

ARTÍCULO ORIGINAL

DISEÑO DE REDES NEURONALES DE FUNCIONES DE BASE RADIAL CON UN ENFOQUE MULTIOBJETIVO COOPERATIVO COMPETITIVO PARA LA PREDICCIÓN DE TEMPERATURAS¹

*RADIAL BASIS NEURAL
NETWORKS DESIGN USING A
COMPETITIVE COOPERATIVE
COEVOLUTIONARY
MULTIOBJECTIVE ALGORITHM FOR
TEMPERATURE PREDICTION¹*

Christian Von Lücken²

Manuel Ávalos Godoy²

Arturo Ferreira Duarte²

Enrique Dávalos²

¹ Trabajo presentado por la Facultad Politécnica y financiado con rubros del Rectorado de la Universidad Nacional de Asunción durante el año 2013.

² Docentes de la Facultad Politécnica. Email: clucken@pol.una.py, mavalosgodoy@gmail.com, aferreiraduarte@gmail.com, edavalos@pol.una.py

RESUMEN

Este trabajo presenta la utilización de un modelo evolutivo de entrenamiento de Redes Neuronales de Base Radial (*Radial Basis Neural Networks* - RBNN) para la predicción de temperaturas.

El entrenamiento se realiza en dos etapas bien diferenciadas. En la primera, se obtienen los pesos de la función de base radial (*Radial basis function* - RBF) y, en una segunda etapa, se utiliza un algoritmo multiobjetivo cooperativo competitivo para determinar los parámetros de cada RBF. El método ha sido aplicado sobre un conjunto de datos de temperatura reales, demostrando resultados similares a los obtenidos por un modelo de entrenamiento que representa el estado del arte en algoritmos bio-inspirados para la predicción de series temporales.

PALABRAS CLAVE: Neural Network training, Radial Basis Neural Networks, Time series prediction, Competitive cooperative coevolutionary multiobjective algorithm

ABSTRACT

This work presents the use of an evolutionary model to train Radial Basis Neural Network (RBNN) for temperature prediction.

Training is performed in two phases. First, weights of the radial basis function (RBF) are obtained and in the second phase, a competitive cooperative co-evolutionary multi-objective

algorithm is used to determine the parameters for each RBF node. The model has been applied to real set of temperature data obtaining similar results than those obtained by a model representing the state-of-the-art in bio-inspired algorithms for time series prediction.

KEY WORDS: Neural Network training, Radial Basis Neural Networks, Time series prediction, Competitive cooperative coevolutionary multiobjective algorithm

INTRODUCCIÓN

Las funciones de base radial fueron utilizadas por primera vez en las redes neuronales por (BROOMHEAD and LOWE, 1988). Las Redes Neuronales de Funciones de Base Radial (RNBR, *Radial Network with Radial Basis*) se basan en un aprendizaje equivalente a encontrar una función multidimensional que provee una mejor aproximación estadística al conjunto de datos de entrenamiento.

La arquitectura básica de una RNBR está compuesta de tres capas: una capa de entrada, una capa oculta y una capa de salida.

Todos los nodos que componen las capas de entrada y salida de una RNBR están conectados con todos los nodos de la capa oculta. A diferencia de otros modelos de redes neuronales, en una RNBR las conexiones entre los nodos de la entrada y la capa oculta no están ponderadas. La capa oculta esta compuesta de nodos que tienen una función de activación radial simétrica.

Las RNBR han sido exitosamente aplicadas en diferentes problemas.

La predicción de series temporales es una de las principales aplicaciones de las RNBR. A fin que las RNBR sean útiles es necesario especificar la arquitectura de la red y sus parámetros: las variables de entrada, el número de unidades ocultas, el centro y el ancho de cada unidad oculta. Determinar la mejor combinación de tales parámetros es determinante en el funcionamiento de una RNBR y por tanto es un tema sumamente importante en la literatura de RNBR.

El gran número de trabajos de investigación que abordan el diseño de las redes neuronales (HAYKIN, 1999; PLATT, 1991; WIDROW and LEHR, 1990), y más concretamente de las Redes Neuronales de Base Radial (*Radial Basis Neural Networks* - RBFN), pone de manifiesto la dificultad de esta tarea y la ausencia de mecanismos generales para ajustar automáticamente los parámetros de la red, de tal forma que se puedan adaptar al entorno sin necesidad de intervención humana.

El algoritmo típico de diseño de las RBFN tiene dos etapas: en la primera etapa se determinan los centros y los radios de las RBF (*Radial basis function*), mientras que en la segunda se calculan sus pesos.

Varios trabajos han propuesto métodos para automatizar el diseño de RNBR. En los últimos años se han propuesto algunos enfoques de meta-heurísticas basadas en poblaciones, las cuales, en su mayoría, trabajan con métodos

evolutivos típicos donde un individuo representa una red completa como: Algoritmos Evolutivos (BUCHTA-LA et al, 2005), Optimización por enjambre de partículas (*Particle Swarm Optimization*) (YANG et al., 2013), Evolución Diferencial (*Differential Evolution*) (DASH et al., 2013), entre otros.

No obstante, para cierto tipo de problemas, el diseño evolutivo presenta algunas dificultades cuando un individuo representa una solución completa compuesta de subcomponentes independientes (POTTER y JONG, 2000).

Un enfoque distinto es la utilización de técnicas cooperativas competitivas donde un individuo representa sólo una parte de la solución y, en conjunto con otros individuos, forman el total de la misma. Un algoritmo evolutivo híbrido que utiliza este enfoque para el entrenamiento de RNBRs es el modelo CO2RBFN (PEREZ-GODOY et al., 2010) donde se utilizan unas funciones de evaluación (aporte, error, solapamiento) sobre cada neurona y por medio de lógica difusa se determinan los operadores genéticos que se aplicarán sobre la misma. En AVALOS et al., (2012) se extiende el CO2RBFN proponiendo la utilización de un modelo evolutivo multi-objetivo.

Desde la década de los 90 los métodos de aprendizaje de máquina se han utilizado cada vez más en las ciencias del ambiente, entre los que la predicción del clima y el tiempo es una de las aplicaciones más importantes (HSIEH, 2009.; PASINI et al., 2008).

Como un ejemplo muy reciente del uso específico de las RNBR para predicción de la temperatura atmosférica mensual, en (XUE et al, 2012) se propone un método híbrido basado en RNBR para tal fin.

En el presente trabajo se utiliza el algoritmo evolutivo cooperativo competitivo para el diseño de RBFN presentado por los autores en (AVALOS et al. 2012) para la predicción de datos de temperaturas mínimas y máximas utilizando un conjunto de datos reales. Dicho algoritmo se basa en las mismas funciones utilizadas por el CO2RBFN, pero utilizando un enfoque multi-objetivo, y una función adicional que determina el grado de incidencia de una neurona sobre el conjunto de entrenamiento actual. En el enfoque propuesto en (AVALOS et al. 2012) no se utiliza un SBRD y por lo tanto se evita la definición de las reglas.

El presente documento está organizado como sigue. En la Sección II se describen las RBFN y su aplicación en la predicción de Series Temporales. En la Sección III se presenta el algoritmo propuesto.

En la Sección IV se presentan aspectos generales sobre la predicción de series temporales.

En la Sección V se analizan los resultados de las pruebas experimentales, y, finalmente, en la Sección VI se discuten las conclusiones junto con los lineamientos para trabajos futuros.

REDES DE FUNCIONES DE BASE RADIAL (RBFN)

Una Función de Base Radial (RBF) φ es una función de segundo orden no lineal que permite obtener un valor que representa la distancia de un patrón de referencia al origen (POWELL, 1997), o alternativamente a algún punto c , llamado *centro*. Una RBF, $\varphi_i: \mathbb{R}^n \rightarrow \mathbb{R}$, se puede expresar como:

$$(\vec{x}) = \varphi (\|\vec{x} - \vec{c}_i\| / r_i) \quad (1)$$

Note que la función tiene una salida simétrica, y se incrementa (o decrementa) de forma monótona con respecto a $c_i \in \mathbb{R}^n$. Tiene un factor escalar, $r_i \in \mathbb{R}$, conocido como radio, y el operador $\|\ \|$ se suele tomar como la distancia euclídea en \mathbb{R}^n .

Una red neuronal de funciones de base radial (BROOMHEAD and LOWE, 1988; YEE and HAYKIN, 2001) es un tipo de red hacia delante (*feedforward neural network*) con tres capas (Fig. 1): la capa de entrada con m nodos, una capa oculta con n neuronas (RBF) y una capa de salida, la cual puede contar con varios nodos pero para el caso de predicción presenta sólo un nodo.

Las n neuronas RBF de la capa oculta contienen una función de activación simétrica radial, que puede tomar diferentes formas (Gaussiana, Multicuadrática, Multicuadrática Inversa, etc.). La función de activación comúnmente utilizada es la función Gaussiana que viene dada por la siguiente expresión:

$$\varphi_i(\vec{x}) = e^{-\left(\frac{\|\vec{x}-\vec{c}_i\|}{r_i}\right)^2} \quad (2)$$

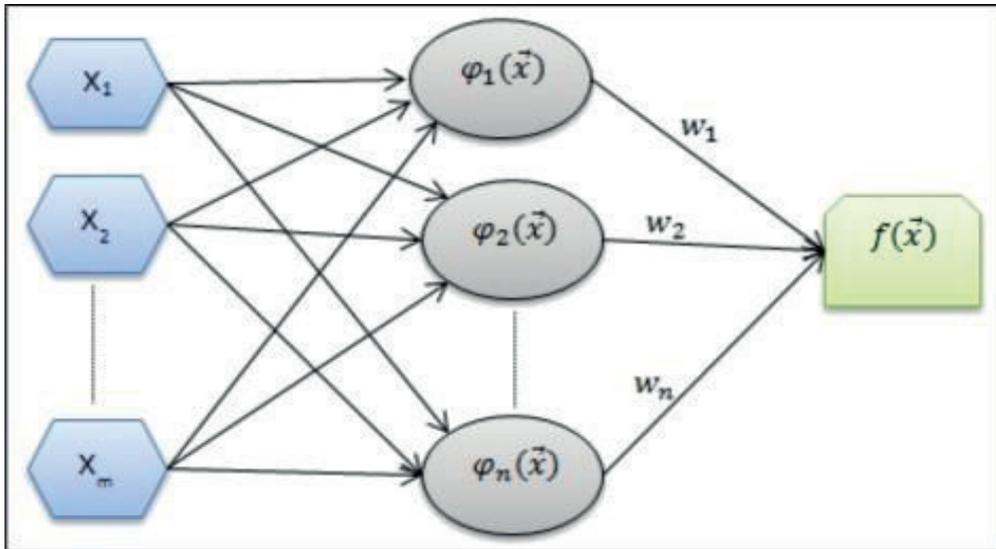
En donde φ_i , corresponde a la función de activación de la neurona i , \vec{x} la entrada actual a la red, \vec{c}_i es el centro de la neurona y r_i , es el radio de la neurona. Para obtener el resultado de la RBFN, el nodo de salida utiliza la

siguiente ecuación para el cálculo del valor de salida, que en este caso viene a ser la predicción obtenida:

En donde n indica la cantidad de neuronas en la capa oculta y $\omega_i \omega_i$ es el peso de la neurona i .

$$f(\vec{x}) = \sum \omega_i \varphi_i(\vec{x}) \quad (3)$$

Fig. 1. Estructura de una RBFN para predicción.



ALGORITMO EVOLUTIVO MULTI-OBJETIVO PARA EL DISEÑO DE RBFN

En este trabajo se propone un algoritmo evolutivo multi-objetivo con un enfoque cooperativo competitivo para la optimización de los parámetros de las neuronas RBF. En los algoritmos cooperativos-competitivos un único individuo no representa una solución completa sino una parte de ella. Entonces, se requiere de un grupo de individuos o todos los individuos de una población para formar la solución del problema. Cada individuo de la población coopera para alca-

nzar la solución óptima, pero también compiten por su supervivencia dado que los mejores adaptados, es decir aquellos que colaboren en obtener mejores soluciones, son los que permanecerán. En este caso la función de *fitness* promueve la competencia entre componentes que realizan sub-tareas similares y la cooperación entre componentes que realizan sub-tareas diferentes, ya que juntos consiguen resolver el problema total. Ejemplos de algoritmos cooperativos-competitivos se pueden encontrar en (WHITEHEAD and CHOATE, 1995; MIA-GKIKH et al., 1997; RIVERA et al., 2007)

En el Algoritmo 1 se presenta un pseudocódigo del modelo utilizado en este trabajo, denominado Multi-Objective Radial Basis Function Network (MORBFN) (AVALOS et al. 2012), el cual se detallará en las siguientes secciones.

Algoritmo 1: MORBFN

```

Generar neuronas
while condición parada modelo != true do
  Obtener pesos de las RBF
  while condición de parada MOEA != true do
    Generar población
    Evaluar población
    Evolucionar población con MOEA
  endwhile
  Actualizar red con Frente Pareto
end while
  
```

A. Obtención de pesos

Durante esta etapa se obtienen los pesos que serán asignados a las RBF. Para la obtención de los pesos se entrena la red utilizando el método LMS, también conocido como el algoritmo

de la Regla Delta. Básicamente consiste en aplicar una regla lineal para la corrección del error tras el procesamiento de una muestra del conjunto de entrenamiento, el Algoritmo 2 muestra un pseudocódigo del LMS.

Algoritmo 2: LMS

```

for k = 1 to p do
  Tomar el par (x_k, y_k)
  Obtener la salida de la red f(x_k)
  Modificar los pesos según la ecuación (4)
end for
  
```

El LMS adapta el valor del vector de pesos mediante la siguiente ecuación:

$$\vec{w}_{k+1} = \vec{w}_k + \delta \frac{e_k \varphi_i(\vec{x}_k)}{|\varphi_i(\vec{x}_k)|^2} \quad (4)$$

donde k indica el número de generación en la que se está y p es la cantidad de patrones en el conjunto de entrenamiento, \vec{w}_{k+1} es el valor actualizado del vector de pesos, \vec{w}_k es el valor actual del vector de pesos, $e_k = y_k - f(\vec{x}_k)$ es el error lineal

entre la respuesta deseada y la obtenida por la red, $\varphi_i(\vec{x}_k)$ es la salida de la función de activación de la i -ésima RBF para la entrada k y la expresión $|\varphi_i(\vec{x}_k)|^2 = \sum_{i=1}^n \varphi_i(\vec{x}_k)$ siendo n el número de RBF.

El parámetro δ es la velocidad de aprendizaje y mide el tamaño de la actualización que se va a realizar, dicho parámetro suele tomar un valor entre 0,1 y 1,0.

El error es el que determina la dirección en la que los pesos se van adaptando.

B. Generación de la población

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

Cada individuo de la población se evalúa de forma independiente y debe pasar por varias generaciones para definir si puede formar parte de la red actual.

Para obtener el centro de cada uno de los individuos se selecciona de forma aleatoria entre todos los patrones de entrada utilizados para el entrenamiento. Para el cálculo del radio se toma la distancia media entre todos los centros del conjunto de entrada y se asigna la mitad del valor obtenido como radio a cada uno de los individuos.

C. Evaluación del individuo

Para determinar si un individuo de la población formará parte de la red se utilizan las funciones definidas en (PEREZ-GODOY et al., 2010) conocidas como aporte, error y solapamiento. Además hemos agregado una función adicional llamada incidencia.

Esta es una variación de la función de aporte y se centra en que una neurona RBF debe abarcar el mayor número de patrones de entrada dentro de su

radio. (PEREZ-GODOY, 2010). Una neurona es mejor mientras obtenga el mayor aporte, menor error individual, menor solapamiento y mayor incidencia dentro de la red. Estos objetivos son utilizados por el algoritmo evolutivo multi-objetivo (MOEA, *Multi-objective Optimization Evolutionary Algorithm*) para determinar el *fitness* de los individuos a evaluar. A continuación se detallan las funciones objetivo utilizadas en este trabajo para evaluar a un individuo.

1. Aporte. Consiste en la contribución de una neurona RBF a la salida de la RBFN. Se calcula teniendo en cuenta tanto el peso de la RBF ω_i , como el número de patrones del conjunto de entrenamiento que están dentro de su radio, pi_i :

$$a_i = \begin{cases} |\omega_i| & \text{si } pi_i > q \\ |\omega_i| * (pi_i/q) & \text{en otro caso} \end{cases} \quad (5)$$

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

2. Error. Es el error Porcentual Absoluto Medio (*Mean Absolute Percentage Error*- MAPE) correspondiente a una neurona. Para determinar este valor, sólo se tienen en cuenta los patrones de entrenamiento que se encuentran dentro (pi_i) del radio de la neurona RBF.

$$a_i = \begin{cases} |\omega_i| & \text{si } pi_i > q \\ |\omega_i| * (pi_i/q) & \text{en otro caso} \end{cases} \quad (6)$$

3. Solapamiento. Indica el nivel de semejanza que tiene una neurona RBF en comparación con las demás neuronas que forman parte de la capa oculta.

$$s_i = \sum_{j=1}^n s_{ij} \quad (7)$$

$$s_{ij} = \begin{cases} 1 - \frac{\|c_i - c_j\|}{r_i} & \text{si } \|c_i - c_j\| < r_i \\ 0 & \text{en otro caso} \end{cases}$$

Se establece que una neurona i se solapa con otra neurona j , si y sólo si, la distancia entre los centros de ambos es menor al radio de la neurona i .

4. Incidencia. Consiste en la cantidad de patrones del conjunto de datos, establecidos como entrada a la red, que se encuentran dentro del radio de una neurona RBF. Un patrón de entrada se encuentra dentro del radio de la neurona i , si la distancia entre el centro (c_i) de la neurona y el patrón de entrada es menor o igual al radio (r_i) de la neurona. Esta función objetivo tiene un alto grado de correlación con la función de aporte.

$$in_i = \sum_{\forall \vec{x} \in X} g(\vec{x}) \quad (8)$$

$$g(\vec{x}) = \begin{cases} 1 & \text{si } \|\vec{x} - c_i\| \leq r_i \\ 0 & \text{en otro caso} \end{cases}$$

Para evaluar a un nuevo individuo, primero se debe encontrar el peso que se le asignará. Entonces, se selecciona de forma aleatoria la posición que tendrá el nuevo individuo dentro de la red, y una vez ubicada dicha posición el individuo toma el valor del peso que se encuentra asignado a la misma.

D. Evolución de la Población

Se utilizó el algoritmo evolutivo NSGA2 (*Non-dominated Sorting Genetic Algorithm*) (DEB et al., 2000), debido a sus buenos resultados y a su creciente utilización. Como es bien conocido, el NSGA2 podría no tener

un buen desempeño en un espacio de más de tres objetivos.

Sin embargo, si existe una gran correlación entre los objetivos esto podría hacer que el comportamiento no se vea afectado negativamente (ISHIBUCHI et al., 2011).

En la implementación del algoritmo fueron utilizados los siguientes operadores genéticos para la evolución de la población:

1) *Operador de cruce.* En base a pruebas empíricas realizadas el operador de cruce seleccionado es una modificación del *Heuristic Crossover* (DE LACERDA, 2003), debido a que se obtuvo un mejor desempeño con respecto al operador, comúnmente utilizado en NSGA2, el *Simulated Binary Crossover* (SBX).

Primeramente se selecciona un número aleatorio entre 0 y 1 llamado *Alpha* (α). El primer hijo lleva el centro del primer padre, y cada coordenada se modifica de la siguiente manera: a la primera coordenada del centro se le suma el producto entre el valor α y la diferencia entre la primera coordenada del centro del primer padre y la primera coordenada del centro del segundo padre, y este proceso se realiza para todas las coordenadas.

$$h_1 = p_1 + \alpha(p_1 - p_2) \quad (8)$$

Para el segundo hijo se utiliza el centro del segundo padre y se realiza el mismo proceso anterior, a diferencia de que en vez de sumar a cada coor-

denada del hijo el valor del producto entre n y la diferencia de las coordenadas de los padres, en este caso se realiza una resta de dicho producto a cada coordenada del hijo.

$$h_2 = p_2 - \alpha(p_1 - p_2) \quad (9)$$

En el caso de los radios cada hijo lleva el radio de uno de los padres y estos radios son modificados mediante la suma del producto entre n por la diferencia en valor absoluto de los radios de los padres. En (11) se presentan las ecuaciones para la asignación de radio para los hijos.

$$\begin{aligned} r_{h_1} &= r_{p_1} + \alpha * \text{abs}(r_{p_1} - r_{p_2}) \\ r_{h_2} &= r_{p_2} + \alpha * \text{abs}(r_{p_1} - r_{p_2}) \end{aligned} \quad (11)$$

2) *Operador de mutación*: Este operador (PEREZ-GODOY, 2010) modifica de forma aleatoria el centro y el radio de la RBF. El radio se modifica con una probabilidad inversamente proporcional a la cantidad de nodos de entrada de la red (u). 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 nodos de entrada de la red. La coordenada a mutar sufre un incremento o decremento en una cantidad aleatoria comprendida entre un 5% y un 50% del valor del radio.

3) *Operador de selección*: La selección es un proceso por medio del cual los individuos son escogidos de la

población de acuerdo al valor de su función de adaptación para someterse a la acción futura de otros operadores. La técnica utilizada es la versión determinista de la selección por torneo binario, descrito en (COELLO COELLO, 2004).

Donde para elegir el primer padre, se seleccionan de manera aleatoria dos individuos y se comparan en base a su *fitness* y el individuo con *fitness* más alto gana el torneo y es seleccionado. Luego se hace el mismo procedimiento para elegir el segundo padre.

E. Actualización de Red

En el CO2RBFN la población del algoritmo evolutivo es la capa oculta de la red RBF, mientras que en el modelo propuesto la capa oculta se construye a partir de las neuronas que forman parte del Frente Pareto obtenido mediante el MOEA.

El proceso de actualización consiste básicamente en recorrer una por una las neuronas del Frente Pareto, de manera a seleccionar aquellas que pueden formar parte de la capa oculta. Una neurona formará parte de la capa oculta si, y sólo si, su inclusión permite obtener un menor error en la salida de la red. Tanto el tamaño de la población como la cantidad de neuronas ocultas están predefinidos.

PREDICCIÓN DE SERIES

Una serie temporal se define como una colección de observaciones de una variable recogida secuencial-

mente en el tiempo (PEÑA, 2005). Estas observaciones se suelen recoger en instantes de tiempo equi-espaciados. Las series pueden tener una periodicidad anual, semestral, trimestral, mensual, otros., según los periodos de tiempo en los que se han recogidos los datos que la componen.

Toda serie temporal está compuesta de cuatro componentes principales que se detallan a continuación (HAN y KAMBER, 2000):

- *Tendencia.* Se puede definir como un cambio a largo plazo que se produce en relación al nivel medio, o el cambio a largo plazo de la media. La tendencia se identifica con un movimiento suave de la serie a largo plazo.
 - *Variaciones Cíclicas.* Representan ciclos que tienen las series. Dichas variaciones pueden o no ser periódicas, es decir, los ciclos pueden no ser completamente iguales después de periodos de tiempo idénticos.
 - *Movimiento Estacional.* estos movimientos se deben a eventos que ocurren con una frecuencia establecida y constante, aunque la amplitud puede ser variable.
 - *Movimiento Aleatorio.* Es el componente que no está sujeto a ninguna periodicidad en el tiempo y son considerados como la parte no predecible por lo cual son valores aleatorios.
- estocásticas* son aquellas series temporales que tienen un carácter probabilístico, por cuanto ha sido generada por alguna variable aleatoria (predomina el movimiento aleatorio) con una distribución de probabilidad determinada aunque habitualmente desconocida. El tipo de análisis así como los modelos en los que se basen el estudio de una serie dependerán en gran medida de si dichos modelos se centran sólo en una serie o en las posibles relaciones de varias series entre sí:
- *Modelos Univariantes:* representan la evolución de una serie temporal y pueden generar predicciones de su comportamiento futuro, basándose únicamente en la evolución histórica de la propia serie. Las predicciones obtenidas por dichos modelos se basan en la hipótesis de que las condiciones futuras serán análogas a las pasadas.
 - *Modelos de Regresión Dinámica o de Función de Transferencia:* estos modelos pretenden encontrar relaciones de dependencia entre la serie estudiada y ciertas variables. Las previsiones univariantes se pueden mejorar incorporando información de la evolución de otras variables y construyendo modelos que tengan en cuenta dichas dependencias.
 - *Modelos Multivariantes:* son modelos que representan conjuntamente las relaciones dinámicas entre un grupo de series y obtienen predicciones simultáneas de sus valores futuros.

Si para una serie se pueden predecir exactamente los valores, se dice que las series son *determinísticas*. Por otro lado, las series denominadas

El interés de este trabajo se centra en métodos utilizados para analizar una serie temporal sin tener en cuenta otras variables que puedan incluir en la misma, es decir, en los métodos univariantes.

En particular interesa la aproximación estocástica, frente a la determinista, por la complejidad en la obtención de la probabilidad que describe la distribución.

PRUEBAS EXPERIMENTALES

Para realizar las pruebas experimentales se obtuvieron series temporales a partir de los datos climatológicos registrados en el centro de datos del NOAA (*National Oceanic and Atmospheric Administration*, (NOAA, 2013)) como entrada de datos:

- **Temperatura Máxima:** Recoge datos desde Setiembre/2011 hasta Diciembre/2012 de la estación de Concepción - Paraguay, dando un total de 495 instancias.
- **Temperatura Mínima:** Recoge datos desde Setiembre/2011 hasta Diciembre/2012 de la estación de Concepción — Paraguay, dando un total de 495 instancias. Para las pruebas los datos se dividieron en dos conjuntos. Uno para entrenamiento con 445 instancias y otro con 50 instancias para realizar las pruebas. En ambos casos se debió realizar un gran trabajo de pre-procesamiento debido a datos faltantes dentro de la serie. Los datos faltantes fueron rellenados utilizando datos históricos de la fecha faltante y datos de los días adyacentes. La Tabla I muestra la división de los datos de las series temporales con temperaturas máxima y mínima.

Tabla 1: División de los datos de las series temporales.

| <i>Serie Temporal</i> | <i>Conjunto</i> | <i>Cantidad de Datos</i> |
|-----------------------|-----------------|--------------------------|
| Temperatura Máxima | CE | 495 |
| | CP | 50 |
| Temperatura Mínima | CE | 495 |
| | CP | 50 |

Los conjuntos de datos están formados de la siguiente manera: Una instancia por cada día del año. De esta forma

para predecir la fecha *dd/mm/yyyy*, la instancia se formará de la siguiente manera:

dd/mm/(yyyy - 3), dd/mm/(yyyy - 2), dd/mm/(yyyy - 1), (dd - 3)/mm/yyyy, (dd - 2)/mm/yyyy, (dd - 1)/mm/yyyy, dd/mm/yyyy

En la tabla II se especifican los parámetros utilizados en NSGA2 para la realización de las pruebas:

Tabla II: Parámetros de NSGA2

| Parámetro | Valor |
|--------------------------|-------|
| Tamaño población | 100 |
| Probabilidad de cruce | 0.6 |
| Probabilidad de mutación | 0.3 |
| Cantidad de generaciones | 30 |

A. Algoritmo Evolutivo Cooperativo

Competitivo para el diseño de Redes de Funciones de Base Radial (CO2RBFN).

El método seleccionado para la comparación con el MORBFN es el CO2RBFN. En el CO2RBFN los individuos cooperan entre sí para lograr la solución óptima, mientras que al mismo tiempo compiten por su supervivencia y maneja un SBRD, el cual representa el conocimiento experto, de manera a decidir la aplicación de los operadores genéticos (Eliminación, Nulo y Mutación) a los individuos de la población. Además utiliza los pasos de un Algoritmo Evolutivo clásico con un solo objetivo, el cual es el de minimizar el error de la red, también presenta el mismo método de entrenamiento de pesos utilizado en este trabajo el cual es el LMS. Los parámetros de configuraciones utilizados fueron los mismos que se indicaban en la literatura.

B. Error de la Predicción

Muchas medidas sobre la precisión de un modelo de predicción se han propuesto en el pasado, y varios autores han hecho recomendaciones acerca de

cuál debe ser utilizado al momento de comparar la exactitud de los métodos de predicción aplicados a datos de series temporales.

Algunos de los tipos de errores utilizados en la literatura (HYNDMAN and KOEHLER, 2006; SWANSON et al, 2011) son por ejemplo: Error Promedio (AE), Desviación Absoluta Media (MAD), Error Cuadrado Medio (MSE), Desviación Típica de los Errores (SDE), Error Porcentual (PE), Error Porcentual Medio (MPE) y Error Porcentual Absoluto Medio (MAPE).

El más comúnmente utilizado para evaluar predicciones es el MAPE (SWANSON et al, 2011). Además, se encuentra en diversos paquetes de software como: *Autobox*, *Nostradamus*, *SmartForecast*, etc.

El MAPE tiene valiosas propiedades estadísticas ya que hace uso de todas las observaciones y tiene una menor variabilidad de muestra en muestra. La fórmula matemática se presenta en la ecuación (12).

$$MAPE = \frac{\sum_{t=1}^n abs \left(\frac{f_{real_t} - f_{model_t}}{f_{model_t}} \right)}{n} \quad (12)$$

Donde n indica la cantidad de instantes de tiempos observados, f_{real} es el valor real observado en el tiempo t y $f_{modelot}$ es el valor predicho para el tiempo t .

C. Procedimiento de Comparación

El procedimiento de comparación de los resultados obtenidos fue el siguiente:

- 1) Primeramente se normalizaron las series temporales, y el resultado se utilizó como entrada de datos a los modelos.
- 2) Se aplicó validación cruzada (*Crossvalidation* 10-Fold) con 10 particiones, para luego realizar 5 corridas por cada partición generada, por lo tanto para cada serie temporal se obtuvieron 50 resultados de predicciones de CE y 50 resultados de predicciones de CP.
- 3) El resultado de las predicciones de los conjuntos de Entrenamiento y Prueba fueron desnormalizados para su posterior análisis.
- 4) En cada resultado del conjunto de Prueba se realizó el cálculo del error MAPE, para cada una de las predicciones.
- 5) Por cada corrida se obtuvo el promedio del error MAPE obtenido. Luego se obtuvo el error MAPE promedio obtenido en las 5 corridas, para finalmente obtener el promedio de todas las particiones.

El modelo propuesto se ha codificado utilizando el lenguaje de programación *Java*, en su versión 1.6,

y como entorno de desarrollo el *Eclipse Indigo*. Para el manejo de los archivos de datos se ha hecho uso del paquete keel. Dataset que se encuentra en el código fuente de la herramienta Keel (ALCALA-FERNANDEZ et al., 2008), este paquete ha sido adoptado debido a que se ha establecido que el formato del archivo de datos de entrada debe cumplir con el formato Keel.

Debido a que el modelo a comparar (CO2RBFN) solo se encuentra disponible en su versión compilada, se tuvo que programar, utilizando lenguaje de programación *Java*, para realizar las operaciones de preprocesamiento y postprocesamiento, como también tareas de pruebas en general.

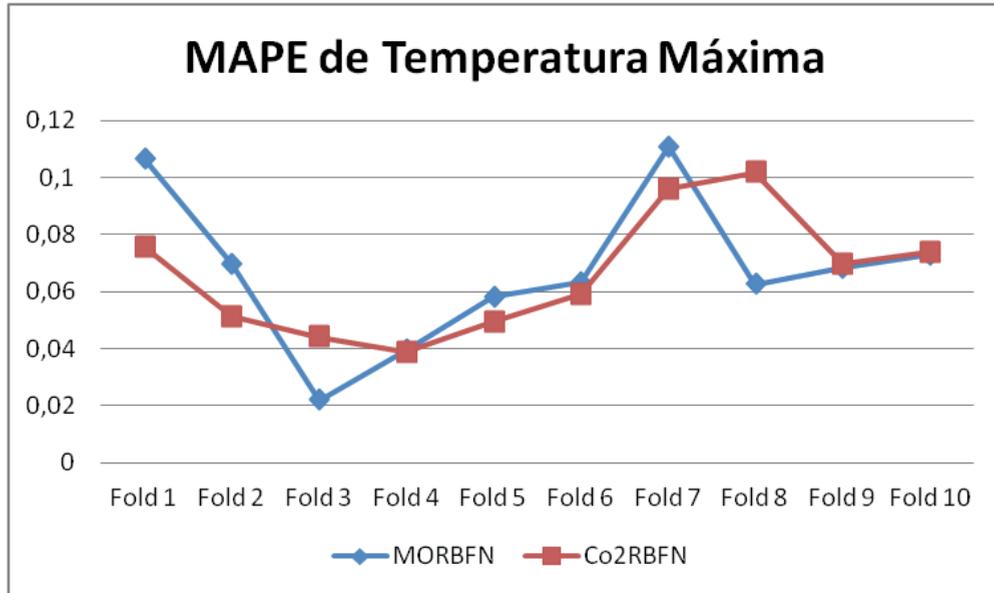
D. Resultados Obtenidos

Las tablas III, IV muestran el error MAPE promedio obtenido por cada modelo para cada una de las series temporales utilizadas para el estudio. Las figuras 2 y 3 presentan la comparación entre los errores obtenidos en cada una de las particiones para ambas series temporales.

- 1) *Temperatura Máxima*: En la tabla III se puede visualizar los resultados de los errores MAPE obtenidos en cada corrida de las pruebas y su desviación estándar, para ambos métodos comparados, correspondiente a la serie de Temperatura Máxima.

Tal y como se puede observar en la figura 2, existe gran semejanza en los errores obtenidos para cada par-tición de la validación cruzada por ambos métodos.

Fig. 2 Comparación de Errores en la serie de Temperatura Máxima



Con esto podemos observar que el modelo CO2RBFN obtuvo mejores resultados: promediando los errores y analizando tanto el mayor error como el menor error. Sin embargo, el modelo propuesto presenta mejores valores en cuanto a la desviación estándar de los errores, lo que indica su robustez en las predicciones.

Tabla III: Errores mape para temperatura mínima

| | MORBFN | CO2RBFN |
|-------------------|-------------|-------------|
| | <i>Test</i> | <i>Test</i> |
| <i>Fold 1</i> | 0,106570717 | 0,075701131 |
| <i>Fold 2</i> | 0,069680915 | 0,05128098 |
| <i>Fold 3</i> | 0,021791495 | 0,043927524 |
| <i>Fold 4</i> | 0,039462369 | 0,038674509 |
| <i>Fold 5</i> | 0,058085638 | 0,049448849 |
| <i>Fold 6</i> | 0,063158351 | 0,059006487 |
| <i>Fold 7</i> | 0,110660083 | 0,095935265 |
| <i>Fold 8</i> | 0,062479385 | 0,101658509 |
| <i>Fold 9</i> | 0,068147464 | 0,069666098 |
| <i>Fold 10</i> | 0,072776132 | 0,073747276 |
| <i>Promedios</i> | 0,067281255 | 0,065910663 |
| <i>Desv. Est.</i> | 0,02535727 | 0,02027236 |
| <i>Máximo</i> | 0,110660083 | 0,101658509 |
| <i>Mínimo</i> | 0,021791495 | 0,038674509 |

2) *Temperatura Mínima*: En la tabla IV se puede visualizar los resultados de los errores MAPE obtenidos en cada corrida de las pruebas y su desviación estándar, para ambos métodos comparados, correspondiente a la serie de Temperatura Mínima.

Fig. 3 Comparación de Errores en la serie de Temperatura Mínima

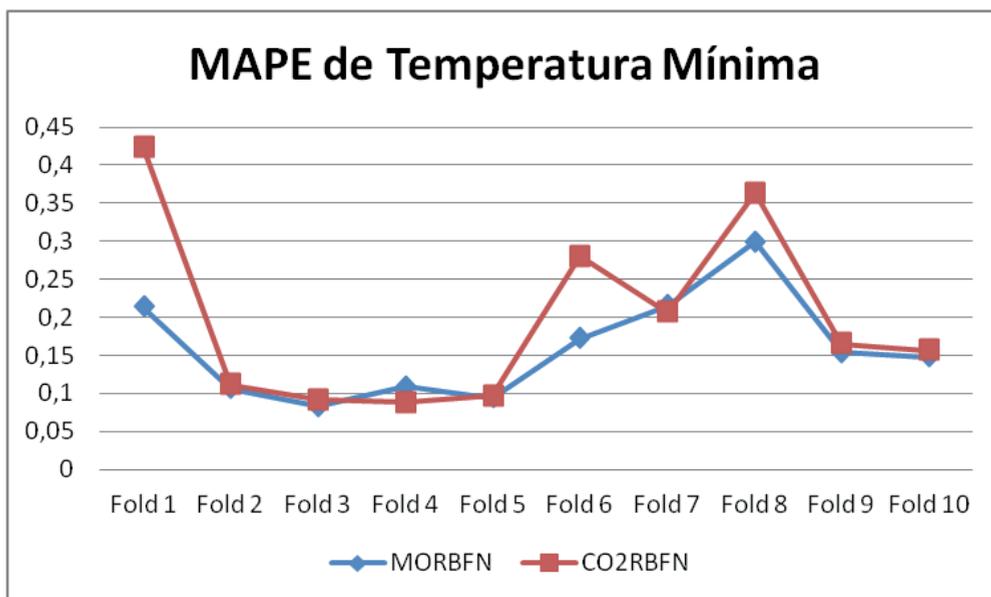


Tabla IV: Errores mape para temperatura minima

| | MORBFN | CO2RBFN |
|-------------------|-------------|-------------|
| | Test | Test |
| Fold 1 | 0,213311171 | 0,424202688 |
| Fold 2 | 0,106130112 | 0,111388772 |
| Fold 3 | 0,083250682 | 0,091551497 |
| Fold 4 | 0,100541592 | 0,007986722 |
| Fold 5 | 0,094158769 | 0,097024928 |
| Fold 6 | 0,172277108 | 0,28015736 |
| Fold 7 | 0,215399244 | 0,206461846 |
| Fold 8 | 0,299174473 | 0,363124387 |
| Fold 9 | 0,153885265 | 0,165011169 |
| Fold 10 | 0,148057901 | 0,15651559 |
| Promedios | 0,159418632 | 0,198342496 |
| Desv. Est. | 0,064330009 | 0,113663924 |
| Máximo | 0,299174473 | 0,424202688 |
| Mínimo | 0,083250682 | 0,087986722 |

CONCLUSIÓN

Este trabajo presenta un nuevo modelo, para el entrenamiento de redes neuronales con funciones de base radial, aplicado a la predicción de series temporales de temperaturas máximas y mínimas para una estación. El entrenamiento es realizado en dos fases.

La primera fase consiste en la utilización del LMS para determinar los pesos asociados a las RBF y la segunda fase consiste en el entrenamiento de los parámetros de las neuronas RBF (centro y radio) mediante el NSGA2. Para el diseño del algoritmo evolutivo se utilizan dos enfoques: el cooperativo-competitivo, donde cada individuo de la población es una potencial neurona que formará parte de la red neuronal, y el multi-objetivo, donde una población de individuos evoluciona teniendo en cuenta cuatro objetivos: aporte de la RBF a la salida de la red, error de la RBF, solapamiento con otras RBF e *incidencia* de la RBF sobre los datos de entrada. Son deseados valores mínimos para el segundo y tercer objetivo y valores máximos para el primer y cuarto objetivo.

Tanto el modelo propuesto como el CO2RBFN se centran en la asignación de los parámetros definidos en la segunda etapa, dejando la primera etapa al método LMS y de esta forma obtener el RBFN que obtenga el menor error.

La serie de temperatura máxima presenta menos variaciones bruscas que la serie de temperatura mínima.

Tanto el algoritmo propuesto como el modelo CO2RBFN se han evaluado con las mismas particiones generadas por la validación cruzada. Se han obtenidos resultados alentadores en cuanto a la serie de temperaturas mínimas se refiere.

Trabajos futuros podrían consistir en realizar comparaciones contra otros métodos clásicos utilizados en la predicción de la serie temporal, como el ARIMA, así como los métodos presentes en el KEEL utilizados para predicción. Además sería interesante la utilización de otras series temporales reales para realizar las comparaciones, distintas a las ya utilizadas, así como reestructurar el modelo propuesto de manera a darle un enfoque paralelo.

AGRADECIMIENTOS

Este trabajo fue posible gracias a la colaboración de Antonio J. Rivera, Profesor Titular en la Universidad de Jaén (España) en Arquitectura y Tecnología de Computadores, que nos ha facilitado todo lo necesario para realizar las comparaciones contra el CO2RBFN.

REFERENCIAS

AVALOS, M.; FERREIRA, A.; VON LÜCKEN, C.; DAVALOS, E. 2012. Radial Basis Neural Network design using a Competitive Cooperative Coevolutionary Multiobjective Algorithm. Conferencia Latinoamericana En Informática (CLEI), 2012 XXXVIII, vol., no.,

- pp.1,9, 1-5 Oct. 2012. doi: 10.1109/CLEI.2012.6427171. URL: <http://ieeexplore.ieee.org/stamp/stamp.6427171&isnumber=6426913>.
- ALCALA-FERNANDEZ, J.; et al. 2008.** Keel: a software tool to assess Evolutionary Algorithms for data mining problems. *Soft Computing*, vol. 13, no. 3, pp. 307–318. Disponible en: <http://www.keel.es>.
- BROOMHEAD, D.S.;LOWE, D. 1988.** Multivariable functional interpolation and adaptive network. *Complex Systems*, 2, 321–355.
- BUCHTALA, O.; KLIMEK, M and SICK, B. 2005.** Evolutionary optimization of radial basis function classifiers for data mining applications. *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics*, vol. 35, P. 928–947.
- COELLO COELLO, C. A. 2004.** Introducción a la Computación Evolutiva - Notas del Curso. Instituto Politécnico Nacional N° 2508.
- DASH, C. S. K., BEHERA, A. K., PANDIA, M. K., & DEHURI, S. 2013. Neural networks training based on Differential Evolution in Radial Basis Function Networks for classification of web logs. In *Distributed Computing and Internet Technology* (pp. 183-194). Springer Berlin Heidelberg.
- DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. 2002.** A fast and elitist multi-objective genetic algorithm: NSGA-II. *Evolutionary Computation*, IEEE Transactions on, 6(2), 182-197.
- DE LACERDA, E. 2003.** Model selection of RBF networks via Genetic Algorithms. PhD diss., Ph. D, Centro de Informática de la Universidad Federal de Pernambuco, Brasil.
- HAN, J.; KAMBER, M.. 2000.** *Data Mining: concepts and techniques*, 1st ed. Morgan Kaufmann.
- HAYKIN, S. 1999. *Neural Networks: a comprehensive foundation*. Prentice Hall.
- HSIEH, W. 2009.** *Machine learning methods in the environmental sciences*. Cambridge University Press. ISBN-10: 052179192.
- HYNDMAN, R.; KOEHLER, A. 2006.** Another look at measures of forecast accuracy. *International Journal of Forecasting*, pp. 679–688.
- ISHIBUCHI, H.; HITOTSUYANAGI, Y.; OHYANAGI, H.; NOJIMA, Y. 2011. Effects of the existence of highly correlated objectives on the behavior of MOEA/D. *Evolutionary Multi-Criterion Optimization*, Lecture Notes en Computer Science, Springer Berlin / Heidelberg, vol. 6576, pp. 166–181.
- MIAGKIKH, V., KASABOV, N., TOPCHY, A., LEBEDKO, O. 1997.** Adaptive training of Radial Basis Function Networks based on Cooperative Evolution and Evolutionary Programming. *Proceedings of the International Conference Neural Information Processing (ICONIP)*, pp. 253–258.

- NOAA, National Climatic Data Center, <http://www.ncdc.noaa.gov/>, 2013, [En línea; accedido 07-nov-2013].
- PASINI, A.; MARZBAN, C.; HAUPT, S. 2008.** Artificial intelligence methods in the environmental sciences. Springer. ISBN-10: 402091176
- PEÑA, D. 2005.** Análisis de series temporales. Alianza Editorial.
- PEREZ-GODOY, M.; PEREZ, P.; RIVERA, A. et al. 2010.** CO2RBFN: predicción de series temporales con un enfoque cooperativo competitivo. Congreso Español de Informática (CEDI), P. 265–272.
- PEREZ-GODOY, M. 2010.** Métodos híbridos evolutivos cooperativos-competitivos para el diseño de redes de funciones de base radial. Tesis de doctorado, Dept. de Informática, Universidad de Jaen.
- PLATT, J. 1991.** A resource-allocating network for function interpolation. *Neural Computation*, vol. 3, pp. 213–225.
- POTTER, M. A.; JONG, K. A. D. 2000.** Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, vol. 8, pp. 1–29.
- POWELL, M. 1987.** Algorithms for approximation, pp. 143–167.
- RIVERA, A. J.; ROJAS, I.; ORTEGA, J.; DEL JESUS, M. J. 2007.** A new hybrid methodology for cooperative-coevolutionary optimization of Radial Basis Function Networks. *Soft Comput.*, vol. 11, no. 7, pp. 655–668.
- SWANSON, D.; TAYMAN, J.; BRYAN, T. 2011.** Mape-r: a rescaled measure of accuracy for cross-sectional subnational population forecasts. *Journal of Population Research*, vol. 28, pp. 225–243, 2011.
- WIDROW, B.; LEHR, M. A. 1990.** 30 years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation. *Proceedings of the IEEE*, 78(9), 1415-1442.
- WHITEHEAD, B. A.; CHOATE, T. D. 1995.** Cooperative - competitive Genetic Evolution of Radial Basis Function centers and widths for time series prediction. *IEEE Transactions on Neural Networks*, vol. 7, pp. 869–880, 1995.
- XUE, S.; WEI, S.; XUE, Y. 2012.** Monthly mean temperature forecasting based on Hybrid Radial Basis Function Neural Network. *Advanced Science Letters*. American Scientific Publishers. Volumen=9, Número 1, P. 712-717.
- YANG, Y; SUN, T; CHIH-LI, H et al. 2013.** A novel self-constructing Radial Basis Function Neural-Fuzzy System. *Applied Soft Computing*. Elsevier.
- YEE, P. V.; HAYKIN, S. 2001.** Regularized radial basis function networks: theory and applications. John Wiley & Sons, Inc.