



UNIVERSIDAD SANTO TOMÁS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA
T U N J A



ingenio Magno

8

No. 1



ISSN (versión impresa)

2145-9282

ISSN (en línea)

2422-2399

Enero-junio de 2017, vol. 8, no. 1

Universidad Santo Tomás

Tunja, Boyacá

Publicación semestral

Hecho el depósito que establece la ley

© Derechos reservados

Universidad Santo Tomás

Suscripción y canje

Unidad de Investigación

Cll. 19 No. 11-64

Universidad Santo Tomás, Tunja-Colombia

PBX: 744 0494

Línea gratuita: 018000 932340

desde cualquier lugar del país

<http://revistas.ustatunja.edu.co/index.php/ingeniomagno>

Los conceptos expresados en los artículos son de exclusiva responsabilidad de sus autores y no comprometen a la Institución o a la publicación.

División de Arquitectura e Ingenierías

INGENIO MAGNO	Tunja Colombia	Vol. 8 No. 1	pp. 1-172	Enero- junio	2017
---------------	-------------------	-----------------	-----------	-----------------	------



UNIVERSIDAD SANTO TOMÁS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA
T U N J A



Directivos

P. Jorge Ferdinando RODRÍGUEZ RUIZ, O.P.
Rector

P. José Antonio BALAGUERA CEPEDA, O.P.
Vicerrector Administrativo-Financiero

P. Javier Antonio CASTELLANOS, O.P.
Vicerrector Académico

P. Samuel Elías FORERO BUITRAGO, O.P.
Decano de División de Ingeniería y Arquitectura

Jimena BOHÓRQUEZ HERRERA, Ph.D.
Directora Unidad de Investigaciones

Édgar Andrés GUTIÉRREZ CÁCERES, Esp.
Director Centro de Investigaciones en Ingeniería
San Alberto Magno - (CIAM)

Editor

Fredy Andrés Aponte Novoa, M.Sc.
Centro de Investigación en Ingeniería Alberto Magno
(CIAM)
ingeniomagno@ustatunja.edu.co

Equipo Editorial

Edwin Blasnilo Rúa Ramírez, M.Sc.
Coeditor Universidad Santo Tomás, Tunja, Colombia
edwin.rua@usantoto.edu.co

José Ricardo Casallas Gutiérrez, M.Sc.
Coeditor Universidad Santo Tomás, Tunja, Colombia
jose.casallas@usantoto.edu.co

John Fredy Guzmán Vargas
Profesional en Filosofía y Letras
Corrector de estilo, redactor
fredyguzmanvargas@gmail.com

Albany Milena Lozano Násner, M.Sc.
Traducción español-portugués
mlnasner@gmail.com

Departamento de Idiomas
Traducción español-inglés
Universidad Santo Tomás, Tunja, Colombia
coordinacion.idiomas@ustatunja.edu.co

David Enrique González Camargo
Administrador OJS
Universidad Santo Tomás, Tunja, Colombia
ing.investigacion@ustatunja.edu.co

Comité Científico

Antonio Moreira Teixeira, Ph.D.
Universidad de Aberta (Lisboa, Portugal)
antonio.teixeira@uab.pt

Electo Eduardo Silva Lora, Ph.D.
Univerdidad Federal de Itajubá (Itajubá M.G., Brasil)
electo@unifei.edu.br

María Julia Mazzarino, Ph.D.
Universidad de Buenos Aires (Buenos Aires, Argentina)
mmazzari@crub.uncoma.edu.ar

Antonio Rico Sulayes, Ph.D.
Universidad de las Américas Puebla (Puebla, México)
antonio.rico@udlap.mx

Carlos Enrique Montenegro Marín, Ph.D.
Universidad Distrital Francisco José de Caldas (Bogotá
D.C., Colombia)
cemontenegrom@udistrital.edu.co

César Darío Guerrero Santander, Ph.D.
Universidad Autónoma de Bucaramanga (Bucaramanga
- Santander, Colombia)
cguerrer@unab.edu.co

Comité Editorial de la Revista

Antonio José Bula Silvera, Ph.D.
Universidad Católica del Norte (Barranquilla - Atlántico,
Colombia)
abula@uinorte.edu.co

Wilson Javier Pérez Holguín, Ph.D.
Universidad Pedagógica y Tecnológica de Colombia
(Tunja - Boyacá, Colombia)
wilson.perez@uptc.edu.co

José Carlos Escobar Palacios, Ph.D.
Investigador Universidad Federal de Itajubá (Itajubá
M.G., Brasil)
jocesobar@unifei.edu.br

Camilo Andrés Lesmes Fabian, Ph.D.
Universidad Santo Tomás (Tunja - Boyacá, Colombia)
camilo.lesmes@usantoto.edu.co

Impresión

Editorial Jotamar Ltda.
Calle 57 No. 3-39
Tel.: (8) 745 7120
editorialjotamar@yahoo.com
Tunja - Boyacá - Colombia

**Optimización en la
elaboración de redes
neuronales artificiales
adaptativas usando
una metodología
de algoritmo de poda**

**Optimization in the
production of
adaptive artificial
neural networks
using a pruning
algorithm methodology**

**Otimização na
elaboração de redes
neuronais artificiais
adaptativas usando
uma metodologia de
algoritmo de poda**

Para citar este artículo / To reference this article / Para citar este artigo: González Salcedo, L. O., Gotay Sardiñas, J., Roodschild, M., Will, A. L. y Rodríguez, S. (2017). Optimización en la elaboración de redes neuronales artificiales adaptativas usando una metodología de algoritmo de poda. *Ingenio Magno*, 8(1), 44-56.

Luis Octavio González-Salcedo

Universidad Nacional de Colombia, sede Palmira,
Facultad de Ingeniería y Administración,
Grupo de Investigación en Materiales
y Medio Ambiente
logonzalezsa@unal.edu.co

Jorge Gotay-Sardiñas

Universidad Tecnológica Nacional,
Facultad Regional Tucumán,
Grupo de Investigación en Tecnologías
Informáticas Avanzadas
jgotay@gitia.org

Matias Roodschild

Universidad Tecnológica Nacional,
Facultad Regional Tucumán,
Grupo de Investigación en Tecnologías
Informáticas Avanzadas
matias.roodschild@gitia.org

Adrián Luis Ernesto-Will

Universidad Nacional de Tucumán,
Facultad de Ciencias Exactas y Tecnología,
Facultad Regional Tucumán,
Laboratorio de Robótica e Inteligencia Artificial,
Grupo de Investigación en Tecnologías
Informáticas Avanzadas
adrian.will@gitia.org

Sebastián Rodríguez

Universidad Tecnológica Nacional,
Facultad Regional Tucumán,
Grupo de Investigación en Tecnologías
Informáticas Avanzadas
sebastian.rodriguez@gitia.org

Resumen

Las redes neuronales artificiales *feedforward* y multicapa (RNA-MFF) han demostrado ser de gran alcance en la aproximación de funciones; sin embargo, su aplicación en problemas reales a menudo se limita a la experimentación del usuario, ya que la elección de arquitectura adecuada es un proceso que requiere conocimiento y experiencia. En este artículo se demuestra la capacidad de adaptación de la metodología del algoritmo de poda para encontrar el número óptimo de neuronas en la capa oculta de una RNA-MFF. La metodología se probó en dos conjuntos diferentes de problemas de referencia: modelación de la resistencia y del asentamiento, en hormigón de alto desempeño. Ambos conjuntos se utilizaron para analizar el tamaño de la arquitectura inicial de la red neuronal artificial y para asegurar que el número superior propuesto de neuronas ocultas se pueda evitar en exceso.

Palabras clave: hormigón de alto desempeño, inteligencia artificial, redes neuronales artificiales, optimización en informática, algoritmo de poda.

Abstract

Feedforward and multi-layer artificial neural networks (RNA-MFF) have been shown to be powerful in the approximation of functions; however, its application to real problems is often limited to user experimentation, since choosing the right architecture is a process that requires knowledge and experience. This article demonstrates the adaptability of the pruning algorithm methodology in finding the optimal number of neurons in the hidden layer of an RNA-MFF. The methodology was tested on two different sets of reference problems: resistance and settling modeling, in high performance concrete. Both sets were used to analyze the initial architecture size of the artificial neural network and to ensure that the proposed higher number of hidden neurons can be avoided in excess.

Keywords: high performance concrete, artificial intelligence, artificial neural networks, computer optimization, pruning algorithm.

Resumo

As redes neurais artificiais avançadas e multicamadas *feedforward* (RNA-MFF) têm demonstrado ter grande alcance na aproximação de funções; porém, a aplicação em problemas reais geralmente é limitada à experimentação do usuário, dado que escolher a arquitetura adequada é um processo que precisa conhecimento e experiência. Neste artigo foi demonstrada a adaptabilidade da metodologia do algoritmo de poda para encontrar o número ótimo de neurônios na camada oculta de uma RNA-MFF. A metodologia foi testada em dois conjuntos diferentes de problemas de referência: modelagem de resistência e sedimentação, em concreto de alto desempenho. Ambos os conjuntos foram utilizados para analisar o tamanho da arquitetura inicial da rede neural artificial e garantir que o maior número de neurônios ocultos propostos possa ser evitado em excesso.

Palavras-chave: concreto de alto desempenho, inteligência artificial, redes neurais artificiais, otimização de computadores, algoritmo de poda.

I. Introducción

Las redes neuronales artificiales (RNA) son una excelente herramienta que permite desarrollar expresiones cuantitativas sin un conocimiento *a priori* del comportamiento o la complejidad del problema. El modelo matemático que describe la relación entre las variables en el conjunto de datos y las variables de salida puede no ser conocido o difícil de aplicar o evaluar, y las RNA son capaces de funcionar correctamente siempre y cuando se dispongan de datos suficientes. La relación funcional se puede encontrar durante el proceso de entrenamiento únicamente a partir de ejemplos; esto las hace ideales para su aplicación en diversos problemas prácticos, especialmente en la industria del hormigón (González *et al.*, 2014).

El hormigón hoy en día es formulado con un incremento sofisticado de arreglos de aditivos, adiciones y refuerzos, incluyendo polímeros, fibras metálicas y sintéticas, y puzolanas (González *et al.*, 2012). Estos adicionales son claves para determinar la resistencia a la compresión y el asentamiento —como especificaciones principales de diseño—, y otras propiedades características mecánicas como el control del asentamiento y el tiempo de curado, (González, 2014). Al mismo tiempo, la mezcla de hormigones de resistencia normal está perfectamente estandarizada (Sánchez, 2000), pero para hormigones de alto desempeño es más complicado de determinar, lo cual es objeto de una investigación muy activa (Constantiner y Dackzo, 2002; Bentz, Peltz y Winpigler, 2009; Bhadeshia *et al.*, 2009).

La topología de red neuronal artificial *feed-forward* multicapa (RNA-MFF) es la más popular, debido a la flexibilidad en los buenos resultados que entrega y a su robustez ante la presencia de ruido. En esta red hay disponibilidad de muchos algoritmos de entrenamiento y

son aproximadores universales; más aún, son capaces de proporcionar buenas respuestas en presencia de datos complejos y ruidosos.

Al respecto, diferentes variaciones de las RNA se han utilizado con éxito (MacKay, 1992). Redes neuronales que incorporan elementos de sesgo, conformadas por una capa oculta sigmoidea y una capa de salida lineal, son capaces de aproximar cualquier función con un número finito de discontinuidades a cualquier precisión (Cybenko, 1989; Hornik, Stinchcombe y White, 1989); sin embargo, se requiere de un número grande pero finito de neuronas ocultas. Sin embargo, si el tamaño de la capa oculta o de la existentes es demasiado grande, se puede generar el problema de que la red neuronal sea insuficientemente caracterizada, es decir, la red neuronal debe optimizar más parámetros que vectores de datos existentes para restringir estos parámetros.

El resultado teórico que se menciona con anterioridad no dice nada sobre el número óptimo de neuronas en la capa oculta, o cuántas capas ocultas y sus respectivas neuronas son necesarias para un problema en particular. El número óptimo de neuronas ocultas/capas podría depender de los tamaños de los vectores de entrada/salida y de los tamaños de los conjuntos de datos de entrenamiento y prueba, con una gran influencia de la complejidad del problema.

La elección de la adecuada arquitectura de la red neuronal para un problema dado (clasificación o regresión) sigue siendo un tema difícil, y a pesar de la existencia de diversas propuestas para resolverlo, no hay un acuerdo sobre la estrategia que se debe seguir con el fin de seleccionar la arquitectura adecuada de la red neuronal (Koza y Rice, 1991; Reed, 1993; Huang y Shurong, 2009; Sharma y Chandra, 2010a).

Varios investigadores han intentado diseñar RNA automáticamente; algunos desarrollos muestran: a) una técnica de búsqueda binaria y secuencial con el fin de encontrar el mejor número de neuronas ocultas (Doukim, Darghamy y Chekima, 2010); b) un método para estimar el tamaño de la capa oculta basado en algoritmos de árbol de decisión (Yuan, Xiong y Huai, 2003); c) una metodología para analizar el comportamiento de las RNA-MFF (Panchal *et al.*, 2011).

Una colección de un grupo de técnicas en su estructura, que está adaptado durante el proceso de entrenamiento para un problema dado, constituye, entonces, el marco de la estructura de adaptación de las RNA. Este enfoque se puede clasificar en dos grandes categorías: los *algoritmos constructivos* (Knowk y Yeung, 1997; Sharma y Chandra, 2010a, 2010b; Subirats, Franco y Jerez, 2012) y los *algoritmos de poda* (Reed, 1993; Hashemi-Nejad y Fekri, 2009); ambos tipos de algoritmos pueden abordarse mediante técnicas evolutivas o no evolutivas.

Se han propuesto muchos algoritmos evolutivos que permiten que la arquitectura de la RNA evolucione junto con sus pesos sinápticos mediante técnicas de optimización global. Un ejemplo de estas son los algoritmos genéticos, la programación genética y las estrategias evolutivas (Koza y Rice, 1991; Yao y Liu, 1997). Los métodos de búsqueda globales como optimización de colonia de hormigas y optimización de enjambre de partículas han sido ampliamente utilizados para determinar la arquitectura óptima durante el aprendizaje (Wei, 2008; Huang y Shurong, 2009; Kiranyaz *et al.*, 2009). Sin embargo, el enfoque evolutivo es bastante exigente en tiempo y parámetros, que generalmente deben ser definidos por el usuario (Kwok y Yeung, 1997).

Las RNA con estructuras adaptativas no evolutivas no requieren la definición de su arquitectura antes de comenzar el entrenamiento, mientras que la estructura adaptativa le permite construir su arquitectura junto con el proceso de entrenamiento. Un algoritmo constructivo añade capas ocultas, nodos y conexiones a una RNA con arquitectura minimalista durante el entrenamiento; mientras que un algoritmo de poda hace lo contrario, es decir, borra las capas ocultas, los nodos y las conexiones redundantes a partir de una base de RNA más grande durante el entrenamiento. Un algoritmo constructivo de poda es un enfoque híbrido en el que la RNA puede ser podada después de finalizado el proceso constructivo o se intercale con el proceso constructivo.

Un problema crítico que afecta los niveles de casi todos los algoritmos de aprendizaje de predicción, y en particular de los algoritmos de poda para RNA, es el problema de sobreajuste (Reed, 1993; Castellano, Fanelli y Pelli, 1997). Este problema se produce cuando, durante el entrenamiento de la RNA, el error en el conjunto de entrenamiento disminuye y aumenta el error en ejemplos no visibles. Hay básicamente dos maneras de atacar el sobreajuste: reducir el número de dimensiones en el espacio de parámetros o reducir el tamaño efectivo de cada dimensión; en el primero aparece la técnica de poda, mientras que el segundo corresponde a la técnica de interrupción temprana (Prechelt, 1998).

En este artículo se reporta la elaboración de una metodología basada en el algoritmo de poda para la resolución de la arquitectura de una RNA adaptativa. La RNA predice la resistencia para diferentes tipos de hormigones de alto desempeño; el problema de sobreajuste se analiza, y la estrategia propuesta en el algoritmo de poda no permite que el sobreajuste

aparezca en el proceso de entrenamiento final. Este enfoque utiliza el conjunto de validación para anticipar el comportamiento en uso real (o en un conjunto de prueba), suponiendo que el error será similar: el error de validación se utiliza como una estimación del error de generalización. El algoritmo de poda se aplica con éxito para encontrar el número suficiente de neuronas ocultas en todos los tamaños de las arquitecturas presentadas.

II. Marco teórico

A. Bases teóricas del algoritmo de poda

La RNA-MFF es un aproximador universal donde se combinan una función de transferencia lineal en la capa de salida y las funciones de transferencia sigmoideal en las capas ocultas. Pero si tanto la capa de salida como las capas ocultas tienen funciones lineales, la aproximación a cualquier función no lineal no será adecuada; esto se debe al hecho de que la composición de las funciones lineales es, una vez más, una función lineal. El resultado anterior contiene la base teórica del algoritmo de poda.

Se asume, por simplicidad, que se tiene una RNA-MFF con k neuronas tangentes hiperbólicas y s neuronas lineales en su única capa oculta, además de una neurona lineal en la capa de salida. Entonces, se tiene que:

$$\begin{bmatrix} a_L \\ a_{NL} \end{bmatrix} = \begin{bmatrix} I W_L u + I b_L \\ \tanh(I W_{NL} u + I b_L) \end{bmatrix} \quad [1]$$

Donde a_L es la parte lineal, a_{NL} es la parte no lineal, $u \in \mathfrak{R}^n$ es el vector de entrada, $I W_L$ es una matriz de pesos de entrada ($s \times n$), $I W_{NL}$ es una matriz de pesos de entrada ($k \times n$) e $I b_L$ es un vector de sesgo de entrada de la capa oculta ($k \times 1$).

Se denomina como LW y b a la matriz de pesos de la capa de entrada y al vector de sesgos, respectivamente. Se divide LW con tamaño $1 \times (a+k)$ dentro de la parte lineal LW_L , de tamaño $s \times n$, y la parte no lineal LW_{NL} , de tamaño $1 \times k$. Entonces, la red neuronal de salida y puede escribirse como:

$$y = LW_{LaL} + LW_{NL aNL} + b \quad [2]$$

$$y = LW_L I W_L u + LW_L I b_L + \dots$$

$$\dots + LW_{NL} \tanh(I W_{NL} u + I b_{NL}) + b \quad [3]$$

Sin embargo, otra red con una sola capa oculta mezclada, pero que ahora contiene una única neurona lineal, y el mismo número de neuronas no lineales, con una sola neurona lineal como capa de salida, pueden producir la misma salida que la red original de [3] para el mismo vector de salida P . Entonces, se tiene:

$$y = \overline{TW} u + LW_{NL} \tanh(I W_{NL} u + I b_{NL}) + \overline{b} \quad [4]$$

Donde:

$$\overline{TW} = LW_L I W_L \quad [5]$$

$$\overline{b} = LW_L I b_L + b \quad [6]$$

Esto muestra que es suficiente para mantener a lo sumo una neurona lineal en la capa oculta, con lo cual se justifica el método.

B. Características fundamentales del algoritmo de poda

Hay cinco cuestiones principales relacionadas en los algoritmos constructivos para tareas de regresión, identificadas por Sharma y Chandra (2010a), que se reformulan para los algoritmos de poda. Estas temáticas son las siguientes:

a) la elección de la arquitectura inicial y la estrategia de poda de la red: ¿cuál es el tamaño apropiado de la red inicial? b) la elección de la función de transferencia: ¿qué función de activación debería ser usada en los nodos ocultos y de salida?; c) la elección de los pesos de congelación: ¿reiniciar los pesos de la red o empezar con los valores obtenidos después de la poda?; d) la elección de la técnica de optimización: ¿qué método de optimización debería ser usado para determinar los pesos de optimización durante el entrenamiento?; e) los criterios de parada del entrenamiento: ¿cuándo parar la poda de los nodos ocultos o, en otras palabras, cuál es el número de nodos ocultos para ser instalado en la red?

La capacidad de generalización y el tiempo de entrenamiento de los algoritmos de poda para las tareas de regresión dependen de cada una de las elecciones que se discuten a continuación.

1. La elección de la función de transferencia

Usualmente, la función sigmoidea o la función hiperbólica aparecen como funciones de activación de las redes neuronales si se requieren salidas positivas o salidas positivas y negativas, respectivamente; pero la razón principal por la que se utilizan es porque estas funciones son diferenciables, lo que es esencial para el proceso de propagación hacia atrás. Los algoritmos constructivos con una función sigmoidal con pendiente de adaptación fueron desarrollados primero por Sharma y Chandra (2010b, 2010c), con base en la observación de que una función paramétrica puede adaptarse mejor a la complejidad del problema.

De manera similar, la función hiperbólica fue parametrizada por Hashemi-Nejad y Fekri (2009), con el objetivo de mejorar la capacidad de adaptación de la función de activación.

Esto se realiza mediante la introducción de un nuevo parámetro p , que sensibiliza a la red con respecto a la complejidad del sistema bajo investigación y, aún más, a la utilización real que la red hace de cada neurona en la red. Esta función se puede definir como:

$$f_p(x) = \tanh(x) \rightarrow f_p(x) = \frac{1}{\ln(p)} \tanh(\ln(p)x) \quad [7]$$

Donde $x \in \mathfrak{R}$. Al desarrollarse la función parametrizada de [7] en series de Taylor, se observa que:

$$\lim_{p \rightarrow 1} f_p(x) = x \quad [8]$$

La función de activación tangente hiperbólica, comúnmente usada en las redes neuronales, no tiene la capacidad de cambiar su forma; sin embargo, mediante la introducción de un nuevo parámetro p de la función de activación, esta mejorará desde un trazado lineal hasta una tangente hiperbólica pura; de esta manera se sensibiliza la red con respecto a la complejidad del sistema bajo investigación. La figura 1 muestra la gráfica de la función de transferencia para diferentes valores p .

2. La elección de la arquitectura inicial

El resultado teórico presentado por Cybenko (1989) y Bhadeshia *et al.* (2009), sobre la RNA-MFF y su capacidad de aproximación universal, no menciona nada acerca de exactamente cuántas neuronas son suficientes o cuál es el número óptimo de neuronas ocultas que se instalarán en la red. Las neuronas ocultas en exceso causarán sobregeneralización, es decir, las redes neuronales sobreestiman la complejidad del problema objetivo (Jinchuan y Xinzhe, 2008). La selección de la arquitectura inicial se realiza de modo que el número de

parámetros no exceda el número de patrones de entrenamiento. Las redes paramétricas con una sola capa oculta calculan funciones del siguiente tipo:

$$y(x, u) = \sum_{i=1}^S LW_i f_p(IW_i u + b_i) + LW_0 \quad [9]$$

Donde f_p es la función parametrizada entregada en [7]; las entradas u son vectores en \mathfrak{R}^n ; el entero S representa el número de neuronas ocultas, que será fijado en [10]; los parámetros $S(n+3)+1$ de la red (resumido por el vector x), a saber: los escalares LW_0, LW_1, \dots, LW_S y b_1, b_2, \dots, b_S , y los n -vectores fila IW_1, IW_2, \dots, IW_S , son considerados como las variables que se pueden sintonizar con el fin de hacer que $y(x, u) \approx y_i$, cuando se da el conjunto de entradas y sus correspondientes salidas objetivo. Se fija, entonces, un límite superior de número de neuronas por:

$$S < \frac{Ndata - 1}{n + 3} \quad [10]$$

Donde $Ndata$ es el número de patrones de entrenamiento.

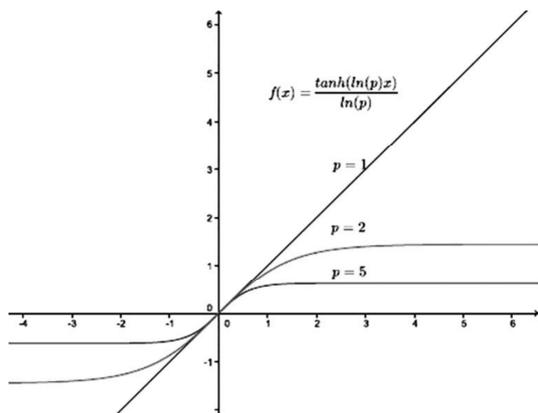


Figura 1. Tangente hiperbólica paramétrica para $p = \{1, 2, 5\}$

3. Criterio de parada

El algoritmo comienza con una arquitectura de la red que satisface a [10]; el parámetro p será entrenado usando el mismo algoritmo como los pesos, con las diferencias impuestas por la derivada de la función de transferencia paramétrica en el algoritmo de propagación hacia atrás. Durante el entrenamiento, el parámetro de la red p se comprueba de manera que permanezca dentro de un rango especificado. Si más de una neurona tiene un parámetro cercano a uno, entonces pueden ser representados por una sola neurona. El algoritmo se inicia de nuevo, pero con una arquitectura más pequeña. El procedimiento de poda se detiene cuando el error en el conjunto de validación comienza a aumentar cada vez más.

III. Materiales y Métodos

A. Procedimiento

Se analiza el desempeño del algoritmo de poda mediante simulaciones numéricas, utilizando dos conjuntos diferentes de datos del “mundo real” sobre hormigón de alto desempeño con dos problemas de referencia. El primero está relacionado con la resistencia a la compresión del hormigón, que es una función altamente no lineal de la edad (desde la preparación de la mezcla) y de su materia prima. El conjunto de datos incluye 1030 diseños de mezcla reales relacionados con nueve atributos. Hay ocho variables de entrada y una variable de salida; este conjunto de datos tiene números reales y ningún valor falta.

El segundo conjunto está relacionado con la prueba de asentamiento del cono de Abrams; el flujo de asentamiento del hormigón no solo está determinado por el contenido de agua, sino que también está influenciado por otros

ingredientes de la mezcla de hormigón. El conjunto de datos incluye 103 diseños de mezcla y diez atributos. Hay siete variables de entrada y tres variables de salida (asentamiento, flujo y resistencia de diseño a la compresión). Sin embargo, solamente la variable del asentamiento fue utilizada en el conjunto para fines de entrenamiento y aplicación del algoritmo de poda. Ambas bases de datos están reportadas en el repositorio UCI (Asunción y Newman, 2007) y en la literatura (Yeh, 1998, 2007).

B. Primera aproximación

Para evaluar la relación entre el tamaño del vector de entrada, el número de ejemplos de entrenamiento y el tamaño de la arquitectura, se divide aleatoriamente cada conjunto de datos en dos partes disjuntas: datos de entrenamiento y datos de prueba. Los datos de entrenamiento se subdividen a su vez de manera aleatoria en un conjunto de ejemplos de entrenamiento utilizados para ajustar los pesos de la red y en un conjunto de ejemplos de validación utilizados para estimar el desempeño de la red durante el entrenamiento. El método de parada temprana se emplea para mejorar la generalización de la red (Prechelt, 1998). Los datos se dividen en un conjunto de entrenamiento (60%), un conjunto de validación (20%) y un conjunto de prueba (20%); varias simulaciones se realizaron en ambas bases de datos. Con la base de datos relacionada con la resistencia a la compresión, el número de neuronas ocultas se fijaron como:

$$N = \{1,2,3,4,5,6,7,8,9,10,15,20,30,40 \dots \\ \dots 50,56,60,80,90,100\} \quad [11]$$

Con la base de datos relacionada con el asentamiento del hormigón, el número de neuronas ocultas se fijaron como:

$$N = \{1,2,3,4,5,6,7,8,9,10\} \quad [12]$$

El proceso se repitió 1000 veces para cada una de las arquitecturas. El promedio del error para cada uno de los conjuntos de entrenamiento, validación y prueba se utilizaron respectivamente.

C. Segunda aproximación

Esta aproximación consiste en aplicar el algoritmo de poda para la estimación de la resistencia a la compresión y del asentamiento del cono de Abrams en hormigones. En el primer problema se utiliza una arquitectura inicial de 56 neuronas en la capa oculta, mientras que en el segundo problema se usa una arquitectura inicial de siete neuronas ocultas. Para obtener una estimación fiable del desempeño, siempre se prefiere un gran número de estimaciones. Se propone desarrollar una repetición de validación cruzada (*k-fold*) (Díaz, Will y González, 2011). En este estudio, una validación cruzada (*4-fold*) es corrida diez veces, resultando así 40 valores precisos; el proceso de prueba para cuando el error promedio de los conjuntos de validación comienza a aumentar. El umbral de p es de 1,28.

D. Criterio de desempeño

En este estudio, el error medio absoluto medido en porcentaje (MAPE) y la raíz del error medio cuadrático (RSME) se utilizaron como criterios de desempeño (Yeh, 2007; Chaouachi *et al.*, 2009). En el caso de la resistencia a la compresión del hormigón, el MAPE se utilizó como se define en [13]. Dada la existencia de algunos valores iguales a cero, en algunos valores del asentamiento el RMSE se usó como criterio de desempeño del algoritmo de poda, en el caso del asentamiento del cono de Abrams.

$$MAPE = \frac{1}{n} \sum_{i=1}^n |T_i - P_i| * 100 \quad [13]$$

Donde T_i es el valor reportado, P_i es el valor estimado y n es el número de ejemplos.

IV. Resultados y discusión

De acuerdo con la figura 2, en la primera aproximación aplicada al problema relacionado con la resistencia a la compresión del hormigón, el error de entrenamiento disminuye rápidamente con el aumento del tamaño de las arquitecturas; además, el error de entrenamiento más bajo comienza a mantenerlo para arquitecturas mayores de 90 neuronas. Esta figura muestra tres áreas diferentes de desempeño de las redes neuronales que se discuten a continuación.

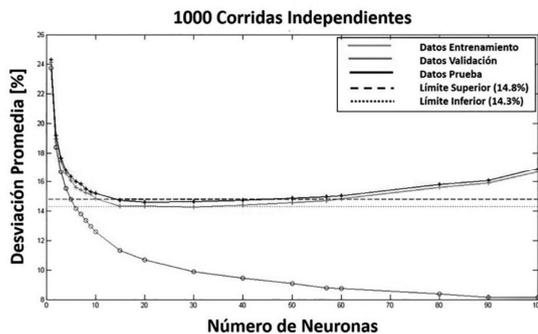


Figura 2. Promedio de los errores (entrenamiento, validación y prueba) versus número de neuronas en el problema relacionado con la resistencia a la compresión

En la primera de ellas existe más de una arquitectura donde las redes hacen su trabajo y generalización con un desempeño muy bueno (arquitecturas con tamaños entre 10 y 56 neuronas). Se debe notar que la arquitectura con 56 neuronas satisface la condición de [10]. En esta parte, el MAPE promedio del error de validación no excede el 0,56%.

En la segunda área se puede distinguir que, en las arquitecturas con tamaños menores a nueve neuronas, el MAPE promedio del error de entrenamiento y del error de validación es mayor al 13%. Esto es un indicativo de que se necesitan más neuronas para explicar la complejidad de la función y se tiene un bajo ajuste de los datos. Respecto a las otras arquitecturas con tamaños mayores a 56 neuronas, vale señalar que mientras las redes parecen ser cada vez mejores —es decir, el error en los conjuntos de entrenamiento disminuye durante el entrenamiento—, ellas empiezan a empeorar en cuanto el error de los conjuntos de validación aumentan (tercera zona). Se usó la técnica de detención temprana durante el entrenamiento, por lo que el sobreajuste ocurre debido a que el número de parámetros que deben optimizarse es más grande que el número de ejemplos para el entrenamiento.

La figura 3 muestra que para la primera aproximación aplicada al problema relacionado con el asentamiento del hormigón, el error de entrenamiento decrece rápidamente con el incremento del tamaño de las arquitecturas. El error del conjunto de validación se incrementa significativamente a partir de siete neuronas.

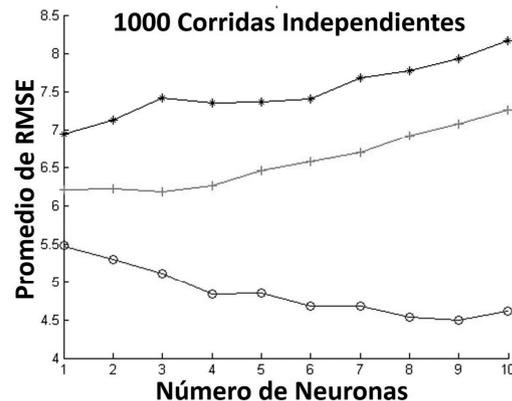


Figura 3. Promedio de los errores (entrenamiento, validación y prueba) versus número de neuronas en el problema relacionado con el asentamiento del hormigón

En la segunda aproximación, el algoritmo de poda para ambos problemas siempre alcanzó la arquitectura adecuada cuando el entrenamiento comienza a partir de la arquitectura inicial entregada por [10], donde se puede asegurar que no hay más parámetros que patrones de entrenamiento. En el problema relacionado con la resistencia a la compresión del hormigón, el algoritmo de poda comienza con 56 neuronas y se detiene en una arquitectura con nueve neuronas, requiriéndose 14.000 épocas. El promedio de los valores del RMSE de los conjuntos de entrenamiento y validación fueron 5,1091 y 6,7204, respectivamente. Una arquitectura con ocho neuronas fue encontrada por Yeh (1998), pero con tamaño de

muestra de 727 vectores de datos (algunos datos fueron descargados en el trabajo original durante la detención temprana de limpieza de información).

Con respecto a la prueba de asentamiento del hormigón, el algoritmo de poda comienza con siete neuronas y se detiene en una arquitectura con tres neuronas, y 4000 épocas fueron usadas. El promedio de los valores del RMSE de los conjuntos de entrenamiento y validación fueron 5,2032 y 6,7653, respectivamente. La tasa de poda depende ampliamente del umbral del parámetro p , el cual varió entre 1,25 y 1,8. La figura 4 muestra el análisis de regresión para el mejor conjunto de entrenamiento.

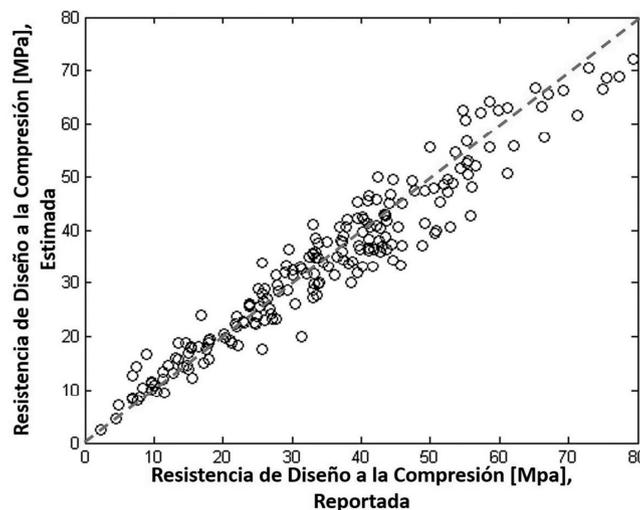


Figura 4. Correlación entre los valores de resistencia a la compresión del hormigón, reportados y estimados por la red neuronal, para el mejor conjunto de entrenamiento

El análisis de regresión muestra cómo para hormigones hasta resistencias de 30 MPa (hormigones de resistencia normal) hay un ajuste con menor dispersión sobre la mejor recta; esta dispersión es mayor para hormigones con mayor resistencia, característica de los hormigones de alto desempeño, lo cual demuestra la complejidad en la predicción de la propiedad mecánica en cuestión,

en particular para estos tipos de hormigones (González, 2014).

Sin embargo, para ambos casos, así como para todos los pares de valores correlacionados (reportados versus estimados), se aprecia gráficamente una bondad en el ajuste, en razón a una relación lineal positiva fuerte entre los dos valores. Así es referido por diversos

investigadores sobre correlaciones estadísticas, como Ascombe (1973), Aschen (1982), Tompkins (1992) y Martínez (2005).

V. Conclusiones

En este trabajo se reportó la elaboración de la arquitectura de una red neuronal artificial, usando una metodología basada en el algoritmo de poda. La red neuronal fue aplicada para predecir la resistencia a la compresión y el asentamiento del hormigón. El algoritmo de poda fue exitoso para encontrar el número apropiado de neuronas ocultas en las arquitecturas presentadas. La selección de la arquitectura inicial entregada por [10] ayudó a superar problemas de sobreajuste. El número de épocas necesario para aplicar el algoritmo de poda es alto (como en el caso de la estimación de la resistencia a la compresión, con un valor de 14.000 épocas), de modo que es importante desarrollar métodos para acelerar esta clase de algoritmos, con lo cual se abre una agenda futura.

Por las razones expuestas, la metodología que usa el algoritmo de poda es una herramienta útil para los usuarios sin experiencia en el uso de redes neuronales artificiales, y en particular para los investigadores que trabajan en el campo del hormigón de alto desempeño.

Referencias

Aschen, C. (1982). *Interpreting and using regression*. Newbury Park: Sage University Paper.

Ascombe, T. (1973). Graphs and Statistical Analysis. *The American Statistician*, 27(1), 17-21.

Asuncion, A. y Newman, D. (2007). UCI Machine Learning Repository. Recuperado de <http://www.ics.uci.edu/~mllearn/MLRepository.html>

Bentz, D., Peltz, M. y Winpighler, J. (2009). Early-age properties of cement-based materials: II. Influence of water-to-cement ratio. *ASCE Journal of Materials in Civil Engineering*, 21(9), 512-517.

Bhadeshia, H., Dimitriu, R., Forsik, S., Pak, J. y Ryu, J. (2009). Performance of neural networks in materials science. *Materials Science and Technology*, 25(4), 504-510.

Castellano, G., Fanelli, A. y Pelillo, P. (1997). An iterative pruning algorithm for feedforward neural networks. *IEEE Transactions on Neural Networks*, 8(3), 519-531.

Chaouachi, A., Kamel, R., Ichikawa, R., Hayashi, H. y Nagasaka, K. (2009). Neural network ensemble-based solar power generation short-term forecasting. *International Journal of Engineering and Mathematical Sciences*, 3(6), 40-46.

Constantiner, D. y Dackzo, J. (2002). Not all applications are created equal: Selecting the appropriate SCC performance targets. *First North American Conference on the Design and Use of Self Consolidating Concrete*. Chicago.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Sigma and Systems*, 2(4), 303-314.

Díaz, W., Will, A. y González, S. (2011). Un método para estimación de calidad de datos en redes neuronales y aplicación a problemas de hormigón de alto performance. *Mecánica Computacional*, 30(43), 3381-3381.

Doukim, C., Dargham, J. y Chekima, A. (2010). Finding the number of hidden neurons for a MLP neural networks using coarse to fine search technique. Recuperado de <https://www.semanticscholar.org/paper/Finding-the->

- number-of-hidden-neurons-for-an-MLP-ne-Doukim-Dargham/c442c3ac2589e1f048d-5d5827556763c5b7fbdfa
- González, S., Guerrero, Z., Delvasto, A. y Will, A. (2012). Red neuronal Artificial para estimar la resistencia a compresión, en concretos fibroreforzados con acero. *Ciencia e Ingeniería Neogranadina*, 22(1), 19-41.
- González, S. (2014). *Diseño de mezclas de concreto reforzado con fibras metálicas y de polipropileno, usando inteligencia artificial* (tesis doctoral). Cali: Universidad del Valle.
- González, S., Guerrero, Z., Delvasto, A. y Will, A. (2014). Estimación del Índice de Tenacidad Flexural I_5 en concretos fibroreforzados, usando redes neuronales artificiales. *Revista Colombiana de Materiales*, 5, 24-29.
- Hashemi-Nejad, M. y Fekri, H. (2009). Short-term wind speed prediction using adaptive neural networks. *Europe Wind Energy Conference and Exhibition 2009*. Marsella, Francia.
- Hornik, K., Stinchcombe, M. y White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359-366.
- Huang, R. y Shurong, T. (2009). Evolving product unit neural networks with particle swarm optimization. *Proceedings of the 2009 Fifth International Conference on Image and Graphics*.
- Jinchuan, K. y Xinzhe, L. (2008). Empirical analysis of optimal hidden neurons in neural networks modeling for stock prediction. *Proceedings of the Pacific-Asia Workshop on Computational Intelligence and Industrial Applications*, 40, 828-832.
- Kiranyaz, S., Ince, T., Yildirim, A. y Gabbouj, M. (2009). Evolutionary artificial neural networks by multi-dimensional particle swarm optimization. *Neural Networks*, 22(10), 1448-1462.
- Koza, J. y Rice, J. (1991). Genetic generation of both the weights and architecture for neural networks. *Proceedings of IEEE, IJCNN*, Seattle (WA), pp. 397-404.
- Kwok, T. y Yeung, D. (1997). Constructive algorithms for structure learning in feedforward neural networks for regression problems. *IEEE Transactions on Neural Networks*, 8(3), 630-645.
- MacKay, D. (1992). A practical bayesian framework for backpropagation networks. *Neural Computation*, 4(3), 448-472.
- Martínez, E. (2005). Errores frecuentes en la interpretación del coeficiente de determinación lineal. *Anuario Jurídico y Económico Escurialense*, 38, 315-332.
- Panchal, G., Ganatra, A., Kosta, Y. y Panchal, D. (2011). Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers. *International Journal of Computer Theory and Engineering*, 3(2), 332-337.
- Prechelt, L. (1998). Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks*, 11(4), 761-767.
- Reed, R. (1993). Pruning algorithms-a survey. *Transactions of Neural Networks*, 4(5), 740-747.
- Sánchez, D. (2000). *Tecnología del Concreto y Mortero*. Bogotá: Bhandar.
- Sharma, S. y Chandra, P. (2010a). Constructive neural networks: A review. *International Journal of Engineering Science and Technology*, 2(12), 7847-7855.
- Sharma, S. y Chandra, P. (2010b). An adaptive slope sigmoidal function cascading neural

networks algorithm. *Proceedings of the IEEE, Third International Conference on Emerging Trends in Engineering and Technology*.

Sharma, S. y Chandra, P. (2010c). An adaptive slope basic dynamic node creation algorithm for single hidden layer neural networks. *Proceedings of the IEEE, International Conference on Computational Intelligence and Communication Systems*.

Subirats, J., Franco, L. y Jerez, J. (2012). C-MANTEC: A novel constructive neural network algorithm incorporating competition between neurons. *Neural Networks*, 26, 130-140.

Tompkins, C. (1992). Using and interpreting linear regression and correlation analyses: Some cautions and considerations. *Clinical Aphasiology Paper*, 21, 35-46.

Wei, G. (2008). Evolutionary neural network based on new ant colony algorithm. *International Symposium on Computational Intelligence and Design IEEE*.

Yao, X. y Liu, Y. (1997). A new revolutionary system for evolving artificial neural networks. *Transactions on Neural Networks*, 8(3), 694-713.

Yeh, I. (1998). Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete Research*, 28(12), 1797-1808.

Yeh, I. (2007). Modeling slump flow of concrete using second-order regressions and artificial neural networks. *Cement and Concrete Composites*, 29(6), 474-480.

Yuan, H., Xiong, F. y Huai, X. (2003). A method for estimating the number of hidden neurons in feed-forward neural networks based on information entropy. *Computers and Electronics in Agriculture*, 40(1-3), 57-64.