

SERIES TEMPORALES APLICADAS AL CONTROL. INFLUENCIA DE UN MÉTODO DE INICIALIZACIÓN PARA UNA RED NEURONAL DIFUSA

**Lic. Tania Hernández Fernández, Lic. Humberto Brito Santana
Lic. Luis Monzón Benítez**

*Universidad de Camagüey, Facultad de Informática
Circunvalación Norte Km. 5.5, Camagüey, Cuba
e-mail: tania@inf.reduc.edu.cu*

Abstract: En los últimos años la comunidad científica ha utilizado las Series Temporales para la modelación y control de diferentes procesos, abarcando desde problemas relativamente simples hasta los de elevada complejidad, uno de los focos de principal atención ha sido la Predicción de Series Temporales. Por otra parte, los Sistemas Difusos utilizando las redes neuronales, han incorporado métodos de entrenamiento que les permiten aprender a partir de ejemplos, dando lugar a las Redes Neuronales Difusas (RND). Las RND incorporan las ventajas en la implementación y la capacidad de aprendizaje de las Redes Neuronales Artificiales y el alto nivel de razonamiento y la toma de decisiones de los Sistemas Difusos. La inicialización de los parámetros es importante en una RND debido a que influye en garantizar una convergencia correcta del algoritmo al punto de mínimo error buscado. En este trabajo se propone un método de inicialización para los parámetros de una RND aplicada a la Predicción de Series Temporales, basado en técnicas de formación de clusters y se realiza una comparación con uno de los modelos mas populares propuesto por Teocharis.

Keywords: Redes Neuronales, Redes Neuronales Difusas, Series Temporales.

1. INTRODUCCIÓN

Los sistemas difusos han incorporado métodos de entrenamiento que les permiten aprender a partir de ejemplos y han dado lugar a lo que se conoce como redes neuronales difusas (RND), debido a su similitud con la estructura y funcionamiento de las redes neuronales artificiales (RNA) [1]. Una de las principales virtudes que se le atribuyen RND es la posibilidad de inicializar sus parámetros con valores que tengan un significado físico, lo que contribuye a

mejorar la velocidad de aprendizaje, en comparación con las RNA, cuya inicialización se realiza de forma aleatoria en la mayoría de los casos [2].

En este trabajo se propone un método de inicialización para los parámetros de la RND desarrollada por Teocharis [3], a partir de un análisis de clusterización que se realiza a los datos disponibles para el entrenamiento, de forma tal que se puedan asignar conjuntos difusos que cubran adecuadamente los diferentes clusters encontrados. El trabajo está organizado de la siguiente forma: En la

sección dos se explica brevemente la RND, su topología y su algoritmo de entrenamiento. En la siguiente sección se introduce el método de inicialización propuesto con diferentes variantes y se discuten sus características en relación con el método tradicional.

Finalmente se señalan las principales conclusiones y las direcciones fundamentales en las que se investiga en estos momentos sobre el referido método.

2. BREVE DESCRIPCIÓN DE LA RED NEURONAL DIFUSA

2.1. Configuración de un Sistema Difuso.

La configuración general de un sistema difuso está formado por cuatro elementos básicos: interfase difusa, base de reglas difusas, máquina de inferencia difusa e interfase dura [3,4]. Cuando el sistema difuso está acoplado a varias entradas y una salida, la regla de control j-ésima del sistema queda descrita como:

$$w^j(x) = \sum_{s=0}^n \mathbf{I}_s^j x_s \quad [x_0 = 1] \quad (1)$$

Donde \mathbf{I}_i^j son parámetros de valores reales, w^j es la salida del sistema desde la regla R^j , x_i ($i=1,2,\dots,n$) son las variables de entrada al sistema difuso y D_i^j son conjuntos difusos que están caracterizados por las funciones de pertenencia difusas $m_{D_i^j}(\cdot)$, de la forma:

$$m_{D_i^j}(x_i) = \exp \left\{ -\frac{1}{2} \left(\frac{x_i - \bar{x}_i^j}{g_i^j} \right)^2 \right\} \quad (2)$$

La interfase dura, toma la base de reglas y las salidas difusas de la máquina de inferencia difusa para generar una salida dura la cual es la salida general del sistema difuso y está determinada por:

$$y(x) = \frac{\sum_{j=1}^m w^j(x) m^j(x)}{\sum_{j=1}^m m^j(x)} \quad (3)$$

Donde $m^j(x)$ proporciona el grado de activación de la regla j.

2.2 Red Neuronal Difusa. Algoritmo de entrenamiento.

El sistema difuso descrito anteriormente tiene una estructura de una red de tres capas. En la capa A están los nodos de entrada cuya función de transferencia es la identidad. Cada uno de estos nodos, está conectado con m nodos S los cuales relacionan la entrada x_i con m conjuntos difusos. Los nodos S se conectan de forma particular con los nodos Π , cada uno de estos nodos corresponde a una regla difusa. Cada nodo de entrada, se conecta además con cada uno de los nodos Σ de la capa B, donde se ejecuta una suma pesada de las entradas y constituyen la parte consecuencia de las reglas difusas. En la capa C se ejecuta la transformación de los valores difusos a valores duros.

La Red Neuronal Difusa descrita anteriormente posee como parámetros desconocidos g_i^j , \bar{x}_i^j y \mathbf{I}_s^j ($j=1,2,\dots,m$), ($i=1,2,\dots,n$), ($s=0,1,2,\dots,n$). Se toman Q pares de entrada-salida deseada (x^p, y_d^p) ($p=1,2,\dots,Q$) donde

$x^p \in U \subset \mathfrak{R}^n$ y $y_d^p \in \mathfrak{R}$. Se emplea el algoritmo de retropropagación de los errores hacia atrás (backpropagation) basado en el método del gradiente (o del descenso más rápido) para la optimización de una función objetivo $J(z)$ que, en este caso, establece un promedio del error cuadrático medio por entrada.

2.3 Inicialización de los parámetros de la RND

La inicialización de los parámetros es fundamental en el proceso iterativo ya que ofrece la proximidad inicial al punto de mínimo error buscado. Si los valores iniciales están muy alejados, la convergencia del algoritmo es lenta y hay mayor posibilidad de que el algoritmo converja a un mínimo local en el espacio de los parámetros [6,7].

Inicialización según Teocharis

Se considera que inicialmente la RND contiene m reglas difusas y se utilizan los m primeros vectores de entrenamiento (x^j, y_d^j) $j=1,\dots,m$ para asignar los valores iniciales a los parámetros [8], por esto, se exige $Q \geq m$ y se obtiene:

$$\bar{x}_i^j(0) = x_i^j$$

$$g_i^j(0) = \frac{1}{m} \left[\max_{1 \leq p \leq m} (x_i^p) - \min_{1 \leq p \leq m} (x_i^p) \right]$$

para $i=1,\dots,n; j=1,\dots,m$ y $p=1,\dots,m$.

De esta forma se obtiene que los valores medios \bar{x}_i^j de las funciones de pertenencia van a quedar centrados directamente en los correspondientes

puntos x_i^j de los m primeros vectores de entrada. Esta inicialización no utiliza toda la información disponible de los vectores de entrenamientos al solo utilizar m de los Q existentes, además se comienza con un número inicial de reglas prefijado.

Propuesta de inicialización de parámetros \bar{x} y s

Nos proponemos crear inicialmente conjuntos difusos a partir de los Q vectores disponibles para el entrenamiento tales que cubran los valores de estos vectores de la mejor forma posible. Según nuestro punto de vista, una buena aproximación consistiría en analizar todos los datos disponibles para el entrenamiento, o sea, los Q vectores x^p , donde $x^p = (x_1^p, \dots, x_n^p)$, $x^p \in \mathfrak{R}^n$ y agruparlos con un método que logre extraer características comunes haciendo posible la formación de clusters en ellos. Sería natural entonces, hallar la media y la varianza de cada uno de los clusters obtenidos y construir con estos parámetros las funciones de pertenencia correspondientes, asociando cada cluster a un conjunto difuso de la red.

Las técnicas de clusterización constituyen un tema vigente en las investigaciones de nuestros días, es por esto que varias de estas técnicas se pueden encontrar en la literatura científica actual. Los modelos neuronales se han unido a ese campo basándose en la existencia de modelos que son capaces de autoorganizarse en el espacio, encontrando en ellos una de las formas de realizar una clusterización. Entre estos modelos se encuentran las redes de Kohonen como un modelo de la auto-organización de las conexiones neuronales. Una variante de las más prometedoras es adoptar una de las técnicas de clusterización o de formación de clusters en un espacio. Uno de estos algoritmos fue propuesto por Lazo en [11] en el cual se utiliza para realizar la clusterización una técnica que parte de tener un determinado conjunto de grupos que en un inicio tiene un único elemento, y escoger cual de los elementos del conjunto tiene menor similitud intragrupal para ser sometido a una técnica de división en otros dos grupos. Obteniéndose así los grupos o clusters que existen en los valores dados. En este algoritmo no hay necesidad de fijar a priori el número de reglas difusas que tendrá inicialmente la

RND, pues este número se obtiene como resultado del proceso de clusterización, o sea, es la cantidad de clusters obtenidos. Con este proceso se asegura que todos los x_i pertenezcan a algún conjunto difuso.

Intervención de los parámetros \mathbf{I} .

La salida de la regla R_j cuando se presenta un vector x a la red, está dada por la expresión:

$$w^j(x) = \mathbf{I}_0^j + \mathbf{I}_1^j x_1 + \dots + \mathbf{I}_n^j x_n$$

Propuesta de Teocharis.

La variante de teocharis se basa en la asignación

$$w^j(x) = y_d^j, \text{ o sea, } \mathbf{I}_0^j + \mathbf{I}_1^j x_1 + \dots + \mathbf{I}_n^j x_n = y_d^j$$

y tomar $\mathbf{I}_0^j = y_d^j$ y $\mathbf{I}_i^j = 0$ para $1 \leq i \leq n$ y $1 \leq j \leq m$. Este método provoca que inicialmente la regla R_j participe completamente en la salida de la red, para la entrada x_j .

Propuesta de inicialización para los parámetros \mathbf{I} .

Teniendo en cuenta la propuesta de inicialización para los parámetros \bar{x} y σ realizada anteriormente, no podremos hacer uso de la forma recién expuesta de asignar los valores iniciales a los parámetros λ , pues los supuestos son completamente diferentes.

En este caso partiremos de la salida de a red para un vector de entrenamiento.. Sustituyendo

$$y_d^p \sum_{j=1}^m \mathbf{m}^j(x^p) = b^p \text{ y } \mathbf{m}^j(x^p) = a_p^j \text{ obtenemos:}$$

$$b^p = \sum_{j=1}^m w^j(x^p) a_p^j(x^p) \text{ sustituyendo } w^j(x^p) \text{ en}$$

$$(1) \text{ queda: } b^p = \sum_{j=1}^m \sum_{i=0}^n \mathbf{I}_i^j x_i^p a_p^j$$

Sustituyendo $x_i^p a_p^j = d_i^{jp}$ el sistema queda de la siguiente forma:

$$b^p = \sum_{j=1}^m \sum_{i=0}^n \mathbf{I}_i^j d_i^{jp}$$

Hemos obtenido un sistema de ecuaciones lineales de Q ecuaciones por $m(n+1)$ incógnitas. Una solución del mismo proporcionaría los valores de λ .

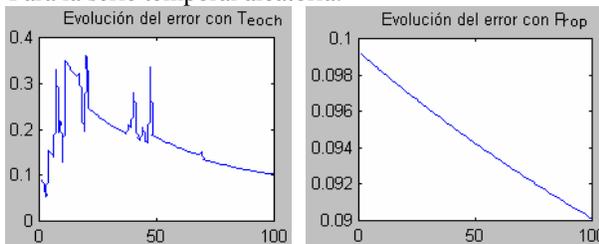
Experimentos Numéricos

Se realizaron dos experimentos, primeramente se generó una serie temporal de 1000 puntos con un

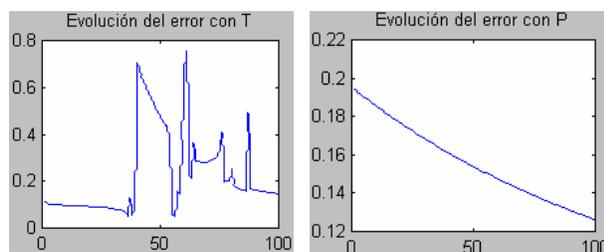
generador de números aleatorios de Matlab, produciéndose una serie temporal aleatoria de distribución normal; y para el segundo experimento una serie temporal, de igual tamaño, generada por el mapa logístico: $X_{t+1}=aX_t(1-X_t)$. La cual, para estos valores presenta un comportamiento caótico.

A cada una de estas series le fue aplicado el algoritmo de clusterización, anteriormente descrito, obteniéndose los grupos que existían en ellas, los cuales, como anteriormente explicamos, fueron utilizados para extraer los valores característicos de los conjuntos difusos de la red. La estructura de la red fue de 5 neuronas de entrada. El conjunto de vectores de entrenamiento estuvo formado en cada paso por 25 pares de entrada-salida deseada y se entrenó para cada experimento la Red Neuronal Difusa con 1000 iteraciones. Posteriormente para cada una de estas series, la red fue entrenada con 1000 iteraciones utilizando el método de inicialización de Teocharis y con el método de inicialización propuesto. Las redes obtenidas fueron utilizadas para la predicción de la serie temporal original y se obtuvieron resultados, de los cuales se muestra la evolución del error durante en el entrenamiento para los diferentes casos.

Para la serie temporal aleatoria:



Para la serie generada por el mapa logístico



Como puede apreciarse, los resultados obtenidos demuestran que el método de inicialización es efectivo y que aproxima la red al punto de error mínimo global, ya que es apreciable la diferencia existente en el momento de comenzar el algoritmo entre los dos métodos así como su evolución en el transcurso de las iteraciones.

3. CONCLUSIONES

En este trabajo se ha propuesto un método de inicialización de los parámetros de una RND específica, basada en la aplicación de técnicas de formación de clusters. Estas técnicas permiten una proximidad inicial al punto mínimo global buscado para la función error, permitiendo una convergencia en menos pasos iterativos y disminuyendo la posibilidad de caer en un mínimo local, obteniéndose resultados satisfactorios que responden al modelo teórico. En estos momentos se trabaja teórica y experimentalmente en la eficiencia y velocidad de los algoritmos utilizados.

REFERENCIAS

- [1]. THEOCHARIS, J. "Adaptative Fuzzy Neural Networks as Identifiers of Discrete-time Nonlinear Dynamic Systems"/ Theocharis, J. and Vachtsevanos, G., Journal of Intelligent and Robotic Systems, Vol. 4, No.6: 21-73, 1996.
- [2]. BUCKLEY, J. "Fuzzy Neural Networks"/ Buckley, J. and Hayashi, Y., Fuzzy Sets and Systems, Vol. 66: 13-54, 1994.
- [3]. MONZÓN, L. "Predicción de series temporales de EEG mediante Redes Neuronales Artificiales"/ Monzón, L. y Carbó, J. R. Memorias del II Taller Iberoamericano de Reconocimiento de Patrones, pp. 339-344, La Habana, Marzo 1997.
- [4]. HERNÁNDEZ, T., "Predicción de series temporales con un modelo de red neuronal difusa". Trabajo de Diploma. Universidad de Oriente. 2001.
- [5]. BRITO, H. "Métodos para la predicción de series temporales". Trabajo de Diploma. Universidad de Oriente, 2000