

CO²RBFN: predicción de series temporales con un enfoque cooperativo-competitivo

M. Dolores Pérez-Godoy¹, Pedro Pérez², A. J. Rivera³, M. J. del Jesus⁴, Pedro L. López⁵

Resumen-- En este trabajo presentamos la adaptación de un algoritmo bioinspirado para el diseño de Redes de Funciones de Base Radial (RBFNs), CO²RBFN, para aplicarlo al problema de predicción de series temporales. 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. El modelo se ha aplicado a problemas reales de predicción de series temporales y los resultados obtenidos se comparan con otros tipos de modelos de minería de datos. Tras un análisis de los resultados se puede observar que la eficiencia del modelo presentado es comparable con la del resto de métodos aplicados.

Palabras clave-- RBFNs, Modelo Cooperativo-Competitivo, Predicción de series temporales.

I. INTRODUCCIÓN

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 [9], clasificación [4] o predicción de series temporales [24]. Las Funciones de Base Radial (RBFs) se usaron inicialmente en interpolación numérica y aproximación funcional [20]. A finales de los 80 tienen lugar las primeras investigaciones de redes neuronales basadas en RBFs [3] [13].

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 [15] o la analogía entre la salida de estas redes y los campos receptivos localizados, encontrados en estructuras biológicas cerebrales.

Básicamente una salida de una RBFN ofrece una suma ponderada, por unos pesos, de las respuestas de cada una de las neuronas/RBFs que conforman la red. Una salida o respuesta de una RBF tiene un carácter local y viene dada en función de un centro y un radio. Así pues, el objetivo a la hora de diseñar

una RBFN es determinar el centro y el radio que caracteriza cada RBF así como su peso para cada salida de la red.

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 [16]. En la segunda etapa y para calcular los pesos se pueden utilizar algoritmos como Least Mean Square (LMS) [25], Singular Value Decomposition (SVD) [8], etc.

El diseño de una RBFN también se puede abordar desde el punto de vista de la computación evolutiva [2]. 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 [19] 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 [21]. En el trabajo actual se propone un marco de evolución cooperativo-competitivo para resolver el problema de predicción de series temporales con nuevas medidas para la asignación de crédito, nuevos operadores, nuevas bases de reglas, nuevos algoritmos para la inclusión de individuos en la población, nuevo mecanismo de reemplazo o una aplicación de operadores a toda la población.

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.

¹ Dpto. Informática. U. de Jaén. E-mail: lperez@ujaen.es

² Dpto. Informática. U. de Jaén. E-mail: pperez2104@gmail.com

³ Dpto. Informática. U. de Jaén. E-mail: arivera@ujaen.es

⁴ Dpto. Informática. U. de Jaén. E-mail: mjjesus@ujaen.es

⁵ D. Informática. U. Jaén. E-mail: pll0001@estudiante.ujaen.es

En el trabajo actual aplicaremos nuestro algoritmo a la predicción de tres series temporales construidas a partir de datos reales y compararemos los resultados obtenidos con los de otros métodos de minería de datos.

La organización del resto de este trabajo se describe a continuación. En la sección II se comentan generalidades sobre las series temporales. En la sección III se introducen las RBFNs y se explica cómo se aplican a problemas de predicción de series temporales. En la sección IV se presenta el algoritmo propuesto y sus resultados se muestran en la sección V. Por último se explican las conclusiones alcanzadas y las líneas de trabajo futuro en la sección VI.

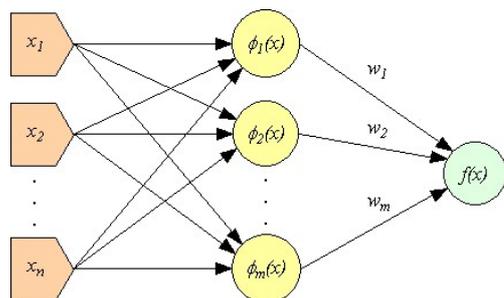


Fig. 1. Topología de una RBFN para predicción.

II. 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 permite 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.

Además, bajo el supuesto de que las condiciones estructurales que conforman la serie objeto de estudio permanecen constantes, también se trata de predecir el comportamiento futuro.

Las series pueden tener una periodicidad anual, semestral, trimestral, mensual, etc., según los periodos de tiempo en los que están recogidos los datos que la componen.

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, los datos se analizan para descubrir un modelo de comportamiento futuro de los mismos para poder estimar con antelación, posibles valores y tendencias.

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 [5][10][18][23]. Normalmente la capacidad de predicción de los métodos de minería de datos está muy relacionada con su característica de aproximadores universales.

III. REDES DE FUNCIONES DE BASE RADIAL Y SU DISEÑO EVOLUTIVO

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 Fig. 1. Las m neuronas/RBFs de la capa oculta ofrecen una activación simétrica radial $\phi: R^n \rightarrow R$, 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 ϕ , $d_i \in R$ es el radio y como $\|\cdot\|$ 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 [4][11][14] 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 [17][21][22][24] que implementan este tipo de paradigma, debido sobre

todo a la dificultad que entraña definir los parámetros de cooperación y competición entre los individuos.

IV. CO²RBFN: 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.

Se utiliza un esquema de codificación real. Cada cromosoma representa el centro (con un número de coordenadas igual al número de características de los ejemplos) y el radio de una RBF.

Las etapas principales de CO²RBFN se explican en las siguientes subsecciones. A continuación se muestra su pseudocódigo:

PRINCIPALES ETAPAS DE CO ² RBFN
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

A. 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.

B. Entrenamiento de las RFBs

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 [25].

C. 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 , o_i para cada RBF ϕ_i .

La contribución, a_i , de la RBF ϕ_i , $i=1\dots m$, 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 ϕ_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, ϕ_i , con otras RBFs se cuantifica usando el parámetro o_i . Este parámetro se calcula partiendo de la base de la metodología *fitness sharing* [7], que intenta mantener la diversidad en la población. El factor se expresa como:

$$o_i = \sum_{j=1}^m o_{ij} \quad (4)$$

donde o_{ij} mide el solapamiento entre las RBF ϕ_i y ϕ_j , $j=1\dots m$.

D. 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 [6] que son similares a las utilizadas por el algoritmo LMS en el cálculo de los pesos. Se calcula el error para los ejemplos que están dentro del radio de la RBF ϕ_j . Para cada coordenada del centro y para el radio se calcula una variación Δc_{ij} y Δd_i respectivamente. Las nuevas coordenadas y el nuevo radio se obtienen modificando (incrementando o decrementando) 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.

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

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

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

- **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 [12], cuyas entradas son los parámetros a_i , e_i y o_i , éstos determinan la asignación de crédito de cada RBF. Estas entradas se consideran como variables lingüísticas va_i , ve_i y vo_i , y las salidas son $p_{eliminas}$, p_{ma} , p_{mi} y p_n , que representan la probabilidad de aplicar los operadores Elimina, Mutación Aleatoria, Mutación Informada y Nulo, respectivamente. La Fig. 2 muestra las funciones de pertenencia para las etiquetas lingüísticas de las entradas y las salidas 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 I muestra las reglas usadas 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.

En el diseño de reglas se tiene en cuenta el hecho de que una RBF es peor si su contribución (a_i) es baja, su error (e_i) es alto y su solapamiento (o_i) también es alto. En el lado opuesto, una RBF es mejor cuando su contribución es alta y su error y solapamiento son bajos. De esta forma, la probabilidad de eliminar una RBF se incrementa cuando ésta tiene un mal comportamiento; la probabilidad de mutar va aumentando a medida que el comportamiento es mejor. El operador nulo se aplicará cuando el comportamiento de una RBF sea muy bueno.

E. 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.

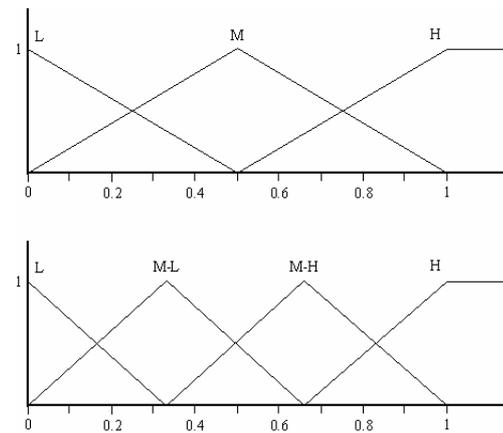


Fig. 2. Arriba: etiquetas lingüísticas para las variables de entrada. Abajo: etiquetas lingüísticas de las variables de salida.

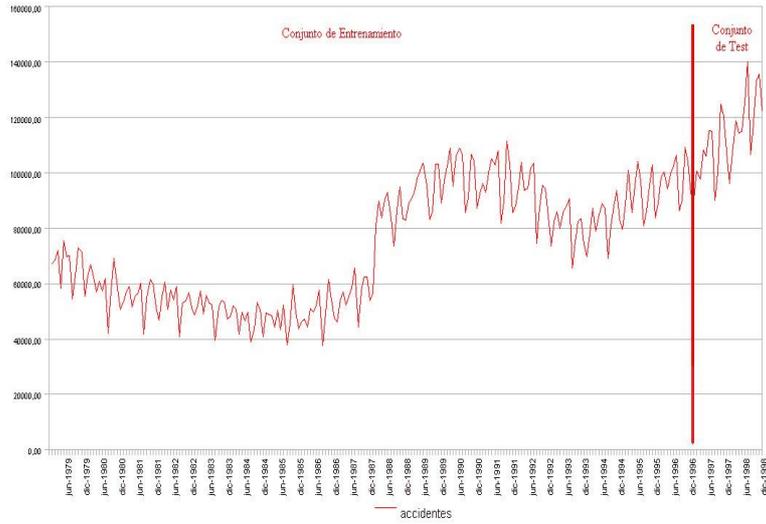


Fig. 3. Accidentes en jornada de trabajo en España.

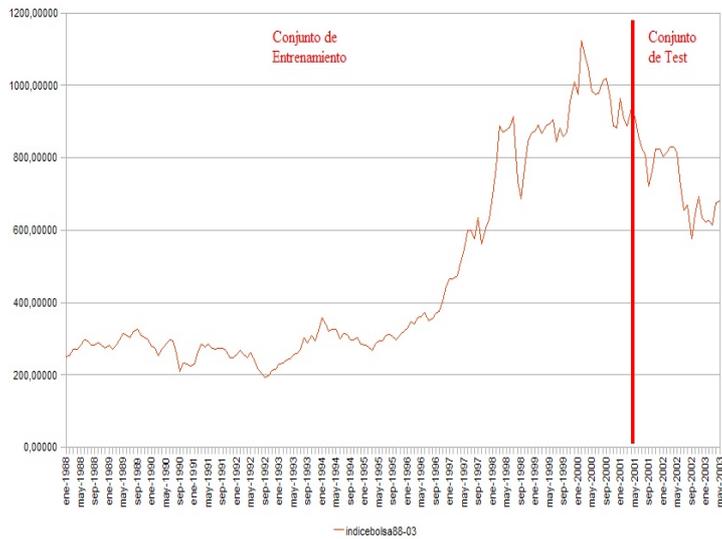


Fig. 4. Índice general de la bolsa de Madrid.

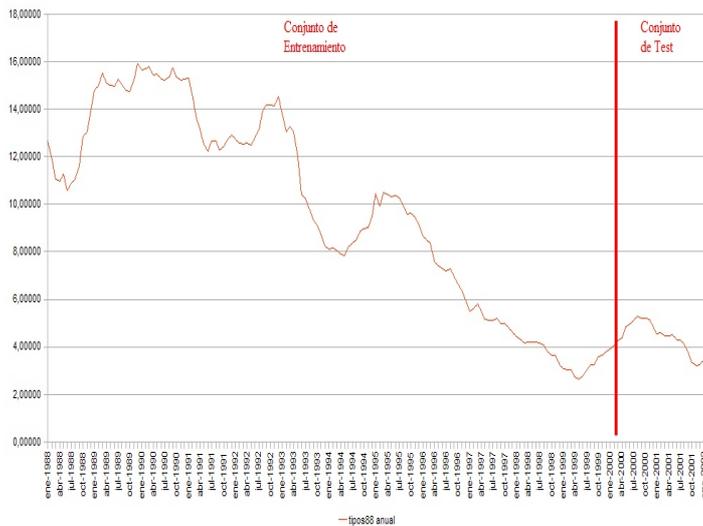


Fig. 5. Tipo de interés interbancario a un año.

TABLA I

		BASE DE REGLAS UTILIZADA					
		Antecedentes			Consecuentes		
	v_a	v_e	v_o	$P_{elimina}$	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

F. 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.

V. ENTORNO DE EXPERIMENTACIÓN

Para realizar la experimentación se va a predecir el comportamiento de tres series temporales (Fig. 3, Fig. 4, Fig. 5), disponibles en <http://www.alianzaeditorial.es/3491099/ejercicios.zip>:

- Accidentes en jornada de trabajo en España: formada por los datos recogidos mensualmente desde enero de 1979 hasta diciembre de 1998. Fuente: INE.
- Índice general de la Bolsa de Madrid: recoge los datos mensuales desde enero de 1988 hasta Mayo de 2003. Fuente: Banco de España e INE.
- Tipo de interés interbancario a un año: recoge los datos mensuales desde enero de 1988 hasta marzo de 2002. Fuente: Ministerio de Economía.

Como conjunto de test se han tomado los últimos 25 datos de la serie y como conjunto de entrenamiento el resto de los datos.

Para formar las muestras de los conjuntos de datos se ha elegido un diseño clásico de patrones ($n-3, n-2, n-1, n, n+1$), donde $n+1$ es el dato a predecir y el resto los datos de entrada al modelo.

Los métodos utilizados para comparar con CO²RBFN son los siguientes:

- Fuzzy GA-P: algoritmo para el diseño de sistemas difusos. Utiliza un método de computación evolutiva híbrido entre la programación genética y los algoritmos genéticos. Implementación obtenida de la herramienta Keel [1].
- MLP-GradConj: algoritmo para el diseño de Redes Perceptrón Multicapa que utiliza el algoritmo de Gradiente Conjugado para el aprendizaje. Implementación obtenida de la herramienta Keel [1].

- NU-SVR: algoritmo para el diseño de una máquina de vectores de soporte utilizando información de segundo orden. Los kernels utilizados son del tipo RBF. Implementación obtenida de la herramienta Keel [1].
- RBFN-LMS: algoritmo para el diseño de RBFNs que usa el método LMS para el entrenamiento de la red. Implementación obtenida de la herramienta Keel [1].

Estos métodos con los que se compara se han elegido de forma que queden representados los modelos más utilizados en minería de datos, como son las redes neuronales, sistemas difusos y máquinas de vectores de soporte. Dentro de éstos paradigmas se han escogido los algoritmos que suelen dar mejores resultados.

Los parámetros utilizados en los métodos con los que comparamos son los recomendados en la literatura. Para CO²RBFN el número de ejecuciones es 200 y el número de RBFs o individuos de la población se ha fijado a 20.

La medida de error considerada, para evaluar la precisión de las predicciones por parte de los métodos, es el error MAPE.

Las series temporales se han diferenciado para evitar problemas derivados de la no estacionalidad de las mismas. Las predicciones se han realizado sobre los datos diferenciados, pero los errores se han calculado después de recomponer la serie original.

Para obtener los resultados se han ejecutado los algoritmos 10 veces y en las tablas (TABLA II, TABLA III y TABLA IV) se muestra la media del error MAPE cometido y su desviación típica.

Las figuras Fig. 6, Fig. 7 y Fig. 8 muestran la mejor predicción conseguida por los métodos para el conjunto de test.

TABLA II
RESULTADOS CON LA SERIE DE ACCIDENTES

Método	MAPE Test
Fuzzy GP-A	0.08276±0.00684
MLPGradConj	0.08311±0.00495
NU-SVR	0.09632±0.00000
RBFN-LMS	0.08019±0.00496
CO ² RBFN	0.08310±0.00351

TABLA III
RESULTADOS CON LA SERIE DE LA BOLSA

Método	MAPE Test
Fuzzy GP-A	0.06094±0.00655
MLPGradConj	0.05563±0.00224
NU-SVR	0.05260±0.00000
RBFN-LMS	0.05962±0.00238
CO ² RBFN	0.05257±0.00119

TABLA IV
RESULTADOS CON LA SERIE INTERÉS INTERBANCARIO

Método	MAPE Test
Fuzzy GP-A	0.04016±0.00280
MLPGradConj	0.03522±0.00125
NU-SVR	0.04049±0.00000
RBFN-LMS	0.03863±0.00326
CO ² RBFN	0.03743±0.00199

El análisis de los resultados muestra:

- Un comportamiento correcto y similar de los métodos utilizados para la predicción de las series temporales.
- La serie más complicada de predecir es la que contienen los datos de los accidentes en jornada de trabajo en España. Tal y como se puede observar esta serie presenta oscilaciones muy bruscas y un comportamiento en el conjunto de test que no se asemeja al del conjunto de entrenamiento.
- La serie que mejor han modelizado los métodos es la que contiene los datos del tipo de interés interbancario a un año. Sus variaciones son más suaves y el comportamiento de los datos en el conjunto de test es similar al de los datos en el conjunto de entrenamiento.
- CO²RBFN obtiene redes que predicen las series con resultados comparables a los obtenidos con los mejores métodos. La desviación típica del error obtenido suele ser baja lo cual implica que es un método robusto.

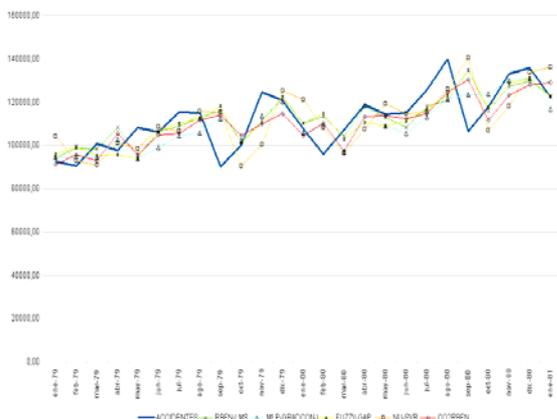


Fig. 6. Predicción en el conjunto de test para la serie de accidentes.

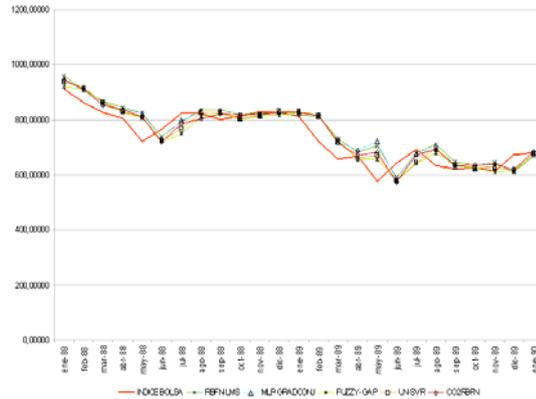


Fig. 7. Predicción en el conjunto de test para la serie de la bolsa.

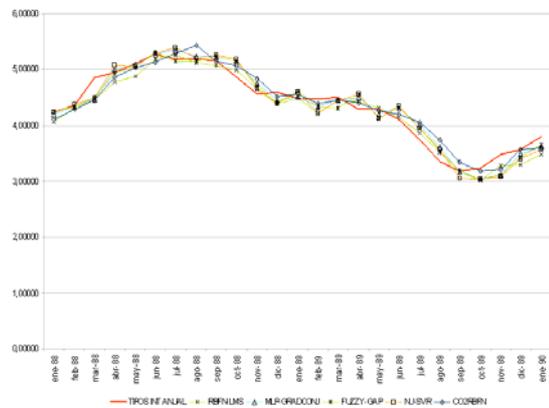


Fig. 8. Predicción en el conjunto de test para la serie del interés interbancario.

VI. CONCLUSIONES

Este trabajo presenta un algoritmo bioinspirado para diseñar redes 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, o_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 o_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 utilizando series temporales de datos reales y los resultados obtenidos son comparables a los de los mejores métodos utilizados en la comparativa.

Como líneas de trabajo futuro se intentará mejorar la eficiencia de los operadores y su sistema de aplicación.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente subvencionado por los proyectos españoles TIN2005-04386-C05-03 y TIN2007-60587.

REFERENCIAS

- [1] Alcalá-Fdez J, Sánchez L, García S, Del Jesus MJ, Ventura S, Garrell JM, Otero J, Romero C, Bacardit J, Rivas VM, Fernández JC, Herrera F. KEEL: A Software Tool to Assess Evolutionary Algorithms to Data Mining Problems. *Soft Comput* (in press). (2008). Doi: 10.1007/s00500-008-0323-y.
- [2] 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.
- [3] Broomhead, D.; Lowe, D. Multivariable functional interpolation and adaptive networks. *Complex System* 2:321-355, 1988.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] Golub, G.; Van Loan, C. *Matrix computations*. J. Hopkins University Press, 3rd ed., 1996.
- [9] 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.
- [10] 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.
- [11] Lacerda, E; Carvalho, A.; Braga A.; Ludermir T. Evolutionary Radial Functions for Credit Assessment. *Applied Intelligence* 22. Springer Netherlands. 167-181, 2005.
- [12] Mamdani, E.; Assilian, S. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man-Machine Stud.*, 7(1): 1-13, 1975.
- [13] Moody, J.; Darken, C, J. Fast learning in networks of locally-tuned processing units. *Neural Computing*, 1: 281-294, 1989.
- [14] Neruda, R.; Kudová, P. Learning methods for radial basis function networks. *Future Generation Computer Systems*, 21(7): 1131-1142, 2005.
- [15] Park, J.; Sandberg, I. Universal approximation using radial-basis function networks. *Neural Comput.*, 3: 246-257, 1991.
- [16] Pedrycz, W. Conditional fuzzy clustering in the design of radial basis function neural networks. *IEEE Transactions on Neural Networks*, 9(4): 601-612, 1998.
- [17] Pérez-Godoy, M; Rivera, A.J.; Jesus, M.J.; Rojas, I. CoEvRBFN: an approach to solving the classification problem with a hybrid cooperative-coevolutionary algorithm. 9th International Work-Conference on Artificial Neural Networks (IWANN07), LNCS 4507, 324-332, 2007.
- [18] 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.
- [19] Potter, M.; De Jong, K. Cooperative Coevolution: an architecture for evolving coadapted subcomponents". *Evolutionary Computation*, 8(1): 1-29, 2000.
- [20] 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.
- [21] 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*. ISSN 1432-7643. 2007 D.O.I: <http://dx.doi.org/10.1007/s00500-006-0128-9>.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] Widrow, B.; Lehr, M.A. 30 Years of adaptive neural networks: perceptron, madaline and backpropagation. *Proceedings of the IEEE*, 78(9): 1415-1442, 1990.