

# Problemas de investigación en la predicción de series de tiempo con redes neuronales artificiales

## Research problems on time series prediction using artificial neural networks

Paola Sánchez, Ph. D (c), & Juan D. Velásquez, Ph. D.

Universidad Nacional de Colombia - Sede Medellín

pasanche@unal.edu.co, jdvelasq@unal.edu.co

Recibido para revisión 27 de mayo de 2010, aceptado 28 de octubre de 2010, versión final 2 de diciembre de 2010

**Resumen**— En el modelado de series de tiempo, las redes neuronales han venido ganando cada vez más terreno, debido a sus reconocidas capacidades de adaptabilidad, generalización y aprendizaje. Si bien, en la literatura se denota un creciente interés por el desarrollo de aplicaciones con dichos modelos y se han presentado muchos reportes exitosos de su desempeño, igualmente, se han reportado resultados inconsistentes acerca de su uso. Algunos autores sostienen que tales inconsistencias son producto de falencias en la implementación del modelo y en la carencia de una metodología válida. En este trabajo se evalúan los efectos de dos factores clave en la construcción del modelo de red neuronal: el algoritmo de entrenamiento y el criterio de selección del número de neuronas ocultas; para exemplificar la discusión se usó la serie de pasajeros en líneas aéreas de Box & Jenkins y los métodos de entrenamiento de regla delta generalizada y RProp. La evidencia experimental demuestra que los métodos de entrenamiento evaluados exhiben comportamientos diferentes a los teóricamente esperados.

**Palabras Clave**— Redes Neuronales, Series de Tiempo, Algoritmo de Entrenamiento, Selección de Neuronas Ocultas.

**Abstract**— In time series modeling, neural networks have been gaining more ground due to his proven ability of adaptation, generalization and learning. While, the literature indicates a growing interest in developing applications with these models and have been submitted multiple reports of successful performance, also have reported inconsistent results regarding its use. Some authors argue that such inconsistencies are the result of shortcomings in the implementation of the model and lack of a reliable methodology. This study evaluates the effects of two key factors in building the neural network model: the training algorithm and the selection criteria the number of hidden neurons; for illustrating the discussion, we used the Box & Jenkins' number of passenger in an airlines and generalized delta rule and RProp

as training methods. Experimental evidence shows that the evaluated training methods exhibit different behaviors than theoretically expected.

**Keywords**— Artificial Neural Networks, Time Series, Training Algorithm, Selection Of Hidden Neurons.

### I. INTRODUCCION

El modelado de series de tiempo ha recibido en las últimas décadas mucha atención, debido a la creciente necesidad de contar con herramientas que faciliten la toma de decisiones y permitan superar las limitaciones teóricas, conceptuales y prácticas que presentan muchas de las técnicas disponibles en la actualidad. Esta motivación ha conducido al surgimiento de una amplia gama de modelos, donde, las redes neuronales, vistas como una técnica de regresión no paramétricas no lineal, han demostrado un amplio potencial debido a sus capacidades de adaptabilidad, generalización, aprendizaje y la posibilidad de representar relaciones no lineales [38]. Algunos aspectos que justifican y favorecen el desarrollo de modelos de redes neuronales son:

El proceso generador de los datos en series reales es, a menudo, desconocido y de difícil identificación, lo cual limita la capacidad de modelos de orden paramétrico de representar adecuadamente el comportamiento de la serie. Las redes neuronales son modelos auto-adaptativos, que no requieren supuestos a priori acerca del problema en estudio, característica altamente deseable en casos donde el mecanismo generador de los datos es desconocido e inestable.

Las series reales evidencian comportamientos inestables. La capacidad de aprendizaje y generalización de la red neuronal le

permite aprender comportamientos complejos directamente desde los datos y, a partir del conocimiento adquirido, inferir correctamente la evolución de la serie en porciones no evaluadas de los datos.

Las relaciones entre la serie de tiempo y sus variables explicativas son desconocidas. El carácter de aproximador universal que poseen las redes neuronales, les permite identificar dependencias ocultas de orden no lineal con base en una muestra finita.

Las series reales pueden exhibir patrones regulares como tendencias, ciclos y estacionalidades; e irregulares como cambios estructurales, datos atípicos, efectos de días calendario, etc. La flexibilidad en la configuración de la red facilitan la representación y extrapolación de comportamientos irregulares, toda vez que permite la construcción de muchos modelos candidatos con diferente número de variables de entrada, neuronas en la capa oculta, función de activación, etc.

Las unidades de los datos pueden ser muy grandes o muy pequeñas. La red neuronal es flexible en los valores que recibe y entrega; sin embargo, en datos muy grandes se prefiere un escalado para facilitar los cálculos y obtener resultados acordes.

Las series de tiempo proviene de múltiples campos del conocimiento y los datos pueden ser medidos en diferentes frecuencias de tiempo. La flexibilidad en la forma funcional que puede adquirir y el carácter de aproximador universal, les permite, a las redes neuronales, representar comportamientos complejos, sin importar el campo del conocimiento de donde provengan los datos, ni la frecuencia de tiempo.

En la literatura se denota un creciente interés por el desarrollo de aplicaciones de redes neuronales en series de tiempo [1], [8], [16], [38]. Si bien, muchos reportes exitosos de su desempeño han sido presentados, igualmente, tal como lo enuncian Zhang et al. [38], a menudo se reportan resultados inconsistentes acerca de su uso. Algunos autores sostienen que el origen de tales inconsistencias puede ser debido a diferentes causas: implementación incorrecta del modelo de red neuronal, criterios de selección del mejor modelo no comparables, configuración inadecuada de la red para las características de los datos, y carencia de evaluaciones confiables y válidas, entre otros. Muchas conclusiones son obtenidas a partir de estudios empíricos, presentando resultados limitados que a menudo no pueden ser extendidos a aplicaciones generales y que no son replicables.

Igualmente, el elevado número de factores incluidos en la configuración de la red, el proceso de entrenamiento, y la muestra de datos que se requiere determinar para lograr un modelo de red con un ajuste adecuado, convierte a las redes neuronales en una técnica inestable, dado que todo cambio en el entrenamiento o en algún parámetro produce grandes cambios en el ajuste [35]. Dos factores determinantes en el proceso de construcción del modelo de red neuronal lo constituyen el

algoritmo de entrenamiento u optimización y la selección del número adecuado de neuronas en la capa oculta, los cuales influencian el desempeño de la red tanto en ajuste como en tiempo de convergencia y robustez.

Partiendo de la premisa que una red neuronal tipo perceptrón multicapa con una capa oculta puede aproximar cualquier función continua con la precisión deseada, si se tiene un número suficiente de neuronas ocultas [17], y que al adicionar más neuronas ocultas, la red se ajusta mejor a los datos, es posible concluir que en este tipo de arquitectura de red, un proceso de adición de neuronas ocultas debe conducir a reducciones sistemáticas en la superficie del error hasta el punto que el error llegue a cero. No obstante, la experiencia práctica de los autores, así como la copiosa literatura sobre este aspecto, señala que la conclusión presentada es válida en lo teórico pero muy difícil de lograr en la práctica.

El objetivo de este trabajo es evaluar el desempeño de los métodos de entrenamiento regla delta generalizada y Rprop, y el nivel de cumplimiento del supuesto de reducción del error en un proceso de adición de neuronas ocultas. Las contribuciones de este trabajo se focalizan en los siguientes aspectos:

1. Se contribuye al campo de la inteligencia computacional, toda vez que se evalúan aspectos teóricos de la construcción de modelos de redes neuronales.
2. Se contribuye al campo del modelado de series de tiempo con redes neuronales, al presentar criterios que deben ser considerados a la hora de construir la representación y que son obviados en su implementación.

Lo restante de este trabajo está organizado como sigue. En la Sección 2 se describe el modelo general de series de tiempo con redes neuronales tipo perceptrón multicapa, y se presentan los pasos para su especificación y construcción. En la Sección 3 se discute las dificultades subyacentes al modelado de series de tiempo con redes neuronales, enfatizando en el problema de selección del algoritmo de entrenamiento y del número adecuado de neuronas en la capa oculta, mediante la evidencia práctica con los algoritmos de Regla Delta Generalizada y RProp. Finalmente, en la Sesión 4 se presentan las conclusiones y se resaltan problemas puntuales extraídos de los experimentos realizados.

## II. MODELADO DE SERIES DE TIEMPO CON REDES NEURONALES ARTIFICIALES

En general, el modelado de series tiempo con redes neuronales ha sido en su arquitectura abordado desde representaciones de alimentación hacia adelante y recurrentes, tales como: perceptrones multicapa, redes con función de base radial [37], [33], redes cascada-correlación [11], modelos neuro-difusos [19], [27], arquitecturas dinámicas [14], redes con dinámicas recurrentes y retrasos en el tiempo [13], [25], [23], entre otros.

La arquitectura de red neuronal más empleada en series de tiempo es del tipo perceptrón multicapa, la cual consta de: una capa de entrada en donde existe una neurona por cada variable explicativa  $x_t$ , que puede corresponder a un rezago de la serie de tiempo; de una o más capas ocultas donde se realiza el procesamiento de la información; y una capa de salida donde la red entrega los valores ajustados de la variable dependiente. La salida de la capa oculta se produce al aplicar a la entrada una función de transformación, conocida como función de activación. La configuración final obtenida se conoce como topología o arquitectura de la red neuronal.

Un perceptrón multicapa con una capa oculta y transferencia lineal en la activación de su capa de salida, puede ser definido matemáticamente como un modelo  $Y_t$ , que representa las relaciones existentes entre el conjunto de variables  $x_t$  y la serie de tiempo  $y_t$ , visibles en el instante de tiempo  $t$ , tal que:

$$Y_t = \alpha_0 + \sum_{j=1}^q \alpha_j g\left[\beta_{0j} + \sum_{i=1}^p \beta_{ij} x_t^{(i)}\right] + \varepsilon_t \quad (1)$$

donde  $\alpha_j$  representan los pesos de la capa oculta a la capa de salida; los coeficientes  $\beta_{ij}$  son los pesos de la capa de entrada a la capa oculta;  $p$  es el número de neuronas en la capa de entrada y  $q$  el número de neuronas en la capa oculta;  $\varepsilon_t$  son los errores aleatorios del modelo los cuales se asume que son independientes e idénticamente distribuidos con media cero y varianza constante  $\sigma_\varepsilon^2$ ; y  $g(u)$ , representa la función de activación o transferencia, la cual corresponde a una fórmula matemática que determina la salida de la capa oculta. Esta puede asumir diferentes formas, la más común es la función logística:

$$g(u) = \frac{1}{1 + \exp(-u)} \quad (2)$$

La función transferencia en la salida de la red neuronal es lineal.

Si  $x_t$  está conformado únicamente por los valores rezagados de  $y_t$ , entonces la entrada a cada neurona de la capa oculta es la parte determinísticas de un modelo autoregresivo, y en consecuencia, puede considerarse que la red neuronal es la suma ponderada de tantos modelos autoregresivos como neuronas posea en la capa oculta. Según la ecuación (1) y usando una función de activación logística, resulta definida en función de sus valores rezagados y de los pesos de la red, en una forma funcional de orden no lineal, por lo que éste modelo en el

contexto de series de tiempo es equivalente a una combinación no lineal de modelos autoregresivos [18], [26].

En el modelado de series de tiempo, Kaastra y Boyd [18] concuerdan en afirmar que la construcción de un modelo de red neuronal, tipo perceptrón multicapa, que tenga un desempeño adecuado, está basada en una secuencia de pasos críticos (Tabla 1), que para su aplicación requiere de juicio experto en muchos de los pasos.

Una crítica generalizada al procedimiento presentado en la Tabla 1 es el alto número de parámetros que deben ser seleccionados experimentalmente para generar la salida deseada, lo que convierte a las redes neuronales en un método inestable, dado que todo cambio en el entrenamiento o en algún parámetro produce grandes cambios en los resultados del modelo [35].

### III. DIFICULTADES EN EL MODELADO DE SERIES DE TIEMPO CON REDES NEURONALES

Para la inteligencia artificial, así como para la investigación de operaciones, el modelado de series de tiempo con redes neuronales es visto como un problema de minimización del error, que consiste en ajustar los parámetros de la red neuronal en función de minimizar el error entre el valor real y la salida obtenida. Si bien, este criterio permite obtener modelos cuya salida es cada vez más próxima a la deseada, va en perjuicio de la parsimonia del modelo, toda vez que conduce a representaciones más complejas (gran cantidad de parámetros). Desde el punto de vista estadístico, un criterio basado únicamente en la reducción del error no es óptimo, y es necesario un desarrollo orientado a la formalización del modelo, el cual requiere el cumplimiento de ciertas propiedades que no son tenidas en cuenta siempre, tales como la estabilidad de los parámetros calculados, la coherencia entre la serie y el modelo, la consistencia con el conocimiento previo y la capacidad predictiva del modelo.

Muchas comparaciones han sido hechas entre las redes neuronales y los modelos económétricos y estadísticos, con el fin de medir el desempeño de ambas aproximaciones. Tal como lo enuncian Zhang et al. [38]:

“There are many inconsistent reports in the literature on the performance of ANNs. The main reason is that a large number of factors including network structure, training method, and sample data may affect the forecasting ability of the networks.”

Casos donde la red neuronal presenta un desempeño peor que los modelos estadísticos lineales u otros modelos, puede ser debido a que las series estudiadas no presentan gran perturbación, a que la red neuronal utilizada para comparar no fue adecuadamente entrenada, a que el criterio de selección del mejor modelo no es comparable, o a que la configuración usada

no es adecuada para las características de los datos.

Muchas conclusiones acerca del desempeño de las redes neuronales son obtenidas a partir de estudios empíricos, presentando así resultados limitados que, a menudo, no pueden ser extendidos a aplicaciones generales. Sin embargo, son pocas las investigaciones sistemáticas acerca del modelado de series de tiempo con redes neuronales, y los avances teóricos obtenidos [31], y es esta razón, quizás, la causa primaria de las inconsistencias reportadas en la literatura.

Muchas de las publicaciones optimistas que indican un desempeño superior de las redes neuronales están relacionadas con paradigmas novedosos o extensiones de métodos existentes, arquitecturas y algoritmos de entrenamiento, pero carecen de una evaluación confiable y válida de la evidencia empírica de su desempeño.

Pocas contribuciones se han hecho en el desarrollo sistemático de metodologías que permita representar series de tiempo con redes neuronales sobre condiciones específicas, limitando el proceso de modelado a técnicas ad-hoc, en lugar de acercamientos científicos que sigan una metodología y proceso de modelado replicable. Consecuencia de ello es que, a pesar de los hallazgos empíricos, los modelos de redes neuronales no son totalmente aceptados en muchas áreas.

**Tabla 1.** Procedimiento para la construcción de un modelo de redes neuronales para series de tiempo

Paso 1: Selección de Variables
Paso 2: Recolección de Datos
Paso 3: Preprocesamiento de Datos
Paso 4: Conjuntos de entrenamiento, validación y verificación
Paso 5: Selección de la configuración interna
Número de capas ocultas
Número de neuronas ocultas
Número de neuronas de salida
Paso 6: Criterio de Evaluación
Paso 7: Entrenamiento de la Red Neuronal
Paso 8: Uso del modelo

Fuente: [16]

La discusión anterior conduce a pensar que, si bien se han realizado avances en el campo, existen aún tópicos abiertos por investigar. La pregunta de sí, por qué y sobre cuáles condiciones son mejores los modelos de redes neuronales está aún vigente.

Ahora bien, con el diseño de una red neuronal artificial se pretende conseguir que para ciertas entradas, la red sea capaz de generar una salida deseada. Para ello, además de una topología de red adecuada (arquitectura), se requiere un proceso de aprendizaje o entrenamiento, que permita modificar los pesos y

el numero de neuronas hasta encontrar una configuración acorde con la relación medida por algún criterio, y de esta forma estimar los parámetros de la red [26], [38]. Bajo el procedimiento enunciado dos procesos críticos los constituyen la selección del algoritmo de entrenamiento u optimización y del número adecuado de neuronas en la capa oculta. A continuación se discuten estos dos tópicos.

#### A. Algoritmos de Entrenamiento

El entrenamiento de una red neuronal es un problema de optimización no lineal sin restricciones en el cual los pesos de la red son iterativamente modificados con el fin de minimizar el error entre la salida deseada y la obtenida, en la búsqueda de un mínimo global. Diversos métodos han sido propuestos en la literatura para el entrenamiento de la red neuronal, siendo las técnicas clásicas de gradiente descendiente también llamado Backpropagation o la Regla Delta Generalizada, el algoritmo más comúnmente usado [15].

Diversos autores han convergido en afirmar que el desempeño del algoritmo de entrenamiento se ve afectado por diversos factores, entre otros:

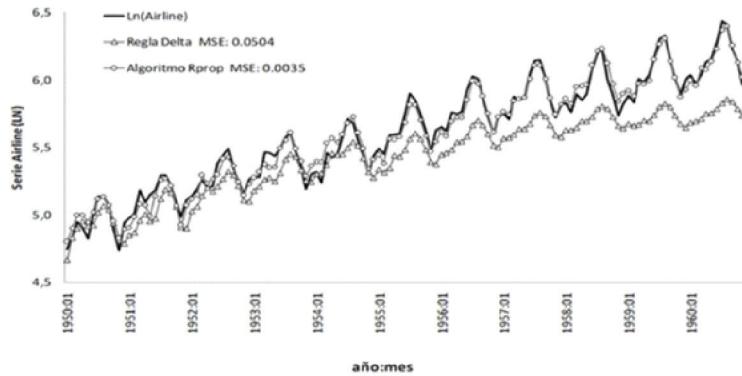
- Dependencia de valores iniciales: el conjunto de pesos iniciales de la red neuronal generalmente se selecciona de manera aleatoria. Sin embargo, pequeñas variaciones realizadas sobre las condiciones iniciales pueden llevar a grandes diferencias en el tiempo de convergencia del algoritmo y los resultados alcanzados, lo que influencia la habilidad de generalización de la red.
- Dificultad para converger a un óptimo global: la superficie del error, a menudo, es compleja y con múltiples óptimos locales, relacionada con el comportamiento no lineal, corriendo el riesgo de que el proceso de entrenamiento quede atrapado en estos.

En la búsqueda de soluciones a los dos problemas expuestos se han propuesto en la literatura desarrollos alternativos; en el primer caso, considerando valores iniciales predefinidos utilizando un esquema de inicialización selectivo, mediante la adición de valores preseleccionados en la ejecución inicial se logra que el error y tiempo de entrenamiento disminuya en gran medida, mediante el uso de un factor de regularización, o incluso calculados mediante algoritmos genéticos (para literatura relacionada véase [22] y [30]). Y en el segundo caso, mediante el desarrollo de procedimientos complejos basados, en algunas ocasiones, en la adición de parámetros que permitan una convergencia más rápida. Las mejoras más comunes aplicadas en la literatura las constituyen algoritmos de optimización dinámica adaptativa [24], [34], Quickprop [10], métodos de segundo orden como Levenberg-Marquardt [7], BFGS y la familia de métodos cuasi-Newton [9], [3]; GRG2 [20], [29]; y Rprop [28], entre otros.

Los resultados que pueden obtenerse con diversos algoritmos de entrenamiento pueden ser ampliamente diferentes; evidencia

de ello se presenta en la Figura 1 donde se muestra el número mensual de pasajeros en líneas aéreas -Airline- entre ENE1949 y DIC1960 [12] usando dos perceptrones multicapa que han sido entrenados por los métodos de la regla delta generalizada y el algoritmo Rprop. El método de la regla delta fue elegido por ser el algoritmo más comúnmente usado en las aplicaciones presentes en la literatura, mientras que el algoritmo RProp fue seleccionado por ser considerado uno de los algoritmos con más alto nivel de desempeño, velocidad de convergencia y robustez [2].

Tal como lo proponen Ghiassi et al. [14] y Faraway y Chatfield [12], para la serie transformada con logaritmo natural, se tomaron como variables de entrada los rezagos 1, 12 y 13, y 2 neuronas en la capa oculta. En la Tabla 2 se resumen las variables y parámetros de la red neuronal tomados para ambos algoritmos de entrenamiento. Igualmente, para establecer una comparación entre los dos algoritmos evaluados se calculó el ajuste mediante la función MSE para 50 iteraciones de cada experimento y se conservó el mejor valor en ambos casos.



**Figura 1.** Modelo de red neuronal de la serie de número de pasajeros en líneas aéreas usando Regla Delta y Algoritmo Rprop

Tal como puede observarse en la Figura 1, el error de ajuste es menor para el algoritmo Rprop, impactando la calidad del ajuste del modelo. Esto demuestra que la diferencia entre los errores de diferentes métodos de entrenamiento, bajo iguales condiciones de red, puede ser muy grande.

En la literatura se presentan a menudo estudios empíricos que permiten el contraste de diferentes métodos de entrenamiento, pero en muchas ocasiones, las comparaciones se hacen con el fin de destacar las ventajas de métodos particulares (el propuesto en el artículo, por ejemplo), y no se realiza una adecuada implementación de los demás algoritmos probados o las condiciones no son las mismas, presentando resultados más empíricos que teóricos que no permiten extraer reglas de comportamiento para una adecuada caracterización de ellos. Es así como a pesar de que se han desarrollado métodos eficientes para el entrenamiento de redes neuronales, no hay una aceptación generalizada de alguno estos, reflejo de la poca investigación sistemática tendiente a la comparación de las características de los métodos y a la carencia de un marco conceptual para la especificación del algoritmo de entrenamiento, más aún, se desconoce bajo qué condiciones estos métodos presentan resultados consistentemente mejores y estables.

Dado que la optimización de los pesos de cada neurona, depende del grado de error existente entre el valor deseado y el pronosticado por la red durante el entrenamiento, y está asociado a su vez a la configuración de la red neuronal, la

selección del número óptimo de neuronas ocultas está directamente relacionada con el proceso de entrenamiento usado. Es así como se han desarrollado métodos de regularización tales como métodos fijos, constructivos y destructivos, los cuales permiten la selección de las neuronas ocultas dentro del proceso de entrenamiento. Los métodos basados en algoritmos constructivos tienen ciertas ventajas sobre otros, puesto que permiten evaluar la conveniencia de adicionar o no una nueva neurona a la red, durante el entrenamiento, según ella disminuya el término del error, lo que los convierte en métodos más eficientes toda vez que permite explorar secuencialmente la superficie del error minimizando el riesgo de quedar atrapado en un óptimo local, aunque con alto costo computacional. Otros desarrollos como los algoritmos de poda (pruning algorithms) [7], [32], algoritmos bayesianos [21], [36], basados en algoritmos genéticos [4], redes neuronales con conjuntos rugosos [19], aprendizaje ensamblado [5], [34] y meta-aprendizaje [6], [35], han demostrado, igualmente, buenos resultados en la tarea de encontrar los parámetros óptimos de la red y el número de neuronas ocultas, pero, estos métodos son usualmente más complejos y son difíciles de implementar.

#### A. Selección del número de neuronas en la capa oculta

El proceso de selección del número adecuado de neuronas ocultas se ha convertido en un importante tópico de investigación en redes neuronales, toda vez que estas pueden conducir a resultados ampliamente diferentes. Un número

pequeño de neuronas en la capa oculta hace que la red no pueda aprender adecuadamente las relaciones existentes en los datos, mientras que un gran número hace que la red memorice los datos con una pobre generalización y una poca utilidad para la predicción. Algunos autores proponen que el número de neuronas ocultas debe ir en función del número de variables de entrada; sin embargo, este criterio esta a su vez relacionado con la extensión de la serie de tiempo y de los conjuntos de entrenamiento, validación y pronóstico.

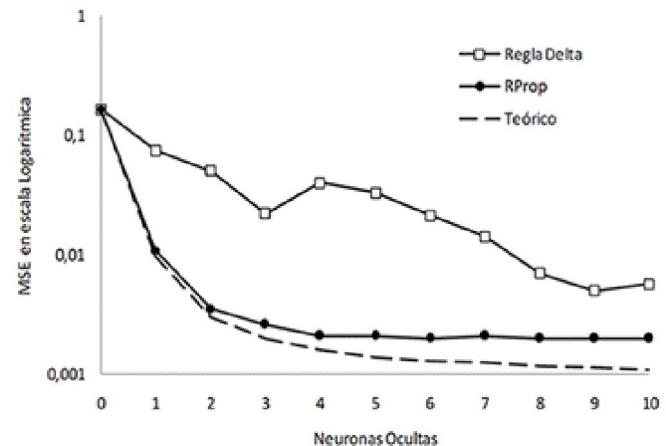
Teóricamente, un proceso constructivo de adicionar neuronas en la capa oculta debe conducir a reducciones sistemáticas en la superficie del error hasta el punto que el error llegué a cero, esto bajo la premisa que al adicionar más términos la red va a memorizar mejor el comportamiento de los datos y por ende va tener un mejor ajuste (véase planteamiento de las redes neuronales como aproximadores universales de funciones [17]); sin embargo, en la práctica, dicho comportamiento no se cumple. Con el fin de evidenciar la afirmación anterior se desarollo un procedimiento constructivo para la serie de Airline usando, nuevamente, como métodos de entrenamiento los algoritmos de Regla Delta Generalizada y RProp, bajo los mismos parámetros enunciados en la Tabla 2, pero adicionando neuronas secuencialmente desde cero (caso lineal), hasta 10. Fueron realizadas 50 iteraciones de cada neurona y cada método de entrenamiento, conservando para el mejor valor para cada caso.

La Figura 2 presenta los resultados del procedimiento constructivo (para resaltar el comportamiento la Figura es presentada en escala logarítmica). Tal como se observa, el algoritmo de Regla Delta, el cual es ampliamente utilizado en la literatura, no es la mejor opción de entrenamiento, toda vez que conduce a resultados inferiores en todos los casos. Si bien la adición de neuronas ocultas permite la mejora en el ajuste, el comportamiento teórico secuencialmente decreciente no es evidente en éste método, toda vez que se manifiesta un aumento del error en la cuarta y decima neurona, no permitiendo alcanzar en la quinta neurona el valor obtenido en la tercera; mientras que el algoritmo de RProp, si bien exhibe un comportamiento

**Tabla 2.** Variables y parámetros usados para la construcción del perceptrón multicapa

Variables y Parámetros	Valores
Serie modelada	Ln Airline
Rezagos	1,12, 13
Neuronas ocultas	2
Función de transferencia	Logística
Pesos iniciales	Aleatorios
Función de Error	MSE

**Fuente:** Construcción propia



**Figura 2.** Procedimiento Constructivo de Adición de Neuronas en la Capa Oculta para Regla Delta y Algoritmo Rprop

#### IV. CONCLUSIONES

En la literatura se evidencia un creciente interés por la realización de modelos de redes neuronales que conduzcan a resultados cada vez más acertados en el ajuste, favorecido por sus características de adaptabilidad, no linealidad y habilidad para aprender comportamientos desconocidos. Si bien se reconoce que hay mucha experiencia ganada en estos modelos, existen muchos factores que pueden afectar su desempeño, y los estudios empíricos a menudo encontrados no son suficientes para dirigir una solución al problema, donde se limita el proceso de modelado a técnicas ad-hoc, en lugar de acercamientos científicos que sigan una metodología y proceso de modelado replicable.

Los dos casos descritos en este trabajo, selección del algoritmo de entrenamiento y del número óptimo de neuronas ocultas, proveen una visión limitada pero útil de las falencias que poseen los modelos de redes neuronales, donde la evaluación de dos de las alternativas de entrenamiento presentes en las literatura condujeron a resultados en contravía de lo que teóricamente debería ser, demostrando así, que siguen siendo problemas abiertos de investigación.

Basados en los experimentos expuestos es posible extraer los siguientes problemas puntuales:

- No existe una vía sistemática aceptada en la literatura para determinar el conjunto adecuado de neuronas ocultas de la red neuronal.
- No hay claridad acerca de procedimientos orientados a la selección de neuronas en la capa oculta que a su vez permitan minimizar el tiempo de entrenamiento de la red.
- No existen razones empíricas, metodológicas o teóricas para preferir un algoritmo de entrenamiento específico entre varias alternativas.

## REFERENCIAS

- [1] Adya, M. y Collopy, F. 1998. How effective are neural networks at forecasting and prediction? A review and evaluation. En: *Journal of Forecasting*, Vol. 17, pp. 481-495.
- [2] Anastasiadis, A.D., Magoulas, G.D., y Vrahatis, M.N. 2003. An Efficient Improvement of the Rprop Algorithm. En: *Proceedings of the First International Workshop on Artificial Neural Networks in Pattern Recognition (IAPR 2003)*, University of Florence, Italy. pp. 197-201.
- [3] Battiti, R. 1992. First and second order methods for learning: Between steepest descent and Newton's method. En: *Neural Computation*, Vol. 4, No. 2, pp. 141-166.
- [4] Branke, J. 1995. Evolutionary Algorithms for Neural Network Design and Training. En: *Proceedings of the First Nordic Workshop on Genetic Algorithms and its Applications*, Vol. 95, No.1, pp. 145-163.
- [5] Carney, J. y Cunningham, P. 2000. Tuning diversity in bagged ensembles. En: *International Journal of Neural Systems*, Vol. 10, pp. 267-280.
- [6] Chan, P. y Stolfo, S. 1993. Metalearning for multistrategy and parallel learning. En: *Proceedings of the Second International Workshop on Multistrategy Learning*, pp. 150-165.
- [7] Cottrell, M., Girard, B., Girard, Y., Mangeas, M., y Muller, C. 1995. Neural modeling for time series: a statistical stepwise method for weight elimination. En: *IEEE Transactions on Neural Networks*, Vol. 6, No. 6, pp. 1355-1364.
- [8] Crone, S. y Kourentzes, N. 2009. Input-variable Specification for Neural Networks - An Analysis of Forecasting low and high Time Series Frequency. En: *Proceedings of the International Joint Conference on Neural Networks, IJCNN'09*, in press.
- [9] Dennis, J.E. y Schnabel, R.B. 1983. Numerical Methods for Unconstrained Optimization and Nonlinear Equations. NJ, Englewood Cliffs: Prentice-Hall.
- [10] Faltings, S. 1989. Faster-learning variations of back-propagation: An empirical study. En: *Proceedings of the 1988 Connectionist Models Summer School*.
- [11] Fahlman, S. y Lebiere, C. 1991. The Cascade-Correlation Learning Architecture. En: *Advances in Neural Information Processing Systems 2*, pp. 524—532.
- [12] Faraway, J. y Chatfield, C. 1998. Time series forecasting with neural networks: a comparative study using the airline data. En: *Appl. Statist.*, Vol. 47, pp. 231-250.
- [13] Gençay, R. y Liu, T. 1997. Nonlinear modelling and prediction with feedforward and recurrent networks. En: *Physica D: Nonlinear Phenomena*, Vol. 108, No. 1-2, pp. 119-134.
- [14] Ghiassi, M., Saidane, H., y Zimbra, D.K. 2005. A dynamic neural network model for forecasting time series events. En: *International Journal of Forecasting*, Vol. 21, pp. 341-362.
- [15] Hagan, M.T., Demuth, H.B., y Beale, M.H. 1996. Neural Network Design. Boston: PWS Publishing.
- [16] Hill, T., O'Connor, M., y Remus, W. 1996. Neural network models for time series forecasts. En: *Management Science*, Vol. 42, pp. 1082-1092.
- [17] Hornik, K., Stinchcombe, M., y White, H. 1989. Multilayer Feedforward Networks are Universal Approximators. En: *Neural Networks*, Vol. 2, No. 5, pp. 359-366.
- [18] Kaastra, L. y Boyd, M. 1996. Designing a neural network for forecasting financial and economic time series. En: *Neurocomputing*, Vol. 10, No. 3, pp. 215-236.
- [19] Kadogiannis, V. y Lolis, A. 2002. Forecasting financial time series using neural network and fuzzy system-based techniques. En: *Neural Computing and Application*, Vol. 11, pp. 90-102.
- [20] Lasdon, L.S. y Waren, A.D. 1986. GRG2 User's Guide. Austin.
- [21] Lauret, P., Fock, E., Randrianarivony, R.N., y Manicom-Ramsamy, J-F. 2008. Bayesian neural network approach to short time load forecasting. En: *Energy Conversion and Management*, Vol. 49, pp. 1156-1166.
- [22] Lehtokangas, M. 1999. Fast initialization for cascade-correlation learning. En: *IEEE Transactions on Neural Networks*, Vol. 10, No. 2, pp. 410-414.
- [23] Mishra, S. y Patra, S.K. 2009. Short term load forecasting using a novel recurrent neural network. En: *International Journal of Computational Intelligence: Theory and Practice*, Vol. 4, No. 1, pp. 39-45.
- [24] Pack, D.C., El-Sharkawi, M.A., Marks II, R.J., y Atlas, L.E. 1991. Electric load forecasting using an artificial neural network. En: *IEEE Transactions on Power Systems*, Vol. 6, No. 2, pp. 442-449.
- [25] Parlols, A.G., Rais, O.T., y Atiya, A.F. 2000. Multi-step-ahead prediction using dynamic recurrent neural networks. En: *Neural Networks*, Vol. 13, pp. 765-786.
- [26] Qi, M. y Zhang, P.G. 2001. An investigation of model selection criteria for neural network time series forecasting. En: *European Journal of Operational Research*, Vol. 132, pp. 666-680.
- [27] Rast, M. 1997. Forecasting Financial Time Series with Fuzzy Neural Networks. En: *IEEE International Conference on Intelligent Processing Systems*, pp. 432-434.
- [28] Riedmiller, M. 1994. Advanced supervised learning in multi-layer perceptrons—from backpropagation to adaptive learning algorithms. En: *Computer Standards and Interfaces*, Vol. 16, pp. 265-278.
- [29] Subramanian, V. y Hung, M. 1993. A GRG2-based system for training neural networks: design and computational experience. En: *ORSA J. Comput.*, Vol. 5, pp. 386-394.
- [30] Thimm, G. y Fiesler, E. 1995. Neural network initialization. En: *Natural to Artificial Neural Computation*, Vol. 930, pp. 535-542.
- [31] Trapletti, A. 2000. On neural networks as time series models. Universidad Técnica de Wien.
- [32] Weigend, A.S., Rumelhart, D.E., y Huberman, B.A. 1991. Generalization by weight-elimination with application to forecasting. En: *Advances in Neural Information Processing Systems*, Vol. 3, pp. 875-882.
- [33] Yan, X-B., Wang, Z., Yu, S-H., y Li, Y-J. 2005. Time Series Forecasting with RBF Neural Network. En: *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, pp. 4680-4683.
- [34] Yu, X.H., Chen, G.A., y Cheng, S.X. 1995. Dynamic learning rate optimization of the backpropagation algorithm. En: *IEEE Transactions on Neural Networks*, vol. 6, No. 3, pp. 669-677.
- [35] Yu, L., Wang, S., y Lai, K.K. 2009. A neural-network-based nonlinear metamodeling approach to financial time series forecasting. En: *Applied Soft Computing*, Vol. 9, pp. 563-574.
- [36] Zhang, T. y Fukushige, A. 2002. Forecasting time series by Bayesian neural networks. En: *Proceedings of the 2002 International Joint Conference on Neural Networks*, Vol.1, pp. 382-387.
- [37] Zhang, D., Han, Y., Ning, X., y Liu, A. 2008. Framework for Time Series Forecasts. En: *Proceedings ISECS International Colloquium on Computing, Communication, Control, and Management*, Vol. 1, pp. 52-56.
- [38] Zhang, P.G., Patuwo, B.E., y Hu, M.Y. 1998. Forecasting with artificial neural networks: the state of the art. En: *International Journal of Forecasting*, Vol. 14, No. 1, pp. 35-62.



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA  
SEDE MEDELLÍN  
FACULTAD DE MINAS

**120** años   
TRABAJO Y RECTITUD