

# CO<sup>2</sup>RBFN: predicción de series temporales con un enfoque cooperativo-competitivo

María Dolores  
Pérez-Godoy

Pedro Pérez-  
Recuerda

Antonio J. Rivera

María José del  
Jesus

Departamento de Informática  
Universidad de Jaén  
lperez, pedro.perez, arivera, mjjesus@ujaen.es

## Resumen

En este trabajo se presenta la aplicación de un algoritmo bioinspirado para el diseño de Redes de Funciones de Base Radial (RBFNs), CO<sup>2</sup>RBFN, a la serie hidrográfica propuesta en la competición de los simposios SICO2010 y TAMIDA2010. El método de aprendizaje está basado en la programación evolutiva con un enfoque cooperativo-competitivo en el que cada individuo representa una neurona y la población al completo la red. En dicho enfoque los individuos cooperan para lograr la solución final y también compiten por su supervivencia. El modelo utiliza un Sistema Basado en Reglas Difusas (SBRD), que representa conocimiento experto, para decidir la aplicación de los operadores a los individuos.

## 1. Introducción

Una serie temporal es una colección de observaciones de una variable realizadas de forma secuencial en el tiempo, en las que el orden de observación es importante. Los valores de una serie temporal van ligados a instantes de tiempo, de manera que el análisis de una serie implica el manejo conjunto de dos variables; la variable en estudio propiamente dicha y la variable tiempo.

Dentro del campo de la minería de datos son muchos los métodos que se utilizan para predicción de series temporales. Así por ejemplo es bien conocida la capacidad de predicción de métodos tales como Redes Neuronales o Sistemas Basados en Reglas Difusas [4][9][20][25]. Normalmente la capacidad de predicción de los métodos de minería de datos está muy relacionada

con su característica de aproximadores universales.

Las Redes de Funciones de Base Radial (RBFNs) constituyen uno de los paradigmas más populares dentro del campo de las redes neuronales. Este modelo de cómputo ha demostrado su solvencia a la hora de abordar problemas como aproximación de funciones [8], clasificación [3] o predicción de series temporales [26]. Las Funciones de Base Radial (RBFs) se usaron inicialmente en interpolación numérica y aproximación funcional [22]. A finales de los 80 tienen lugar las primeras investigaciones de redes neuronales basadas en RBFs [2] [12].

Son muchas las características que despiertan el interés por este tipo de redes entre las cuales cabe destacar: su topología simple con tan solo una capa oculta, su capacidad de ser un aproximador universal [14] o la analogía entre la salida de estas redes y los campos receptivos localizados, encontrados en estructuras biológicas cerebrales.

Las técnicas empleadas en el diseño de RBFNs son muy diversas. El algoritmo típico de diseño de RBFNs tiene dos etapas. En la primera etapa se determinan los centros y los radios de las RBFs, mientras que en la segunda etapa se calculan sus pesos. Para determinar los centros y los radios se pueden emplear técnicas de clustering [15]. En la segunda etapa y para calcular los pesos se pueden utilizar algoritmos como Least Mean Square (LMS) [27], Singular Value Decomposition (SVD) [7], etc.

El diseño de una RBFN también se puede abordar desde el punto de vista de la computación evolutiva [1]. En la mayoría de estas propuestas evolutivas, un individuo representa una RBFN completa. De esta manera, los operadores

evolutivos actúan sobre los individuos añadiendo RBFs, eliminándolas o modificándolas.

Sin embargo y según el trabajo de Potter [21] la computación evolutiva tradicional presenta ciertos problemas relacionados con la evaluación de subcomponentes independientes en el diseño de ciertos modelos. Así se propone la coevolución cooperativa para extender el modelo evolutivo tradicional y conseguir un entorno de diseño en el que los individuos de la población representen una parte de la solución evolucionando en paralelo. Ahora los individuos no sólo compiten por sobrevivir sino que deben cooperar para alcanzar una solución.

Los autores ya han desarrollado un importante trabajo en diseño híbrido de RBFNs [17][23]. En el trabajo actual se usa un marco de evolución cooperativo-competitivo, definiéndose unas medidas para la asignación de crédito basada en tres parámetros, operadores evolutivos que consigan un buen equilibrio entre exploración y explotación, y un sistema difuso de base de reglas que decidirá la aplicación de estos operadores.

El algoritmo CO<sup>2</sup>RBFN se aplicará a la serie hidrológica propuesta en la competición de series temporales por los simposios SICO2010 y TAMIDA2010.

La organización del resto de este trabajo se describe a continuación. En la sección 2 se comentan generalidades sobre las series temporales. En la sección 3 se introducen las RBFNs y se explica cómo se aplican a problemas de predicción de series temporales. En la sección 4 se presenta el algoritmo propuesto y sus resultados se muestran en la sección 5. Por último se explican las conclusiones y las líneas de trabajo futuro en la sección 6.

## 2. Predicción de series temporales

Las razones que motivan la investigación de las series temporales son muy diversas: la necesidad de predecir el comportamiento de una variable en el futuro, el control de un proceso, aumentar los beneficios de la empresa, anticipar caídas o subidas en el mercado, la simulación de fenómenos que no pueden llevarse a la práctica y que permiten el estudio de los diferentes estados o situaciones que pueden acontecer, la generación de nuevas teorías físicas o biológicas, etc. En general, el objetivo final es siempre aumentar

nuestro conocimiento sobre un fenómeno o aspecto de nuestro entorno partiendo de datos pasados y presentes. Por tanto, se pretende extraer las regularidades que se observan en el comportamiento pasado de la variable, es decir, obtener el mecanismo que la genera, para tener un mejor conocimiento de la misma en el tiempo.

El tipo de análisis, así como los modelos en los que basemos el estudio, dependerán en gran medida del tipo de preguntas que queramos responder. Cuando las observaciones corresponden a una única variable, el análisis de series temporales suele tener como objetivo construir un modelo para explicar la estructura (descripción) y prever la evolución (predicción) de la variable de interés. Por tanto, en el modo descriptivo podemos utilizar los datos para encontrar patrones de comportamiento de los datos, reglas de hechos y asociaciones entre sus ocurrencias. También se buscan anomalías de comportamiento y en general se emplea para caracterizar los datos. En el modo predictivo, que es el que nos ocupa, los datos se analizan para descubrir un modelo de comportamiento futuro de los mismos para poder estimar con antelación, posibles valores y tendencias.

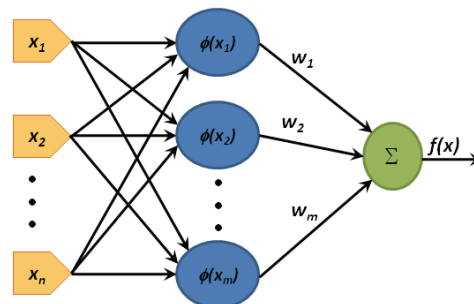


Figura 1. Red de Funciones de Base Radial

## 3. Descripción y diseño de RBFNs

Una Red de Funciones de Base Radial es un tipo de red neuronal hacia delante con tres capas: la capa de entrada con  $n$  nodos, una capa oculta con  $m$  neuronas o RBFs, y una capa de salida que en el caso de predicción presenta sólo un nodo, ver Figura 1. Las  $m$  neuronas/RBFs de la capa oculta

ofrecen una activación simétrica radial  $\phi_i: R^n \rightarrow R$ , que puede tomar diferentes formas aunque la más común es la función gaussiana que viene dada por la expresión:  $\phi_i(\vec{x}) = \phi_i(e^{-\|\vec{x}-\vec{c}_i\|/d_i})^2$ , donde  $\vec{c}_i \in R^n$  es el centro de la función base  $\phi_i$ ,  $d_i \in R$  es el radio, también conocido como  $r_{is}$  y como  $\| \|$  suele utilizarse la norma euclídea en  $R^n$ .

El nodo de salida implementa la siguiente ecuación:

$$f(\vec{x}) = \sum_{i=1}^m w_i \phi_i(\vec{x}) \quad (1)$$

En la bibliografía especializada se pueden encontrar distintos métodos evolutivos [3][10][13] para el diseño de RBFNs. Sin embargo, la mayoría de las aproximaciones existentes trabajan con métodos evolutivos típicos donde un individuo representa una red completa, por lo que pueden tener problemas de alto coste computacional y convergencia prematura a mínimos locales. Como se ha comentado anteriormente estos problemas pueden mitigarse utilizando técnicas cooperativas-competitivas donde un individuo representa una neurona o RBF tal y como se propone en este trabajo. Hasta ahora en la bibliografía no son muchos los trabajos [23][24][26] que implementan este tipo de paradigma, debido sobre todo a la dificultad que entraña definir los parámetros de cooperación y competición entre los individuos.

#### 4. CO<sup>2</sup>RBFN: un algoritmo evolutivo cooperativo-competitivo para el diseño de RBFNs

Se propone un algoritmo evolutivo híbrido con un enfoque cooperativo-competitivo, para la predicción de series temporales. En esta propuesta cada individuo de la población representa una función base y la población entera es la responsable de la solución final. Se presenta un entorno en el que los individuos cooperan para alcanzar la solución definitiva, no obstante, también compiten por la supervivencia, dado que si el trabajo de un individuo no es bueno dicho individuo será eliminado.

Este escenario tiene en cuenta la respuesta local de las neuronas y la interpretabilidad de esta

clase de redes, lo cual establece una importante guía de diseño.

En este entorno de cooperación-competición en el que la solución depende del comportamiento de muchos componentes, el *fitness* de cada individuo es conocido como asignación de crédito. Para medir la asignación de crédito de un individuo, se proponen tres factores que evalúan el papel de cada RBF en la red. Dichos factores tienen en cuenta la aportación de la RBF a la red, el error cometido por la RBF y la posible interacción de una RBF con otras. El algoritmo usa un Sistema Basado en Reglas Difusas (SBRD) para generar la probabilidad de aplicación de los operadores a las RBFs.

Las etapas principales de CO<sup>2</sup>RBFN se explican en las siguientes subsecciones. A continuación se muestra su pseudocódigo:

- 
1. Inicialización de las RBFs
  2. Entrenamiento de las RBFs
  3. Evaluación de las RBFs
  4. Aplicación de los operadores a las RBFs
  5. Sustitución de las RBFs que han sido eliminadas
  6. Selección de las mejores RBFs
  7. Salto al paso 2 si no se verifica la condición de Parada
- 

Algoritmo 1. Principales etapas de CO<sup>2</sup>RBFN

##### 4.1. Inicialización de la Red

El proceso para definir la red inicial es simple. Las neuronas se colocan aleatoriamente sobre muestras del conjunto de entrenamiento. El número de RBFs está especificado como un parámetro (es el tamaño de la población,  $m$ ).

El centro de cada RBF,  $\vec{c}_i$ , se inicializa a partir de un patrón al azar del conjunto de entrenamiento. El radio,  $d_i$ , se inicializa a la mitad de la media de las distancias entre los centros. Finalmente los pesos,  $w_{ij}$ , se ponen a cero.

##### 4.2. Entrenamiento de las RBFs

Durante esta etapa se entrenan los pesos de las RBFs. El propósito de entrenar los pesos es el de explotar la información local extraída del

comportamiento de las RBFs. Dicho entrenamiento se realiza mediante la técnica LMS [27].

#### 4.3. Evaluación de las RBFs

Se requiere un mecanismo de asignación de crédito para poder evaluar el papel de cada función base dentro del entorno cooperativo-competitivo. Para este propósito se consideran tres parámetros,  $a_i$ ,  $e_i$ ,  $s_i$  para cada RBF  $\phi_i$ .

La contribución,  $a_i$ , de la RBF  $\phi_i$ , a la salida de la RBFN, se determina considerando su peso,  $w_i$ , y el número de patrones de entrenamiento dentro de su radio,  $pi_i$ . Se penaliza una RBF con poco peso y pocos patrones dentro de su radio:

$$a_i = \begin{cases} |w_i| & \text{if } pi_i > q \\ |w_i| * (pi_i / q) & \text{en otro caso} \end{cases} \quad (2)$$

donde  $q$  es la media de los valores  $pi_i$  menos su desviación típica.

La medida de error,  $e_i$ , para cada RBF  $\phi_i$ , se obtiene calculando el error Porcentual Absoluto Medio (MAPE) cometido:

$$e_i = \frac{abs\left(\sum_{\forall pi_i} \frac{\text{salidaReal} - \text{salidaModelo}}{\text{salidaModelo}}\right)}{pi_i} \quad (3)$$

El solapamiento de la RBF,  $\phi_i$ , con otras RBFs se cuantifica usando el parámetro  $s_i$ . Este parámetro se calcula partiendo de la metodología fitness sharing [6], que intenta mantener la diversidad en la población. Concretamente:

$$s_i = \sum_{j=1}^m s_{ij} \\ s_{ij} = \begin{cases} \left(1 - \frac{\|\phi_i - \phi_j\|}{r_i}\right) & \text{si } \|\phi_i - \phi_j\| < r_i \\ 0 & \text{en otro caso} \end{cases} \quad (4)$$

donde  $s_{ij}$  mide el solapamiento entre las RBF  $\phi_i$  y  $\phi_j$ ,  $j=1 \dots m$ .

#### 4.4. Aplicación de los operadores a las RBFs

En el algoritmo existen cuatro operadores que se pueden aplicar a las RBFs:

- Operador Elimina: es un operador que elimina una RBF.
- Operador Mutación Aleatoria: es un operador que modifica de forma aleatoria el centro y el radio de la RBF. El radio se modifica con una probabilidad inversamente proporcional al número de características en el problema ( $n$ ). La modificación, en caso de llevarse a cabo, puede oscilar en un porcentaje entre un 5% y un 50% sobre el radio antiguo. En cuanto a la modificación del centro, el número de coordenadas que se van a mutar se obtiene aleatoriamente entre un 1% y un 25 % del total de características del conjunto de entrenamiento. La coordenada a mutar sufre un incremento o decremento en una cantidad aleatoria comprendida entre un 5% y un 50% de valor del radio.
- Operador Mutación Informada: este operador también puede modificar el radio y las coordenadas del centro de la RBF, usando información de su entorno. Las modificaciones del centro y radio siguen las recomendaciones dadas en [5]. Se calcula el error para los ejemplos que están dentro del radio de la RBF  $\phi_i$ . Para cada coordenada del centro y para el radio se calcula una variación  $\Delta c_{ij}$  y  $\Delta d_i$  respectivamente. Las nuevas coordenadas y el nuevo radio se obtienen modificando sus valores antiguos en una cantidad aleatoria, entre un 5% y un 50% del radio actual de la RBF, en función del signo de la variación calculada.

$$\Delta d_i = \sum_k e(\bar{p}_k) * w_i \quad (5)$$

$$\Delta c_{ij} = \text{signo}(c_{ij} - p_{kj}) * e(\bar{p}_k) * w_i \quad (6)$$

donde  $e(\bar{p}_k)$  es el error cometido con el ejemplo  $\bar{p}_k$ .

- Operador Nulo: no se realiza ninguna acción sobre la RBF.

Estos operadores se aplican a la población total de RBFs. La probabilidad de elegir un operador para una RBF dada viene determinada por un sistema difuso tipo Mamdani [11], cuyas entradas son los parámetros  $a_i$ ,  $e_i$  y  $s_i$ , éstos determinan la asignación de crédito de cada RBF.

Estas entradas se consideran como variables lingüísticas  $va_i$ ,  $ve_i$  y  $vs_i$ , y las salidas son  $p_{elimina}$ ,  $p_{ma}$ ,  $p_{mi}$  y  $p_n$ , que representan la probabilidad de aplicar los operadores Elimina, Mutación Aleatoria, Mutación Informada y Nulo, respectivamente. El número de etiquetas lingüísticas se ha determinado empíricamente, con centros y bases directamente relacionados con su significado. Hay tres etiquetas lingüísticas: L (bajo), M (medio), H (alto), para definir cada entrada. Para definir las salidas se consideran cuatro etiquetas lingüísticas: L (baja), M-L (medio-baja), M-H (medio-alta), y H (alta).

La Tabla 1 muestra las reglas usadas, basadas en conocimiento experto, para relacionar los antecedentes y los consecuentes. En la tabla cada fila representa una regla y así por ejemplo la interpretación de la primera regla es: Si la aportación de una RBF es baja Entonces la probabilidad de aplicar el operador elimina es medio-alta, la del operador de mutación aleatoria es medio-alta, la del operador mutación informada es baja y la del operador nulo es baja.

Tabla 1. Base de reglas utilizada

	Antecedentes			Consecuentes			
	$va$	$ve$	$vo$	$P_{elimina}$	$P_{ma}$	$P_{mi}$	$P_{nulo}$
R1	L			M-H	M-H	L	L
R2	M			M-L	M-H	M-L	M-L
R3	H			L	M-H	M-H	M-H
R4		L		L	M-H	M-H	M-H
R5		M		M-L	M-H	M-L	M-L
R6		H		M-H	M-H	L	L
R7			L	L	M-H	M-H	M-H
R8			M	M-L	M-H	M-L	M-L
R9			H	M-H	M-H	L	L

#### 4.5. Introducción de nuevas RBFs

En este punto el algoritmo sustituye las RBFs eliminadas por otras nuevas. Existen dos alternativas para situar las nuevas RBFs, ambas con probabilidad de 0.5: en un patrón dentro la zona donde se esté cometiendo el máximo error y que esté fuera de cualquier RBF o bien, en un patrón obtenido de forma aleatoria que no esté dentro del radio de ninguna RBF.

#### 4.6. Selección de las mejores RBFs

Después de aplicar los operadores de mutación aparecen nuevas RBFs y éstas son comparadas con sus padres para determinar cuáles tienen un mejor comportamiento en la red. Las mejores RBFs serán las elegidas para formar parte de la nueva población.

### 5. Experimentación

Tal y como se ha comentado los datos que se manejan son los que se proporcionan para la competición propuesta en los simposios SICO2010 y TAMIDA2010. Concretamente se proveen 549 datos relacionados con una cuenca hidrogáfica, de los que no se desvela su significado y se propone predecir los siguientes 30 datos. La serie se muestra en la Figura 2.

CO<sup>2</sup>RBFN ha obtenido buenos resultados hasta ahora en problemas de predicción a corto plazo [16][18][19], por lo que supone un reto enfrentarse a la competición que aquí se plantea. Siguiendo el modo de trabajo que se ha utilizado hasta ahora lo que se ha hecho es preparar un conjunto de entrenamiento para cada uno de los 30 datos u horizontes que se tienen que predecir.

Las muestras de cada uno de los conjuntos de entrenamiento tendrá la siguiente composición ( $n-k$ ,  $n-k+1$ , ...,  $n-2$ ,  $n-1$ ,  $n$ ,  $n+h$ ) donde  $h$  es el horizonte a predecir en ese conjunto y  $k$  es igual a  $h$  salvo para los 5 primeros conjuntos donde  $k$  se ha establecido a 5. De cada uno de estos conjuntos de entrenamiento se extraen una serie de muestras, una de cada 25 empezando por el final y equidistribuidamente, para de esta manera formar los que llamaremos conjuntos de validación.

Para cada horizonte/conjunto de entrenamiento CO<sup>2</sup>RBFN se ejecuta 10 veces obteniendo 10 modelos o RBFNs. A cada uno de estos 10 modelos se le pasa su correspondiente conjunto de validación y que evidentemente depende del horizonte que estemos prediciendo. Se calcula el error de predicción para cada modelo obtenido y el modelo que menor error consiga será utilizado para la predicción de ese horizonte. Esta metodología se utilizará para predecir cada horizonte.

El número de individuos/RBFs empleados en la predicción de los primeros 19 horizontes es 10 y para el resto de los horizontes es 15. Estos valores se han determinado empíricamente.

En la Tabla 2 se muestra para cada horizonte la predicción que ha obtenido el modelo y el error relativo medio cometido en cada conjunto de validación.

Tabla 2. Para cada horizonte: predicción y error cometido en cada conjunto de validación

<i>Horizonte</i>	<i>Error validación</i>	<i>Predicción</i>
1	0,05526	3,07458
2	0,07088	3,07676
3	0,07590	3,11548
4	0,07853	3,07110
5	0,07060	3,14041
6	0,08741	3,01530
7	0,10949	3,08729
8	0,09040	2,97406
9	0,08197	3,15007
10	0,08601	2,85390
11	0,09658	3,25490
12	0,08131	3,44280
13	0,07947	3,36731
14	0,10542	3,16163
15	0,09032	3,08984
16	0,09431	3,05376
17	0,09460	2,93842
18	0,09911	3,07505
19	0,10215	2,79322
20	0,11192	3,06002
21	0,06706	3,01226
22	0,09632	3,14203
23	0,09205	2,88003
24	0,08528	2,90357
25	0,09454	2,83364
26	0,09585	2,73267
27	0,08442	2,23629
28	0,10545	2,14050
29	0,10053	2,46966
30	0,11643	2,63701

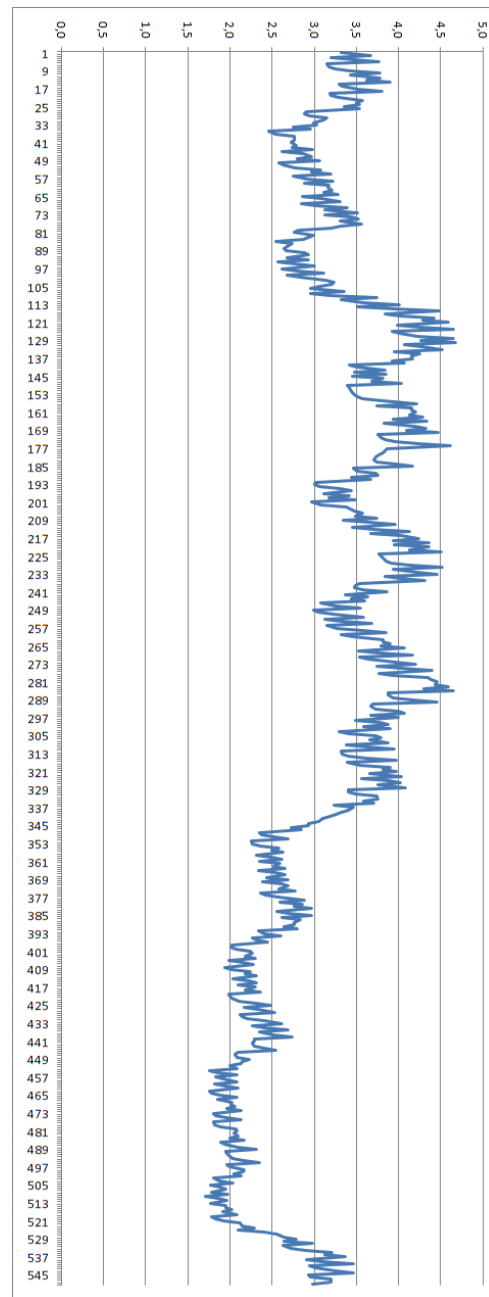


Figura 2. Serie temporal

En la Figura 3 se muestra predicción realizada.

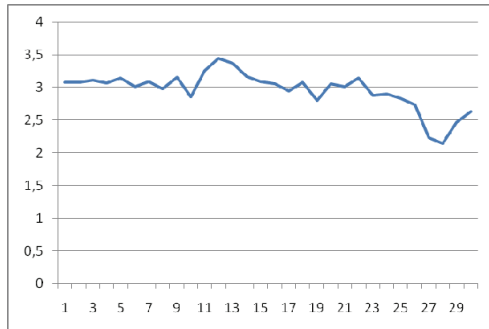


Figura 3. Datos de la predicción realizada

## 6. Conclusiones

Este trabajo presenta un algoritmo bioinspirado para diseñar redes de funciones de base radial aplicado a la predicción de series temporales. En él se mantiene una población de RBFs que cooperan para lograr una solución final y que compiten por su supervivencia. El comportamiento (asignación de crédito) de cada función base dentro de la red completa se mide en base a tres factores: la contribución,  $a_i$ , de la RBF a la salida de la red, el error,  $e_i$ , de la RBF y el solapamiento,  $s_i$ , de la RBF con otras RBFs.

Se utilizan cuatro operadores para poder aplicar a una RBF dada en el proceso evolutivo: Elimina, Mutación Aleatoria, Mutación Informada y Nulo. Con las dos modalidades de mutación se consigue un equilibrio adecuado entre las cualidades de explotación y exploración que todo algoritmo evolutivo debe poseer. Por un lado la mutación informada utiliza información del entorno de una neurona para modificar ésta de forma que se adapte óptimamente a su entorno. La mutación aleatoria promueve modificaciones de forma que se favorezca la exploración del entorno y se huya de óptimos locales.

La aplicación de dichos operadores es determinada mediante un Sistema basado en Reglas Difusas. Las entradas a dicho sistema son los parámetros  $a_i$ ,  $e_i$ , y  $s_i$ , usados para medir la asignación de crédito y las salidas son las

probabilidades de aplicación de los distintos operadores.

El algoritmo propuesto se ha evaluado en la serie temporal proporcionada para la competición propuesta por los simposios SICO2010 y TAMIDA2010.

Como líneas de trabajo futuro se intentará incorporar al método elementos que mejoren su eficiencia en la predicción a largo plazo.

## Agradecimientos

Este trabajo ha sido parcialmente subvencionado por el proyecto del Ministerio de Ciencia y Tecnología TIN2008-06681-C06-02, por el proyecto del plan Andaluz de Investigación TIC-3928 y por el proyecto de la Universidad de Jaén UJA-08-16-30.

## Referencias

- [1] Bäck, T.; Hammel, U.; Schwefel, H. Evolutionary computation: comments on the history and current state. *IEEE Trans. on Evolutionary Computation*, 1(1): 3-17, 1997.
- [2] Broomhead, D.; Lowe, D. Multivariable functional interpolation and adaptive networks. *Complex System* 2:321-355, 1988.
- [3] Buchtala, O.; Klimek, M.; Sick, B. Evolutionary optimization of radial basis function classifiers for data mining applications. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 35(5): 928-947, 2005.
- [4] Co, H.C.; Boosarawongse, R. Forecasting Thailand's rice export: Statistical techniques vs. artificial neural networks. *Computers and Industrial Engineering*, 53(4): 610-627, 2007.
- [5] Ghost, J.; Deuser, L.; Beck, S. "A neural network based hybrid system for detection, characterization and classification of short-duration oceanic signals, *IEEE JI. Of Ocean Engineering*, 17(4):351-363, 1992.
- [6] Goldberg, D.; Richardson J. Genetic algorithms with sharing for multimodal function optimization. In Grefenstette (ed.), *Proc. of the 2nd International Conference on Genetic Algorithms*. Lawrence Erlbaum Associates, 41-49, 1987.
- [7] Golub, G.; Van Loan, C. *Matrix computations*. J. Hopkins University Press, 3rd ed., 1996.

- [8] González, J., Rojas, I., Ortega, J., Pomares, H., Fernández, J., Fco, A. Multiobjective evolutionary optimization of the size, shape, and position parameters of radial basis function networks for function approximation. *IEEE Transactions on Neural Networks*, 14(6): 1478-1495, 2003.
- [9] Khashei, M; Reza, S.; Bijari, M. A new hybrid artificial neural networks and fuzzy regression model for time series forecasting. *Fuzzy Sets and Systems*, 159 (7): 769-786, 2008.
- [10] Lacerda, E; Carvalho, A.; Braga A.; Ludermir T. Evolutionary Radial Functions for Credit Assessment. *Applied Intelligence* 22. Springer Netherlands. 167-181, 2005.
- [11] Mamdani, E.; Assilian, S. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man-Machine Stud.*, 7(1): 1-13, 1975.
- [12] Moody, J.; Darken, C, J. Fast learning in networks of locally-tuned processing units. *Neural Computing*, 1: 281-294, 1989.
- [13] Neruda, R.; Kudová, P. Learning methods for radial basis function networks. *Future Generation Computer Systems*, 21(7): 1131-1142, 2005.
- [14] Park, J.; Sandberg, I. Universal approximation using radial-basis function networks. *Neural Comput.*, 3: 246-257, 1991.
- [15] Pedrycz, W. Conditional fuzzy clustering in the design of radial basis function neural networks. *IEEE Transactions on Neural Networks*, 9(4): 601-612, 1998.
- [16] Pérez-Godoy, M. D.; Rivera, A. J. y del Jesus, M. J.  $CO^2RBFN$ : predicción de series temporales con un enfoque cooperativo-competitivo. *Actas del Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB'09)*, pp. 269-276, 2009.
- [17] Perez-Godoy, M. D.; Rivera, A. J.; Berlanga, F.J.; Del Jesus, M.J.  $CO^2RBFN$ : an evolutionary cooperative-competitive RBFN design algorithm for classification problems. *Soft Computing* 14(9): 953-971. 2010.
- [18] Pérez-Godoy, M. D.; Pérez, P.; Rivera, A. J.; del Jesus, M. J.; Frías, M. P. y Parras, M.  $CO^2RBFN$  for short-term forecasting of the extra virgin olive oil price in the Spanish market. *International Journal of Hybrid Intelligent Systems*, 7(1): 75-87, 2010
- [19] Pérez-Godoy, M. D.; Pérez-Recuerda, P.; Frías, M.P.; Rivera, A. J.; Carmona, C. J. y Parras, M.:  $CO^2RBFN$  for short and medium term forecasting of the extra-virgin olive oil price. *Proceedings of the International Workshop on Nature Inspired Cooperative Strategies for Optimization (NICSO'10)*, 2010
- [20] Pino, R. ; Parreno J.; Gomez, A. ; Priore, P. Forecasting next-day price of electricity in the Spanish energy market using artificial neural networks. *Engineering Applic. of Artificial Intelligence*, 21(1):53-62, 2008.
- [21] Potter, M.; De Jong, K. Cooperative Coevolution: an architecture for evolving coadapted subcomponents". *Evolutionary Computation*, 8(1): 1-29, 2000.
- [22] Powell, M. Radial basis functions for multivariable interpolation: A review. In *IMA. Conf. on Algorithms for the approximation of functions and data*, 143-167, 1985.
- [23] Rivera, A.J.; Rojas, I.; Ortega, J.; del Jesús, M.J.; A new hybrid methodology for cooperative-coevolutionary optimization of radial basis function networks. *Soft Computing*. 11 (7): 655-668. 2007.
- [24] Topchy A.; Lebedko, O.; Miagkikh V.; Kasabov N.; Adaptive training of radial basis function networks based on co-operative evolution and evolutionary programming. *Prog. in connectionist-based information syst.* N. Kasabov et al (eds), Springer, 253-258, 1998.
- [25] Ture, M.; Kurt, I. Comparison of four different time series methods to forecast hepatitis A virus infection. *Expert Systems with applications*, 31(1):41-46, 2006.
- [26] Whitehead, B; Choate, T. Cooperative-competitive genetic evolution of Radial Basis Function centers and widths for time series prediction. *IEEE Trans. on Neural Networks*, 7(4):869-880, 1996.
- [27] Widrow, B.; Lehr, M.A. 30 Years of adaptive neural networks: perceptron, madaline and backpropagation. *Proceedings of the IEEE*, 78(9): 1415-1442, 1990.