Comparación de Redes Neuronales aplicadas a la predicción de Series de Tiempo

Comparison of Neural Network applied to prediction of Time Series

Darwin Mercado Polo[1], Luis Pedraza Caballero[2], Edinson Martínez Gómez[3]
[1]Master en lenguajes y sistemas informáticos, Docente Tiempo Completo, Universidad de la Costa - CUC, Ingeniería de Software y Redes, Barranquilla – Colombia.
[2]Magister en Ciencias de la computación (c), Docente Investigador, Universidad de la Costa - CUC, Educación, investigación e Innovación en Ciencias Básicas EDICBAS, Barranquilla – Colombia.
[3]Ingeniero de sistemas, Coordinador de desarrollo, Tecnosystems IT, Barranquilla – Colombia
dmercado@cuc.edu.co

Recibido 13/11/14
Aceptado 15/05/2015

Cite this article as: D. Polo, L. Pedraza, E. Martinez, "Comparison of Neural Network applied to prediction of times Series", *Prospect*, Vol 13, N° 2, 88-95, 2015.

RESUMEN

El presente estudio tiene como objetivo principal presentar la comparativa de las redes neuronales artificiales (RNA) tipo perceptrón multicapa (MLP) y de funciones de base radial (RBF) aplicadas a la predicción de series de tiempo. Se utilizó resilient backpropagation como algoritmo de aprendizaje para la red MLP y una combinación entre el algoritmo de los k-emanes y el método de la matriz pseudoinversa para la RBF. La implementación de las RNA se realizó utilizando un sistema basado en arquitectura cliente-servidor, previendo una futura integración con aplicaciones en tiempo real. Para la evaluación de las RNA se utilizaron conjuntos de datos de diferentes características y cantidad de datos.De acuerdo a los resultados obtenidos se concluye que para la utilización e integración de técnicas de inteligencia computacional en sistemas web, es preferible el uso de las RBF, debido a que obtiene mejores tiempos de ejecución. Es importante resaltar también que en calidad de respuesta los dos tipos de redes neuronales obtienen resultados similares.

Palabras clave: Redes neuronales artificiales; Predicción; Series de tiempo; Perceptrón multicapa; Funciones de base radial.

ABSTRACT

The main objective of the present work is to compare artificial neural networks (ANN) Multilayer Perceptron (MLP) and radial basis function (RBF) applied to time series prediction. The learning algorithm used was the resilient backpropagation for MLP network and a combination of the K-Means algorithm and pseudoinverse matrix method for the RBF. The implementation of the RNA was performed using a client-server-based system architecture, anticipating a future integration with real-time applications. For the evaluation of RNA, datasets with different characteristics and amount of data was used.According to the results, it can be concluded that the use and integration of computational intelligence techniques in web systems, it is preferable to use the RBF, because you get better runtimes. It is also important to note that response of both types of neural networks obtained similar results.

Keywords: Artificial Neural Networks; Forecasting; Time Series; Multilayer Perceptron; Radial Basis Functions.

Doi: http://dx.doi.org/10.15665/rp.v13i2.491

1. INTRODUCCIÓN

Una serie de tiempo es un conjunto de datos obtenidos a partir de la observación de un fenómeno determinado durante periodos de tiempos iguales, representando el cambio de una variable especifica de tipo económico, físico, químico, financiero, biológico, entre otros.

Cuando se trabaja con series temporales, una de las tareas más importantes es la de predecir los datos futuros de la serie, es decir, a partir de los datos del pasado proyectar los valores que tomará la variable determinada. Para llevar a cabo esta actividad se construye un modelo matemático que capture, total o parcialmente las características de esta, por ejemplo los modelos ARIMA. Estos, son modelos autorregresivos integrados de medias móviles y han demostrado gran utilidad en la predicción a corto plazo de series de alta frecuencia. En contraste a los modelos ARIMA y métodos estadísticos, las redes neuronales artificiales son consideradas más robustas, especialmente en la representación de relaciones complejas que exhiben comportamientos no lineales.

Las Redes Neuronales Artificiales (RNA) son sistemas de procesamiento de la información cuya estructura y funcionamiento están inspirados en las redes neuronales biológicas. En los últimos años se ha masificado el uso de las RNA para la predicción de series temporales gracias a su capacidad generalizadora por el hecho de aprender a partir de ejemplos, tal y como lo hace el cerebro humano.

En [1] se demuestra el uso de algoritmos combinados con un modelo hibrido de funciones de base radial para la modelación y pronóstico de series de tiempo no lineales, obteniendo resultados interesantes en cuanto a eficacia y eficiencia utilizando este tipo de red neuronal.

Un estudio teórico y experimental de diferentes tipos de RNA como el multilayer perceptron, Fir neural network y elman neural network es realizado en [2]. Destacan las ventajas y desventajas de cada uno de estos en cuanto al dominio del pronóstico y predicción de series de tiempo. De acuerdo a los resultados experimentales obtenidos es necesario realizar una buena configuración de la red para lograr una buena aproximación a los datos reales, dependiendo del tipo de RNA que se esté utilizando.

Por otra parte, Villa Garzón [3], en su tesis de maestría utiliza las redes cascada correlación para el modelado y predicción del precio de la electricidad en mercados de corto plazo liberalizados, tomando como punto de partida y de comparativa el multilayer perceptrón con algoritmo de aprendizaje resilient backpropagation.

Teniendo en cuenta que los sistemas que utilizan arquitectura cliente-servidor son desplegados en la web, lo que implica que son accesibles en cualquier lugar y momento, la importancia de este trabajo radica en la comparación e implementación de redes neuronales artificiales aplicadas a la predicción de series de tiempo utilizando este tipo de arquitecturas para su posterior uso en sistemas de tiempo real.

El presente proyecto pretende realizar una implementación web para redes neuronales artificiales aplicadas a la predicción de series de tiempo, utilizando herramientas de última generación que permiten construir interfaces de usuario muy elegantes y excelentes resultados en procesamiento de datos. Por otra parte, se desea que el usuario final no deba preocuparse por los detalles de los elementos internos de la implementación, sino ocuparse de la definición de la arquitectura de las RNA implementadas.

2. TEORÍA

2.1. Antecedentes

En las últimas décadas, la aplicación de las redes neuronales artificiales en la predicción de series de tiempo ha ido creciendo por las características ideales que ofrecen las RNA para trabajar con modelos no lineales. Así mismo, el desarrollo de aplicaciones que faciliten el trabajo a la hora de realizar las simulaciones con Redes Neuronales Artificiales continúa en aumento.

En [4] se destacan las redes neuronales artificiales como uno de los métodos de predicción para series temporales gracias a su gran capacidad de adaptación y capacidad de representación de procesos no lineales.

Saranli et al [5] presentan las redes tipo RBF para la predicción de series de tiempo caóticas utilizando como algoritmo de aprendizaje el algoritmo relocating-lms, resaltando la consecución de un error mínimo al comparar los datos reales de la serie de tiempo con los predichos por la red.

En [6] se presenta una nueva técnica para la predicción no lineal de series de tiempo a través de las redes neuronales de funciones de base radial con atribución de centros gaussianos de las funciones de base radial por descomposición de los espacios de datos en subespacios.

RBF son utilizadas en [7] para la predicción del caudal de los ríos, en este se destaca el proceso de aprendizaje al que es sometido la red neuronal y los resultados con las distintas configuraciones utilizadas, demostrando las variaciones que esto puede generar en las simulaciones de este tipo de problemas. Además se demuestra la eficacia de las RBF en la predicción de series de

tiempo. También desarrolla un sistema hibrido inteligente para la predicción y el pronóstico de caudales en el cual utiliza las redes neuronales como una de las herramientas inteligentes que ayudan al sistema para el desarrollo del objetivo.

Por otro lado, basados en la red tipo MLP [8] se propone una nueva versión no lineal del modelo airline; reemplazando la componente lineal de promedios móviles por un perceptrón multicapa con el objetivo de realizar pronósticos de series de tiempo con tendencia y ciclo estacional.

En [9] es utilizado un modelo que combina un modelo lineal autorregresivo (AR) con un perceptrón multicapa (MLP) con una única capa oculta, permitiéndole unir las ventajas de los modelos autorregresivos y de las redes neuronales, de tal forma que es más fácil capturar dinámicas complejas, tal como es el caso de los precios de electricidad. Esto para realizar el modelado del precio spot de la electricidad en Brasil.

En cuanto al desarrollo de aplicaciones que involucren redes neuronales artificiales y las series temporales, Velásquez et al [10], desarrollan ARNN: un paquete para la predicción de series de tiempo usando redes neuronales autorregresivas, basado en el lenguaje de programación R. implementando como funciones principales la creación y estimación del modelo de RNA y el pronóstico de la serie temporal.

Para el caso de plataformas web que trabajen con redes neuronales se destaca el desarrollo realizado por Watta, Hassoun, y Dannug de applets que permiten simular varios tipos de redes neuronales [11]. Además, en [12] desarrollan una herramienta web para la simulación de redes neuronales artificiales con objetivo educativo, resaltando características importantes de este tipo de proyectos como lo son: la accesibilidad con respecto a software y hardware y la facilidad en el despliegue de la plataforma, olvidándose de los obstáculos de la instalación. Hay que resaltar que esta plataforma muestra una interfaz de usuario poco elegante.

Es evidente la importancia de las series de tiempo en las ramas del conocimiento humano en donde las RNA juegan un papel central al permitir realizar pronósticos a las series temporales. El desarrollo de aplicaciones que ayuden a optimizar el proceso de simulación de estas toma mucha relevancia en especial cuando los proyectos desarrollados son implementados en internet.

2.2. Redes Neuronales Artificiales

Las Redes Neuronales Artificiales (RNA) o sistemas conexionistas son sistemas de procesamiento de la información cuya estructura y funcionamiento están inspirados en las redes neuronales biológicas. Consisten en un conjunto de elementos simples de procesamiento llamados nodos o neuronas conectadas entre sí por conexiones que tienen un valor numérico modificable llamado peso [13].

La actividad que una unidad de procesamiento o neurona artificial realiza en un sistema de este tipo es simple. Normalmente, consiste en sumar los valores de las entradas (inputs) que recibe de otras unidades conectadas a ella, comparar esta cantidad con el valor umbral y, si lo iguala o supera, enviar activación o salida (output) a las unidades a las que esté conectada. Tanto las entradas que la unidad recibe como las salidas que envía dependen a su vez del peso o fuerza de las conexiones por las cuales se realizan dichas operaciones [14].

Una neurona artificial se comporta como la neurona biológica pero de una forma muy simplificada (tabla 1).

Tabla 1. Analogía neurona biológica y neurona artificial [14].
Table 1. Biological neuron and artificial neuron analogy [14].

Neurona Biológica	Neurona Artificial
Soma	Neurona
Dendrita	Entrada
Axón	Salida
Sinapsis	Peso

La salida de la red neuronal viene dada de la siguiente manera:

$$O_j = f\left(\sum w_{ij} x_j - \theta_i\right) O_j = f\left(\sum w_{ij} x_j - \theta_i\right) \quad (1)$$

Donde f(.), es la función de activación de la neurona. Dentro de las más utilizadas se encuentran la función identidad, escalón, lineal a tramos, sigmoidea y sinusoidal.

El aprendizaje en una RNA es un proceso de ajuste o modificación de los valores o pesos de las conexiones, en el que se logra que las salidas del sistema sean lo más parecidas a las salidas deseadas proporcionadas por el usuario. Las redes neuronales artificiales pueden clasificarse de acuerdo con el tipo de aprendizaje que utilizan.

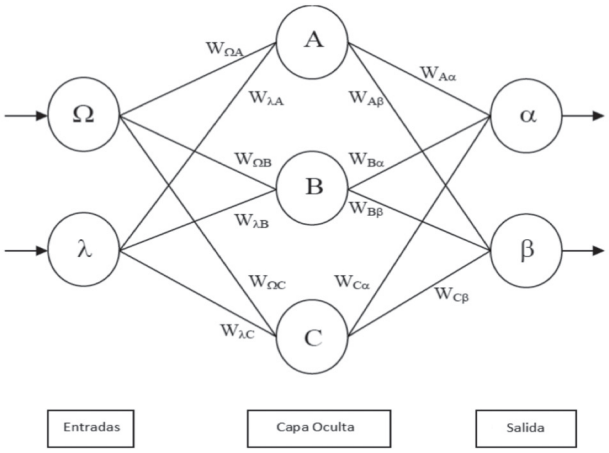
2.2.1.Perceptron Multicapa – MLP

El perceptrón multicapa (Multilayer Perceptron MLP) propuesto por Rumelhart es el componente principal de una red neuronal, proporciona la base para la ma-

yoría de las aplicaciones de las redes neuronales.

El MLP es una red formada por una capa de entrada, al menos una capa oculta y una capa de salida, tal como se muestra en la figura 1.

Figura 1. Red Neuronal Perceptrón Multicapa [3].
Figure 1. Multilayer Perceptron Neural Network [3].



Dentro de las características más importantes del perceptrón multicapa se encuentran las siguientes:

- Se trata de una estructura altamente no lineal.
- Presenta tolerancia a fallos.
- El sistema es capaz de establecer una relación entre dos conjuntos de datos.
- Existe la posibilidad de realizar una implementación hardware.

El algoritmo más popular de aprendizaje para el perceptrón multicapa es el backpropagation, el cual consiste en utilizar el error generado por la red y propagarlo hacia atrás, es decir, reproducirlo hacia las neuronas de las capas anteriores.

El algoritmo backpropagation para el MLP presenta ciertas desventajas, como son: lentitud de convergencia, precio a pagar por disponer de un método general de ajuste funcional, puede incurrir en sobre aprendizaje, fenómeno directamente relacionado con la capacidad de generalización de la red. Y no garantiza el mínimo global de la función de error, tan solo un mínimo local.

Ante las desventajas presentadas se han desarrollado variantes que buscan mitigar estos inconvenientes, tal es el caso del resilient backpropagation, considerado como uno de los algoritmos basados en gradiente más adecuados para entrenar redes neuronales artificiales.

El algoritmo resilient backpropagation es considerado como uno de los algoritmos más robustos para la estimación de los parámetros (o pesos) de una red neuronal. En este proceso se busca encontrar los valores de los parámetros la red neuronal tal que se minimice la diferencia entre los valores deseados y los valores calculados por la red.

El algoritmo RPROP, y sus variantes, difiere de la técnica clásica de propagación hacia atrás del error (o algoritmo backpropagation) en que las derivadas parciales de la función de error sólo son usadas para determinar el sentido en que deben ser corregidos los pesos de la red pero no las magnitudes de los ajustes. Los algoritmos basados en backpropagation modifican los valores de los parámetros proporcionalmente al gradiente de la función de error, de tal forma que en regiones donde el gradiente tiende a ser plano, el algoritmo avanza lentamente. RPROP tampoco se ve afectado por la saturación de las neuronas de la red neuronal, ya que solamente se usa la derivada para determinar la dirección en la actualización de pesos. Consecuentemente, converge más rápidamente que los algoritmos basados en backpropagation [3].

2.2.2. Redes Neuronales de Funciones de Base Radial – RBF

Una red de funciones de base radial es una red neuronal que utiliza funciones de base radial como funciones de activación. La arquitectura utilizada por las RBF es muy similar a la del perceptrón multicapa, con la característica de que las RBF utilizan siempre tres capas; una capa de entrada, una capa oculta y una de salida, mientas que los MLP pueden tener más.

En las RBF la capa oculta realiza una transformación no lineal del espacio de entrada. Las neuronas de esta capa son las funciones de base radial y cada neurona de la capa de salida es un combinador lineal [14].

De modo general, el valor generado por una red RBF con una única salida, viene definido por la ecuación:

$$y_i = \sum_{j=1}^{N} w_j f_j \left(\frac{\|x_j - u_j\|}{\sigma_j} \right) y_i = \sum_{j=1}^{N} w_j f_j \left(\frac{\|x_j - u_j\|}{\sigma_j} \right) \quad (2)$$

Las redes RBF son excelentes aproximadores universales y los parámetros que definen el proceso de aproximación son:

- Los pesos entre los centros y las neuronas del nivel de salida.
- La posición de los centros.
- Las funciones de Gauss de los centros.

Los pesos se ajustan utilizando el algoritmo de mínimos cuadrados ordinarios (LMS, Least Mean Square), pero se pueden utilizar también otros algoritmos como el método de la pseudo-inversa. Para ajustar los pesos utilizando el método de la pseudo-inversa, es necesario representar matricialmente las salidas de-

seadas como el producto de las salidas de los centros por el vector de pesos, como se muestra en la figura 2:

Figura 2. Representación matricial de las salidas deseadas en una red RBF [15].

Figure 2. Matrix representation of the desired outputs in a RBF network [15].

$$\begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} \exp\left[-\frac{\|x_1-u_1\|^2}{2\sigma_1^2}\right] & \cdots & \exp\left[-\frac{\|x_1-u_n\|^2}{2\sigma_n^2}\right] \\ \exp\left[-\frac{\|x_2-u_1\|^2}{2\sigma_1^2}\right] & \cdots & \exp\left[-\frac{\|x_2-u_n\|^2}{2\sigma_n^2}\right] \\ \vdots & \vdots & \vdots \\ \exp\left[-\frac{\|x_n-u_1\|^2}{2\sigma_1^2}\right] & \cdots & \exp\left[-\frac{\|x_n-u_n\|^2}{2\sigma_n^2}\right] \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$

De esta manera se obtiene la siguiente relación:

$$D = G * W \quad (3)$$

Donde D es el vector de salidas deseadas, W es el vector de pesos y G es la matriz de salida de los centros de la RBF.

De la expresión anterior se obtiene el ajuste de pesos aplicando la pseudo inversa de la matriz G:

$$W = G^{+} * D \quad (4)$$

Donde G^{+}es la pseudo-inversa de la matriz G:

$$G^{+} = (G^{T}G)^{-1}G^{T} \quad (5)$$

Los centros de la capa intermedia se pueden seleccionar utilizando diferentes métodos:

• **Selección aleatoria de centros:** Algunos vectores del espacio de entrada se pueden elegir como centros. Si es posible, deben ser seleccionados de tal forma que estén repartidos de manera regular por todo el espacio de entrada.

• **Autoselección de centros:** Una posibilidad es la utilización de métodos de entrenamiento que no requieran supervisión. Los centros se sitúan en aquellas zonas del espacio de entrada, donde haya un número significativo de datos. La regla de los k-vecinos más próximos se puede utilizar para ajustar el valor de los centros.

Los citados algoritmos no garantizan la convergencia de la red, ya que E no está linealmente relacionada con la ubicación de todos los centros y, por tanto, se pueden alcanzar mínimos locales [15].

2.3.Redes Neuronales Artificiales y Series de Tiempo

Una serie de tiempo viene determinado por un conjunto de observaciones que están ordenadas en el tiempo, representando el cambio de una variable ya sea de tipo económico, físico, químico, biológico, etc. a lo largo de un período determinado. Por lo general, el conjunto de observaciones disponibles se encuentra almacenado a intervalos de tiempo iguales [16].

El objetivo del análisis de las serie de tiempo es el conocimiento de su patrón de comportamiento, para así poder prever su evolución en el futuro cercano, suponiendo que las condiciones no variarán significativamente. Si bien el comportamiento de cualquier serie de tiempo puede observarse gráficamente, no en todos los casos es posible distinguir las particularidades que cada una puede presentar. Existen ciertos movimientos o variaciones características que pueden medirse y observarse por separado. Estos movimientos son llamados a menudo componentes de una serie de tiempo, y se asume que son causados por fenómenos distintos.

Matemáticamente las series de tiempo están representadas mediante la relación tiempo observación, descrita a través de un conjunto de datos numéricos, como se muestra a continuación [17]:

$$\{y(t_1), y(t_2), \ldots., y(t_n)\} = \{y(t): t \in T \subseteq R\} \quad (6)$$

El objetivo entonces es encontrar un modelo que permita predecir los valores futuros a través de los ya obtenidos, de la siguiente manera:

$$Y(t_{n+1}) = f(y(t_n), y(t_{n-1}), \ldots..) \quad (7)$$

Este modelo generalmente abstrae las características más importantes de la serie, de tal manera que se puedan obtener las predicciones de la serie para intervalos de tiempo determinados.

En muchas áreas del conocimiento las observaciones de interés son obtenidas en instantes sucesivos del tiempo, por ejemplo, a cada hora, durante 24 horas, mensuales, trimestrales, semestrales o bien registradas por algún equipo en forma continua. De aquí la importancia de este tópico.

A través del tiempo se han desarrollado un gran número de técnicas para la predicción y el modelado de series de tiempo, siendo las redes neuronales artificiales consideradas las más robustas y eficaces para esta tarea.

Para lograr predecir una serie de tiempo mediante una red neuronal artificial, esta debe configurarse de ma-

nera que las entradas de la red sean valores pasados al que se desea predecir.

El entrenamiento de la red consiste en ajustar los pesos para obtener los valores deseados de la serie en un instante de tiempo específico.

2.4. Arquitectura Cliente-Servidor

La arquitectura cliente-servidor(C/S) es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, que le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En la arquitectura C/S el remitente de una solicitud es conocido como cliente. Sus características son:

• Es quien inicia solicitudes o peticiones, tienen por tanto un papel activo en la comunicación.
• Espera y recibe las respuestas del servidor.
• Por lo general, puede conectarse a varios servidores a la vez.
• Normalmente interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.

Al receptor de la solicitud enviada por el cliente se conoce como servidor. Sus características son:

• Al iniciarse esperan a que lleguen las solicitudes de los clientes, desempeñan entonces un papel pasivo en la comunicación.
• Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
• Por lo general, aceptan conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado).
• No es frecuente que interactúen directamente con los usuarios finales.

Hoy en día el desarrollo de aplicaciones que utilizan la arquitectura cliente-servidor, generalmente trabajan implementando en este último el patrón de diseño MVC (Modelo, Vista, Controlador), ya que se ha convertido en una guía para el diseño de arquitecturas de aplicaciones y brinda una fuerte interactividad con los usuarios. La intención de implementar este patrón es separar los datos de la aplicación, la interfaz de usuario y la lógica de control.

3. METODOLOGÍA

Para lograr el objetivo del proyecto se plantea una metodología basada en fases que contempla las siguientes etapas:

Fase 1.Exploración y selección de los diferentes tipos de redes neuronales artificiales a ser implementadas.

En esta fase se realiza un levantamiento bibliográfico de los diferentes tipos de redes neuronales artificiales aplicadas al tratamiento de las series de tiempo, para lo cual se consulta con expertos, revistas científicas, artículos web, entre otras fuentes. Esto permitió seleccionar los tipos de red neuronal RBF y MLP.

Fase 2.Diseño de la plataforma

En esta fase se toma la ingeniería de software como referente para el establecimiento de diseños y diagramas que representen y abstraigan las características más importante de los tipos de redes neuronales artificiales seleccionados en la fase anterior.

Fase 3.Desarrollo e implementación de la plataforma.

En esta fase se implementa en Python los algoritmos de aprendizaje para las redes neuronales seleccionadas y se construye utilizando ExtJs las interfaces gráficas de usuario. Finalmente se realiza el despliegue en el servidor web apache.

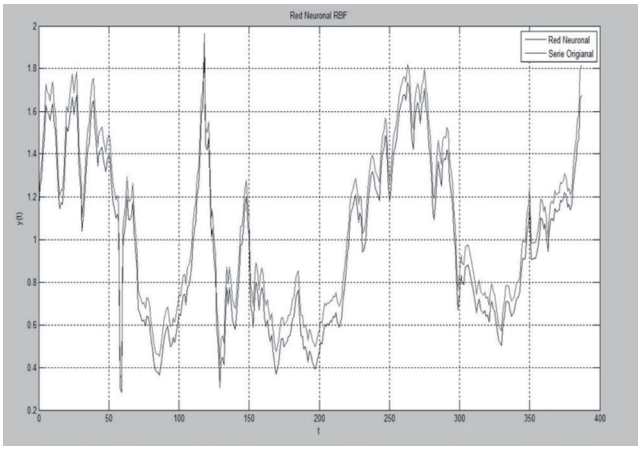
Fase 4.Test y comparación de resultados

Esta etapa se basa en la comparación de los resultados de los tipos de red neuronal implementados aplicando dos series de tiempo categorizadas en el área de finanzas. Se toma un conjunto de test y otro de entrenamiento para determinar la efectividad y tiempos de respuestas de las RNA. Se realiza la simulación en forma repetitiva y se toman los mejores resultados.

4. RESULTADOS

Se desarrolló una plataforma web utilizando el web server apache, el lenguaje de programación python para implementar los algoritmos de aprendizaje de las redes neuronales artificiales seleccionadas y ExtJs para la construcción de las interfaces gráficas de usuario. La plataforma permite importar los datos de entrenamiento y test de la red a partir de archivos delimitados por comas (CSV) y posterior a la ejecución de la simulación son desplegados los resultados obtenidos por la RNA (figura 3). También es posible realizar la representación del error obtenido y de los resultados de la red.

Figura 3. Salida deseada y la salida de la red en la fase de entrenamiento.
Figure 3. Desired output and real output of the network in the training phase.



Para evaluar las redes neuronales artificiales se utilizaron conjuntos de datos de diferentes características y cantidad de datos. Se realizaron 10 replicaciones para cada dataset variando las configuraciones de las redes neuronales, utilizando 4, 5, 6 y 7 neuronas en la capa oculta. Se seleccionó la mejor respuesta obtenida por las redes neuronales en cuanto a porcentaje de efectividad. Los resultados son mostrados en la tabla 2.

Tabla 2 Resultados de las redes neuronales para los diferentes datasets [18].
Table 2. Results of neural networks for different datasets [18].

Dataset (Id)	Porcentaje Efectividad (%)	
	MLP	RBF
1	65,4	66,5
2	66,5	70,1
3	74,5	72,3
4	75,4	77,2
5	66,5	68,5
6	74,7	75,5
7	76,8	75,2
8	75,8	76,4
9	75,6	77,9
10	78,5	76,4

De los valores obtenidos se puede observar que el porcentaje de efectividad de los dos tipos de redes neuronales es próximo. Con 95% de confianza no se puede establecer que en cuanto a calidad de respuesta un tipo de red neuronal es mejor que otro.

Otro aspecto importante a tener en cuenta es el tiempo de ejecución de cada red neuronal, en la tabla 3 son mostrados los tiempos de ejecución en segundos para los datasets testados:

Tabla 3. Tiempos de ejecución en segundos de las redes neuronales para los diferentes datasets [18].
Table 3. Execution times in seconds of neural networks for different datasets [18].

Dataset (Id)	Tiempo (segundos)	
	MLP	RBF
1	6,1	3,6
2	7,2	3,5
3	9,3	4,1
4	7,5	3,8
5	6,4	3,7
6	9,2	4,5
7	6,2	3,6
8	7,3	3,8
9	7,1	3,6
10	6,5	3,4

La tabla de tiempos de ejecución muestra claramente que la RBF presenta mejores tiempos de ejecución. Con 95% de confianza es posible afirmar que la RBF obtiene mejores resultados en comparación con la red MLP. Este aspecto es de mucha relevancia dado que se prevé la utilización de este tipo de técnicas de inteligencia computacional para integrarlos en sistemas Web.

5. CONCLUSIONES

En este artículo se presentó una comparación de dos tipos de redes neuronales artificiales aplicadas al dominio de la predicción de series de tiempo. Se utilizó un sistema con arquitectura cliente-servidor para evaluar el desempeño de las RNA, en cuanto a tiempo y calidad de respuesta. Se optó por utilizar este tipo de arquitectura previendo la integración con aplicaciones web 2.0, en donde las técnicas de inteligencia computacional abarcan un factor importante. Los resultados muestran claramente que:

• En cuanto a tiempo de ejecución la red neuronal tipo RBF obtiene las mejores respuestas, siendo este un factor relevante cuando se trabaja con sistemas web.

• En cuanto a calidad de respuesta, la RBF y MLP obtienen resultados muy próximos y las herramientas estadísticas utilizadas establecen con 95% de confianza que las RBF y MLP no son totalmente diferentes.

• En la integración de redes neuronales aplicadas a la predicción de series de tiempo es preferible el uso de las RBF en comparación con las MLP.
• El uso de estas redes neuronales sería un gran aporte

sobre la predicción en todos los proyectos que incluyen series de tiempo como: estimación y predicción de la moneda, predicción de las condiciones ambientales, toma de decisiones desde el punto financiero y predicción en el crecimiento poblacional.

REFERENCIAS

[1] M. Nasir, (1993, August) Time Series Modelling and Prediction Using Neural Networks. Masters thesis. Universiti Teknologi Mara.

[2] T. Koskela, M. Lehtokangas, J. Saarinen & K. Kaski. (1996, September) Time Series Prediction with Multilayer Perceptron, FIR and Elman Neural Networks. Presentado en World Congress on Neural Networks. [En Línea] Disponible: <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=8E47EA4845D879E4251A9706E58EB6C2?doi=10.1.1.35.1631&rep=rep1&type=pdf>.

[3] F. Villa, (2010, Octubre) Modelado y Predicción del Precio de la Electricidad en Mercados de Corto Plazo Liberalizados Usando Redes Cascada Correlación. Tesis de Maestría. Universidad Nacional de Colombia.

[4] V. Ribeiro, Goldschmidt R. & Choren R., "Métodos para Previsão de Séries Temporais e suas Tendências de Desenvolvimento", Monografías en Sistemas e Computação, 3, 1-26, 2009.

[5] A. Saranli, B. Baykal, (2010) Chaotic Time-Series Prediction and the Relocating-LMS (RLMS) Algorithm for Radial Basis Function Networks [Internet], Imperial College of Science. Disponible desde: <http://www.eurasip.org/Proceedings/Eusipco/1996/paper/pas_8.pdf> [Acceso 20 de Octubre 2014]

[6] M. Filipetto, (2001, Marzo) Predição Não-Linear de Séries Temporais Usando Redes Neurais RBF por Descomposição em Componentes Principais. Tese de Doutorado. Universidad Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação.

[7] C. Fajardo, D. González, B. Soto & F. Fernández-Riverola, "Water flows modelling and forecasting using a RBF neural network", Sistemas y Telemática, 6 (12), 13-31, 2008.

[8] J.D. Velásquez & C.J. Franco, "Pronóstico de series de tiempo con tendencia y ciclo estacional usando el modelo airline y redes neuronales artificiales", Ingeniería y Ciencia, 8 (15), 171-189, 2012.

[9] J. Velásquez, D. Isaac & R. Souza, "Ingeniare Spot Price Modelling in Brasil Using an Autoregressive Neural Network", Revista Chilena de Ingeniería, 6 (3), 394-403, 2008.

[10] J. Velazquez, C. Zambrano & L. Velez, "ARNN: A packages for time series forecasting using autoregressive neural netowork", Revista Avances en Sistemas e Informática, 8 (2), 177-181, 2011.

[11] P. Watta, M. Hassoun & N. Dannug, (1996) A Backprop Learning Tool for Function Approximation [Internet], Wayne State University. Disponible desde: <http://neuron.eng.wayne.edu/bpFunctionApprox/bpFunctionApprox.html>[Acceso 21 de Octubre 2014].

[12] M. Manic, B. Wilamowki & A. Malinowski. (2002, November) IEEE 2002 28th Annual Conference of the Industrial Electronics Society, [En Línea]. Disponible: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=1182851&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D1182851.

[13] I. Chaman, (2010, Septiembre) Integración Numérica con Redes Neuronales. Tesis de Pregrado. Benemérita Universidad Autónoma de Puebla. Facultad de Ciencias de la Computación.

[14] J. Montaño, (2002, Septiembre) Redes Neuronales Artificiales aplicadas al Análisis de Datos. Tesis Doctoral. Universitat de les Illes Balears. Facultad de Psicología.

[15] M.F. Bouami, (2005, Julio) Desarrollo y Optimización de Nuevos Modelos de Redes Neuronales Basadas en Funciones de Base Radial. Tesis Doctoral. Universidad de Granada. Departamento de Arquitectura y Tecnología de Computadores.

[16] C. Fajardo, (2008, Julio) Sistema Inteligente para la Estimación y Pronóstico de Caudales. Tesis Doctoral. Universidad de Vigo. Departamento de Informática.

[17] M. Arellano. (2001) Introducción al Análisis Clásico de Series de Tiempo [Internet], Universidad de Zaragoza. Disponible desde: <http://www.5campus.com/leccion/seriest> [Acceso 22 de Octubre 2014].

[18] M. Lichman. (2013) UCI Machine Learning Repository [Internet], University of California. Disponible desde: <http://archive.ics.uci.edu/ml> [Acceso 10 de Agosto 2013].