

Una Comparación entre Estrategias Evolutivas y RPROP para la Estimación de Redes Neuronales

A Comparison between Evolutionary Strategies and RPROP for Estimating Neural Networks

Diana Ortíz, Est., Fernán Villa, Est., Juan Velásquez, PhD.(c)

Grupo de Finanzas Computacionales

Escuela de Ingeniería de Sistemas, Facultad de Minas

Universidad Nacional de Colombia Sede Medellín

dianamortizs@gmail.com, fernan_villa@msn.com, jdvelasq@gmail.com

Recibido para revisión 26 de Marzo de 2007, aceptado 15 de Junio de 2007, versión final 31 de julio de 2007

Resumen—Rprop ha sido reconocido como uno de los más poderosos algoritmos para entrenar redes neuronales artificiales; sin embargo, el algoritmo de estrategias de evolución es un fuerte competidor para resolver problemas de optimización debido a su capacidad para buscar el óptimo global sin la necesidad de usar información sobre el gradiente. En este artículo, se comparan ambos algoritmos usando tres series de tiempo no lineales del mundo real, con el fin de determinar cual algoritmo ofrece mejores resultados en la práctica. Los resultados indican que estrategias de evolución converge más rápido que Rprop al punto de óptima local, pero en la mayoría de los casos los resultados obtenidos con Rprop son mejores en magnitud, aunque en la práctica los valores de la función objetivo son muy cercanos.

Palabras Clave—Redes Neuronales Artificiales, Estrategias Evolutivas, RPROP, Optimización, Series Temporales.

Abstract—Rprop has been recognized as one of the most powerful algorithms for training artificial neural networks; however, the evolution strategies algorithm is a strong competitor for solving optimization problems due to its capacity for searching the global optimum without the necessity of using information about of the gradient. In this paper, we compare both algorithms using three non-linear time series from the real world, with the aim of determinate which algorithm offers betters results in the practice. The results indicate that evolution strategies converges faster than Rprop to the local optima, but in most cases the results obtained using Rprop are better in magnitude, even in practice the values of the objective function are very near.

Keywords—Artificial Neural Networks, Evolution Strategies, RPROP, Optimization, Time Series

I. INTRODUCCIÓN

DURANTE la última centuria, se han desarrollado un gran número de técnicas para la predicción y el modelado de series temporales, debido principalmente a la importancia de este problema en muchas áreas de la ciencia. Por ejemplo, dichos modelos son usados en el campo empresarial para pronosticar los cambios en la demanda de un determinado producto, tomar decisiones sobre niveles de inventario, insumos, y demás.

Ha sido comúnmente aceptado que muchas series temporales poseen comportamientos que no pueden ser modelados de la mejor forma usando un modelo lineal. Es así entonces, como se han venido aplicando diferentes técnicas no lineales al modelado y la predicción de series temporales. De dichas técnicas, los modelos de redes neuronales han sido ampliamente usados en muchos casos prácticos, y de ha demostrado su utilidad y valor en la solución de este problema. Por ejemplo, en [1] Velásquez y Montoya desarrollan un modelo híbrido para la predicción del Índice de Precios al Consumidor en Colombia, mientras que en [2] Velásquez y González modelan la dinámica del índice de tipo de cambio real colombiano. Su éxito se debe a que dichos modelos son aproximadores universales de funciones que estén definidas en un dominio compacto [3, 5]. No obstante, la estimación de sus parámetros se ha caracterizado por ser un problema particularmente difícil debido a la multitud de puntos de mínima que hacen que las técnicas de gradiente sean de poca utilidad, por lo que se han venido desarrollando una cantidad importante de algoritmos que pretenden obviar este problema. De estos algoritmos, Rprop [6, 7], y sus

variantes, ha sido considerado como uno de los más eficientes; sin embargo, debido a sus características podría adolecer de las mismas falencias generales que sufren todas las técnicas basadas en gradientes.

Por otra parte, las técnicas de computación evolutiva, y particularmente las Estrategias Evolutivas (EE), se han posicionado como fuertes competidores de los algoritmos tradicionales de optimización debido a que realizan una búsqueda global en el espacio de soluciones sin tener en cuenta la información del gradiente, por lo que pueden converger al óptimo global del problema. Aunque han sido aplicadas a muchos problemas de optimización, no se han encontrado referencias relevantes sobre la aplicación de esta técnica a la estimación de los pesos de redes neuronales. Así un objetivo de este artículo es establecer una base para la comparación de esta técnica con otras que han sido tradicionales en la literatura de redes neuronales, y particularmente con Rprop.

En este mismo sentido, en la literatura más relevante no se encuentra ninguna comparación empírica entre dichas técnicas (Rprop y EE) que permite establecer cual técnica podría ser superior. Así, otro objetivo del artículo es analizar tres casos reales para determinar si realmente existen ganancias derivadas del uso de EE en comparación con iRprop+ cuando se estiman modelos de redes neuronales.

Para cumplir con los objetivos propuestos, el resto de este artículo está organizado como sigue. En la Sección II se discute el modelo utilizado. Las técnicas de optimización evaluadas son descritas en la Sección III. Seguidamente en la Sección IV, se describen los casos de análisis utilizados y se analizan los resultados obtenidos. Finalmente se concluye en la Sección V.

II. EL MODELO

Un perceptrón multicapa (MLP, por sus siglas en inglés) es un tipo de red neuronal que imita la estructura masivamente paralela de las neuronas del cerebro. Desde un punto de vista matemático, es un modelo que puede aproximar cualquier función continua definida en un dominio compacto con una precisión arbitraria previamente establecida [3, 5]. En la práctica, los MLP se han caracterizado por ser muy tolerantes a información incompleta, inexacta o contaminada con ruido [8], por lo que han sido usados en la modelación empírica de series temporales no lineales; en [9] Zhang et al presentan una revisión general sobre el estado del arte, mientras que aplicaciones específicas son presentadas en [10, 15] entre muchos otros.

Una serie temporal se define como una secuencia de observaciones en el tiempo y_1, y_2, \dots, y_T , para la que se pretende construir una función que permita obtener y_t en función de sus valores pasados, $y_{t-1}, y_{t-2}, \dots, y_{t-P}$. Dicha función puede ser especificada como un perceptrón multicapa que puede ser representado matemáticamente como:

$$y_t = \beta_* + \sum_{h=1}^H \beta_h \cdot g\left[\frac{1}{2\sigma_y^2} \left(\alpha_{*,h} + \sum_{p=1}^P \alpha_{p,h} y_{t-p} \right) \right] + \varepsilon_t \quad (1)$$

Donde los parámetros $\Omega = [\beta_*, \beta_h, \alpha_{*,h}, \alpha_{p,h}], h = 1 \dots H, p = 1 \dots P$ son estimados usando el principio de máxima verosimilitud de los residuales, el cual equivale a la minimización de una función de costo que es definida usualmente como el error cuadrático medio. La ecuación definida por (1) equivale a un modelo estadístico no paramétrico de regresión no lineal [16]. En (1) se asume que ε_t sigue una distribución normal con media cero y varianza desconocida σ^2 . H representa el número de neuronas en la capa oculta. P es el número de rezagos de la variable dependiente. $Y(g(\cdot))$ es la función de activación de las neuronas de la capa oculta. En el contexto de las series temporales, el modelo puede ser entendido como una combinación lineal ponderada de la transformación no lineal de varios modelos autorregresivos. En la Figura 1 se presenta una representación pictórica del modelo postulado en (1).

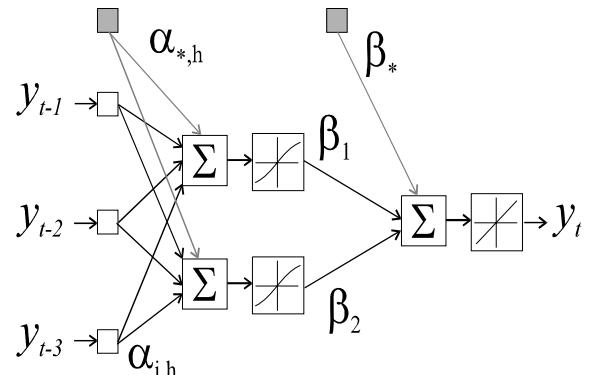


Figura 1. Perceptrón multicapa con tres entradas y una salida.

El problema de identificación o estimación de parámetros está relacionado con la multiplicidad de puntos de mínima local de la función de error usada para la estimación de los parámetros, debido a que:

- El modelo puede estar mal condicionado, lo cual es síntoma de que las diferencias entre las magnitudes de los parámetros Ω son muy grandes. Esto puede causar que las neuronas en la capa oculta estén en su nivel de saturación (diferentes entradas netas producen la misma salida), y consecuentemente que el gradiente de la función de verosimilitud sea cercano a cero, por lo que la optimización numérica se hace ineficiente.
- Se pueden obtener modelos diferentes con el mismo error al permutar las neuronas de la capa oculta sin variar los valores de los parámetros, lo que equivale a intercambiar algunos de los elementos de Ω .
- Se pueden obtener modelos diferentes con el mismo error, cambiando de signo los parámetros asociados con las conexiones que entran y salen de la misma

neurona oculta, ya que para funciones de activación que cumplen con $g(u) = \text{sgn}(u)|u|$, el cambio de signos no altera el resultado del MLP.

El efecto nocivo de éstas características puede ser mitigado al modificar la función de activación de las neuronas ocultas y al imponer algunas restricciones a los parámetros del modelo:

- Tal como en [17] algunos autores han sugerido que las funciones tipo sigmoidea que son simétricas alrededor del origen convergen más rápidamente que la función sigmoidea tradicional. Adicionalmente, la incorporación de un término lineal puede ayudar a la convergencia, ya que se evita la saturación de la neurona y garantiza un gradiente mínimo cuando la salida neta de la función de activación es cercana a sus valores extremos. Consecuentemente con las razones expuestas, el modelo propuesto activa sus neuronas de la capa oculta usando la función:

$$g(u) = \frac{1 - \exp(-u)}{1 + \exp(-u)} + 0.01u \quad (2)$$

- El conjunto de parámetros β_h , $h = 1, \dots, H$ pueden restringirse a ser positivos, ya que si uno de ellos es negativo, su signo puede alterarse multiplicando los parámetros de las conexiones que entran y salen de la correspondiente neurona por menos uno; esto se debe a que (2) es simétrica alrededor del origen, por lo que $g(u) = -g(-u)$.
- El conjunto de parámetros β_h , $h = 1, \dots, H$ se restringe a que sus componentes estén ordenados de forma creciente $0 < \beta_1 \leq \beta_2 \leq \dots \leq \beta_H$, para evitar que se puedan obtener nuevas configuraciones permutando las neuronas de la capa oculta.

La división de la entrada neta a la neurona por $2\sigma_y$ escala automáticamente las entradas al modelo, de tal forma que no es necesario realizar su preprocesamiento.

III. ALGORITMOS UTILIZADOS

A. Técnica de optimización: RPROP

El algoritmo de optimización RPROP desarrollado por Redmiller y Braun [6, 7], es considerado como uno de los algoritmos más poderosos para la estimación de los parámetros (o pesos) de una red neuronal. En este proceso se busca encontrar los valores del vector de parámetros Ω del modelo definido en (1) tal que se minimice la diferencia entre los valores reales y_t^* y los valores calculados y_t . Sea E la función de error utilizada, y w_i las componentes de Ω . En esta investigación se utilizó la variante iRprop+ de Igel y Hüskens [18], en la cual los parámetros del modelo en la iteración $t+1$ son actualizados como:

$$w_i^{(t+1)} = w_i^{(t)} + \Delta w_i^{(t+1)} \quad (3)$$

donde $\Delta w_i^{(t+1)}$ es estimado con una función del cambio de signo de la derivada del error entre las iteraciones t y $t-1$, y del tamaño del paso Δ_i tal como se realiza tradicionalmente en las técnicas basadas en el gradiente.

Así, si el signo de la derivada no cambia en las dos últimas iteraciones, entonces el tamaño de paso Δ_i es incrementado en un factor η^+ pero limitado a que su valor máximo no supere Δ_{\max} que corresponde al tamaño máximo de la modificación de w_i .

Cuando se presenta el cambio de signo de las derivadas el algoritmo sobrepasa el punto de mínima; consecuentemente, el tamaño de paso es reducido en un factor η^- pero limitando el tamaño mínimo de modificación de w_i a un valor Δ_{\min} . Si la derivada es cero, no se modifica el tamaño de paso.

El algoritmo iRprop+ se basa en el uso de técnicas de gradiente por lo que puede adolecer de la misma problemática general que sufren esta clase de técnicas. Una amplia discusión es presentada en [17]. De ahí que es necesario que exista la necesidad de optimizar desde diferentes puntos de arranque, y de utilizar algoritmos de inicialización de pesos, para intentar que el algoritmo no quede atrapado en un mínimo local.

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 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*.

B. Estrategias Evolutivas

Esta técnica de optimización basada en el principio de evolución de Darwin, fue originalmente desarrollada por Rechenberg [19, 20], y llevada a su forma actual por Schwefel [21, 22].

En EE, cada individuo está formado por un vector que contiene tantas variables objeto, x_i , como parámetros tiene el vector Ω , y sendas variables de estrategia, σ_i . De esta forma un individuo tiene la siguiente estructura:

$$I = [x_1, x_2, \dots, x_n, \sigma_1, \sigma_2, \dots, \sigma_n] \quad (4)$$

La optimización es realizada sobre una población conformada por N individuos siguiendo el algoritmo descrito en la Figura 2.

El algoritmo se inicia generando una población de N individuos donde las variables x_i son generadas de forma aleatoria; y las variables σ_i se inician en 3 unidades siguiendo

las sugerencias dadas por Bäck en [23].

El desempeño de cada individuo se evalúa como función de la medida de error utilizada en el modelo de redes neuronales. La selección de padres en EE, se realiza de forma probabilística asignando a cada parente una probabilidad igual de ser seleccionado.

```

t:=0;
Iniciar_Poblacion P(t);
Evaluar_Desempeño P(t);
Mientras no criterio hacer
t := t + 1;
P'(t):= Seleccionar_Padres P(t);
Reproducir P'(t);
Mutar P'(t)
Evaluar_Desempeño P'(t)
P := Seleccionar P(t), P'(t)
Fin Mientras

```

Figura 2. Seudocódigo del algoritmo evolutivo.

El proceso de reproducción se realiza escogiendo dos padres de forma aleatoria entre la población, para luego proceder a realizar un proceso de combinación entre ellos; existen diversas formas para realizar este proceso, pero se escogió la reproducción sexual panmínica intermedia para maximizar la capacidad de búsqueda del algoritmo. En este tipo de reproducción sexual, cada una de las componentes del nuevo individuo, c_i , se calculan como:

$$c_i = c_{S,i} + \chi_i \times (c_{T,i} - c_{S,i}) \quad (5)$$

donde los subíndices S y T representan cada uno de los padres seleccionados, y el valor χ_i representa un número aleatorio uniforme entre cero y la unidad.

Una vez calculada la población de hijos, se procede a mutar cada uno de ellos; esto se hace, de la siguiente forma:

$$\sigma'_i = \sigma_i \exp(\tau' \cdot N[0, 1] + \tau \cdot N_i[0, 1]) \quad (6)$$

$$x'_i = x_i + \sigma'_i \cdot N_i[0, 1] \quad (7)$$

Donde $N[0, 1]$ representa un número aleatorio normal con media cero y desviación estándar unitaria, y los símbolos τ' y τ representan constantes.

Una vez se han mutado los individuos usando (6) y (7), se procede a escoger la nueva población con la que se iniciará el ciclo nuevamente. En este caso en particular, se realiza una selección del tipo +, en la cual se hace un ordenamiento del conjunto de padres unido con el conjunto de hijos, y se selecciona la nueva población de esta unión.

Las EE poseen la propiedad de auto-adaptación, lo cual implica que no sólo evolucionan las variables del problema, sino también los parámetros mismos del algoritmo (variables de estrategia).

IV. CASOS DE APLICACIÓN

En esta sección se presenta la comparación entre iRprop+ y EE para distintas series de tiempo reales y para varias configuraciones de los modelos de redes neuronales. Las series utilizadas son 'Pasajeros de una Aerolínea' de Box y

Jenkins, 'Linceos Canadienses' y el 'Número de Manchas Solares' de Wolf; ellas han sido ampliamente estudiadas en la literatura por diversos autores. Cada serie se pronostica con diferentes modelos de Redes Neuronales. Las Redes Neuronales son optimizadas mediante EE y RPROP. Se compara el desempeño de RPROP contra el de EE, y además, con fines ilustrativos, se presentan los resultados obtenidos en otras investigaciones. Las tres series mencionadas fueron elegidas debido a su comportamiento no lineal como se puede apreciar en las Figuras 3, 4 y 5.

Todos los algoritmos utilizados para cada uno de los modelos de cada caso se implementaron y se ejecutaron en MatLab versión 7.0. EE se ejecutó una sola vez con 20 individuos y un máximo de 1000 generaciones para cada configuración y cada caso considerado. iRprop+ se ejecutó desde 20 puntos de arranque diferentes y con un máximo de 1000 iteraciones.

A. Primer Caso: Pasajeros de una Aerolínea.

More Esta serie de tiempo contiene el registro del número total de pasajeros transportados por mes por una aerolínea, desde enero de 1949 hasta diciembre de 1960. Cada uno de los trece modelos presentados en la Tabla I fue estudiado por Faraway y Chatfield [13]. Para cada modelo, los datos de la serie se transformaron utilizando la función logaritmo natural (base - e); igualmente se usaron los primeros 120 datos para entrenamiento y los 12 últimos para validación, tal como fue realizado en [13].

En la Referencia [13] los autores reportaron para cada uno de los modelos de la Tabla I la sumatoria del error cuadrático (SSE) para las muestras de entrenamiento y validación. Igualmente para cada uno de dichos modelos, se procedió a realizar la estimación de sus parámetros usando EE y iRprop+; posteriormente, se estimaó el estadístico de ajuste para las muestras de calibración y de validación. Se puede apreciar que todos los errores obtenidos con iRprop+ y EE son menores que los reportados en [13]. Los errores alcanzados con iRprop+ en los modelos 2, 3 y 7 – 11, tanto en entrenamiento como en validación, son menores que los obtenidos con EE, mientras que en los modelos 4, 12 y 13 con EE se obtuvo el menor error. En los Modelos 1, 5 y 6 tanto con EE como con RPROP se obtuvieron los mismos errores tanto en validación como en entrenamiento. En la Figura 3 se muestran los valores reales y los pronosticados usando el modelo 1 de la Tabla I.

En el 53.85% de los modelos con iRprop+ se obtuvo el menor error tanto en validación como en entrenamiento, en el 23.08% de los modelos con EE y en el 23.08% de los modelos con ambos (iRprop+ y EE). Resulta conveniente utilizar iRprop+ para esta serie, ya que para la mayoría de los modelos se obtuvo menores errores utilizando este algoritmo de optimización.

El tiempo promedio para el pronóstico de la serie, con el mínimo error, mediante iRprop+ fue de 335.28 segundos y con EE de 313.75, EE es 6.86% más rápido que RPROP; este tiempo depende de la estructura de la red neuronal, entre más rezagos

y neuronas en la capa oculta posea más aumenta el tiempo.

B. Segundo Caso: Lince Canadienses

En esta serie se encuentra registrada la cantidad de lince capturados anualmente, desde 1821 hasta 1934, en los alrededores del río Mackenzie ubicado en el distrito de Northem Canadá. Esta serie fue estudiada por Capmbell y Walker [24], Rao y Garb [25], y Zhang [26]. Los datos de la serie se transformaron utilizando la función logaritmo base-10; de sus 114 datos se tomaron los 100 primeros para entrenamiento y los últimos 14 para validación tal como se ha realizado en estudios pasados.

En la Tabla II se presenta el error cuadrático medio (MSE) calculado para distintos modelos estimados. En [26] se reporta 0.021 como el MSE de validación al pronosticar la serie de este caso con un modelo de 5 neuronas en la capa oculta; para un modelo similar se obtuvo con iRprop+ 0.039 y 0.015 en entrenamiento y validación respectivamente; para el mismo modelo con EE se obtuvo 0.049 y 0.017. Los errores obtenidos en validación con iRprop+ y EE son menores que el reportado en [26]. En la Figura 5 se presentan los valores reales de la serie de tiempo y los pronosticados con el modelo 1 de la Tabla II.

Los errores alcanzados con iRprop+, al pronosticar la serie de este caso, con los modelos 1 – 9 de la Tabla II, son menores que los obtenidos con EE durante la fase de

entrenamiento, mientras que en la validación los menores errores se obtuvieron con EE. El menor error se obtuvo con el modelo 10 tanto en entrenamiento como en validación, se obtuvo con iRprop+, mientras que, en el modelo 10 con EE.

En el 82% de los modelos de la Tabla III, se obtuvo el menor error en entrenamiento con iRprop+, mientras que, con EE se obtuvo el menor error en validación; en el 9% de los modelos el menor error tanto en entrenamiento como en validación fue obtenido con iRprop+, mientras que en el 9. % con EE. Dados estos resultados, no se puede decidir cual de los dos algoritmos sería más conveniente utilizar para pronosticar esta serie, ya que en la mayoría de los modelos, ambos no reducen a la vez el error en entrenamiento y en validación.

El tiempo promedio para la estimación de los modelos fue de 699.83 seg. cuando se usó iRprop+; y de 311.95 para EE. Así, EE es 124% más rápida que iRprop+. La diferencia de velocidades es más del doble, por este criterio se puede decidir que es mejor utilizar el método de EE para optimizar el modelo que pronostica la serie de lince; además, la diferencia relativa entre los errores usando ambas técnicas para pronosticar la serie, y para los modelos 1, 4, 7, 10 y 11 es poco significativa, como se muestra en la Tabla III.

Tabla 1. Sumatoria del error cuadrático para diferentes modelos pronosticando la serie del Primer Caso.

Modelo	Rezagos	Neuronas	F&C	Entrenamiento		Validación		
				RPROP	EE	F&C	RPROP	EE
1	1-4	2	7.74	1.27	1.27	1.03	0.16	0.16
2	1-13	2	0.73	0.73	2.33	0.71	0.10	0.26
3	1-13	4	0.26	0.21	1.60	1.12	0.01	0.20
4	1,12	2	2.30	0.66	0.57	0.34	0.06	0.04
5	1,12	4	2.16	0.46	0.46	0.44	0.01	0.01
6	1,12	10	1.77	0.45	0.45	0.59	0.03	0.03
7	1,2,12	2	2.17	0.65	1.22	0.29	0.11	0.15
8	1,2,12	4	1.91	0.42	1.23	1.403	0.03	0.16
9	1,2,12,13	2	0.99	0.59	1.21	0.52	0.08	0.14
10	1,2,12,13	4	0.81	0.50	1.20	0.52	0.08	0.14
11	1,12,13	1	1.18	0.99	1.22	0.50	0.16	0.15
12	1,12,13	2	1.03	0.67	0.60	0.50	0.06	0.04
13	1,12,13	4	0.84	0.47	0.39	0.62	0.04	0.02

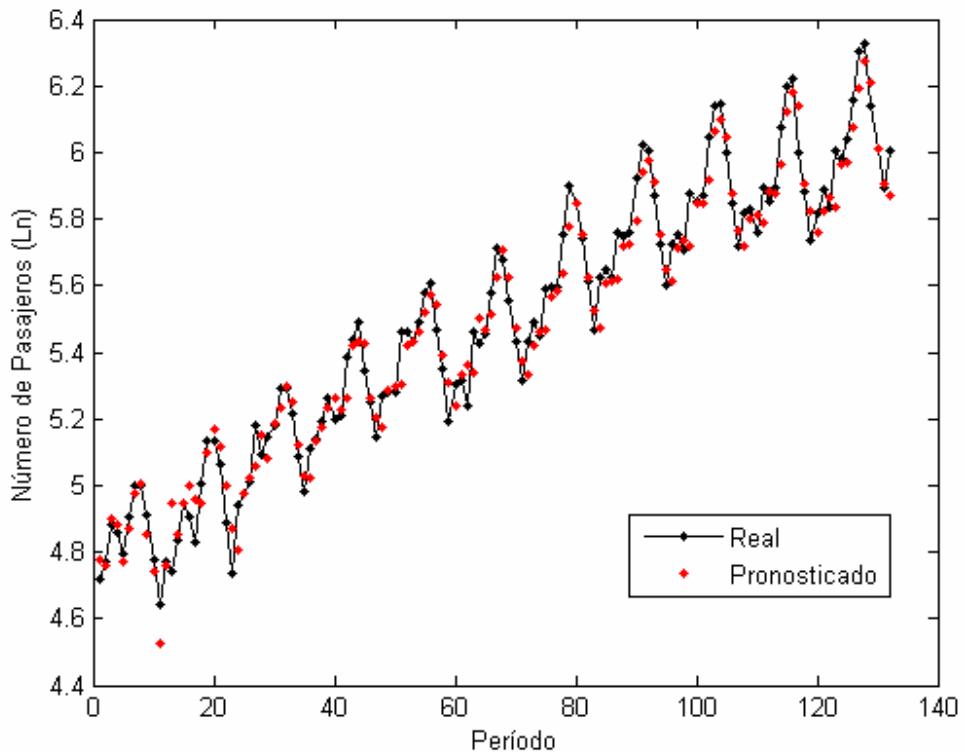


Figura 3. Valores real y pronosticado para la serie de pasajeros de una aerolínea

C. Tercer Caso: Manchas Solares

Esta serie contiene el número anual de las manchas solares sobre la cara del sol durante el período de 1700 a 1956, para un total de 256 datos. Esta serie ha sido estudiada por varios investigadores; entre ellos se encuentran Cottrell et al. [27], de Groot & Wurtz [28], Zhang [26].

En [27] Cottrell et al., utilizaron de las 256 observaciones de la serie, las primeras 221 para entrenamiento y las últimas 35 para validación; reportó los resultados del MSE durante las fases de entrenamiento y validación para los modelos presentados en la Tabla IV. En la Figura 5 se presentan los valores reales de la serie de tiempo y los pronosticados con el modelo 2 de la Tabla IV.

Los modelos propuestos en [27] fueron estimados utilizando EE y iRprop+. Las Tablas IV y V resumen los MSE obtenidos, tanto para entrenamiento como para validación.

En la Tabla V puede observarse que los errores obtenidos con iRprop+ y EE son menores que los reportados en [27]. Además, con iRprop+ se obtienen los menores errores tanto en entrenamiento como en validación para los modelos 1, 2, 4 y 6 – 9 de la Tabla V; mientras que con EE se obtienen los menores errores para el modelo 5. En el modelo 2 de la Tabla IV y en el modelo 3 de la Tabla V, se tiene el mismo comportamiento de los modelos 1 – 9 de la Tabla III utilizados

en el Caso 2, donde el menor error para entrenamiento se logra con iRprop+ y el de validación con EE.

En el total de modelos utilizados en este caso, (véase las Tablas IV y V), en el 73% de los modelos, se obtuvo el menor error con iRprop+, tanto en entrenamiento como en validación; mientras que, para el 9% de los modelos, con EE se obtuvo el menor error en entrenamiento y validación. Para el 18% el menor error para entrenamiento se obtuvo con RPROP y el de validación con EE.

A partir de estos resultados se puede concluir que es conveniente utilizar RPROP para pronosticar esta serie, ya que para la mayoría de los modelos se obtuvo menores errores utilizando este algoritmo de optimización.

El tiempo promedio para el pronóstico de la serie de este caso, con el mínimo error, mediante RPROP fue de 1039.02 seg. y con EE de 27.83 seg. Así, EE es 3633% más rápido que RPROP. La diferencia de velocidades es enorme; si se desea más velocidad, por este criterio se puede decidir, que es mejor utilizar el método de EE para optimizar el modelo que pronostica la serie de Manchas Solares.

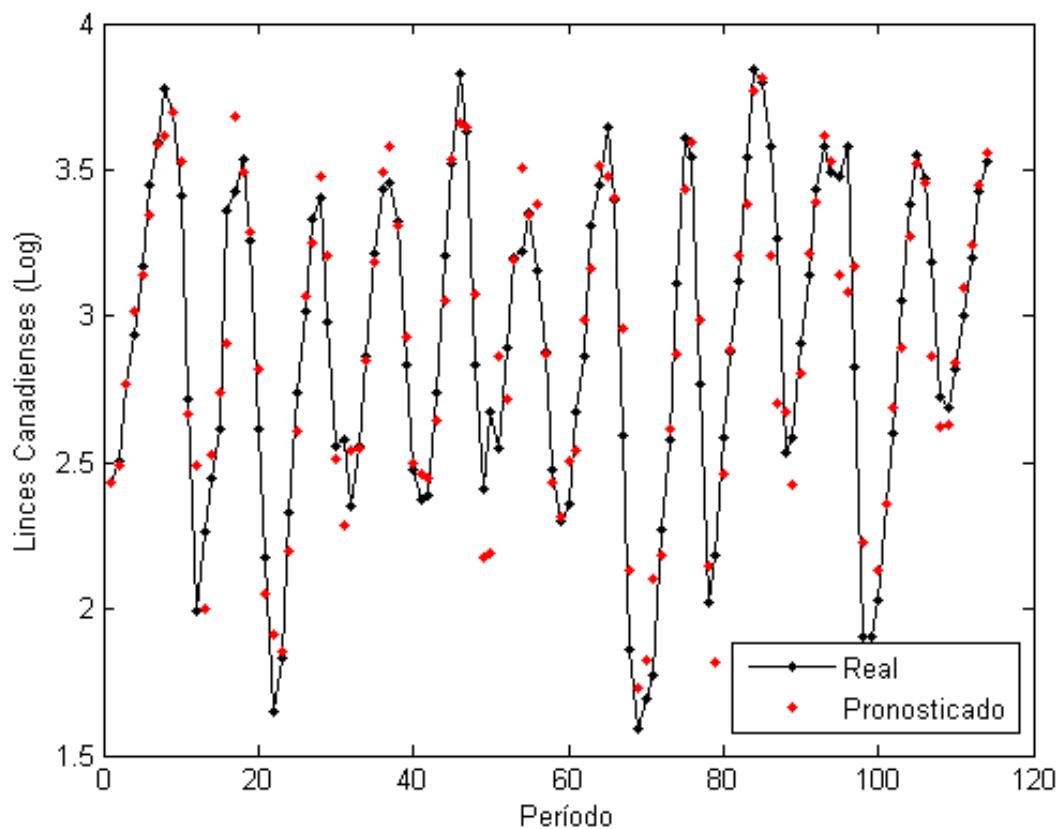


Figura 4. Valores real y pronosticado para la serie de tiempo de Linceos Canadienses

Tabla 2. Error cuadrático medio para diferentes modelos pronosticando la serie del Segundo Caso.

Modelo	Rezagos	Neuronas	Entrenamiento		Validación	
			RPROP	EE	RPROP	EE
1	1 – 7	4	0.042	0.044	0.026	0.010
2	1,2,3,8,9,10	2	0.035	0.050	0.032	0.027
3	1,2,3,8,9,10	4	0.034	0.042	0.035	0.030
4	1 – 6	2	0.048	0.050	0.017	0.015
5	1 – 10	3	0.033	0.045	0.019	0.016
6	1,2,3,4,8,9,10	2	0.037	0.045	0.044	0.033
7	1,2,3,4,8,9,10	5	0.033	0.039	0.036	0.026
8	1 – 8	4	0.035	0.047	0.021	0.012
9	1 – 9	5	0.032	0.045	0.033	0.026
10	1 – 6	4	0.039	0.046	0.013	0.018
11	1 – 5	5	0.047	0.045	0.012	0.010

Tabla 3. Diferencia Relativa del MSE obtenido con EE respecto al de RPROP para diferentes modelos de la Tabla 2.

Modelo	Entrenamiento	
	EE / RPROP	
1	4.76%	
2	42.86%	
3	23.53%	
4	4.17%	
5	36.36%	
6	21.62%	
7	18.18%	
8	34.29%	
9	40.63%	
10	17.95%	
11	-4.26%	

Tabla 4 Error cuadrático medio para diferentes modelos pronosticando la serie del Tercer Caso

Modelo	Rezagos	Neuronas	Entrenamiento			Validación		
			Cot.	RPROP	EE	Cot.	RPROP	EE
1	1,2,3,4	5	123	134	164	129	119	210
2	1,2,9,11	2	135	133	148	N/A	154	144

Tabla 5. Media del error cuadrático (MSE) para diferentes modelos pronosticando la serie del Tercer Caso

Modelo	Rezagos	Neuronas	Entrenamiento		Validación	
			RPROP	EE	RPROP	EE
1	1 – 11	2	159	177	123	188
2	1 – 11	4	123	180	112	162
3	1,2,9,11	5	129	144	137	130
4	1,2,3,9,10,11	1	153	198	183	212
5	1,2,3,9,10,11	2	118	141	138	132
6	1,2,3,9,10,11	5	113	155	123	145
7	1,3,4,9,10,11	1	177	185	172	180
8	1,3,4,9,10,11	2	147	171	164	185
9	1,3,4,9,10,11	5	105	158	170	173

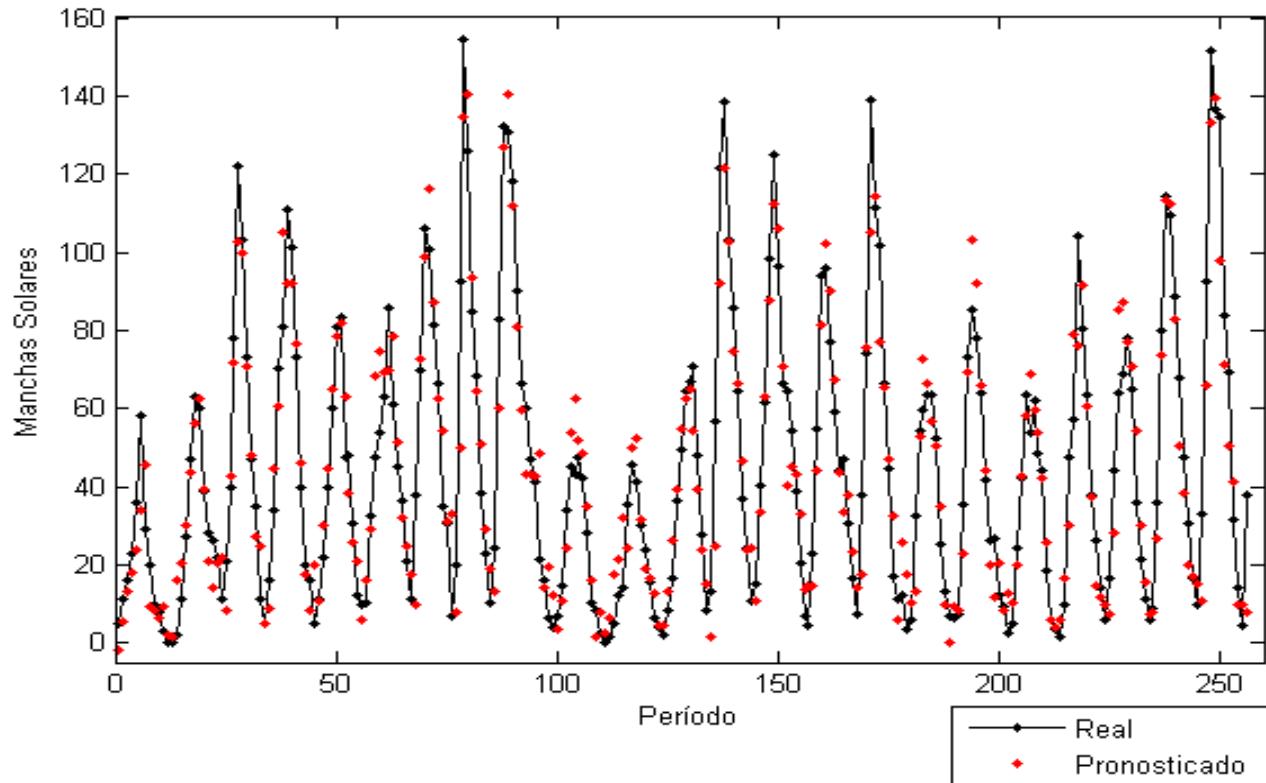


Figura 5. Serie de Manchas Solares

V. CONCLUSIONES

A pesar que en muchos casos iRprop+ ofrece una mejor aproximación de los pesos o soluciones de la red neuronal al modelar las series de tiempo, con EE converge más rápido a un error mínimo, que en varios casos es menor o muy cercano al obtenido con iRprop+.

En todas las series pronosticadas se obtuvieron errores, tanto con iRprop+ como con EE, menores que los reportados por otros autores, por ello se puede afirmar que los dos métodos estudiados en éste artículo son una opción razonable a la hora de optimizar redes neuronales.

REFERENCIAS

- [1] J.D. Velásquez, S.F. Montoya, "Modelado del índice de precios al consumidor usando un modelo híbrido basado en redes neuronales artificiales," Revista Dyna, Año 72, Nro. 147, pp. 85 - 93. Medellín, Noviembre de 2005.
- [2] J.D. Velásquez, L.M. González, "Modelado del índice de tipo de cambio real colombiano usando redes neuronales artificiales," Cuadernos de Administración. Vol 19, No 32, p 319-336, 2007.
- [3] K. Hornik, M. Stinchcombe, H. White, "Multilayer feedforward networks are universal approximators," Neural Networks, 2, 359-366, 1989.
- [4] G. Cybenko, "Approximation by superpositions of a sigmoidal function," Mathematics of Control, Signals and Systems 2, 202-314, 1989.
- [5] K. Funahashi, "On the approximate realization of continuous mappings by neural networks," Neural Networks, 2, 183-192, 1989.
- [6] M. Riedmiller, "Advanced supervised learning in multi-layer perceptrons – from backpropagation to adaptive learning algorithms," Computer Standards and Interfaces, 16:265–278, 1994.
- [7] M. Riedmiller, H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," In Proceedings of the IEEE International Conference on Neural Networks, pages 586–591. IEEE Press, 1993.
- [8] T. Masters, "Practical neural network recipes in C++," New York: Academic Press, 1993.
- [9] G. Zhang, B. Patuwo, M. Hu, "Forecasting with artificial neural networks: the state of the art. International Journal of Forecasting," 14, 35-62, 1998.
- [10] S. Heravi, D. Osborn, C. Birchenhall, "Linear versus neural network forecasts for european industrial production series," International Journal of Forecasting (20), 435-446, 2004.
- [11] N. Swanson, H. White, "Forecasting economic time series using adaptive versus nonadaptive and linear versus non-linear econometric models," International Journal of Forecasting (13), 439-461, 1997.
- [12] Swanson, N. y White, H. "A model selection approach to real time macroeconomic forecasting using linear models and artificial neural networks," Review of Economics and Statistics (39), 540-550, 1997.
- [13] J. Faraway, C. Chatfield, "Time series forecasting with neural networks: A comparative study using the airline data," Applied Statistics, 47(2), 231–250, 1998.

- [14] G. Darbellay, M. Slama, "Forecasting the short-term demand for electricity: do neural networks stand a better chance?," International Journal of Forecasting, 16, 71-83, 2000.
- [15] C. Kuan, T. Liu, "Forecasting exchange rates using feedforward and recurrent neural networks," Journal of Applied Econometrics (10), 347-364, 1995.
- [16] W. Sarle, "Neural networks and statistical models," Documento presentado en The 19th Annual SAS Users Group Int. Conference, Cary, NC: SAS Institute, pp. 1538-1550, 1994.
- [17] Y. LeCun, L. Bottou, G. B Orr, K. R. Muller, Neural Networks - Tricks of the Trade, Springer Lecture Notes in Computer Sciences 1524, chapter Efficient Backprop, pp. 5-50, 1998.
- [18] C. Igel, M. Hüskens, "Improving the Rprop learning algorithm," Proceedings of the Second International Symposium on Neural Computation, NC2000. ICSC Academic Press. pp 115-121
- [19] I. Rechenberg, "Cybernetic solution path of an experimental problem," Royal Aircraft Establishment, Library translation No. 1122, Farnborough, Hants, UK. August, 1965.
- [20] I. Rechenberg, "Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution," Frommann-Holzboog, Stuttgart, 1973.
- [21] H-P. Schwefel, "Kybernetische Evolution als Strategie der experimentellen Forschung in der Stromungstechnik," Diplomarbeit, Technische Universität Berlin, 1965.
- [22] H-P. Schwefel, "Evolutionsstrategie und numerische Optimierung," Dissertation, Technische Universität Berlin, May 1975.
- [23] T. Bäck, "Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms," Oxford University Press, 1996.
- [24] M. J. Campbell, A. M. Walker, "A survey of statistical work on the mackenzie river series of annual canadian lynx trappings for the years 1821-1934 and a new analysis". Journal of the Royal Statistical Society. Series A. Statistics in Society, 140, 411 -431, 1977.
- [25] T. S. Rao, M. Gabr, "An introduction to bispectral analysis and bilinear time series models," Lecture Notes in Statistics, vol. 24. (pp. 528- 535) New York7 Springer-Verlag, 1984.
- [26] G. Zhang, "Time Series forecasting using a hybrid ARIMA and neural network model," Neurocomputing. 50, 159-175, 2003.
- [27] M. Cottrell, B. Girard, Y. Girard, M. Mangeas, C. Muller, "Neural modeling for time series: A statistical stepwise method for weight elimination," IEEE Transactions on Neural Networks, 6(6), 1355-1364, 1995.
- [28] C. de Groot, D. Wurtz, "Analysis of univariate time series with connectionist nets: A case study of two classical examples.Neurocomputing," 3, 177-192, 1991.

Diana Marcela Ortíz Salazar, es estudiante de 10 semestre de Ingeniería de Sistemas e Informática de la Universidad Nacional de Colombia Sede Medellín. Es miembro del grupo de Investigación en Finanzas Computacionales. Sus áreas de interés son la inteligencia computacional, metodologías de predicción, y optimización heurística.

Fernán Alonso Villa Garzón, es estudiante de 10 semestre de Ingeniería de Sistemas e Informática de la Universidad Nacional de Colombia Sede Medellín. Es miembro del grupo de Investigación en Finanzas Computacionales. Sus áreas de interés son la inteligencia computacional, metodologías de predicción, y optimización heurística.

Juan David Velásquez Henao, es Profesor Asociado de la Escuela de Sistemas de la Universidad Nacional de Colombia Sede Medellín. Es director del grupo de Investigación en Finanzas Computacionales. Sus áreas de interés están relacionadas con la econometría no lineal, la estadística computacional y las técnicas no lineales de predicción. Sus trabajos han sido publicados en diferentes revistas de circulación nacional e internacional.