ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Instituto de Ciencias Matemáticas

Ingeniería en Estadística e Informática

"Pronóstico de ventas: comparación de la precisión de la predicción con diferentes métodos"

TESIS DE GRADO

Previa a la obtención del Título de:

INGENIERO EN ESTADÍSTICA E INFORMÁTICA

Presentada por:

Andrés Guillermo Abad Robalino

GUAYAQUIL - ECUADOR

AÑO

2005

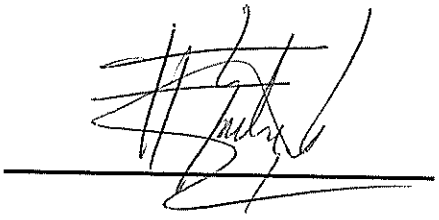
AGRADECIMIENTO

A todas las personas que de una u otra manera colaboraron en la realización de este trabajo, a mi director de tesis el Msc. Fernando Sandoya.

DEDICATORIA

A DIOS
A MIS PADRES
A MIS HERMANOS
A MI FAMILIA

TRIBUNAL DE GRADUACIÓN



MSC. FERNANDO SANDOYA
DIRECTOR DE TESIS

MAT. JHON RAMÍREZ
PRESIDENTE

ING. ENRIQUE BAYOT
VOCAL

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta tesis de grado, me corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL"

ANDRÉS GUILLERMO ABAD ROBALINO

RESUMEN

El presente trabajo pretende aportar a la estadística, realizando una comparación entre los métodos convencionales de predicción de los valores de una serie de tiempo, y el método de la aplicación de las redes neuronales. La importancia de esta comparación radica en la obtención de criterios para la decisión entre uno u otro método dada una serie de tiempo específica.

Capítulo 1, en donde se presentan antecedentes históricos del desarrollo de las redes neuronales, así también como sus primeras aplicaciones.

Capítulo 2, en donde se presenta la teoría matemática que da forma a las redes neuronales, entre otras cosas se presentan clasificaciones de las redes neuronales y derivamos las fórmulas matemáticas del método de aprendizaje llamado Back-Propagation, que más adelante será utilizado.

Capítulo 3, en donde presentamos la teoría convencional utilizada para la predicción de los valores de una serie de tiempo.

Capítulo 4, en donde le damos una forma o arquitectura específica a la red neuronal para que cumpla la función de predicción de una serie de tiempo.

Capítulo 5, en donde se utilizan tres series de tiempo distintas de ventas para realizar la comparación práctica, de los métodos convencionales y las redes neuronales.

Finalmente se exponen las conclusiones y recomendaciones basadas en los resultados que obtuvimos en nuestra investigación.

INDICE GENERAL

Pág.

II. PRESENTACIÓN DE LA TEORÍA PARA LAS REDES NEURONALES

III. SERIES DE TIEMPO

IV. APLICACIÓN DE REDES NEURONALES AL PRONÓSTICO DE UNA SERIE DE TIEMPO

INTRODUCCION

Una de las principales aplicaciones que tiene la estadística es la de la predicción de los valores de una serie de tiempo. Para esto se ha desarrollado todo una rama de la estadística aplicada, con el consecuente desarrollo de un conjunto muy variado de herramientas.

El problema particular de la predicción de los valores de una serie de tiempo es de especial interés dentro de la estadística. Es muy común que cualquier investigador tenga los datos de cierto experimento a pronosticar, tomados a intervalos de tiempos iguales, es decir, si la segunda observación es tomada una hora después de la primera, la tercera será tomada también una hora después de la segunda y así sucesivamente. Con este marco teórico ya podemos aplicar la vasta teoría existente para la predicción de los valores de una serie de tiempo.

Los principales métodos que se utilizan son los modelos autoregresivos o AR (p) y los de medias móviles o MA (q), (o modelos compuestos por estos dos modelos básicos denominados ARIMA (p,q)). Conforme estos modelos eran más estudiados, fueron poco a poco refinándose cada vez más, y aparecieron modelos muy específicos para series de tiempo que presentaban ciertas características particulares. Estos modelos han llegado a ser

bastantes precisos en sus pronósticos. Sin embargo existe la desventaja de que estos modelos requerían que el investigador sea un experto en la teoría estadística a aplicar.

Mientras se desarrollaban estos métodos para la predicción de los datos de una serie de tiempo, también se hacían avances significativos en otra rama de las matemáticas, nos referimos a las redes neuronales; y así, para mediados de los años 1970, la teoría base de las redes neuronales ya estaba desarrollada. Aunque con este despegue de esta nueva teoría, surgieron muchas aplicaciones, no fue sino hasta los años 1980, que se realizaron los primeros intentos de aplicar las redes neuronales a las predicciones estadísticas.

La presente tesis de grado, realiza una comparación de la precisión de la predicción de los valores de una serie de tiempo utilizando los métodos convencionales y el método de las redes neuronales, encontrando ventajas de un método sobre el otro o encontrando patrones bajo los cuales resulta mejor utilizar uno u otro método. Aportando así con el desarrollo de la estadística.

CAPÍTULO 1

1. LAS REDES NEURONALES

1.1. La neurona

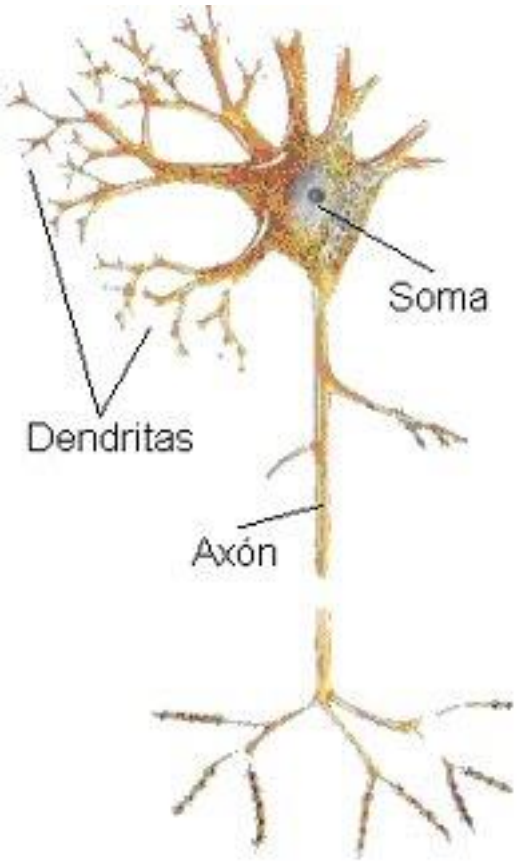
A finales del siglo XIX se logró una mayor claridad sobre el conocimiento del funcionamiento del cerebro debido a los trabajos de Ramón y Cajal en España y Sherrington en Inglaterra. El primero trabajó en la anatomía de las neuronas y el segundo en los puntos de conexión de las mismas o sinápsis.

El tejido nervioso es el más diferenciado del organismo y está constituido por células nerviosas, fibras nerviosas y la neuroglia, que está formada por varias clases de células. La célula nerviosa se denomina neurona, que es la unidad funcional del sistema nervioso. Hay neuronas bipolares, con dos prolongaciones de fibras y multipolares, con numerosas prolongaciones. Pueden ser neuronas

sensoriales, motoras y de asociación. Se estima que en cada milímetro del cerebro hay cerca de 50.000 neuronas. La estructura general de una neurona se muestra en la figura 1.1

Figura 1.1

Estructura general de una neurona



Fuente: www.cienciateca.com

Elaboración: Ciencia Teca

El tamaño y la forma de las neuronas es variable, pero con las mismas subdivisiones que muestra la figura. El cuerpo de la neurona o Soma contiene el núcleo, el cual se encarga de todas las actividades metabólicas de la neurona y recibe la información de otras neuronas vecinas a través de las conexiones sinápticas.

Las dendritas son las conexiones de entrada de la neurona. Por su parte el axón es la "salida" de la neurona y se utiliza para enviar impulsos o señales a otras células nerviosas. Cuando el axón esta cerca de sus células destino se divide en muchas ramificaciones que forman sinápsis con el soma o axones de otras células. Esta unión puede ser "inhibidora" o "excitadora" según el transmisor que las libere. Cada neurona recibe de 10.000 a 100.000 sinápsis y el axón realiza una cantidad de conexiones similar.

La transmisión de una señal de una célula a otra por medio de la sinápsis es un proceso químico. En él se liberan substancias transmisoras en el lado del emisor de la unión. El efecto es elevar o disminuir el potencial eléctrico dentro del cuerpo de la célula receptora. Si su potencial alcanza el umbral se envía un pulso o potencial de acción por el axón. Se dice, entonces, que la célula se disparó. Este pulso alcanza otras neuronas a través de la distribución de los axones.

1.2. La red neuronal

El sistema de neuronas biológico está compuesto por neuronas de entrada (censores) conectados a una compleja red de neuronas "calculadoras" (neuronas ocultas), las cuales, a su vez, están conectadas a las neuronas de salida que controlan, por ejemplo, los músculos. Los censores pueden ser señales de los oídos, ojos, etc. las respuestas de las neuronas de salida activan los músculos correspondientes. En el cerebro hay una gigantesca red de neuronas "calculadoras" u ocultas que realizan los cálculos necesarios. De manera similar, una red neuronal artificial debe ser compuesta por censores del tipo mecánico o eléctrico.

1.3. Breve historia de las redes neuronales

Los intentos por imitar el funcionamiento del cerebro han seguido a la evolución del estado de la tecnología. Por ejemplo, al finalizar el siglo 19 se le comparó con la operación de la bomba hidráulica; durante la década de 1920 a 1930 se intento utilizar la teoría de la conmutación telefónica como punto de partida de un sistema de conocimiento similar al del cerebro. Entre 1940 y 1950 los científicos comenzaron a pensar seriamente en las redes neuronales utilizando como concepto la noción de que las neuronas del cerebro funcionan como interruptores digitales (*on - off)* de manera también similar al recién desarrollado computador

digital. Así nace la idea de "revolución cibernética" que maneja la analogía entre el cerebro y el computador digital.

1943 Teoría de las Redes Neuronales Artificiales

En este año, Walter Pitts junto a Bertran Russell y Warren McCulloch intentaron explicar el funcionamiento del cerebro humano, por medio de una red de células conectadas entre sí, para experimentar ejecutando operaciones lógicas. Partiendo del menor suceso psíquico (estimado por ellos): el impulso todo/nada, generado por una célula nerviosa.

Así, el bucle "sentidos - cerebro - músculos", mediante la retroalimentación producirían una reacción positiva si los músculos reducen la diferencia entre una condición percibida por los sentidos y un estado físico impuesto por el cerebro.

Estos científicos también definieron la memoria como un conjunto de ondas que reverberan en un circuito cerrado de neuronas.

1949 Conductividad de la sinápsis en las Redes Neuronales.

Seis años después de que McCulloch y Pitts mostraran sus Redes Neuronales, el fisiólogo Donald O. Hebb (de la McGill University) expuso que estas (las redes neuronales) podían aprender. Su propuesta tenía

que ver con la conductividad de la sinápsis, es decir, con las conexiones entre neuronas. Hebb expuso que la repetida activación de una neurona por otra a través de una sinápsis determinada, aumenta su conductividad, y la hacía más propensa a ser activada sucesivamente, induciendo a la formación de un circuito de neuronas estrechamente conectadas entre sí.

1951 Primera Red Neuronal

A principio de los años 1950, el estudiante de Harvard, Marvin Minsky conoció al científico Burrhus Frederic Skinner, con el que trabajó algún tiempo ayudándole en el diseño y creación de máquinas para sus experimentos. Minsky se inspiró en Skinner para gestar su primera idea "oficial" sobre inteligencia artificial, su Red Neuronal. Por aquel entonces entabló amistad con otro brillante estudiante, Dean Edmonds, el cual estaba interesado en el estudio de una nueva ciencia llamada Electrónica.

Durante el verano de 1951, Minsky y Edmonds montaron la primera máquina de redes neuronales, compuesta básicamente de 300 tubos de vacío y un piloto automático de un bombardero B-24. Llamaron a su creación "Sharc", se trataba nada menos que de una red de 40 neuronas artificiales que imitaban el cerebro de una rata. Cada neurona hacia el papel de una posición del laberinto y cuando se activaba daba a

entender que la "rata" sabía en que punto del laberinto estaba. Las neuronas que estaban conectadas alrededor de la activada, hacían la función de alternativas que seguir por el cerebro, la activación de la siguiente neurona, es decir, la elección entre "derecha" o "izquierda" en este caso estaría dada por la fuerza de sus conexiones con la neurona activada. Por ejemplo, la "rata" completaba bien el recorrido eligiendo a partir de la quinta neurona la opción "izquierda" (que correspondería a la sexta), es entonces cuando las conexiones entre la quinta y sexta se hacen más fuertes (dicha conexión era realizada por el piloto automático), haciendo desde este momento más propensa esta decisión en un futuro. Pero las técnicas Skinnerianas (que eran las que se habían puesto en funcionamiento en esta red neuronal) no podrían llevar muy lejos a este nuevo engendro, la razón es que esto, en sí, no es inteligencia, pues la red neuronal así creada nunca llegaría a trazar un plan.

Después de su Red Neuronal, Minsky escribió su tesis doctoral acerca de esta, en ella describía "cerebros mucho mayores", exponiendo que si se realizaba este proyecto a gran escala, con miles o millones de neuronas más y con diferentes censores y tipos de retroalimentación, la máquina podría ser capaz de razonar, pero el sabía que la realización de esta Red Neuronal era imposible y así, decidió buscar otra forma de crear inteligencia.

1.4. Las Redes Neuronales Artificiales

1.4.1. Sistema Experto

Un método más avanzado para representar el conocimiento, es el sistema experto. Típicamente está compuesto por varias clases de información almacenada: Las reglas *If - Then* le dicen al sistema como se debe reaccionar ante los estados del "mundo". Una regla del sistema experto puede ser *if Y* es un hombre, *Then Y* es mortal. Los hechos describen el estado del "mundo". Por ejemplo: Juan es mortal. Por último, una máquina de inferencia relaciona los hechos conocidos con las reglas *If - Then* y genera una conclusión. En el ejemplo: Juan es mortal. Esta nueva conclusión se añade a la colección de hechos que se almacena en los medios ópticos o magnéticos del computador digital. De esta forma, un sistema experto sintetiza nuevo conocimiento a partir de su "entendimiento" del mundo que le rodea. Es decir, un sistema experto es un método de representación y procesamiento del conocimiento, mucho más rico y poderoso que un simple programa de computador. Sin

embargo, con respecto a la manera en que opera el cerebro humano, sus limitaciones son múltiples. Los problemas planteados en términos difusos o ambiguos , por ejemplo, son muy complejos de analizar o "conocer" con sistemas de procesamiento simbólico, como los sistemas expertos o programas de computador.

1.4.2. Método de transmisión de la información en el cerebro

Primero conviene saber que en los primeros tiempos de la informática a las computadoras se los llamaba calculadoras de cifras electrónicas o simplemente calculadoras digitales. Los sistemas digitales trabajan con cifras en código binario que se transmiten en formas de impulsos (bits). Los sistemas analógicos procesan señales continuamente cambiantes, como la música o la palabra hablada. Por suerte para nuestro propósito de imitar con una computadora el cerebro, este también codifica la información en impulsos digitales. En los humanos las sensaciones se generan digitalmente y se transmiten así a través del sistema nervioso. En otras palabras cuando la luz se hace más intensa, el sonido más alto o la presión más fuerte, entonces no es que fluya mas corriente a través de los nervios, sino que la frecuencia de los impulsos digitales aumenta.

En principio las computadoras trabajan de manera semejante. Así una sensación mas fuerte corresponde en un equipo informático a una cifra más alta (o en una palabra mas larga). Sin embargo en una computadora los datos se transmiten siempre a un mismo ritmo; la frecuencia base es inalterable. Por eso las cifras mas altas tardan mas tiempo en ser transmitidas. Como por lo general las computadoras no trabajan en tiempo real, esto no tiene mayor importancia, pero cuando se trata de un computador en tiempo real, como son los empleados en proceso industrial, hace falta de ampliar él numero de canales de transmisión para que en el mismo espacio de tiempo pueda fluir mayor cantidad de datos.

1.4.3. Compuertas lógicas

Sabemos que los elementos básicos de una computadora son las compuertas lógicas, en el cerebro también existen aunque no son idénticas a las de una computadora.

En una computadora las compuertas And, Or etc. tienen una función perfectamente determinada e inalterable. En el cerebro también hay elementos de conexión parecidos, las llamadas sinapsis, donde confluyen en gran número las fibras nerviosas.

1.4.4. Funcionamiento de las sinapsis

Cientos de datos fluyen por los nervios hasta cada sinapsis, donde son procesados. Una vez analizada y tratada la información esta sale ya transformada por los canales nerviosos.

En los seres vivos no pueden permitirse el lujo de la especialización ya que si algo se rompe otro elemento debe hacerse cargo de la función. Por eso cada sinapsis es simultáneamente una compuerta Ad, Or, Not etc.

Una sinapsis suma las tensiones de los impulsos entrantes. Cuando se sobrepasa un determinado nivel de tensión; el llamado umbral de indicación; esta se enciende, esto es deja libre el camino para que pasen los impulsos. Si el umbral de indicación de tensión es demasiado bajo, la sinapsis actúa como una puerta lógica del tipo Or, pues en tal caso pocos impulsos bastan para que tenga lugar la conexión. En cambio cuando el umbral de indicación es alto, la sinapsis actúa como una puerta And, ya que en ese caso hace falta que lleguen la totalidad de los impulsos para que el camino quede libre. También existen

conducciones nerviosas que tienen la particularidad de bloquear el paso apenas llegan los impulsos. Entonces la sinapsis hace la función de una compuerta inversora. Esto demuestra la flexibilidad del sistema nervioso.

1.4.5. Diferencias entre el cerebro y una computadora

La diferencia más importante y decisiva es cómo se produce el almacenamiento de información en el cerebro y en la computadora.

Computadora: Los datos se guardan en posiciones de memoria que son celdillas aisladas entre sí. Así cuando se quiere acceder a una posición de memoria se obtiene el dato de esta celdilla. Sin que las posiciones de memoria aldeanas sé den por aludidas.

Cerebro: La gestión es totalmente diferente. Cuando buscamos una información no hace falta que sepamos donde se encuentra almacenada y en realidad no lo podemos saber ya que nadie sabe, hasta hoy en día, donde guarda el cerebro los datos.

Pero tampoco es necesario ya que basta con que pensemos en el contenido o significado de la información para que un mecanismo, cuyo funcionamiento nadie conoce, nos proporcione automáticamente no solo la información deseada sino que

también las informaciones vecinas, es decir, datos que de una u otra manera hacen referencia a lo buscado.

Los expertos han concebido una serie de tecnicismos para que lo incomprensible resulte algo más comprensible. Así a nuestro sistema para almacenar información se lo llama memoria asociativa. Esta expresión quiere dar a entender que los humanos no memorizan los datos direccionándolos en celdillas, sino por asociación de ideas; esto es, interrelacionando contenidos, significados, modelos.

En todo el mundo, pero sobre todo en Estados Unidos y Japón, científicos expertos tratan de dar con la clave de la memoria asociativa. Si se consiguiera construir un chip de memoria según el modelo humano, la ciencia daría un paso gigante en la fascinante carrera hacia la inteligencia artificial. Y además el bagaje del saber humano quedaría automáticamente enriquecido.

1.4.6. Similitudes entre el cerebro y una computadora.

- Ambos codifican la información en impulsos digitales.
- Tanto el cerebro como la computadora tienen compuertas lógicas.
- Existen distintos tipos de memoria.

- Los dos tienen aproximadamente el mismo consumo de energía.

1.4.7. Una super-computadora llamado cerebro

El hombre necesita un sistema de proceso de datos de múltiple propósito capaz de tratar gran cantidad de información muy distinta y en muy poco tiempo y con el mayor sentido práctico(pero no necesariamente con exactitud), para inmediatamente poder actuar en concecuencia. Las computadoras, en cambio, son altamente especializados con capacidad para procesar con exactitud información muy concreta (en principio solo números) siguiendo unas instrucciones dadas.

El cerebro humano posee más de diez millones de neuronas las cuales ya están presentes en el momento del nacimiento, y conforme pasa el tiempo se vuelven inactivas, aunque pueden morir masivamente.

Nuestro órgano de pensamiento consume 20 Watios/hora de energía bioquímica, lo que corresponde a una cucharada de azúcar por hora. Las computadoras domésticos consumen una cantidad semejante. Las necesidades de oxígeno y alimento son

enormes en comparación con las necesidades del resto del cuerpo humano: casi una quinta parte de toda la sangre fluye por el cerebro para aprovisionar de oxígeno y nutrientes. La capacidad total de memoria es difícil de cuantificar, pero se calcula que ronda entre 10^{12} y 10^{14} bits.

La densidad de información de datos de un cerebro todavía no se ha podido superar artificialmente y en lo que se refiere a velocidad de transmisión de datos, a pesar de la lentitud con que transmite cada impulso aislado, tampoco está en desventaja, gracias a su sistema de proceso en paralelo: la información recogida por un ojo representa 10^6 bits por segundo.

Según todos los indicios el cerebro dispone de dos mecanismos de almacenamiento de datos: la memoria intermedia; que acepta de cinco a diez unidades de información, aunque solo las mantiene durante algunos minutos, y la memoria definitiva, que guarda las informaciones para toda la vida, lo que no significa que nos podamos acordar siempre de todo. La memoria inmediata trabaja como una especie de cinta continua: la información circula rotativamente en forma de impulsos eléctricos por los registros. El sistema es comparable a la memoria dinámica de una computadora, en la que la información tiene que

ser refrescada continuamente para que no se pierda. En cambio, la memoria definitiva parece asemejarse más bien a las conocidas memorias de celdillas de las computadoras. Se cree que esta memoria funciona gracias a formaciones químicas de las proteínas presentes en el cerebro humano.

1.4.8. Aplicaciones Estadísticas de las redes neuronales.-

Uno de los principales objetivos de la estadística inferencial es el de la predicción de los valores de una serie de tiempo. Para esto se han desarrollado numerosos métodos, entre los que tenemos la regresión lineal, los modelos de medias móviles, autoregesivos e integrados, los de alisamiento exponencial, entre otros. También existen métodos no paramétricos, heurísticos; entre los que podemos citar el de redes neuronales. Es de particular importancia la comparación del poder de predicción de los métodos convencionales frente a los métodos alternativos.

Uno de los métodos convencionales más utilizados es el de los modelos ARIMA, mientras los recientes avances en la teoría de las Redes Neuronales han convertido a este método en el método de predicción no convencional más popular. Por consiguiente un estudio comparativo entre ambos métodos es de mucho interés.

Una de las preguntas abiertas en la teoría de las Redes Neuronales es precisamente en que situaciones se las prefiere frente a otras técnicas. En esta tesis se busca obtener conclusiones sobre la preferencia de las redes neuronales frente a los modelos convencionales en el problema de la predicción de series de tiempo.

Así el tema central de estudio será la determinación de si en la predicción de series temporales la utilización de las Redes Neuronales resuelve mejor el problema que los métodos convencionales de predicción. Para lo cual se usarán series de datos reales con un horizonte suficientemente grande como para realizar las predicciones. El modelo ARIMA se implementará en el software matemático MATLAB, y será de interés el análisis de algunos índices de rendimiento de los modelos, entre otros: la suma cuadrática de los errores, el estadístico F, un ploteo de los errores, etc.

Para la predicción utilizando las Redes Neuronales solamente necesitaremos diseñar una red de dos capas, ya que como ha sido demostrado por Funashashi, en 1989; dado que la función de

activación no es lineal, una red de dos capas (es decir, de una sola capa oculta) es suficiente para aproximar cualquier función con un número finito de discontinuidades a un error de precisión arbitrario. No podemos utilizar una red neuronal de una sola capa, ya que aún cuando el teorema de convergencia universal demuestra que de existir una solución, la regla de aprendizaje del perceptrón encontrará una solución en un número finito de pasos, este tipo de redes de capa única se ajustan a funciones lineales, es decir, obtendríamos resultados del mismo poder de predicción que los que obtenemos al realizar una autoregresión en la serie.

La estructura o topología que utilizaremos en nuestro modelo de Redes Neuronales consiste en 12 nodos de entrada, 6 nodos escondidos o internos y un nodo de salida. Los 12 nodos de entrada son los 12 datos históricos inmediatamente anteriores a la fecha que deseamos predecir; es decir, si deseamos predecir las ventas en un periodo t, entonces como entrada recibiremos los datos del periodo t-1, t-2,…,t-12 (en nuestro estudio llegaremos hasta t-12, por la característica cíclica anual que se esperaría), así podremos variar estos nodos dependiendo de la cantidad de datos anteriores que consideremos influirán en la predicción actual. Los nodos intermedios generalmente son difíciles o hasta imposibles

de explicar, por lo que se suele decir que las redes neuronales son una caja negra, indescifrable. Debemos tener presente que este método es un método heurístico. El nodo de salida es la predicción de ventas del periodo que deseamos.

Una de las características que diferencia a las redes neuronales de otros métodos de predicción es la capacidad de aprendizaje que éstas poseen.

Para los modelos de dos o más capas, la regla de aprendizaje más utilizada es la de Backpropagation, (propagación hacia atrás), que tal como su nombre lo indica es un método que una vez realizado un ensayo, compara el resultado con el deseado, y ajusta la red, de capa en capa y de atrás hacia delante. El ajuste comienza en la capa de salida y una vez que todas las unidades han sido ajustadas, continúa con la capa inmediata anterior, para luego hacerlo con la siguiente y así sucesivamente. Esta técnica, presentada por Rumelhart en el año de 1986, significó un gigantesco avance en la teoría de las redes neuronales, de hecho, fue gracias a este trabajo que las redes neuronales tomaron otra vez fuerza. En 1969, Minsky & Papert plantearon la forma en que una red de multicapas puede superar las limitaciones que presentaban las redes de una sola capa y aproximar una función

no lineal con una precisión deseada, aunque no incluyeron una regla de aprendizaje, lo que hizo que no se encontraran mayores aplicaciones a sus resultados.

En este trabajo, la implementación de la red neuronal en la computadora se hará a través de Neural Network, un toolbox de MATLAB, el cual constituye un conjunto de herramientas que permiten la implementación rápida de las redes neuronales. Aquí especificaremos la topología de la red, el número de unidades de entrada, de unidades intermedias o escondidas y de unidades de salida, luego realizaremos el entrenamiento y finalmente la predicción.

CAPÍTULO 2

2. PRESENTACIÓN DE LA TEORÍA PARA LAS REDES NEURONALES

2.1. Las Redes Neuronales y la Modelación

El estudio de las Redes Neuronales comenzó en la década de 1960, generando primero un gran interés que sin embargo decayó al poco tiempo y que dio lugar luego al escepticismo en la comunidad científica. Su lento despegue se debió principalmente al incipiente desarrollo de la Computación en esa época, rama fuertemente ligada a las Redes Neuronales; y a la lenta aparición de modelos matemáticos precisos.

En los tiempos actuales, con el creciente desarrollo de las computadoras, las Redes Neuronales han tenido un desarrollo muy importante, así como también han encontrado aplicaciones en

diferentes campos del conocimiento humano, entre los que podemos citar:

El Aprendizaje artificial.-

- Auto programación de una computadora para realizar tareas.- se tiene un conjunto de datos experimentales para la red, y mediante la retroalimentación, la red se ajusta a sí misma para la realización de determinada tarea.
- Optimización.- en el área de la investigación de operaciones, dado un conjunto de restricciones y una función objetivo, encontrar la solución óptima. Como ejemplo tenemos el problema del agente viajero.
- El problema de clasificación.- asignar a los elementos de un conjunto un grupo. Como ejemplo podemos citar los análisis de conglomerados y el reconocimiento de letras escritas a mano.

En las ciencias cognoscitivas.-

- Modelamiento de niveles altos de razonamiento
- Modelamiento de niveles bajos de razonamiento

Neurobiología.-

- En la modelación del cerebro humano para entender su funcionamiento.

Matemáticas.-

- En la estadística no paramétrica
- En los modelos de regresión y de predicción.

Filosofía./Psicología.-

- Resolviendo problemas filosóficos profundos como por ejemplo: ¿Puede el comportamiento humano ser explicado únicamente a través de símbolos o se necesita de un modelo de bajo nivel como el de las Redes Neuronales?

2.1.1. Áreas específicas de aplicación

Entre las áreas específicas en donde se están utilizando con éxito las redes neuronales tenemos:

- El Procesamiento de señales.- reducción de ruidos, eliminación de resonancias y otras aplicaciones
- La Robótica.- para el reconocimiento visual. De hecho esta fue una de las principales aplicaciones prácticas que se le dio a las redes neuronales. El experimento consistía en un automóvil con una cámara incorporada, y con foquitos a los lados de una carretera. El automóvil debía ser capaz de auto-conducirse sin salirse de la vía.
- La Reproducción del Habla
- El Reconocimiento del Habla

- La Visión 3D, en el reconocimiento de caras y fronteras.
- Los negocios.- en la concesión de créditos, y en tasas de seguros.

En la construcción de programas de decisión que tengan un alto grado de concordancia con las decisiones de los profesionales (sistemas expertos).

- Las finanzas.- en la predicción del mercado bursátil.
- Para la compresión de datos.

2.1.2. Modelamiento del cerebro humano en una red neuronal.-

El cerebro humano esta formado por alrededor de un billón de células nerviosas que se conocen como neuronas, éstas a su vez se conectan con otras neuronas a través de lo que conocemos como sinápsis. Así los elementos principales de toda neurona son: el cuerpo de la célula, las dendritas (o entradas), y los axones (o salidas). Partiendo de esta forma física y funcional de las células que componen al cerebro humano creamos un modelo matemático que las imite.

El marco que define a esta red está conformado por los siguientes elementos.

- Un conjunto de unidades de proceso (los nodos propiamente dichos).
- Un conjunto de "estados" y_k de activación para cada unidad k.

- Una conexión entre las unidades, que denominaremos w_{jk}, que representa al peso o conexión entre la unidad j y la unidad k. Esta determina la magnitud del efecto entre unidades, y es primordial en el proceso de ajuste.

- Una regla de propagación, la cual determina la entrada efectiva s_k de una unidad de entre sus entradas externas.

- Una función de activación F_k, que depende de la entrada efectiva s_k y del estado de activación actual y_k. A esta función la llamaremos la renovación.

- Una entrada externa θ_k para cada unidad.

- Un método para reunir información (una regla de aprendizaje).

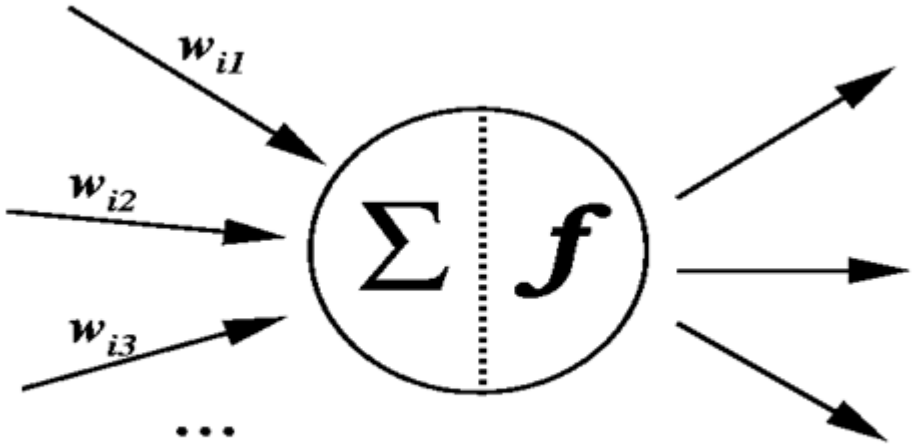
- Un ambiente o entorno dentro del cual el sistema debe operar. El cual le proporcionará las señales de entrada y si es posible las señales de error.

Este es el marco general de las redes neuronales, ahora veamos como ajustamos esto a nuestro interés, es decir a pronosticar los valores en una serie temporal.

2.2. Redes de una sola capa

Las redes neuronales pueden ser caracterizadas de acuerdo al número de capas que tienen. Las unidades de una red se dividen en tres tipos, las unidades de entrada, las unidades de salida y las unidades escondidas o internas. Todas las unidades, a excepción de las de entrada, reciben ingresos de una o más unidades de un nivel anterior; así mismo todas las unidades, a excepción de las de salida, son ingresos de una o más unidades de un nivel posterior. Así, según el número de niveles de una red, se la denomina como de una, dos, o más capas. Por convención no se cuenta al nivel conformado por unidades de entrada, es decir, a una red conformada por dos o más unidades de entradas y una o más unidades de salida la clasificaremos como de una sola capa. En la figura 2.1 podemos observar un esquema sencillo de una neurona.

Figura 2.1
Esquema sencillo de la modelación de una neurona



Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos.
Elaboración: Andrés Abad

Veremos primero el caso más sencillo de este tipo de redes, llamado el perceptrón. El perceptrón está conformado por 2 unidades de entrada y una de salida.

La información que ingresa a la unidad de salida será una suma ponderada de las unidades de ingresos x_j, con los pesos w_j para $j = 1,2$. Además incluimos un término constante w_0, al que por simplicidad lo consideraremos como el peso de una variable ficticia $x_0 \equiv 1$. Con la notación utilizada, el ingreso s a la unidad de salida j

será igual a $s = \sum_{i=0}^{2} x_i w_i$. Por el momento no especificaremos una función de activación F de la unidad de salida.

Una vez definida la estructura de esta red nuestro interés se centrará en la determinación de los pesos w_i. Esto es lo que se llama el aprendizaje de la red, y se logra utilizando una regla de aprendizaje, y utilizando los datos de entrenamiento (es decir los $x \in D$, donde D es el conjunto de datos de aprendizaje).

2.3. Entrenamiento de redes neuronales artificiales

Una red neuronal debe ser configurada de tal manera que con la aplicación de un conjunto de datos de entrada se produzca un deseado conjunto de salidas. Existen varios métodos para definir los valores iniciales del conjunto de pesos de la red. Una forma de definir estos valores es obtener un conocimiento *a priori*. Otra forma es "entrenar" a la red alimentándola con datos de aprendizaje y permitiéndole modificar sus pesos de acuerdo a lo que llamamos una regla de aprendizaje.

2.3.1. Tipos de aprendizaje

Podemos categorizar a los tipos de aprendizaje en dos grupos. Estos son:

2.3.1.1. Aprendizaje Supervisado o asociativo.- es cuando la red es entrenada a través del empleo de entradas junto con sus respectivas salidas. Estos pares pueden ser provistos por un "juez externo", o por algún sistema contenido en la propia red, caso en el que se la denomina auto-supervisada.

2.3.1.2. Aprendizaje no Supervisado o auto-organizado.- este se tiene cuando una unidad de salida es entrenada para responder a conglomerados o clusters, dentro del conjunto de datos de entrada. En este tipo de aprendizaje el sistema está listo a descubrir estadísticamente algunas características inherentes en la población de entrada. A diferencia del método supervisado de aprendizaje, aquí no existe un conocimiento *a priori*, sino que el sistema debe desarrollar sus propios criterios para la clasificación de sus entradas.

En la actualidad existen algunas propuestas para esta regla de aprendizaje, algunas muy exóticas. Como introducción veremos dos, la regla de aprendizaje simple, o del perceptrón, y la regla de los mínimos errores cuadráticos, MEC; también llamada la regla delta.

Ambas reglas parten de una misma idea, la de ajustes iterativos de los pesos, es decir $w' = w + \Delta w$. Lo que se hace es incrementar (o disminuir), en cada entrenamiento una cantidad Δw al valor del peso anterior. La diferencia entre los dos métodos que estudiaremos radica en la cantidad Δw.

2.3.2. Regla de aprendizaje del perceptrón

Esta regla utiliza como función de activación F a la función sgn, es decir a la función signo, es decir $\text{sgn}(s) = 1$ si $s > 0$ y $\text{sgn}(s) = -1$ si $s \leq 0$. Por lo tanto el estado y de la unidad de salida después de recibir los ingresos de las unidades de entrada será 1 ó -1, y estará dado por $y = \text{sgn}(\sum x_i w_i)$. El Δw que propone esta regla de aprendizaje es sencillamente $\Delta w_i = d(x) \cdot x_i$, donde $d(x)$ es el valor deseado cuando se ingresan los valores de entrenamiento x. Dado que $d(x)$ toma únicamente los valores de 1 o -1, lo que estamos definiendo es simplemente un incremento o decrecimiento directamente proporcional al valor de la entrada x_i. Si el signo es positivo estaremos hablando de un “estímulo”, y si es negativo diremos que se trata de una “inhibición”. El algoritmo para esta regla de aprendizaje es:

1.- Asignar valores aleatorios a los pesos de la red. Generalmente valores pequeños.

2.- Mientras no se alcance un nivel deseado de predicción repetir:

- Tomar un dato de entrenamiento x y obtener la salida y.

- Si $d(x) \neq y$ entonces $w'_i = w_i + d(x) \cdot x_i$.

3.- FIN

Debe notarse que este tipo de algoritmos son muy buenos en funciones de clasificación, en los cuales si la salida es 1 pertenece a cierta clase y si es -1 a otra clase, como en las funciones booleanas. Veremos ahora la regla delta, cuya aplicación se halla un poco menos restringida, ya que no es necesario utilizar la función signo como la función de activación.

2.3.3. Regla de aprendizaje MEC o delta.-

Como ya dijimos esta segunda regla de aprendizaje utiliza el mismo principio que la anterior, la diferencia se halla en el término Δw, y en que no se necesita que la salida sea booleana. Utilizaremos una medida de error, la que definiremos como $E^p = \frac{1}{2}(d^p - y^p)^2$, donde E^p es el error cuando se utiliza el dato de entrenamiento p, d^p es el resultado deseado, y y^p es el resultado obtenido. La medida que

buscamos será $E = \sum_{p \in D} E^p = \frac{1}{2} \sum (d^p - y^p)^2$, y como es de esperarse lo que busca la regla delta es encontrar los w_i que minimicen este error cuadrático.

El Δw para esta regla de aprendizaje es $\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$, donde η representa una tasa de aprendizaje (usualmente pequeña, por ejemplo $\eta = 0.05$). La idea es definir un vector gradiente $\nabla \cdot E = \left\langle \frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, ..., \frac{\partial E}{\partial w_m} \right\rangle$, y encontrar la dirección en la que este vector hace menor al error E. Para calcular estas derivadas parciales hacemos lo siguiente: $\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial w_i}$, por la regla de la cadena. Dada la linealidad de y tenemos que $\frac{\partial y}{\partial w_i} = x_i$, ahora tenemos que $\frac{\partial E}{\partial y} = -(d^p - y^p)$, finalmente tenemos que $\Delta w_i = \eta \delta^p x_i$, donde $\delta^p = (d^p - y^p)$. El algoritmo es exactamente el mismo, la única diferencia es el cálculo de los Δw.

Existe una generalización del método de aprendizaje delta, para redes neuronales de dos o más capas. Esta técnica es conocida como Backpropagation.

2.4. Redes de dos Capas

2.4.1. Método de Backpropagation.-

Una red de una sola capa tiene severas restricciones en cuanto al conjunto de funciones que puede ajustar, como por ejemplo el problema del O excluyente (XOR). En general, todos aquellos problemas en donde, graficados en el plano, los datos que pertenecen a dos grupos distintos no puedan ser separados por un línea recta.

En 1969 Minsky y Papert, mostraron que una red de dos capas con alimentación hacia delante puede sobreponerse a muchas restricciones, pero no presentaron una solución al problema de cómo ajustar los pesos de entrada de las unidades escondidas. Una solución a esta dificultad fue presentada por Rumelhart, Hinton y Williams en 1986. La idea central detrás de esta solución es que los errores de las unidades escondidas (1era capa), son determinados por propagación hacia atrás de los errores de las unidades de salida. Por esta razón el método es generalmente llamado regla de aprendizaje Back-propagation. Esta regla también puede ser considerada como una generalización de la regla delta para funciones de activación no lineales, y redes de multicapas.

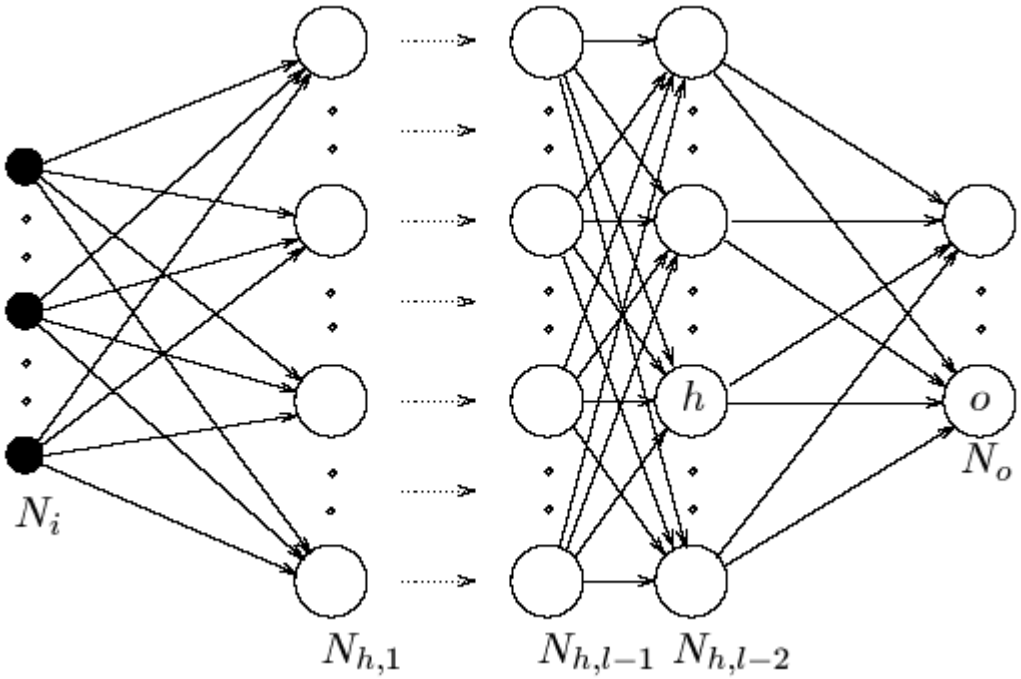
2.4.2. Redes de Multicapas con alimentación hacia delante

Una red de alimentación hacia delante tiene una estructura de capas. Cada capa consiste en unidades que reciben sus entradas de unidades directamente detrás de éstas y envían sus salidas a unidas en capas directamente adelante. No hay conexiones dentro de una capa. Las unidades N_i de entrada alimentan a la primera capa de $N_{h,1}$ unidades escondidas. Las unidades de entrada son simplemente unidades de “ventilación”; ningún proceso se lleva a cabo en estas unidades. La activación de las unidades escondidas es una función F_i de las entradas con sus respectivos pesos y una constante, según la ecuación:

$$y_k(t+1) = F_k(s_k(t)) = F_k\left(\sum_j w_{jk}(t) y_j(t) + \theta_k(t)\right)$$

Las salidas de las unidades escondidas son distribuidas hacia la siguiente capa de $N_{h,2}$ unidades escondidas, y así sucesivamente hasta la última capa de N_o unidades de salidas. En la figura 2.2, a continuación podemos ver un esquema de una red de multicapas.

Figura 2.2
Diseño de la red multicapa



Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005
Elaborado: Andrés Abad Robalino

Aunque el método back-propagation puede ser aplicado a redes con cualquier número de capas, se ha demostrado que una sola capa de unidades escondidas es suficiente para aproximar con una precisión arbitraria, cualquier función que tenga un número finito de discontinuidades. Siempre que la red tenga funciones de activación no lineales en las capas escondidas. Este resultado es conocido como el teorema de aproximación universal. En la mayoría de las aplicaciones de redes con alimentación hacia delante y con una capa escondida

simple, se utiliza una función de activación sigmoide para las unidades.

2.4.3. La regla delta generalizada

Para utilizar unidades que tienen una función de activación no lineal, debemos generalizar la regla delta, que presentamos en este mismo capítulo. Ya no usaremos funciones de activación lineales sino un conjunto de funciones de activación no lineales. Esta función de activación será una función diferenciable para el total de las entradas, dada por

$$y_k^p = F\left(s_k^p\right), \tag{1}$$

en donde

$$s_k^p = \sum_j w_{jk} y_j^p + \theta_k . \tag{2}$$

Para plantear la correcta generalización de la función delta, como fue presentada en la sección previa, debemos fijar

$$\Delta_p w_{jk} = -\gamma \frac{\partial E^p}{\partial w_{jk}}. \tag{3}$$

La medida de error E^p es definida como el error cuadrático total para el conjunto p de entradas y de sus correspondientes salidas:

$$E^p = \frac{1}{2}\sum_{o=1}^{N_o}\left(d_o^p - y_o^p\right)^2, \tag{4}$$

Donde d_o^p es la salida deseada para la unidad o cuando el conjunto p es ingresado. Luego fijaremos $E = \sum_p E^p$ como la suma de errores cuadráticos. Podemos escribir

$$\frac{\partial E^p}{\partial w_{jk}} = \frac{\partial E^p}{\partial s_k^p} \frac{\partial s_k^p}{\partial w_{jk}}. \tag{5}$$

Por la ecuación 2 vemos que el segundo factor es

$$\frac{\partial s_k^p}{\partial w_{jk}} = y_j^p. \tag{6}$$

Si definimos

$$\delta_k^p = -\frac{\partial E^p}{\partial s_k^p}, \tag{7}$$

Conseguiríamos una regla actualizada que es equivalente a la regla delta descrita previamente, resultando esto en un descenso por el gradiente sobre la superficie de error. Si hacemos que los pesos se reajusten de acuerdo a:

$$\Delta_p w_{jk} = \gamma \delta_k^p y_j^p. \tag{8}$$

Aquí interpretaremos lo que δ_k^p debería ser para cada unidad k en la red. El resultado interesante, que conseguiremos ahora, es que existe un cálculo recursivo simple para estos δ que puede ser interpretado como la propagación de las señales de error hacia atrás a lo largo de la red.

Para calcular δ_h^p aplicaremos la regla de la cadena. Escribiremos esta derivada parcial como el producto de dos factores, un factor que refleje los cambios en el error como una función de las salidas de las unidades y otro que refleje el cambio en las salidas como una función de los cambios de las entradas. Tendremos

$$\delta_k^p = -\frac{\partial E^p}{\partial s_k^p} = -\frac{\partial E^p}{\partial y_k^p}\frac{\partial y_k^p}{\partial s_k^p}. \tag{9}$$

Calculando el segundo factor, por la ecuación 1, vemos que

$$\frac{\partial y_k^p}{\partial s_k^p} = F\left(s_k^p\right), \tag{10}$$

es simplemente la derivada de la función de alisamiento F, para la $k$ésima unidad, evaluada en las entradas s_k^p. Para calcular el primer factor de la ecuación 9, consideraremos dos casos. Primero, asumiremos que la unidad $k = o$ es una unidad de salida de la red. En este caso, se sigue de la definición de E^p que

$$\frac{\partial E^p}{\partial y_o^p} = -(d_o^p - y_o^p), \tag{11}$$

El cuál es el mismo resultado que obtuvimos con la función delta estándar. Substituyendo esto y la ecuación 10 en la ecuación 9, obtenemos

$$\delta_o^p = (d_o^p - y_o^p)F(s_o^p) \tag{12}$$

para cualquier unidad o de salida. Segundo, si es que k no es una unidad de salida, sino una unidad escondida $k = h$, realmente conocemos todavía la contribución de la unidad al error de salida de la red. Sin embargo, la medida del error puede ser escrita como una función de las entradas de las capas escondidas hacia las capas de salida; $E^p = E^p\left(s_1^p, s_2^p, ..., s_j^p, ..\right)$ y utilizamos la regla de la cadena para escribir

$$\frac{\partial E^p}{\partial y_h^p} = \sum_{0=1}^{N_o} \frac{\partial E^p}{\partial s_o^p} \frac{\partial s_o^p}{\partial y_h^p} = \sum_{j=1}^{N_o} \frac{\partial E^p}{\partial s_o^p} \frac{\partial}{\partial y_h^p} \sum_{j=1}^{N_h} w_{ko} y_j^p = \sum_{o=1}^{N_o} \frac{\partial E^p}{\partial s_o^p} w_{ho} = -\sum_{o=1}^{N_o} \delta_o^p w_{ho}. \quad (13)$$

Substituyendo esto en la ecuación 9 obtenemos

$$\delta_h^p = F(s_h^p) \sum_{o=1}^{N_o} \delta_o^p w_{ho}. \quad (14)$$

Las ecuaciones 12 y 14 proporcionan un procedimiento recursivo para calcular el δ para todas las unidades de la red, que son usadas después para calcular los cambios de los pesos de acuerdo a la ecuación 8. Este procedimiento constituye la generalización de la regla delta para una red multicapas con alimentación hacia adelante de unidades no lineales.

2.4.3.1. Explicación de Back-propagation

La ecuación que acabamos de obtener podrá ser matemáticamente correcta, pero, ¿Qué es lo que realmente significa? ¿Hay alguna otra

forma de comprender el método back-propagation que recitando las ecuaciones necesarias?

La respuesta es, por supuesto, sí. De hecho, todo el proceso de back-propagation es intuitivamente muy claro. Cuando un grupo de aprendizaje es utilizado, los valores de activación son propagados a las unidades de salida, y las salidas producidas son comparadas con las deseadas, generalmente terminamos con un error en cada una de las unidades de salida. Llamaremos a este error e_o para una unidad de salida particular o. Debemos llevar e_o hacia cero.

El método más simple para hacer esto es cambiando los pesos. Lo cual se hace de tal manera, que la próxima vez, el error e_o será cero para este grupo en particular. Sabemos de la regla delta, que para poder reducir el error a cero, debemos adaptar los pesos respectivos de acuerdo a

$$\Delta w_{ho} = (d_o - y_o) y_h. \tag{15}$$

Ese es el paso uno, pero esto no es suficiente. Cuando solo aplicamos esta regla, los pesos de las unidades de entrada a las escondidas nunca son cambiados, y no alcanzamos a abarcar en su cabalidad el poder de representación de la red, como es requerido por el Teorema de Aproximación Universal. Con el fin de adaptar los

pesos de unidades de entrada a las unidades escondidas, de nuevo aplicaremos la regla delta. En este caso, no tenemos un valor para δ de las unidades escondidas. Esto es resuelto por la regla de la cadena, la cual distribuye el error de una unidad de salida o a todas las unidades escondidas que se conectan a ella. De manera diferente, una unidad escondida h recibe un δ de cada unidad de salida o igual al delta de esa unidad de salida, multiplicada por el peso de la conexión entre esas dos unidades. En símbolos: $\delta_h = \sum_o \delta_o w_{ho}$. Solo falta la función de activación de las unidades escondidas; F tiene que ser aplicada al δ , antes que el proceso de back-propagation pueda continuar.

2.4.3.2. Trabajando con Back-Propagation

La aplicación de la regla delta generalizada involucra dos fases: durante la primera fase, la entrada $\mathbf{x}$ es presentada y propagada hacia delante, a través de la red para poder calcular los valores de salida y_o^p para cada unidad de salida. Esta salida es comparada con su respectiva salida deseada d_o , resultando en una señal de error δ_o^p para cada unidad de salida. La segunda fase comprende un recorrido por la red hacia atrás durante el cual, la señal de error es

pasada a cada unidad de la red y luego son calculados los cambios apropiados en los pesos.

2.4.3.3. Ajuste de pesos con una función sigmoide de activación.-

Los resultados previos pueden ser resumidos en tres ecuaciones:

- El peso de una conexión es ajustada por una cantidad proporcional al producto de una señal de error δ, en la unidad k que recibe la entrada y la salida de la unidad j, enviando esta señal por la conexión:

$$\Delta_p w_{jk} = \gamma \delta_k^p y_j^p. \tag{16}$$

- Si la unidad es una unidad de salida, la señal de error está dada por:

$$\delta_o^p = (d_o^p - y_o^p) F(s_o^p). \tag{17}$$

Si tomamos como función de activación F, a la función sigmoide, definida así:

$$y^p = F(s^p) = \frac{1}{1+e^{-s^p}}. \tag{18}$$

En este caso, la derivada es igual a

$$F(s^p) = \frac{\partial}{\partial s^p} \frac{1}{1+e^{-s^p}}$$

$$= \frac{1}{\left(1+e^{-s^p}\right)^2} (-e^{-s^p})$$

$$= \frac{1}{\left(1+e^{-s^p}\right)}\left(\frac{e^{-s^p}}{1+e^{-s^p}}\right)$$

$$= y^p(1-y^p). \tag{19}$$

La señal de error para una unidad de salida pueda ser escrita como

$$\delta_o^p = (d_o^p - y_o^p)y_o^p(1-y_o^p). \tag{20}$$

- La señal de error para una unidad escondida es determinada recursivamente en términos de las señales de error de las unidades a las que se conecta directamente, y de los pesos de esas conexiones. Para el caso particular de la función sigmoide de activación, tendremos:

$$\delta_h^p = F(s_h^p)\sum_{o=1}^{N_o}\delta_o^p w_{ho} = y_h^p(1-y_h^p)\sum_{o=1}^{N_o}\delta_o^p w_{ho}. \tag{21}$$

2.4.3.4. Aprendizaje por datos de entrenamiento.-

El método de back-propagation ejecuta un descenso por el gradiente en la superficie del error total siempre que los pesos sean ajustados después de que se haya utilizado el conjunto completo de datos de entrenamiento. Lo más común es que la regla de aprendizaje sea aplicada a cada dato de entrenamiento por separado, es decir, se utiliza un dato p, E^p es calculado, y los pesos son adaptados (p=1,2,…,P). Existen resultados

empíricos que indican que esto produce una convergencia más rápida. Es de notar que se deberá tener cuidado con el orden en el cuál los datos de entrenamiento son utilizados. Por ejemplo, cuando se usa la misma secuencia una y otra vez, la red se puede enfocar más en los primeros datos. Este problema se puede superar fácilmente permutando el conjunto de datos de entrenamiento.

CAPÍTULO 3

3. SERIES DE TIEMPO

Las series de tiempo son un registro metódico a intervalos de tiempo fijos de las características de una variable, o su observación numérica. Se usan para describir y analizar fenónemos a través del tiempo. Las series de tiempo son en definitiva procesos estocáticos, pero con la restricción de que estén indexados por el tiempo y que los cortes se hagan a intervalos fijos. Definiremos a continuación un proceso estocástico.

3.1 Procesos Estocásticos

Un proceso estocástico es una familia de variables aleatorias asociada a un conjunto índice de números reales: es decir a cada elemento del conjunto de números índice, le corresponde una y solo una variable aleatoria.

Sea T el conjunto de números reales índice. Definimos $Z(\tau)$ como la variable aleatoria correspondiente al elemento τ de T (es decir, τ está en T). Definimos al proceso estocástico como la familia (o el conjunto) de variables aleatorias $\{ Z(\tau),\ \tau \in T \}$

Las series de tiempo discretas, $(\tau_1, \tau_2, \tau_3, \ldots, \tau_n)$ son las observaciones de una variable en el tiempo (1,2...n). El proceso estocástico respectivo será: { Z(τ_1), Z(τ_2), Z(τ_3),.....Z(τ_n) }. Es decir, una familia (o conjunto) de variables aleatorias. En lo sucesivo el nombre de la variable en (t) y su valor observado, se denotarán por $\boldsymbol{Z_t}$.

Una serie de tiempo observada es simplemente una realización de un proceso estocástico: siempre habrá un elemento probabilístico en las observaciones registradas y observadas.

El comportamiento de una variable aleatoria $(\boldsymbol{Z})$ se describe por su función de densidad $\boldsymbol{f(Z)}$. El comportamiento de dos variables aleatorias $\boldsymbol{Z_1, Z_2}$, queda descrito por su función de densidad conjunta $\boldsymbol{f(Z_1, Z_2)}$, y si éstas son variables aleatorias independientes: $f(Z_1, Z_2) = f(Z_1) \cdot f(Z_2)$.

En el análisis de series de tiempo se establece el supuesto de que las observaciones no provienen de variables aleatorias independientes: se supone que existe toda una estructura de correlación entre las observaciones; por lo que no es fácil obtener la función de densidad conjunta.

3.1.1. Ventajas de aplicar procesos estocásticos a series de tiempo:

- Flexibilidad para representar un amplio número de fenómenos mediante una sola clase general de modelos.
- Facilidad y precisión en pronósticos.
- Generalizar métodos de análisis de variables individuales a grupos de variables.

Ahora analizaremos los procesos determinísticos. Para ello, vamos utilizar ciertas herramientas de las ecuaciones en diferencia.

- Su solución.
- Condiciones para llegar al equilibrio.

“Temporalmente” hacemos caso omiso a la aleatoriedad. Luego, ya haremos un símil entre equilibrio y estacionariedad.

3.2 Ecuaciones en diferencia:

Operador incremento Δ:

$$\Delta \boldsymbol{Z}_{\boldsymbol{t}} = \boldsymbol{Z}_{\boldsymbol{t}+1} - \boldsymbol{Z}_{\boldsymbol{t}}$$

La ecuacion: $\boldsymbol{Z}_{\boldsymbol{t}} = \boldsymbol{a}_0 + \boldsymbol{a}_1 \boldsymbol{Z}_{\boldsymbol{t}-1}$

Se puede escribir como: $(1 - \boldsymbol{a}_1 \boldsymbol{B})\boldsymbol{Z}_{\boldsymbol{t}} = \boldsymbol{a}_0$

La solución general de dicha ecuación es:

$$\boldsymbol{Z}_{\boldsymbol{t}} = [\boldsymbol{a}_0 / (1 - \boldsymbol{a}_1)] + \boldsymbol{s}\boldsymbol{a}_1^{\boldsymbol{t}}$$

Es fácil probar que la solución de la ecuación diferencial 1, está dada por la ecuación 2 : aplicando a la ecuación 2 el operador; $1 - \boldsymbol{a}_1 \boldsymbol{B}$ se obtiene la ecuación (1).

La solución particular requiere información adicional para calcular el valor de $(\boldsymbol{s})$. Más generalmente, conociendo la condición inicial $\boldsymbol{Z}_0$, se tendría que $\boldsymbol{Z}_0 = [\boldsymbol{a}_0 / (1 - \boldsymbol{a}_1)] + \boldsymbol{s}$, y la solución de la ecuación en diferencia sería: $\boldsymbol{Z}_{\boldsymbol{t}} = [\boldsymbol{a}_0 / (1 - \boldsymbol{a}_1)] + [\boldsymbol{Z}_0 - (\boldsymbol{a}_0 / (1 - \boldsymbol{a}_1))]\boldsymbol{a}_1^{\boldsymbol{t}}$ si suponemos que $|\boldsymbol{a}_1| < 1$, cuando $\boldsymbol{t} \to \infty$; $\boldsymbol{a}_1^{\boldsymbol{t}}$ tiende a cero y $(\boldsymbol{Z}_{\boldsymbol{t}})$ tiende a: $\boldsymbol{Z}_{\boldsymbol{t}} = \boldsymbol{a}_0 / (1 - \boldsymbol{a}_1)$

El método iterativo o “pedestre”, permite también resolver una ecuación en diferencia.

$$Z_1 = a_0 + a_1 Z_0$$

$$Z_2 = a_0 + a_1(a_0 + a_1 Z_0) = a_0(1 + a_1) + a_1^2 Z_0$$

$$Z_3 = a_0 + a_1 Z_2 = a_0 + a_0(a_1 + a_1^2) + a_1^3 Z_0$$

$$Z_3 = a_0(1 + a_1 + a_1^2) + a_1^3 Z_0$$

$$Z_4 = a_0(1 + a_1 + a_1^2 + a_1^3) + a_1^4 Z_0$$

$$Z_t = a_0(1 + a_1 + a_1^2 + a_1^3 + ...a_1^{t-1}) + a_1^t Z_0$$

$$Z_t = a_0[1 - a_1^t / (1 - a_1)] + a_1^t Z_0$$

Ecuaciones en diferencia de segundo orden:

$$Z_t = a_0 + a_1 Z_{t-1} + a_2 Z_{t-2}$$

3.3. Tipos de series de tiempo:

3.3.1. Series de tiempo estacionarias.-

Una vez definida la función de autocavarianza, podremos definir la estacionariedad de un proceso estocástico, en este caso, de una serie de tiempo (más especificamente nos referimos a la estacionalidad

débil). En general, cuando se habla de estacionaridad nos referimos a estacionaridad débil.

3.3.1.1. Estacionaridad o estacionaridad débil.- la serie de tiempo $\{X_t, t \in Z\}$, donde Z es el conjunto de los números enteros es estacionaria sí y solo sí:

i) $E[X_t^2] < \infty$, $\forall t \in Z$

ii) $E[X_t] = \mu$, $\forall t \in Z$

iii) $Cov[X_s, X_t] = Cov[X_{s+h}, X_{t+h}]$, $\forall s,t,h \in Z$.

En resumen para que una serie de tiempo sea estacionaria se necesita que su varianza se mantenga finita, que su primer momento sea constante a lo largo del tiempo y que la covarianza entre dos variables de la serie solo dependan de el lapso entre estas y no en su ubicación en el tiempo (es decir la covarianza entre X_s y X_t solo dependedan de $(s-t)$, y no de s ni de t.

3.3.1.3. Estacionaridad Estricta.-

Se dice que la serie de tiempo $\{X_t, t \in Z\}$ es estrictamente estacionaria si la distribución conjunta de $(X_{t_1}.X_{t_2}, X_{t_3},..., X_{t_4})$ es la misma que la de $(X_{t_1+h}.X_{t_2+h}, X_{t_3+h},..., X_{t_4+h})$.

Debe notarse que en la estacionaridad estricta no se necesita que los X_t

Mantengan su varianza finita, por consiguiente la estacionaridad estricta no implica necesariamente estacionaridad débil. También es de notar que toda función no lineal de un proceso estrictamente estacionario sigue siendo estrictamente estacionario, sin embargo no sucede lo mismo con las funciones no lineales de procesos estacionarios débiles. Por ejemplo el cuadrado de un proceso estacionario débil puede no tener una varianza finita.

Las propiedades de los procesos estacionarios y no estacionarios son muy diferentes, y requieren diferentes métodos para realizar inferencias. Un procedimiento, para estudios empíricos, sencillo y útil de reconocer entre un proceso estacionario y uno no estacionario es ploteando los datos. Si una serie parece no tener una media constante o varianza, entonces es muy probable que no sea estacionaria.

3.3.1.3. Estacionariedad de las series de tiempo:

En la práctica muchas series no son estacionarias; pero sí sus primeras y segundas diferencias. El propósito de diferenciar una serie es volver estacionaria al diferencial de dicha serie. No obstante, debe recordarse que si se toman diferencias de series que ya son estacionarias; estas diferencias también serán estacionarias. Luego puede darse una sobre diferenciación de las series; lo que acarrea problemas de identificación respecto a aquel modelo que representa mejor el proceso que sigue la serie y se incrementa su varianza.

Una serie de tiempo es estacional cuando además de su tendencia y ciclo de largo plazo, muestra fluctuaciones que se repiten periódicamente.

- Observaciones mensuales: puede haber similitud de comportamiento para observaciones del mismo mes; por ejemplo, venta de juguetes en los “meses de diciembre”. También puede haber un patrón de comportamiento periódico con duración menor a un año; por ejemplo, “cada seis meses” a partir de junio. Las observaciones de los “meses de junio” y los “meses de diciembre” serán similares en su comportamiento, además de un comportamiento similar de las

observaciones de los “meses de diciembre” entre sí, y de los “meses de junio” entre sí.

- Es conveniente considerar genéricamente un periodo E donde esperamos observar comportamiento estacional de la variable.

3.3.1.4. Procesos Ergódicos

La ley de los grandes números de Kolmogorov establece que si los X_i´s son indepedientes y están todos identicamente distribuidos con media μ y varianza σ^2 entonces tenemos el siguiente límite:

$$\bar{X} = n^{-1}\sum_{i=1}^{n} X_i \rightarrow \mu .$$

En series de tiempo en lugar de tener un promedio de un conjunto de individuos en un instante dado, tendremos un promedio a través del tiempo del mismo individuo. Para entender mejor esta diferencia consideremos primero un conjunto de estudiantes formanado un curso, se tomó primero el promedio de todo el curso de cierto examen final. Luego se tomó el promedio de la carrera estudiantil de un solo estudiante. Ciertamente son medidas diferentes. En el primer caso la ley de los grandes números afirma que el promedio del curso convergerá al μ de la distribución. Para poder hacer una aplicación

de esta ley, a las series de tiempo es necesario que éstas sean ergódicas y estacionarias.

Para definir lo que es un proceso ergódico, primero debemos definir lo que son las medias temporales de un proceso. Si $Y(t)$ es un proceso estocástico, entonces su media temporal, $\langle Y(t) \rangle$, se define así:

$$\langle Y(t) \rangle = \lim_{k \to \infty} \frac{1}{2T} \int_{-T}^{T} Y(t) \cdot dt$$

Ahora si podremos definir lo que es un proceso ergódico.

3.3.1.4. Proceso ergódico

$X(t)$ es un proceso ergódico si y solo si todas sus medias estadísticas de la familia pueden ser intercambiadas por sus correspondientes medias temporales. Es decir una simple realización temporal es representativa de todo el proceso. Que un proceso sea ergódico implica que éste sea estacionario, pero al revés no.

Por lo tanto: Si la serie de tiempo X_t es estacionaria y ergódica con $E[X_t] = \mu$, entonces el promedio de la serie de tiempo converge a μ, es decir:

$$\bar{X}_t = t^{-1}\sum_{i=1}^{n} X_i \to \mu .$$

3.3.1.5. Procesos estocásticos lineales:

Ocurren choques aleatorios independientes de una variable. Estos choques, generan series de tiempo para otras ciertas variables, cuyos valores sucesivos pueden ser altamente dependientes: esta es la idea genial que introdujeron Yule(1927) y Box & JenKins (1970).

$(\boldsymbol{a_t})$ es la variable sujeta a choques aleatorios independientes. Sin embargo, los valores que se generan para $\boldsymbol{Z_t}$ relacionada con un polinomio de rezagos de $\boldsymbol{a_t}$, son altamente dependientes con valores sucesivos de $\boldsymbol{Z}$.

Por ejemplo: $a_{t,}$ la variable sujeta a choques aleatorios independientes, puede ser “la cantidad de lluvia”, la cual tiende afectar “las cosechas de maíz”. Es factible que la cosecha de maíz en el período t, denotada por Z_t esté afectada por la cantidad de

lluvia del año t, la de un período pasado y la del período antepasado; por lo que la cosecha actual y la del año próximo estarán relacionadas : ambas dependientes de las lluvias actuales y las del año pasado.

$$\boldsymbol{Z_t} = \mu + \boldsymbol{a_t} - \Psi_1 \boldsymbol{a_{t-1}} - \Psi_2 \boldsymbol{a_{t-2}} - .... = \mu + \left[1 - \Psi_1 \boldsymbol{B} - \Psi_2 \boldsymbol{B}^2 - \Psi_3 \boldsymbol{B}^3 + ...\right] \boldsymbol{a_t}$$

$$\boldsymbol{Z_t} = \mu + \Psi(\boldsymbol{B}) \cdot \boldsymbol{a_t}$$

El polinomio de retraso para $\boldsymbol{a_t}$ "convierte", a través de una relación, el proceso estocástico $\{\boldsymbol{a_t}\}$ en el proceso $\{\boldsymbol{Z_t}\}$.

3.3.1.6. Características de una serie de tiempo.-

Para realizar inferencias acerca de estas variables, muchas veces es útil "descomponer" la serie de tiempo por sus principales componentes:

- Tendencia y ciclo: representa el movimiento de largo plazo de la serie.
- Estacionalidad: representa efectos de fenómenos que ocurren o se reproducen periódicamente (fin de semana, diciembres, los viernes, etc.).
- Irregularidad: movimientos impredecibles o aleatorios.

Tendencia y ciclo y estacionalidad, conforman la parte determinística de la serie, mientras que la irregularidad representa la parte no-determinística o estocástica de la serie.

Muchos usuarios de la información se limitan a desestacionalizar las series estocásticas (en parte por la generalización de métodos de desestacionalización), sin intentar un análisis estadístico mas completo.

3.4. Modelos de series de tiempo

3.4.1. Operadores y polinomios:

Los polinomios de retraso son muy útiles, porque permiten representar en forma concisa y simple modelos que son muy valiosos (pero que parecen complejos).

- **Operador de retraso o "backward" B**, aplicable a Z_t. Nos indica que se debe retrasar la variable un periodo:

es decir, $B\,Z_t = Z_{t-1,}$

también, $B^2 Z_t = B\,[B\,Z_t] = B[Z_{t-1}] = Z_{t-2}$,

y en general, $B^k Z_t = Z_{t-k.}$

- **Operador diferencia** ∇, aplicable a Z_t. Nos indica que se debe obtener las diferencia entre Z_t y su valor rezagado:

$$\nabla Z_t = Z_t - Z_{t-1} = (1-B)Z_t$$

$$\nabla^2 Z_t = \nabla(Z_t - Z_{t-1}) = (Z_t - Z_{t-1}) - (Z_{t-1} - Z_{t-2})$$

- **Polinomios formados por observaciones presentes y pasadas, ponderadas:**

$$G(B)Z_t = Z_t - g_1 Z_{t-1} - g_2 Z_{t-2} - ...g_k Z_{t-k} = Z_t - \Sigma g_j Z_{t-j}$$

- **Polinomios de retraso racionales:**

$$G(B) = A(B)/C(B)$$

$A(B) = 1-\sum a_j B^j$; $C(B)= 1-\sum c_j B^j$

Ejemplo: probar que el siguiente operador (polinomio de retrasos) es "racional":

$G(B) = 1+ gB +g^2 B^2 + g^3 B^3 + g^4 B^4$..... Para: $|g| < 1$.

Es claro que en este caso G(B) es un polinomio de retraso racional, puesto que es equivalente a la relación :

G (B) = A (B) / C(B) =1 / (1-g B). Donde: A (B) = 1 ; C(B) = (1-g B).

Modelos Autorregresivos (AR):

- Deterministicos. Son del tipo:

$A(B)Z_t$ = constante

- Estocásticos. Se introduce una variable aleatoria:

$A(B)Z_t =$ constante $+ a_t$

Donde $\{a_t\}$ es un proceso de ruido blanco. $A(B)$ es un polinomio de rezagos de la forma: $(1-\phi_1 B-\phi_2 B^2-\phi_3 B^3+.....\phi_p B^p)$. Luego el modelo estocástico se puede escribir como:

$[1-\phi_1 B-\phi_2 B^2-\phi_3 B^3-\phi_4 B^4+....\phi_p B^p]Z_t =$ constante $+ a_t$

Si: $E(Z_t) = \mu$, para todo t, implica que la constante sea igual a $(1 - \phi_1 - \phi_2 ... \phi_p)\mu$. Y que el proceso estocástico se pueda escribir como: $\phi(B)\tilde{Z} = [1 - \phi_1 B - \phi_2 B^2 - ... \phi_p B^p]\tilde{Z}_t = a_t \quad \tilde{Z} = (Z_t - \mu)$.

El proceso es autorregresivo porque también se puede escribir como:

$$Z_t = (1 - \phi_1 - \phi_2 ... - \phi_p)\mu + \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \phi_p Z_{t-p} + a_t$$

Es importante determinar si el proceso estocástico tiene estacionariedad; es decir si (Z_t) tiende a localizarse (la mayoría de las veces) en la vecindad de su valor medio.

Tenemos:

$\phi(B)\tilde{Z} = a_t$. El proceso será estacionario si para el polinomio de rezagos, ϕ (B), equivalente a:

$\phi(B) = (1 - g_1 B)(1 - g_2 B)...(1 - g_p B)$, $|g_i| < 1$, para $i = 1,2,3...p$

Representación de procesos (o modelos) autorregresivos (AR):

$$(Z_t - \mu) = \phi_1(Z_t - \mu) + \phi_2(Z_{t-1} - \mu) + ...\phi_n(Z_{t-k} - \mu) + a_t$$

$$\left[1 - \phi_1 B - \phi_2 B^2 - \phi_3 B^3 - .....\phi_p B^p\right](Z_t - \mu) = a_t$$

$$\phi(B)[Z_t - \mu] = a_t$$

El modelo AR será de orden (1,2,3,…p), de acuerdo con el número (1,2,3,..p) de rezagos que el polinomio operador de rezagos θ(B) realice.

Modelo autorregresivo de orden uno. AR (1):

$\tilde{Z}_t - \phi\tilde{Z}_{t-1} = a_t$, que genera una serie de tiempo conocida como serie de Markov.

El proceso se puede re-escribir como: $(1 - \phi B)\tilde{Z}_t = a_t$.

La raíz de la ecuación: $1 - \phi X = 0$, deberá encontrarse fuera del circulo unitario: es decir $|\phi| < 1$.

Modelo AR (2):

$$\left[1-\phi_1 \boldsymbol{B}-\phi_2 \boldsymbol{B}^2\right]\tilde{\boldsymbol{Z}}_t=\boldsymbol{a}_t. \qquad [\boldsymbol{Yule}-1927]$$

$$\left(\tilde{\boldsymbol{Z}}_t=\boldsymbol{Z}_t-\mu\right)$$

Modelo AR (p):

$$\left[1-\phi_1 \boldsymbol{B}-\phi_2 \boldsymbol{B}^2-\phi_3 \boldsymbol{B}^3-.....\phi_p \boldsymbol{B}^p\right]\tilde{\boldsymbol{Z}}_t=\boldsymbol{a}_t.$$

$$\tilde{\boldsymbol{Z}}_t=\phi_1\tilde{\boldsymbol{Z}}_{t-1}+\phi_2\tilde{\boldsymbol{Z}}_{t-2}+\phi_3\tilde{\boldsymbol{Z}}_{t-3}.....+\phi_p\tilde{\boldsymbol{Z}}_{t-p}+\boldsymbol{a}_t$$

Modelos promedios móviles (MA).

Introducidos por Yule (1926) y Slutsky (1927). Son procesos estocásticos $\{\boldsymbol{Z}_t\}$, cuyos valores pueden ser dependientes entre sí, porque corresponden a una suma finita ponderada de choques aleatorios independientes $\{\boldsymbol{a}_t\}$.

$$Z_t = (1-\theta_1 B-\theta_2 B^2-\theta_3 B^3-\theta_4 B^4 \ldots\ldots\ldots-\theta_q B^q)\, a_t$$

El término “promedios móviles” no es estrictamente correcto, porque θ_i puede ser negativo y $\sum \theta_i \neq 1$.

Sí θ_i es menor a infinito y considerando un número finito de sumandos, el proceso es estacionario.

Representación de promedios móviles por medio de polinomios de retrasos:

$$Z_t = \mu + a_t + \theta_1 a_{t-1} + \theta_2 a_{t-2} + ... + \theta_n a_n$$

$$(Z_t - \mu) = (1 + \theta_1 B + \theta_2 B^2 + \theta_3 B^3 + ... + \theta_n B^n) a_t$$

Donde $\{a_t\}$ es una sucesión de variables aleatorias con ciertas características.

La representación compacta del modelo de promedios móviles (MA) será:

$Z_t - \mu = \theta(B) a_t$. A este modelo se le conoce como el modelo de promedios móviles, o modelo MA por sus siglas en inglés. De acuerdo con el número de rezagos que el polinomio operador de rezagos θ(B) realice (1,2,3, …q), el modelo MA alcanzará un orden (1,2,3, …q).

Modelo MA(1):

$\tilde{Z}_t = (1- \theta B)a_t = a_t - \theta a_{t-1}$

$E(\tilde{Z}_t) = 0$; VAR $\tilde{Z}_t = (1- \theta^2)$VAR (a) $=(1- \theta^2)\sigma_a^2$.

La autocovarianza entre $\tilde{Z}_t$ y $\tilde{Z}_{t-k}$ será:

$E[(a_t - \theta a_{t-1})(a_t - \theta a_{t-k-1}) = -\theta\sigma_a^2$,para K=1 y cero para $K \geq 2$;

por lo que el coeficiente de autocorrelación entre Z_t y Z_{t-k} será:

ρ_k =Autocovarianza($\tilde{Z}_t$; $\tilde{Z}_{t-k}$) / VAR (a) = $-\theta(1-\theta^2)$ para K=1; cero para $K \geq 2$.

El proceso MA(1) “no recuerda” más allá de lo ocurrido el periodo anterior; es decir tiene una memoria limitada a un periodo. Note que $|\rho_k| \leq 0.5$.

Los procesos autorregresivos estacionarios pueden representarse por modelos de promedios móviles (siempre que $|\phi|$ sea menor a

uno); de igual forma los modelos MA (1) pueden representarse en forma de un modelo autorregresivo si

$|\theta| \leq 1$.

En general cuando un proceso AM(q) se puede expresar mediante un modelo AR(p) se dice que dicho proceso es invertible.

Modelo MA (2):

$\tilde{\mathbf{Z}}_t = (1- \theta B - \theta_2 B^2)\, a_t$

$\left(\tilde{Z}_t = Z_t - \mu\right)$

E $(\tilde{\mathbf{Z}}_t) = 0$; VAR$(\tilde{\mathbf{Z}}_t) = (1 + \theta + \theta^2)\ \sigma_a^2$.

Modelo MA (q)

$\tilde{\mathbf{Z}}_t = (1- \theta B - \theta_2 B^2 - \ldots.. - \theta_q B^q)\, a_t$

$\tilde{\mathbf{Z}}_t = a_t - \theta a_{t-1} - \theta_2 a_{t-2} \ldots\ldots\ldots\ldots - \theta_q a_{t-q}.$

$E(\tilde{Z}_t) = 0$ y un coeficiente de autocorrelación de la serie ρ_k >0 para K= 1,2,3 ...q y cero para $k \geq q+1$; lo que nos señala que el proceso MA (q) tiene “memoria limitada” a q periodos.

ARMA (p, q):

Modelo autorregresivo y de promedios móviles: Proceso estocástico que sigue la variable aleatoria Z_t, cuya desviación con respecto a su valor esperado μ lo denotamos por:

$\tilde{Z}_t = Z_t - \mu$

El modelo lo expresamos de la siguiente forma:

1) $\phi(B)Z_t = \theta(B)a_t$,

Donde ϕ(B), θ(B) son operadores de rezagos de orden p y q respectivamente, $\{a_t\}$ es una variable aleatoria con proceso de ruido blanco (media cero y varianza finita).

Una forma alterna de escribir el proceso que sigue la variable $\tilde{Z}_t$ sería:

1.1) $(1+\phi_1 B+\phi_2 B^2+...+\phi_p B^p)Z_t = (1+\theta_1 B+\theta_2 B^2+...+\theta_q B^q)a_t$

O bien:

1.2) $Z_t+\phi_1 Z_{t-1}+\phi_2 Z_{t-2}+...+\phi_p Z_{t-p} = a_t+\theta_1 a_{t-1}+\theta_2 a_{t-2}+...+\theta_q a_{t-q}$

El modelo ARMA (p, q) es una generalización de los modelos AR y MA, combinando ambas clases de modelos. Tal generalización

surge de observar que las series de tiempo presentan, simultáneamente, características de procesos AR Y MA. Además, el principio de parsimonia sugiere construir modelos que incluyan el menor número posible de parámetros.

Es de esperarse que no todas las series de tiempo sean estacionarias, supuesto bajo el cual está construido el modelo ARMA. No obstante, sabemos que para casi cualquier serie no estacionaria, la primera, segunda o tercera diferencia de la serie si es estacionaria. Bajo estas condiciones Yaglom (1955), consideró que si el proceso original $\{\tilde{Z}_t\}$ adolece de no estacionariedad causada por una tendencia polinomial no determinista (a la cual se le denomina no estacionariedad homogénea) es posible construir un proceso estacionario $\{W_t\}$, tal que:

2) $W_t = \nabla^d \tilde{Z}_t$, para toda t. Para esta nueva serie es posible obtener un modelo ARMA:

ϕ (B) $W_t = \theta(B)\, a_t$, equivalente a considerar el modelo ARIMA : ϕ(B) $\nabla^d \tilde{Z}_t = \theta(B)\, a_t$

En el modelo ARIMA el término “integración” proviene de que Z_t equivale a la suma de un número infinito de valores actuales y

pasados de $W_{t\,.}$ Consideremos la ecuación 2, para d =1. El valor de Z_t se puede obtener multiplicando ambos lados de dicha ecuación por el operador ∇^{-1}, obtendríamos:

$Z_t = \nabla^{-1} W_t = (1-B)^{-1} W_t = W_t + W_{t-1} + W_{t-2} + W_{t-3} + \ldots\ldots \ldots$, una suma de un numero infinito de términos.

Modelo ARIMA (p, d, q).-

Dado que en muchas ocasiones el proceso estocástico que sigue $[Z_t - \mu] = \tilde{\mathbf{Z}}_t$ no es de estacionariedad; pero si su diferencial de primero, segundo, tercer..... enésimo orden, se puede formular una generalización del modelo ARMA para llegar a lo que se conoce como modelo ARIMA.

Tendremos finalmente:

$\phi(B)[\nabla^d(Z_t - \mu)] = \phi(B)\nabla^d \tilde{\mathbf{Z}}_t = \theta B a_{t.}$

Que constituye el llamado modelo autorregresivo integrado y de promedios móviles, o modelo ARIMA por sus siglas en inglés (autorregresive, integreted, moving average).

El modelo ARIMA se describe más precisamente como: ARIMA (p, d, q). Donde p es el número de rezagos que el polinomio operador de rezagos ϕ (B) realiza, d es el número de diferenciaciones sobre $\tilde{\mathbf{Z}}_t$ que el operador ∇^d realiza y q es el número de rezagos que el polinomio operador de rezagos θ (B) realiza. (Ver figura 4).

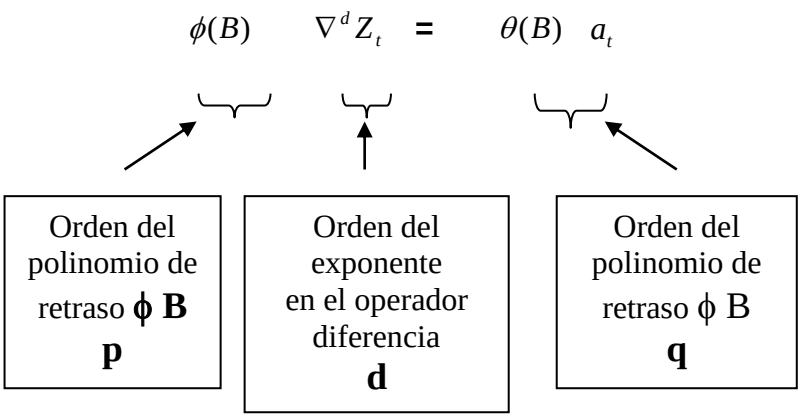


Fig. 4

Un modelo ARIMA (p, d, q) indica que el modelo consta de un polinomio autorregresivo de orden p, de una diferenciación en la variable de estudio $\tilde{\boldsymbol{Z}}_t$ de orden d, y de un polinomio de promedios móviles de orden q, tal como se muestra en la figura 3.

Modelo Multiplicativo Estacional (ARIMA (p, d , q) X (P ,D , Q)$_E$):

A fin de incorporar los efectos estocásticos estacionales y no estacionales a que están sujetos los valores observados de cierta característica de la población, o series de tiempo, Box y Jenkins (1970) propusieron un modelo general del tipo:

$$\Phi(B^E)\ \nabla_E^D\ (Z_t - \mu) = \Theta(B^E)\ \alpha_t$$

Donde las variables $\{\alpha_t\}$ no se suponen ruido blanco, sino generadas por un proceso ARIMA (p,d,q), o sea:

$$\phi(B)\ \nabla^d\ \alpha_t = \theta(B)\ a_t$$

Con (a_t) un proceso de ruido blanco. De estas dos últimas expresiones se obtiene el modelo multiplicativo estacional:

$$\phi(B)\Phi(B^E)\ \nabla_E^D\ (Z_t - \mu) = \theta(B)\Theta(B^E)\ a_t$$

El cual lo denotamos por: modelo ARIMA (p, d , q) X (P ,D , Q)$_E$.

Como es de esperarse, a mayor complejidad del modelo corresponde una estructura de autocorrelación más compleja.

El modelo ARIMA multiplicativo estacional para series con observaciones mensuales permite;

1) Considerar la relación que puede existir entre las observaciones de meses contiguos dentro de los años.

2) Considerar la relación que puede haber entre años, para las observaciones de los mismos meses.

Es decir se “captura” simultáneamente, los efectos estacionales y de tendencia del proceso “multiplicativa” o de “auto-refuerzo” de manera de tales efectos.

CAPÍTULO 4

4. APLICACIÓN DE REDES NEURONALES AL PRONÓSTICO DE UNA SERIE DE TIEMPO

4.1. Definición

Las Redes Neuronales son una herramienta muy poderosa para la modelación matemática, especialmente cuando se utilizan para modelar problemas no lineales, como es el caso de las series de tiempo. Así, su propiedad de ajustarse a un conjunto de datos de entrenamiento, las hacen muy apropiadas para tratar el problema de la extrapolación o predicción de un conjunto de datos.

La predicción de eventos futuros de una serie de tiempo con ruido, generalmente se lleva a cabo utilizando diversas técnicas estadísticas. Aplicados a un problema formulado apropiadamente, los modelos de redes de neuronales que se utilizan para este fin están fuertemente

relacionados con modelos estadísticos y a estimaciones de probabilidad bayesianas *a posteriori.* Los modelos de series de tiempo han tenido gran éxito en varios problemas de reconocimiento de patrones, y fácilmente pueden generalizarse del problema de la predicción de valores de una serie temporal, al reconocimiento de patrones y tendencias. Lo que hacen las redes neuronales es mapear un número de datos previos en el valor a predecir.

En breves rasgos el proceso de predicción de una serie de tiempo utilizando redes neuronales es el siguiente: primero se recopilan la mayor cantidad posible de observaciones ordenadas e indexadas en el tiempo (a intervalos iguales). Luego se analiza el ploteo de estos datos, buscando principalmente tendencias cíclicas, esto se hace con el fin de escoger un correcto número de datos anteriores para predecir el dato actual. A continuación se escoge una arquitectura de red adecuada y definimos la red. Del conjunto de datos escogeremos los valores que conformaran el llamado “grupo de entrenamiento”, y el resto serán el “grupo de simulación”, que nos permitará el correcto cálculo de errores de la red para poderla comparar con otros modelos. Se acostumbra dejar alrededor del 30% de los datos para el grupo de simulación. El siguiente paso es “entrenar” a la red. Esto se hace aplicando los valores del grupo de entrenamiento y realizando los ajustes respectivos según

ciertos parámetros definidos previamente en la topología o arquitectura de la red. Una vez ajustados los “pesos” de la red, procedemos a simular la red y a comparar estos datos simulados con los datos pertenecientes al grupo de simulación. Si los resultados son satisfactorios el proceso de modelación de la serie temporal conlcuye.

4.2. Ventajas y desventajas.-

Una de las principales ventajas de utilizar las redes neuronales para la predicción de series de tiempo es la escencia heurística intrínseca de las redes neuronales. En este sentido, no necesitamos supuestos teóricos, tales como la normalidad, la independencia del ruido, la media cero, etc; ni tampoco es necesario reconocer o asumir estacionalidad dentro de la serie. Su modelamiento es más sencillo y general que el de modelos estadísticos más utilizados. El investigador tiene mayor libertad en cuanto a manejo de parámetros y con habilidad se puede llegar a ajustar el modelo a la serie.

Pero sin duda la mayor ventaja que presenta la modelación de series de tiempo a través de redes neuronales es la capacidad que éstas poseen para aproximar problemas no lineales, lo que los modelos ARIMA solo logran hacer hasta cierta medida. Las series de tiempo ciertamente son

mucho mejor representadas con funciones no lineales, debido a su gran cantidad de factores y variables que estas reflejan.

Como desventajas es de notar que series con ruidos muy volátiles y no estacionales dificultan la predicción utilizando redes neuronales. Otras desventajas son:

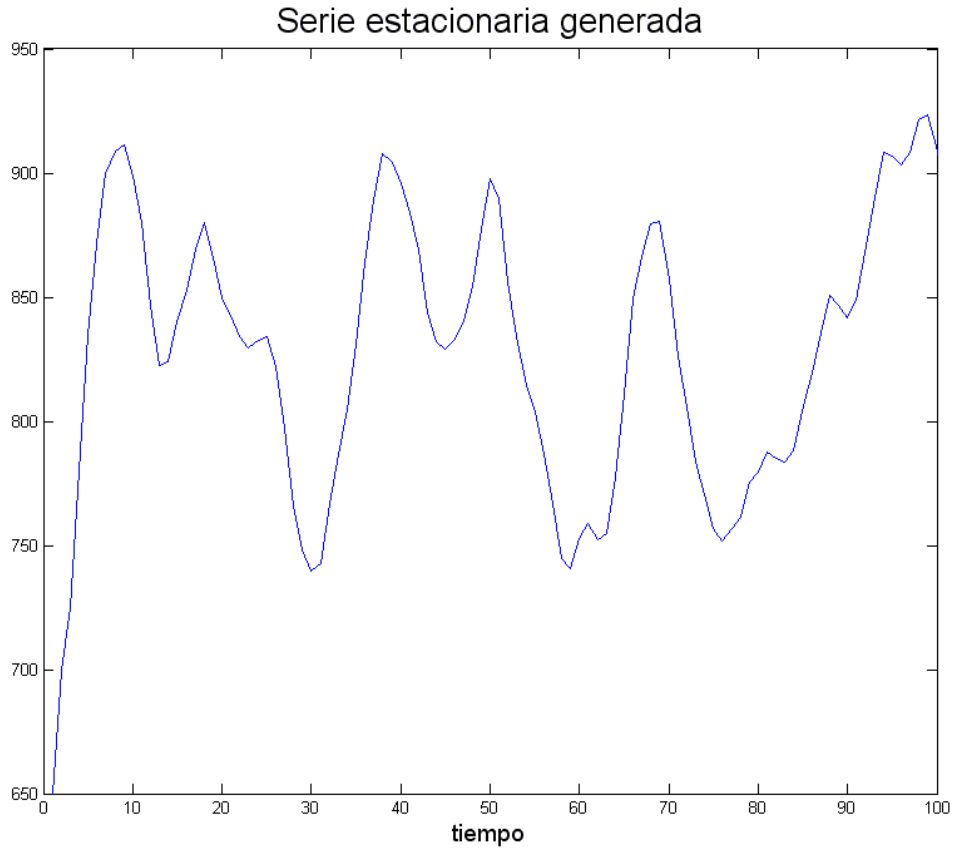
1) El problema del aprendizaje de un conjunto de datos es complejo, ya que existe un número infinito de modelos que se ajustan con la precisión deseada a los datos de entrenamiento, pero pocos de ellos logran generalizar bien esta simulación a datos fuera del conjunto de datos de entrenamiento. Se desea contar con la mayor cantidad de datos de entrenamiento, sin embargo, en series de tiempo no estacionarias el aumento de datos de entrenamiento aportará con características estadísticas que no son muy relevantes para los valores a predecir.

2) Fuertes ruidos y un conjunto de datos de entramiento pequeño hacen que el modelo no sea preciso. Correlaciones al azar entre los datos de entrada y los de salida pueden presentar grandes dificultades. Los modelos no reconocen la relación temporal explícita entre los datos de entrada y los de salida, es decir, no reconocen entre las correlaciones que ocurren en orden temporal y las que no.

4.3. Simulación de la serie y Desarrollo del modelo

Para propósitos prácticos utilizaremos una serie generada. El modelo utilizado fue un AR(4), más precisamente la fórmula utilizada para generar la serie fue $X_t = 50 + 2 * X_{t-1} - 1{,}5 * X_{t-2} + 0{,}5 * X_{t-3} - 0{,}0625 * X_{t-4} + e_t$, donde e_t es un ruido blanco con media 0 y varianza 75. Se puede comprobar fácilmente que la serie es estacionaria, para ver este resultado encontramos las raices del polinomio de resagos y verificando que efectivamente todas se encuentran fuera del círculo unitario, suficiente para la estacionalidad de la serie. En la figura 4.1 podemos ver el gráfico de la serie.

Figura 4.1
Gráfico de la serie estacionaria generada



Fuente: Pronóstico de ventas: comparación de la precisión de la predicción con diferentes métodos, TESIS 2005
Elaboración: Andrés Guillermo Abad Robalino

La serie que generamos se extiende por 100 observaciones, lo cual, para series de variables económicas no es tan fácil de conseguir. A menos que las observaciones sean mensuales, esto implicaría datos estadísticos de demasiados años.

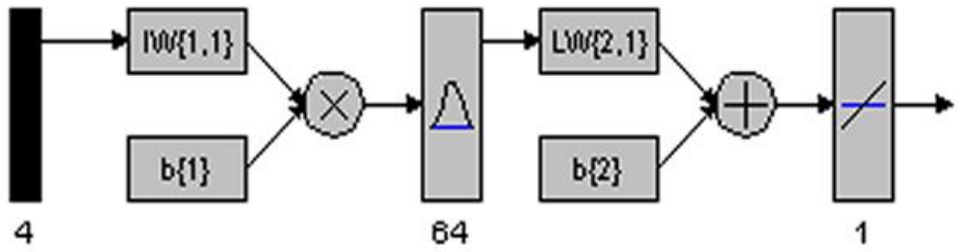
Para el modelamiento de esta serie con redes neuronales utilizaremos tres arquitecturas diferentes, para analizar las diferencias en cuanto a calidad

de predicción y eficiencia de los métodos utilizados. En todos los casos respetaremos el principio de parsimonia, tratando de reducir al mínimo el número de parámetros utilizados.

4.3.1. Red 1

La primera red que presentaremos se conoce como Radial Basis (exact fit) y no representa un modelo muy refinado. No trabajamos con ninguna regla de aprendizaje, de hecho se puede ver a esta red como un método que utiliza la fuerza bruta para la predicción. Topológicamente la red tiene dos capas: una escondida y la capa de salida. Recibe cuatro valores de entrada, X_{t-1} , X_{t-2}, X_{t-3} y X_{t-4} y entrega como salida X_t. Dado que vamos a utilizar el 70% de los datos de la serie para entrenamiento y el restante 30% para evaluar el poder de predicción fuera del conjunto de entrenamiento, la capa intermedia o escondida tiene 64 neuronas (una para cada uno de las entradas), un número un poco exagerado. Esto se debe a que la red busca ajustarse perfectamente a los datos de entrenamientos y de ahí, predecir los valores fuera de estos. La función de transferencia de la última capa es una función lineal, ya que las funciones *sigmoid* y la función *hardlimit* tienen como imagen a [-1,1], por lo tanto no podríamos predecir los valores de una serie de tiempo. En la figura 4.2 podemos ver el diseño de esta red.

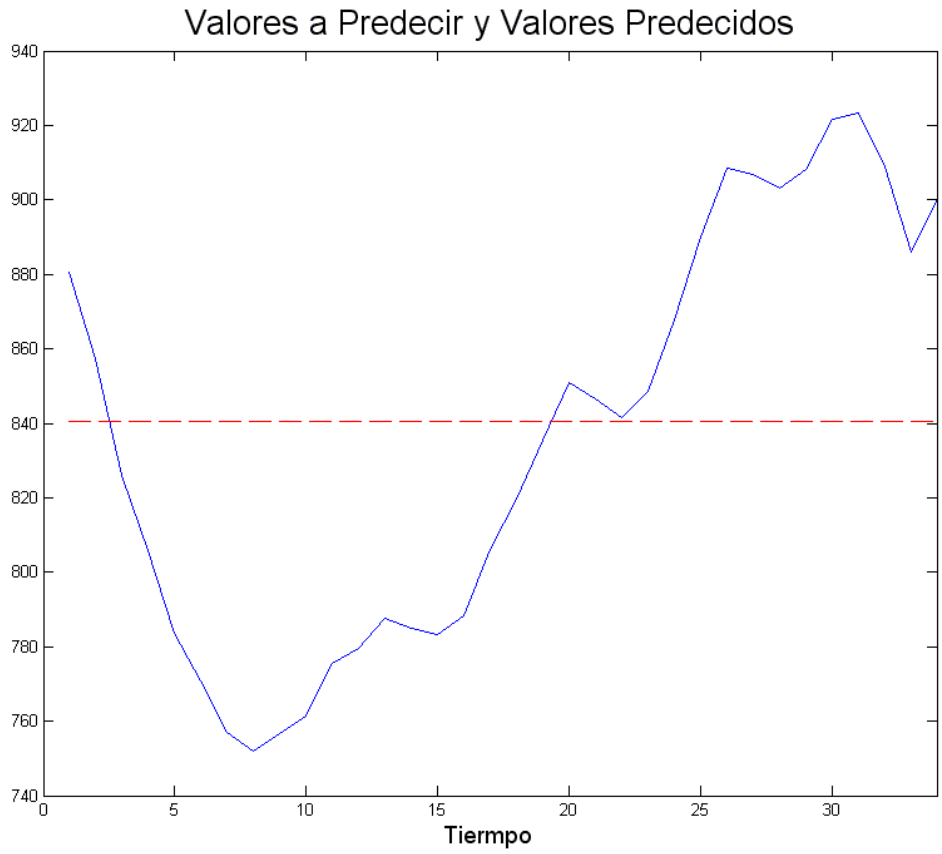
Figura 4.2
Esquema de la red neuronal de 64 nodos en la capa oculta.



Fuente: Pronóstico de ventas: comparación de la precisión de la predicción con diferentes métodos, TESIS 2005
Elaboración: Andrés Guillermo Abad Robalino

Por lo tanto cuando simulamos la red utilizando los datos con los que fue entrenada la red, el vector de errores que obtenemos es casi un vector de ceros. Lo que nos interesa es la predicción del 30% de los datos que guardamos como datos para verificar el poder de predicción de la red, y su consiguiente vector de errores. En la figura 4.3 podemos ver, en línea contínua los valores a predecir, y en línea entrecortada los valores predecidos por la red neuronal, y comprobamos que aunque el modelo predice casi a la perfección los datos de entrenamiento, su poder de predicción para valores fuera del conjunto de entrenamiento es muy pobre.

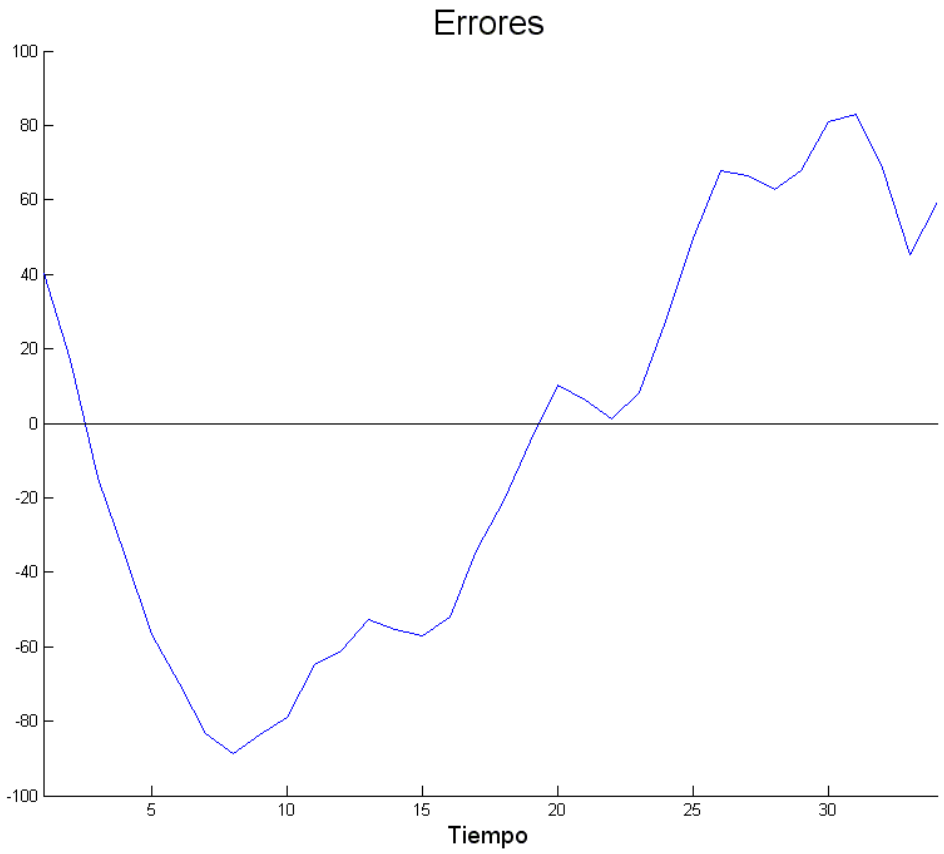
Figura 4.3
Valores a predecir y valores predichos por la red



Fuente: Pronóstico de ventas: comparación de la precisión de la predicción con diferentes métodos, TESIS 2005
Elaboración: Andrés Guillermo Abad Robalino

El modelo predice los valores como la media de los datos de entrenamiento. Analizando el vector de errores comprobamos lo que era de esperarse. El gráfico de estos errores tiene la misma forma de los datos a predecir pero desplazada y centrada alrededor del cero, esto se debe a que lo que hizo el modelo fue entregar como salida el valor medio para cada una de las predicciones.

Figura 4.4
Gráfico de los errores



Fuente: Pronóstico de ventas: comparación de la precisión de la predicción con diferentes métodos, TESIS 2005
Elaboración: Andrés Guillermo Abad Robalino

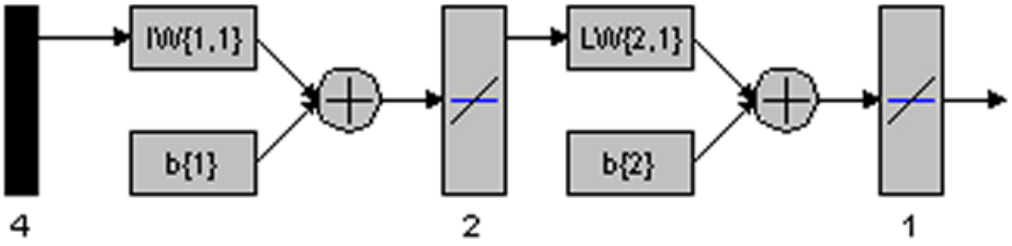
Finalmente calcularemos la suma cuadrática de estos errores, vemos que su valor es SSE=105530. Una cantidad muy grande.

En conclusión vemos que este modelo no cumple con la función de predecir datos en una serie de tiempo, trataremos de refinar este modelo para así poder ser más precisos en nuestros objetivos.

4.3.2. RED 2

Esta segunda red que analizaremos se denomina Feed Forward con Back-Propagation debido a que cada capa recibe los datos de la capa inmediata anterior, no existen conexiones de neuronas hacia atrás ni tampoco dentro de una misma capa y los ajustes se hacen comenzando por la capa de salida de atrás hacia delante y según lo influyente que es cada neurona al error total de la red. Como vemos en la figura 9 la red tiene 2 capas, una escondida y la capa de salida. Para predecir el valor X_t, la red recibe como entrada los cuatro valores previos de la serie, X_{t-1}, X_{t-2}, X_{t-3} y X_{t-4}. La función de transferencia de la capa de entrada a la capa escondida es una función lineal. En la capa intermedia o escondida tenemos dos neuronas. Finalmente la función de transferencia de esta capa a la de salida es también la función lineal.

Figura 4.5
Esquema de la red con 2 neurona en la capa escondida



Fuente: Pronóstico de ventas: comparación de la precisión de la predicción con diferentes métodos, TESIS 2005
Elaboración: Andrés Guillermo Abad Robalino

La arquitectura escogida para esta red tiene como función de entrenamiento a la función TRAINLM o función de Levenberg-Marquardt backpropagation. La función TRAINLM recibe su nombre debido a que esta ajusta los pesos y la constante de las neuronas según la regla de optimización propuesta por Levenberg-Marquadt. Esta función puede se aplicada a cualquier red neuronal que posea como funciones de transferencia a funciones derivables, como ciertamente es el caso de la función lineal.

Inicializamos los pesos de la primera capa con los valores 0.068158, -0.38142, 0.13614 y 0.40548 para la neurona 1 y 0.45423, 0.67699, -0.25917 y 0.093142 para la neurona 2 respectivamente y para la

constante de la primera capa neurona 1, -0.11024 y para la neurona 2 0.38913. Estos resultados se resumen en la tabla I.

Tabla I
Pesos iniciales de la capa 1

CAPA 1					
	W1	W2	W3	W4	B
Neurona 1	0.068158	-0.38142	0.13614	0.40548	-0.11024
Neurona 2	0.45423	0.67699	-0.25917	0.093142	0.38913

Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaboración: Andrés Guillermo Abad Robalino

Así mismo para la segunda capa o capa de salida tenemos 0.24262 y 0.58964 respectivamente. Nótese que en esta segunda capa solo tenemos dos datos de entrada. Su constante es 0.91369. Se resumen en la tabla II.

Tabla II
Pesos iniciales de la capa 2

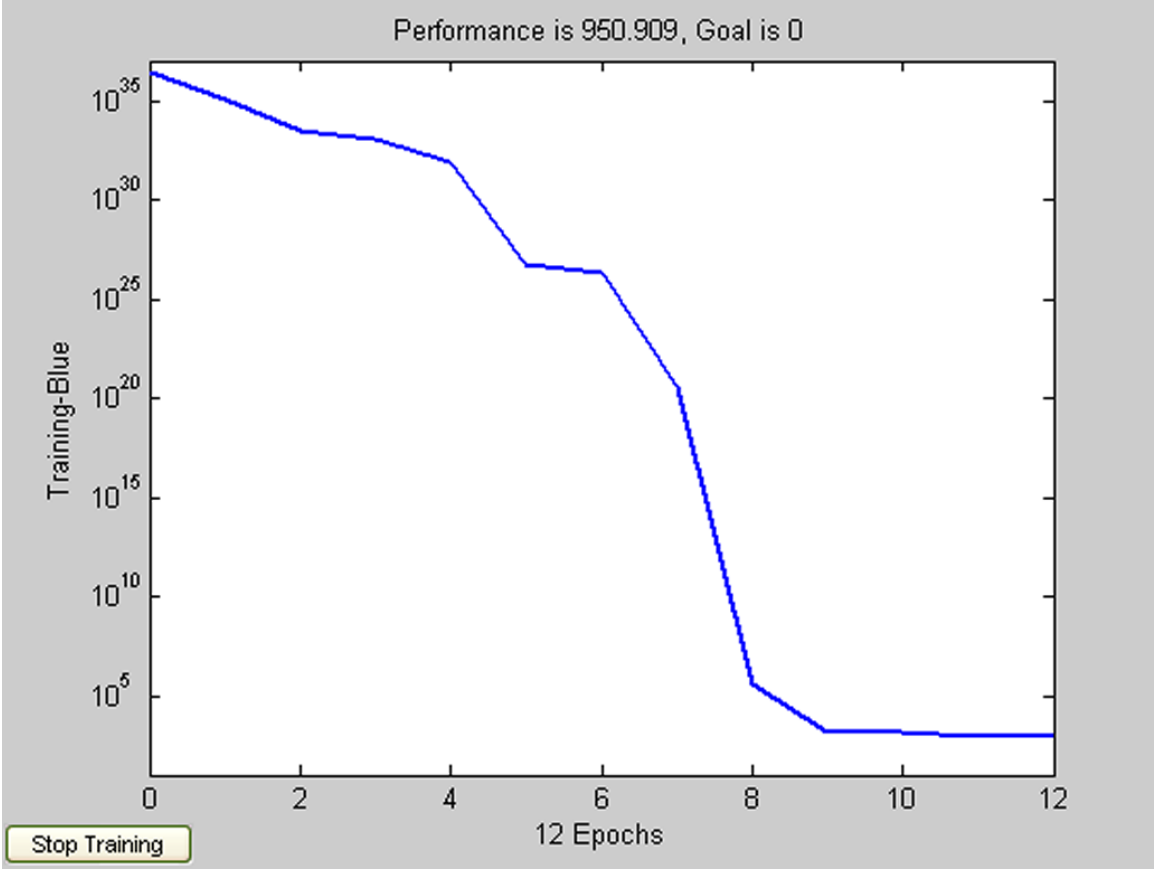
CAPA 2			
	W1	W2	B
Neurona 1	0.24262	0.58964	0.91369

Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaboración: Andrés Guillermo Abad Robalino

Estos valores fueron asignados de manera aleatoria como valores iniciales. Al realizar el entranamiento con el 70% de los datos de la serie tenemos la siguiente función de aproximación en la figura 4.6. Esta función nos muestra como se fue ajustando las salidas de la red a nuestros valores de entrenamiento.

Figura 4.6
Función de aproximación



Fuente: Pronóstico de ventas: comparación de la precisión de la predicción con diferentes métodos, TESIS 2005
Elaboración: Andrés Guillermo Abad Robalino

Vemos que el algoritmo de entrenamiento alcanzó su criterio de salida en 12 iteraciones. Los valores ajustados para los pesos y las constantes son resumidos en la tabla III.

Tabla III
Pesos ajustados de la capa 1 y 2

CAPA 1					
	W1	W2	W3	W4	B
Neurona 1	-6222926.51	3010643.54	-4867641.42	5353776.97	260316.55
Neurona 2	-10936654.07	5291138.65	-8554770.85	9409143.20	457500.51

CAPA 2			
	W1	W2	B
Neurona 1	-42934511.242	24429620.497	-2230.769

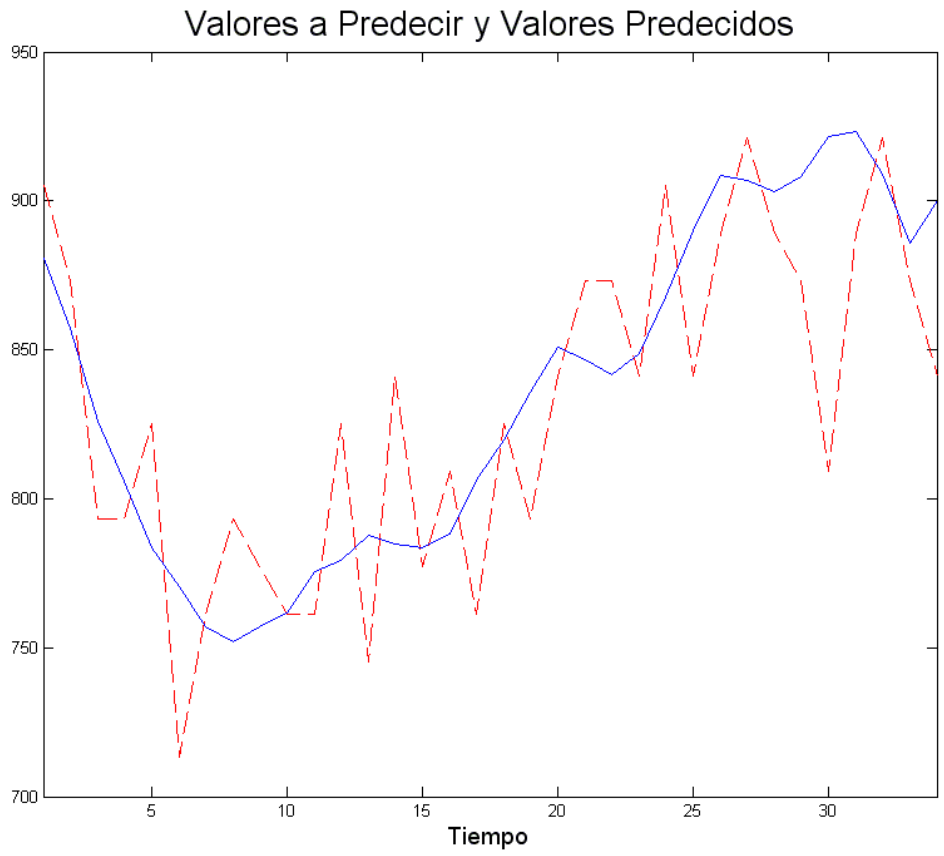
Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaboración: Andrés Guillermo Abad Robalino

Una vez entrenada la red procedemos a verificar que tan bien se ajustan las predicciones que obtenemos, a los datos de la serie. Simulamos la red con el 30% de datos que asignamos a este propósito y obtenemos un conjunto de datos simulados. En la figura 4.7 vemos que los datos pronosticados por esta segunda red se ajustan mucho mejor que los de la red anterior. Vemos que la tendencia central de la serie temporal es simulada por la red, pero la

simulación es algo errática. En el gráfico la línea entrecortada representa los datos simulados por la red, mientras que la línea contínua representa los datos reales de la serie.

Figura 4.7
Valores a predecir y valores predecidos por la red

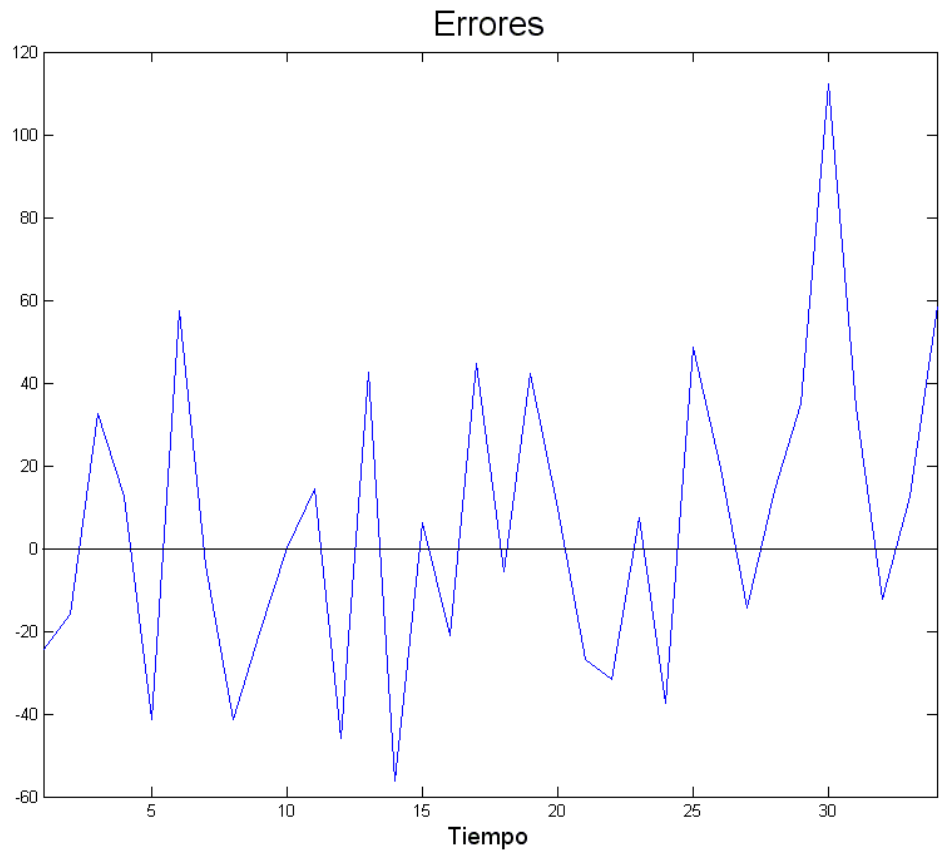


Fuente: Pronóstico de ventas: comparación de la precisión de la predicción con diferentes métodos, TESIS 2005
Elaboración: Andrés Guillermo Abad Robalino

Restando los datos simulados por la red de los datos reales de la serie obtenemos un vector de errores. Como es de esperarse queremos

que cada elemento de este vector sea tan pequeño como podamos. En la figura 4.8 vemos el ploteo de estos errores.

Figura 4.8
Gráfica de los errores para la segunda red



Fuente: Pronóstico de ventas: comparación de la precisión de la predicción con diferentes métodos, TESIS 2005
Elaboración: Andrés Guillermo Abad Robalino

Como indicador del tamaño total de error tomaremos la suma cuadrática de los errores. La SSE es simplemente la suma del cuadrado de todos los elementos del vector de errores, siendo en este caso 46003.

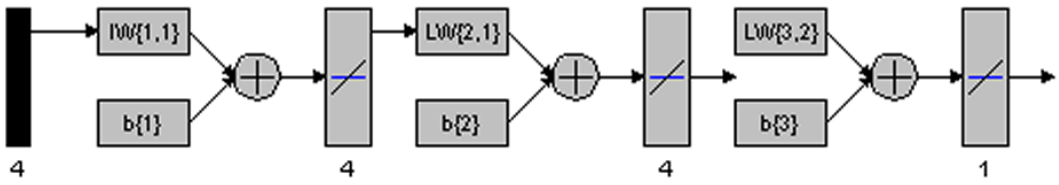
Como conclusión de este segundo modelo tenemos que se ajusta muy bien al pronóstico de los datos de la serie. La suma cuadrática de los errores en esta segunda red fue mucho menor que la de la primera, de hecho fue menos de la mitad. La figura 4.8 nos muestra que el ajuste a la serie real fue muy buena. Y vemos que el principio de parsimonia, que recomienda que la menor cantidad de parámetros necesarios predecirá mejor al modelo, nos condujo a un mejor resultado, ya que en la primera red teníamos cerca de 70 neuronas y en esta segunda red tenemos solamente 3.

4.3.3. Red 3

Una vez que hemos mejorado el primer modelo y hemos encontrado un modelo que responde bastante bien a nuestros propósitos de predicción procederemos a refinar más aún este modelo para así encontrar la arquitectura de red que más se ajuste al problema de predecir los datos de una serie de tiempo. Asumimos que los datos son trimestrales (esto está implícito en el hecho que escogemos los cuatro periodos anteriores como datos de entrada). Es de recalcar que siendo las redes neuronales un método heurístico, la habilidad del investigador en encontrar la mejor topología de red para cada problema específico es muy importante.

Vimos que el modelo feed-forward con back-propagation se ajustó bastante bien al problema de predicción, así lo que haremos ahora es refinar este mismo modelo, aumentando dos neuronas en la capa intermedia existente e incrementando una capa intermedia más, con cuatro neuronas. Utilizaremos la misma regla de aprendizaje, TRAINLM, y la función de transferencia de esta nueva capa es así mismo PURELIN o función lineal. La arquitectura de la capa de entrada se mantendrá igual, así como la arquitectura de la capa de salida. En la figura 4.9 podemos ver el esquema de esta tercera red.

Figura 4.9
Esquema de una red neuronal con dos capas intermedias



Fuente: Pronóstico de ventas: comparación de la precisión de la predicción con diferentes métodos, TESIS 2005
Elaboración: Andrés Guillermo Abad Robalino

La iniciación de los pesos se hizo aleatoriamente. Por lo general esta es la manera de hacer esta inicialización ya que es muy difícil o hasta imposible interpretar los pesos de una red, por lo tantos no se puede tener criterios *a priori* de estos valores. La tabla IV presenta estos valores iniciales de los pesos para las capas 1 ,2 y 3 respectivamente.

Tabla IV
Pesos iniciales para las capas 1, 2 y 3

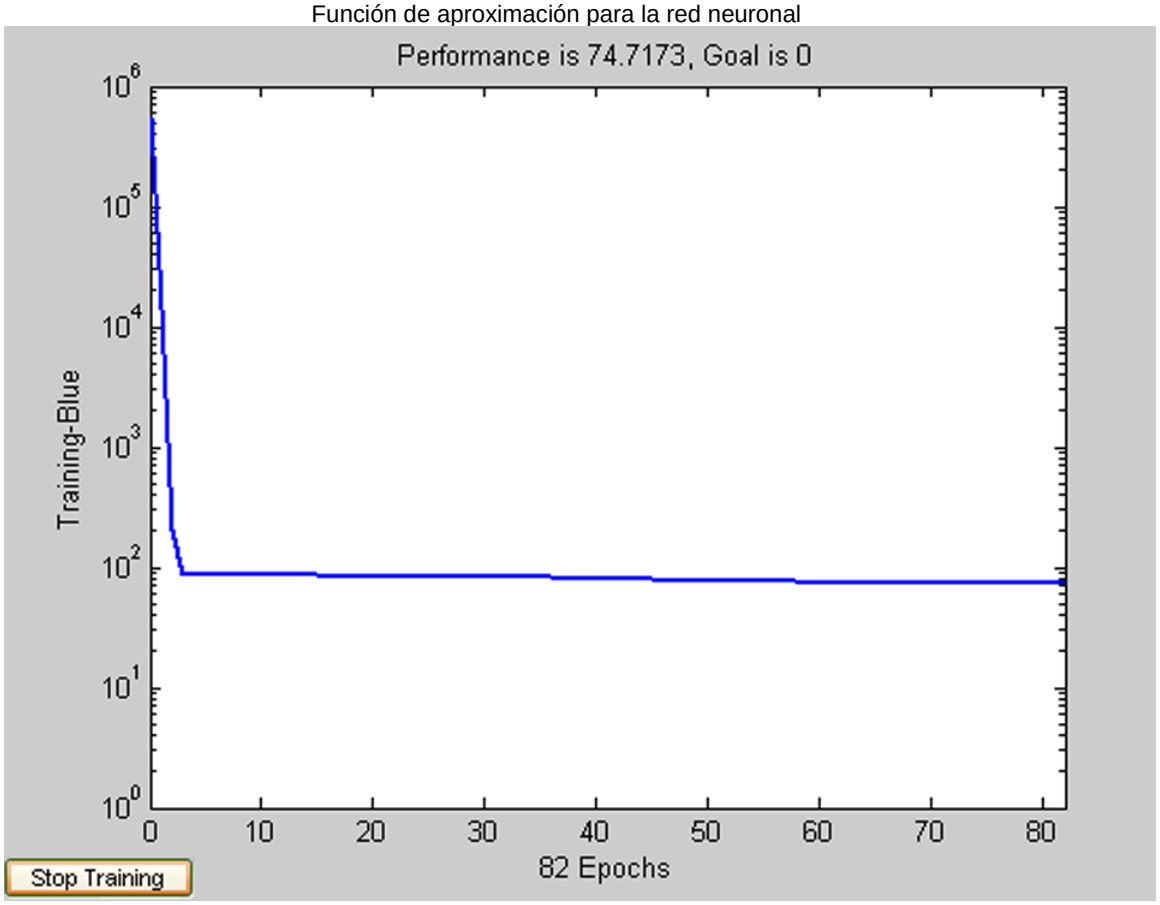
CAPA 1					
	W1	W2	W3	W4	B
Neurona 1	0.44396	-0.066475	-0.43673	0.97232	0.60903
Neurona 2	-0.38758	-0.97066	-0.47636	-0.053315	0.65773
Neurona 3	-0.77567	0.3281	0.41694	0.80564	-0.66746
Neurona 4	-0.11342	0.44812	0.56772	-0.097882	-0.21219
CAPA 2					
	W1	W2	W3	W4	B
Neurona 1	0.041515	0.10939	-0.39493	0.11588	0.95418
Neurona 2	0.43625	-0.82451	0.70369	-0.97153	-0.55618
Neurona 3	0.13838	-0.11303	0.51896	0.19235	0.40737
Neurona 4	-0.078388	-0.2674	0.89952	0.63241	0.044122
CAPA 3					
	W1	W2	W3	W4	B
Neurona 1	0.86579	0.42671	-0.54392	-0.10072	-0.6556

Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaboración: Andrés Guillermo Abad Robalino

Después de inicializar los pesos procedemos a entrenar la red con el conjunto de datos que separamos para este propósito. Para ver con que velocidad alcanzó los criterios de ajuste, ploteamos la función de aproximación y la mostramos en la figura 4.10. Vemos que en este caso esta se acerca mucho más al cero que en la red anterior, y converge asintóticamente hacia un valor por debajo de 10^2, mientras que la red 2 convergía a un valor entre 10^3 y 10^4. Aún cuando la segunda red toma más epochs o corridas para alcanzar su convergencia, ésta es mucho más cercana al cero que en la anterior.

Figura 4.10
Función de aproximación para la red neuronal



Fuente: Pronóstico de ventas: comparación de la precisión de la predicción con diferentes métodos, TESIS 2005
Elaboración: Andrés Guillermo Abad Robalino

Después de este entrenamiento los pesos y constantes han sido modificados para adaptarse a estos datos, y poder predecir valores fuera del conjunto de entrenamiento. Los nuevos valores están presentados en la tabla V, correspondiente a una de las neuronas.

Tabla V
Pesos ajustados para la capa 1, 2 y 3

CAPA 1					
	W1	W2	W3	W4	B
Neurona 1	-1.6147	-0.45804	-0.62535	0.62688	1.7845
Neurona 2	0.66942	0.73682	1.2249	0.20999	-0.63122
Neurona 3	1.1026	0.011882	-0.52073	-0.66404	1.0938
Neurona 4	0.45904	0.38124	0.56477	0.14161	3.1298
CAPA 2					
	W1	W2	W3	W4	B
Neurona 1	-0.72571	0.58185	-1.2023	-2.5696	-2.3085
Neurona 2	0.63536	0.36378	-0.047503	0.51849	2.6831
Neurona 3	-0.47288	1.0381	0.056528	-1.4685	-1.2709
Neurona 4	-0.80673	-0.41315	-0.47819	-0.83425	-0.71164
CAPA 3					
	W1	W2	W3	W4	B
Neurona 1	-3.2448	2.333	-2.0007	-0.80901	4.0984

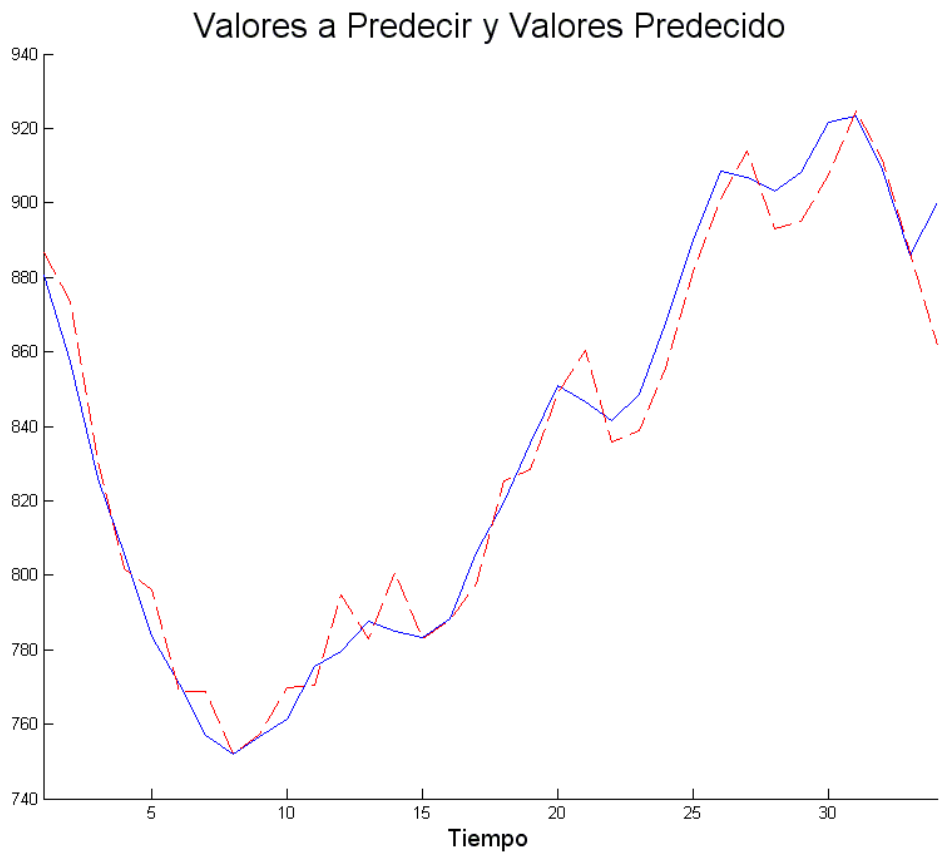
Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaboración: Andrés Guillermo Abad Robalino

Con los pesos ajustados podemos predecir o hacer una simulación de la red con los datos del conjunto a predecir. En la figura 4.11 podemos

ver que el ajuste de los valores simulados por la red, graficados con la línea entrecortada, es bastante cercana a la serie original, graficada con la línea continua.

Figura 4.11
Valores a predecir y valores predicho por la red

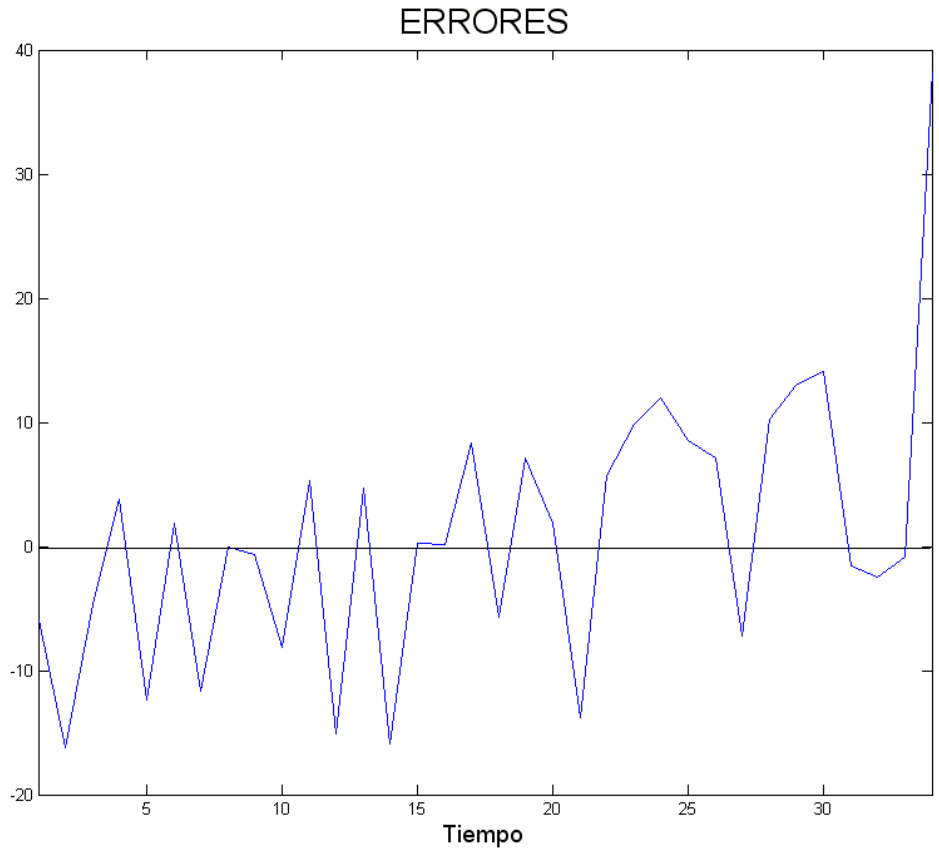


Fuente: Pronóstico de ventas: comparación de la precisión de la predicción con diferentes métodos, TESIS 2005
Elaboración: Andrés Guillermo Abad Robalino

Como podemos ver, la línea entrecortada (que representa la predicción hecha por la red neuronal) esta casi encima en cada punto de la serie original. Haciendo un gráfico de los errores vemos que

estos oscilan muy cercanos al cero. Podemos ver el plot en la figura 4.12.

Figura 4.12
Gráfico de los errores



Fuente: Pronóstico de ventas: comparación de la precisión de la predicción con diferentes métodos, TESIS 2005
Elaboración: Andrés Guillermo Abad Robalino

Y como indicador de estos errores procedemos a calcular la suma cuadrática de los errores o SSE. Y obtenemos, como era de esperarse, que es un valor bastante pequeño, mucho menor a la SSE de las otras dos redes. Y es 3973.9.

En conclusión tenemos que esta tercera red tiene un poder de predicción mucho mayor a las otras dos expuestas y es por eso que esta será la topología de red que utilizaremos para el análisis de series reales, dada por las cuentas trimestrales del Ecuador. Realmente la serie es predecida de manera muy precisa por esta tercera red. Aquí vemos todo el poder de predicción del que son capaces las redes neuronales.

4.3.4. Comparación de las tres redes expuestas

Ahora veremos en la tabla VI un cuadro comparativo de las principales características, tanto de arquitectura como de desempeño de estas tres redes.

Tabla VI
Cuadro comparativo de las principales características de las 3 redes

	Red 1	Red 2	Red 3
# de capas	2	2	3
# total de neuroanas	65	3	9
# total de pesos y constante	385	13	45
Regla de aprendizaje	-	TRAINLM	TRAINLM
Función de transferencia	PURELIN	PURELIN	PURELIN
Iteraciones en entrenamiento	-	13	85
SSE en datos de predicción	105530	46003	3973.9

Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaboración: Andrés Guillermo Abad Robalino

Analizando el cuadro comparativo vemos que la magnitud de la suma cuadrática de los errores es significativamente menor en la red 3. Aunque el número total de parámetros en la red 3 casi triplica al de la red 2 este incremento se justifica con la mejora en predicción de la red 3. También tenemos que el número de parámetros de la red 1 es inmanejable. En conclusión la red 3 es la que cumplió mejor con la función de predicción que buscábamos.

CAPÍTULO 5

5. COMPARACIÓN DE LA PRECISIÓN DE LA PREDICCIÓN CON DIFERENTES MÉTODOS A TRAVÉS DE TRES SERIES DE VENTAS REALES

En esta sección introducimos un análisis comparativo entre las técnicas convencionales de análisis de series de tiempo y las técnicas que se desprenden de la teoría de las redes neuronales para el pronóstico de valores, y que fueron analizados en el capítulo anterior. Para esto utilizaremos 3 series de tiempo reales, correspondientes a series de de ventas de diferentes compañías; medidas a intervalos trimestrales, comenzando en enero de 1990 y terminando en octubre del 2002. Las cuentas que utilizaremos serán denominadas como: SERIE 1, SERIE 2 y SERIE 3. El propósito de utilizar tres series distintas para la comparación

es precisamente tratar de que nuestro análisis sea lo más representativo de la realidad posible.

Aunque en la sección anterior ya realizamos un pequeño análisis comparativo entre los métodos convencionales y el método propuesto, la medida de comparación quizá no fue del todo representativa, ya que la serie que utilizamos fue una serie generada, ahora realizaremos las comparaciones utilizando series totalmente reales, de ventas registradas.

Los modelos ARIMA utilizados fueron construidos utilizando el software DEMETRA 2.0, el cual hace un análisis automático por diferentes modelos predeterminados y rechaza según criterios predeterminados por el usuario, los que no se ajustan a cada serie; así, al utilizar el software, encontramos el mejor modelo para cada una de las seis series de tiempo respectivamente, por lo tanto, el método de red neuronal se estará comparando al mejor modelo del método convencional de predicción.

5.1. Selección del modelo ARIMA

Los modelos ARIMA que ajusta DEMETRA son de la forma $(p,d,q)\times(P,D,Q)_s$, donde p representa el orden de la parte autoregresiva de la parte no estacionaria del modelo, d representa la diferenciación de la parte no estacionaria, q es el orden del polinomio de las medias móviles de la parte no estacionaria. De la misma manera P representa el orden de la parte autoregresiva del modelo de la parte estacionaria, D representa el orden de la diferencia de la parte estacionaria, Q representa el orden la parte de las medias móviles de la parte estacionaria del modelo y finalmente s representa el ciclo de estacionalidad del modelo.

El modelo $(p,d,q)\times(P,D,Q)_s$, expresado explícitamente tiene la forma:

$$\phi_p(B)\Phi_P(B^s)\nabla^d\nabla_s^D z_t = \theta_q(B)\Theta_Q(B^S)a_t$$

El modelo ARIMA utilizado en el proceso de desestacionalización puede ser proveído por el usuario o puede ser seleccionado en forma automática por el programa. En el caso de seleccionar la opción automática, el programa DEMETRA sigue la metodología de Box y

Jenkins para la identificación del modelo. Para determinar si el modelo seleccionado se ajusta bien a la serie, se emplean tres criterios:

1. La prueba de ajuste desarrollada por Box y Pierce, con la corrección de varianza para muestras pequeñas introducida por Ljung y Box. Con ella la hipótesis nula de aleatoriedad de los residuos es probada al 5% de nivel de significancia.
2. La media absoluta de los residuos expresados como porcentajes de los pronósticos de los últimos tres años debe ser menor o igual a 15%.
3. En los parámetros no debe haber evidencia de sobre-diferenciación. Automáticamente DEMETRA prueba, en el orden que se presentan, cada uno de los siguientes cinco modelos y el primero que se ajuste a la serie, de acuerdo a los criterios señalados arriba, es el seleccionado.

1. $L(0,1,1)\ (0,1,1)_S$
2. $L(0,1,2)\ (0,1,1)_S$
3. $L(2,1,0)\ (0,1,1)_S$
4. $L(0,2,2)\ (0,1,1)_S$
5. $L(2,1,2)\ (0,1,1)_S$

En donde L = transformación logarítmica.

El número limitado de modelos para seleccionar en el programa se debe a que, para el desarrollo del programa, originalmente se probó un conjunto de 12 modelos ARIMA en 174 series económicas que cubrían un período de 15 años y que contenían información con periodicidad mensual y trimestral. Estas series provenían de los siguientes sectores: del Sistema Nacional de Cuentas Nacionales, del sector manufacturero, precios, empleo, construcción, comercio doméstico y finanzas. Los doce modelos probados fueron:

1. $(1,1,1)\ (1,1,1)_s$
2. $(2,1,2)\ (0,1,1)_s$
3. $(2,0,1)\ (0,1,2)_s$
4. $(1,1,2)\ (0,1,2)_s$
5. $(2,0,0)\ (0,1,1)_s$
6. $(1,1,2)\ (1,0,2)_s$
7. $(2,1,1)\ (0,1,2)_s$ log
8. $(0,1,2)\ (1,1,2)_s$ log
9. $(0,1,1)\ (0,1,1)_s$ log
10. $(0,1,1)\ (1,2,2)_s$ log
11. $(1,2,2)\ (0,1,1)_s$ log
12. $(2,1,1)\ (0,1,1)_s$ log

Se encontró que el modelo No. 2 ajustaba y pronosticaba bien al 73% de las series. Para el conjunto de series que no correspondía este modelo, el modelo No. 11 produjo resultados aceptables para el 19% de las restantes (ó 5% del total). Para el resto de las series el modelo No. 9 probó ser adecuado al ajustar bien un 2% adicional del total de series. En consecuencia, los modelos 2, 11 y 9 en forma conjunta ajustaron bien al 80% de las series. Un 1% adicional podría ser ajustado por los otros nueve modelos restantes, mientras que ninguno de los 12 modelos produjo resultados aceptables para el restante 19% de las series.

En posteriores revisiones del modelo y con mayor evidencia empírica, se agregaron los otros dos modelos (indicados anteriormente) a las posibilidades de selección automática dentro del programa.

El objetivo de un procedimiento automático es encontrar modelos adecuados para una gran variedad de series a un bajo costo. En este contexto eso implica tener un número pequeño de modelos que cubran una gran cantidad de series económicas.

5.2. Selección de la Red Neuronal utilizada para cada serie.-

La selección de la topología o arquitectura de la red neuronal utilizada en cada una de las aplicaciones que se les pueda dar está íntimamente ligada al criterio, experiencia o conocimientos *a priori* del investigador. Esto se debe a que las redes neuronales son un procedimiento heurístico y éstas se presentan al usuario final como una caja negra. Para nuestros propósitos de predicción utilizaremos una arquitectura bastante parecida a la que definimos en el capítulo anterior, ya que tenemos un conocimiento *a priori* de que la serie de tiempo esta muestreada a intervalos de tiempo trimestrales, podremos asumir un ciclo de cuatro periodos (un año). Por tanto, aún cuando en ciertos casos realizaremos modificaciones a esta topología, esta será la que en general utilizaremos para predecir, y luego compararemos el poder de predicción de cada red con su respectivo modelo ARIMA utilizado.

Una vez presentada la metodología de trabajo, podemos comenzar con los pronósticos y comparaciones para cada una de las seis series.

5.3. Presentación de las Series

SERIE 1

A continuación presentamos el contenido de la serie 1:

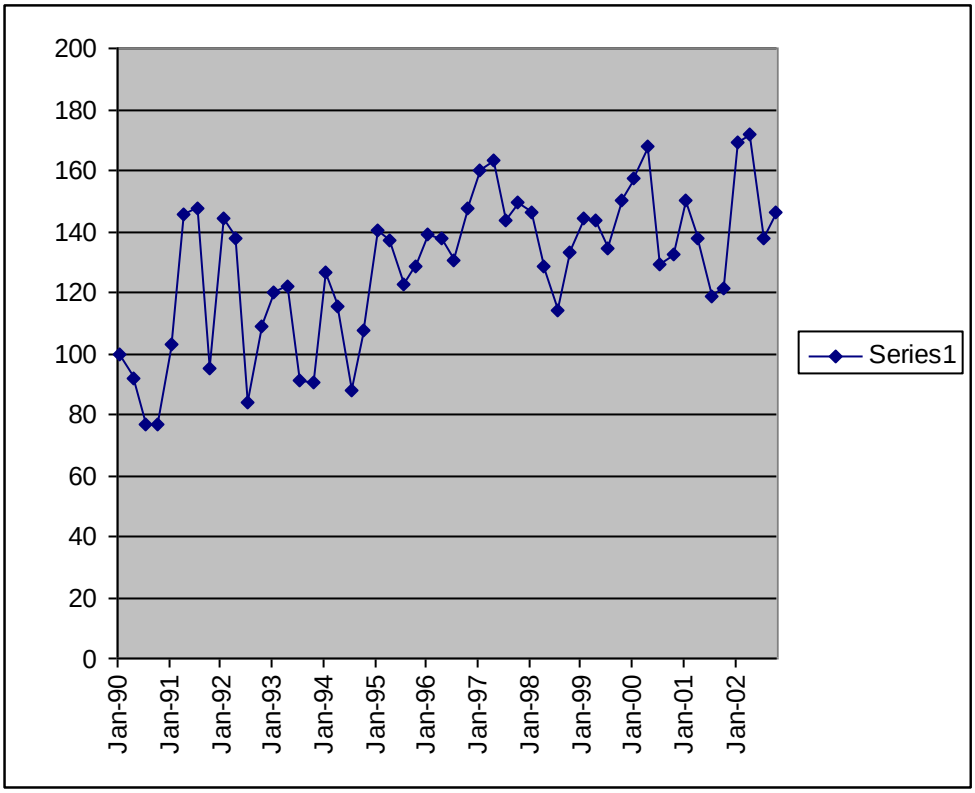
Jan-90	100	Jul-96	130.752232
Apr-90	91.95373841	Oct-96	147.492188
Jul-90	76.96847958	Jan-97	159.903543
Oct-90	76.96847958	Apr-97	163.018704
Jan-91	103.0410989	Jul-97	143.736754
Apr-91	145.4206363	Oct-97	149.279118
Jul-91	147.7205708	Jan-98	146.321632
Oct-91	94.87491882	Apr-98	128.270109
Jan-92	144.3462483	Jul-98	113.846845
Apr-92	137.3803949	Oct-98	133.432165
Jul-92	84.21904725	Jan-99	144.474315
Oct-92	108.8450641	Apr-99	143.817548
Jan-93	119.8551522	Jul-99	134.207710
Apr-93	121.6441934	Oct-99	149.972304
Jul-93	90.88041906	Jan-00	157.619301

Oct-93	90.79937709	Apr-00	167.551612
Jan-94	126.426923	Jul-00	129.493043
Apr-94	115.614807	Oct-00	132.755029
Jul-94	88.033759	Jan-01	150.333594
Oct-94	107.736126	Apr-01	137.847119
Jan-95	140.455935	Jul-01	118.464604
Apr-95	137.136725	Oct-01	121.588254
Jul-95	122.819537	Jan-02	169.4005027
Oct-95	128.297445	Apr-02	172.0863039
Jan-96	139.084372	Jul-02	137.9952987
Apr-96	137.586422	Oct-02	146.2562964

En la figura 5.1 podemos ver un gráfico de la serie.

Figura 5.1

Gráfico de la serie 1



Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaboración: Andrés Guillermo Abad Robalino

SERIE2

A continuación vemos los valores de la serie 2:

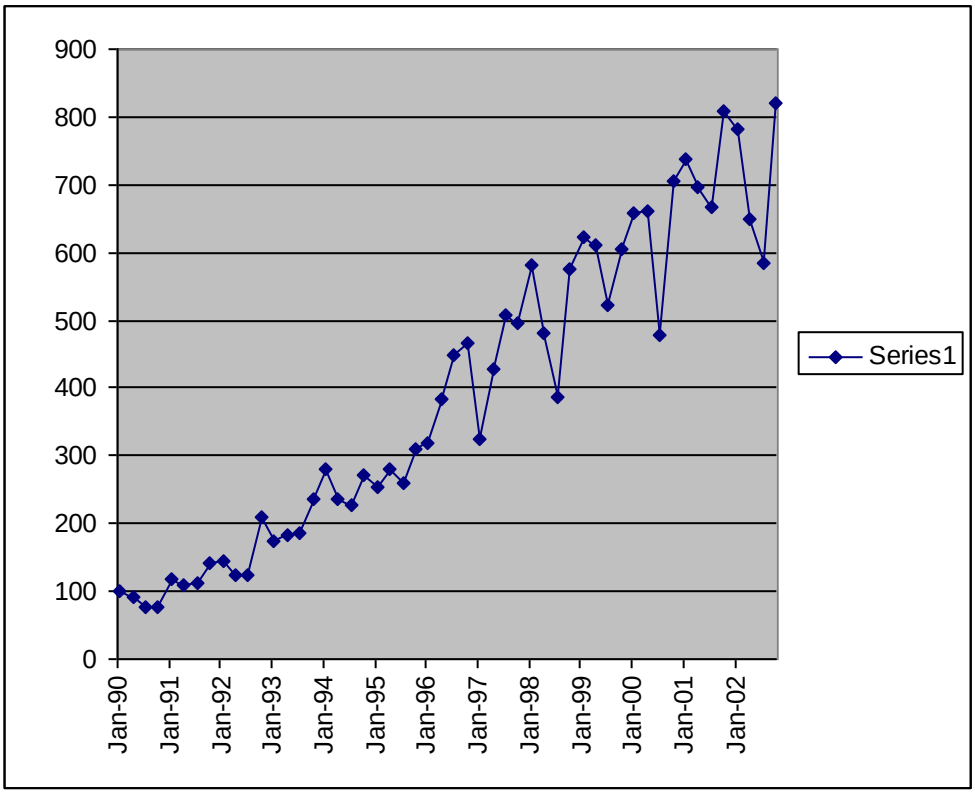
Jan-90	100	- Jul 96	448.923086
Apr-90	92.61762268	Oct-96	466.270085
Jul-90	77.7163537	Jan-97	324.592614
Oct-90	77.7163537	Apr-97	426.581873
Jan-91	118.8360378	Jul-97	506.102142
Apr-91	109.7733299	Oct-97	496.946691
Jul-91	111.3766147	Jan-98	580.354578
Oct-91	141.6257985	Apr-98	480.198385
Jan-92	144.6884819	Jul-98	386.593644
Apr-92	122.8588749	Oct-98	576.835459
Jul-92	124.6074934	Jan-99	622.392334
Oct-92	208.214218	Apr-99	610.624813
Jan-93	173.6504779	Jul-99	523.132623
Apr-93	182.5902599	Oct-99	604.024317
Jul-93	184.4847752	Jan-00	658.812270
Oct-93	235.4645388	Apr-00	659.640105
Jan-94	279.380654	Jul-00	476.961952
Apr-94	235.546394	Oct-00	704.854950
Jul-94	227.269718	Jan-01	739.044552
Oct-94	271.307229	Apr-01	696.551305

Jan-95	254.002087	Jul-01	667.207608
Apr-95	280.034077	Oct-01	809.409004
Jul-95	259.909976	Jan-02	783.2359864
Oct-95	310.741245	Apr-02	648.7294235
Jan-96	317.299897	Jul-02	585.1991767
Apr-96	383.608269	Oct-02	821.1486572

En la figura 5.2 vemos los valores de esta serie.

Figura 5.2

Gráfico de la serie 2



Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaboración: Andrés Guillermo Abad Robalino

SERIE3

Presento los valores de la serie 3 a continuación:

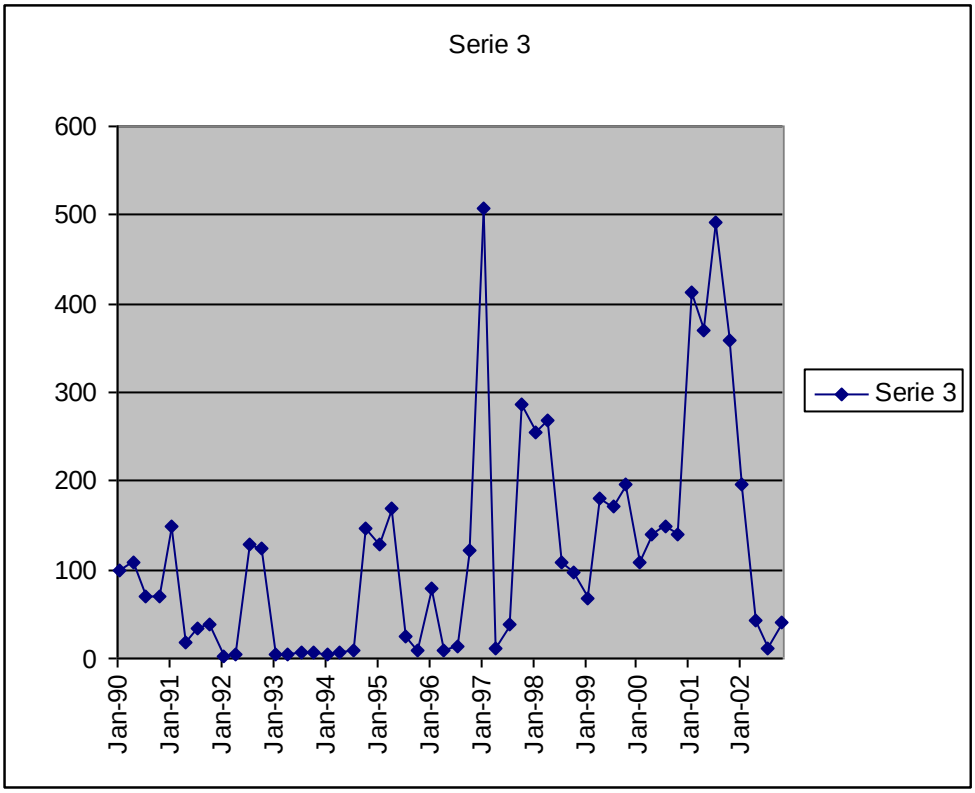
Jan-90	100	Jul-96	14.002687
Apr-90	108.0773589	Oct-96	121.909789
Jul-90	70.89469702	Jan-97	507.373502
Oct-90	70.89469702	Apr-97	11.972620
Jan-91	149.0932139	Jul-97	37.406677
Apr-91	18.15992271	Oct-97	287.218456
Jul-91	34.56823613	Jan-98	254.017082
Oct-91	38.30809076	Apr-98	268.912732
Jan-92	3.156905792	Jul-98	107.945775
Apr-92	4.449749549	Oct-98	96.445210
Jul-92	129.5637866	Jan-99	68.724776
Oct-92	124.4017249	Apr-99	179.816253
Jan-93	4.500086324	Jul-99	172.552749
Apr-93	3.896071267	Oct-99	197.039133
Jul-93	6.771816351	Jan-00	109.083972
Oct-93	6.583825313	Apr-00	140.845363
Jan-94	4.927851	Jul-00	148.267887
Apr-94	6.068284	Oct-00	139.520451

Jul-94	8.762304	Jan-01	412.466006
Oct-94	146.282722	Apr-01	368.869338
Jan-95	128.357438	Jul-01	492.436821
Apr-95	169.674343	Oct-01	357.591605
Jul-95	25.338141	Jan-02	195.1216819
Oct-95	9.483211	Apr-02	42.81410417
Jan-96	78.997812	Jul-02	10.99555799
Apr-96	9.700127	Oct-02	41.17165439

Presentamos la gráfica de la serie en la figura 5.3

Figura 5.3

Gráfico de la serie 3.



Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaboración: Andrés Guillermo Abad Robalino

Podemos ver que la serie se muestre bastante irregular y no es fácil visualizar una tendencia.

Una vez presentadas las tres series que analizaremos procedemos ahora a encontrar el modelo que mejor se ajuste a cada serie utilizando DEMETRA.

5.4. Modelo ARIMA

Basándonos en la prueba de Box y Pierce, en que la media absoluta de los residuos expresada como porcentaje de los pronósticos de los últimos tres años debe ser igual o menor a 15% y que en los parámetros no debe haber evidencia de sobre diferenciación, el modelo que DEMETRA determinó que mejor se ajustaba a cada una de las series fue:

Serie 1

Para la primera serie obtuvimos que, según los criterios predefinidos, el modelo ARIMA $(0,1,1)\times(0,1,1)_4$ es el que mejor se ajusta al pronóstico. Los valores encontrados para los parámetros del modelo son $\overline{\theta}_1 = 0.412882989659$ y $\overline{\Theta}_1 = 0.981681373220$. Así el modelo finalmente toma la forma explícita:

$$z_t - z_{t-1} - z_{t-4} + z_{t-5} = a_t - 0.41288298 \cdot a_{t-1} - 0.98168137 \cdot a_{t-4} + 0.40514891 \cdot a_{t-5}$$

La serie que ajusto este modelo esta expuesta a continuación.

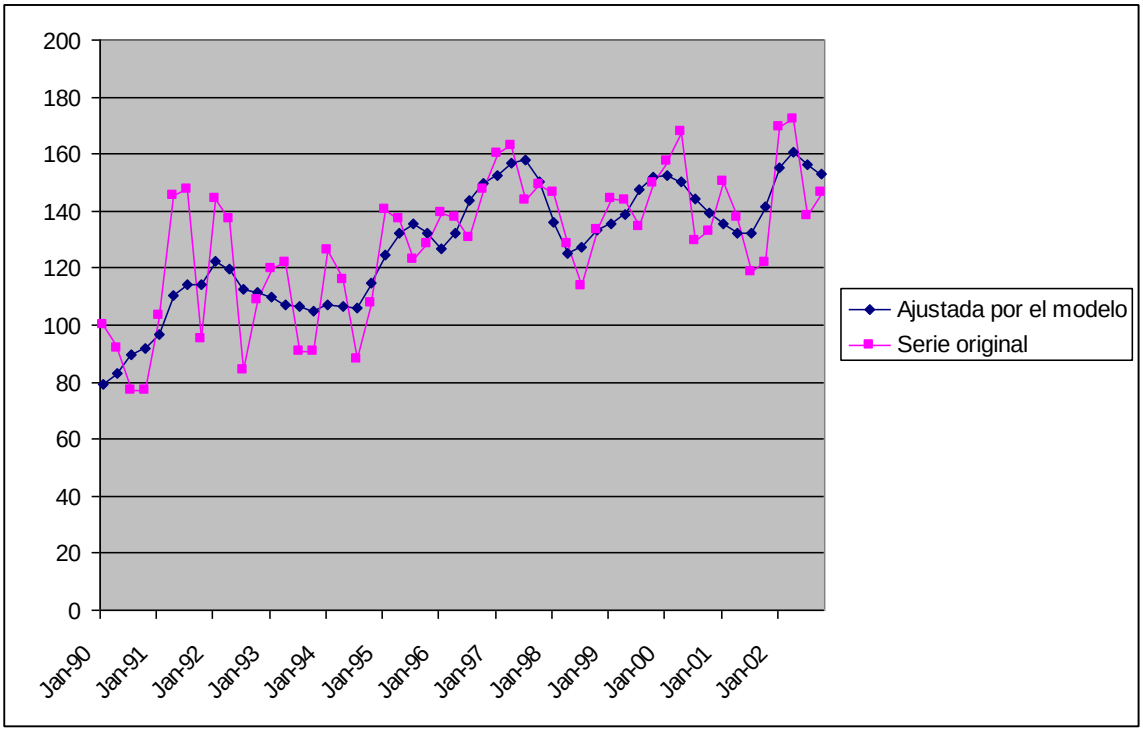
Jan-90	78.97411273	Jul-96	143.6861313
Apr-90	83.12406486	Oct-96	149.5647569
Jul-90	89.45292546	Jan-97	152.596967
Oct-90	91.96560946	Apr-97	156.7146636
Jan-91	96.45491057	Jul-97	157.8036869
Apr-91	110.4967961	Oct-97	150.131418
Jul-91	114.171541	Jan-98	135.9937795
Oct-91	114.3948142	Apr-98	125.3716432
Jan-92	122.5373217	Jul-98	127.4334933
Apr-92	119.7000108	Oct-98	133.553809
Jul-92	112.6609156	Jan-99	135.2839822
Oct-92	111.3027242	Apr-99	138.9611599
Jan-93	109.912091	Jul-99	147.6743384
Apr-93	107.2917408	Oct-99	152.147744
Jul-93	106.5597751	Jan-00	152.2421337
Oct-93	105.0468764	Apr-00	150.1028646
Jan-94	107.0696706	Jul-00	144.5144083
Apr-94	106.6963214	Oct-00	139.2387322

Jul-94	105.77493	Jan-01	135.5018392
Oct-94	114.951329	Apr-01	132.1433772
Jan-95	124.7704212	Jul-01	132.0151477
Apr-95	132.0787896	Oct-01	141.3304723
Jul-95	135.7757141	Jan-02	154.9502127
Oct-95	132.3823607	Apr-02	160.799086
Jan-96	126.9885332	Jul-02	156.3740498
Apr-96	132.3005209	Oct-02	152.7918027

En la figura 5.4 vemos el gráfico de la serie junto con el gráfico de la serie ajustada por el modelo.

Figura 5.4

Gráfico de la serie 1 y del ajuste conseguido por el modelo ARIMA



Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaboración: Andrés Guillermo Abad Robalino

Ahora calcularemos la suma cuadrática de los errores. Para esto primero calculamos el vector de errores, que no es más que la diferencia de cada elemento de la serie ajustada por el modelo con su respectivo elemento de

la serie original. Luego elevamos cada uno de estos elementos y los sumamos todos. Así obtenemos la suma cuadrática de los errores, o square sum of errors (SSE). Encontramos que para la serie ajustada por este modelo, la SSE es 10679.56881.

SERIE2

Para esta segunda serie, tenemos que el modelo que satisface mejor los criterios de selección es ARIMA $(0,1,2)\times(0,1,1)_4$. Los valores de los parámetros para el modelo son: $\theta_1 = 0.443445708728$, $\theta_2 = 0.556552843709$ y $\Theta = 0.953021898029$. Explícitamente el modelo toma la siguiente forma:

$$z_t - z_{t-1} - z_{t-4} + z_{t-5} = a_t - 0.4434 \cdot a_{t-1} - 0.5565 \cdot a_{t-2} - 0.9530 \cdot a_{t-4} + 0.42260 \cdot a_{t-5} + 0.53040 \cdot a_{t-6}$$

Los valores ajustados por esta serie se presentan a continuación:

Jan-90	94.95347327	Jul-96	462.2227821
Apr-90	93.98413891	Oct-96	432.8069225
Jul-90	93.00408001	Jan-97	386.5603018

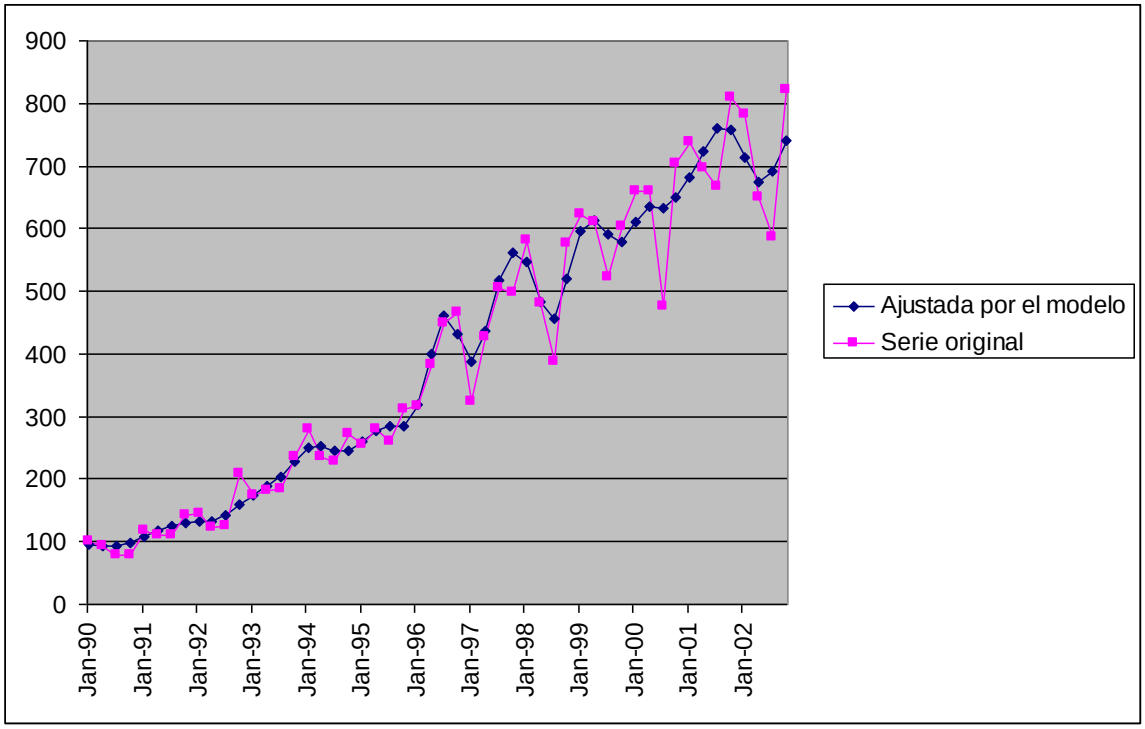
Oct-90	98.70485753	Apr-97	435.990859
Jan-91	108.9317871	Jul-97	518.2762795
Apr-91	117.7314589	Oct-97	561.1214602
Jul-91	124.9246273	Jan-98	547.7235239
Oct-91	131.1903269	Apr-98	483.463905
Jan-92	131.7563311	Jul-98	456.0836217
Apr-92	131.6772718	Oct-98	520.9487347
Jul-92	142.7683036	Jan-99	596.1037463
Oct-92	159.2670307	Apr-99	613.0372793
Jan-93	172.9562419	Jul-99	590.411484
Apr-93	187.9512661	Oct-99	578.1109517
Jul-93	203.0257546	Jan-00	610.4084868
Oct-93	227.7674761	Apr-00	633.9260585
Jan-94	250.4576862	Jul-00	633.2810141
Apr-94	252.2708368	Oct-00	650.5746881
Jul-94	245.4488079	Jan-01	681.6109099
Oct-94	246.2433301	Apr-01	723.4787711
Jan-95	258.9165939	Jul-01	759.8687611
Apr-95	277.1747269	Oct-01	758.0770403
Jul-95	283.8874906	Jan-02	712.8166961

Oct-95	285.1520817	Apr-02	673.679583
Jan-96	319.0603721	Jul-02	690.5765338
Apr-96	400.0344153	Oct-02	740.7835426

En la figura 5.5 graficaremos estos datos obtenidos del modelo, sobre los datos originales de la serie a pronosticar.

Figura 5.5

Gráfica de la serie 2 y del ajuste conseguido con el modelo ARIMA



Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaboración: Andrés Guillermo Abad Robalino

Vemos que en esta serie el modelo es bastante preciso en su ajuste. La tendencia de la serie es reflejada correctamente por los datos ajustados. Calculemos la suma cuadrática de los errores. Tenemos que SSE= 101357.7382.

Serie 3

Encontraremos ahora el modelo que más se ajusta a la serie3 utilizando los mismos criterios. El modelo que obtenemos es el ARIMA $(2,0,1)\times(0,0,0)_4$, es decir, dado que los últimos tres parámetros son todos cero, este modelo no necesita de una parte estacionaria para realizar el mejor ajuste, por ende es razonable que no tengamos tampoco orden de diferenciación, ya que este orden se utiliza precisamente para lograr la no estacionalidad de la serie. Pudimos haber previsto este resultado analizando la gráfica. Los coeficientes para el modelo encontrados son: $\phi_1 = 1.379441893074$, $\phi_2 = -0.713523503328$ de la parte autoregresiva y no estacionaria, y $\theta = 0.999926964265$ de la parte de la media móvil y no estacionaria. Explícitamente el modelo es:

$$z_t - 1.37944 \cdot z_{t-1} + 0.71352 \cdot z_{t-2} = a_t - 0.99992 \cdot a_{t-1}$$

La serie ajustada obtenida a través de este modelo es:

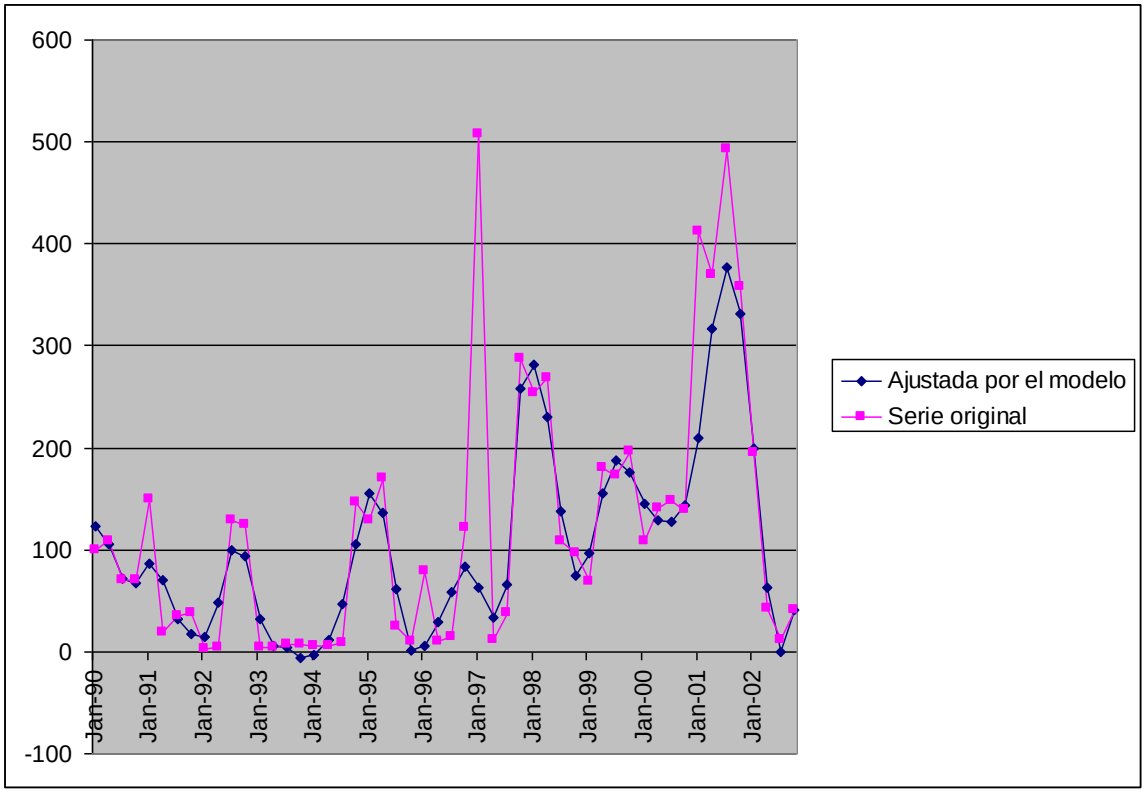
Jan-90	123.1626235	Jul-96	58.24172119
Apr-90	104.842325	Oct-96	83.42034201
Jul-90	71.20734942	Jan-97	62.69292459
Oct-90	66.73290019	Apr-97	33.1018571
Jan-91	86.16584276	Jul-97	66.11481309
Apr-91	70.24444011	Oct-97	257.9541433
Jul-91	32.05099225	Jan-98	281.1281707
Oct-91	17.90324236	Apr-98	230.3849771
Jan-92	14.36145946	Jul-98	137.628731
Apr-92	48.12355957	Oct-98	73.94435664
Jul-92	100.0192655	Jan-99	96.94218878
Oct-92	93.96780785	Apr-99	154.7326401
Jan-93	32.39378788	Jul-99	187.2896942
Apr-93	6.079273629	Oct-99	175.4240192
Jul-93	3.838247714	Jan-00	145.3117502
Oct-93	-6.191039416	Apr-00	129.0677042
Jan-94	-3.597566069	Jul-00	127.3667043
Apr-94	11.45641682	Oct-00	144.0277333

Jul-94	46.31363048	Jan-01	209.8044313
Oct-94	106.115603	Apr-01	317.4643493
Jan-95	155.0129233	Jul-01	377.1516508
Apr-95	135.6290709	Oct-01	331.8260337
Jul-95	61.51361305	Jan-02	198.9051944
Oct-95	1.868375437	Apr-02	62.6683305
Jan-96	6.264465748	Jul-02	0.322089907
Apr-96	29.40078472	Oct-02	40.88930393

Ahora en la figura 5.6 graficaremos estos datos y los compararemos con los datos de la serie original para poder visualizar que tan bueno es el ajuste de este modelo a la serie original.

Figura 5.6

Gráfico de la serie 3 junto con el ajuste conseguido por el modelo ARIMA



Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaboración: Andrés Guillermo Abad Robalino

Al ver la gráfica podemos decir que el ajuste que realiza el modelo ARIMA no es muy preciso. También es de notar que en la serie original no se ve

fácilmente alguna tendencia. La suma cuadrática de los errores de el ajuste es 293608.61.

5.4.1. Red NEURONAL

Definiremos una red para cada una de las series, y procederemos a entrenar (ajustar) la red para que “aprenda” a predecir los valores de cada serie respectivamente. Luego, calcularemos los valores de la red, y restándolos de los valores reales a predecir obtenemos el vector de errores. Finalmente calculamos la suma cuadrática de estos errores sumando todos los elementos elevados al cuadrado, para así, con la SSE como índice poder comparar éste método con el de la modelación convencional.

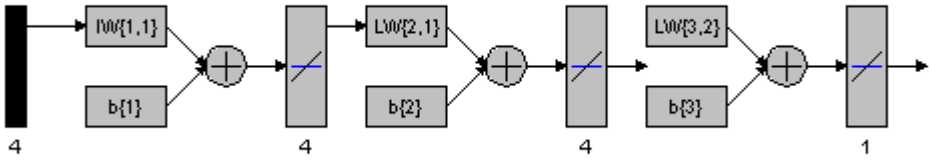
SERIE 1

Para la primera serie definiremos una arquitectura en donde disponemos de cuatro neuronas de entrada, así recibimos la información de periodicidad de ciclo anual que asumimos que la serie tiene (son cuatro dado que los datos están muestreados trimestralmente). Luego definiremos dos capas ocultas o intermedias, ambas con cuatro neuronas, y finalmente una neurona de salida. Las funciones de

transferencia son todas, la función PURELIN o función lineal, dado que el rango del conjunto de partida de la función debe ser la recta real. En la figura 5.7 se presenta el esquema de la red.

Figura 5.7

Esquema de la red neuronal utilizada en la serie 1



Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

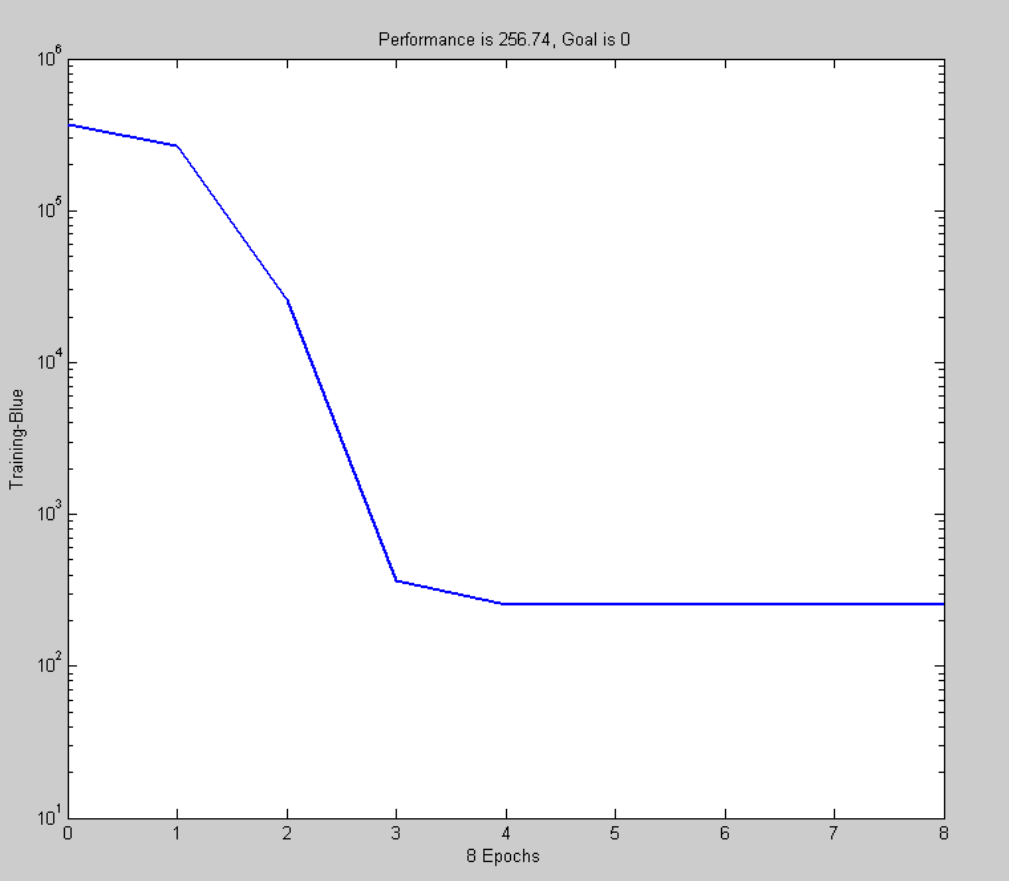
Elaboración: Andrés Guillermo Abad Robalino

Construida la red neuronal aplicaremos los datos de la serie de tiempo para entrenarla, es decir que los pesos de las neuronas se ajusten y minimicen el error total. Es de interés la función de ajuste, que nos muestra el número de iteraciones necesarias para que la red neuronal se

estabilice. En la figura 5.8 presentamos la función de entrenamiento para la serie1.

Figura 5.8

Función de entrenamiento de la red para la serie 1



Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaboración: Andrés Guillermo Abad Robalino

En la gráfica vemos que fue necesario un total de ocho iteraciones para lograr el ajuste de la red según criterios predefinidos. En la tabla VII se resumen los resultados de este entrenamiento. Presentamos los valores de los pesos ya ajustados de cada una de las tres capas de la red neuronal respectivamente.

Tabla VII

Pesos ya ajustados de la red utilizada en la serie 1

CAPA 1					
	W1	W2	W3	W4	B
Neurona 1	-0.5552	-0.68794	0.59232	0.88507	0.69437
Neurona 2	0.44318	0.1267	-0.20228	0.22625	-0.89903
Neurona 3	-0.37885	-0.52389	-0.27085	0.31179	-1.8959
Neurona 4	-0.023034	0.33516	0.70719	-0.64643	-4.1954
CAPA 2					
	W1	W2	W3	W4	B
Neurona 1	0.48125	0.25765	1.1676	0.37112	-2.3756
Neurona 2	-0.19899	1.0909	1.1056	0.87478	-2.9795
Neurona 3	0.52034	-0.45987	-0.094864	-0.72528	2.7495
Neurona 4	0.56348	0.29765	1.0526	0.33157	-0.85541

CAPA 3					
	W1	W2	W3	W4	B
Neurona 1	-2.0783	-2.3561	1.7887	0.1304	2.0437

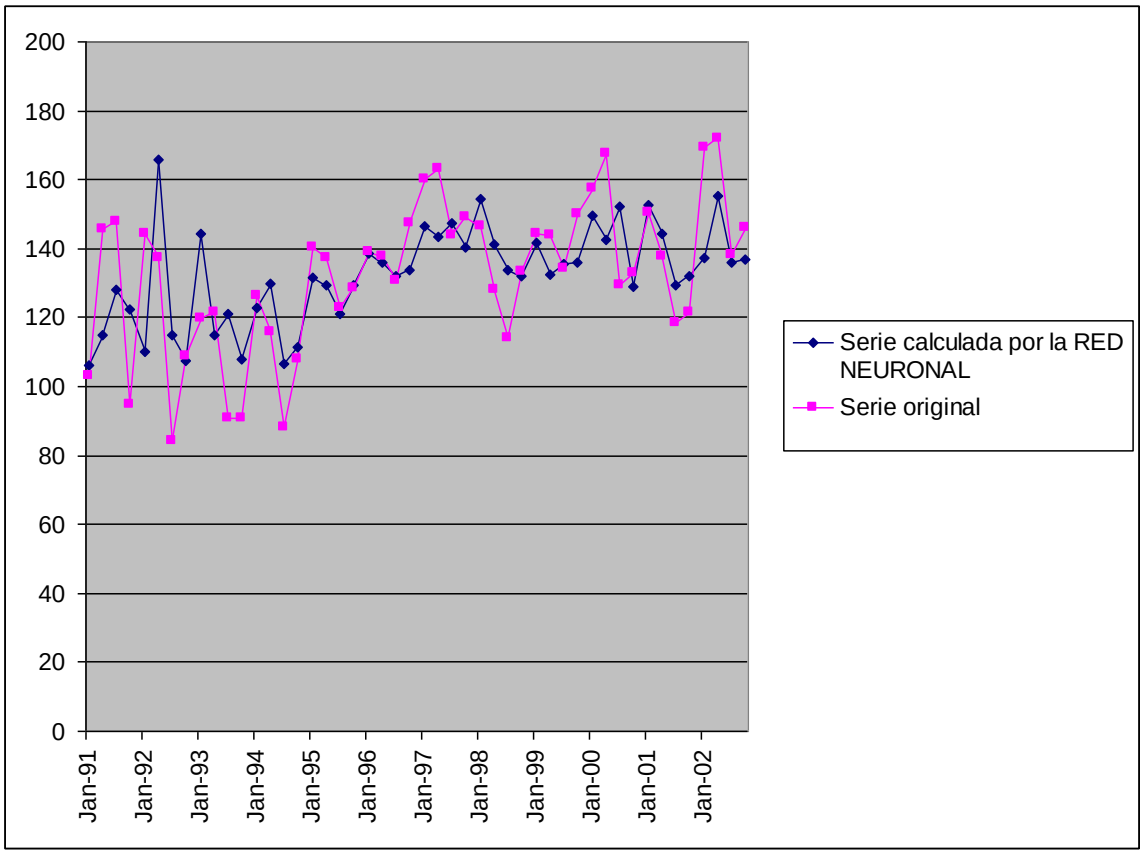
Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaboración: Andrés Guillermo Abad Robalino

Estos pesos minimizan el error total de la red neuronal para la serie1. Ahora obtendremos los valores de la serie1 calculados por la red. En la figura 5.8 graficamos los datos de la serie calculados por la red junto con los datos originales de la serie.

Figura 5.8

Gráfica de la serie 1 junto con el ajuste conseguido por la red neuronal



Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaboración: Andrés Guillermo Abad Robalino

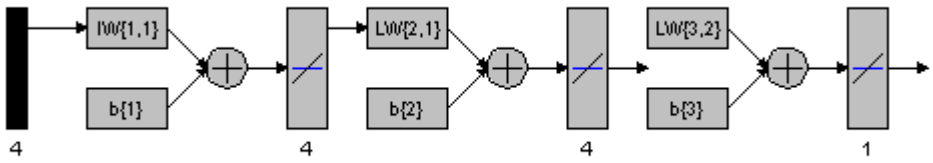
Visualmente el ajuste que obtuvimos con la red neuronal es bastante preciso. A continuación obtendremos la suma cuadrática de los errores de la serie. SSE= 12324,44.

SERIE 2

Para esta segunda serie, utilizaremos una topología de red similar a la que utilizamos en la aplicación teórica de las redes neuronales. En la figura 21 vemos que la serie tiene una marcada tendencia y vemos algo de periodicidad, por lo cuál las cuatro neuronas de entrada se ajustaran bien a la serie. En esta segunda red neuronal utilizaremos como función de transferencia para todas las capas a la función PURELIN. En la figura 5.9 vemos el esquema de esta red.

Figura 5.9

Esquema de la red neuronal utilizada en la serie 2



Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

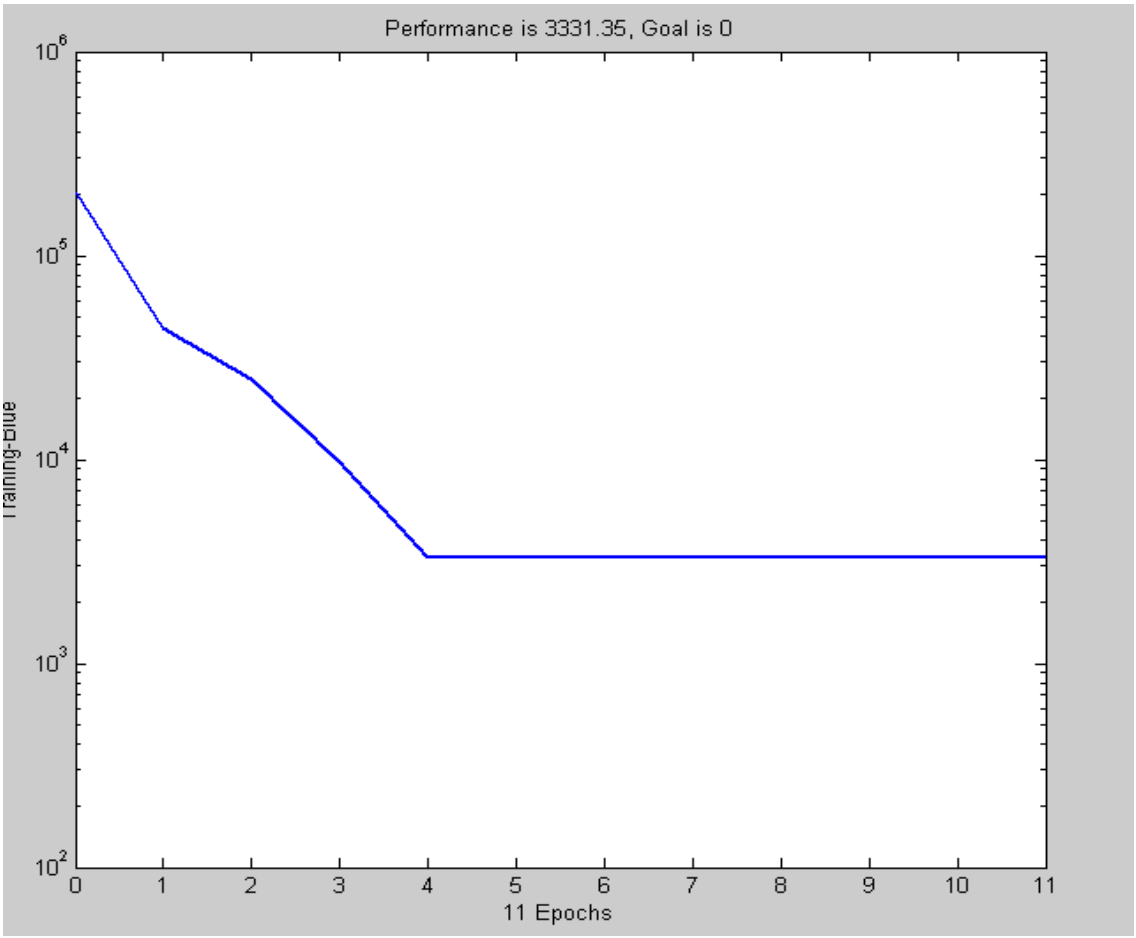
Elaboración: Andrés Guillermo Abad Robalino

Con la topología de la red definida debemos entrenarla con el conjunto de datos de la serie2. Es de interés conocer el tiempo, en este caso medido en iteraciones, en el cual la red alcanza sus criterios de ajuste

predefinidos por el investigador. Para esto graficaremos la función de ajuste obtenida al entrenar a esta red. En la figura 5.10 vemos esta función.

Figura 5.10

Función de entrenamiento de la red utilizada en la serie 2



Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaboración: Andrés Guillermo Abad Robalino

Analizando la figura 27 vemos que fueron necesarias once iteraciones para ajustar la red, es un tiempo bastante corto. Con los pesos de cada neurona encontrados ya podemos calcular los valores de una red. En la tabla VIII presentamos estos valores.

Tabla VIII

Pesos ya ajustados de la red utilizada en la serie 2

CAPA 1					
	W1	W2	W3	W4	B
Neurona 1	0.64663	-0.062417	-0.032797	-0.003846	2.0286
Neurona 2	1.209	0.080937	-0.84724	-0.13681	2.2462
Neurona 3	-0.35117	0.43902	0.52334	0.16185	1.0027
Neurona 4	-0.56858	-0.37391	-0.14477	-0.018309	1.0185
CAPA 2					
	W1	W2	W3	W4	B
Neurona 1	0.89829	0.7053	0.95924	0.99302	3.9611
Neurona 2	-0.3172	0.14712	-0.49814	0.13704	-1.1463
Neurona 3	-0.16052	-0.71194	0.29301	0.2274	-0.28459
Neurona 4	-0.33329	1.1767	0.47571	0.98914	1.2898

CAPA 3					
	W1	W2	W3	W4	B
Neurona 1	4.1439	-0.21466	1.4644	0.21707	3.2365

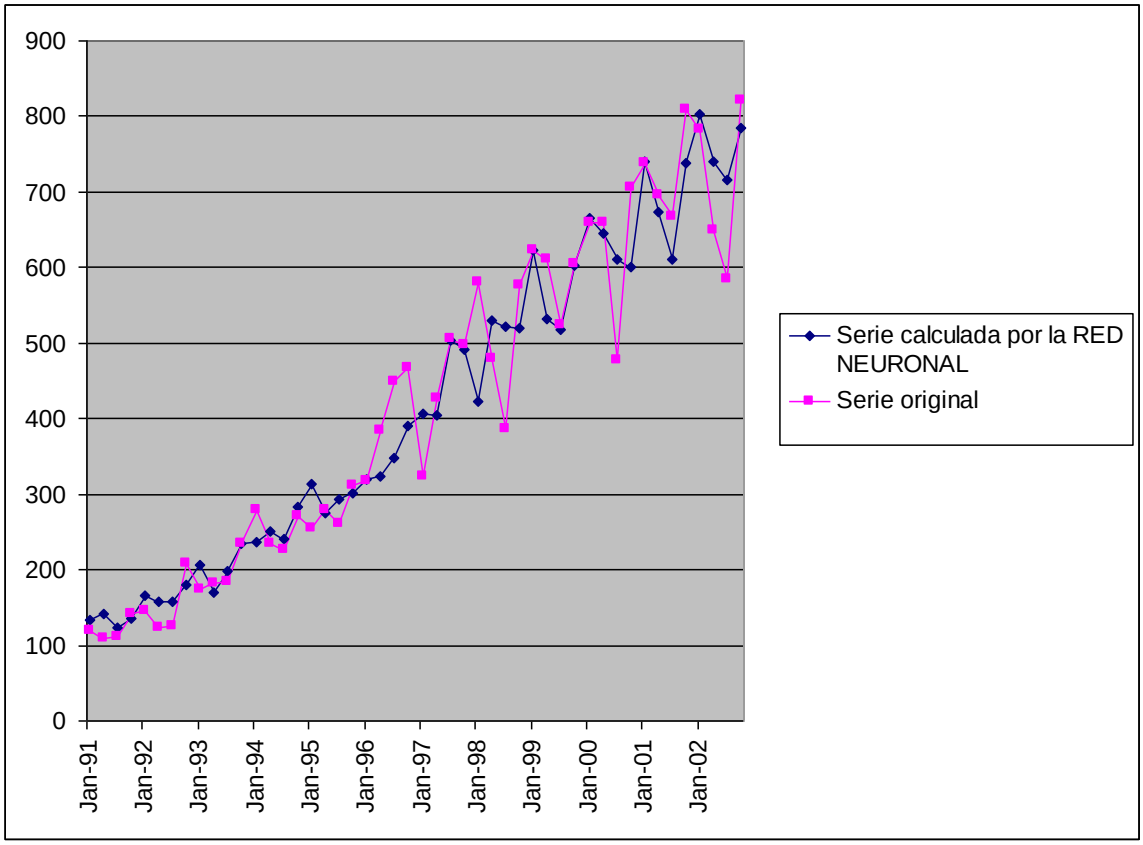
Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaboración: Andrés Guillermo Abad Robalino

Con los pesos ajustados ya podemos calcular la serie y compararla con la serie original. En la figura 5.11 vemos las dos series: la serie generada por la red neuronal y la serie objetivo.

Figura 5.11

Gráfica de la serie 2 junto con el ajuste conseguido por la red neuronal



Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaboración: Andrés Guillermo Abad Robalino

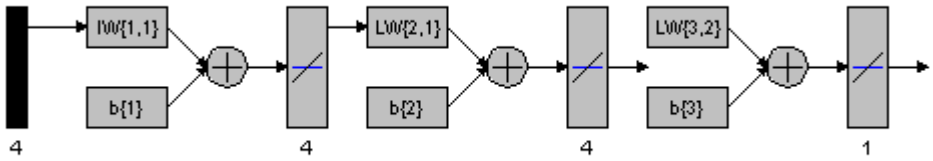
En la figura 5.11 vemos que la red neuronal capturó bien la tendencia de la serie, sin embargo vemos que el ajuste no es tan preciso. La suma de los errores cuadráticos del ajuste fue SSE= 159901.82 .

SERIE 3

Para la tercera red utilizaremos la misma arquitectura que utilizamos en las dos series anteriores con el fin de ver como se ajusta a esta serie específica. Como se puede ver en la figura 5.3 la serie3 es bastante volátil. No se aprecia tendencia ni periodicidad, por lo tanto su predicción o ajuste probablemente no será muy preciso. En la figura 5.12 presentamos la arquitectura de esta tercera red neuronal.

Figura 5.12

Esquema de la red neuronal utilizada para la serie 2



Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

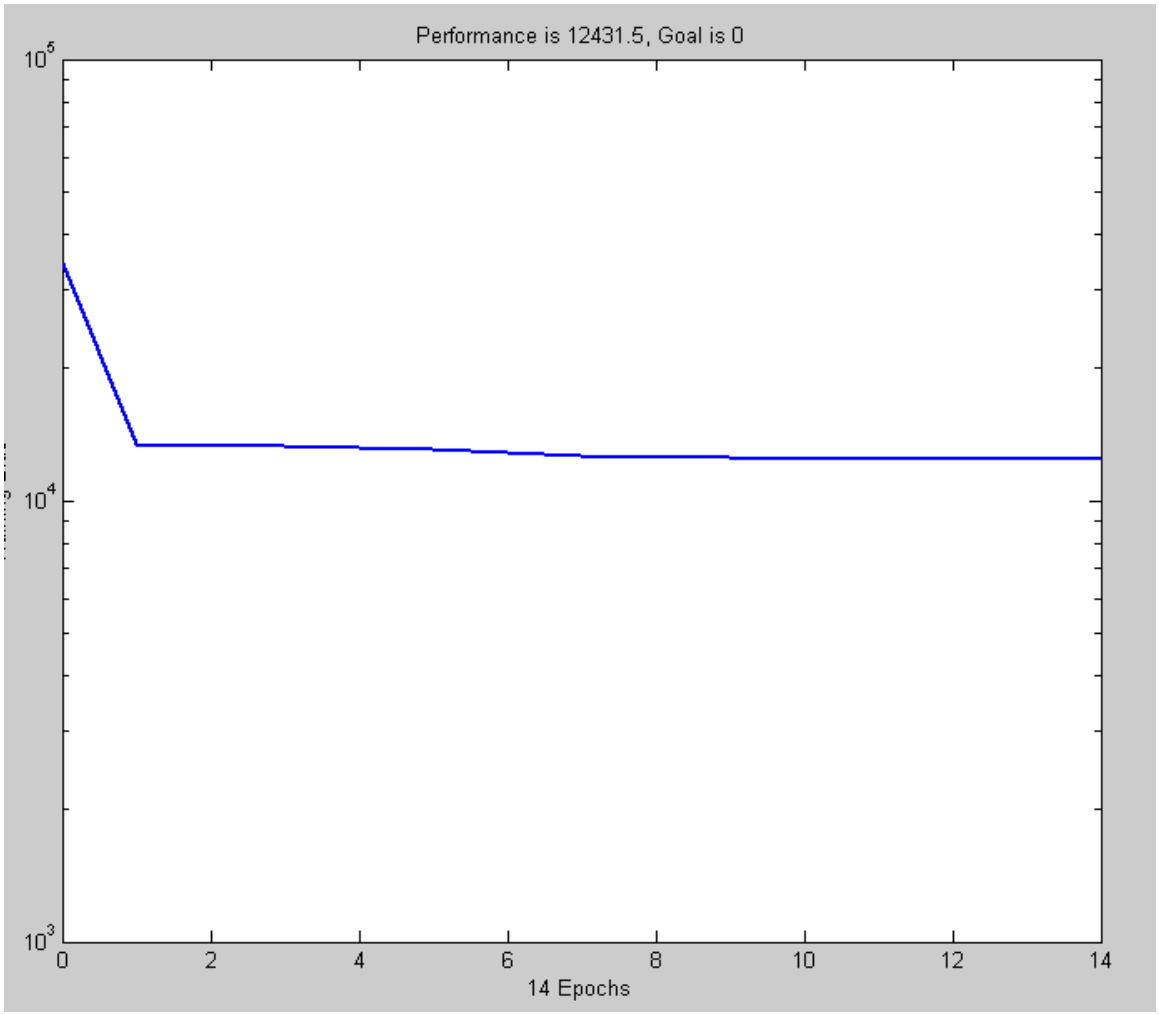
Elaboración: Andrés Guillermo Abad Robalino

Con la red definida ya somos capaces de entrenar esta red para poder ajustar la red a los datos de la red. Al hacer el entrenamiento, obtenemos la función de ajuste, la cual nos muestra en cuantas iteraciones se

alcanzó el ajuste predefinido. En la figura 5.13 presentamos la función de ajuste para esta red neuronal.

Figura 5.13

Función de entrenamiento de la red utilizada para la serie 3



Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaboración: Andrés Guillermo Abad Robalino

En la gráfica vemos que la función converge rápidamente pero alcanza sus criterios de salida en la iteración 14. En la tabla IX presentamos los valores de los coeficientes que minimizan el error total de la predicción.

Tabla IX
Pesos ya ajustados de la red utilizada en la serie 3

CAPA 1					
	W1	W2	W3	W4	B
Neurona 1	0.093621	-0.257	0.11018	0.34493	1.9482
Neurona 2	0.56321	-0.16422	-0.70451	0.30052	1.8188
Neurona 3	-0.20769	0.57498	0.5604	-0.29773	1.6113
Neurona 4	-0.43696	-0.30018	-0.15603	-0.22943	1.3794
CAPA 2					
	W1	W2	W3	W4	B
Neurona 1	1.5567	1.2168	1.5434	1.0507	2.882
Neurona 2	-0.55696	0.18789	-0.62195	0.11095	-0.5766
Neurona 3	-0.25702	-0.33093	0.38972	0.27989	0.48318
Neurona 4	0.01316	1.0657	0.62981	0.99207	0.29774
CAPA 3					
	W1	W2	W3	W4	B
Neurona 1	3.9054	-0.75404	0.36712	1.2012	1.8103

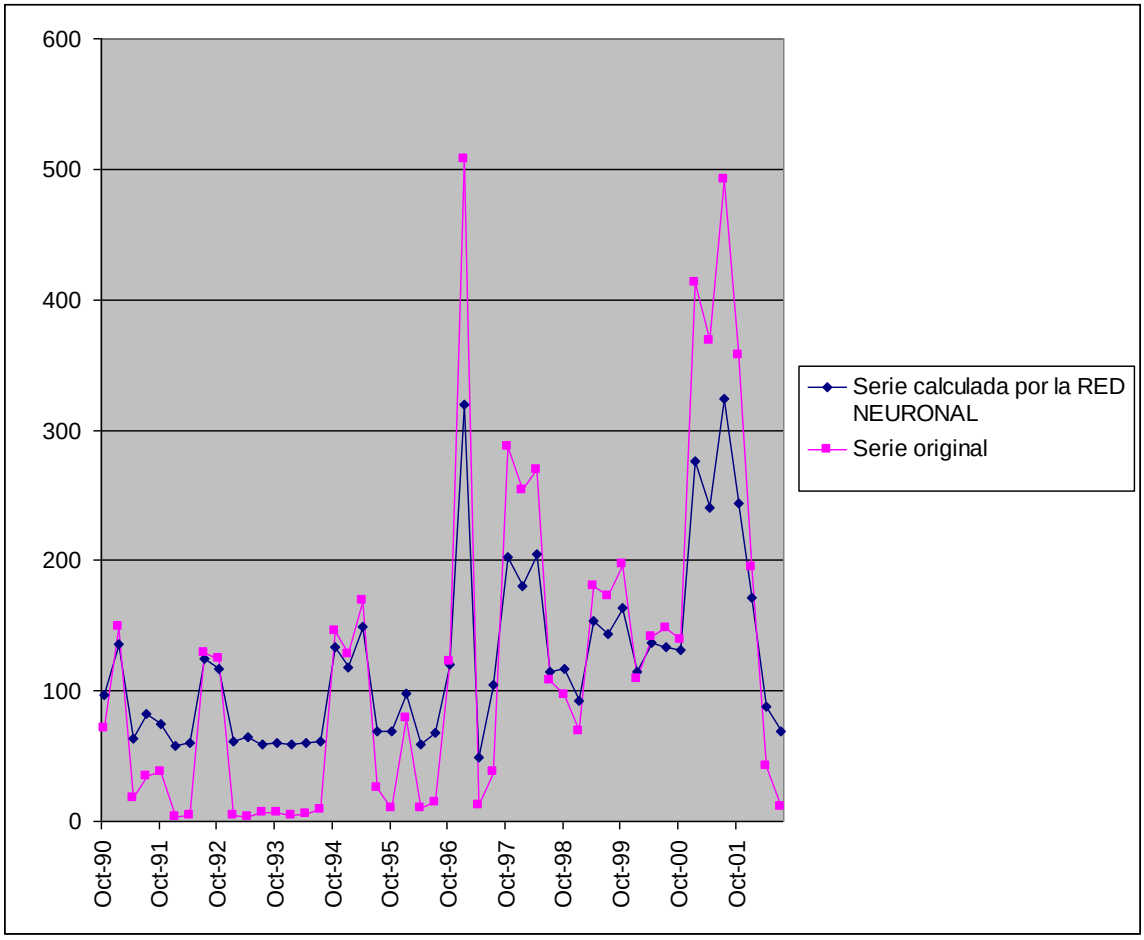
Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaboración: Andrés Guillermo Abad Robalino

Con los parámetros ajustados ya podemos realizar el cálculo de los valores de la serie a través de la serie de tiempo. En la figura 5.14 presentaremos estos resultados. Podemos ver las dos series, la original y la serie calculada por la red neuronal.

Figura 5.14

Gráfica de la serie 3 junto con el ajuste conseguido por la red neuronal



Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaboración: Andrés Guillermo Abad Robalino

En la gráfica podemos ver que aunque la serie es algo complicada de ajustar, la red neuronal hace un trabajo bastante aceptable. El cálculo del error cuadrático medio fue SSE= 189897.73

5.4.2. Resumen

Los resultados del análisis y ajuste de las tres series con los métodos se presentan resumidos en la tabla X a continuación.

Tabla X

Resultados comparativos entre los diferentes métodos de predicción

	SSE aplicando ARIMA	**SSE aplicando Red Neuronal**	**Elección**
Serie 1	11679.56881	12324,44	ARIMA
Serie 2	101357.7382	159901.82	ARIMA
Serie 3	293608.61	189897.73	Red Neuronal

Fuente: Pronóstico de Ventas: Comparación de la precisión de la predicción con diferentes métodos, TESIS 2005

Elaboración: Andrés Guillermo Abad Robalino

En la serie 1 el modelo preferido fue el modelo ARIMA, ya que tiene una SSE cerca de un 10% menor. En la segunda serie vemos que también se prefirió al modelo ARIMA, pero ahora el porcentaje fue de casi un 50%. Finalmente en la serie tres se prefiere al método de las redes neuronales.

Estos resultados constituyen el principal objetivo de la presente tesis, ya que nos muestran los índices respectivos para los diferentes métodos, realizando así la comparación de la precisión de la predicción utilizando series de tiempo correspondientes a ventas.

CONCLUSIONES

1. Las Redes neuronales representan una excelente alternativa como método de predicción para series temporales.

2. Una de las principales ventajes que encontramos de las redes neuronales sobre los métodos ARIMA, está la no necesidad de supuestos estadísticos, principalmente en cuanto a asignaciones de distribuciones de probabilidad a la serie de tiempo.

3. Las redes neuronales presentan un mayor poder de predicción en series de tiempo que no presentan una marcada estacionalidad ni tendencia que los modelos ARIMA convencionales.

4. Al aplicar las redes neuronales el investigador no esta forzado a hacer un análisis a priori de la serie de tiempo y de sus implicaciones, esto se convierte finalmente en la capacidad de hacer predicciones con precisiones aceptables obviando hasta cierto punto la teoría de predicciones convencional.

5. Cuando el modelo presenta periodicidad los modelos ARIMA presentan un mayor poder de predicción que las redes neuronales, aunque esta diferencia no es demasiado notable.

6. Al definir la topología de la red, debemos tener cuidado en no sobreparametrizar la red; ya que, aún cuando el error de ajuste dentro del conjunto de datos de entrenamiento tiende a reducirse a cero, el poder de extrapolación de la red se reduce significativamente para datos fuera del conjunto de entrenamiento.

7. Si trabajamos con una serie de tiempo con algún periodo cíclico debemos absorber esta información en la red neuronal ajustando el

número de neuronas de entrada correspondientes a los periodos anteriores al valor a predecir.

8. Una vez definida una topología de red que se ajusta bien a la serie, el conjunto de datos de entrenamiento puede ser manipulado de diferentes maneras, obteniendo así diferentes resultados.

RECOMENDACIONES

1. Para los investigadores carentes de bases estadísticas sólidas recomendaría el aprendizaje de la aplicación de las redes neuronales al problema del pronóstico; ya que, aunque su poder de predicción es en general un poco más bajo, una vez conocida la metodología de la aplicación de las redes neuronales, su empleo es relativamente fácil y adaptable a diferentes series.

2. Sería de interés probar con otros reglas de aprendizaje diferentes a la de Back-propagation que fue la que se presentó aquí, ya que esto se refleja directamente en el ajuste de los pesos y por consiguiente en el ajuste final de la red a la serie.

3. La utilización de indicadores distintos al SSE, ya que éstos nos podrían entregar más información acerca de las ventajes y desventajas de las redes neuronales sobre los métodos convencionales.

4. Dado que existen un número infinito de arquitecturas de redes neuronales, sería interesante probar diferentes topologías a las redes presentadas, buscando un mejor ajuste que el que conseguimos.

5. El diseño de un código que ejecute una barrida sobre diferentes topologías de redes y que, según criterios predefinidos y guardando el principio de parsimonia, determine que topología consiguió un mejor ajuste a la serie.

6. El gráfico de todas las etapas de el proceso de predicción brinda muchas ventajas e información útil para el investigador.

BIBLIOGRAFÍA

1. Krose Ben , Van der Smagt Patrick, 1996. An introduction to Neural Networks, The University of Amsterdam.

2. Jenkins Gwilym M, Reinsel Gregory C. 1994, Time series Analysis: Forecasting and Control, Prentice Hall.

3. Veelenturf L.P.J, 1995, Analysis and Applications of Artificial Neural Networks, Prentice Hall.

4. Johnston Jack, Dinardo John, 1997, Econometric Methods, McGraw-Hill.

5. Pınar Ural Beyza, 2002, Seasonal Adjustment in Economic Time Series, The Central Bank of the Republic of Turkey.

6. Gómez Víctor, Maravall Agustín, 2002, Seasonal Adjustment Interface for Tramo/Seats and X-12-Arima, Statistical Office of the European Communities.

7. Mitchell Tom M. , Machine learning, 1997, Mc Graw-Hill

8. Lee Giles G., Lawrence Steve, Chung Tsoi Ah, 2001, Noisy Time Series Prediction using a Recurrent Neural Network and Grammatical Inference, Princeton Press.

9. Masters Timothy, 1994, Signal and image processing with neural networks, John Wiley and Sons.

10. *Demuth Howard, Beale Mark,* Neural Network Toolbox, 2000, MathWorks.

11. Kutsurelis Jason E., 1998, Forecasting financial markets using neural networks: an analysis of methods and accuracy, Monterey, California Naval school press.

12. Guillén Varela Fabricio, Salguero López Bernardo, 2003, Introducción a las Redes Neuronales en la Minería De Datos, Universidad de Costa Rica.

13. 2004, http://ingenieria.udea.edu.co/investigacion/mecatronica/mectronics/redes.htm

14. 2004, www.gc.ssr.upm.es/inves/neural/ann2/anntutorial.html

15. 2005, ohm.utp.edu.co/neuronales/main.htm

16. 2005, www.neural-forecasting.com/

17. 2005, citeseer.ist.psu.edu/atiya99comparison.html

18. 2004, www.nd.com/appcornr/papers/finpapr5.htm