UNIVERSIDAD DE LOS ANDES
FACULTAD DE CIENCIAS ECONÓMICAS Y SOCIALES
INSTITUTO DE ESTADÍSTICA APLICADA Y COMPUTACIÓN

PREDICCIÓN CON REDES NEURONALES: COMPARACIÓN CON LAS METODOLOGÍAS DE BOX Y JENKINS

Autora
Joanna Verónica Collantes Duarte

Tutor
Prof. Giampaolo Orlandoni Merli

Jurado
Prof. Gerardo Colmenares La Cruz
Prof. Luciano Maldonado

TRABAJO DE GRADO
Presentado ante la
UNIVERSIDAD DE LOS ANDES
En cumplimiento parcial de los requisitos
para optar al título de
MAGISTER SCIENTIAE EN ESTADÍSTICA APLICADA

Mérida, Venezuela
2001

RESUMEN

El objetivo principal de esta investigación es comparar las metodologías de Box y Jenkins, ARIMA y Función de Transferencia, utilizadas frecuentemente en estadística para predicción con series de tiempo, con una técnica de la Inteligencia Artificial denominada Redes Neuronales (RN).

Las nociones básicas de análisis de series de tiempo en el campo estadístico se exponen brevemente. Así como también se reseñan los conceptos fundamentales de Redes Neuronales que permitan diseñar RN adecuadas para predicción.

Una metodología para predicción con redes neuronales es propuesta. Ésta pretende servir de guía para tener éxito en la predicción con series de tiempo.

La comparación planteada se basa en el estudio de dos aplicaciones, la primera, es la serie del número de nacimientos mensuales ocurridos en España. A esta serie se le aplica la metodología ARIMA y RN para predicción (caso univariable). El modelo ARIMA proporciona un mejor ajuste, pero la RN logra mejores predicciones.. La segunda aplicación comprende dos series: el gasto de publicidad mensual (en miles de $) y el número de ventas mensuales (en miles de casos). Se desea predecir el número de ventas en base al gasto de publicidad. En este caso se aplica el Modelo de Función de Transferencia y RN para predicción (caso bivariable). La RN resulta ser muy superior al MFT, tanto en el ajuste como en la predicción.

El punto más álgido en el diseño del modelo de la red son las entradas o retrasos de la serie. En forma empírica, se llegó a la conclusión de que puede utilizarse la metodología ARIMA como una herramienta de preprocesamiento de datos, considerando como entradas los retrasos involucrados en el modelo proporcionado por esta metodología. En el caso del MFT no podemos llegar a la misma conclusión, seguramente, porque en el ejemplo estudiado no se logra captar el patrón de comportamiento del proceso. Sin embargo, herramientas muy útiles resultaron ser los correlogramas simple y parcial, y el gráfico de correlación cruzada para identificar los retrasos de interés candidatos a ser entradas.

A mi hija,

mi esposo

y mis padres

con todo mi amor

AGRADECIMIENTO

Al tutor Prof. Giampaolo Orlandoni Merli, por guiar pacientemente esta investigación y estar dispuesto a compartir sus conocimientos.

Al Prof. Gerardo Colmenares La Cruz, por sus aportes y siempre oportunas correcciones.

Al Prof. Francklin Rivas Echeverría, por sus valiosas ideas y explicaciones.

Al Lic. Milton Quero, por las sugerencias metodológicas y por tener siempre una palabra de aliento.

A mi esposo Hussein El Fakih Rodríguez, por su colaboración e inconmensurable paciencia.

A la Virgen de Coromoto. Gracias.

INDICE

INDICE DE FIGURAS

CAPÍTULO I

INTRODUCCIÓN

Este capítulo introductorio establece que el enfoque de este estudio es de tipo comparativo, entre las técnicas estadísticas de Box y Jenkins, utilizadas en predicción con series de tiempo y la técnica computacional, derivada de la Inteligencia Artificial, denominada Redes Neuronales. Así como también, pretende transmitir la esencia del trabajo realizado, sus objetivos, su importancia, sus alcances y limitaciones.

1.1 PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA

Planteamiento del Problema

La predicción basada en series de tiempo es de gran interés práctico, pues permite conocer con un margen de error valores futuros de una serie basándose en sus valores pasados (caso univariable), lo cual podría facilitar la toma de decisiones y la planificación. En ocasiones existe una relación causal entre dos series, en este caso valores pasados de una serie ayudan a predecir valores futuros de otra serie (caso bivariable).

La estadística clásica para poder predecir utiliza, con mucha frecuencia, las metodologías de Box y Jenkins (Box, Jenkins y Reinsel, 1994). Estas metodologías requieren del criterio de un experto, los pasos a realizar dependen del tipo de datos, estos datos deben ser estacionarios, se analizan los gráficos de correlaciones, se evalúa la adecuación del modelo, se mide el error de predicción, etc. Estos procedimientos pueden resultar complejos, lo cual nos conduce a intentar dar solución a este problema mediante las nuevas técnicas computacionales que han tomado auge en los últimos años.

La Inteligencia Artificial ha surgido como una nueva área del conocimiento. Está formada por un conjunto de técnicas que intentan imitar, en forma artificial, las habilidades relacionadas con la inteligencia humana. Una de estas técnicas que trata de simular el pensamiento humano mediante conexión de neuronas es la denominada "Redes Neuronales", utilizada recientemente con un relativo éxito para la predicción de series de tiempo. Pueden citarse algunos trabajos en el área como los realizados por: Wong 1991, Hill et al 1996, Wedding II y Cios 1996, Faraway y Chatfield 1998, entre otros.

Formulación del Problema

¿Cuál es la capacidad predictiva de las Redes Neuronales en comparación con las Metodologías de Box y Jenkins?

1.2. OBJETIVOS

Objetivo General

- Comparar los métodos de predicción: Metodologías de Box y Jenkins y Redes Neuronales.

Objetivos Específicos

- Diseñar Redes Neuronales para que realicen procesos de predicción univariable y bivariable.
- Predecir valores de series de tiempo utilizando los métodos: Redes Neuronales y las Metodologías de Box y Jenkins: ARIMA y Función de Transferencia.
- Analizar y comparar los resultados obtenidos con las diferentes metodologías para predicción, utilizando como criterios comparativos los errores de ajuste y predicción.

1.3. ALCANCES Y LIMITACIONES

Con esta investigación se pretende analizar en forma comparativa y mediante casos de estudio, las Metodologías de Box y Jenkins, usadas ampliamente en estadística para la predicción con series de tiempo, y una técnica de la Inteligencia Artificial denominada Redes Neuronales. Un objetivo es proponer ciertos diseños de redes neuronales que pudieran ser útiles al momento de predecir, tanto en el caso de disponer de una sola serie de tiempo (caso univariable), como en el caso de la existencia de otra serie de tiempo que tenga una relación causal con la primera y que ayude a mejorar su predicción (caso bivariable).

El desarrollo de esta investigación se presenta mediante seis capítulos: El primero es un capítulo introductorio, donde se plantea el problema, objetivos, alcances y limitaciones de la investigación. El segundo capítulo proporciona las nociones básicas de series de tiempo, en el ámbito estadístico, y se hace énfasis en las Metodologías de Box y Jenkins: Metodología ARIMA que permite predecir valores de una serie de tiempo en base a sus valores pasados (caso univariable), y la metodología de Función de Transferencia que proporciona valores futuros de una serie de tiempo en base a sus valores pasados y a otra serie con la cual se tiene una relación causal (caso bivariable). El tercer capítulo introduce al área de Redes Neuronales, se definen y establecen los aspectos más importantes y por último, se exponen varios modelos de redes usados para predicción por diversos autores. En el cuarto capítulo se plantea una metodología para construir los modelos de redes neuronales que se utilizarán para la predicción, tanto en el caso univariable como bivariable. Se establece la forma de medir el error, lo cual se utilizará como criterio comparativo entre las Metodologías de Box y Jenkins y las Redes Neuronales. El quinto capítulo se dedica, primero, a la aplicación de la Metodología ARIMA y a la construcción del modelo de Red Neuronal Univariable

para la serie de tiempo del número de nacimientos mensuales ocurridos en España entre enero de 1960 y diciembre de 1999; y segundo, a la aplicación de la técnica de Función de Transferencia y a la construcción del modelo de Red Neuronal Bivariable para las series de gastos mensuales de publicidad y número de ventas mensuales de cierta empresa. En cada uno de los casos se discuten los resultados obtenidos según los criterios de comparación establecidos. Por último, en el sexto capítulo se plantean las conclusiones, así como algunas recomendaciones para el camino a explorar en esta línea de investigación.

Este trabajo no pretende establecer la superioridad de una metodología sobre otra, sino de evaluar, mediante ejemplos, la posibilidad de predecir utilizando Redes Neuronales, y discutir las ventajas y desventajas de éstas comparadas con las metodologías de Box y Jenkins, utilizadas comúnmente en estadística.

CAPÍTULO II

SERIES DE TIEMPO: METODOLOGÍAS DE BOX Y JENKINS

Muchos datos se almacenan en forma de series de tiempo, mensuales, trimestrales, anuales y es de gran importancia para las empresas u organismos poder pronosticar sus valores para la planificación y la toma de decisiones.

Un método frecuentemente utilizado en la estadística clásica para la predicción con series de tiempo de una variable es el descrito por Box y Jenkins (Box, Jenkins y Reinsel, 1994), basado en el proceso Autorregresivo Integrado de Promedio Móvil: ARIMA (de sus siglas en inglés, "Autoregressive Integrated Moving Average"). En el marco de las Metodologías de Box y Jenkins se consideran los modelos de función de transferencia para predecir valores de una serie de tiempo en base a valores pasados de esa serie y de otras series que tienen una relación causal con ésta. En este capítulo se estudiarán las nociones básicas de series de tiempo, así como éstos métodos de predicción mencionados.

2.1. NOCIONES BÁSICAS DE SERIES DE TIEMPO Y PREDICCIÓN

2.1.1. Definición de Serie de Tiempo

De acuerdo a Bowerman y O'Connel, 1993, una serie de tiempo es una secuencia cronológica de observaciones de una variable particular.

Conseguimos series de tiempo en los distintos campos del saber: en economía, mercadeo, demografía, meteorología, ingeniería, etc. Muchos son los ejemplos de series de tiempo que podrían citarse, tales como:

- Las ventas mensuales de una empresa en la última década.
- El número de automóviles producidos por año de determinada marca en el periodo 1985-2000.
- La temperatura diaria promedio en los últimos 6 meses.

2.1.2. Objetivos del Análisis de Series de Tiempo

Según Chatfield, 1978, son varios los objetivos por los cuales se puede querer analizar una serie de tiempo:

- Descripción: Cuando tenemos una serie de tiempo, el primer paso en el análisis es graficar los datos y obtener medidas descriptivas simples de las propiedades principales de la serie.
- Explicación: Cuando las observaciones son tomadas sobre dos o más variables, es posible usar la variación en una serie para explicar la variación en otras series.
- Predicción: Dada una serie de tiempo se puede querer predecir los valores futuros de la serie. Este es el objetivo más frecuente en el análisis de series de tiempo.
- Control: Cuando una serie de tiempo se genera por mediciones de calidad de un proceso, el objetivo del análisis puede ser el control del proceso.

2.1.3. Componentes de una Serie de Tiempo

Una serie de tiempo frecuentemente es examinada con la intención de descubrir patrones históricos que puedan ser útiles en la predicción. Para identificar estos patrones es

conveniente pensar que una serie de tiempo consiste de varios componentes (Bowerman y O'Conell, 1993):

1. Tendencia: Una serie de tiempo tiene tendencia cuando por largos periodos los valores crecen o decrecen. También puede definirse como cambios en la media.

2. Ciclos: Se refiere a movimientos hacia arriba y hacia abajo alrededor del nivel de la tendencia. Estas fluctuaciones, medidas de pico a pico, pueden tener una duración larga.

3. Variaciones Estacionales: Son patrones periódicos que ocurren y se repiten cada determinado tiempo, por ejemplo: anualmente. Estas variaciones son usualmente causadas por factores como el clima y las costumbres.

4. Fluctuaciones Irregulares: Son movimientos erráticos en una serie de tiempo que no siguen un patrón regular, ni reconocible. Tales movimientos representan "lo que queda" en una serie de tiempo después de que la tendencia, ciclos y variaciones estacionales han sido explicadas.

2.1.4. Métodos de Predicción

Pueden obtenerse valores futuros de una serie de tiempo observada mediante una gran variedad de métodos de predicción. Estos métodos pueden clasificarse fundamentalmente en tres tipos:

1. Subjetivo: Las predicciones se hacen sobre bases subjetivas usando el criterio, la intuición, el conocimiento en el área y otra información relevante (Chatfield, 1978). Entre estos métodos están: Ajuste de una curva subjetiva, el Método Delphi y comparaciones tecnológicas en tiempo independiente. Ninguno de estos métodos se considerará en este estudio.

2. Univariado: Este tipo de método obtiene valores futuros de la serie basándose en el análisis de sus valores pasados, se intenta conseguir un patrón en estos datos, se asume que este patrón continuará en el futuro y se extrapola para conseguir tales predicciones. Son muchos los métodos que encajan en esta categoría, entre éstos se encuentran: Extrapolación de curvas de tendencia, suavización exponencial, método de Holt-Winters y método de Box y Jenkins (ARIMA). Este

último es un método ampliamente utilizado y es en el que centraremos nuestro interés.

3. Causal o Multivariado: Involucra la identificación de otras variables que están relacionadas con la variable a predecir. Una vez que esas variables han sido identificadas, se desarrolla un modelo estadístico que describe la relación entre esas variables y la variable a predecir. La relación estadística derivada es entonces usada para predecir la variable de interés (Bowerman y O'Connell, 1993). Entre estos métodos están: Regresión múltiple, modelos econométricos y método de Box y Jenkins (Modelo de Función de Transferencia). Este último método es una extensión del modelo ARIMA que consiste en describir la relación entre la variable de entrada y la variable de salida. Aunque el método puede generalizarse para varias variables de entrada, nos concentraremos únicamente en el caso bivariable (una variable de entrada y una de salida).

2.2. MÉTODOS DE BOX Y JENKINS

Se estudiarán dos metodologías desarrolladas por Box y Jenkins que permiten predecir valores futuros de una serie de tiempo basándose en valores pasados de una sola variable o dos variables entre las que existe una relación causal.

2.2.1. ARIMA: Modelo Univariable

El proceso Autorregresivo Integrado de Promedio Móvil: ARIMA (de sus siglas en inglés, "Autoregressive Integrated Moving Average") es denominado también Método (Univariante) de Box y Jenkins.

Este enfoque de Box y Jenkins es una de las metodologías de más amplio uso para el análisis de series de tiempo. Es popular debido a su generalidad y cuenta con programas de computación bien documentados. Si bien Box y Jenkins no fueron los creadores ni quienes contribuyeron de manera más importante en el campo de los modelos ARMA, si fueron quienes los popularizaron y los hicieron accesibles (Maddala, 1996).

La metodología de Box y Jenkins requiere que la serie sea estacionaria o que sea estacionaria después de una o más diferenciaciones.

Una serie de tiempo es estacionaria, si su media, su varianza y su covarianza (en los distintos rezagos) permanecen constantes sin importar el momento en el cual se midan

(Gujarati, 1997). Para el tratamiento de la no estacionariedad en la media se propone la diferenciación sucesiva de la serie, aprovechando la propiedad de que gozan una gran parte de los procesos estocásticos, de convertirse en estacionarios al diferenciarlos cierto número de veces. Las primeras diferencias de los valores de la serie de tiempo $y_1, y_2, ..., y_n$ son:

$$z_t = y_t - y_{t-1} \quad \text{donde } t = 2,3,...,n \tag{2.1}$$

Las segundas diferencias son:

$$z_t = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \quad \text{para } t = 3,4,...,n \tag{2.2}$$

La metodología de Box y Jenkins aplica métodos autorregresivos, promedio móvil, autorregresivos y de promedio móvil y, autorregresivo integrado de promedio móvil. A continuación se explicarán brevemente cada uno de estos procesos. Las series de tiempo se suponen en adelantes estacionarias a menos que se especifique lo contrario. Considérese $Y_t = y_t - \mu$, donde y_t son los valores originales de la serie de tiempo, μ es la media de todos los valores de la serie, y Y_t es la desviación del proceso respecto a la media. ε_t es una perturbación aleatoria o ruido blanco (con media cero, varianza constante y covarianza cero). Los diferentes procesos son:

a) Proceso Autorregresivo (AR)

Definición: Al modelo

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p} + \varepsilon_t \tag{2.3}$$

donde p denota el número de términos autorregresivos y $\phi_1, \phi_2, ..., \phi_p$ es un conjunto finito de pesos o parámetros, se le denomina proceso autorregresivo de orden p o AR(p). Los términos ϕ son coeficientes determinados por regresión lineal. Esos coeficientes son multiplicados por los p valores previos de la serie. Este modelo relaciona el valor pronóstico de Y_t con la suma ponderada de sus valores en periodos pasados, más una perturbación aleatoria en el tiempo t. Equivalentemente, y haciendo uso del operador retardo B, que funciona de la siguiente manera $BY_t = Y_{t-1}$, un proceso autorregresivo puede expresarse como:

$$(1 - \phi_1 B - \phi_2 B^2 - ... - \phi_p B^p) Y_t = \varepsilon_t \tag{2.4}$$

o en forma abreviada

$$\phi_p(B) Y_t = \varepsilon_t \tag{2.5}$$

b) Proceso de Promedio Móvil (MA)

Definición: Y_t podría ser generado por el modelo:

$$Y_t = \varepsilon_t - \theta_1\varepsilon_{t-1} - \theta_2\varepsilon_{t-2} -\ldots- \theta_q\varepsilon_{t-q} \quad (2.6)$$

donde q denota el número de términos de promedio móvil y $\theta_1, \theta_2, \ldots, \theta_q$, son el conjunto finito de pesos o parámetros. A este modelo se le denomina proceso de promedio móvil de orden q, o MA(q). Los términos θ son coeficientes determinados mediante métodos iterativos no lineales y son multiplicados por los q errores de predicción previos. En este proceso se relaciona el valor pronóstico a los errores de predicciones previas.

El modelo MA(q) también puede escribirse equivalentemente como:

$$Y_t = (1- \theta_1B - \theta_2B^2 -\ldots- \theta_qB^q)\ \varepsilon_t \quad (2.7)$$

o en forma abreviada

$$Y_t = \theta_q(B)\varepsilon_t \quad (2.8)$$

c) Proceso Autorregresivo y de Promedio Móvil (ARMA)

Se sabe que un proceso de promedio móvil finito (ver Box, Jenkins y Reinsel, 1994)

$$Y_t = \varepsilon_t - \theta_1\varepsilon_{t-1} \qquad |\theta| < 1$$

Puede ser escrito como un proceso autorregresivo infinito

$$Y_t = \varepsilon_t - \sum_{i=1}^{\infty} \theta_1^i Y_{t-i}$$

Por lo tanto, si el proceso fuera realmente MA(1), se podría obtener una representación no parsimoniosa en términos de un modelo autorregresivo. Contrariamente, un AR(1) no puede ser representado usando un proceso de promedio móvil. En la práctica, para obtener una parametrización parsimoniosa, algunas veces es necesario incluir ambos términos, autorregresivo y promedio móvil. Así,

$$Y_t = \phi_1Y_{t-1} + \phi_2Y_{t-2} +\ldots+\phi_pY_{t-p} + \varepsilon_t - \theta_1\varepsilon_{t-1} - \theta_2\varepsilon_{t-2} -\ldots- \theta_q\varepsilon_{t-q} \quad (2.9)$$

o en forma abreviada

$$\phi_p(B)Y_t = \theta_q(B)\varepsilon_t \quad (2.10)$$

es llamado *proceso autorregresivo y de promedio móvil de orden (p,q) o ARMA(p,q).* (Box, Jenkins y Reinsel, 1994).

d) Proceso Autorregresivo Integrado de Promedio Móvil (ARIMA)

Los modelos de series de tiempo analizados están basados en el supuesto de que las series de tiempo consideradas son estacionarias (media y varianza constantes). Pero se sabe que muchas series de tiempo son no estacionarias, es decir, son integradas.

Si se debe diferenciar una serie de tiempo d veces para hacerla estacionaria y luego aplicar a ésta el modelo ARMA(p,q), se dice que la serie de tiempo original sigue un *proceso autorregresivo integrado de promedio móvil o ARIMA(p,d,q),* donde p denota el número de términos autorregresivos, d el número de veces que la serie debe ser diferenciada para hacerse estacionaria y q el número de términos de promedio móvil. (Gujarati, 1997).

El modelo ARIMA(p,d,q) puede ser escrito como:

$$\phi_p(B)(1-B)^d Y_t = \theta_q(B)\varepsilon_t \qquad (2.11)$$

Los procesos ARIMA son suficientes para explicar procesos con tendencia pero incapaces de representar procesos con estacionalidad y se hace necesaria una generalización de estos para lograr explicar los comportamientos estacionales. Los modelos estacionales consideran los retrasos del proceso y de la perturbación aleatoria periódicamente, es decir, cada s periodos. Por ejemplo, si los datos son mensuales es lógico considerar el periodo s=12. El objeto de estos retardos estacionales (s) es explicar la dependencia que tienen entre sí iguales periodos de años sucesivos, por ejemplo, marzo del 94 con marzo del 93 y marzo del 92 directamente y a través de los errores (perturbaciones no explicadas) asociadas a estos períodos.

Los modelos estacionales se denotan anteponiéndoles la letra S y el orden de sus parámetros se escribe con mayúscula, como sigue: modelos autorregresivos estacionales SAR(P), promedios móviles estacionales SMA(Q) y, autorregresivo y de promedios móviles estacionales SARMA(P,Q). Los modelos SARMA son análogos al proceso ARMA pero considerando los retardos del ruido blanco y del proceso de s en s.

Sin embargo estos modelos SARMA no son capaces de explicar todos los movimientos estacionales, pues si estos crecieran de año en año, los SARMA serian incapaces de recoger esta evolución pues al igual que los ARMA son estacionarios. Esta dificultad se resuelve a través de los modelos autorregresivos de promedios móviles integrados estacionales SARIMA(P,D,Q).

La unión de modelos estacionales con modelos no estacionales conduce a un modelo de gran capacidad de adaptación que puede reflejar tanto la tendencia como la estacionalidad de una serie (enfoque de Box y Jenkins). La combinación de estos modelos se logra a través de la multiplicación de los operadores polinomiales que caracterizan a cada modelo obteniendo los modelos conocidos como ARIMA(p,d,q)xSARIMA(P,D,Q). La notación que emplearemos para esta combinación de modelos es en términos generales un ARIMA(p,d,q)x$(P,D,Q)_s$, donde los parámetros no considerados en el modelo seleccionado serán de orden cero :

- p : orden de un modelo autorregresivo AR
- d : orden de diferenciación en la parte regular o no estacionaria de la serie
- q : orden de un modelo promedio móvil MA
- P : orden de un modelo autorregresivo estacional SAR
- D : orden de diferenciación en la parte estacional de la serie
- Q : orden de un modelo promedio móvil estacional SMA

El modelo ARIMA(p,d,q)x$(P,D,Q)_s$ puede escribirse como:

$$\phi_P(B^s)\ (1-B^s)^D\ \phi_p(B)(1-B)^d Y_t = \theta_Q(B^s)\ \theta_q(B)\varepsilon_t \qquad (2.12)$$

como $Y_t = y_t - \mu$, entonces el modelo puede expresarse como:

$$\phi_P(B^s)\ (1-B^s)^D\ \phi_p(B)(1-B)^d y_t = \delta + \theta_Q(B^s)\ \theta_q(B)\varepsilon_t \qquad (2.13)$$

donde δ es una constante.

Herramientas para la determinación de los órdenes del modelo

El número de términos en la parte AR y MA del modelo final no son escogidos arbitrariamente. Para esto, Box y Jenkins proporcionan un método estructurado que determina cual modelo ajusta mejor a la serie en cuestión y recomiendan que el modelo se mantenga tan simple como sea posible (principio de parsimonia). En general, no hay más de tres términos AR o MA. (Wedding y Cios, 1996). La herramienta principal para determinar los órdenes del modelo son los correlogramas simple y parcial, que son la representación gráfica de las funciones de autoccorrelación simple y parcial, las cuales se explican brevemente a continuación:

a) Función de Autocorrelación Simple

Considere la serie de tiempo $Y_1, Y_2,..., Y_n$. La autocorrelación simple muestral en el retraso k, denotada por r_k, es:

$$r_k = \frac{\sum_{t=1}^{n-k} (Y_t * Y_{t+k})}{\sum_{t=1}^{n} Y_t^2} \qquad (2.14)$$

Esta cantidad mide la relación lineal entre las observaciones de la serie de tiempo separadas por un retraso de k unidades de tiempo. La autocorrelación r_k toma valores entre –1 y 1. Para conocer si el valor de r_k es significativo se utiliza el estadístico t_{r_k} :

$$t_{r_k} = \frac{r_k}{s_{r_k}}$$

donde s_{r_k} es el error estándar de r_k, dado por:

$$s_{r_k} = \frac{\left(1 + 2\sum_{j=1}^{k-1} r_j^2\right)^{1/2}}{n^{1/2}}$$

La función de autocorrelación simple es el conjunto de autocorrelaciones simples muestrales en los retrasos k=1,2,....; a la representación gráfica de estas autocorrelaciones se le denomina correlograma simple. (Bowerman y O'Connell, 1993).

b) Función de Autocorrelación Parcial

La autocorrelación parcial muestral en el retraso k es:

$$r_{kk} = \begin{cases} r_1 & \text{si } k=1 \\ \dfrac{r_k - \sum_{j=1}^{k-1} r_{k-1,j} r_{k-j}}{1 - \sum_{j=1}^{k-1} r_{k-1,j} r_j} & \text{si } k=2,3,... \end{cases} \qquad (2.15)$$

donde $r_{kj} = r_{k-1,j} - r_{kk} r_{k-1,k-j}$ para j=1,2,...,k-1

El valor de estas autocorrelaciones puede pensarse intuitivamente como la relación de las observaciones de la serie de tiempo separadas por un retraso de k unidades de tiempo, eliminando el efecto de las observaciones intermedias.

Para conocer si el valor de r_{kk} es significativo se utiliza el estadístico $t_{r_{kk}}$:

$$t_{r_{kk}} = \frac{r_{kk}}{s_{r_{kk}}}$$

donde $s_{r_{kk}}$ es el error estándar de r_{kk}, dado por:

$$s_{r_{kk}} = \frac{1}{n^{1/2}}$$

La función de autocorrelación parcial es el conjunto de las autocorrelaciones parciales muestrales en los retrasos k=1,2,....; a la representación gráfica de estas autocorrelaciones se le denomina correlograma parcial. (Bowerman y O'Connell, 1993).

Etapas del Método Univariado de Box y Jenkins o Métodología ARIMA

Para construir un modelo ARIMA que recoja suficientemente bien las características de la serie se hace uso de la metodología de Box y Jenkins que puede ser estructurada en cinco etapas:

1. Análisis Exploratorio de la Serie

Consiste en un primer análisis, mediante gráficos y pruebas estadísticas para obtener estacionariedad (media y varianza constante)

a) Se grafica la serie a través del tiempo, de manera de observar a priori sus componentes: tendencia, estacionalidad y ciclos. Podría notarse la necesidad de aplicar diferencias, en la parte no estacional o regular, para hacer que la media sea constante. Podría observarse estacionalidad mediante una pauta repetida de acuerdo con el periodo estacional s, lo que implicaría la necesidad de diferencias en la parte estacional. Esto se confirmaría en la segunda etapa mediante los correlogramas.

b) Se realiza un diagrama de caja (por año) que permita estudiar el comportamiento de la varianza. Al sospechar que la varianza no es constante, es conveniente aplicar la prueba de Levene. Si el estadístico es significativo, se rechaza la hipótesis nula de homocedasticidad y será necesario realizar alguna transformación a la serie. Entre las más utilizadas están las transformaciones de Box-Cox. Si λ es el estadístico de Box-Cox, entonces la serie y_t se transforma de la siguiente manera:

$$y_t^{(\lambda)} \begin{cases} = \dfrac{(y_t^{\lambda} - 1)}{\lambda} & \text{para } \lambda \neq 0 \\ = \ln y_t & \text{para } \lambda = 0 \end{cases} \qquad (2.16)$$

Obsérvese que para λ=1, la transformación apenas si tiene influencia sobre los valores originales. El valor que más se suele utilizar es el de λ=0 porque simplemente consiste en tomar logaritmos neperianos de los valores originales y con frecuencia es una transformación exitosa. Podría realizarse nuevamente la Prueba de Levene para verificarlo.

2. Identificación del Modelo

En esta etapa se debe sugerir un conjunto reducido de posibles modelos.

a) Selección del conjunto de estimación: conjunto de datos que se usará para la estimación y adecuación del modelo; y del conjunto de predicción: conjunto de datos que se guardará para evaluar las predicciones.

b) Determinación de los correlogramas o funciones de autocorrelación simple y parcial para establecer, conjuntamente con lo observado en la primera etapa, el número de diferencias que se aplicarán y que convertirán el proceso en estacionario. Una serie no es estacionaria en la parte regular si en el correlograma simple se observa que los valores decrecen lentamente en los retardos 1,2,...

c) Determinación de los órdenes del componente autorregresivo (p) y promedio móvil (q) del modelo ARMA (p,q), haciendo uso de los patrones que se observan en los correlogramas simple y parcial:

Correlograma Simple	Correlograma Parcial	Modelo
Decae lentamente	Se corta después del retardo p	AR(p)
Se corta después del retardo q	Decae lentamente	MA(q)
Decae lentamente	Decae lentamente	ARMA(p,q)

d) Estudio de la estacionalidad. En caso de presentar estacionalidad con periodo s, se aplica una diferencia estacional $(1-B)^s$ para convertir la serie en estacionaria. La estacionalidad se manifiesta en el gráfico de la serie (etapa 1) y en el correlograma simple que presentará valores positivos que decrecen lentamente en los retardos s, 2s, 3s,...

e) Determinación de los órdenes P y Q del proceso SARMA(P,Q), de la misma manera que en la parte regular, pero, considerando solamente los valores de los correlogramas en los retardos s,2s,3s,...

f) Especificación del modelo ARIMA identificado y se sugieren otros modelos posibles.

3. Estimación de los Parámetros del Modelo

Una vez identificado el modelo,

$$\phi_P(B^s)\,(1-B^s)^D\,\phi_p(B)(1-B)^d Y_t = \theta_Q(B^s)\,\theta_q(B)\varepsilon_t \qquad (2.17)$$

los valores de los parámetros ϕ y θ se estiman mediante la minimización de la suma de cuadrado de los errores ε_t. Esto se hace utilizando el método de mínimos cuadrados condicional o el método de mínimos cuadrados incondicional. Generalmente estas estimaciones se hacen con la ayuda de herramientas computacionales como SPSS, SAS, etc.

4. Adecuación del Modelo

Una idea, a priori, de la adecuación del modelo se logra graficando la serie original y la ajustada por el modelo ARIMA contra el tiempo, de manera que se puedan observar sus similitudes y diferencias.

Un modelo ARIMA es adecuado para representar el comportamiento de una serie si se cumple lo siguiente:

a) Los residuos, diferencia entre el valor original de la serie y el valor estimado por el modelo, se aproximan al comportamiento de un ruido blanco (media cero, varianza σ^2 y covarianza cero). Al observar los correlogramas no deberán observarse valores significativamente diferentes de cero, como indicativo de ausencia de correlación serial, así como tampoco patrones (tendencia, ciclos) que indicarían que el modelo no extrajo toda la información posible.

b) Los parámetros del modelo ARIMA seleccionado son significativamente diferentes de cero.

c) Los parámetros del modelo están poco relacionados entre sí.

d) El grado de ajuste es elevado en comparación al de otros modelos alternativos. La bondad del ajuste puede evaluarse con la Desviación Estándar Residual (DER), Criterio de Información de Akaike (AIC), y con el Criterio Bayesiano de Schwarz (SBC), entre otros. A continuación se da una breve explicación de cada uno (Faraway y Chatfield, 1998):

- Desviación Estándar Residual (DER): Su expresión matemática es:

$$\hat{\sigma} = \sqrt{S/(T - r)} \tag{2.18}$$

donde S es la suma cuadrática de los residuos, T es el número de las observaciones efectivas que se usan en el ajuste del modelo (recuerde que se pierden observaciones por diferenciación) y r es el número de parámetros estimados en el modelo, incluyendo la constante. La DER es un criterio de selección de modelo, un valor pequeño indica una mayor adecuación del modelo.

- Criterio de Información de Akaike: El estadístico AIC propuesto por Akaike está dado por:

$$AIC = T\ln(S/T) + 2r \tag{2.19}$$

Este criterio permite seleccionar un modelo. Se prefiere el modelo que tenga el menor valor de AIC.

- Criterio Bayesiano de Schwarz: La expresión matemática de este estadístico es:

$$SBC = T\ln(S/T) + r\ln(T) \tag{2.20}$$

Este criterio es un método para la selección de un modelo. Se prefiere el modelo que tenga el mínimo valor del estadístico. El SBC penaliza los parámetros adicionales más severamente que el AIC, conduciendo a modelos más simples.

5. Predicción

a) La predicción se basa en el modelo ARIMA seleccionado.

b) Se predicen "m" períodos, correspondientes al tamaño del conjunto de predicción, con sus intervalos de confianza.

c) Se calculan los errores de predicción. Es importante juzgar la adecuación del modelo en función de qué tan bien se pronostican los datos no empleados para la estimación del modelo. Para evaluar la predicción, se utilizarán dos tipos de medición del error (Makridakis y Wheelwright, 1994):

- Error Medio Absoluto

$$\text{EMA} = \frac{1}{m}\sum_{i=1}^{m}|y_{t+i} - \hat{y}_{t+i}| \qquad (2.21)$$

donde y_{t+i} son los valores observados de la serie que pertenecen al conjunto de predicción y, $\hat{y}_{t+i}$ son los valores pronosticados por el modelo ARIMA.

- Raíz del Error Cuadrado Medio Porcentual

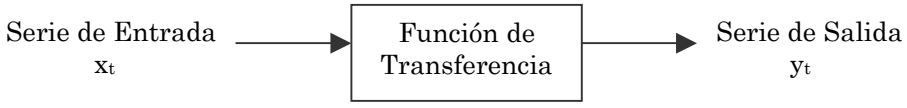
$$RECMP = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\frac{y_{t+i} - \hat{y}_{t+i}}{y_{t+i}}\right)^2} \qquad (2.22)$$

Otra forma de evaluar la predicción es a través de la correlación entre los valores observados y los pronosticados por el modelo. Valores altos de esta correlación indican una buena adecuación del modelo.

d) Si la serie pareciera cambiar en el tiempo, pudiera ser necesario recalcular los parámetros, o desarrollar un nuevo modelo.

2.2.2. Función de Transferencia: Modelo Bivariable

En la metodología de Box y Jenkins, el término Modelo de Función de Transferencia (MFT) se refiere a un modelo que predice valores futuros de una serie de tiempo en base a valores pasados de esa serie y a una o más series relacionadas con la serie de tiempo a predecir. Es decir, se estudian modelos multivariables. En esta investigación, se considerará el caso más sencillo de un MFT, se usará este modelo para predecir valores futuros de una serie de tiempo, llamada serie de salida, en base a valores pasados de esa serie y en base a otra serie de tiempo relacionada, llamada serie de entrada. Es decir, se considerará solamente el modelo bivariable. (Bowerman y O'Connell, 1993). Esquemáticamente:



Se supone que existe una relación causal unidireccional a priori, desde la serie de entrada hacia la serie de salida, eliminando la posibilidad de retroalimentación.

Etapas para Construir un Modelo de Función de Transferencia

Bowerman y O'Connell proporcionan una metodología de tres etapas para construir un modelo de función de transferencia, la cual explicamos a continuación:

1. Identificar un modelo que describa a la serie de entrada y preblanquear a la serie de entrada y salida.

   Se denota el t-ésimo término de la serie de entrada por x_t y el de la serie de salida por y_t. Se asume que la transformación necesaria para hacer a la serie de entrada estacionaria es la misma que para la serie de salida. A la serie de entrada estacionaria la denotaremos por $Z_t^{(x)}$. Se deberán seguir los siguientes pasos:

   a) Análisis de los correlogramas simple y parcial de la serie de entrada x_t, de la misma manera que se hizo en el caso univariable, para determinar la necesidad de alguna transformación que conduzca a que la serie de tiempo sea estacionaria $Z_t^{(x)}$.

   b) Análisis de los correlogramas simple y parcial de $Z_t^{(x)}$ para identificar el modelo ARIMA (p,q,P,Q). Esto fue explicado en la página Nº 15.

   c) Estimación del modelo. Seleccionar un modelo cuyos parámetros sean significativos.

$$\phi_P(B^s)\,\phi_p(B)Z_t^{(x)} = \theta_Q(B^s)\,\theta_q(B)\varepsilon_t \qquad (2.23)$$

   d) Adecuación del modelo mediante el análisis de los residuos, de la misma manera que el caso univariable (pág. 16).

   e) Preblanqueo de x_t $(Z_t^{(x)})$. Se despeja ε_t de (2.23) y a este error del modelo para la serie de entrada se le denomina α_t.

$$\alpha_t = \frac{\hat{\phi}_p(B^s)\hat{\phi}_p(B)}{\hat{\theta}_Q(B^s)\hat{\theta}_q(B)} Z_t^{(x)} \qquad (2.24)$$

   f) Preblanqueo de y_t $(Z_t^{(y)})$. Al error del modelo para la serie de salida se le denomina β_t.

$$\beta_t = \frac{\hat{\phi}_p(B^s)\hat{\phi}_p(B)}{\hat{\theta}_Q(B^s)\hat{\theta}_q(B)} Z_t^{(y)} \tag{2.25}$$

2. Identificar un MFT preliminar que describa la serie de entrada.

   a) Para identificar una función de transferencia preliminar que describa la relación entre y_t y x_t, se calcula la función de correlación cruzada (CC) entre los valores de α_t y los valores de β_t. La CC es un conjunto, para los retrasos k = -10,-9,...,0,...,9,10, de valores de:

$$r_k(\alpha_t, \beta_t) = \frac{\sum_{t=b}^{n-k}(\alpha_t - \overline{\alpha})(\beta_{t+k} - \overline{\beta})}{\left[\sum_{t=b}^{n}(\alpha_t - \overline{\alpha})^2\right]^{1/2}\left[\sum_{t=b}^{n}(\beta_t - \overline{\beta})^2\right]^{1/2}} \tag{2.26}$$

   donde

$$\overline{\alpha} = \frac{\sum_{t=b}^{n}\alpha_t}{n-b+1} \quad \text{y} \quad \overline{\beta} = \frac{\sum_{t=b}^{n}\beta_t}{n-b+1}$$

   $r_k(\alpha_t, \beta_t)$ es llamada la correlación cruzada entre los valores de α_t y β_t en el retraso k y es una medida de la relación lineal entre los valores de α_t y β_{t+k}. Se puede representar mediante el gráfico de CC.

   b) Verificar que no existan valores significativos en las CC antes del retraso cero (en los MFT se asume que valores presentes de y_t están relacionados con valores presentes y/o pasados de x_t y no al revés).

   c) Identificar b, el retraso donde la CC es estadísticamente diferente de cero.

   d) Identificar s. El valor de s se fija igual al número de retrasos que están entre la primera CC significativa y el comienzo del patrón de decaimiento. s es el número de valores pasados de x que influyen sobre y.

   e) Identificar r. La práctica ha demostrado que puede determinarse r examinando el decaimiento de la CC después del retraso (b+s). Si cae en forma exponencial amortiguada, r=1. Si cae en forma sinusoidal amortiguada, r=2. r representa el número de valores pasados de y que están relacionados con y_t

f) El MFT preliminar general es de la forma:

$$z_t^{(y)} = \mu + \frac{Cw(B)}{\delta(B)} B^b z_t^{(x)} + \eta_t \qquad (2.27)$$

donde:

μ: constante que se incluye en el modelo si es estadísticamente diferente de cero.

C: es un parámetro escalar desconocido

η_t: representa un componente de error

$w(B) = 1 - w_1B - w_2B^2 - \ldots - w_sB^s$

$\delta(B) = 1 - \delta_1B - \delta_2B^2 - \ldots - \delta_rB^r$

3. Identificación de un modelo que describa a η_t y del modelo final de función de transferencia.

a) Calcular la función de correlación cruzada de los residuos con los valores de entrada x_t y las autocorrelaciones simple y parcial de los residuos.

b) Verificar que se cumpla una condición necesaria para la validez del MFT, que establece que la serie de entrada preblanqueada α_t debe ser estadísticamente independiente del componente del error η_t. Esto se cumple cuando los valores de la probabilidad en las correlaciones cruzadas son grandes y no se puede rechazar la hipótesis de independencia.

c) Encontrar, por medio de los correlogramas simple y parcial de los residuos, el modelo que describe a η_t.

$$\eta_t = \frac{\hat{\theta}_Q(B^s)\hat{\theta}_q(B)}{\hat{\phi}_p(B^s)\hat{\phi}_p(B)} a_t \qquad (2.28)$$

donde a_t es una variación aleatoria en el tiempo t.

d) Se sustituye η_t en el MFT preliminar (ecuación 2.27) y esto constituye el modelo final de función de transferencia:

$$z_t^{(y)} = \mu + \frac{Cw(B)}{\delta(B)} B^b z_t^{(x)} + \frac{\hat{\theta}_Q(B^s)\hat{\theta}_q(B)}{\hat{\phi}_p(B^s)\hat{\phi}_p(B)} a_t \qquad (2.29)$$

Una vez construido el MFT se generan los valores correspondientes al ajuste y predicción de la serie y_t, y se calculan sus respectivos errores, como en la metodología ARIMA.

CAPÍTULO III

REDES NEURONALES

La Inteligencia Artificial (IA) es un área del conocimiento compuesta por un conjunto de técnicas que se basan en imitar computacionalmente las distintas habilidades relacionadas a la inteligencia del ser humano, como por ejemplo: reconocimiento de patrones, diagnóstico, clasificación, entre otros. Una de estas técnicas imita, específicamente, el comportamiento de las neuronas en el cerebro humano, por lo cual se le ha denominado Redes Neuronales Artificiales (RNA). Es de interés en este trabajo la aplicación de esta metodología para la predicción utilizando series de tiempo.

En este capítulo se introducirán los conceptos básicos de las RNA, su arquitectura y varios modelos que han sido utilizados con fines de pronóstico.

3.1. NEURONAS BIOLÓGICAS

Las RNA persiguen imitar ciertas habilidades humanas atribuibles al cerebro y a millones de elementos interconectados llamados Neuronas.

Según Colina y Rivas (1998): "Las neuronas son células nerviosas que constituyen los elementos primordiales del sistema nervioso central. Son capaces de recibir señales provenientes de otras neuronas, procesar estas señales, generar pulsos nerviosos, conducir estos pulsos, y transmitirlos a otras neuronas".

Las neuronas están formadas por tres componentes principales: dendritas, cuerpo celular y axón (ver figura 3.1).

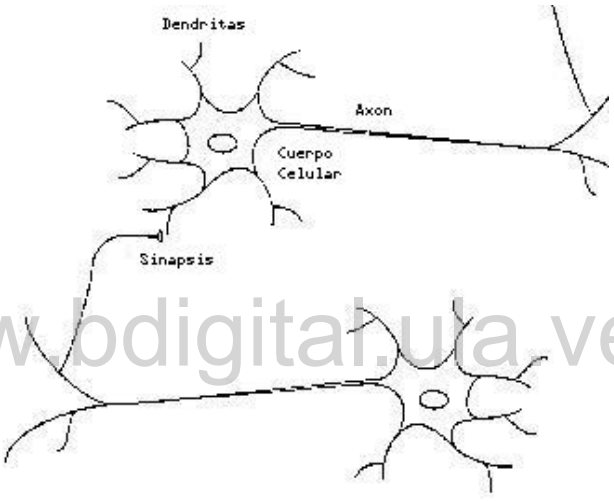


Fig. 3.1. Dibujo Esquemático de Neuronas Biológicas

Las dendritas son como un árbol de redes receptivas de fibras nerviosas que llevan señales eléctricas al cuerpo de la célula. (Hagan, Dermuth y Beale, 1996)

El cuerpo celular contiene el núcleo, tiene forma esférica y es aquí donde se ejecutan todas las transformaciones necesarias para la vida de la neurona. (Colina y Rivas, 1998)

El axón transmite la señal de salida a otras neuronas. El intercambio químico de información entre una neurona y otra se hace a través de la sinapsis. Ésta es el punto de interconexión entre neuronas.

La neurona artificial es una versión muy simple de la neurona biológica, pero puede establecerse fácilmente una equivalencia entre ambas, como lo manifiestan Colina y Rivas (1998):

"La operación de la neurona es un proceso en donde la célula ejecuta una suma de señales que llegan por sus dendritas (entradas). Cuando esta suma es mayor que cierto umbral, la neurona responde transmitiendo un pulso a través de su axón (salida)". La siguiente figura ilustra el modelo artificial y la equivalencia entre la neurona artificial y biológica:

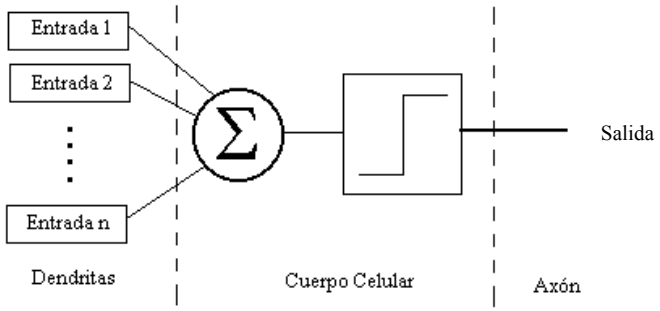


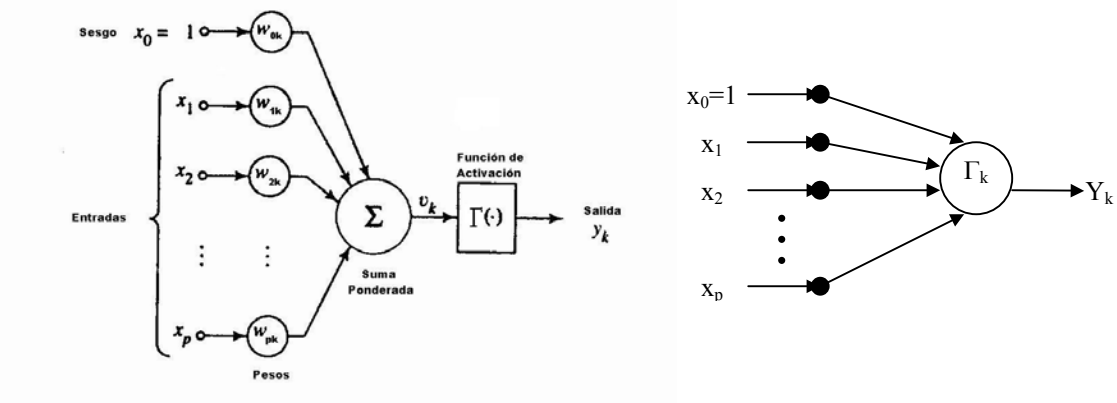
Fig. 3.2: Equivalencia entre la Neurona Artificial y Biológica

3.2. MODELO DE UNA NEURONA ARTIFICIAL

Una neurona artificial o nodo es una unidad de procesamiento de información que es fundamental para la operación de una red neuronal (Haykin, 1994). En un modelo de una neurona pueden identificarse los siguientes elementos:

- Entradas o nodos de entrada: son escalares que se le proporcionan a la red, de acuerdo al problema en estudio.
- Salidas o nodos de salida: son los valores que arroja la red como resultado del aprendizaje.
- Un conjunto de pesos sinápticos o simplemente pesos: son valores numéricos que expresa la importancia de la entrada correspondiente. El valor de la entrada x_i se dirige a la neurona k multiplicado por el peso w_{ik}. (El primer subíndice de los pesos se refiere a la entrada y el segundo a la neurona en cuestión).
- Un punto de suma de entradas ponderadas: aquí se realiza la combinación lineal o suma de todas las entradas multiplicadas por sus correspondientes pesos.
- Una función de activación: es una función, que puede ser lineal o no lineal, que limita el rango de la salida de la neurona.
- Sesgo: es un valor formado por una entrada fija e igual a 1, denominada x_0, multiplicada por el peso w_{0k}.

La figura 3.3 (a) ilustra en detalle el modelo de una neurona y, equivalentemente, la figura 3.3 (b) representa una forma más abreviada del mismo:



(a) (b)

Fig. 3.3: Modelo de una Neurona

En la figura 3.3 (a) se puede observar que la neurona k puede describirse mediante las siguientes ecuaciones:

a) La acumulación

$$v_k = \sum_{i=0}^{p} x_i w_{ik} \tag{3.1}$$

b) La función de activación

$$y_k = \Gamma(v_k) \tag{3.2}$$

donde $x_0, x_1, \ldots, x_p$ son las entradas de la neurona k; $w_{0k}, w_{1k}, \ldots, w_{pk}$ son los pesos de la neurona k; v_k es la suma de las entradas multiplicadas por los pesos correspondientes; $\Gamma(.)$ es la función de activación y y_k es la salida de la neurona k.

Los pesos son parámetros escalares que se van ajustando según la aplicación de una regla de aprendizaje de manera de cumplir con la relación entrada/salida y la función de activación se selecciona de acuerdo al objetivo del problema y al rango de valores en que se requiera la salida.

3.3. FUNCIONES DE ACTIVACIÓN

La selección de la Función de Activación (FA) depende del criterio del investigador y del problema en estudio, en muchas ocasiones se selecciona por ensayo y error. Existen diversos tipos de FA, entre los más utilizados están:

1. **Función Paso**: La salida de este tipo de FA puede ser 0 o 1, dependiendo si el parámetro de la función es positivo o negativo. Se usa para problemas de clasificación. En forma matemática puede expresarse como:

$$\Gamma(v) = \begin{cases} 1 & \text{si } v \geq 0 \\ 0 & \text{si } v < 0 \end{cases} \qquad (3.3)$$

y gráficamente,

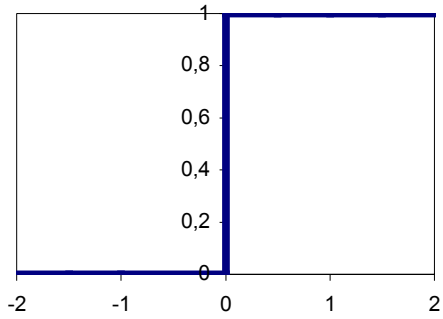


Fig. 3.4. Función Paso

2. **Función Lineal**: La entrada de la FA es igual a la salida. Se usa en diversos tipos de redes, con frecuencia, en la capa de salida. Su expresión matemática es:

$$\Gamma(v) = v \qquad (3.4)$$

Gráficamente,

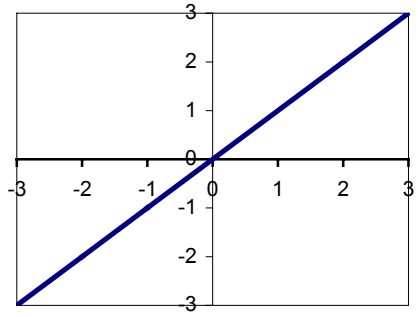


Fig. 3.5. Función Lineal

3. **Función Rampa**: Su salida está entre –1 y 1. Matemáticamente se expresa como:

$$\Gamma(v) = \begin{cases} -1 & \text{si } v \le -1 \\ v & \text{si } -1 < v < 1 \\ 1 & \text{si } v \ge 1 \end{cases} \qquad (3.5)$$

Gráficamente,

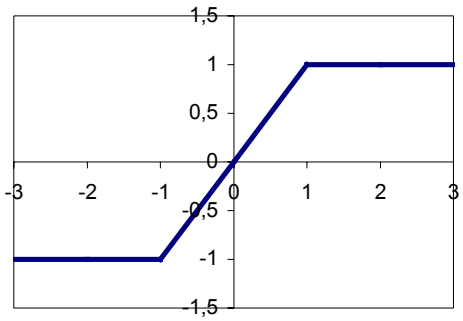


Fig. 3.6. Función Rampa

4. **Función Logística**: Su salida comprende valores entre 0 y 1. Es la FA más usada en redes neuronales y se recomienda para problemas de predicción. Su expresión matemática es:

$$\Gamma(v) = \frac{1}{1+e^{-v}} \qquad (3.6)$$

En forma gráfica,

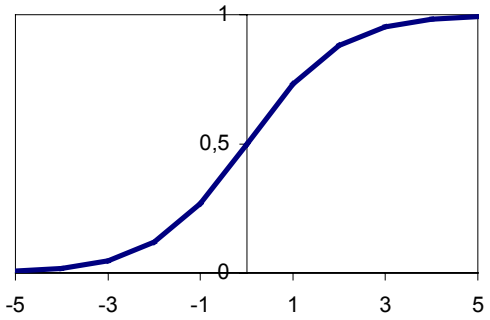


Fig. 3.7. Función Logística

5. **Función Tangente Hiperbólica**: Es semejante a la función logística, pero su salida está entre –1 y 1. Se utiliza con frecuencia en redes multicapas. En forma matemática se expresa como:

$$\Gamma(v) = \text{TanH}(v) \tag{3.7}$$

y en forma gráfica:

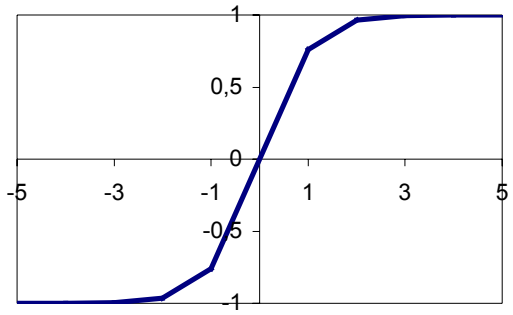


Fig. 3.8. Función Tangente Hiperbólica

6. **Función Gaussiana**: Su rango está entre 0 y 1. Se utiliza en redes neuronales de función de base radial, las cuales pueden aplicarse a problemas de predicción. Su expresión matemática es:

$$\Gamma(v) = e^{-v^2/\sigma^2} \tag{3.8}$$

Gráficamente,

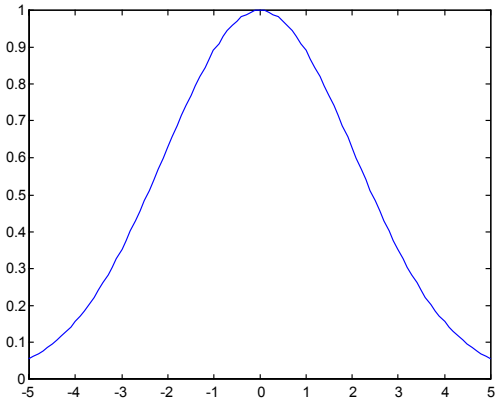


Fig. 3.9. Función Gaussiana

3.4. ARQUITECTURA DE RED

El término arquitectura de red se refiere a la forma o estructura de una RNA. La arquitectura de un modelo de RNA incluye los siguientes componentes: entradas, capas y salidas; así mismo, la interconexión y dirección de la red, que se refiere a la forma en que se interrelacionan las neuronas. El algoritmo de aprendizaje está muy relacionado con la arquitectura de la red, pero se estudiará más adelante.

Componentes

- Entradas: Es el canal de alimentación de la red. Se deberán establecer el número de entradas de acuerdo al caso en estudio. Los datos de entradas son numéricos y en muchos casos puede ser conveniente escalarlos y/o preprocesarlos. El escalamiento se refiere a un cambio de escala, convertirlos a datos entre 0 y 1, entre –1 y 1, o estandarizarlos, de acuerdo al rango de las funciones de activación involucradas. El preprocesamiento se refiere a la aplicación de algún método estadístico de exploración de datos que permita cualquier transformación que mejore el conjunto original de datos en beneficio de un mejor desempeño de la red.

- Capas. También se les denomina capas ocultas o intermedias, por encontrarse entre la entrada y la salida de la red. Con frecuencia no es suficiente una sola neurona para resolver un problema, sino que se requiere de varias neuronas que operen en paralelo, lo que se denomina capa, e inclusive pudieran ser necesarias varias capas. A la red con una sola capa se le denomina red unicapa, mientras que a la red con dos o más capas se le denomina red multicapa.

  El número de capas y de neuronas o nodos que componen cada capa debe especificarse en la arquitectura. El número de capas deberá ser mayor si el problema es no lineal y complejo, pero, en general, un problema podrá representarse bastante bien con una o dos capas ocultas. El número de neuronas por capa, puede variar entre una capa y otra. Aunque existen algunos criterios para determinar el número de nodos por capa, éstos deberán determinarse por ensayo y error. Un criterio que pudiera ser útil es considerar al promedio entre el número de entradas y salidas como un valor referencial del número de nodos ocultos.

- Salidas: Son neuronas o nodos de salida. El número de salidas de la red dependerá del problema en estudio. La salida de la red deberá estar expresada en la misma escala de los datos originales, es decir, si los datos fueron escalados, deberán ser regresados a su escala inicial. Si se realizó un preprocesamiento deberá hacerse un postprocesamiento.

Interconexión

Las interconexiones entre capas de las redes pueden clasificarse como:

- Totalmente conectadas: la salida de una neurona de la capa i es entrada a todas las neuronas de la capa i+1.
- Localmente conectadas: la salida de una neurona de la capa i es entrada a una región de neuronas de la capa i+1.

Dirección

La dirección de la información de las redes pueden clasificarse en:

- Redes de alimentación adelantada: las salidas de las neuronas de una capa sólo se propagan a las neuronas de la capa siguiente. Es decir, la información fluye solamente de la entrada a la salida.
- Redes retroalimentadas: las salidas de las neuronas de una capa pueden ser entradas de las neuronas de las capas anteriores.
- Redes de alimentación lateral: las salidas de las neuronas pueden ser entradas de neuronas de la misma capa
- Redes recurrentes: existen lazos cerrados.

Un ejemplo de red unicapa totalmente conectada, de alimentación adelantada, con 3 entradas, 2 neuronas o nodos en la capa oculta, y una salida sería:

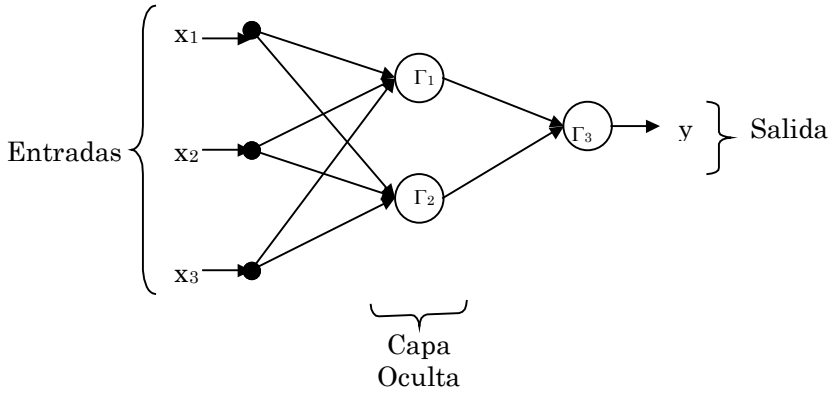


Fig. 3.10. Ejemplo de una arquitectura de red

Esta red, por ejemplo, está formada por una sola capa que consta de dos neuronas en paralelo. Las entradas se orientan a la salida. Las 3 entradas se comunican con las 2 neuronas de la única capa, éstas dos neuronas se comunican con la neurona de salida para proporcionar la salida de la red. No se consideran como capas las entradas ni las salidas, así lo señalan algunos autores: Colina y Rivas (1998), y Hagan, Demuth y Beale (1996).

3.5. ENTRENAMIENTO Y GENERALIZACIÓN

Un modelo de red neuronal al ser construido origina dos actividades fundamentales para la posterior utilización. Estas son: entrenamiento o aprendizaje y generalización o prueba.

3.5.1. Entrenamiento

El entrenamiento de una red se lleva a cabo mediante una regla o algoritmo de aprendizaje. Este algoritmo es un procedimiento para modificar los pesos de una red y su propósito es entrenar la red para ejecutar alguna tarea (Hagan, Dermuth y Beale, 1996). En otras palabras, los algoritmos de aprendizaje proporcionan la habilidad para aprender a capturar información, mediante el ajuste de los pesos o ponderaciones de interconexión (Colina y Rivas, 1998).

Terminología

En el proceso de entrenamiento se usa cierta terminología que conviene explicar previamente:

- Patrón: una observación o experimento del conjunto o patrones de entrenamiento.
- Patrones de Entrenamiento: son ejemplos o grupos de datos suministrados a la red para su aprendizaje y están compuestos por patrones de entrada y patrones de salida.
- Ciclo: Es un barrido de todos los patrones del conjunto o patrones de entrenamiento.
- Criterios de Parada del Entrenamiento:
  - Se repiten los ciclos hasta que el error alcance un valor máximo permisible.
  - Se alcance un número máximo de ciclos.

Algoritmos de Entrenamiento

Los algoritmos de entrenamiento, en general, pueden clasificarse en dos tipos:

- Supervisados: "un maestro" guía la red en cada etapa del aprendizaje, indicándole el resultado correcto (Colina y Rivas, 1998). La misión del algoritmo es ajustar los pesos de la red de manera tal, que dado un conjunto de entradas las salidas proporcionadas por la red deberán coincidir lo más posible con las salidas especificadas en el patrón de entrenamiento. Ejemplos: regla de Hebb, aprendizaje de Widrow-Hoff y algoritmo de retropropagación.
- No Supervisados: los pesos son modificados únicamente en respuesta a las entradas de la red, es decir no están disponibles las salidas (u objetivos). La mayoría de estos

algoritmos tienen como misión realizar algún tipo de agrupamiento. Éstos aprenden a categorizar los patrones de entrada en un número finito de clases (Hagan, Dermuth y Beale, 1996). Ejemplos: regla de Hebb no supervisada y regla de Kohonen.

En esta investigación utilizaremos únicamente algoritmos de entrenamiento supervisados. Debido a su adecuación para resolver problemas de predicción. Son muchos los algoritmos de aprendizaje, pero se expondrán brevemente los que se consideran de interés para la comprensión de este punto y de este trabajo.

- Perceptrón Discreto: se utiliza para clasificar mediante una recta, es un dicotomizador. El algoritmo de aprendizaje para un perceptrón discreto con una sola neurona puede expresarse como sigue:

$$W_{(k+1)} = W_{(k)} + ceX \tag{3.9}$$

donde:

$W_{(k)}$: es el vector de pesos para el patrón de entrenamiento k

$W_{(k+1)}$: es el vector de pesos para el patrón de entrenamiento k+1

c: factor de aprendizaje o paso

$e = y_d - y_n$: es el error, la diferencia entre la salida deseada (proveniente del patrón de entrada del conjunto de entrenamiento) y la salida estimada por la red neuronal.

X: vector de entradas (proveniente del patrón de entrada del conjunto de entrenamiento)

Mediante un ejemplo es fácil entender su funcionamiento. Suponga un dicotomizador que clasifique un punto (X_1,X_2) en clase 1 y clase 2, mediante una recta.

Los patrones de entrenamiento tabulados y en forma gráfica son los siguientes:

	Entradas		Salida
Patrón	**X1**	**X2**	**Yd**
1	-1	3	1
2	-2	6	1
3	1	7	1
4	3	1	0
5	7	-3	0
6	9	-5	0

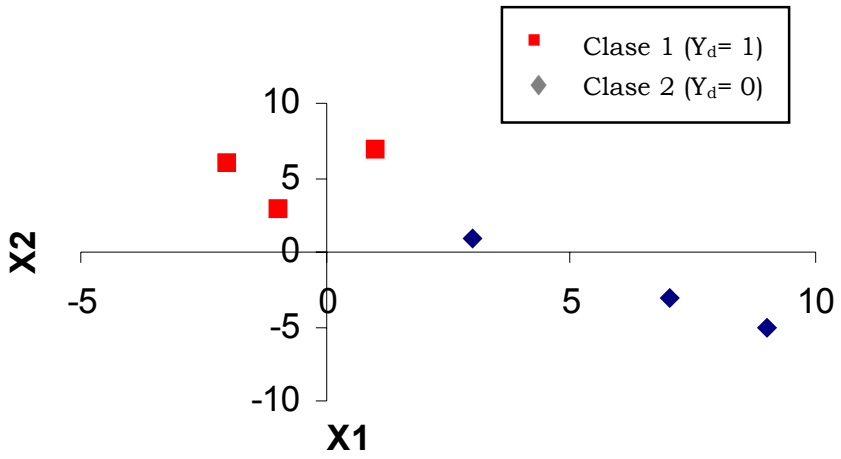


Fig. 3.11. Patrón de entrenamiento en forma tabular y gráfica

El diseño de una red neuronal con dos entradas, una sola neurona en la capa oculta, una salida y función de activación Paso sería:

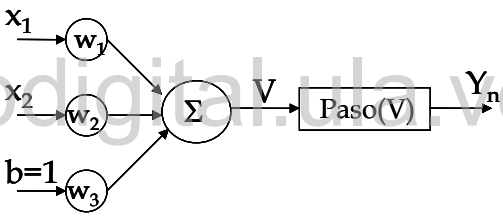


Fig. 3.12. Red neuronal

Diseño:

donde $W=[w_1,w_2,w_3]^T$, $X=[x_1,x_2,1]^T$ y b es el sesgo, por lo tanto

$V=w_1 x_1 + w_2 x_2 + w_3 * 1$ y la salida de esta red estaría dada por:

$$Yn = Paso(V) = \begin{cases} 1 & \text{si } V \geq 0 \Rightarrow \text{Clase 1} \\ 0 & \text{si } V < 0 \Rightarrow \text{Clase 2} \end{cases}$$

Entrenamiento:

El entrenamiento propiamente consiste en:

- Inicialización de los pesos: con frecuencia es aleatoria, pero, en este caso, consideremos los siguientes pesos $W_{(0)} = [1, 2, 3]^T$.
- Ciclo 1
  - Patrón 1:

x_1	x_2	b	Y_d
-1	3	1	1

$Y_n = Paso\,(w_1x_1 + w_2x_2 + w_3*1)$

$= Paso\,[1*(-1) + 2*3 + 3*1]$

$= Paso\,(8) = 1$

$e = Y_d - Y_n = 1 - 1 = 0$ => No se actualizan los pesos

– Patrón 2:

x_1	x_2	b	Y_d
-2	6	1	1

$Y_n = Paso\,(w_1x_1 + w_2x_2 + w_3*1)$

$= Paso\,[1*(-2) + 2*6 + 3*1]$

$= Paso\,(13) = 1$

$e = Y_d - Y_n = 1 - 1 = 0$ => No se actualizan los pesos

– Patrón 3:

x_1	x_2	b	Y_d
1	7	1	1

$Y_n = Paso\,(w_1x_1 + w_2x_2 + w_3*1)$

$= Paso\,(1*1 + 2*7 + 3*1)$

$= Paso\,(18) = 1$

$e = Y_d - Y_n = 1 - 1 = 0$ => No se actualizan los pesos

– Patrón 4:

x_1	x_2	b	Y_d
3	1	1	0

$Y_n = Paso\,(w_1x_1 + w_2x_2 + w_3*1)$

$= Paso\,(1*3 + 2*1 + 3*1)$

$= Paso\,(8) = 1$

$e = Y_d - Y_n = 0 - 1 = -1$ => Se actualizan los pesos

$W_{(1)} = W_{(0)} + c\,e\,X$, donde c se fija en 1 (apropiado para el caso discreto)

$$= \begin{bmatrix}1\\2\\3\end{bmatrix} + (-1)\begin{bmatrix}3\\1\\1\end{bmatrix} = \begin{bmatrix}-2\\1\\2\end{bmatrix}$$

– Patrón 5:

x_1	x_2	b	Y_d
7	-3	1	0

$Y_n = Paso\,(w_1x_1 + w_2x_2 + w_3*1)$

$= Paso\,[-2*7 + 1*(-3) + 2*1]$

$= Paso\,(-15) = 0$

$e = Y_d - Y_n = 0 - 0 = 0$ => No se actualizan los pesos

– Patrón 6:

x_1	x_2	b	Y_d
9	-5	1	0

$Y_n = Paso\ (w_1x_1 + w_2x_2 + w_3*1)$

$= Paso\ [-2*9 + 1*(-5) + 2*1]$

$= Paso(-21) = 0$

$e = Y_d - Y_n = 0 - 0 = 0$ => No se actualizan los pesos

- Ciclo 2: se repite el procedimiento para todos los patrones. Si los errores son siempre cero, el entrenamiento ha sido completado, lo cual ocurre en este caso, y los pesos definitivos son $W_{(1)}$. Así, la salida de la red está definida por la siguiente función: $Y_n = Paso\ (-2\ x_1 + x_2 + 2)$
  Con la cual puedo tomar un punto (x_1, x_2) nuevo y clasificarlo.

- Perceptrón Discreto Unicapa (múltiples neuronas): El algoritmo de perceptrón discreto, dado por la ecuación (3.9), actualiza el vector de pesos de una sola neurona. Este algoritmo puede generalizarse para el caso de múltiples neuronas como sigue:

$$W_{ij(k+1)} = W_{ij(k)} + ceX_i \qquad (3.10)$$

Los vectores de pesos del perceptrón discreto ahora se convierten en matrices de pesos. Éstas contienen los pesos entre la entrada y las neuronas de la única capa y los pesos entre estas neuronas y la salida.

- Perceptrón Continuo: La función de activación del perceptrón discreto se cambia en el perceptrón continuo a fin de lograr un control más fino del proceso de entrenamiento. Esta función deberá ser diferenciable, permitiendo de esa manera el cálculo de errores por gradiente. Para el ajuste de los pesos se minimiza la función del error utilizando procedimientos de máximo descenso o por técnicas de gradiente. El procedimiento básico de descenso consiste en calcular el gradiente de la función del error, $\nabla E(W)$, empezando con un vector de pesos arbitrariamente seleccionado W; el próximo valor de W es obtenido moviéndonos en la dirección negativa del gradiente sobre la superficie de error multidimensional. La dirección de gradiente negativo es aquella de descenso máximo. El algoritmo puede expresarse como:

$$W_{(k+1)} = W_{(k)} - \eta \nabla E(W_{(k)}) \qquad (3.11)$$

donde η es una constante positiva llamada factor de aprendizaje y $E = (y_d - y_n)^2$ es el error para el k-ésimo patrón de entrenamiento, el cual consiste en la diferencia cuadrática entre la salida deseada y la salida de la red neuronal. (Colina y Rivas, 1998).

- Algoritmo de Retropropagación o del gradiente descendente: Es la generalización del algoritmo para perceptrón continuo cuando se tienen múltiples capas. Popularmente utilizado para la solución de problemas de clasificación y pronóstico.

  El uso del término "retropropagación" aparece después de 1985, sin embargo, la idea básica de retropropagación fue descrita primero por Werbos en su tesis doctoral en 1974. Subsecuentemente, fue redescubierto por Rumelhart, Hinton y Williams en 1986, y popularizado mediante la publicación del libro titulado "Parallel Distributed Processing" por los autores Rumelhart y McClelland. Una generalización similar del algoritmo fue derivada independientemente por Parker en 1985, y también por LeCum en 1985 (Haykin, 1994).

  El método general de entrenamiento puede resumirse en los siguientes pasos:

  Pasos hacia delante:

  1. Seleccionar un patrón de entrada del conjunto de entrenamiento.
  2. Aplicar esta entrada a la red y calcular la salida.

  Pasos hacia atrás:

  3. Calcular el error entre la salida de la red neuronal y la salida deseada para el patrón de entrada usado.
  4. Ajustar los pesos para que el error cometido entre la salida de la red neuronal y la salida deseada sea disminuido.
  5. Repetir los pasos 1 al 4 para todas los patrones de entrenamiento, hasta que el error global sea aceptablemente bajo.

  Los valores de los pesos se buscan de forma tal que se reduzca la función del error (diferencia cuadrática entre la salida deseada y la salida de la red neuronal), y este proceso se realiza por el método del gradiente descendente. La idea del método es realizar un cambio en los pesos inversamente proporcional a la derivada del error respecto al peso para cada patrón.

Durante la fase de asociación o clasificación, la red opera enteramente en forma de cascada directa o hacia delante, es decir sin retroalimentación. Sin embargo, el ajuste de los pesos obtenido por la regla de entrenamiento se realiza desde atrás hacia delante, pasando por las capas ocultas hasta el nivel de entrada (Colina y Rivas, 1998). Como el error se propaga en la red de atrás hacia delante, se le denominó a este procedimiento algoritmo de retropropagación.

Error de Entrenamiento

Para medir el error de entrenamiento comúnmente se utiliza la suma cuadrática del error de entrenamiento(SCEE), cuya expresión matemática es:

$$SCEE = (Y_d - Y_n)^2 \tag{3.12}$$

donde:

Y_d: Y deseada. Es la salida del modelo, especificada en el patrón de entrenamiento.

Y_n: Y neuronal. Es la salida proporcionada por la red, cuando ésta ya ha sido entrenada y se le proporcionan las entradas del patrón de entrenamiento.

3.5.2. Generalización

Una vez que una red neuronal ha sido entrenada mediante un algoritmo supervisado, esto implica que los pesos de interconexión han sido ajustados de forma que se minimice el error. Esta red deberá estar en capacidad de proveer salidas próximas a los valores deseados cuando se le proporcionan nuevos ejemplos, es decir, entradas que no pertenecen al conjunto de entrenamiento, sino que forman parte de los patrones de prueba. A este proceso se le conoce como generalización de un modelo de RNA.

Para que ocurra una buena generalización deben cumplirse las siguientes condiciones:

- Las entradas de la red deben contener suficiente información en relación con el objetivo o salida deseada, de manera que exista una función matemática que relacione correctamente (con el grado de exactitud deseado) las salidas con las entradas.
- La función que se trata de aprender debe ser, en cierto sentido, suave. Es decir, cambios pequeños en las entradas, producen cambios pequeños en las salidas.
- El número de patrones de entrenamiento deberá ser suficientemente grande y representativo del conjunto de los casos que se quiere generalizar.

Error de Generalización

Para medir el error de generalización comúnmente se utiliza la suma cuadrática del error de generalización (SCEG), cuya expresión matemática es:

$$SCEG = (Y_d - Y_n)^2 \quad (3.13)$$

donde:

Y_d: Y deseada. Es la salida del modelo, especificada en el patrón de prueba.

Y_n: Y neuronal. Es la salida proporcionada por la red, cuando ésta ha sido entrenada previamente y se le proporcionan las entradas del patrón de prueba.

3.6. MODELOS DE REDES NEURONALES USADOS PARA PREDICCIÓN

Son muchos los autores que han intentado hacer predicciones sobre los valores futuros de una variable mediante redes neuronales, nombraremos algunos que se consideran de interés y se explicará brevemente el modelo de red que utilizaron:

- Faraway y Chatfield (1998) exponen un caso de estudio ajustando una variedad de modelos de redes neuronales a los bien conocidos datos de la línea aérea y compara los resultados de la predicción con los obtenidos con el método de Box-Jenkins y Holt-Winters. Se consideran RN de alimentación adelantada, una capa oculta con función de activación logística, las entradas corresponden a distintos retrasos de la variable y una salida con función de activación lineal que es la predicción en el tiempo t. La fig. 3.13 representa un ejemplo de una red neuronal para predicción, con tres entradas: una entrada con valor constante 1, el valor de la variable en el tiempo t-1, y el valor de la variable en el tiempo t-12, una capa oculta con dos nodos o neuronas y una sola salida:

  Procedimiento

  1. Se calcula $v_j = \Sigma\, w_{ij} y_i$ donde los w_{ij} denotan los pesos de las conexiones entre la entrada y_i y la j-ésima neurona. Los valores de entrada, en este caso, son $y_1 = 1$, $y_2 = x_{t-1}$ y, $y_3 = x_{t-12}$.
  2. La suma lineal, v_j, es transformada por la función de activación logística: $z_j = \frac{1}{(1+\exp(-v_j))}$, para j=1,2; la cual arroja valores entre 0 y 1.
  3. Una operación similar se aplica a z_1, z_2 y a la constante de entrada para obtener la salida o predicción. Se aplica una función lineal.

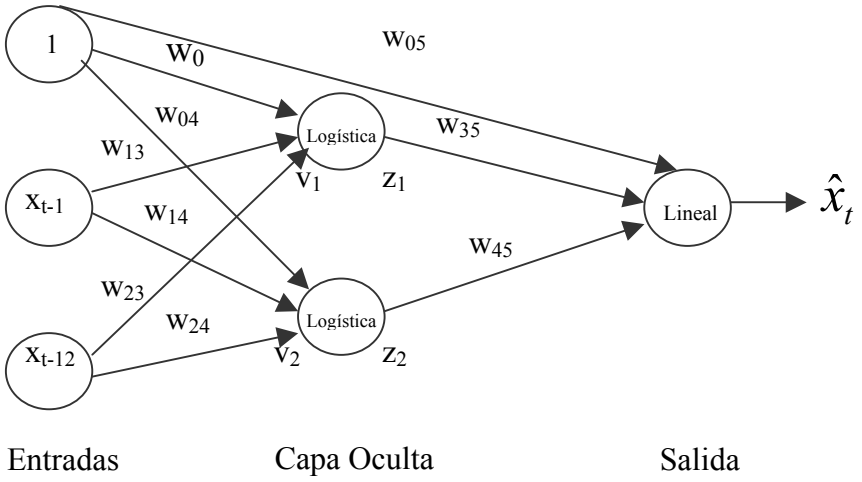


Fig. 3.13. RN de alimentación adelantada con una capa oculta para la predicción de series de tiempo con datos mensuales.

Ecuación de Predicción General

$$\hat{x}_t = \phi_o\left\{w_{co} + \sum_h w_{ho}\phi_h\left(w_{ch} + \sum_i w_{ih}x_{t-i}\right)\right\} \tag{3.14}$$

donde $\{w_{ch}\}$ denota los pesos para las conexiones entre la entrada constante y la neurona oculta y w_{co} denota el peso de la conexión directa entre la entrada constante y la salida. Los pesos $\{w_{ih}\}$ y $\{w_{ho}\}$ denotan los pesos para las otras conexiones entre las entradas y las neuronas ocultas, y entre estas neuronas y la salida, respectivamente. Las dos funciones ϕ_h y ϕ_o denotan las funciones de activación usadas en la capa oculta y en la salida respectivamente.

Los pesos a ser usados en el modelo de RN son estimados de los datos minimizando la suma de cuadrados de los errores de predicción,

$$S = \sum_t \left(\hat{x}_t - x_t\right)^2, \tag{3.15}$$

sobre la primera parte de la serie de tiempo, llamada el conjunto de entrenamiento (los pesos iniciales se generan aleatoriamente 50 veces y se toma el modelo que tenga S mínimo). Se utiliza el algoritmo de retropropagación.

- Hill, O'Connor y Remus (1996): El experimento consistió en comparar las predicciones producidas mediante RNA con las predicciones obtenidas por expertos, utilizando seis métodos estadísticos de series de tiempo, en la famosa competencia

de Makridakis en 1982. Una arquitectura de RNA fue implementada para las series anuales, otra para las series trimestrales y otra para series mensuales. En todos los casos, la RNA es de alimentación adelantada, totalmente conectada, tiene una sola capa oculta con función de activación logística y se usa el algoritmo de retropropagación para su entrenamiento.

Para datos anuales, se consideraron tres entradas, dos nodos en la capa oculta y una salida, es decir, los datos de los años t-2 hasta t fueron usados para predecir el año t+1. Para datos trimestrales, se usaron cuatro entradas, dos nodos en la capa oculta y una salida; los datos de los trimestres t-3 hasta t fueron usados para predecir el trimestre t+1. Para los datos mensuales, se consideraron nueve entradas, cuatro nodos en la capa oculta y una salida; los datos de los meses t-8 hasta t fueron usados para predecir el mes t+1. Para hacer predicciones más allá del periodo t+1, se obtiene primero la predicción t+1 y se usa como una de las entradas a la red para producir la predicción para el periodo t+2. Este procedimiento se repite para todo el horizonte de predicción.

- Wong (1991): También utilizó para la predicción con series de tiempo redes neuronales de alimentación adelantada, completamente conectadas, con una o dos capas intermedias, una o más salidas y el algoritmo de retropropagación para entrenarlas. Como regla utilizó en la primera capa oculta el doble de nodos que el número de entradas, la mitad de los nodos con función de activación logística y la otra mitad sinusoidal. En la segunda capa utilizó únicamente la función logística.

- Wedding y Cios (1996): Utilizan una RNA denominada Función de Base Radial (FBR) para predicción con series de tiempo. Esta red está compuesta por tres capas. La primera es la de entrada, alimenta los datos de entrada a cada uno de los nodos de la segunda capa o capa oculta. La segunda capa difiere de las anteriores en que cada nodo representa un grupo de datos, el cual es centrado en un punto particular y tiene un radio dado. La tercera capa o capa de salida consiste de un solo nodo. Este suma las salidas de los nodos de la segunda capa para alcanzar el valor de decisión.

  Cuando un vector de entrada x es alimentado, en cada nodo de la capa oculta se calcula la distancia del vector de entrada al centro del nodo c. El valor de la distancia es transformado mediante una función gaussiana y el resultado es la salida del nodo. Ese valor es multiplicado por una constante o valor ponderado. El producto es alimentado en el nodo de la tercera capa, el cual suma todos los productos y cualquier entrada numérica constante, en este caso, se usa la media de

todos los datos de entrada. Por último, la salida del nodo de la tercera capa proporciona el valor de decisión.

Representación gráfica de una RN RBF con 3 centros de datos:

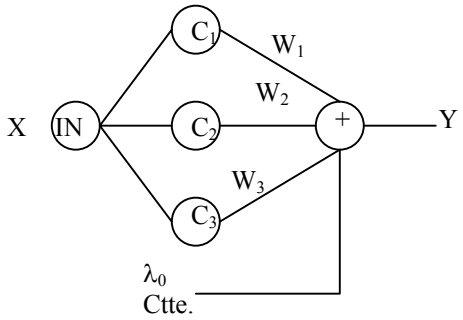


Fig. 3.14. Red Neuronal Función de Base Radial con 3 centros de datos

En forma matemática la RN recibe un vector de entrada k dimensional y sale un valor escalar usando la siguiente fórmula:

$$f_x(\bar{x}) = \lambda_0 + \sum_{i=1}^{n} \lambda_i \phi(v_i) \tag{3.16}$$

donde n es el nº de nodos de la capa oculta, λ_i son los pesos y v es el vector de longitud n que contiene la distancia medida del vector de entrada x a cada uno de los centros.

Usualmente, la Norma Euclideana es usada para calcular la distancia, pero cualquier otra métrica puede ser usada. La ecuación de la Norma Euclideana es dada por:

$$v_i = \sqrt{\sum_{j=1}^{k} (x_j - c_{ji})^2} \tag{3.17}$$

c: centro del grupo (los centros se seleccionan del vector de aprendizaje usando mínimos cuadrados ortogonales).

$\phi(v)$ es la Función Gaussiana y está dada por:

$$\phi(v) = \exp\left(-v^2 / \beta^2\right) \tag{3.18}$$

donde β es el radio del grupo (se fija en 0,1).

Los datos de entrenamiento y prueba son normalizados y sus predicciones tienen un sesgo añadido: la media.

CAPÍTULO IV

UNA METODOLOGÍA PARA PREDICCIÓN CON REDES NEURONALES

Como se estudió en el capítulo anterior muchos autores han incursionado en el área de predicción de series de tiempo haciendo uso de redes neuronales con diferentes arquitecturas, diferentes algoritmo de entrenamiento, etc. En este capítulo se definirá una metodología propia para la aplicación de redes neuronales con la cual se realizarán las predicciones de series de tiempo univariable y las predicciones de una serie de tiempo que tiene una relación causal con otra (caso bivariable).

4.1. METODOLOGÍA PARA PREDICCIÓN DE UN MODELO UNIVARIABLE CON REDES NEURONALES

A continuación se especifican los pasos a seguir para construir, entrenar y probar una red neuronal para predecir valores futuros de una serie de tiempo, basada únicamente en sus valores pasados:

1. Escalamiento de los datos

- Transformar los datos a valores comprendidos entre 0 y 1, utilizando la siguiente fórmula:

$$z_t = \frac{y_t - \text{Min}}{\text{Max} - \text{Min}} \tag{4.1}$$

donde:

y_t: son los valores originales de la serie de tiempo

Min: valor mínimo de la serie de tiempo

Max: valor máximo de la serie de tiempo

z_t: serie de tiempo transformada en valores entre 0 y 1

2. Patrones de Entrenamiento y Prueba

Los valores de la serie de tiempo se dividen en dos conjuntos de datos:

- Patrones de entrenamiento: Está formado por el 80 % de los datos de la serie. Se seleccionan en forma consecutiva y ordenada. Este conjunto de datos es el que se utilizará para el entrenamiento de la red neuronal .

- Patrones de prueba: Está formado por el 20% de los datos de la serie. Corresponden a los datos restantes, una vez que se han seleccionado los patrones de entrenamiento. Este conjunto de datos se utiliza para evaluar la capacidad de generalización o predicción de la red.

3. Topología de la RN

- Dirección de la información: Alimentación adelantada
- Tipo de interconexión: Totalmente conectada
- Nº de entradas: p + 1 (una constante de valor 1, denominada sesgo o intercepto)

- Nº de capas ocultas: 1
- Nº de nodos en la capa oculta: q
- Nº de salidas: 1
- Función de activación de los nodos de la capa oculta: logística
- Función de activación de la salida: logística

4. Determinación de las entradas (p)

Pueden considerarse varias recomendaciones que ayudarán en la selección de las entradas a la RNA:

- La periodicidad de los datos: Como en esta investigación se utilizan únicamente series de tiempo mensuales, es lógico pensar en considerar 12 o 13 retrasos. Pudiera ser conveniente construir una primera red con 13 entradas, correspondientes a los 13 retrasos y analizar los pesos, de manera que se seleccionen las entradas asociadas a los pesos de mayor magnitud, como lo sugieren Faraway y Chatfield (1998). Este análisis de los pesos ayuda a identificar las variables de entrada más importantes.

- Una vez determinado el modelo ARIMA, seleccionar como entradas a los valores correspondientes a los retrasos de y_i involucrados en este modelo.

- Otra herramienta que podría considerarse son los correlogramas simple y parcial de la serie estacionaria. Las entradas a la red serían los datos correspondientes a los retrasos que resultaran con una correlación significativamente diferente de cero.

- Pruebas por ensayo y error: Otras pruebas pudieran ser de interés, por ejemplo, considerar los datos retrasados 1,4,8,12 periodos, 1,3,6,9,12 periodos, o 1,2,3,4 periodos.

5. Determinación del número de nodos de la capa oculta (q)

- Una regla ad hoc, que en experimentos previos ha resultado de utilidad, es asumir que el valor inicial del número de nodos de la capa oculta sea igual al promedio entre el número de entradas y salidas, es decir: (# Entradas + # Salidas)/2 (si el valor obtenido es decimal se redondea).

- Pueden realizarse pruebas por ensayo y error, agregando más nodos, y comparando los errores de ajuste y predicción.

6. Algoritmo de Entrenamiento: Retropropagación

7. Selección de los pesos iniciales

La escogencia de los pesos iniciales puede ser crucial y es recomendable probar con diferentes conjuntos de valores iniciales para tratar de obtener buenos resultados. Los pesos iniciales se generan aleatoriamente 50 veces (Faraway y Chatfield, 1998). Se selecciona el modelo que obtenga el menor promedio entre la suma de cuadrados de los errores de ajuste y predicción (ecuación 3.15).

8. Entrenamiento de la RN seleccionada

- Para entrenar la red es necesario establecer los siguientes parámetros:
  - El número máximo de ciclos y el error permitido de convergencia se fijará por ensayo y error.
  - Tasa de aprendizaje, incremento de la tasa de aprendizaje y momento pueden ser fijados en 0,05; 1,05 y 0,95, respectivamente. Es conveniente realizar pruebas cambiando estos valores, y evaluando el comportamiento de los errores de entrenamiento y generalización.
- Una vez definida la RNA, con su ecuación se generan los valores de la serie de tiempo ajustada o producida por la red, utilizando los patrones de entrenamiento.
- Se calcula el error de entrenamiento.

9. Predicción

- Usando la ecuación de predicción definida por la RNA se obtiene el valor de predicción t+1. Para hacer predicciones más allá del periodo t+1, se utiliza ésta como entrada para producir la predicción t+2 y así sucesivamente para todo el conjunto de predicción.
- Cálculo del error de generalización.

10. Comparación entre RNA con diferentes entradas

Mediante el error de entrenamiento y de generalización se comparan las RNA generadas y se selecciona aquella en la que ambos valores sean mínimos. No es conveniente que el error de entrenamiento sea muy pequeño en comparación con el error de generalización, pues esto indica un sobreajuste o memorización. La correlación entre los valores originales de la serie y los estimados por la RNA puede usarse como una medida de la exactitud de la predicción.

4.2. METODOLOGÍA PARA PREDICCIÓN DE UN MODELO BIVARIABLE CON REDES NEURONALES

La metodología a seguir en el caso bivariable es esencialmente la misma que en el caso univariable, la diferencia radica en las entradas de la red. No sólo se deben considerar retrasos de la serie a predecir (y), sino también valores presentes y pasados de la serie relacionada (x). Para definir las entradas pueden considerarse los siguientes aspectos:

- Periodicidad de los datos: Si se consideran datos mensuales, es lógico pensar en un máximo de 12 o 13 retrasos por serie.

- Los retrasos de interés para las series de entrada y salida pueden identificarse, como en el caso univariable, mediante el modelo ARIMA o los correlogramas simple y parcial (se seleccionan los retrasos cuyas correlaciones sean significativamente diferentes de cero).

- Modelo de Función de Transferencia: Pudieran considerarse como entrada los valores de la serie de entrada (x) correspondientes a los b retrasos, indicados en el MFT. También los valores de la serie de salida (y) correspondientes a los r retrasos.

- Correlación Cruzada: Una herramienta disponible es considerar como entradas los valores correspondientes a los retrasos de la serie x que tengan correlaciones cruzadas significativamente diferentes de cero.

- Pruebas por ensayo y error: Otras pruebas con diferentes retrasos para ambas series pudieran ser de interés.

Se selecciona la RNA con los menores errores de entrenamiento y generalización.

CAPÍTULO V

EXPERIMENTOS Y RESULTADOS

La experimentación es la fase central de cualquier investigación. Aquí se enfocará en el estudio de dos casos: el primero, caso univariable, se analiza la serie de tiempo número de nacimientos mensuales ocurridos en España entre 1960 y 1999; y el segundo, caso bivariable, se estudia la serie de tiempo ventas en base a la serie publicidad (Bowerman y O'Connell). El análisis consiste en aplicar, en el primer caso, la metodología ARIMA, y en el segundo la metodología de función de transferencia, y a ambos casos se les aplicará la metodología para predicción de RNA especificada en el capítulo IV. Por último se hará un análisis comparativo de los resultados obtenidos con estas técnicas estadísticas contra los resultados alcanzados con las RNA.

5.1. SERIE NACIMIENTOS: CASO UNIVARIABLE

En el caso de predicciones con una sola serie de tiempo se analizará la serie de tiempo del número de nacimientos mensuales, ocurridos en España desde enero de 1960 hasta diciembre de 1999. Esta serie puede conseguirse en la página web del Instituto Nacional de Estadística de España (2001). El análisis de la serie de tiempo se hará aplicando la metodología ARIMA y RN.

5.1.1. Metodología ARIMA

La Metodología ARIMA o Método Univariado de Box y Jenkins se aplicará siguiendo las siguientes etapas y con el uso del paquete estadístico SPSS® 8.0 de SPSS Inc.:

Etapa 1: Análisis Exploratorio de la Serie

- Análisis gráfico de la serie completa:

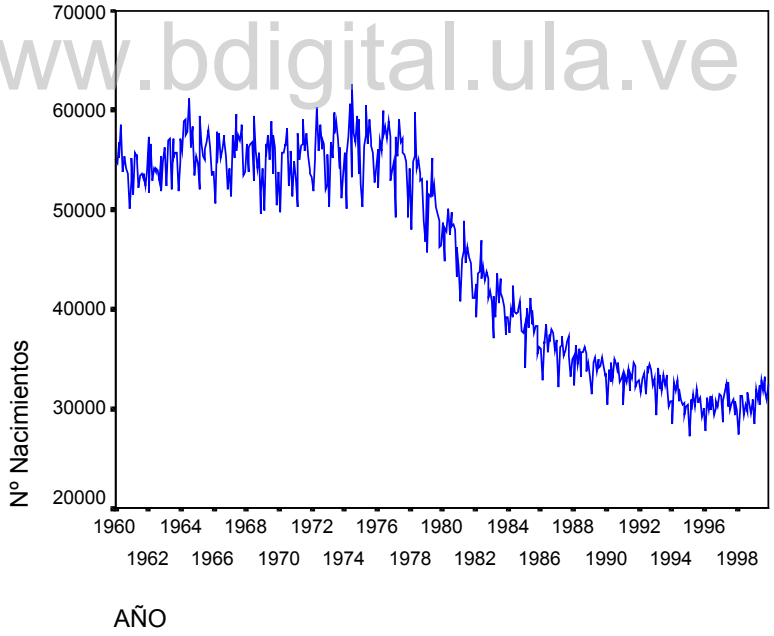


Fig. 5.1. Secuencia del número de nacimientos ocurridos en España desde enero de 1960 hasta diciembre de 1999

Aproximadamente hasta 1980 no hay tendencia, a partir de este año comienza a observarse una marcada tendencia decreciente en el número de nacimientos, que intenta estabilizarse hacia los años noventas. Es claro, que la media no es constante, por

lo tanto la serie debe diferenciarse. También se observa estacionalidad. La serie no es estacionaria.

- Puede obtenerse una idea del comportamiento de la varianza, mediante un diagrama de caja por año:

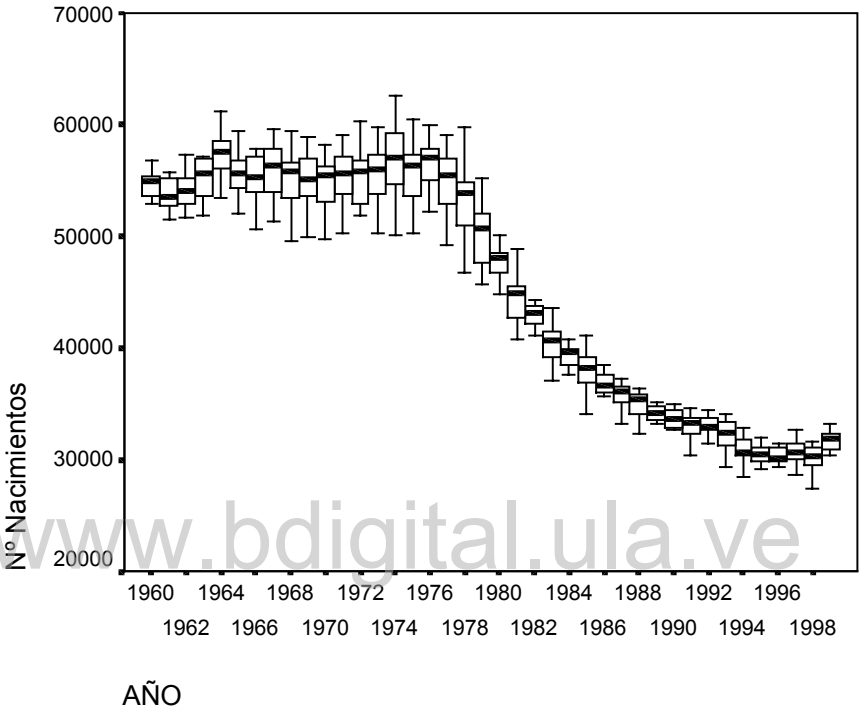


Fig.5.2. Diagramas de caja por año para el número de nacimientos ocurridos en España entre 1960 y 1999

Parece existir heterocedasticidad, lo cual puede ser verificado con la Prueba de Homogeneidad de Varianzas de Levene:

Prueba de homogeneidad de varianzas

Nº Nacimientos

Estadístico de Levene	gl1	gl2	Sig.
2,393	39	440	0,000

Fig.5.3. Tabla de los resultados de la prueba de homogeneidad de varianza de Levene para el nº de nacimientos.

Dada la significación obtenida (0,000) se rechaza la hipótesis nula de homogeneidad de varianzas y se verifica la presencia de heterocedasticidad. Se puede aplicar una transformación logarítmica que con frecuencia conseguirá que las variaciones sean constantes y se aplica otra vez la Prueba de Levene:

Prueba de homogeneidad de varianzas

LN(Nº de Nacimientos)

Estadístico de Levene	gl1	gl2	Sig.
0,891	39	440	0,661

Fig.5.4. Tabla de los resultados de la prueba de homogeneidad de varianza de Levene para el ln(nº de nacimientos).

No se rechaza la hipótesis nula de homogeneidad de varianzas (p>0,66) y se considera a la serie ln(nº de nacimientos) como homocedástica.

- Al diferenciar la serie (y_t - y_{t-1}) y transformarla aplicando logaritmo neperiano, se obtiene, gráficamente, la siguiente serie:

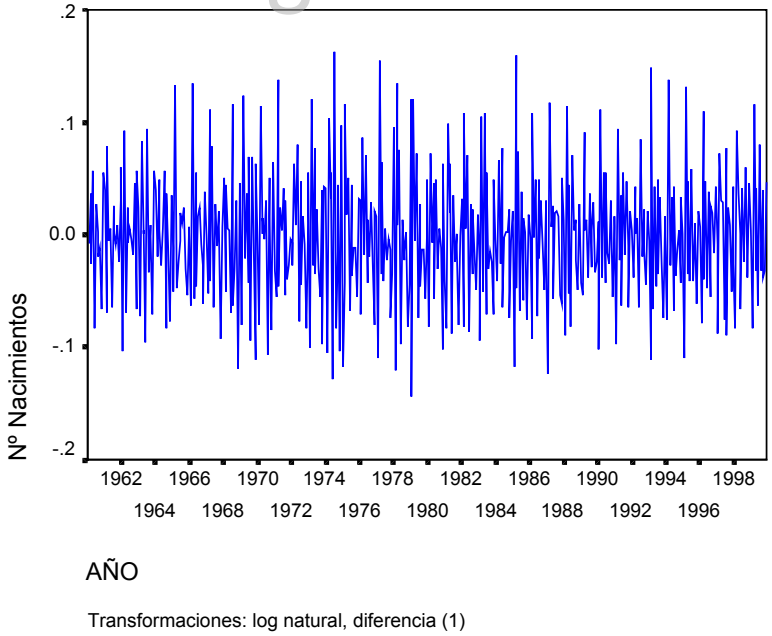


Fig. 5.5. Secuencia del número de nacimientos con las transformaciones: logaritmo neperiano y 1era. diferencia en la parte regular (Enero de 1960-Diciembre de 1999)

Ahora, la serie aparece sin tendencia (media constante) y también se cumple la homogeneidad de varianza, gracias a la transformación logarítmica.

Etapa 2: Identificación del Modelo

Para la identificación del modelo se considerará el 80% de los datos, lo cual corresponde a los meses comprendidos entre enero de 1960 hasta diciembre de 1991, esto equivale a 384 meses de los 480 meses que comprende el lapso de tiempo estudiado, enero de 1960 hasta diciembre de 1999. El conjunto de datos de prueba para validar el modelo, estaría formado por los 96 valores correspondientes a los meses comprendidos entre enero de 1992 hasta diciembre de 1999.

- Correlogramas simple y parcial. Una vez aplicada la primera diferencia en la parte regular y la transformación logarítmica, se obtiene:

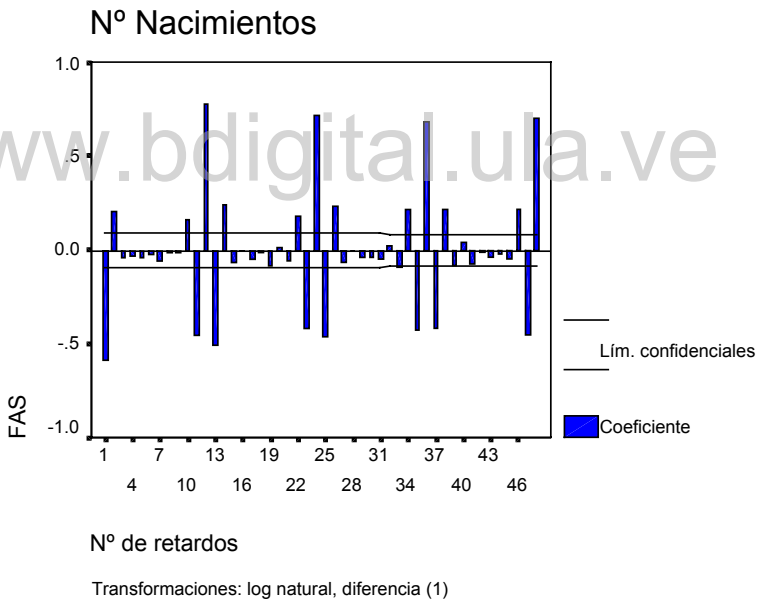


Fig. 5.6. Correlograma Simple con las transformaciones: logaritmo natural y 1 era. diferencia en la parte regular

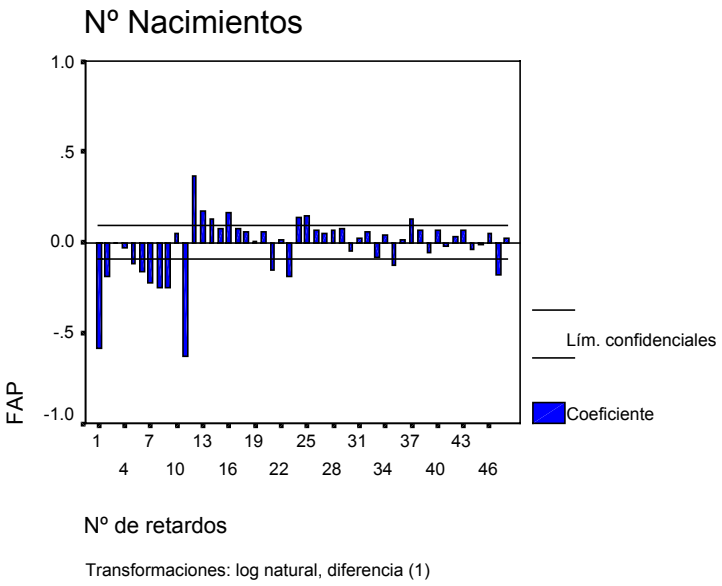


Fig. 5.7. Correlograma Parcial con las transformaciones: logaritmo natural y 1 era. diferencia en la parte regular

En el correlograma simple se observa estacionalidad de periodo 12, se debe aplicar 1era. diferencia en la parte estacional.

- Correlograma Simple y Parcial con transformación logarítmica, 1era. diferencia en la parte regular y en la parte estacional:

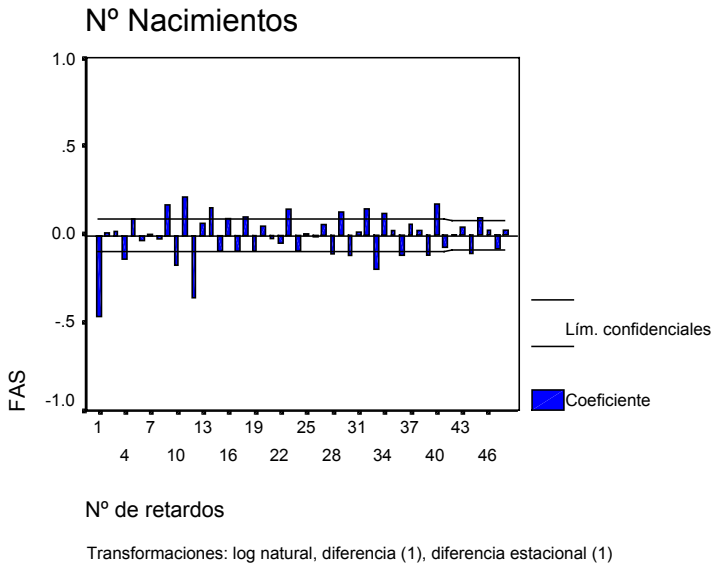


Fig. 5.8. Correlograma Simple con las transformaciones: logaritmo natural, 1 era. diferencia en la parte regular y 1 era. diferencia en la parte estacional

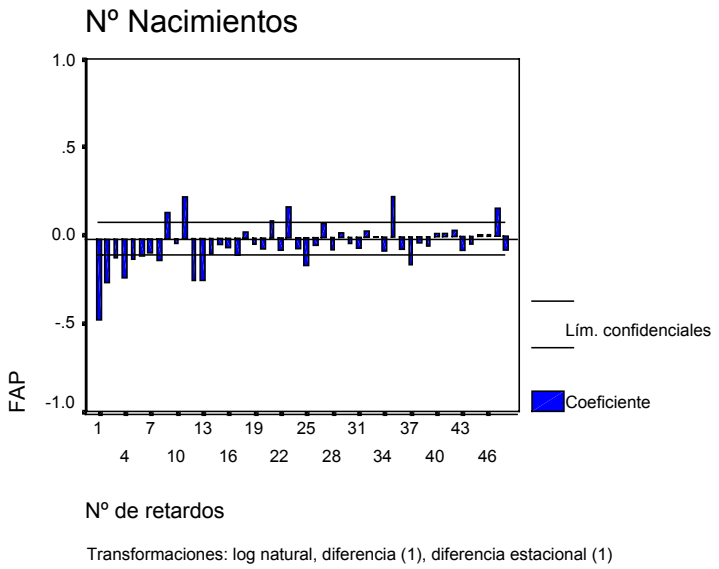


Fig. 5.9. Correlograma Parcial con las transformaciones: logaritmo natural, 1 era. diferencia en la parte regular y 1 era. diferencia en la parte estacional

En la parte regular se observa un patrón MA (Promedio Móvil) de orden 1 (q=1)

- Correlogramas Simple y Parcial para los 10 primeros retardos (120 meses) estacionales:

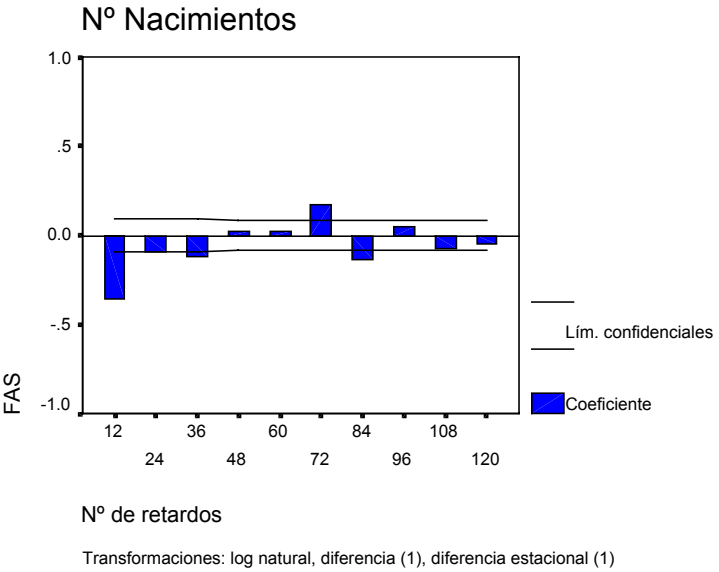


Fig. 5.10. Correlograma Simple (estacional) con las transformaciones: logaritmo natural, 1 era. diferencia en la parte regular y 1 era. diferencia en la parte estacional

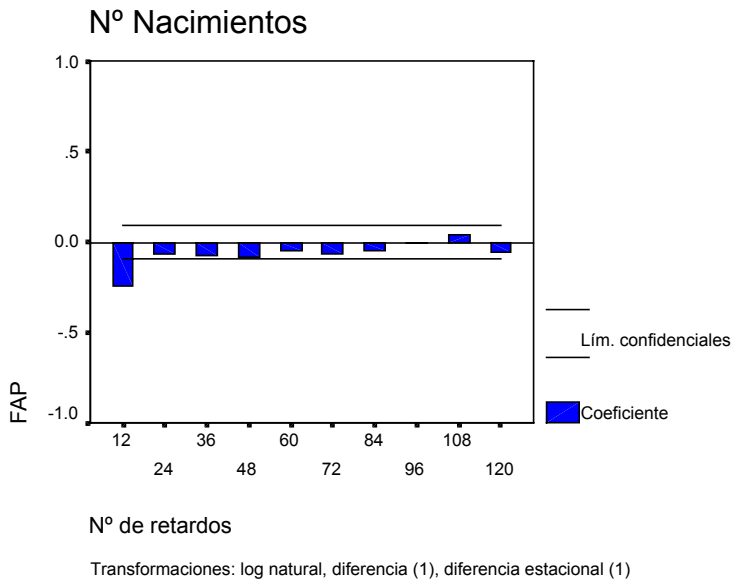


Fig. 5.11. Correlograma Parcial (estacional) con las transformaciones: logaritmo natural, 1 era. diferencia en la parte regular y 1 era. diferencia en la parte estacional

En la parte estacional se observa un patrón AR (autoregresivo) de orden 1 (P=1).

- Modelo identificado ARIMA(0,1,1)*(1,1,0)$_{12}$. Se deben realizar pruebas con modelos afines.

<u>Etapa 3: Estimación de Parámetros</u>

- Se realizaron numerosas pruebas, siendo los parámetros significativos y obteniéndose resultados satisfactorios con los siguientes modelos:

  A) ARIMA(0,1,1)*(1,1,0)$_{12}$ sin constante
  B) ARIMA(0,1,1)*(0,1,1)$_{12}$ sin constante
  C) ARIMA(1,1,0)*(0,1,1)$_{12}$ sin constante

  El modelo de mejor ajuste resultó ser el B, con menor Desviación Estándar Residual (1371,96), AIC (5362,20) y SBC (5370,03).

  Los parámetros del modelo ARIMA(0,1,1)*(0,1,1)$_{12}$ sin constante, para el periodo de ajuste (enero 1960-diciembre 1991) son:

PARAMETRO	ESTIMACIÓN	T-RATIO	APPROX. PROB.
MA1	0,74350215	21,256739	0,0000000
SMA1	0,71156701	17,871405	0,0000000

El modelo estimado es:

$$(1-B^{12})(1-B)\, y_t^* = (1-0{,}71156701\, B^{12})\,(1-0{,}74350215\, B)\,\varepsilon_t \qquad (5.1)$$

donde y_t^* es la serie transformada por logaritmo natural.

Los parámetros del modelo considerando la serie completa, tanto el periodo de estimación como el de predicción, son:

PARAMETRO	ESTIMACIÓN	T-RATIO	APPROX. PROB.
MA1	0,69853659	21,028877	0,0000000
SMA1	0,73616397	21,722571	0,0000000

Estos parámetros cambian ligeramente en relación con los determinados para el periodo de ajuste, por lo tanto puede decirse que la estructura de la serie no varía y sigue siendo válido el modelo seleccionado.

Etapa 4: Adecuación del Modelo

- En la siguiente gráfica se representa la serie original y ajustada:

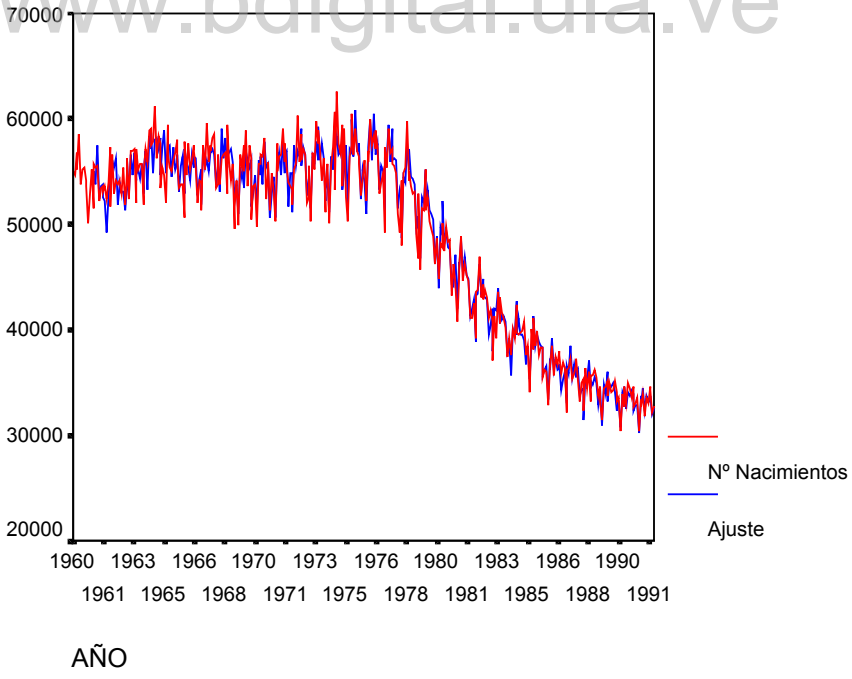


Fig. 5.12. Serie original y ajustada. Periodo: enero de 1960 - diciembre de 1991

Al comienzo el ajuste no es muy bueno, pero va mejorando paulatinamente. Puede decirse que la serie ajustada representa bastante bien a la serie original.

- Análisis de los residuos

La Prueba de Kolmogorov Smirnov permite verificar si los residuos siguen una distribución Normal. Se obtuvo una significación de 0,999, lo que indica que no puede rechazarse la hipótesis nula de normalidad de los residuos. La media de los errores es –0,00045, aproximadamente cero.

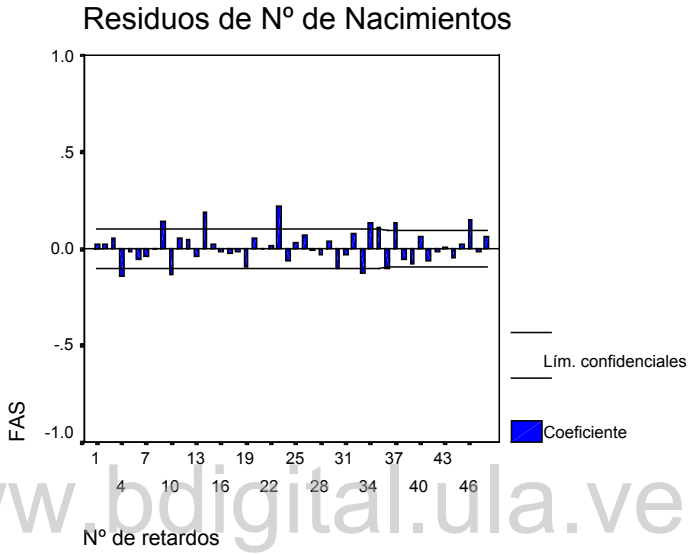


Fig. 5.13. Correlograma Simple de los residuos para la serie nacimientos

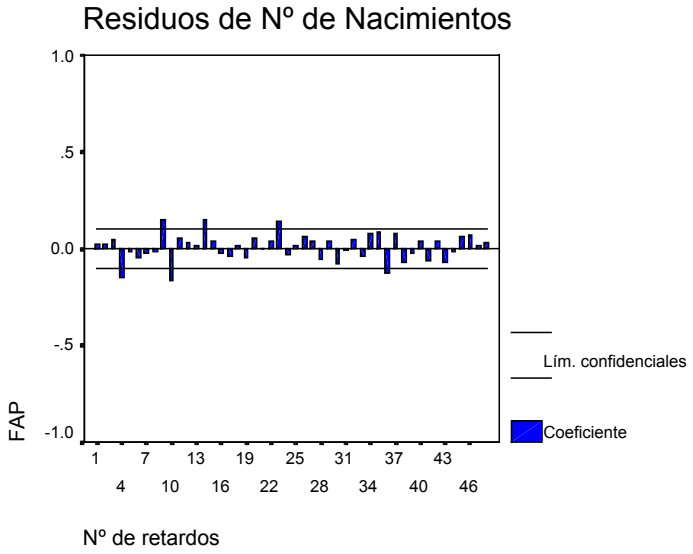


Fig. 5.14. Correlograma Parcial de los residuos para la serie nacimientos

En los correlogramas de los residuos se observan valores significativos, lo que indica que queda cierta información sobre la estructura de la serie (ver fig. 5.13 y 5.14).

Etapa 5: Predicción

En el apéndice 1 se presentan los valores de la serie original de nacimientos, las predicciones obtenidas con el Modelo ARIMA$(0,1,1)*(0,1,1)_{12}$ y el error para el lapso comprendido entre enero de 1992 y diciembre de 1999.

En la Fig. 5.15 se representan los valores originales de la serie nacimiento y las predicciones para el periodo enero 92 - diciembre 1999. Se observa que hasta el año 1996 la predicción es próxima a los valores originales de la serie, pero a partir del año 1997 la predicción se aleja notablemente de los valores reales, posiblemente la serie cambió de patrón. Esto pudiera corregirse mediante una técnica llamada análisis de intervención (ver Box, Jenkins y Reinsel, 1994, cap. 12)

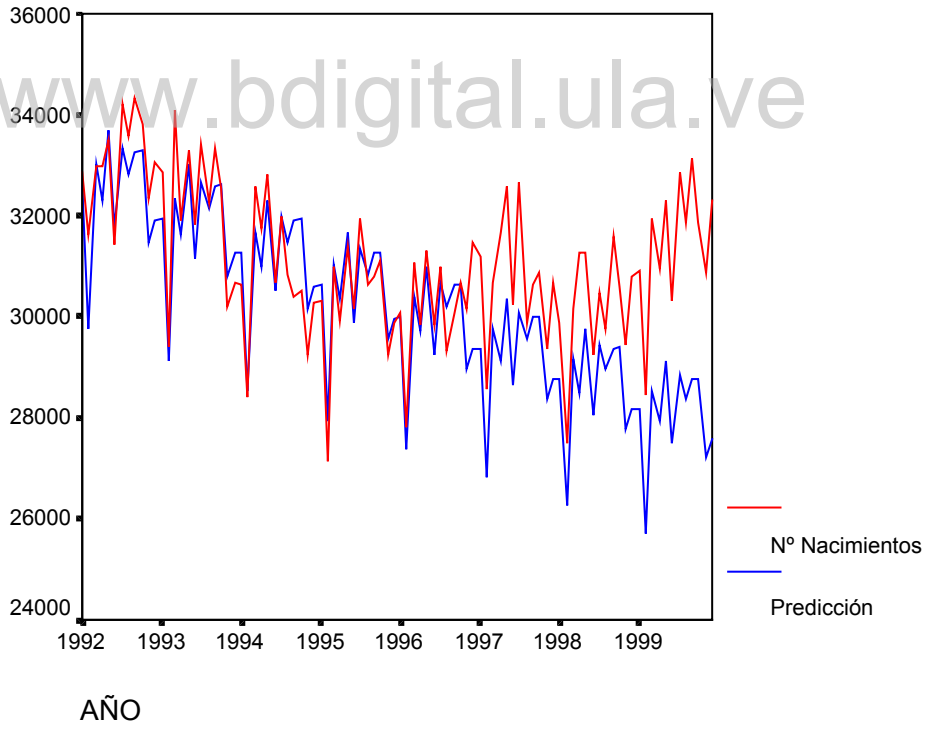


Fig. 5.15. Serie original y la predicción. Periodo: enero de 1992 - diciembre de 1999

5.1.2. Redes Neuronales

Se sigue la metodología especificada en el Capítulo IV, sección 4.1. para predecir valores futuros de la serie de tiempo nacimientos, basándose en ciertos valores pasados. Se utilizó el paquete MatLAb® 5.1 producido por MathWorks, Inc. para desarrollar todas las rutinas computacionales, entrenamiento y prueba de las RN (ver apéndice 2).

1. Escalamiento de los datos

   Todos los datos de la serie de tiempo son transformados a valores entre 0 y 1, mediante una rutina computacional descrita en el apéndice 2 (a).

2. Patrones de Entrenamiento y Prueba

   Los valores de la serie de tiempo están comprendidos entre enero de 1960 y diciembre de 1999, lo que equivale a 40 años. Estos datos se dividen en dos conjuntos:

   Patrones de Entrenamiento: Enero 1960 - Diciembre 1991. Comprende 384 meses (32 años) que equivale al 80% de los datos.

   Patrones de Prueba: Enero 1992 – Diciembre 1999. Comprende 96 meses (8 años) que equivale al 20% de los datos.

3. Topología de la RN

   Todas las RN que serán probadas son de alimentación adelantada, totalmente conectadas, con una capa oculta, un nodo adicional tanto en la capa de entrada como en la capa oculta de valor constante 1 (sesgo) y función de activación logística en los nodos de la capa oculta y en el nodo de salida.

4. Determinación de las entradas (p)

   La selección de los retrasos de entrada se realiza según las recomendaciones dadas en el capítulo IV, sección 4.1. Se considerarán, los siguientes retrasos como entradas para las RN a probar:

   - Los retrasos del 1 al 13: Se consideran los retrasos del 1 al 12 debido a que los datos son anuales y el retraso 13 por la estacionalidad del modelo ARIMA.

- Los retrasos 1, 12 y 13: Son los retrasos utilizados en el modelo ARIMA encontrado.

- Los retrasos 1, 2, 4, 9, 10, 11, 12 y 13: Son los retrasos correspondientes a las correlaciones significativamente diferentes de cero, en los correlogramas simple y parcial usados en la metodología ARIMA.

- Los retrasos 1, 2, 12 y 13: Como una prueba adicional se le agregó el retraso 2 a los retrasos utilizados en la metodología ARIMA.

- Los retrasos 3, 5 y 9: Una sugerencia de Faraway y Chatfield (1998) es el análisis de las matrices de pesos. A continuación se plantea una propuesta para seleccionar las entradas analizando los pesos de una RN que tiene como entradas los retrasos del 1 al 13. Los pesos entre la capa oculta y la de salida son:

  1.7130 -2.3105 2.3124 1.8086 -3.1162 -1.8028 -2.3698

  La homogeneidad en los valores de los pesos puede considerarse como un indicativo de que el número de nodos en la capa oculta es apropiado, y también señala que no existen grandes diferencias en relación a la importancia de un nodo con respecto a otro, por lo cual se procede al estudio de la matriz de pesos entre las entradas (E) y los nodos de la capa oculta (N):

N\E	1	2	3	4	5	6	7	8	9	10	11	12	13
1	-0.94	0.09	0.3	-2.33	-0.32	-0.67	1.17	-0.91	2.81	-3.97	2.03	-1.12	-1.49
2	0.89	-3.59	-1.53	0.3	-0.5	-1.72	1.37	1.99	2.44	2.34	-1.34	-2.72	0.83
3	0.91	0.83	-0.87	0.97	2.12	1.58	-1.26	-0.19	1.76	1.73	-0.18	1.91	-0.77
4	-2.44	3.61	-1.22	0.47	0.68	-2.3	1.41	0.43	0.8	-1.14	-1.18	-1.24	3.21
5	1.37	-0.73	-1.28	-0.88	0.89	1.9	-0.82	-3.13	-2.63	-2.7	0.56	2.61	0.58
6	0.27	-0.86	0.3	0.23	3.38	0.77	-0.6	-1.13	3.46	-0.46	-2.5	-0.39	-2.49
7	-0.83	1.2	-2.17	1.55	0.7	-1.55	-2.43	-1.99	-1.94	-0.33	2.04	-1.53	-2.12

Fig. 5.16. Matriz de pesos entre las 13 entradas y los 7 nodos de la capa oculta

Las medias de los pesos por entradas son:

Media	-0.11	0.08	**-0.92**	0.04	**0.99**	-0.28	-0.17	-0.7	**0.96**	-0.65	-0.08	-0.36	-0.32

Fig. 5.17. Medias de los pesos para las 13 entradas

Se consideran como entradas para esta red los tres retrasos con medias mayores en valor absoluto (3, 5 y 9).

El establecimiento de las entradas para cada red se hace mediante el programa ConjuntosEP (ver apéndice 2 c).

5. Determinación del número de nodos de la capa oculta (q)

   El número de nodos de la capa oculta se tomó en todos las RN como el promedio entre el número de entradas (p) y el número de salidas (1), para valores decimales se redondea por defecto. Así:

Red Neuronal Nº	Retrasos (Entradas)	Nº de Entradas (p)	Nº de Nodos de la Capa Oculta (q)
RN1	Del 1 al 13	13	7
RN2	1,12,13	3	2
RN3	1,2,4,9,10,11,12,13	8	4
RN4	1,2,12,13	4	2
RN5	3,5,9	3	2

Fig. 5.18. Tabla con las RN seleccionadas, nº de entradas y nº de nodos de la capa oculta

   Se realizaron algunas pruebas aumentando y disminuyendo el número de nodos, pero no hubo diferencias significativas, por tanto no se consideraron.

6. Algoritmo de entrenamiento: Las 5 RN se entrenan con el algoritmo de retropropagación

7. Selección de los pesos iniciales

   Los pesos iniciales se generan 50 veces en cada caso, mediante la función NWLOG (generador aleatorio Nguyen-Widrow para neuronas LOGSIG), y de los 50 resultados obtenidos se selecciona la red que obtenga el promedio mínimo entre la suma de cuadrados de los errores de entrenamiento (ajuste) y generalización (predicción). Para esto se desarrolló un procedimiento computacional descrito en el apéndice 2 (e)

8. Ajuste o entrenamiento de la RN

   - Fijación de los parámetros:

     - Número máximo de épocas: 2000
     - Error permitido de Convergencia: 0,01
     - Tasa de Aprendizaje: 0,01
     - Incremento en la tasa de aprendizaje: 1,05
     - Momento: 0,95

     Se hicieron pruebas variando estos parámetros, en general, se encontraron las menores suma de cuadrados de los errores de entrenamiento y generalización con estos valores.

- Con la ecuación general de cada una de las cinco RN se generan los valores de la serie de tiempo ajustada, utilizando las entradas de los patrones de entrenamiento.

9. Predicción

   Por medio de la ecuación general de cada una de las cinco RN se generan las predicciones para la serie de tiempo, utilizando las entradas de los patrones de prueba.

10. Comparación con las diferentes RN

    Se comparan las cinco RN seleccionadas a través de los errores de ajuste y predicción:

RN Nº	Ajuste			Predicción		
	DER	AIC	SBC	EMA	RECMP	CORR
RN1	2.977,5	6.147,1	6.562,3	1.083,0	4,47	0,54
RN2	**2.365,6**	**5.786,4**	**5.829,5**	996,7	**4,07**	**0,66**
RN3	2.949,5	6.010,1	6.170,7	1.133,5	4,82	0,50
RN4	2.920,6	5.946,8	5.997,7	1.113,4	4,64	0,53
RN5	2.639,5	5.867,7	5.910,8	**985,08**	4,19	0,65

Fig. 5.19. Errores de ajuste y predicción para las cinco RN seleccionadas

Se selecciona la RN2 por obtener los menores errores de ajuste y dos de los menores errores de predicción (RECMP y CORR). Podría considerarse la RN5, pues tiene el menor EMA y los demás errores están cerca de los valores mínimos encontrados para la RN2.

El programa que realiza el cálculo de los errores de ajuste y predicción se presenta en el apéndice 2 f.

5.1.3. Análisis Comparativo: Ajuste y Predicción

En la siguiente tabla se presentan los errores de ajuste (entrenamiento) y predicción (prueba), tanto para la mejor RN obtenida como para el Modelo ARIMA seleccionado:

RN Nº	Ajuste			Predicción		
	DER	AIC	SBC	EMA	RECMP	CORR
RN2	2.365,6	5.786,4	5.829,5	**996,7**	**4,07**	**0,66**
ARIMA	**1.372,0**	**5.362,2**	**5.370,0**	1173,8	5,09	0,48

Fig. 5.20. Errores de ajuste y predicción para la mejor RN y el Modelo ARIMA

La RN2 tiene como entradas los mismos retrasos utilizados en el modelo ARIMA: 1, 12 y 13; y en la capa oculta posee 2 nodos. Note que en el modelo ARIMA los errores de ajuste son menores que en la RN, lo que implica un mejor ajuste a la serie en la fase de

entrenamiento, mientras que los errores de predicción son menores para la RN, indicando una mejor predicción o generalización con el uso de la RN. A continuación, puede observarse esto mediante los gráficos para el ajuste y predicción con RN y el modelo ARIMA comparado con los verdaderos valores de la serie nacimiento.

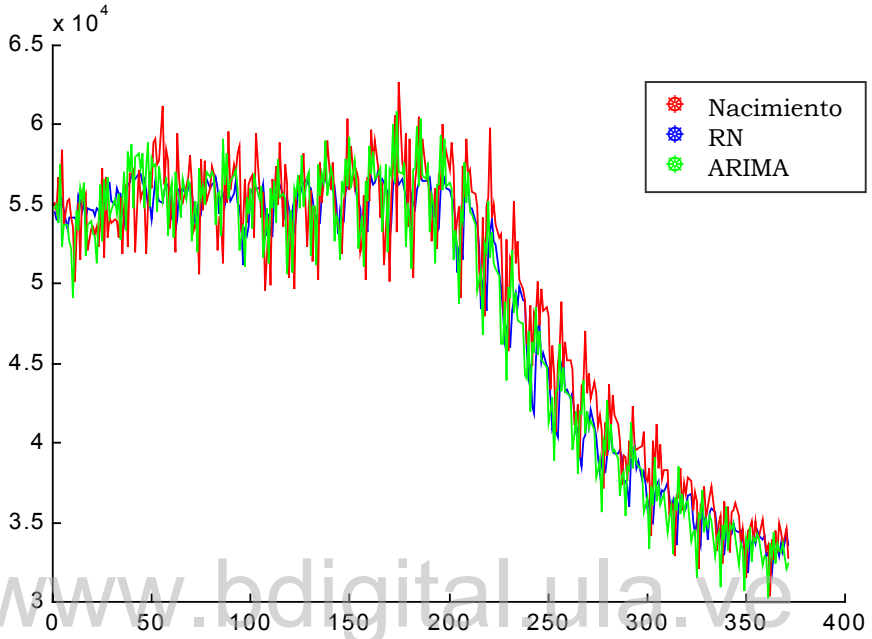


Fig. 5.21. Gráfico del ajuste de la serie nacimiento con RN y ARIMA

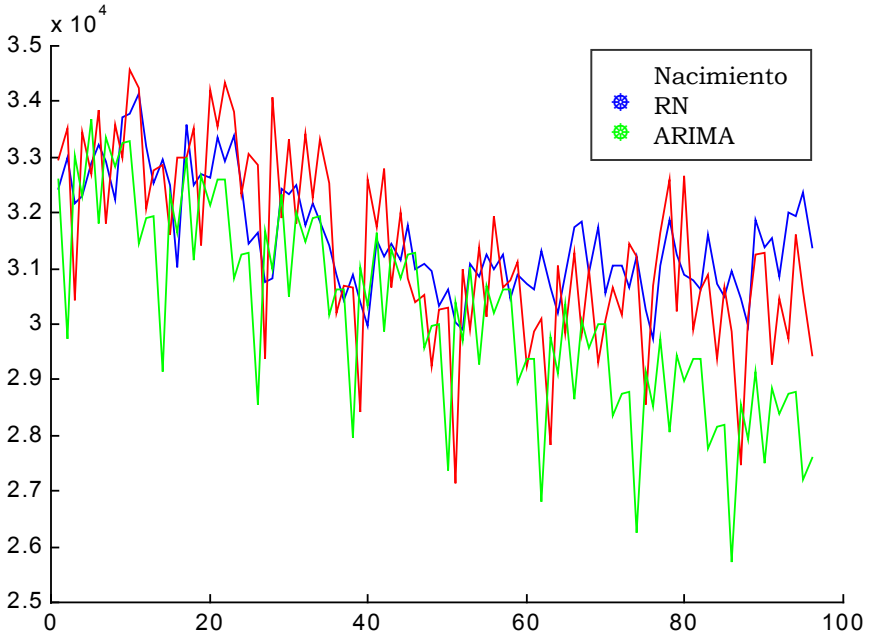


Fig. 5.22. Gráfico de la predicción de la serie nacimiento con RN y ARIMA

5.2. SERIES PUBLICIDAD Y VENTAS: CASO BIVARIABLE

En el caso bivariable se analizarán los datos presentados por Makridakis, Whelwright y McGee (1983) referenciados por Bowerman y O'Connell (1994). Estos datos consisten de 100 observaciones sobre dos series de tiempo: número de ventas mensuales (en miles de casos) y gasto mensual en publicidad (en miles de dólares). El objetivo es predecir el número de ventas (y_t) en base a sus valores pasados y a la publicidad (x_t), considerando que existe una relación causal entre ambas. El conjunto de datos se dividirá en dos subconjuntos: conjunto de ajuste o entrenamiento formado por 80 pares de datos (equivalente al 80%) y conjunto de predicción o prueba formado por 20 pares de datos (equivalente al 20%).

5.2.1. Función de Transferencia

Para construir el modelo de función de transferencia (MFT) se seguirá la metodología señalada en el capítulo II, sección 2.2.2, haciendo uso de los paquetes SPSS® 8.0 de SPSS Inc. y SAS® 3.95 de SAS Institute Inc. (el programa puede verse en el apéndice 3).

Se denotará a la serie de entrada por x_t: gasto mensual en publicidad, en miles de dólares y a la serie de salida por y_t: Número de ventas mensuales, en miles de casos.

Etapa 1: Identificación de un modelo para la serie publicidad y preblanqueo de las series.

- Análisis de la serie publicidad para convertirla en estacionaria

  La gráfica de secuencia para la serie publicidad es:

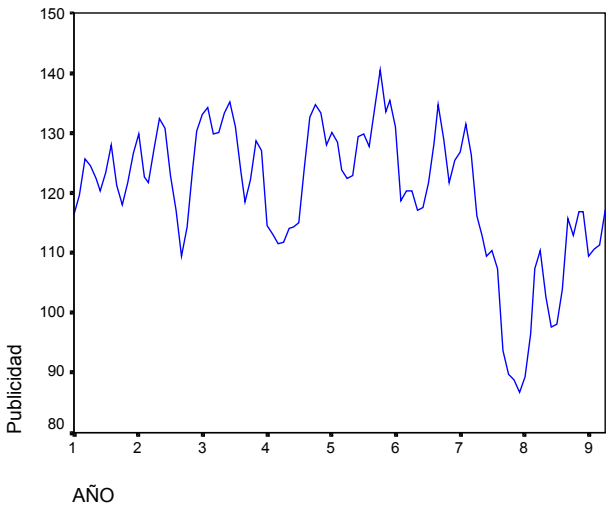


Fig. 5.23. Gráfico de secuencia de la serie Publicidad

- Se obtienen los correlogramas simple y parcial:

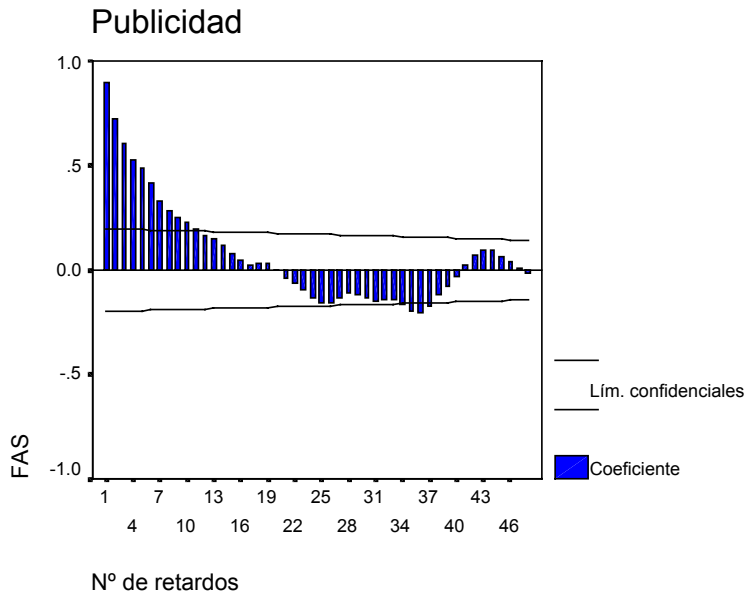


Fig. 5.24. Correlograma Simple de la serie Publicidad

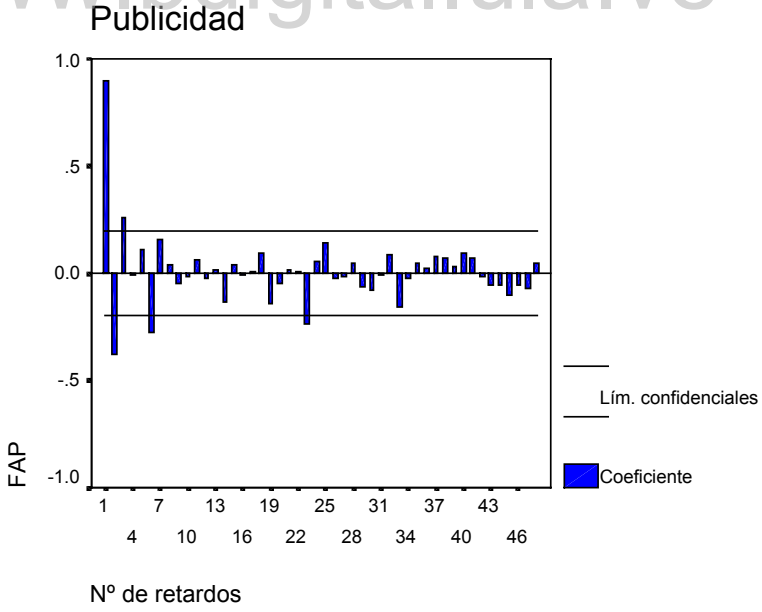


Fig. 5.25. Correlograma Parcial de la serie Publicidad

La serie no es estacionaria se debe aplicar 1era. diferencia en la parte regular.

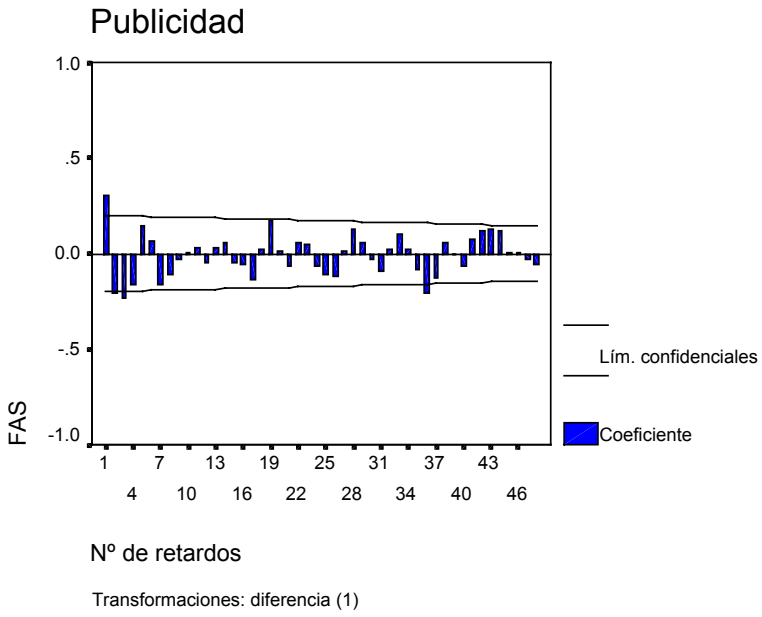


Fig. 5.26. Correlograma Simple con 1era. diferencia regular

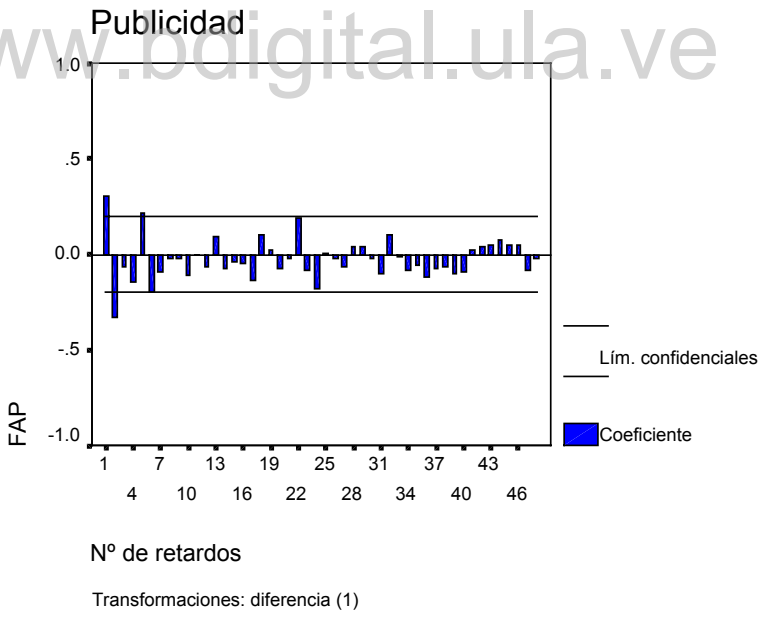


Fig. 5.27. Correlograma Parcial con 1era. diferencia regular

En la parte regular se observa un patrón AR y/o MA, y en la parte estacional no hay valores significativos en ninguno de los dos correlogramas, indicando ausencia de los parámetros.

- Estimación del modelo. Los datos se dividen en dos conjuntos: 80 observaciones para el ajuste y 20 para la predicción. El modelo con mejor ajuste fue: ARIMA(0,1,1) sin constante, cuyo parámetro es:

PARAMETRO	ESTIMACIÓN	T-RATIO	APPROX. PROB.
MA1	-0,56072117	-5,8611009	0,00000010

El modelo estimado es:

$$(1\text{-}B)y_t = (1+ 0{,}56072117\ B)\varepsilon_t \qquad (5.2)$$

- Adecuación del modelo. En la siguiente gráfica se representa la serie original y su ajuste:

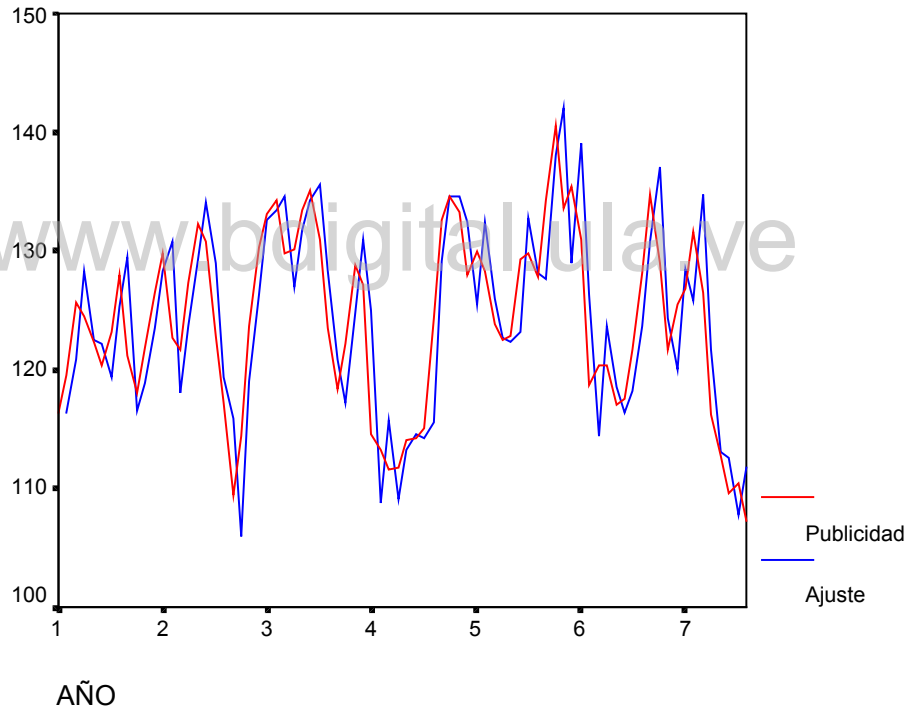


Fig. 5.28. Serie Publicidad original y ajustada. Periodo de entrenamiento: 80 datos

El ajuste es bastante bueno. A continuación se presentan los correlogramas para los residuos:

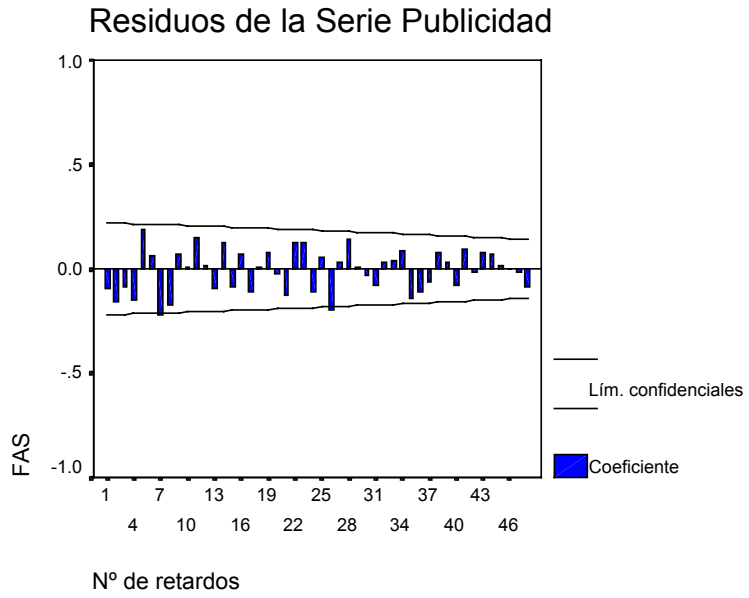


Fig. 5.29. Correlograma Simple de los residuos

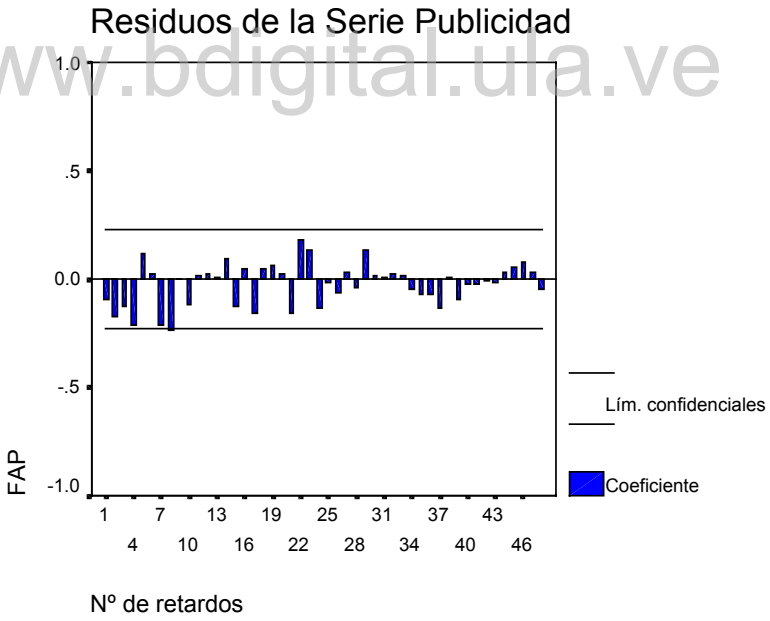


Fig. 5.30. Correlograma Parcial de los residuos

Se observan pocos valores significativamente diferentes de cero. La media de los residuos es -0,09, aproximadamente cero y la hipótesis de normalidad de los residuos, utilizando la prueba de Kolmogorov-Sminov, no se pudo rechazar con una significación

de 0,301. Según el análisis anterior puede decirse que existe una buena adecuación del modelo.

- Preblanqueo de x_t ($Z_t^{(x)}$)

$$\alpha_t = \varepsilon_t = \frac{(1-B)}{1+0{,}5607} Z_t^{(x)} \tag{5.3}$$

- Preblanqueo de y_t ($Z_t^{(y)}$)

$$\beta_t = \frac{(1-B)}{1+0{,}5607} Z_t^{(y)} \tag{5.4}$$

Etapa 2: Identificación de un MFT preliminar que describa a la serie Publicidad

- Se obtiene el gráfico de Correlación Cruzada, usando el paquete SAS:

```
Lag Covariance Correlation -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 10
 -2   6.350187    0.08581  |                .    |**  .                |
 -1   6.034884    0.08155  |                .    |**  .                |
  0   2.398283    0.03241  |                .    |*   .                |
  1 -19.790461   -0.26744  |                .*****|    .                |
  2  28.155537    0.38048  |                .    |********            |
  3  30.215283    0.40832  |                .    |********            |
  4  14.611108    0.19745  |                .    |****.                |
  5  -4.368075   -0.05903  |                .   *|    .                |
  6 -36.247664   -0.48984  |          **********|    .                |
  7 -17.276818   -0.23347  |               *****|    .                |
  8   1.679972    0.02270  |                .    |    .                |
  9   9.219688    0.12459  |                .    |**  .                |
 10  -3.799448   -0.05134  |                .   *|    .                |
 11   3.941615    0.05327  |                .    |*   .                |
 12  -6.884200   -0.09303  |                .  **|    .                |
 13  10.960898    0.14812  |                .    |*** .                |
 14  10.740801    0.14515  |                .    |*** .                |
 15  -8.027227   -0.10848  |                .  **|    .                |
```

Fig. 5.31. Gráfico de Correlación Cruzada

No se observan valores significativos antes del retraso cero. Se considera que la primera correlación significativamente diferente de cero ocurre en el retraso 2, por lo tanto b=2. La más alta correlación se presenta en el retraso 6. El patrón de descenso comienza inmediatamente después de este retraso, lo que indica que s=0. Se considera r=1 porque el patrón de descenso después del retraso b+s=6+0=6, tiene forma exponencial.

Así el MFT preliminar tiene la forma:

$$z_t^{(y)} = \mu + \frac{C}{1-\delta_1 B} B^2 z_t^{(x)} + \eta_t \qquad (5.5)$$

Otros posibles modelos fueron probados, siendo éste el mejor.

Etapa 3: Identificación de un modelo que describa al error y de un MFT final.

- Se obtienen los correlogramas simple y parcial de los residuos, y la correlación cruzada de los residuos con x_t.

Autocorrelation Plot of Residuals

```
Lag Covariance Correlation -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 10
  0    233.745    1.00000  |                    |********************|
  1    105.627    0.45189  |                .   |*********           |
  2 -38.033532   -0.16271  |              .  ***|     .              |
  3   -111.307   -0.47619  |          **********|     .              |
  4 -55.390373   -0.23697  |             .*****|     .              |
  5  -5.200041   -0.02225  |             .      |      .             |
  6  -9.136548   -0.03909  |             .     *|      .             |
  7 -37.154346   -0.15895  |             .   ***|      .             |
  8 -45.195224   -0.19335  |             .  ****|      .             |
  9   3.923050    0.01678  |             .      |      .             |
 10  36.974600    0.15818  |             .      |***   .             |
 11  60.695146    0.25966  |             .      |***** .             |
 12  28.749307    0.12299  |             .      |**    .             |
```

Fig. 5.32. Correlograma simple de los residuos

Partial Autocorrelations

```
Lag Correlation -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 10
  1    0.45189  |                .   |*********           |
  2   -0.46107  |           *********|    .               |
  3   -0.25599  |               *****|    .               |
  4    0.15265  |               .    |*** .               |
  5   -0.22109  |               .****|    .               |
  6   -0.26382  |               *****|    .               |
  7   -0.11365  |               .  **|    .               |
  8   -0.23981  |               *****|    .               |
  9   -0.02576  |               .   *|    .               |
 10   -0.15795  |               . ***|    .               |
 11    0.06035  |               .    |*   .               |
 12   -0.09906  |               .  **|    .               |
```

Fig. 5.33. Correlograma parcial de los residuos

Crosscorrelation Check of Residuals with Input X

To Lag	Chi Square	DF	Prob	Crosscorrelations					
5	46.77	4	0.000	-0.259	0.124	0.326	0.118	-0.410	-0.537
11	61.99	10	0.000	-0.249	0.197	0.272	0.198	0.004	0.011
17	66.49	16	0.000	0.130	-0.022	0.026	-0.077	-0.163	-0.114
23	70.42	22	0.000	0.010	0.061	0.139	0.157	-0.053	-0.070

Fig. 5.34. Correlación cruzada de los residuos con x_t

- No se cumple que la serie de entrada preblanqueada α_t sea estadísticamente independiente del error η_t, porque los valores de la probabilidad en las correlaciones cruzadas son cero y por lo tanto, se rechaza la hipótesis de independencia.

- En los correlogramas simple y parcial de los residuos se observa un patrón promedio móvil de orden 1 en la parte regular y un patrón autorregresivo de orden 1 en la parte estacional con periodo 3. Así, el modelo que describe a η_t es:

$$\eta_t = \frac{(1+\theta_1 B)}{1+\phi_1 B^3} a_t \tag{5.6}$$

Otros posibles modelos fueron probados, siendo éste el mejor.

- Se sustituye η_t en el MFT preliminar (ecuación 5.5) y se constituye el MFT final. La estimación de los parámetros se obtiene mediante el paquete SAS:

Conditional Least Squares Estimation

Parameter	Estimate	Approx. Std Error	T Ratio	Lag	Variable	Shift
MA1,1	-0.53403	0.10727	-4.98	1	Y	0
AR1,1	-0.53405	0.11124	-4.80	3	Y	0
SCALE1	1.97394	0.28903	6.83	0	X	2
DEN1,1	0.45978	0.11620	3.96	1	X	2

Fig. 5.35. Estimación de los parámetros del MFT mediante SAS

- Todos los parámetros resultaron ser significativos (T Ratio mayores que 2 en valor absoluto). La constante μ no resultó significativa, siendo entonces el MFT final el siguiente:

$$z_t^{(y)} = \frac{1{,}97394}{1-0{,}45978B} B^2 z_t^{(x)} + \frac{1-0{,}53403B}{1-0{,}53405B^3} a_t \tag{5.7}$$

- Las predicciones del MFT para la serie ventas, sus valores verdaderos y el error se presentan en el apéndice 4. En la fig. 5.35 se observa que la estimación proporcionada por el MFT, aunque está entre el rango de los valores verdaderos, es demasiado suave, no logrando captar la estructura de la serie y proporcionando valores bastante alejados de los originales.

  Los autores Bowerman y O'Connell (1993) analizaron estas series, utilizando para la estimación los 100 datos disponibles, y construyen un MFT diferente con el cual se obtiene unas predicciones que parecieran más realistas, pero al tomar los primeros 80 datos para el ajuste, el modelo cambia radicalmente.

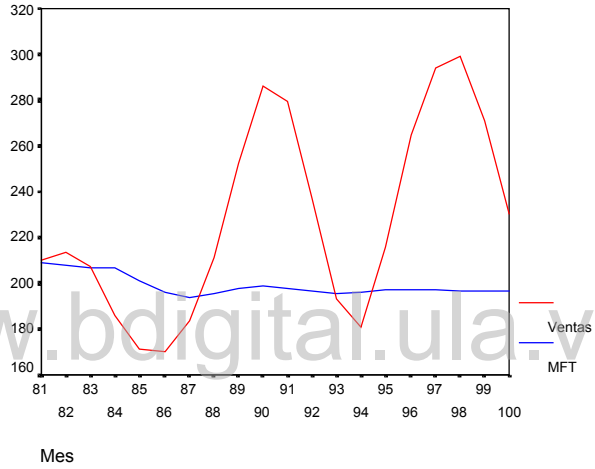


Fig. 5.36. Serie Original de Ventas y la predicción. Periodo: 20 últimos meses

5.2.2. Redes Neuronales

Se aplica la metodología especificada en el capítulo IV, de forma tal de predecir valores futuros de la serie ventas mensuales basándose en ciertos valores pasados y a la serie publicidad, utilizando el paquete MatLab 5.1.

1. Escalamiento de los datos

Los valores de ambas series de tiempo son transformados a valores entre 0 y 1, mediante una rutina computacional descrita en el apéndice 5 (a).

2. Patrones de Entrenamiento y Prueba

Los valores de las series de tiempo comprenden 100 meses. Estos datos se dividen en dos conjuntos:

Patrones de Entrenamiento: Del mes 1 al 80, lo que equivale al 80% de los datos.

Patrones de Prueba: Del mes 81 al 100, lo que equivale al 20% de los datos.

3. Topología de la RN

Todas la RN que serán probadas son de alimentación adelantada, totalmente conectadas, con una capa oculta, un nodo adicional tanto en la capa de entrada como en la capa oculta de valor constante 1 (sesgo) y función de activación logística en los nodos de la capa oculta y en el nodo de salida.

4. Determinación de las entradas (p)

La selección de los retrasos de entrada se realiza según las recomendaciones dadas en el capítulo IV, sección 4.2. Se considerarán, los siguientes retrasos como entradas para las RN a probar:

RN1. Los retrasos del 1 al 13 para ambas series de tiempo: Se consideran estos retrasos debido a la periodicidad de los datos.

RN2. Los retrasos 1, 2, 3, 5 y 6 de la serie publicidad, y los retrasos 1 y 2 de la serie ventas: Son los retrasos correspondientes a las correlaciones significativamente diferentes de cero, en los correlogramas simple y parcial usados en la metodología univariable ARIMA.

RN3. El retraso 1 de la serie publicidad y los retrasos 1 y 2 de la serie ventas: Son los retrasos utilizados en los modelos ARIMA encontrados.

RN4. Los retrasos 2, 3 y 6 de la serie publicidad, y los retrasos 1 y 2 de la serie ventas: Los retrasos de la serie publicidad son los correspondientes a las correlaciones cruzadas significativamente diferentes de cero, y los retrasos de la serie ventas son las correlaciones significativamente diferentes de cero, en los correlogramas simple y parcial usados en la metodología univariable ARIMA para esta serie.

RN5. El retraso 2 de la serie publicidad y el retraso 1 de la serie ventas: Son los retrasos considerados en el MFT.

RN6. Los retrasos 1, 2 y 3 de ambas series: como una prueba por ensayo y error.

5. Determinación del número de nodos de la capa oculta (q)

El número de nodos de la capa oculta se tomó en todos las RN como el promedio entre el número de entradas (p) y el número de salidas (1), para valores decimales se redondea. Así:

Red Neuronal Nº	Retrasos Publicidad	Retrasos Ventas	Nº de Entradas (p)	Nº de Nodos de la Capa Oculta (q)
RN1	Del 1 al 13	Del 1 al 13	26	13
RN2	1,2,3,5,6	1,2	7	4
RN3	1	1,2	3	2
RN4	2,3,6	1	4	2
RN5	2	1	2	1
RN6	1,2,3	1,2,3	6	3

Fig. 5.37. Tabla con las RN seleccionadas, nº de entradas y nº de nodos de la capa oculta

6. Algoritmo de entrenamiento

   Las seis RN se entrenaron con el algoritmo de retropropagación

7. Selección de los pesos iniciales

   Se generan los pesos iniciales 50 veces en cada una de las seis redes, mediante la función NWLOG (generador aleatorio Nguyen-Widrow para neuronas LOGSIG), y de los 50 resultados obtenidos se selecciona la red que obtenga el promedio mínimo entre la suma de cuadrados de los errores de entrenamiento (ajuste) y generalización (predicción). Para esto se desarrolló un procedimiento computacional descrito en el apéndice 2 (e)

8. Ajuste o entrenamiento de la RN

   - Fijación de los parámetros:
     - Número máximo de épocas: 2000
     - Error permitido de Convergencia: 0,01
     - Tasa de Aprendizaje: 0,01
     - Incremento en la tasa de aprendizaje: 1,05
     - Momento: 0,95

     Se hicieron pruebas variando estos parámetros, en general, se encontraron las menores suma de cuadrados de los errores de entrenamiento y generalización con estos valores.

- Con la ecuación general de cada una de las seis RN se generan los valores de la serie de tiempo ajustada, utilizando las entradas de los patrones de entrenamiento.

11. Predicción

Por medio de la ecuación general de cada una de las seis RN se generan las predicciones para la serie de tiempo, utilizando las entradas de los patrones de prueba.

12. Comparación con las diferentes RN

Se comparan las seis RN seleccionadas a través de los errores de ajuste y predicción:

RN Nº	Ajuste			Predicción		
	DER	AIC	SBC	EMA	RECMP	CORR
RN1	**3,9**	911,7	1.716,4	10,3	5,8	0,97
RN2	6,1	315,7	397,3	**6,3**	**3,8**	**0,98**
RN3	10,1	332,2	356,4	10,6	6,0	0,96
RN4	5,8	280,5	**329,0**	8,2	4,2	**0,98**
RN5	15,4	376,1	387,2	21,3	10,6	0,87
RN6	5,6	**280,3**	335,4	8,7	4,9	0,97

Fig. 5.38. Errores de ajuste y predicción para las seis RN seleccionadas

Se selecciona la RN2 por obtener la mejor predicción (menor EMA y RECMP, y la más alta correlación entre los valores verdaderos y la predicción de la RN). Podría considerarse la RN4, pues tiene el menor AIC y los demás valores están cerca de los encontrados para la RN2. En la RN2 se consideraron como entradas los retrasos que resultaron con correlaciones significativamente diferentes de cero en los correlogramas simple y parcial del análisis univariable de las series. La RN4 tiene como entradas los retrasos de la serie publicidad que resultaron significativamente diferentes de cero en las correlaciones cruzadas entre ambas series, así como los retrasos correspondientes a las correlaciones significativamente diferentes de cero en los correlogramas simple y parcial de la serie ventas.

5.2.3. Análisis Comparativo: Ajuste y Predicción

En la siguiente tabla se presentan los errores de ajuste (entrenamiento) y predicción (prueba), tanto para la mejor RN obtenida como para el MFT seleccionado:

RN Nº	Ajuste			Predicción		
	DER	AIC	SBC	EMA	RECMP	CORR
RN2	6,1	315,7	397,3	6,3	3,8	0,98
MFT	18,26	463,25	544,83	39,35	19,18	-0,22

Fig. 5.39. Errores de ajuste y predicción para la mejor RN y el MFT

Se observa claramente la superioridad de la RN. Incluso si se compara el MFT con la RN5 (ver fig. 5.37), cuyas entradas son los retrasos correspondientes al MFT, esta última también resulta ser superior.

Para ilustrar este punto, a continuación se presenta el gráfico de los valores verdaderos de la serie ventas, la predicción con el MFT y la RN2:

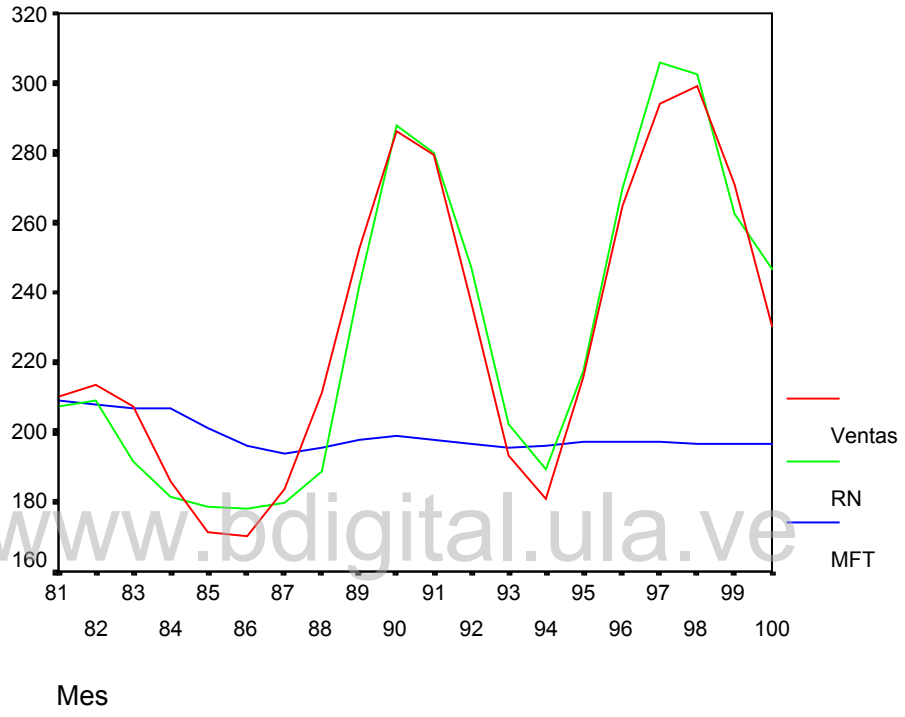


Fig. 5.40. Gráfico del periodo de predicción: valores verdaderos de la serie ventas, RN y MFT

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

La reflexión sobre las actividades realizadas y sus resultados es sin duda lo que proporciona el aprendizaje. Por esta razón este último capítulo es fundamental. Aquí se discuten los logros alcanzados en esta investigación, las ventajas y desventajas de las técnicas bajo estudio y se proponen futuras investigaciones que podrían resultar de interés en la predicción con series de tiempo.

6.1. CONCLUSIONES

Las redes neuronales (RN) como herramienta para la predicción han sido de interés recientemente para muchos autores. Una de sus limitaciones para alcanzar un uso generalizado es el establecimiento de la arquitectura de la red. En este trabajo se proporciona una metodología o conjunto de pasos que pudieran servir de guía para tener éxito en la predicción con series de tiempo.

El punto más álgido en el diseño del modelo de la red son las entradas o retrasos de la serie. En forma empírica, se llegó a la conclusión de que puede utilizarse la metodología ARIMA como una herramienta de preprocesamiento de datos, considerando como entradas los retrasos involucrados en el modelo proporcionado por esta metodología. En el caso del MFT no podemos llegar a la misma conclusión, seguramente, porque en el ejemplo estudiado no se logra captar el patrón de comportamiento del proceso. Sin embargo, herramientas muy útiles resultaron ser los correlogramas simple y parcial, y el gráfico de correlación cruzada para identificar los retrasos de interés candidatos a ser entradas.

En el ejemplo estudiado para el caso univariable, el modelo ARIMA proporcionó un mejor ajuste, mientras que la RN con un ajuste similar, pero cuantitativamente inferior, logra mejores predicciones. El modelo ARIMA no logra captar el cambio de patrón que ocurre al final de la serie, pero existe una herramienta que permite corregir esto denominada Análisis de Intervención (ver Box, Jenkins y Reinsel, 1994). En contraste, las RN si logran captar este cambio. Una razón para justificar este fenómeno podría ser la no linealidad del modelo de RN, y la linealidad del modelo ARIMA.

En el caso bivariable la RN resulta ser muy superior al MFT, tanto en el ajuste como en la predicción. El hecho de dividir el conjunto de datos en 80% de los datos para el ajuste y el 20% restante reservarlo para predicción afecta al MFT, que cambia radicalmente con relación al modelo obtenido por Bowerman y O'Connell (1994) utilizando el 100% de los datos para el ajuste. Esto podría ser un indicativo de la sensibilidad del MFT a la cantidad de información suministrada, y de la robustez de la RN que logra captar la estructura del proceso a pesar de que le sean suprimidos en su entrenamiento 20 datos (equivalente al 20%).

6.2. RECOMENDACIONES

Una mayor experimentación con nuevos casos de estudio podría ser recomendable para corroborar lo expuesto en la sección anterior.

El algoritmo de aprendizaje utilizado es el muy conocido algoritmo de retropropagación, pero sería de mucho interés experimentar con nuevos algoritmos y modelos de red que pudieran ser útiles para predicciones con series de tiempo, tales como la RN función de base radial y la red neurodifusa. Esto permitiría establecer comparaciones entre diferentes tipos de redes y probablemente mejorar las predicciones.

Debido a la experiencia necesaria para determinar la arquitectura apropiada de una RN sería muy provechoso la creación de un sistema experto que ayude a identificar las entradas más adecuadas para la red, mediante el uso de los correlogramas simple y parcial, el número de capas ocultas y sus nodos, en fin, todos los aspectos que involucra el diseño de una RN para predicción con series de tiempo.

REFERENCIAS BIBLIOGRÁFICAS

- Bowerman, B. y O' Connel, R. (1993). *Forecasting and time series: an applied approach* (3ª ed.). California, USA: Duxbury Press.
- Box, G., Jenkins, G. y Reinsel, G. (1994). *Time series analysis, forecasting and control.* (3ª ed.). New Jersey, USA: Prentice Hall.
- Chatfield, C. (1978). *The analysis of time series: theory and practice.* Londres: Chapman and Hall.
- Colina, E. y Rivas, F. (1998). *Introducción a la inteligencia artificial*. Mérida, Venezuela: Universidad de Los Andes.
- Faraway, J. y Chatfield, C. (1998). Time series forecasting with neural networks: a comparative study using the airline data. Applied Statistic, 47 (2), 231-250.
- Gujarati, D. (1997). *Econometría* (3ª ed.). Santa Fé de Bogotá, Colombia: McGraw-Hill.
- Hagan, M., Demuth, H. y Beale, M. (1996). *Neural network design*. Boston, USA: PWS Publishing Company.
- Haykin, S. (1994). *Neural networks: a comprehensive foundation*. USA: Macmillan College Publishing Company.
- Hill, T., O'Connor, M. y Remus, W. (1996). Neural network models for time series forecasts. Management Science, 42 (7), 1082-1092.
- Instituto Nacional de Estadística (2001). [On line]. Disponible en: http:\www.ine.es
- Maddala, G. (1996). *Introducción a la econometría*. (2ª ed.). México: Prentice may Hispanoamericana.
- Makridakis, S. y Wheelwright, S. (1994). *Manual de técnicas de pronósticos*. México: Limusa Noriega Editores.

- Wedding II y Cios. (1996). Time series forecasting by combining RBF network, certainty factors, and the Box-Jenkins model. Neurocomputing, 10, 149-168.

- Wong, F. (1991). Time series forecasting using backpropagation neural networks. Neurocomputing, 2, 147-159.

APÉNDICES

APÉNDICE 1

Valores originales de la serie nacimientos, la predicción ARIMA y el error

Año	Mes	Nacimiento	Predicción	Error
1992	1	32866	32599.45	266.55
1992	2	31622	29748.07	1873.93
1992	3	32988	33019.54	-31.54
1992	4	32996	32294.48	701.52
1992	5	33507	33694.88	-187.88
1992	6	31414	31799.13	-385.13
1992	7	34211	33354.94	856.06
1992	8	33565	32820.47	744.53
1992	9	34349	33271.27	1077.73
1992	10	33828	33292.19	535.81
1992	11	32350	31459.72	890.28
1992	12	33051	31909.19	1141.81
1993	1	32870	31928.81	941.19
1993	2	29386	29136.09	249.91
1993	3	34073	32340.26	1732.74
1993	4	31902	31630.11	271.89
1993	5	33307	33001.70	305.30
1993	6	31815	31144.96	670.04
1993	7	33422	32668.76	753.24
1993	8	32256	32145.28	110.72
1993	9	33335	32586.81	748.19
1993	10	32528	32607.30	-79.30
1993	11	30208	30812.52	-604.52
1993	12	30684	31252.74	-568.74
1994	1	30653	31271.96	-618.96
1994	2	28417	28536.69	-119.69
1994	3	32601	31674.95	926.05
1994	4	31737	30979.41	757.59
1994	5	32804	32322.78	481.22
1994	6	30658	30504.24	153.76
1994	7	31995	31996.69	-1.69
1994	8	30829	31483.98	-654.98
1994	9	30407	31916.42	-1509.42
1994	10	30529	31936.49	-1407.49
1994	11	29255	30178.64	-923.64
1994	12	30263	30609.81	-346.81
1995	1	30304	30628.63	-324.63
1995	2	27142	27949.63	-807.63
1995	3	30976	31023.33	-47.33
1995	4	29917	30342.10	-425.10
1995	5	31392	31657.83	-265.83

1995	6	30140	29876.70	263.30
1995	7	31955	31338.44	616.56
1995	8	30651	30836.28	-185.28
1995	9	30775	31259.83	-484.83
1995	10	31103	31279.49	-176.49
1995	11	29238	29557.80	-319.80
1995	12	29876	29980.09	-104.09
1996	1	30093	29998.53	94.47
1996	2	27826	27374.64	451.36
1996	3	31059	30385.11	673.89
1996	4	29835	29717.89	117.11
1996	5	31294	31006.56	287.44
1996	6	29833	29262.07	570.93
1996	7	30993	30693.74	299.26
1996	8	29317	30201.91	-884.91
1996	9	30077	30616.75	-539.75
1996	10	30670	30636.00	34.00
1996	11	30166	28949.73	1216.27
1996	12	31463	29363.34	2099.66
1997	1	31210	29381.39	1828.61
1997	2	28562	26811.49	1750.51
1997	3	30692	29760.02	931.98
1997	4	31650	29106.53	2543.47
1997	5	32585	30368.69	2216.31
1997	6	30229	28660.08	1568.92
1997	7	32667	30062.31	2604.69
1997	8	29893	29580.59	312.41
1997	9	30624	29986.90	637.10
1997	10	30889	30005.75	883.25
1997	11	29366	28354.17	1011.83
1997	12	30668	28759.27	1908.73
1998	1	29883	28776.95	1106.05
1998	2	27481	26259.92	1221.08
1998	3	30141	29147.79	993.21
1998	4	31260	28507.75	2752.25
1998	5	31267	29743.94	1523.06
1998	6	29260	28070.48	1189.52
1998	7	30473	29443.86	1029.14
1998	8	29750	28972.06	777.94
1998	9	31599	29370.00	2229.00
1998	10	30586	29388.47	1197.53
1998	11	29430	27770.87	1659.13
1998	12	30800	28167.63	2632.37
1999	1	30918	28184.95	2733.05
1999	2	28461	25719.69	2741.31
1999	3	31948	28548.16	3399.84
1999	4	30959	27921.28	3037.72

1999	5	32287	29132.04	3154.96
1999	6	30331	27493.01	2837.99
1999	7	32878	28838.13	4039.87
1999	8	31855	28376.04	3478.96
1999	9	33128	28765.79	4362.21
1999	10	31869	28783.88	3085.12
1999	11	30874	27199.56	3674.44
1999	12	32301	27588.16	4712.84

APÉNDICE 2

Rutinas desarrolladas en MatLab para la predicción con RN (caso univariable)

a) Programa Principal

```
%Programa Principal para la preparación, escalamiento, entrenamiento y
prueba de RN para predicción;
clc
clear all
display('Preparación de la serie Nacimientos');
display(' ');
load nac.prn;
%Escalamiento de los datos entre 0 y 1;
mini=min(nac);
R=max(nac)-min(nac);
nac01=(nac-mini)/R;
%Retraso de los datos t=13;
N=nac01; %si se quieren los datos escalados;
retrasos;
nac13=B;
%Conjunto de Entrenamiento y Prueba;
CE=nac13(1:371,:); % 32 años;
CP=nac13(372:467,:); % 8 años;
conjuntosEP; %Preparación de los conjuntos de entrenamiento y prueba;
Lectura; %Lectura de los datos de entrada y salida para la construcción y
prueba del modelo;
Entren; %Entrenamiento de las 50 RN y selección de la mejor;
errorAP; %Cálculo de los errores de ajuste y predicción;
```


b) Programa Retrasos

```
%Retraso de los datos t=13;
[f,c]=size(N);
for j=1:14;
      for i=j:f;
      A(i-(j-1),j)=N(i);
   end;
end;
B=A(1:f-13,1:14);
```

c) Programa ConjuntosEP

```
%Prepara las entradas y salidas de los conjuntos de entrenamiento y
prueba;
display('Preparado Conjuntos de entrenamiento, xe entrada, ye salida');
xe1=CE(:,1:13); % Conjunto 1: RN(1:13);
xe2=[CE(:,1) CE(:,12:13)]; %Conjunto 2 (ARIMA): RN(1,12,13);
xe3=[CE(:,1:2) CE(:,4) CE(:,9:13)]; %Conjunto 3 (Correlogramas):
RN(1:2,4,9:13);
xe4=[CE(:,1:2) CE(:,12:13)]; %Conjunto 4: RN(1,2,12,13);
xe5=CE(:,1:4); % Conjunto 5: RN(1:4);
xe6=[CE(:,1) CE(:,12)]; %Conjunto 6: RN(1,12);
xe7=[CE(:,1) CE(:,4) CE(:,8) CE(:,12)]; %Conjunto 7: RN(1,4,8,12);
```

```
xe8=[CE(:,1) CE(:,3) CE(:,6) CE(:,9) CE(:,12)]; %Conjunto 8:
RN(1,3,6,9,12);
xe9=[CE(:,1) CE(:,6) CE(:,12)]; %Conjunto 9: RN(1,6,12);
xe10=CE(:,1:3); % Conjunto 10: RN(1:3);
xe11=[CE(:,3) CE(:,5) CE(:,9)]; %Conjunto 11: RN(3,5,9), generado por los
pesos;
ye=CE(:,14); % Salida
display('Preparado Conjuntos de prueba, xp entrada, yp salida');
xp1=CP(:,1:13); % Conjunto 1: RN(1:13);
xp2=[CP(:,1) CP(:,12:13)]; %Conjunto 2 (ARIMA): RN(1,12,13);
xp3=[CP(:,1:2) CP(:,4) CP(:,9:13)]; %Conjunto 3 (Correlogramas):
RN(1:2,4,9:13);
xp4=[CP(:,1:2) CP(:,12:13)]; %Conjunto 4: RN(1,2,12,13);
xp5=CP(:,1:4); % Conjunto 5: RN(1:4);
xp6=[CP(:,1) CP(:,12)]; %Conjunto 6: RN(1,12);
xp7=[CP(:,1) CP(:,4) CP(:,8) CP(:,12)]; %Conjunto 7: RN(1,4,8,12);
xp8=[CP(:,1) CP(:,3) CP(:,6) CP(:,9) CP(:,12)]; %Conjunto 8:
RN(1,3,6,9,12);
xp9=[CP(:,1) CP(:,6) CP(:,12)]; %Conjunto 9: RN(1,6,12);
xp10=CP(:,1:3); % Conjunto 10: RN(1:3);
xp11=[CP(:,3) CP(:,5) CP(:,11)]; %Conjunto 11: RN(3,5,9), generado por
los pesos;
yp=CP(:,14); % Salida
save dnac xe1 xe2 xe3 xe4 xe5 xe6 xe7 xe8 xe9 xe10 xe11 ye xp1 xp2 xp3
xp4 xp5 xp6 xp7 xp8 xp9 xp10 xp11 yp;
```

d) Programa Lectura


```
%Lectura de los datos de entrada y salida para la construcción y prueba
del modelo;
Inp = input('Indique el nombre del archivo de entrada para el
entrenamiento: ');
clc
disp('Presione cualquier tecla al estar listo.');
pause
echo off
save Input.mat Inp;
Out = input('Indique el nombre del archivo de salida para el
entrenamiento: ');
clc
disp('Presione cualquier tecla al estar listo.');
pause
echo off
save Output.mat Out;
Inp = input('Indique el nombre del archivo de entrada para la prueba: ');
clc
disp('Presione cualquier tecla al estar listo.');
pause
echo off
save TestInp.mat Inp;
Out = input('Indique el nombre del archivo de salida para la prueba: ');
clc
disp('Presione cualquier tecla al estar listo.');
pause
echo off
save TestOut.mat Out;
```

e) Programa Entren

```
clc;
fprintf('\n\n   ENTRENAMIENTO mediante Backpropagation\n\n')
load Output.mat;
load Input.mat;
T1='logsig';
T2='tansig';
[M,N]=size(Inp');
P=Inp'; % Patrones de Entrada;
T=Out'; %Patrones de Salida;
number_inputs=M; % Número de Nodos de entrada;
number_patterns=N; % Número de patrones de entrada;
[R,C]=size(P);
[R1,C1]=size(T);
S1=fix((R+R1)/2); % S1: número de neuronas ocultas;
S2=R1;
% Entrenamiento de la red;
% Fijación de parametros por defecto;
% desired error: error_goal
% maximumm number of epochs: max_epochs
% learning rate: lr
disp_freq =100;
max_epochs=2000;
error_goal=0.01;
lr = 0.01;
lr_inc = 1.05;
lr_dec = 0.7;
mom_const = 0.95;
err_ratio = 1.04;
F1 ='logsig';
F2 = 'logsig';
% Despliegue de los paranetros por defecto;
  fprintf('\n PARAMETROS PRESELECCIONADOS:\n')
  fprintf('    Frecuencia de resultados: %d \n    Número máximo de pasos
(epochs): %d \n',disp_freq,max_epochs)
  fprintf('    Error permitido: %f \n    Tasa de Aprendizaje: %f
\n',error_goal,lr)
  fprintf('    Incremento del aprendizaje: %f \n    Momento:
%f\n',lr_inc,mom_const)
  fprintf('    Función de activación para la capa oculta: %s \n',F1)
  fprintf('    Función de activación para la capa de salida: %s\n',F2)
  fprintf('    El número de nodos en la capa oculta sugerido es:
%d\n',S1)
par = menu('Selección de los parámetros de entonación de la red',...
           'Fijar nuevos prámetros',...
           'Establecer los parámetros sugeridos');
disp('');

if par ==1
% Input the new parameters
  S1 = input(' Número de nodos ocultos ');
  if size(S1) == [0,0]
     S1 = fix((R+R1)/2);
  end
  disp_freq=input(' Frecuencia de la presentación de resultados: ');
  if size(disp_freq) == [0,0]
```

```
    disp_freq = 100;
  end
  max_epochs=input(' Máximo número de pasos(epochs): ');
  if size(max_epochs) == [0,0]
    max_epochs = 2000;
  end
  error_goal = input(' Error permitido de convergencia: ');
  if size(error_goal) == [0,0]
    error_goal = 0.01;
  end
  lr = input(' Tasa de aprendizaje: ');
  if size(lr) == [0,0]
    lr = 0.01;
  end
  lr_inc = input(' Incremento de la tasa de aprendizaje: ');
  if size(lr_inc) == [0,0]
    lr_inc = 1.05;
  end
  mom_const= input(' Nuevo momento: ');
  if size(mom_const) == [0,0]
    mom_const = 0.95;
  end
  fprintf(' Typo de función de activación: tansig (-1,1), logsig
(0,1)\n');
  F1 = input(' Función de activación para la capa oculta: ', 's');
  if F1 ~= T1 | F1 ~= T2
    F1='logsig';
  end
  F2 = input(' Función de activación para la capa de salida: ','s');
  if F2 ~= T1 | F2 ~= T2
    F2 = 'logsig';
  end
end
% Se generan 50 veces los pesos iniciales, y se selecciona la mejor red;
for i=1:50
 % Generación de pesos aleatorios;
 % NWLOG Nguyen-Widrow random generator for LOGSIG neurons.
   [W1,B1]=nwlog(S1,R);
   [W2,B2]=nwlog(S2,S1);
   TP = [disp_freq max_epochs error_goal lr lr_inc lr_dec mom_const
err_ratio];
   [NW1,NB1,NW2,NB2,epoch,error] = trainbpx(W1,B1,F1,W2,B2,F2,P,T,TP);
   n_col=(S1*R)+S1+S1+1; % N° de columnas de Matriz50: n° de pesos
   Matriz50(i,1:n_col)=[reshape(NW1',1,S1*R) reshape(NB1,1,S1)
reshape(NW2,1,S1) NB2];
   Ajuste=logsig(NW2*logsig(NW1*P,NB1),NB2); %Valores escalados del
ajuste de la RN
   SSE_A=sumsqr(T-Ajuste); %Suma del error cuadrático del ajuste en
relación al objetivo
   load TestInp.mat;
   load TestOut.mat
   Predic=logsig(NW2*logsig(NW1*Inp',NB1),NB2);
   SSE_P=sumsqr(Out'-Predic); %Suma del error cuadrático de la predicción
en relación al verdadero valor
   Mat_SSE(i,1)=SSE_A;
   Mat_SSE(i,2)=SSE_P;
   Mat_SSE(i,3)=(SSE_A+SSE_P)/2;
```

```
end
[M_SSE,i]=min(Mat_SSE(:,3));
fprintf(' La RN con promedio mínimo entre el error de ajuste y predicción
es la #: %d \n',i)
fprintf(' Error de Ajuste SSE_A = %f \n',Mat_SSE(i,1))
fprintf(' Error de Predicción SSE_P = %f \n',Mat_SSE(i,2))
%pesos en forma lineal y reconversión de las matrices de pesos
VW1=Matriz50(i,1:S1*R);
NW1=(reshape(VW1,R,S1))';
VB1=Matriz50(i,(S1*R)+1:(S1*R)+S1);
NB1=VB1';
NW2=Matriz50(i,(S1*R)+S1+1:(S1*R)+2*S1);
NB2=Matriz50(i,n_col);
Ajuste=logsig(NW2*logsig(NW1*P,NB1),NB2);
Predic=logsig(NW2*logsig(NW1*Inp',NB1),NB2);
save RN.mat NW1 NB1 NW2 NB2 Ajuste Predic;
```

f) **Programa ErrorAP**

```
load RN;
load nac.prn;
ajusteN=(Ajuste*(max(nac)-min(nac)))+min(nac);
[mE,nE]=size(Ajuste);
predicN=(Predic*(max(nac)-min(nac)))+min(nac);
[mP,nP]=size(Predic);
nacE=nac(1:371,:); % 32 años;
nacP=nac(372:467,:); % 8 años;
figure;
hold on
plot(ajusteN,'r');
plot(nacE,'b');
hold off;
figure;
hold on
plot(predicN,'r');
plot(nacP,'b');
hold off;
display('Errores de Ajuste');
errorA=nacE-ajusteN';
S=sum(errorA.^2) %Suma cuadratica del error;
DER=sqrt(S/nE) %Desviación Estándar Residual;
r=(number_inputs*S1)+S1+S1+1; %S1:# de nodos ocultos y r: # parametros;
AIC=nE*log(S/nE)+(2*r)
SBC=nE*log(S/nE)+(r*log(nE))
CorrA=corrcoef(ajusteN,nacE); %Correlación entre el valor observado y el
ajuste;
display('Errores de Predicción');
errorP=nacP-predicN';
EMA=sum(abs(errorP))/nP %Error Medio Absoluto;
RECMP=sqrt((sum((errorP./nacP).^2))/nP)*100 %Raíz del Error Cuadrado
Medio Porcentual;
CorrP=corrcoef(predicN',nacP); %Correlación entre el valor observado y la
predicción;
save Resultados.mat NW1 NB1 NW2 NB2 ajusteN predicN S DER AIC SBC CorrA
EMA RECMP CorrP;
```

APÉNDICE 3

Programa en SAS para la construcción del modelo de función de transferencia

```
DATA PUBVEN;
INFILE 'C:\misdoc~1\tesis\pubven80.prn';
INPUT X Y @@;
* ESTUDIO PRELIMINAR DE X QUE SE TRANSFIERE A Y;
PROC ARIMA;
*IDENTIFY VAR=X; *REQUIRE UNA DIFERENCIA REGULAR;
IDENTIFY VAR=X(1); *Y SE LE APLICA ESTA PRIMERA DIFERENCIA TAMBIEN A Y;
IDENTIFY VAR=Y(1);

*ESTIMACION Y AJUSTE DEL MODELO DE X(1);
PROC ARIMA;
IDENTIFY VAR=X(1);
ESTIMATE Q=1 NOCONSTANT PRINTALL PLOT;

*PREBLANQUEO Y CORRELACION CRUZADA;
PROC ARIMA;
IDENTIFY VAR=X(1) NOPRINT;
ESTIMATE Q=1 NOCONSTANT;
IDENTIFY VAR=Y(1) CROSSCOR=(X(1)) NLAG=20;
ESTIMATE INPUT=(2$(0)/(1)X) NOCONSTANT PRINTALL
      ALTPARM MAXIT=30 BACKLIM=-3 PLOT;

*MFT FINAL;
ESTIMATE Q=1 SP=1 PERIOD=3 INPUT=(2$(0)/(1)X)  NOCONSTANT PRINTALL
      ALTPARM MAXIT=30 BACKLIM=-3 PLOT;
FORECAST LEAD=20;
PROC PRINT;
RUN.
```

APÉNDICE 4

Valores de la serie original de ventas, predicción usando el MFT y cálculo del error

Mes	Ventas	Predicción	Error
81	209.94	221.43	-11.49
82	213.3	245.43	-32.13
83	207.19	266.14	-58.95
84	186.13	284.31	-98.18
85	171.2	291.77	-120.57
86	170.33	302.73	-132.4
87	183.69	314.36	-130.67
88	211.3	320.37	-109.07
89	252.66	323.48	-70.82
90	286.2	325.08	-38.88
91	279.45	325.9	-46.45
92	237.06	326.33	-89.27
93	193.4	326.55	-133.15
94	180.79	326.67	-145.88
95	215.73	326.72	-110.99
96	264.98	326.75	-61.77
97	294.07	326.77	-32.7
98	299.08	326.78	-27.7
99	271.1	326.78	-55.68
100	230.56	326.78	-96.22

APÉNDICE 5

Rutinas desarrolladas en MatLab para la predicción con RN (caso bivariable)

a) Programa PrincipalXY

Los programas Retrasos, Lectura y Entren se especifican en el Apéndice 2

```
%Programa Principal para la preparación, escalamiento, entrenamiento y
prueba de RN para predicción con dos series de tiempo: Publicidad y
Ventas;
clc
clear all
load pub_ven.prn;
datos=pub_ven;
X=datos(:,1);
Y=datos(:,2);
display('Preparación de la serie X');
display(' ');
%Escalamiento de los datos entre 0 y 1;
mini=min(X);
R=max(X)-min(X);
X01=(X-mini)/R;
display('Preparación de la serie Y');
display(' ');
%Escalamiento de los datos entre 0 y 1;
mini=min(Y);
R=max(Y)-min(Y);
Y01=(Y-mini)/R;
%Retraso de la serie X escalada t=13;
N=X01;
retrasos;
X13=B;
%Retraso de la serie Y escalada t=13;
N=Y01;
retrasos;
Y13=B;
% C es el conjunto de los retrasos de X y Y;
C=[X13 Y13];
%Conjunto de Entrenamiento y Prueba;
[nf,nc]=size(datos);
r=13;
n3=nf-r;
d20=round(0.2*nf)
n2=n3-d20+1;
n1=n2-1;
CE=C(1:n1,:); % 80%;
CP=C(n2:n3,:); % 20%;
conjuntosEP_XY; %Preparación de los conjuntos de entrenamiento y prueba;
Lectura; %Lectura de los datos de entrada y salida para la construcción y
prueba del modelo;
Entren; %Entrenamiento de las 50 RN y selección de la mejor;
errorAP_XY; %Cálculo de los errores de ajuste y predicción;
```

b) Programa ConjuntosEP_XY

```
%Prepara las entradas y salidas de los conjuntos de entrenamiento y
prueba;
display('Preparando Conjuntos de entrenamiento, xe entrada, ye salida');
xe1=[CE(:,1:13) CE(:,15:27)] ; % Conjunto 1: 13 retrasos de X y 13
retrasos de Y;
xe2=[CE(:,8:9) CE(:,11:13) CE(:,26:27)]; % Conjunto 2: Correlogramas
simple y parcial. Retrasos de X:1,2,3,5,6 y retrasos Y:1,2;
xe3=[CE(:,13) CE(:,26:27)]; % Conjunto 3:ARIMA, retraso 1 de X y 1 y 2 de
Y. MA(1) AR(2)
xe4=[CE(:,8) CE(:,11:12) CE(:,26:27)]; % Conjunto 4: CC. Retrasos 2,3 y 6
de X y 1 y 2 de Y.
xe5=[CE(:,12) CE(:,27)]; % Conjunto 5:MFT. Retraso 2 de X y 1 de Y
xe6=[CE(:,11:13) CE(:,25:27)]; %Conjunto 6: retrasos 1,2 y 3 de X, y 1,2
y 3 de Y.
ye=CE(:,28); % Salida
display('Preparando Conjuntos de prueba, xp entrada, yp salida');
xp1=[CP(:,1:13) CP(:,15:27)] ; % Conjunto 1: 13 retrasos de X y 13
retrasos de Y;
xp2=[CP(:,8:9) CP(:,11:13) CP(:,26:27)]; % Conjunto 2: retrasos de
X:1,2,3,5,6 y retrasos Y:1,2;
xp3=[CP(:,13) CP(:,26:27)]; % Conjunto 3:ARIMA, retraso 1 de X y 1 y 2 de
Y. MA(1) AR(2)
xp4=[CP(:,8) CP(:,11:12) CP(:,26:27)]; % Conjunto 4: CC. Retrasos 2,3 y 6
de X y 1 y 2 de Y.
xp5=[CP(:,12) CP(:,27)]; % Conjunto 5:MFT. Retraso 2 de X y 1 de Y
xp6=[CP(:,11:13) CP(:,25:27)]; %Conjunto 6: retrasos 1,2 y 3 de X, y 1,2
y 3 de Y.
yp=CP(:,28); % Salida
```


c) Programa ErrorAP_XY

```
load RN;
ajusteN=(Ajuste*(max(Y)-min(Y)))+min(Y);
[mE,nE]=size(Ajuste);
predicN=(Predic*(max(Y)-min(Y)))+min(Y);
[mP,nP]=size(Predic);
h=nf-d20;
YE=Y(14:h,:); % 80%;
YP=Y(h+1:nf,:); % 20%;
figure;
hold on
plot(ajusteN,'r');
plot(YE,'b');
hold off;
figure;
hold on
plot(predicN,'r');
plot(YP,'b');
hold off;
display('Errores de Ajuste');
errorA=YE-ajusteN';
S=sum(errorA.^2) %Suma cuadratica del error;
DER=sqrt(S/nE) %Desviación Estándar Residual;
r=(number_inputs*S1)+S1+S1+1; %S1:# de nodos ocultos y r: # parametros;
AIC=nE*log(S/nE)+(2*r)
```

```
SBC=nE*log(S/nE)+(r*log(nE))
CorrA=corrcoef(ajusteN,YE) %Correlación entre el valor observado y el
ajuste;
display('Errores de Predicción');
errorP=YP-predicN';
EMA=sum(abs(errorP))/nP %Error Medio Absoluto;
RECMP=sqrt((sum((errorP./YP).^2))/nP)*100 %Raíz del Error Cuadrado Medio
Porcentual;
CorrP=corrcoef(predicN',YP) %Correlación entre el valor observado y la
predicción;
save Resultados.mat NW1 NB1 NW2 NB2 ajusteN predicN S DER AIC SBC CorrA
EMA RECMP CorrP;
```

GLOSARIO DE ACRÓNIMOS

AIC: Criterio de Información de Akaike

AR: Proceso Autorregresivo

ARIMA: Proceso Autorregresivo Integrado de Promedio Móvil

ARMA: Proceso Autorregresivo y de Promedio Móvil

CC: Correlación Cruzada

CORR: Correlación

DER: Desviación Estándar Residual

EMA: Error Medio Absoluto

FA: Función de Activación

FAP: Función de Autocorrelación Parcial

FAS: Función de Autocorrelación Simple

FBR: Función de Base Radial

GL: Grados de Libertad

IA: Inteligencia Artificial

MA: Proceso de Promedio Móvil

MFT: Modelo de Función de Transferencia

RECMP: Raíz del Error Cuadrático Medio Porcentual

RN: Redes Neuronales

RNA: Red Neuronal Artificial

SBC: Criterio Bayesiano de Schwarz

SCEE: Suma Cuadrática del Error de Entrenamiento

SCEG: Suma Cuadrática del Error de Generalización