmar que se trata de un sistema que es:

1. Controlable: es decir que es posible transferir el estado del sistema desde un estado inicial al origen de los estados, en  $k$  pasos.
2. Observable: que es posible obtener el vector de estado a partir de las entradas y de las salidas que son la únicas variables que se conocen del proceso.
3. Asintóticamente Estable: es decir que los efectos de las condiciones iniciales se desvanecen asintóticamente con el tiempo.
4. Representación Minimal: es decir que el número de estados elegido es mínimo y suficiente para representarlo.

Con estas propiedades las matrices que representan al sistema en espacio de estado balanceado, tienen determinadas características que son útiles a la hora de realizar el proceso de identificación y estimación. Basándose en esas propiedades, se llevan a cabo los correspondientes algoritmos que nos permiten estimar dichas matrices.

#### 4.2. *Estimación de los modelos en espacio de estado.*

En general el modelizador, que ha elegido este tipo de representación, desconoce las matrices con las que representar el sistema, sólo conoce los datos del proceso a modelizar y a partir de ellos, deberá encontrar estas matrices. Este proceso, se realiza utilizando el método de mínimos cuadrados de forma conveniente, tomando los datos del proceso y las entradas, que serán aleatorias en unos casos, y en otros, se pueden considerar como entradas, las propias variables exógenas retardadas. Una vez identificadas las matrices que representan al sistema (modelización), se puede analizar el mismo de cara a su comportamiento predictivo o con otros fines.



Según sea el proceso, trataremos con representaciones deterministas para el sistema, o estocásticas. Aunque los procesos reales son habitualmente del segundo tipo, en determinadas ocasiones, se puede trabajar sin calcular la matriz  $K$ . Si fuera necesario este tipo de representación no habría más que modificar la ecuación (3) y agrupar como "entradas", reales o estimadas tanto la  $x_t$  como la  $w_t$ , y realizar el proceso como si el sistema tuviera más entradas (Suárez y Mayo, 1995). El resto del proceso no variaría sustancialmente. En otros casos, si no se desea modelizar de forma estocástica, podría utilizarse un filtro, por ejemplo el de Kalman (Suárez y Mayo, 1994).

#### 4.3. Previsión en espacio de estado.

Si tenemos un sistema determinista, representado en espacio de estado, del siguiente modo:

$$\left. \begin{aligned} z_{t+1} &= \Lambda z_t + Bx_t \\ y_t &= Cz_t + Dx_t \end{aligned} \right\} \quad (3)$$

podemos obtener a partir de él, las matrices de observabilidad y de controlabilidad:

$$O = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \end{bmatrix} \quad (4)$$

$$C_x = [B \quad AB \quad A^2B \quad \dots]$$

El producto, de estas dos matrices, se denomina matriz de Hankel.

$$H_x^* = OC_x = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \end{bmatrix} [B \quad BA \quad BA^2 \quad \dots] \quad (5)$$

Es importante resaltar que estas matrices han de ser de rango finito, dado que, el sistema del que parten es de dimensión finita, por lo que posteriormente, se trabajará con dicha matriz finita. Por otro lado, si para un sistema se conoce la matriz de Hankel, es posible la determinación de las matrices que representan a dicho sistema, previo cálculo de las matrices de observabilidad y controlabilidad, siempre y cuando estemos tratando con sistemas balanceados, puesto que en estos sistemas estas matrices tienen las cuatro propiedades antes comentadas que nos asegurarán su existencia.

Bajo el punto de vista del cálculo de previsiones, el modelizador no conoce las matrices que representan al sistema, por lo que ha de estimarlas a partir de los datos del proceso, o de transformaciones de los mismos (si el proceso lo requiere por no ser estacionario, transformaciones tipo Box-Cox, diferencias, etc.), que es lo único que se conoce. Para estimar las matrices A, B, C y D, es necesario llevar a cabo la identificación de la matriz de bloques de Hankel, pieza clave para determinar las matrices del sistema, y para ello emplearemos un procedimiento mínimo cuadrático que exponemos a continuación. Para llevar a cabo tal proceso, es conveniente conocer el orden del sistema o una aproximación del mismo, eso se puede conseguir por medio del cálculo del AIC o de estadísticos similares como el FPE o BIC. (Granger y Newbold, (1986); Ljung (1987), Wei (1990), etc.). También, es necesario elegir un tamaño para la muestra con la que se va a trabajar, no siendo necesario tomar muestras demasiado grandes. Por una parte, conviene que no sean inferiores a cuatro veces el orden estimado para el sistema, pues se distorsionan los cálculos y se producen singularidades no deseadas. Por otra parte, muestras demasiado grandes tampoco son convenientes, pues los procesos con los que se trabaja pueden presentar alinealidades fuertes, lo que hace que no sea posible modelizarlos de forma única con un modelo lineal en espacio de estado.

Los pasos a seguir para llevar a cabo la identificación de dichas matrices, son los siguientes:

- En primer lugar, se calcula la matriz de parámetros, a partir de la cual, se irán construyendo los bloques de la matriz de Hankel. Esta matriz de parámetros se representa por la matriz de bloques:

$$\Theta = [M_0^x \quad M_1^x \quad M_2^x \quad \dots \quad M_k^x]^T \quad (6)$$

la cual, se calcula a partir de la operación matricial dada por:

$$\hat{\Theta} = (X^T X)^{-1} X^T Y \quad (7)$$

donde X e Y, son matrices finitas confeccionadas al efecto. En estas matrices, N representa el tamaño elegido de la muestra y k el orden estimado para el sistema.

$$Y = [Y_{k+1} \quad Y_{k+2} \quad \dots \quad Y_N]$$

$$X^T = \begin{bmatrix} X_{k+1} & X_{k+2} & \dots & X_N \\ X_k & X_{k+1} & \dots & X_{N-1} \\ \vdots & \vdots & & \vdots \\ X_1 & X_2 & \dots & X_{N-k} \end{bmatrix} \quad (8)$$

- Dados los parámetros  $M_i^x$ , ( $i = 0, 1, 2, \dots$ ), podemos construir las matrices  $H_x^x$ ,  $O$  y  $C_x$  a partir de las cuales podemos deducir las matrices del sistema. Es decir, construimos la matriz de Hankel finita del siguiente modo:

$$\hat{H}_x = \begin{bmatrix} \hat{M}_1^x & \hat{M}_2^x & \dots & \hat{M}_k^x \\ \hat{M}_2^x & \hat{M}_3^x & \dots & 0 \\ \vdots & \vdots & & \vdots \\ \hat{M}_k^x & 0 & \dots & 0 \end{bmatrix} \quad (9)$$

Para obtener el modelo en espacio de estado balanceado, se emplea la técnica de descomposición del valor singular (SDV), imprescindible por otro lado para realizar la correspondiente descomposición de la matriz de Hankel finita.

$$\hat{H}_x = \hat{O}\hat{C}_x \quad (10)$$

La SDV de  $\hat{H}_x$ , conduce a  $\hat{H}_x = \hat{U}\hat{Q}\hat{V}^T$ , donde  $\hat{U}^T\hat{U} = I$  y  $\hat{V}^T\hat{V} = I$ , y  $\hat{Q}$  es una matriz diagonal que contiene los valores singulares de  $\hat{H}_x$  ordenados de mayor a menor:

$$\hat{q}_1 \geq \hat{q}_2 \geq \dots \geq \hat{q}_{n_{\max}} \geq 0 \quad \text{con } n_{\max} = K \cdot \min(m, r)$$

Los factores de  $\hat{H}_x$ , se calculan por medio de las siguientes expresiones:

$$\begin{aligned} \hat{O} &= \hat{U}\hat{Q}^{1/2} \\ \hat{C}_x &= \hat{Q}^{1/2}\hat{V}^T \end{aligned} \quad (11)$$

donde  $\hat{Q}^{1/2}$ , denota la matriz diagonal que contiene la raíz cuadrada de los valores singulares de la descomposición. Una vez obtenidos, deducimos las matrices del sistema (3), como se describirá posteriormente.

El usar la SDV para obtener  $\hat{O}$  y  $C_x$ , es lo que origina que, la parametrización elegida, sea llamada representación en Espacio de Estado Balanceada. Se le denomina así, si los productos de las matrices de observabilidad y controlabilidad por sus respectivas transpuestas son diagonales e iguales, ver (Otter (1985), Aoki (1991)).

$$\begin{aligned} \hat{O}^T\hat{O} &= \hat{Q}^{1/2}\hat{U}^T\hat{U}\hat{Q}^{1/2} = \hat{Q} \\ \hat{C}_x\hat{C}_x^T &= \hat{Q}^{1/2}\hat{V}^T\hat{V}\hat{Q}^{1/2} = \hat{Q} \end{aligned} \quad (12)$$

El utilizar un determinado orden para el proceso, queda ratificado según el valor que tengan los coeficientes de la matriz diagonal de la descomposición singular, ya que si son próximos a cero se pueden eliminar, con la consiguiente

disminución del orden y sin necesidad de repetir todo el proceso; solo truncando las matrices estimadas en la aproximación de mayor orden. Esta es una propiedad que poseen internamente las representaciones en espacio de estado balanceadas, característica por otro lado que es muy atractiva, debido a que representaciones de más baja dimensión, pueden ser conseguidas simplemente eliminando las líneas sobrantes de las matrices A, B y C, y están anidadas en las obtenidas de mayores dimensiones. Una consecuencia importante de esto, es pues, que las aproximaciones anidadas de dimensión más baja, son fáciles de obtener sin tener que realizar de nuevo la estimación, operación que sí es necesario realizar en los modelos convencionales. Por tanto, los modelos de predicción de dimensión menor, se pueden evaluar con bajo coste y pueden servir como criterio de selección del modelo definitivo, dato muy interesante a tener en cuenta a la hora de realizar los estudios prácticos de aplicación de esta teoría, y que ha sido utilizado en el presente trabajo.

Una vez que se conocen las matrices de parámetros (6) y de Hankel (9), es posible determinar las restantes matrices siguiendo las indicaciones que se exponen a continuación:

- $D = M_0^\lambda$  (la matriz D del sistema es igual a  $M_0^\lambda$ ).
- La matriz C, viene dada por las primeras m filas de O.
- La matriz B, está dada por las primeras r columnas de  $C_x$ .
- Para deducir la forma de la matriz A y sus elementos, definimos la versión transformada de la matriz de observabilidad, que simbolizaremos por  $O^\dagger$  y que tiene la propiedad de ser igual al producto O por A. También, podemos reflejar la relación entre ellas de la siguiente forma

$$A = O^\# O^\dagger \quad (13)$$

donde  $O^\#$ , representa la inversa generalizada de la matriz O.

Una vez realizados los cálculos de las matrices estimadas, ya se tiene representado el sistema y se puede pasar a la etapa de predicción, de forma bastante automática para el número de previsiones que se desee. Hay que tener siempre en cuenta, que si se trabaja con la variable transformada, es necesario deshacer todas las transformaciones, así como ir construyendo las entradas al sistema de la forma conveniente, si se quieren conocer los valores reales estimados y los de predicción.

En todo el proceso, se puede trabajar con un estado inicial cero, o realizar un proceso de estimación del mismo (Suárez y Mayo, 1994). Los resultados de operar

de una forma o de otra, no son fundamentales en el desarrollo de las predicciones, debido a las características de la representación.

En el primer ejemplo de aplicación realizado en este trabajo (serie UN05), se han considerado como entradas la variable original transformada por la primera diferencia, tomándola desde el instante 12 como punto de partida y como salidas a modelizar, la variable original transformada, es decir, diferenciada trece veces; cara a conseguir un proceso estacionario. Según el método descrito anteriormente, se confeccionaron las matrices X e Y para la estimación de los parámetros; el orden elegido para este sistema ha sido 13, puesto que con órdenes mayores no se mejora el modelo y los elementos diagonales de Q se hacen muy pequeños. Se ha trabajado con una muestra de 121 datos, ya que los 13 primeros se perdieron en el proceso de diferenciación y se han reservado los 10 últimos para compararlos con las previsiones calculadas. Los resultados aparecen representados en la figura 10.

De forma similar se realizaron todas las demás previsiones para las otras tres series temporales, adecuando en cada caso la entrada (desencadenante del proceso), y salidas (serie en estudio o una transformación de la misma), obteniéndose los resultados que aparecen representados en las figuras 8, 9 y 11.

## **5. Análisis Comparativo.**

En este apartado, nuestro objetivo es comparar las previsiones obtenidas utilizando los tres métodos ya comentados, y extraer ciertas conclusiones acerca de la idoneidad de cada uno de dichos métodos con respecto al tipo de serie con la que se esté trabajando. Por ello, vamos a examinar tanto la "calidad de las previsiones" (calculando el error cuadrático medio de las previsiones calculadas por cada uno de los tres métodos), como el hecho de que pronostiquen con exactitud los cambios de tendencia de la serie original (los llamados "turning points"), ya que en muchas ocasiones es preferible una previsión menos exacta en cuanto al valor de la serie en un período determinado pero que detecte con cierta exactitud los períodos en los que la serie pasa de tener tendencia creciente a decreciente y viceversa. Este último aspecto es especialmente importante en el caso de utilizar las predicciones como herramienta de control de un proceso, ya que saber en qué momento va a variar la tendencia de la serie temporal puede ser importante para tomar decisiones.

	Serie BJ06	Box-Jenkins	Red Neuronal	Espacio de Estados
90	0.950	0.950	0.950	0.950
91	0.970	1.101	0.899	1.133
92	0.780	1.048	0.730	0.769
93	0.600	0.871	0.893	0.594
94	0.450	0.662	0.608	0.462
95	0.480	0.492	0.353	0.358
96	0.310	0.397	0.332	0.531
97	0.170	0.381	0.188	0.141
98	0.080	0.422	0.010	0.163
99	0.380	0.491	0.133	0.124
100	0.750	0.558	0.859	0.698
<b>MSE</b>		0.042575	0.021297	0.016672

Tabla 4. Previsiones BJ06.

En todos los casos, se presenta una tabla con los datos numéricos de las previsiones calculadas con los tres métodos con el valor del error cuadrático medio (MSE) para cada uno de ellos, y un gráfico en el que se pueden comprobar las previsiones comparadas con los valores reales de las cuatro series.

	Serie UN17	Box-Jenkins	Red Neuronal	Espacio de Estados
150	5.4470	5.4470	5.4470	5.4470
151	5.4330	5.4420	5.4443	5.4272
152	5.4150	5.4550	5.4745	5.4467
153	5.4130	5.4840	5.5184	5.4302
154	5.4380	5.5240	5.5724	5.3974
155	5.4580	5.5720	5.6306	5.4161
156	5.4690	5.6250	5.6909	5.4688
157	5.4860	5.6810	5.7508	5.4771
158	5.5260	5.7390	5.8098	5.4777
159	5.5270	5.7970	5.8689	5.5018
160	5.5230	5.8560	5.9219	5.5704
161	5.5590	5.9150	5.9746	5.5020
162	5.6450	5.9730	6.0245	5.5204
163	5.7730	6.0310	6.0695	5.6150
164	5.8730	6.0890	6.1075	5.7448
165	5.9500	6.1460	6.1354	5.9191
166	5.9870	6.2030	6.1514	5.9428
167	6.0150	6.2600	6.1557	6.0084
168	6.0040	6.3150	6.1492	5.9942
169	6.0000	6.3710	6.1349	6.0336
170	5.9910	6.4270	6.1153	5.9795
171	6.0030	6.4830	6.0943	5.9898
172	6.0060	6.5390	6.0747	5.9788
173	6.0180	6.5950	6.0589	6.0246
174	6.0570	6.6510	6.0471	6.0150
175	6.1250	6.7070	6.0406	6.0373
176	6.1810	6.7630	6.0395	6.1122
177	6.2400	6.8190	6.0422	6.1965
178	6.3140	6.8740	6.0490	6.2248
<b>MSE</b>		0.1343668	0.0458639	0.00355394

Tabla 5. Previsiones UN17.

Como ya se ha comentado, se han calculado previsiones para dos series no estacionales y otras dos estacionales. En las tablas 4 y 5 y en las figuras 8 y 9 están representadas las previsiones de las series temporales no estacionales BJ06 y UN17 respectivamente. Observando el error cuadrático medio, se comprueba que los métodos de Redes Neuronales Artificiales y Espacio de Estados mejoran

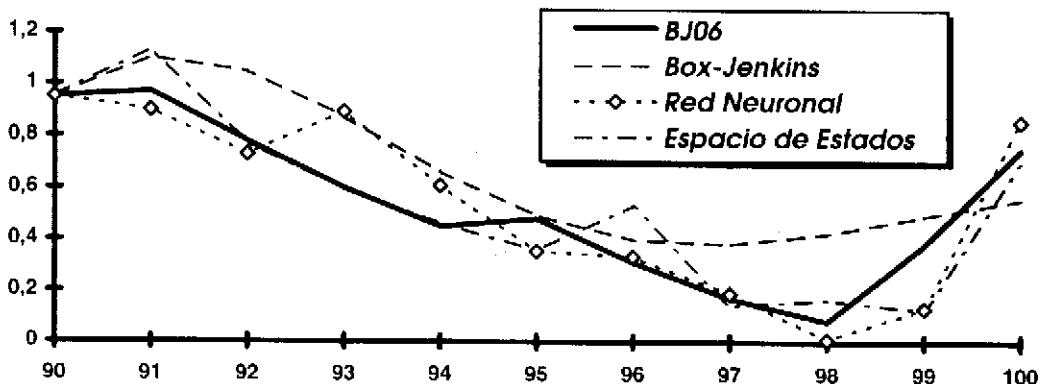


Figura 8.

significativamente los resultados obtenidos utilizando la metodología clásica de Box-Jenkins (sobre todo para UN17). Si observamos los gráficos de las figuras 8 y 9, podemos comprobar que, para BJ06, de los cuatro "turning points" que existen (en los periodos 91, 94, 95 y 98), Box-Jenkins detecta el primero y falla en los tres restantes, la red neuronal acierta sólo el último y además crea otros dos, en los periodos 92 y 93, lo cual puede ser un problema si lo que se pretende es precisamente buscar los "turning points" de la serie temporal. Por su parte, Espacio de Estados predice todavía mejor que la red neuronal en cuanto al error cuadrático medio, pero no se comporta con la misma exactitud a la hora de detectar los "turning points", se puede decir que excepto el del periodo 91, que sí lo detecta perfectamente, los otros tres los desplaza un periodo hacia adelante (de los periodos 94, 95 y 98, los coloca en 95, 96 y 99 respectivamente), además de calcular otros dos (97 y 98) que en la serie real no aparecen.

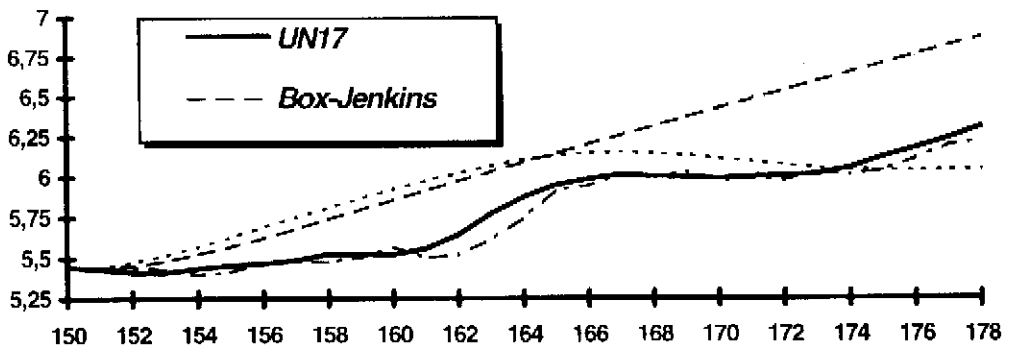


Figura 9.

Para la otra serie no estacional (UN17, figura 9), de nuevo la metodología de Espacio de Estados parece claramente superior a las otras dos, tanto en lo referente al error cuadrático medio como en el seguimiento de las pre- visiones a la serie original. En este caso, Box-Jenkins vuelve a comportarse peor que los otros dos métodos mientras que la red neuronal aunque produce unas previsiones mejores en cuanto al error cuadrático medio, se puede observar que en el periodo 168 cambia de tendencia y permanece con tendencia decreciente hasta el periodo 176, con lo que introduce dos "turning points" que en la serie real no existen.

	Serie BJ08	Box-Jenkins	Red Neuronal	Espacio de Estados
134	3.910	3.910	3.910	3.910
135	4.190	4.603	3.774	4.531
136	4.610	4.487	3.510	4.065
137	4.720	4.672	4.066	4.937
138	5.350	5.406	5.499	5.209
139	6.220	6.148	6.027	6.200
140	6.060	6.228	5.921	6.265
141	5.080	5.216	5.051	5.048
142	4.610	4.581	4.817	4.373
143	3.900	4.032	4.090	4.174
144	4.320	4.489	4.814	4.320
MSE		0.028963	0.22128132	0.065537092

Tabla 6. Previsiones BJ08.

En el caso de las series estacionales (BJ08 y UN05), cuyas previsiones aparecen en las tablas 6 y 7, y las representaciones gráficas en las figuras 10 y 11, se puede comprobar que la metodología clásica de Box-Jenkins da unos resultados significativamente mejores que los otros dos métodos para la serie UN05, y también algo mejores para la serie BJ08, sin embargo, de las representaciones gráficas de las figuras 10 y 11, se puede concluir que los tres métodos son

	Serie UN05	Box-Jenkins	Red Neuronal	Espacio de Estados
134	1269	1269	1269	1269
135	1231	1266	1262	1248
136	1351	1374	1337	1343
137	1337	1328	1319	1299
138	1367	1362	1348	1342
139	1381	1400	1379	1378
140	1249	1284	1250	1258
141	1231	1258	1232	1232
142	1364	1354	1397	1331
143	1464	1462	1434	1412
144	1326	1339	1320	1302
145	1333	1327	1276	1302
146	1345	1324	1316	1306
147	1281	1321	1301	1268
148	1396	1429	1399	1348
149	1397	1383	1400	1314
150	1443	1417	1375	1361
151	1402	1455	1401	1397
152	1293	1340	1296	1275
153	1321	1313	1414	1254
154	1472	1409	1498	1353
<b>MSE</b>		876.8500	1123.1787	2222.4500

Tabla 7. Previsiones UN05.

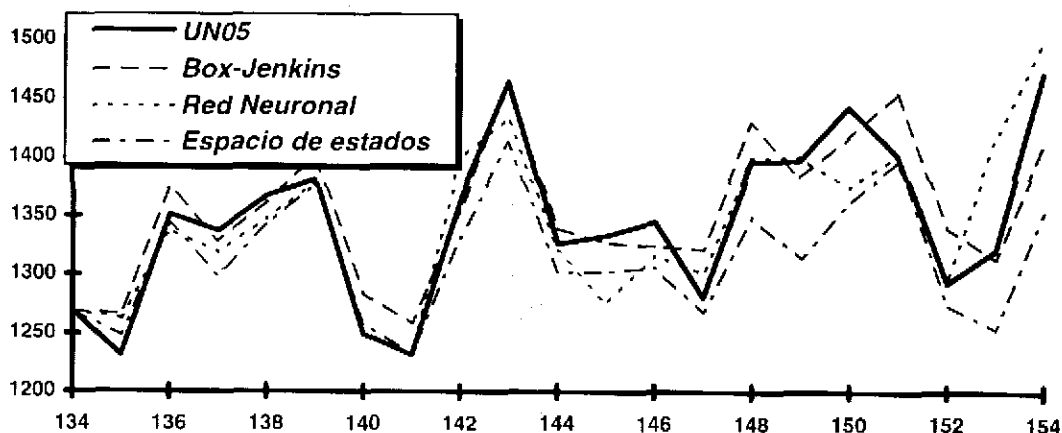


Figura 10.

igualmente aceptables a la hora de calcular previsiones, ya que todos detectan con bastante exactitud los puntos de cambio de tendencia (lo cual puede ser interesante en series temporales como la BJ08, en la que en el horizonte temporal utilizado, sólo hay dos "turning points", que son perfectamente calculados por la red neuronal (aunque la red neuronal introduce uno nuevo en el período 136, que en la serie real no existe), y "casi" por los otros dos métodos -en lugar de ponerlo en el período 139, lo colocan en el 140-). Es destacable en el caso de la serie BJ08, el hecho de que con la excepción de las tres primeras previsiones para la red neuronal, las dos de espacio de estados y la primera de Box-Jenkins, las demás previsiones se van ajustando



con bastante exactitud a la serie real. Este hecho también se da en la otra serie estacional, UN05, aunque en este caso las mejores previsiones se obtienen al principio.

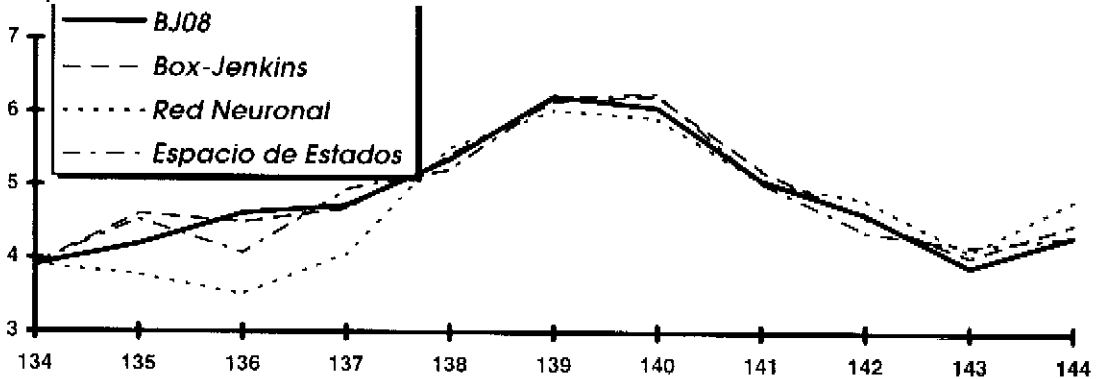


Figura 11.

## 6. Conclusiones

Como conclusiones finales, podemos decir que los resultados obtenidos utilizando los tres métodos: Box-Jenkins, Redes Neuronales Artificiales y Espacio de Estados, son bastante similares e igualmente aceptables en cuanto a la calidad de las previsiones que calculan. Sin embargo, hemos comprobado que las dos últimas metodologías (redes neuronales y espacio de estados) parecen mejorar ligeramente los resultados de Box-Jenkins en el caso de series temporales NO ESTACIONALES, mientras que precisamente la metodología de Box-Jenkins la que da los mejores resultados cuando la serie que se está tratando tiene el patrón de ESTACIONALIDAD.

Esto hace que la elección final del método de previsión a utilizar, dependa en definitiva de otros factores, como pueden ser el conocimiento más profundo de alguna de las tres técnicas o bien la relativa sencillez del aprendizaje y coste de aplicación para cada usuario en cuestión, de un método con respecto a los otros dos.

La mejor solución posiblemente sea no dejar de lado ninguno de los tres métodos, y utilizarlos conjuntamente de forma que los resultados obtenidos finalmente puedan ser validados por los otros dos métodos, de manera que se pueda asegurar que la metodología se ha aplicado correctamente y que no se han cometido errores que puedan hacer que las previsiones no sean correctas.

## 7. Bibliografía

- ALONSO, G., BECERRIL, J.L.(1993). "Introducción a la Inteligencia Artificial". Multimedia Ediciones S.A.
- AOKI, M. (1990). "State Space Modelling of Time Series". Springer-Verlag. Berlin.
- BOX, G.E.P., and JENKINS, G.M. (1976). "Time Series Analysis: Forecasting and Control". 2nd. Ed. San Francisco: Holden Day.
- CHAPMAN, A.J. (1994). "Stock market trading systems through neural networks: developing a model". Int. Journal of Applied Expert Systems. Vol. 12, N.2
- EBERHART, R.C. and DOBBINS, R.W. (1990). "Neural Network PC Tools". Academic Press.
- FARMER, J.D. and SIDOROWICH, J.J. (1987). "Predicting chaotic time series". Physical Review Letters, 59, 845-848.
- GRANGER, C.W.J. and NEWBOLD, P. (1986). "Forecasting economic time series" (2nd.ed.). Orlando, FL: Academic Press.
- HARVEY, A.C. (1989). "Forecasting time series models and the Kalman Filter" U.K.: Cambridge University Press.
- LIUNG, L. (1987). "System Identification. Theory for the User". Prentice Hall. Englewood Cliffs, New Jersey.
- KIMOTO, T., ASAKAWA, K., YODA, M. and TAKEOKA, M. (1990). "Stock market prediction system with modular neural networks". IJCNN Int. Joint Conference on Neural Networks, San Diego, CA.
- LAPEDES, A. and FARBER, R. (1987). "NonLinear signal processing using neural networks". IEEE Conference on Neural Information Processing System - Natural and Synthetic.
- LIPPMANN, R.P. (1987) "An Introduction to Computing with Neural Nets". IEEE ASSP Magazine. April 1987.
- LÜTKEPOHL, H. (1991). "Introduction to Multiple Time Series analysis". Springer-Verlag. Berlin.
- MITTNIK, S. (1989). "Multivariable Time Series Analysis with State Space Models", in S. Mittnik (ed.). System Theoretic Methods in Economic Modelling I. Oxford: Pergamon Press.
- MITTNIK, S.(1990). "Forecasting with Balanced State Space Representations of Multivariable Distributed Lag Models". Journal of Forecasting. Wiley Publishers. Vol. 9. pp. 207-218.
- MOORE, B.C.(1981). "Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction". IEEE Transactions in Automatic Control. AC-26. pp.17-32.
- NEUROSHELL (1989). "Neural Network Shell Program". Ward Systems Group Inc.
- OTTER, P. W. (1985). "Dynamic Feature Space Modelling, Filtering and Self-Turning Control of Stochastic Systems". Berlin: Springer-Verlag.

- OTTER, P. W. and VAN DAL, R. (1987). "State Space and Distributed Lag Modelling of dynamic Economic Processes Based on Singular Value Decomposition". *Anales D'Economie et de Statistique*. 6/7. pp. 253-277.
- PRIESTLEY, M.B. (1988). "Non-Linear and non-stationary time series analysis". San Diego, CA: Academic Press.
- REILLY, D.P. (1980). "Experiences with an Automatic Box-Jenkins Modeling Algorithm". in *Time Series Analysis*. Ed. O.D. Anderson. (Amsterdam: North Holland), pp. 493-508.
- RICHESON, L. and ZIMMERMANN, R.A. (1994) "Predicting consumer credit performance: can neural networks outperform traditional statistical methods?". *Int. Journal of Applied Expert Systems*. Vol.2, n.2, February, 1994.
- SALCHENBERGER, L.M., CINAR, E.M. and LASH, N.A. (1992) "Neural Networks: a new tool for predicting thrift failures". *Decision Sciences*, July-August, 1992.
- STRANG, G. (1980). "Linear Algebra". Academic Press. Inc. New York.
- SUAREZ RUESTRA, C. (1994). "Modelización y predicciones para la serie temporal del IPC, utilizando el Espacio de Estados Balanceado". *Estudios de Economía Aplicada*, Nº 2, 145-157.
- SUAREZ RUESTRA, C. (1994). "Sistemas Automáticos de Diseño en Espacio de Estado para la modelización de Series Temporales". Tesis Doctoral. U.N.E.D. Madrid.
- SUAREZ, C. y MAYO, J.L. (1994). "Predicciones usando un Espacio de Estado Balanceado. Un caso de estudio". VII Reunión Anual de ASEPELT. Palma de Mallorca.
- SUAREZ, C. y MAYO, J.L. (1994). "A New Procedure of Modelling VARMA Processes in Balanced State Space", *SYSID'94, 10th IFAC Symposium on System Identification* Copenhagen. Denmark.
- TIAO, G.C., BOX, G.E.P. (1983). "An Introduction to Applied Multiple Time Series Analysis". Illinois, USA: Scientific Computing Associated.
- TIAO, G.C., TSAY, R.S. (1989). "Model specification in multivariate time series". *Journal of the Royal Statistical Society: B* 51, 157-213.
- TONG, H. (1983). "Threshold models in non-linear time series analysis". *Lecture Notes in Statistic*, 21. New York: Springer-Verlag.
- TONG, H. (1990). "Non-Linear time series: A dynamical system approach". Oxford: Oxford University Press.
- WEY, W.W.S. (1990). "TIME SERIES ANALYSIS: Univariate and Multivariate Methods". California, USA. Addison-Wesley Publishing Company.
- ZAREMBA, T. (1990) "Technology in Search of a Buck". *Neural Network PC Tools*. Academic Press Inc.
- ZURADA, J. (1991) "Using Nworks: An Extended Tutorial for Neural Works Professional II/Plus and NeuralWorks Explorer". Pittsburg.

## ANEXO.

## Serie UN17

3.097	3.188	3.240	3.268	3.265	3.179	3.078	3.007	2.964	2.959
2.945	2.944	2.950	2.949	2.924	2.907	2.911	2.876	2.823	2.733
2.630	2.520	2.406	2.294	2.193	2.123	2.125	2.153	2.196	2.282
2.355	2.428	2.495	2.557	2.585	2.599	2.574	2.566	2.616	2.675
2.694	2.744	2.763	2.773	2.787	2.830	2.907	3.010	3.125	3.248
3.359	3.461	3.559	3.640	3.697	3.757	3.799	3.829	3.856	3.889
3.935	4.000	4.084	4.170	4.266	4.356	4.443	4.520	4.611	4.701
4.756	4.825	4.931	5.079	5.268	5.428	5.559	5.674	5.734	5.744
5.706	5.627	5.533	5.466	5.415	5.403	5.431	5.466	5.502	5.520
5.522	5.517	5.527	5.561	5.607	5.614	5.600	5.535	5.463	5.398
5.320	5.245	5.213	5.178	5.126	5.109	5.119	5.128	5.129	5.142
5.160	5.222	5.343	5.525	5.713	5.887	6.046	6.190	6.280	6.315
6.316	6.279	6.210	6.121	6.049	5.999	5.992	6.004	6.016	5.982
5.929	5.866	5.782	5.700	5.619	5.524	5.430	5.383	5.415	5.494
5.569	5.620	5.622	5.598	5.589	5.590	5.575	5.532	5.476	5.447
5.433	5.415	5.413	5.438	5.458	5.469	5.486	5.526	5.527	5.523
5.559	5.645	5.773	5.873	5.950	5.987	6.015	6.004	6.000	5.991
6.003	6.006	6.018	6.057	6.125	6.181	6.240	6.314		

## Serie UN05

718	724	778	865	688	792	848	810	934	822
962	936	985	1126	1005	1204	981	1076	1241	1039
1233	1170	1195	1215	1211	1309	1407	1351	1231	1345
1402	710	638	696	834	678	812	797	853	929
910	1067	864	997	1125	958	1159	997	1090	1100
1084	1261	1212	1295	1173	1227	1346	1284	1337	1364
1281	1293	762	656	735	779	833	909	711	875
915	879	1005	889	999	1000	994	1202	1042	1225
1004	1147	1299	1119	1262	1226	1219	1230	1272	1367
1464	1396	1321	839	621	730	740	762	850	765
906	865	902	1008	980	1146	924	1001	1210	1003
1190	1032	1162	1199	1161	1277	1291	1324	1203	1269
1381	1326	1397	1472	800	577	781	729	796	843
863	1002	783	925	1000	940	1109	957	1020	1060
1059	1200	1117	1274	1124	1222	1313	1184	1276	1299
1231	1249	1333	1443	1574					