

# **P**REDICCIÓN DE SISTEMAS CAÓTICOS CON REDES NEURONALES: UN ESTUDIO COMPARATIVO DE LOS MODELOS DE PERCEPTRÓN MULTICAPA Y FUNCIONES DE BASE RADIAL

## **PREDICTING CHAOTIC SYSTEMS WITH NEURAL NETWORKS: A COMPARATIVE STUDY OF MULTI-LAYER PERCEPTRON AND RADIAL BASED FUNCTIONS MODELS**

■ Mauricio Paletta

Email(s): mpaletta@ucab.edu.ve, mpaletta@uneg.edu.ve

Facultad de Ingeniería, Escuela de Ingeniería en Informática,  
Universidad Católica Andrés Bello  
Departamento de Ciencia y Tecnología – Universidad Nacional  
Experimental de Guayana  
Ciudad Guayana, Estado Bolívar, 8050, Venezuela

Fecha de Recepción: 27 de mayo de 2008

Fecha de Aceptación: 25 de noviembre de 2008

### Resumen

La predicción de sistemas caóticos ha sido y sigue siendo un tema de mucho interés científico. Esto gracias a los beneficios que se pueden obtener al conocer comportamientos futuros de este tipo de sistemas, en base a data conocida del pasado. Sin embargo, la búsqueda de estrategias de solución con márgenes de error aceptables, basado en la diferencia entre los valores predichos versus los reales, presenta niveles de complejidad que deben ser tratados convenientemente. Basta citar, por ejemplo, problemas específicos como la predicción del clima, un terremoto, la bolsa de valores, para darse cuenta el beneficio que significaría el tener una buena estrategia automatizada de predicción. Ahora bien, al estar este tipo de problemas relacionados con manejo de conocimiento, la mayoría de los intentos de obtener una herramienta de predicción con resultados aceptables, están relacionados con técnicas heurísticas. En este sentido, las Redes Neuronales son una de las técnicas heurísticas más utilizadas para este fin, específicamente mediante la configuración de los modelos Perceptrón Multicapa y Funciones de Base Radial. Cabe preguntarse entonces, ¿cuál de estos modelos es el más idóneo para abordar este tipo de problemas? ¿Se puede garantizar una herramienta de predicción con márgenes de error reducidos al utilizar este tipo de técnica? En este artículo se presenta un estudio comparativo de los modelos de

Redes Neuronales antes indicados, dando respuesta a las dos preguntas anteriores así como también el aporte de conclusiones que sirven de ayuda al mundo científico a la hora de abordar un problema específico de predicción con este tipo de técnica.

**Palabras claves:** Predicción, Redes Neuronales, Perceptrón Multicapa, Función de Base Radial.

### Abstract

The prediction of chaotic systems has been and continues to be a subject with great scientific interest. This is because the benefits that can be obtained knowing the future behaviors of this kind of systems based on previous known data. However, the search for solution strategies with an acceptable error margin presents some levels of complexity that must be treated conveniently, based on the difference between the predicted values and the real values. Enough is to say, for instance, specific problems like climate and seismic predictions and the stock market, to realize of the benefit that would mean to have a good strategy based on prediction. Since these problems are related to managing knowledge, most attempts to obtain a predictive tool with acceptable results are related to technical heuristics. In this regard, Neural Networks is one of the most commonly used, specifically by configuring Multi-layer Perceptron and Radial Based Functions models. We must ask ourselves then: Which one of these models is most appropriate to solve this kind of problems? Can we guarantee a tool of prediction with low error margins using this kind of technique? This article presents a comparative study of patterns of Neural Networks above, responding to the previous two questions as well as providing conclusions that are intended to assist the scientific world to address a specific prediction problem with this technique.

**Keywords:** Prediction, Neural Networks, Multi-layer Perceptron, Radial Based Function.

## 1. Introducción

Predecir un terremoto o un desastre climático son algunos ejemplos específicos de problemas de predicción. En general, estos problemas consisten en determinar lo que puede ocurrir en el futuro sabiendo lo que ha ocurrido en el pasado, siguiendo una dinámica característica de sistemas de comportamiento caótico [1], es decir, con ausencia de reglas claras [2].

Diversas técnicas se han utilizado para obtener una herramienta automatizada que permita modelar problemas de predicción [3]. El objetivo principal que se busca en este tipo de herramientas es dar una buena respuesta (eficacia) en un tiempo razonable (eficiencia). La única información que se dispone para ello es data del pasado (experiencia), por lo tanto, lo convierte en un tipo de problema adecuado para ser abordado con técnicas para el manejo de conocimiento o técnicas heurísticas [2].

Sin embargo, las características propias de comportamiento caótico sobre este tipo de problema, aumentan considerablemente el nivel de complejidad para obtener una herramienta automatizada que logre un adecuado balanceo en la relación eficacia-eficiencia.

Por otro lado, de las técnicas heurísticas más comúnmente usadas para abordar problemas de predicción, están las Redes Neuronales Artificiales (RNAs). Más específicamente con la configuración y uso de los modelos Perceptrón Multi-Capa (MLP – de sus siglas en inglés Multi-Layer Perceptron) y Redes de Funciones de Base Radial (RBFN – de sus siglas en inglés Radial Based Functions Network).

El principal objetivo de este artículo es mostrar un estudio comparativo de la forma en la cual los modelos MLP y RBFN se comportan en problemas de predicción. Es importante mencionar que la idea no es fijar una posición en defensa del uso de los modelos neuronales MLP y RBFN para abordar este tipo de problemas, en relación a otros modelos de RNAs u otras técnicas de resolución de problemas. Este artículo puede ayudar a aquellos investigadores que requieran abordar un problema de predicción y han decidido usar una RNA para dar solución a la automatización del mismo.

En este artículo se incluyen, además, detalles de la experimentación realizada y resultados obtenidos sobre un par de casos de uso escogidos para este estudio, con niveles de dificultad baja y alta respectivamente.

El artículo está organizado de la manera que se explica a continuación. La sección 2 presenta el entorno introductorio de la tecnología de RNA. Algunos trabajos relacionados con problemas de predicción abordados con RNAs se presentan en la sección 3. La sección 4 muestra el estudio comparativo de los modelos MLP y RBF incluyendo los resultados de los ensayos realizados. Finalmente, la sección 5 presenta las conclusiones de la investigación, así como también algunos aspectos relacionados con trabajo futuro.

## 2. Un repaso a las redes neuronales

### 2.1. Introducción

Las RNAs [4][5][6] son estructuras cognitivas de procesamiento de información basadas en modelos de las funciones cerebrales. Formalmente hablando, se pueden describir como un sistema dinámico altamente paralelo con la topología de un grafo dirigido. Una RNA ejecuta el procesamiento de la información en términos de su respuesta o reacción a una entrada o estímulo externo (representado tanto con valores continuos como discretos).

Desde el punto de vista estructural, las RNAs son arreglos de unidades de procesamiento (neuronas artificiales) que están interconectadas entre sí formando la red. Cada una de estas unidades puede tener múltiples entradas pero sólo una salida. Las entradas, o vienen del exterior o son copias de las señales de salida de otras unidades, logrando así la propagación de la información entre las neuronas.

En este mismo orden de ideas, los lados o aristas que conectan una neurona con otra tienen asociado un peso que hace el papel de la sinapsis<sup>1</sup> en el sistema neurofisiológico. Estos pesos permiten almacenar la información o memoria que una RNA es capaz de asimilar para un problema específico. En otras palabras, mediante un ajuste adecuado de estos pesos sinápticos, las RNAs son capaces de "aprender" la asociación entre un conjunto característico de vectores de entrada (estímulos) y salida (respuesta) relativos a un problema específico. Este ajuste de pesos se conoce en la literatura como proceso de aprendizaje de la RNA.

El proceso de aprendizaje/entrenamiento se basa en un algoritmo, por lo general repetitivo, que va gradualmente haciendo ajustes en los pesos desde un valor inicial dado a éstos (comúnmente generados de forma aleatoria). Cabe resaltar que este ajuste se realiza basado en una expresión característica de cada RNA y que se conoce en la literatura como ley de aprendizaje. De esta manera se forma una dinámica que busca converger a un estado de configuración adecuada de los pesos, partiendo con un estado inicial específico, por lo general desordenado. La cantidad de ciclos (también llamado en la literatura como época – del inglés "epoch"), la manera de inicializar los pesos para empezar la dinámica y el tiempo que dura este proceso

dependen de la complejidad del problema a representar y del tipo de RNA que se utilice.

Continuando con el proceso de aprendizaje, cabe resaltar que, éste se realiza basado en un conjunto de patrones característicos asociados al problema que se quiere representar y que han sido seleccionados previamente. Además de la información de entrada, estos patrones pueden o no venir acompañados de la información de salida que corresponde con la entrada específica. Si es este el caso, entonces se dice que el aprendizaje es supervisado. Caso contrario se dice que es no supervisado. El hecho que el proceso de entrenamiento de una RNA sea supervisado o no, depende de la ley de aprendizaje correspondiente.

En otro orden de ideas, además del aprendizaje, una RNA debe también ejecutar un proceso de reconocimiento/ propagación. En este otro proceso se busca que la RNA genere una respuesta (salida) basado en un estímulo (entrada) dado. Es por este medio que los usuarios de una RNA pueden hacer uso de ella para lidiar con el problema específico por el cual ha sido entrenada.

Por otro lado, un modelo específico de RNA (también conocido en la literatura como paradigma neuronal) describe una serie de elementos o propiedades básicas (ver detalles a continuación) que permiten diferenciar la estructura y funcionamiento de una RNA con respecto a otra. Estos elementos son: 1) una función de transferencia no lineal asociada a cada unidad de procesamiento; 2) la topología de interconexión de las unidades y 3) la ley de aprendizaje o ajuste de pesos. MLP y RBF son dos ejemplos particulares de modelos neuronales, los cuales son los utilizados en este estudio y cuyo detalle, basado en los tres elementos antes mencionados, se describe en las siguientes sub-secciones.

### 2.2. El modelo MLP

MLP [7] es uno de los modelos neuronales más utilizados en aplicaciones de diversa índole. Se trata de una red con topología a capas en la cual la información fluye en una dirección, desde una capa de entrada (elementos sensores) hasta una capa de unidades que manifiestan la salida del sistema. En el recorrido, la información es procesada parcialmente por diferentes capas intermedias (escondidas) de unidades.

En términos matemáticos, se puede ver la relación entrada – salida como una correspondencia que la RNA a capas es capaz de aprender y, en cierta forma general-

<sup>1</sup> Puntos de transmisión de información entre una neurona y otra. Su funcionamiento es modificado por la experiencia (tanto para excitar como para inhibir la fuerza o peso de la conexión). Se consideran los sitios de almacenamiento de la memoria.

zar, ya que este tipo de redes provee una representación de esta correspondencia, razón por la cual son también llamadas redes de correspondencia o relacionales.

En lo que respecta a la topología, en MLP el número de unidades de cada capa es arbitrario y las salidas de las unidades de una capa sólo se conectan (son entrada) con todas las unidades de la capa siguiente (no hay interconexión entre las unidades de la misma capa).

Cabe destacar la importancia que tiene seleccionar una adecuada topología de la red (dado en este caso por el número de capas escondidas y el número de unidades en cada capa). En este sentido, es importante tener en cuenta que si la red es pequeña no será capaz de sintetizar un buen modelo de la data y, si la red es grande, podrá implementar muchas soluciones consistentes pero la mayoría, aproximaciones pobres del problema, comportamiento conocido en la literatura como "overfitting". Para revisar una discusión básica sobre "overfitting" se puede consultar [8]; para ver una revisión más profunda se puede consultar los artículos [9] y [10].

En general no hay forma de saber a priori el valor adecuado del número de unidades escondidas que debe tener la red, por lo que es necesario aplicar ensayo y error.

La recomendación es empezar con la red más pequeña posible e ir incrementando gradualmente. Para ello, es posible usar las cotas mínima y máxima cuyas expresiones se muestra en (1), con:

$N$ : número de unidades en la capa de entrada.  
 $n$ : número de unidades en la capa de salida.  
 $m$ : número de patrones de entrenamiento.

$$n_{ocultos\_min} = \sqrt{\frac{N+n}{2}}; \quad n_{ocultos\_max} = m \quad (1)$$

El aprendizaje es del tipo supervisado y provee un método para ajustar los pesos sinápticos de forma tal que la red aprenda la correspondencia  $(\vec{y}, \vec{s})$ , siendo  $\vec{y}$  el vector de entrada y  $\vec{s}$  el vector de salida asociado. El algoritmo de aprendizaje más popular es el que se conoce con el nombre de retro-propagación del error (en inglés backpropagation) [11].

La base de este algoritmo es el método de descenso de gradiente que se emplea para minimizar una función de calidad, siendo la más utilizada el error cuadrático medio cuya expresión se puede ver en (2). A esta ley

o regla de aprendizaje se le llama regla delta generalizada.

$$E(\vec{w}) = \frac{1}{2} \sum_{\mu=1}^m (s_i^{\mu} - O_i^{\mu})^2 \quad (2)$$

La expresión para calcular el cómputo o propagación de las unidades escondidas se observa en (3), mientras que en (4) se muestra la expresión para realizar el cómputo de las unidades de salida del modelo. Siendo:

$O$ : unidad de salida.

$V$ : unidad de la capa escondida.

$y$ : unidad de entrada.

$f$ : función de transferencia (los casos más comunes son la sigmoide, cuando el intervalo de los valores de entrada es  $[0, 1]$  ó tangente hiperbólica cuando el intervalo es  $[-1, +1]$ ).

$i$ : índice para las unidades de la capa de salida.

$j$ : índice para las unidades de la capa escondida.

$k$ : índice para las unidades de la capa de entrada.

$\mu$ : índice para los patrones.

$$V_j = f(h_j^{\mu}) = f\left(\sum_k w_{jk} y_k^{\mu}\right) \quad (3)$$

$$O_i = f\left(\sum_j W_{ij} V_j\right) \quad (4)$$

$$f(x) = \frac{1}{1 + \exp(-x)} \quad \text{ó} \quad f(x) = \tanh(x) \quad (5)$$

Las expresiones para el ajuste de los pesos sinápticos (para el entrenamiento) se originan por el gradiente de la expresión de error dada en (2) en relación a los pesos  $W_{ij}$  que representan las variables de este proceso. En este sentido, la expresión dada en (6) muestra el ajuste de pesos entre la capa de salida y la última capa escondida conectada con ésta y la expresión dada en (7) muestra la expresión para el ajuste de pesos entre las demás capas escondidas hasta llegar a la capa de entrada. Siendo:

W: la matriz de pesos sinápticos.  
 $\Delta W$ : proporción del peso a ajustar acorde a su valor actual.  
 $\eta$ : factor de ajuste.  
s: salida deseada (entrenamiento supervisado).  
f: derivada de f.

$$V_j = f\left(-\sum_k \frac{(y_k^\mu - w_{jk})^2}{2\sigma^2}\right) \quad (8)$$

$$O_i = \sum_j W_{ij} V_j \quad (9)$$

$$f(x) = \exp(x) \quad (10)$$

$$\Delta W_{ij} = -\eta \frac{\partial E}{\partial W_{ij}} = \eta \sum_{\mu} (s_i^\mu - O_i^\mu) f\left(\sum_j W_{ij} V_j\right) V_j^\mu \quad (6)$$

$$\Delta W_{ij} = \eta \sum_{\mu} \delta_i^\mu V_j^\mu ; \delta_i^\mu = f\left(\sum_j W_{ij} V_j\right) (s_i^\mu - O_i^\mu)$$

$$\Delta W_{ij} = -\eta \frac{\partial E}{\partial w_{jk}} = \eta \sum_{\mu} \delta_i^\mu W_{ij} f\left(\sum_k w_{jk} y_k^\mu\right) y_k^\mu \quad (7)$$

$$\Delta W_{ij} = \eta \sum_m \delta_j^\mu y_k^\mu ; \delta_j^\mu = f\left(\sum_k w_{jk} y_k^\mu\right) \sum_i \delta_i^\mu W_{ij}$$

Nótese que el factor  $(s_i - O_i)$  representa el error cometido basado en la salida dada por la red (O) y lo que ésta debería dar (s).

### 2.3 El modelo RBFN

Las RBFs fueron originalmente desarrolladas por Hardy para tratar problemas que involucraban la adaptación de contornos irregulares topográficos mediante un conjunto de datos geográficos [12][13]. Las redes neuronales basadas en esta técnica (conocidas en la literatura como RBFN) son unas de las mejores opciones en modelos de propagación hacia delante, como alternativa para lograr excelentes resultados en la aproximación de datos originados, tanto por funciones estocásticas como determinísticas [14].

Se trata de un modelo formado por tres capas de unidades de procesamiento: la capa de entrada, la capa de salida y una capa de unidades escondidas. Haciendo uso de la misma nomenclatura usada para presentar las expresiones de MLP, en (8) se observa la expresión de cómputo de las unidades escondidas y en (9) las que corresponden a las unidades de salida. La función de activación de este modelo que se muestra en (10), corresponde a la exponencial.

Tal como se puede apreciar en las expresiones de cómputo antes presentadas y las de ajuste de pesos que aparecen a continuación, este modelo tiene su basamento en la distribución Gaussiana. De allí el uso del término  $\sigma$  como desviación estándar. La expresión para el ajuste de peso entre la capa de salida y la capa escondida se muestra en (11) mientras que las expresiones para el ajuste de los pesos y las desviaciones estándares entre la capa escondida y la capa de entrada se pueden observar en (12) y (13) respectivamente.

$$\Delta W_{ij} = \eta \sum_{\mu} (s_i^\mu - O_i^\mu) \left(\sum_j W_{ij} V_j\right) V_j^\mu \quad (11)$$

$$\Delta W_{ij} = \eta \sum_{\mu} \delta_i^\mu V_j^\mu ; \delta_i^\mu = \left(\sum_j W_{ij} V_j\right) (s_i^\mu - O_i^\mu)$$

$$\Delta w_{jk} = \eta \sum_{\mu} (s_i^\mu - O_i^\mu) \left(\sum_j W_{ij} V_j\right) W_{ij} \left(\frac{y_k^\mu - w_{jk}}{\sigma_{jk}^2}\right) f\left(-\sum_k \frac{(y_k^\mu - w_{jk})^2}{2\sigma_{jk}^2}\right) \quad (12)$$

$$\Delta \sigma_{jk} = \eta \sum_{\mu} (s_i^\mu - O_i^\mu) \left(\sum_j W_{ij} V_j\right) W_{ij} \left(\frac{(y_k^\mu - w_{jk})^2}{\sigma_{jk}^3}\right) f\left(-\sum_k \frac{(y_k^\mu - w_{jk})^2}{2\sigma_{jk}^2}\right) \quad (13)$$

En la próxima sección se presentan algunos trabajos que involucran los modelos neurales MLP y RBF en problemas de predicción.

### 3. Trabajos relacionados

Hay una gran cantidad de trabajos relacionados con el uso de RNAs del tipo MLP y RBFN para una

diversidad de problemas de predicción. Se puede citar por ejemplo a Serrano y compañía [15] quienes plantean el uso de una RNA del tipo MLP para tratar de predecir la quiebra bancaria. Un trabajo similar y más reciente es el de Santana [16] quien presenta una solución basada en MLP para predecir series temporales tomando como caso particular una aplicación a la inflación colombiana.

Por otro lado, en [17] los autores presentan un estudio comparativo de modelos de predicción basados en RNAs (específicamente haciendo uso de MLP) con otros modelos basados en métodos estocásticos, siendo una de sus conclusiones que las redes neuronales alcanzaron resultados más acertados en comparación al método estadístico utilizado.

En este mismo orden de ideas, en [18] se tiene un modelo basado en MLP para predecir heladas basado en 24 años de datos meteorológicos recolectados en una región de Argentina. En [19] los autores presentan un interesante trabajo en la cual se busca predecir el éxito en estudios universitarios con una RNA del tipo MLP. Borracci y Rubio [20] presentan un trabajo relacionado con modelos de MLP para predecir resultados individuales de la cirugía cardíaca. Una de las conclusiones a las cuales llegaron los autores de este trabajo es que las RNAs resultaron modelos aplicables para predecir los resultados individuales pos-cirugía cardíaca. Al igual que el caso anterior, otro trabajo relacionado con el ámbito médico es el que se encuentra en [21] en la cual los autores presentan un modelo de predicción de cáncer de próstata basado en la biopsia transceral. En este caso los autores llegan a la conclusión de que la RNA presenta una capacidad de predicción significativamente superior a los métodos de diagnósticos unimodales y similar al método de regresión logística.

En lo que respecta al modelo RBF, se pueden citar los trabajos de Wang y compañía [22] sobre la predicción de tráfico automotor; Rivas y compañía [23] sobre la predicción de consumo de agua; Panda y compañía en [24] sobre la predicción del desgaste en brocas de taladros.

Los mencionados anteriormente son sólo algunos ejemplos de la gran cantidad de trabajos que basan el uso de RNAs en problemas de predicción, lo que muestra la importancia que tiene el considerar esta tecnología a la hora de abordar este tipo de problemas en una gran variedad de dominios. Lo que es un común en la gran mayoría de estos trabajos es el uso de los modelos MLP y RBF para abordar este tipo de pro-

blemas. En la próxima sección se presenta el estudio comparativo de estos modelos neuronales para tratar problemas de predicción.

#### 4. Estudio comparativo entre mlp y rbf para tratar problemas de predicción

##### 4.1. Implementación y experimentación

Con el objeto de hacer experimentos para realizar el análisis comparativo entre los modelos neuronales MLP y RBFN en problemas de predicción, tales modelos fueron implementados haciendo uso de C++. Los experimentos se llevaron a cabo en una plataforma de hardware Intel T2600 (2.16 GHz) con 2 GB RAM.

Para este estudio se escogieron dos casos de prueba cuyos resultados se muestran a continuación en esta sección. Por un lado se hace predicción sobre el mapa logístico [25] y por el otro se busca hacer predicción con la data del terremoto de Kobe<sup>2</sup>. La idea es hacer el estudio basado en la experimentación hecha con problemas de niveles de complejidad y razón de ser diferentes.

Para los ensayos que se realizaron en este estudio se tomaron en cuenta las siguientes consideraciones:

- Una sola capa de unidades escondidas para el modelo MLP con el objeto de homologar ambos modelos.
- Dado que es la más comúnmente utilizada, se usó la sigmoide como función de transferencia para MLP.
- Los pesos sinápticos fueron inicializados al azar en un rango de valores del intervalo [-1, +1].
- Se planteó como objetivo buscar una topología que tenga el menor número de unidades escondidas.
- No se hizo ningún tipo de pre-procesamiento de la data de entrada a fin de comparar el rendimiento de cada modelo sin contaminar el experimento.
- En MLP, el factor de aprendizaje se inicializa en un valor grande (1.0) y va decreciendo a razón de 0.01 por ciclo hasta alcanzar el valor de 0.125; la idea es realizar saltos grandes al comienzo del proceso hasta que el descenso de gradiente encuentre la dirección correcta,

<sup>2</sup> Ocurrido en la ciudad de Kobe, Japón, el 17 de enero de 1995.

para lo cual hay que hacer saltos pequeños a fin de evitar las transiciones.

- En MLP se hace uso de un factor de momento [26]. El factor de aprendizaje para este término se inicializa en un valor pequeño (0.0) y va creciendo a razón de 0.01 por ciclo hasta alcanzar el valor de 0.9. La idea es darle al algoritmo algo de tiempo para encontrar la dirección correcta y acelerar el entrenamiento cuando esto ocurra.
- En ambos modelos no se considera explícitamente el umbral en las expresiones. En su defecto se agrega una conexión adicional que se ajusta de la misma manera que el resto de los pesos sinápticos [27].

### Caso de estudio 1: El mapa logístico

El mapa logístico se trata de un sistema dinámico del tipo mapa (estados del sistema medidos de forma continua y el tiempo de forma discreta) con una sola variable de estado en el intervalo [0, 1]. La regla de transición entre estados viene dada por la expresión mostrada en (14), siendo  $\lambda$  una constante. Dependiendo del valor de la constante  $\lambda$ , el sistema puede tomar diferentes órbitas (por ejemplo: punto fijo para  $\lambda = 2.0$ ; 2-periódico para  $\lambda = 3.2$  y caótico para  $\lambda = 4.0$ ) [28]. La Figura 1 muestra las órbitas generadas por el mapa logístico dependiendo del valor de  $\lambda$ .

$$x_{t+1} = \lambda x_t (1 - x_t) \tag{14}$$

La idea original del mapa logístico es servir como un modelo demográfico de tiempo discreto en la cual,  $\lambda$  representa una tasa que combina tanto la reproducción como el hambre (dependiente de la densidad de la mortalidad). Para este estudio se consideró el caso cuando  $\lambda$  es igual a 4.0 (comportamiento caótico).

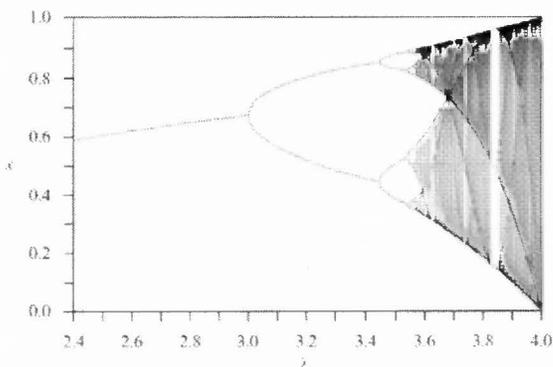


Figura 1. Dinámica del mapa logístico basado en diferentes valores de  $\lambda$ . (Fuente: Wikimedia Commons)

Para este caso, la RNA se entrenó con 100 términos (que hacen el papel de la data que se conoce) y luego se probó con los 50 términos siguientes de la serie caótica. Por otro lado, hay que tener en cuenta que la predicción puede variar dependiendo de la cantidad de data previa que se suministre. Es decir, no es igual predecir el siguiente término conociendo el anterior, a predecir el término que sigue a dos previos dados. En este sentido, se hicieron tres ejercicios de predicción basado en 1, 2 y 3 términos previos conocidos para estimar el término siguiente. Dado que los valores de la data del mapa logístico están en el intervalo [0, 1], al igual como ocurre con las funciones de transferencia sigmoide (caso MLP) y exponencial (caso RBFN), esta data no fue pre-procesada y por ende fue usada tal cual fue generada de la expresión dada en (14).

x	x'	ε	x	x'	ε
0.886594	0.890937	-0.00434	0.676563	0.717201	-0.04064
0.402181	0.365818	0.036363	0.875301	0.885430	-0.01013
0.961725	0.920343	0.041382	0.436595	0.411003	0.025592
0.147239	0.134120	0.013119	0.983919	0.927500	0.056419
0.502237	0.484542	0.017695	0.063288	0.095368	-0.03208
0.999980	0.932205	0.067775	0.237130	0.191938	0.045192
0.000080	0.074366	-0.07429	0.723598	0.768145	-0.04455
0.000320	0.081402	-0.08108	0.800016	0.840390	-0.04037
0.001281	0.081664	-0.08038	0.639961	0.687098	-0.04714
0.005118	0.082720	-0.07760	0.921643	0.906245	0.015398
0.020368	0.087092	-0.06672	0.288868	0.238941	0.049927
0.079813	0.107129	-0.02732	0.821694	0.850694	-0.02900
0.293772	0.237731	0.056041	0.586053	0.618300	-0.03225
0.829880	0.856182	-0.02630	0.970379	0.923589	0.046790
0.564717	0.589409	-0.02469	0.114973	0.117527	-0.00255
0.983247	0.927504	0.055743	0.407016	0.358220	0.048796
0.065889	0.096366	-0.03048	0.965416	0.921592	0.043824
0.246192	0.198640	0.047552	0.133553	0.126799	0.006754
0.742326	0.786388	-0.04406	0.462867	0.430388	0.032479
0.765112	0.813976	-0.04886	0.994484	0.930625	0.063859
0.718862	0.772982	-0.05412	0.021941	0.080967	-0.05903
0.808398	0.846181	-0.03778	0.085837	0.109458	-0.02362
0.619562	0.661889	-0.04233	0.313876	0.256290	0.057586
0.942820	0.914300	0.028520	0.861431	0.875498	-0.01407
0.215643	0.177665	0.037978	0.477471	0.467218	0.010253

Tabla 1. Resultados de la predicción conocido 1 término previo con MLP (2 unidades escondidas)

La Tabla 1 muestra los resultados obtenidos al realizar el ensayo de predicción con una RNA del tipo MLP (2 unidades escondidas) conociendo un término previo, es decir, un patrón de entrenamiento de la forma  $x_t \rightarrow x_{t+1}$ . Los valores mostrados en la tabla corresponden al reconocimiento de los 50 términos siguientes a los primeros 100 utilizados en el entrenamiento, siendo  $x$  el valor deseado,  $x'$  el valor predicho y  $\epsilon$  el error o diferen-

cia. El proceso de entrenamiento para este caso duró 10.000 ciclos y se logró un error promedio de 0,000953. La desviación estándar para las diferencias entre los valores reales y los predichos es de 0,044519.

La Figura 2 muestra un par de gráficas con los resultados expresados en la Tabla 1. En la gráfica de la izquierda se puede observar qué tan cerca están los valores predichos (ordenada) con los reales (abscisa). Nótese que el 100% de eficiencia, es decir, cuando los valores predichos coinciden en su totalidad con los valores reales, corresponde a la recta  $x = y$ . La gráfica de la derecha da una idea del desorden de la data ya que se muestra según la escala temporal de ocurrencia de los términos, de izquierda a derecha. Los valores que aparecen en el eje de abscisas corresponden también a los valores de los términos, al igual que el eje de ordenadas.

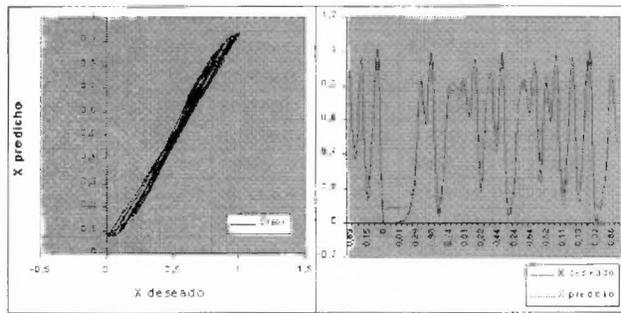


Figura 2. Predicción (1 término previo) con MLP (2 unidades escondidas)

Por otro lado, en la misma gráfica de la derecha también se pueden notar los intervalos en los cuales hay mayor diferencia entre los valores predichos (en azul) y los valores reales o deseados (en magenta). Nótese que estas diferencias se observan en su gran mayoría cerca de los picos (mínimos y máximos) de la curva.

x	x'	D	x	x'	D
0,961725	0,958980	0,002745	0,875301	0,869996	0,005305
0,147239	0,162198	-0,01496	0,436595	0,435742	0,000853
0,502237	0,500865	0,001372	0,983919	0,981723	0,002196
0,999980	0,998119	0,001861	0,063288	0,084677	-0,02139
0,000080	0,026845	-0,02677	0,237130	0,248116	-0,01099
0,000320	0,029288	-0,02897	0,723598	0,718364	0,005234
0,001281	0,030161	-0,02888	0,800016	0,793737	0,006279
0,005118	0,033658	-0,02854	0,639961	0,634041	0,005920
0,020368	0,047564	-0,02720	0,921643	0,917314	0,004329
0,079813	0,102026	-0,02221	0,288868	0,294845	-0,00598
0,293772	0,301419	-0,00765	0,821694	0,816658	0,005036
0,829880	0,824915	0,004965	0,586053	0,580977	0,005076
0,564717	0,560077	0,004640	0,970379	0,967412	0,002967

0,983247	0,980709	0,002538	0,114973	0,132307	-0,01733
0,065889	0,087065	-0,02118	0,407016	0,409122	-0,00211
0,246192	0,256617	-0,01043	0,965416	0,962759	0,002657
0,742326	0,73704	0,005286	0,133553	0,149502	-0,01595
0,765112	0,758633	0,006479	0,462867	0,462800	0,000067
0,718862	0,712359	0,006503	0,994484	0,992548	0,001936
0,808398	0,802190	0,006208	0,021941	0,046796	-0,02486
0,619562	0,613920	0,005642	0,085837	0,107569	-0,02173
0,942820	0,939037	0,003783	0,313876	0,320429	-0,00655
0,215643	0,225967	-0,01032	0,861431	0,856812	0,004619
0,676563	0,671652	0,004911	0,477471	0,475191	0,002280
0,961725	0,958980	0,002745	0,875301	0,869996	0,005305

Tabla 2. Resultados de la predicción conocido 1 término previo con RBFN (1 unidad escondida)

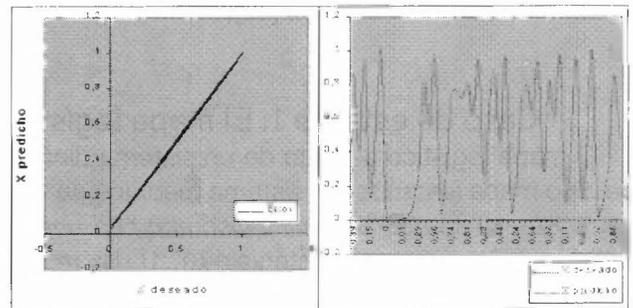


Figura 3. Predicción (1 término previo) con RBFN (1 unidad escondida)

x	x'	D	x	x'	D
0,402181	0,398619	0,003562	0,875301	0,876003	-0,00070
0,961725	0,921574	0,040151	0,436595	0,448295	-0,01170
0,147239	0,114976	0,032263	0,983919	0,928123	0,055796
0,502237	0,465703	0,036534	0,063288	0,082245	-0,01896
0,999980	0,928794	0,071186	0,237130	0,178295	0,058835
0,000080	0,066763	-0,06668	0,723598	0,781304	-0,05771
0,000320	0,081861	-0,08154	0,800016	0,826949	-0,02693
0,001281	0,089018	-0,08774	0,639961	0,725946	-0,08599
0,005118	0,090013	-0,08490	0,921643	0,912383	0,00926
0,020368	0,094133	-0,07377	0,288868	0,254850	0,034018
0,079813	0,113063	-0,03325	0,821694	0,854412	-0,03272
0,293772	0,239728	0,054044	0,586053	0,586806	-0,00075
0,829880	0,864681	-0,03480	0,970379	0,926068	0,044311
0,564717	0,556230	0,008487	0,114973	0,116074	-0,00110
0,983247	0,929123	0,054124	0,407016	0,336279	0,070737
0,065889	0,092417	-0,02653	0,965416	0,921802	0,043614
0,246192	0,184354	0,061838	0,133553	0,108693	0,02486
0,742326	0,798995	-0,05667	0,462867	0,409386	0,053481
0,765112	0,798154	-0,03304	0,994484	0,928475	0,066009
0,718862	0,802386	-0,08352	0,021941	0,070700	-0,04876
0,808398	0,863968	-0,05557	0,085837	0,105790	-0,01995
0,619562	0,702608	-0,08305	0,313876	0,258212	0,055664
0,942820	0,918757	0,024063	0,861431	0,882161	-0,02073
0,215643	0,184428	0,031215	0,477471	0,429669	0,047802
0,676563	0,715012	-0,03845	0,997970	0,931943	0,066027

Tabla 3. Resultados de la predicción conocido 2 términos previos con MLP (2 unidades escondidas)

Haciendo uso de los mismos 100 patrones para entrenar la MLP descrita anteriormente, se entrenó la RBFN con una unidad escondida. Luego de 920 ciclos de entrenamiento se alcanzó un error promedio de 0.000100 obteniendo los resultados que se muestran en la Tabla 2 y la Figura 3. La desviación estándar en este caso es de 0,011971. Hay que resaltar que la topología de esta red tiene 1 unidad en cada una de las tres capas, es decir, hay sólo un peso sináptico entre la capa de entrada y la escondida y un segundo peso sináptico entre la capa escondida y la de salida. Por lo tanto, estos resultados se logran ajustando solo dos pesos sinápticos.

En el experimento de predicción con dos términos previos, el patrón de entrenamiento tiene la forma  $x_i, x_{i+1} \rightarrow x_{i+2}$ . Para la red MLP con 2 unidades escondidas se hizo un entrenamiento de 7.000 ciclos obteniendo un error promedio de 0,000995 y una desviación estándar de 0,050554. Los resultados de este experimento se muestran en la Tabla 3 y la Figura 4.

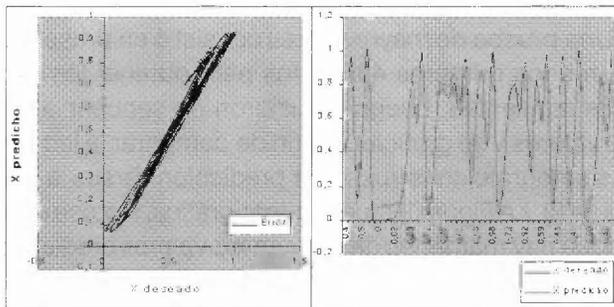


Figura 4. Predicción (2 términos previos) con MLP (2 unidades escondidas)

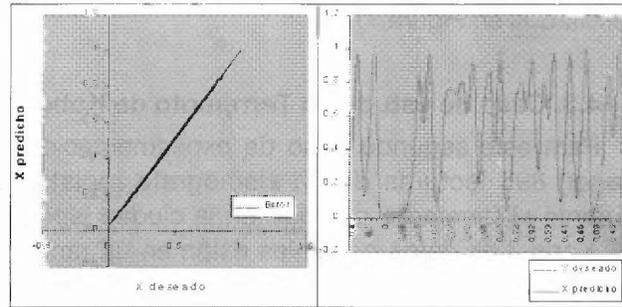


Figura 5. Predicción (2 términos previos) con RBFN (1 unidad escondida)

El ensayo de este mismo caso (predicción basado en dos términos previos conocidos) con RBFN y una unidad escondida, dio como resultado los datos expresados en la Tabla 4 y las gráficas mostradas en la Figura 5. Se realizaron 990 ciclos de entrenamiento y se alcanzó un error promedio de 0,000100. La desviación estándar de las diferencias es, en este caso de 0,011996.

X	x'	D	x	x'	D
0,402181	0,402607	-0,00043	0,875301	0,871717	0,003584
0,961725	0,957881	0,003844	0,436595	0,435640	0,000955
0,147239	0,163058	-0,01582	0,983919	0,980700	0,003219
0,502237	0,499249	0,002988	0,063288	0,085393	-0,0221
0,999980	1,000000	-0,00002	0,237130	0,246344	-0,00921
0,000080	0,027315	-0,02724	0,723598	0,720314	0,003284
0,000320	0,027405	-0,02709	0,800016	0,795364	0,004652
0,001281	0,031824	-0,03054	0,639961	0,633768	0,006193
0,005118	0,035319	-0,03020	0,921643	0,916711	0,004932
0,020368	0,049233	-0,02887	0,288868	0,294864	-0,00600
0,079813	0,10372	-0,02391	0,821694	0,815326	0,006368
0,293772	0,303202	-0,00943	0,586053	0,582361	0,003692
0,829880	0,826896	0,002984	0,970379	0,966697	0,003682
0,564717	0,561438	0,003279	0,114973	0,132498	-0,01753
0,983247	0,97995	0,003297	0,407016	0,407446	-0,00043
0,065889	0,087325	-0,02144	0,965416	0,964752	0,000664
0,246192	0,254848	-0,00866	0,133553	0,150341	-0,01679
0,742326	0,738994	0,003332	0,462867	0,461156	0,001711
0,765112	0,760220	0,004892	0,994484	0,994519	-3,5E-05
0,718862	0,712013	0,006849	0,021941	0,04741	-0,02547
0,808398	0,801750	0,006648	0,085837	0,105724	-0,01989
0,619562	0,613665	0,005897	0,313876	0,322223	-0,00835
0,94282	0,938392	0,004428	0,861431	0,858800	0,002631
0,215643	0,226050	-0,01041	0,477471	0,476455	0,001016
0,676563	0,670169	0,006394	0,997970	0,995166	0,002804

Tabla 4. Resultados de la predicción conocido 2 términos previos con RBFN (1 unidad escondida)

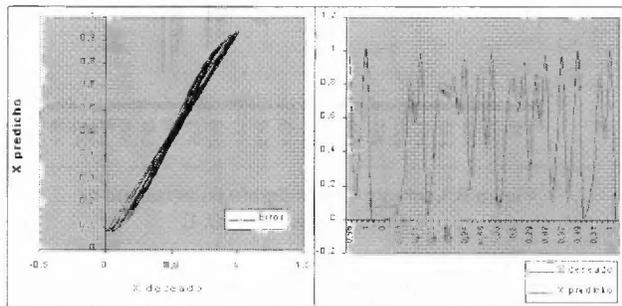


Figura 6. Predicción (3 términos previos) con MLP (2 unidades escondidas)

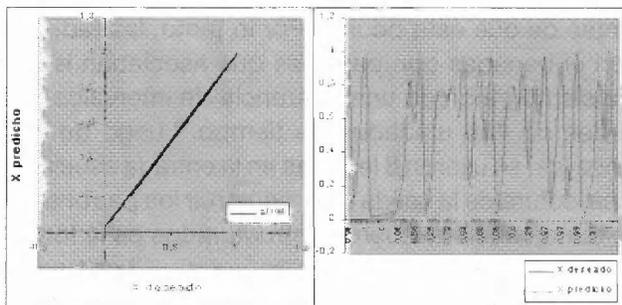


Figura 7. Predicción (3 términos previos) con RBFN (1 unidad escondida)

Finalmente, para la predicción a tres intervalos ( $x_t, x_{t+1}, x_{t+2} \rightarrow x_{t+3}$ ), se mantiene la misma tendencia de dos unidades escondidas en MLP con un entrenamiento de 10.000 ciclos y un error promedio igual a 0,000982 (ver resultados en Figura 6). Para el modelo RBFN en este mismo caso, se mantiene una sola unidad escondida; se hicieron 1070 ciclos de entrenamiento y el error promedio alcanzado es igual a 0,000100 (ver resultados en Figura 7).

### 4.3. Caso de estudio 2: Terremoto de Kobe

Para este segundo caso de experimentación, se tienen 895 lecturas de un sismógrafo registradas durante el terremoto ocurrido en la ciudad japonesa de Kobe. Los datos originales están en un rango de valores  $\pm 42.500$ . Por lo tanto, para ser utilizados con los modelos MLP y RBFN, los 895 valores fueron normalizados adecuadamente a la escala  $[0, 1]$ . La Figura 8 muestra la correlación de estos datos luego de su normalización.

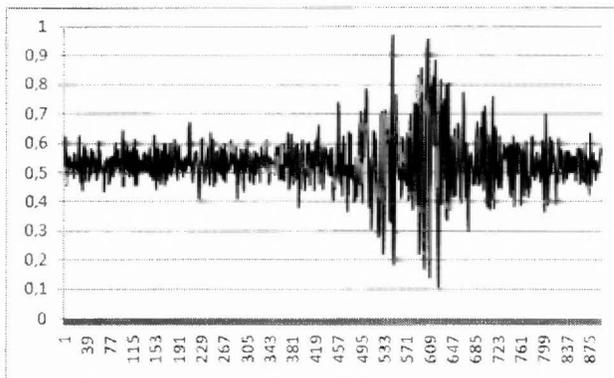


Figura 8. Lecturas de sismógrafo en terremoto de Kobe normalizados a  $[0, 1]$

Como se observa en la Figura 8, el terremoto ocurrió aproximadamente entre las lecturas 440 y 800. En este sentido, la idea con este experimento es tratar de predecir el terremoto conociendo las lecturas previas antes de que éste ocurra. Por lo tanto, las redes fueron entrenadas con patrones que asociaban lecturas de sismógrafo con una diferencia de intervalos por el orden de 446 unidades de tiempo. Luego de varios ensayos, se usaron 8 lecturas en la entrada asociadas a una lectura de la salida para construir los patrones. Por lo tanto, los datos fueron estructurados de la forma  $x_t, x_{t+1}, x_{t+2}, x_{t+3}, x_{t+4}, x_{t+5}, x_{t+6}, x_{t+7}, x_{t+8} \rightarrow x_{t+446}$ . Los primeros 446 datos fueron separados al azar en parejas de 223

patrones, un grupo fue usado para el entrenamiento y el otro para probar las redes ya entrenadas.

Por otro lado, para este experimento se hizo caso omiso sobre el criterio de reducir la topología de la red, es decir, aunque se mantiene una sola capa de unidades escondidas, no se buscó reducir la cantidad de unidades para esta capa.

En relación al entrenamiento con la red MLP, se realizaron unos 700.000 ciclos en una red con 24 unidades escondidas. El error promedio obtenido fue de 0,00394049 y la desviación estándar de las diferencias entre las salidas reales y las predichas fue de 0,06104946. Al probar la red con el otro conjunto de 223 patrones se obtuvo un error promedio de 0,14352885 y una desviación estándar de 0,15274571.

Una prueba de mayor interés consistió en presentar a la red los primeros 446 datos para obtener los 446 datos siguientes. Luego se unieron en secuencia los 892 valores y se graficaron a fin de comparar la lectura del sismógrafo construida por predicción versus la real (Figura 8). La Figura 9 muestra esta gráfica. Nótese que, aunque hay márgenes de error en los valores predichos versus los reales, la red es capaz de identificar una situación anormal en el intervalo adecuado.

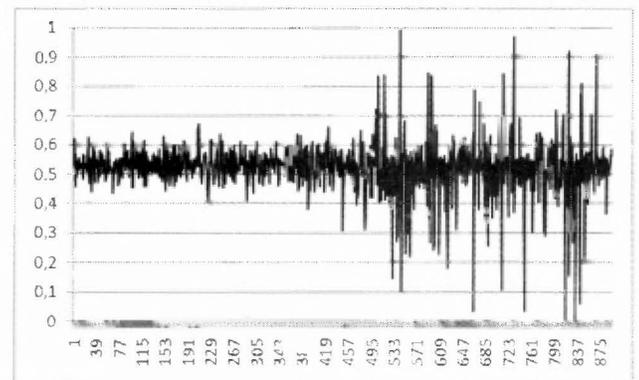


Figura 9. Lecturas de sismógrafo de terremoto de Kobe predichas con MLP

Con respecto a la red RBFN, luego de varios intentos de ensayo y error, no se pudo obtener una configuración adecuada que permitiera manejar este problema. El entrenamiento quedaba estancado por lo general en un error promedio alto con valores que estaban alrededor del 0,008. Las pruebas con una red entrenada bajo estas condiciones arrojaron resultados no satisfactorios.

#### 4.4. Análisis de los resultados

Con respecto al primer caso de estudio, se evidencian mejores resultados obtenidos con el modelo RBFN versus los resultados obtenidos con el modelo MLP, desde diferentes puntos de vista:

1. En relación a la topología de la red, se tiene un menor número de unidades escondidas, lo que implica un menor número de variables asociadas a los pesos sinápticos.
2. Menor cantidad de ciclos de entrenamiento en una diferencia que es de casi 10 a 1.
3. Más eficaz en las predicciones alcanzando un error promedio más bajo en una proporción aproximada de 10 a 1.

Otra observación interesante a considerar es que RBFN mantiene una respuesta homogénea en todos los ensayos. Se puede notar que todos los errores promedios alcanzados están en el orden de 0.000100 y ver en detalle las gráficas de los experimentos correspondientes (Figura 3, Figura 5 y Figura 7).

En relación al segundo caso de estudio, es de notar cómo fue posible obtener un resultado satisfactorio con MLP, mientras que no fue posible converger adecuadamente el entrenamiento de la red RBFN para modelar este problema.

En base a estos hechos, quiere decir entonces que no es posible fijar una posición sobre cuál de los dos modelos es el más adecuado para representar problemas de predicción. La decisión depende de varios factores, uno de los cuales es el problema específico a tratar. Criterios específicos como minimización de la topología de la red, facilidad de implementación y configuración, rapidez en el entrenamiento y otros, son otros factores que pudieran ser utilizados a la hora de hacer un cuadro comparativo entre los dos modelos sobre un problema específico.

En este sentido, sería recomendable, de ser posible, intentar experimentar con los dos modelos y, luego de analizar los resultados y fijar los criterios que se deseen, tomar la decisión de cuál modelo seguir.

#### 5. Conclusiones y trabajo futuro

La automatización de procesos que tienen que ver con la predicción de sistemas dinámicos ha sido y sigue siendo un interés para entender el comportamiento de diversos fenómenos que ocurren en la naturaleza. Siendo ésta una tarea de alta complejidad, la mayoría

de las propuestas de solución relacionados con esta área tienen que ver con el uso de técnicas heurísticas, donde las RNAs tienen principal participación. Muchos son los trabajos que se han propuesto para predecir algún tipo de problema sobre una gran variedad de dominios (medicina, ambiente, banca, industria, etc.), haciendo uso de una RNA. La gran mayoría de estos trabajos tienen que ver con un modelo basado en una RNA a capas, ya sea MLP o RBFN.

En este artículo se presentó un estudio comparativo sobre los modelos MLP y RBFN para tratar problemas de predicción. Para el estudio se realizaron experimentos basados en dos casos de estudio de diferente complejidad, el primero de ellos relativo a la predicción de la serie caótica generada por el mapa logístico. El segundo tiene que ver con la predicción de terremotos, basado en la data del terremoto de la ciudad japonesa de Kobe.

En relación a las dos interrogantes formuladas como motivación de este artículo, los resultados muestran, por un lado, que con una adecuada configuración y adaptación del modelo neuronal al problema tratado, las RNAs son, en efecto, una adecuada opción para tratar de automatizar problemas de predicción. Por otro lado, los resultados muestran que no se puede fijar una decisión de cuál modelo neuronal es mejor para tratar este tipo de problemas. Varios criterios, siendo el problema específico uno de ellos, pueden ser usados para evaluar la conveniencia de usar MLP y/o RBFN. La recomendación es experimentar con ambos y analizar los resultados obtenidos.

Actualmente se está llevando a cabo una investigación en la Universidad Nacional Experimental de Guayana, en colaboración con la Universidad Politécnica de Madrid, que tiene que ver con una propuesta de modelo predictivo del comportamiento que los nodos de un sistema basado en Grid Computing [29] [30] tienen, en base a la cooperación entre sí para el mejor aprovechamiento de los recursos. El diseño de este modelo está basado en una red RBFN; los datos para el entrenamiento y prueba de la red se están tomando de una base de datos generada por AMBLE (de sus siglas en inglés An Agents-Based Cooperative Awareness Model to Cover Load Balancing Delivery in Grid Environments) [31]. Resultados preliminares de esta investigación se pueden consultar en [32].

## 6. Referencias

- [1] Devaney R. *An Introduction To Chaotic Dynamical Systems*, Second Edition. Addison-Wesley Studies in Nonlinearity, ISBN: 978-0201130461, 1989.
- [2] Paletta M. "¿Sistemas Inteligentes, realidad o fantasía? ¿Qué hay detrás de los programas que se dice son Inteligentes?". Guayana Sustentable 8, Publicaciones UCAB (Universidad Católica Andrés Bello), Primera Edición, ISSN: 1856-7916, pp. 77-95, julio 2007.
- [3] Shepperd M. y Kadoda G. "Comparing Software Prediction Techniques Using Simulation". IEEE Transactions on Software Engineering, 27 (11), ISSN: 0098-5589, pp. 1014-1022, noviembre 2001.
- [4] Hertz J., Krogh A. y Palmer R.G. *Introduction to the theory of Neural Computation*. Addison Wesley Publishing Company, ISBN: 0-201-50395-6, 1991.
- [5] Muller B., Reinhardt J. y Strickland M.T. *Neural Networks An Introduction*. (2nd ed.). Berlin, Heidelberg, New York: Springer-Verlag. ISBN: 3-540-60207-0, 1995.
- [6] Ritter H., Martinetz T. y Schulten K. *Neural Computation and Self-Organizing Maps: An Introduction*. Addison Wesley, ISBN: 978-0201554434, 1992.
- [7] Haykin S. *Neural Networks: A Comprehensive Foundation*, 2. Prentice Hall, ISBN: 0-132-73350-1, 1998.
- [8] Smith M. *Neural Networks for Statistical Modeling*. International Thomson Computer Press, ISBN: 1-850-32842-0, 1996.
- [9] Geman S., Bienenstock E. y Doursat R. "Neural Networks and the Bias/Variance Dilemma". *Neural Computation*, 4, pp. 1-58, enero 1992.
- [10] Lawrence S., Giles C.L. y Tsoi A.C. (1997, julio). *Lessons in Neural Network Training: Overfitting May be Harder than Expected*. Presentado en Fourteenth National Conference on Artificial Intelligence (AAAI-97), AAAI Press, pp. 540-545. Disponible: [http://clgiles.ist.psu.edu/papers/AAAI-97.overfitting.hard\\_to\\_do.pdf](http://clgiles.ist.psu.edu/papers/AAAI-97.overfitting.hard_to_do.pdf).
- [11] Werbos P. *The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting*. Wiley-Interscience, Adaptive and Learning Systems for Signal Processing, Communications and Control Series. ISBN: 0-471-59897-6, 1994.
- [12] Lingireddy S. y Ormsbee L.E. "Neural Networks in Optimal Calibration of Water Distribution Systems". En *Artificial Neural Networks for Civil Engineers: Advanced Features and Applications*, Ed: I. Flood y N. Kartam, American Society of Civil Engineers, 1998, pp. 53-76.
- [13] Shahsavand A. y Ahmadpour A. "Application of Optimal Rbf Neural Networks for Optimization and Characterization of Porous arterials". *Computers and Chemical Engineering*, 29, pp. 2134-2143, octubre 2005.
- [14] Jin R., Chen W. y Simpson T.W. "Comparative Studies of Metamodeling Techniques under Multiple Modeling Criteria". *Journal of Structural Optimization*, 23(1), pp. 1-13, abril 2001
- [15] Serrano C. y Martín B. "Predicción de la Quiebra Bancaria mediante el empleo de Redes Neuronales Artificiales". *Revista Española de Financiación y Contabilidad*, XXIII (74), pp. 153-176, marzo 1993.
- [16] Santana J.C. "Predicción de series temporales con redes neuronales: una aplicación a la inflación colombiana". *Revista Colombiana de Estadística*, 29 (1), pp. 77-92, enero-junio 2006.
- [17] Nojek S., Britos P., Rossi B. y García R. "Pronóstico de Ventas: Comparación de Predicción basada en Redes Neuronales versus Método Estadístico". *Reportes Técnicos en Ingeniería del Software*, 5 (1), CAPIS-EPG-ITBA, ISSN: 1667-5002, pp. 1-12, 2003.
- [18] Ovando G., Bocco M. y Sayago S. "Redes Neuronales para Modelar Predicción de Heladas". *Agricultura Técnica (Chile)*, 65 (1), pp. 65-73, enero-marzo 2005.
- [19] Molera L. y Caballero M. (2001, septiembre). *Predicción del éxito en estudios universitarios mediante Redes Neuronales*. Presentado en X Jornadas de la Asociación de Economía de la Educación, pp. 505-513. Disponible: <http://www.pagina-aede.org/Murcia/E07.pdf>.
- [20] Borracci R.A. y Rubio M. "Aplicabilidad de redes neuronales artificiales para la predicción de los resultados individuales de la cirugía cardíaca". Estudio preliminar. *Revista Argentina*

- de Cardiología, 71 (5), pp. 351-358, sept-oct 2003.
- [21] Rodríguez A., Pértega S., González A., Fernández S., Suárez G. y Cuerpo M.A. "Utilidad de las redes neuronales artificiales en la predicción de cáncer de próstata en la biopsia transrectal". Actas urológicas españolas: Órgano oficial de difusión de la Asociación Española de Urología, 30 (1), ISSN: 0210-4806, pp. 18-24, 2006.
- [22] Wang D., Zhou Y. y He X. "Radial basis function neural network-based model predictive control for freeway traffic systems". International Journal of Intelligent Systems Technologies and Applications, 2 (4), pp. 370-388, 2007.
- [23] Rivas V.M., Arenas M.G., Merelo J.J. y Prieto A. (2007, septiembre). *Predicción de consumo de agua mediante redes neuronales de funciones base radiales evolutivas*. Presentado en II Congreso Español de Informática (CEDI 2007), Zaragoza, España, pp. 403-413. Disponible: <http://sci2s.ugr.es/keel/pdf/keel/congreso/SICO07-cedi2005.pdf>.
- [24] Panda S.S., Chakraborty D. y Pal S.K. (2006, marzo). *Prediction of Drill Flank Wear Using Radial Basis Function Neural Network*. Presentado en National Conference on Soft Computing Techniques for Engineering Applications, pp. 94-102.
- [25] May R.M. "Simple mathematical models with very complicated dynamics". Nature, 261 (5560), pp. 459-467, junio 1976.
- [26] Swanston D.J., Bishop J.M. y Mitchell R.J. "Simple adaptive momentum: new algorithm for training multilayer perceptrons". Electronics Letters, 30 (18), pp. 1498-1500, septiembre 1994.
- [27] Chande P.K. y Shrivastava, M. (1993, octubre). *Rule based design of a multilayer perceptron Neural Networks*. Presentado en International Joint Conference on Neural Networks, 3 (25-29), pp. 2865-2868. Disponible: <http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel4/5797/15470/00714320.pdf?temp=x>.
- [28] Wolfram, S. *A New Kind of Science*. Bulletin (New Series) of The American Mathematical Society, 40(1), pp. 143-150, octubre 2002.
- [29] Buyya R., Abramson D., Giddy J. y Stockinger H. "Economic Models for Resource Management and Scheduling in Grid Computing". Journal of Concurrency and Computation: Practice and Experience, edición especial 14/13-15, Wiley Press, pp. 1507-1542, enero 2002.
- [30] Foster I., Kesselman C. y Tuecke S. "The Anatomy of the Grid: Enabling Scalable, Virtual Organizations". International Journal of Supercomputer applications and High Performance Computing, 15 (3), ISSN:1094-3420, pp. 200-222, marzo 2001.
- [31] Herrero P., Bosque J. L. y Pérez M- S. (2007, noviembre). *An Agents-Based Cooperative Awareness Model to Cover Load Balancing Delivery in Grid Environments*. Presentado en OTM 2007 Workshops, Lecture notes in computer science 2536, On the Move to Meaningful Internet Systems 2007: ISSN: 0302-9743, ISBN: 978-3-540-76887-6, Springer Verlag, 4805, pp. 64-74. Disponible: <http://www.springerlink.com/content/q2472211744p6t2/>.
- [32] Paletta M. y Herrero P. (2008, diciembre). *Learning Cooperation in Collaborative Grid Environments to Improve Cover Load Balancing Delivery*. Presentado en IEEE/WIC/ACM Joint Conferences on Web Intelligence and Intelligent Agent Technology. IEEE Computer Society E3496, ISBN: 978-0-7695-3496-1, pp. 399-402. Disponible: [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=4740656](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4740656).