

Predicción de una serie de tiempo caótica con técnicas de inteligencia artificial

Prediction of a chaotic time series with artificial intelligence techniques

LUIS FERNANDO PEDRAZA MARTÍNEZ

Ingeniero Electrónico de la Universidad Distrital Francisco José de Caldas, magister en Ciencias de la Información y las Comunicaciones de la Universidad Distrital Francisco José de Caldas, docente tiempo completo de la Universidad Distrital Francisco José de Caldas e integrante de los grupos de investigación Semaforización Inteligente e Internet Inteligente. Correo electrónico: lfpedrazam@udistrital.edu.co

CÉSAR AUGUSTO HERNÁNDEZ SUÁREZ

Ingeniero Electrónico de la Universidad Distrital Francisco José de Caldas, especialista Servicios Telemáticos e Interconexión de Redes, magíster en Ciencias de la Información y las Comunicaciones de la Universidad Distrital Francisco José de Caldas, y aspirante a Magister en Economía de la Universidad de los Andes. Docente e investigador de la Universidad Distrital Francisco José de Caldas, e integrante del grupo de investigación ARMOS e Internet Inteligente. Correo electrónico: cahernandezs@udistrital.edu.co

ÓSCAR FABIÁN CORREDOR CAMARGO

Ingeniero Electrónico. Profesor Catedrático Universidad Distrital Francisco José de Caldas. Correo electrónico: ofccaing@gmail.com

Clasificación del artículo: investigación (Conciencias)

Fecha de recepción: 26 de abril de 2008

Fecha de aceptación: 29 de julio de 2008

Palabras clave: algoritmos genéticos, defusificación, error al cuadrado, fusificación, peso de cumplimiento, red neuronal.

Key words: genetics algorithms, defuzzification, squared error, fuzzification, firing strength, neural network.

RESUMEN

En este artículo se presenta el procedimiento y el resultado principal de un estudio comparativo, el cual está basado en el uso de dos herramientas de inteligencia computacional, aplicadas en una tarea de predicción de una serie de tiempo caótica. Los métodos de predicción de series de tiempo compa-

rados son: el algoritmo ANFIS (Sistema de Inferencia Difuso Basado en Redes Adaptativas) y un algoritmo genético evolutivo. Luego, se presentan y se analizan los resultados de este estudio, bajo el criterio de la suma del error al cuadrado y el tiempo de procesamiento requerido.

ABSTRACT

In this paper is showed the procedure and the main result of a comparative study based on the use of two tools of computational intelligence applied in a task of prediction of a chaotic series of time. The methods of prediction of time series compared are: the ANFIS

algorithm (Adaptive Network Based Fuzzy Inference System) and an evolutionary genetic algorithm. Then the results of this study are presented and analyzed, under the criterion of the sum of squared error and required processing time.

* * *

1. Introducción

En la planeación, para futuras necesidades de cualquier sistema, la exactitud en el pronóstico de tráfico es realmente importante para definir la capacidad futura requerida y planear los cambios. Un modelo de series de tiempo bastante exacto podría predecir varios años hacia el futuro, cuya habilidad es una ventaja para la planeación de futuros requerimientos [1].

Según la teoría matemática actual, un sistema caótico se define por mostrar “sensibilidad a las condiciones iniciales”; es decir, para predecir el estado futuro de un sistema con certeza, se necesita saber las condiciones iniciales con una exactitud infinita, puesto que los errores aumentan rápidamente con la más ligera inexactitud [2]. Es por esto que sistemas como: financieros, climatológicos, de producción, de minería de datos, entre otros, son tan difíciles de pronosticar [3]. Por tanto, este artículo pretende mostrar la conveniencia de algunas técnicas de inteligencia artificial en esta tarea.

Predecir una serie de tiempo caótica con sistemas difusos es algo difícil, ya que se busca la identificación de un modelo difuso óptimo [4]. Para resolver este problema, este estudio presenta un acercamiento para la construcción de un modelo difuso a partir de los datos caóticos de una señal, utilizando ANFIS y Algoritmos Genéticos. El conjunto de datos de la serie caótica que se va a utilizar viene del modelado del tráfico microscópico de un automóvil a través de una sucesión de semáforos

(figura 1), presentado en [5]. Por tanto, el uso de la computación evolutiva para la predicción de series de tiempo caóticas es un instrumento de aplicación fundamental para sistemas que no se pueden modelar matemáticamente de manera sencilla, como lo son los sistemas de tráfico vehicular.

Uno de los métodos utilizados en este trabajo para la predicción de la serie de tiempo es el ANFIS, el cual consiste en un modelo para el aprendizaje neuro-difuso adaptativo. Esta técnica mantiene el método del modelado difuso y lo usa para aprender a partir de la información disponible sobre un conjunto de datos; estos últimos se usan para inicializar los parámetros de las funciones de pertenencia que permiten mejorar la inferencia del sistema difuso asociado, para enlazar los datos de entrada-salida. Este método de aprendizaje trabaja de forma similar a aquéllos basados en redes neuronales, bajo el modelo Takagi-Sugeno [6]. Éste pone a punto los parámetros de las funciones de pertenencia usando el algoritmo de entrenamiento propagación hacia atrás [7]. Este modelo es implementado para predecir la serie de tiempo caótica, la cual es mostrada en la figura 1, la cual se logra sincronizando 209 semáforos a una frecuencia específica. Este estudio podría ayudar en el futuro a diseñar sistemas de predicción de tráfico macroscópico para controlar los periodos de congestamiento vehicular.

Los algoritmos genéticos aplicados al pronóstico de series de tiempo son algo que sólo se han implementado en los últimos años y es el segundo método utilizado en este trabajo para predecir la

serie de tiempo caótica. El primer acercamiento al problema de la serie de tiempo que usa algoritmos genéticos es hecho en [8]; en éste la idea principal es encontrar las reglas de predicción de los datos de la serie de tiempo.

Un algoritmo genético es un método de búsqueda dirigida, la cual está basada en probabilidad bajo una condición muy débil –que el algoritmo mantenga elitismo, es decir, guarde siempre al mejor elemento de la población sin hacerle ningún cambio–, se puede demostrar que el algoritmo converge en probabilidad al óptimo [9], siendo este criterio utilizado en el presente trabajo para modificar los parámetros de un modelo difuso.

El software para la implementación de estas dos técnicas de predicción se desarrollo en Matlab.

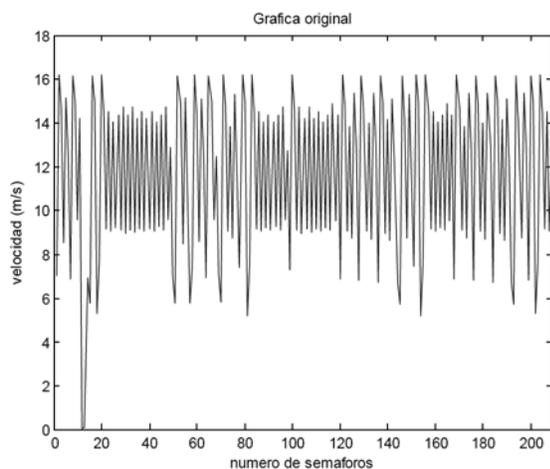


Figura 1. Velocidad de un vehículo a través de 209 semáforos

2. Estructura del anfis

ANFIS tiene una estructura tipo red, similar al de una red neuronal, la cual mapea las entradas a través de funciones de pertenencia y sus parámetros asociados y, luego, a través de las funciones de pertenencia de la salida y los parámetros asociados a la salida, este sistema se puede interpretar como el mapeo entrada-salida. Los parámetros asociados

a las diferentes funciones de pertenencia cambian a través del proceso de aprendizaje. La combinación de parámetros (o su ajuste) es realizada por un vector gradiente, el cual provee una medida, que también ajusta el sistema de inferencia y que obtiene para él, modelando el conjunto de datos entrada-salida para un conjunto de parámetros dado. Una vez el vector gradiente es obtenido, cualquiera de las muchas rutinas de optimización podría ser aplicada con el fin de ajustar los parámetros, así como para reducir la medida del error –definida en este caso por la suma del cuadrado de la diferencia entre el valor de entrada y el de la salida obtenida–. Como se menciono anteriormente, ANFIS usa el algoritmo llamado propagación hacia atrás para estimar los parámetros de la función de pertenencia [7].

La estructura ANFIS aquí descrita está basada en el modelo Takagi-Sugeno, el cual, según lo demostrado en [10], se puede representar como redes neuronales difusas de 5 capas. Este ejemplo de una red neuro-difusa de 5 capas se muestra en la figura 2. La primera capa se utiliza para la fusificación de las entradas. En la segunda capa, se calcula el peso de cumplimiento de las reglas difusas. La tercera capa es la capa de normalización. En la cuarta capa, los valores de las reglas de los consecuentes son calculados y multiplicados por el peso de cumplimiento de las respectivas reglas y la quinta capa realiza la desfusicación [11].

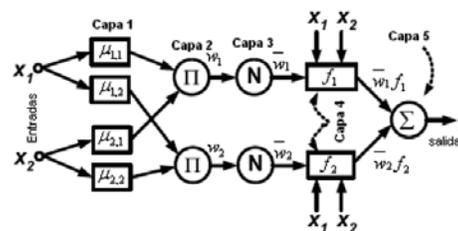


Figura 2. Arquitectura de la red neuro-difusa con dos reglas

Para este caso de estudio, inicialmente se utiliza en el entrenamiento del modelo neuro-difuso dos muestras anteriores ($y(k-1)$ y $y(k-2)$) de la señal por

predecir ($y(k)$), por tanto, se tiene dos universos de entrada ($y(k-1)$ y $y(k-2)$), en los cuales cada uno posee dos conjuntos sigmoidales (mf1 y mf2) y un universo de salida ($u(k)$) con 4 conjuntos lineales de salida (mf1, mf2, mf3 y mf4), (Figura 3). A su vez, esto permite ajustar las siguientes reglas:

- SI $y(k-1)$ es mf1 y $y(k-2)$ es mf1 ENTONCES $u(k)$ es mf1 (1)
- SI $y(k-1)$ es mf1 y $y(k-2)$ es mf2 ENTONCES $u(k)$ es mf2 (2)
- SI $y(k-1)$ es mf2 y $y(k-2)$ es mf1 ENTONCES $u(k)$ es mf3 (3)
- SI $y(k-1)$ es mf2 y $y(k-2)$ es mf2 ENTONCES $u(k)$ es mf4 (4)

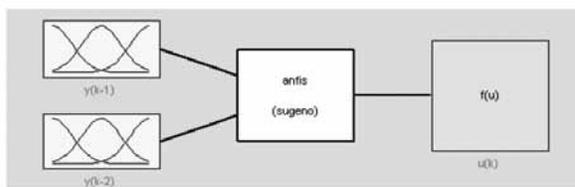


Figura 3. Sistema ANFIS basado en el modelo Takagi-Sugeno

2.1 Algoritmo de entrenamiento “propagación hacia atrás”

A continuación se realiza una breve descripción del algoritmo utilizado para el entrenamiento de los parámetros de las funciones de pertenencia:

Dado un par de parejas de entrenamiento:

$$\{(X^{(1)}, d^{(1)}), \dots, (X^{(i)}, d^{(i)}), \dots, (X^{(p)}, d^{(p)})\} \quad (5)$$

Paso No.1 Inicialice pesos y puntos iniciales con pequeños valores aleatorios.

Paso No.2 Presente un nuevo vector de entrada y de salida deseada. Presente el nuevo vector de entrada y la correspondiente salida deseada $(X^{(i)}, d^{(i)})$. Calcule la salida real.

Paso No.3 Calcule el error del gradiente.

Paso No.4 Adapte los pesos de los vectores y las condiciones iniciales:

$$w(k+1) = w(k) - \alpha \nabla_w E(w(k), t(k)), \quad (6)$$

$$t(k+1) = t(k) - \beta \nabla_t E(w(k), t(k)) \quad (7)$$

Donde

$w(k+1)$: vector de peso ajustado.

$w(k)$: vector de pesos inicial.

$\nabla_w E(w(k), t(k))$: Gradiente del error en función de los pesos y el tiempo

α, β : valor del peso fijado

Paso No.5 Repetir, regresando al paso No.2 [12].

3. Algoritmo genético evolutivo

Los algoritmos genéticos son el resultado de un esfuerzo para modelar los fenómenos de adaptación que se presentan tanto en los sistemas naturales como artificiales. Generalmente, la observación cuidadosa y metódica de las modificaciones estructurales sucesivas revelan un conjunto básico de modificadores estructurales u operadores. La acción repetida de estos operadores explica la secuencia de las modificaciones observadas [12].

El diagrama de flujo del algoritmo genético utilizado en este caso de estudio se muestra en la figura 4, en el cual se genera una población inicial de individuos aleatoria, los cuales se evalúan, se seleccionan y se reproducen, a través de 200 generaciones, para establecer una nueva población con mejores características. Para esta técnica se utiliza el alfabeto binario y la población para dos muestras anteriores corresponde al conjunto de parámetros de las funciones de pertenencia sigmoidales de los universos de entrada $y(k-1)$ y $y(k-2)$ y de las 4 funciones de pertenencia del universo de salida $u(k)$. Las reglas utilizadas son las mismas de ANFIS (1), (2), (3) y (4).

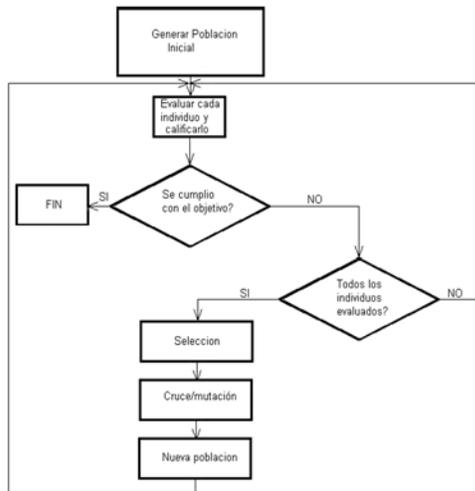


Figura 4. Diagrama de bloques del algoritmo genético

En este proceso se escogió una población (N) de 20 individuos, ya que según [13], para un alfabeto binario se debe tener una población mínima de:

$$N \approx [1 + \log(-\ell / \ln P_2^*) / \log 2] \approx 19 \text{ individuos} \quad (8)$$

Donde $P_2^* = 99.9\%$ es la probabilidad de que por lo menos un alelo (formas variantes de un gen) esté presente en cada sitio, para así lograr ser encontrado y ℓ en la longitud del conjunto. Para dos muestras anteriores se necesita manejar 2 parámetros por cada conjunto sigmoidal de entrada (dos conjuntos por entrada) y son dos entradas; se tiene entonces 8 parámetros por manejar, más los 12 parámetros de los 4 conjuntos lineales del universo de salida, para un total de 20 parámetros. El tamaño de cada fenotipo se escogió de 8 bits, ya que es un alfabeto binario, por tanto, la longitud total del conjunto de parámetros es de $\ell = 160$ bits.

A continuación se presenta la comparación de los dos métodos de predicción.

4. Resultados experimentales

Para dar validez a los métodos, se realiza la implementación de éstos en el software Matlab, aquí se toma la serie caótica original $y(k)$ mostrada en la figura 1 y se calcula el error entre ésta y la serie pronosticada $\hat{y}(k)$ por cada método, para esto se utiliza el concepto de la suma del error al cuadrado:

$$\sum_0^k e^2(k) = \sum_0^k \left(\hat{y}(k) - y(k) \right)^2 \quad (9)$$

También se tiene en cuenta el tiempo de procesamiento utilizado por cada método para predecir la serie.

4.1 Prueba realizada con el método ANFIS para dos muestras anteriores

En este método se obtiene la $\sum_0^k e^2(k) = 780,9258$ y un tiempo de procesamiento de 1,281 segundos, en la figura 5 se puede ver la tendencia de la serie pronosticada con respecto a la serie original y en la figura 6 se observa la diferencia (error) entre éstas.

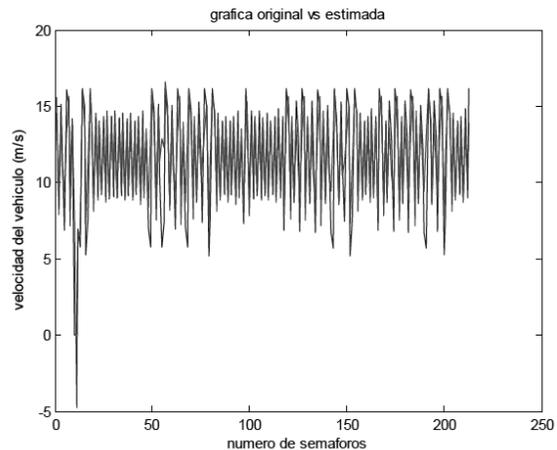


Figura 5. Serie original (azul) vs. Serie estimada (verde) por ANFIS para dos muestras anteriores

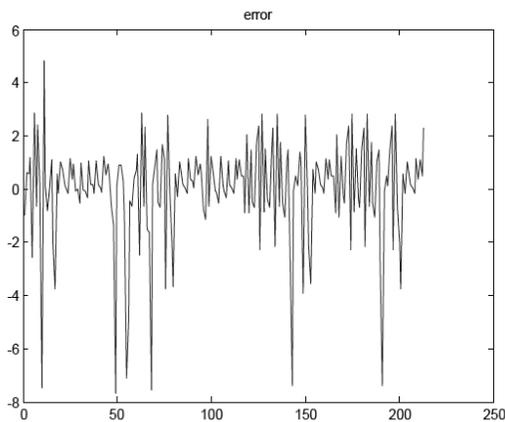


Figura 6. Diferencia entre la serie original y la pronosticada por ANFIS para dos muestras anteriores

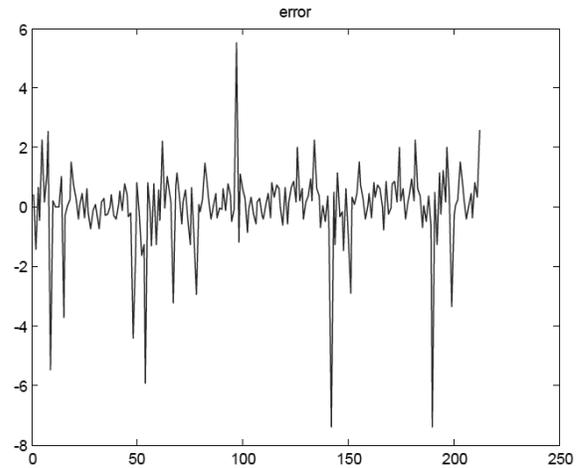


Figura 8. Diferencia entre la serie original y la pronosticada por ANFIS para tres muestras anteriores

4.2 Prueba realizada con el método anfis para tres muestras anteriores

Con la misma metodología usada para la implementación de dos muestras anteriores, se obtiene la figura 7 para tres muestras anteriores, donde $\sum_0^k e^2(k) = 401,04$ y el tiempo de procesamiento es de 1,53 segundos. En la figura 8 se observa el error producido en este caso.

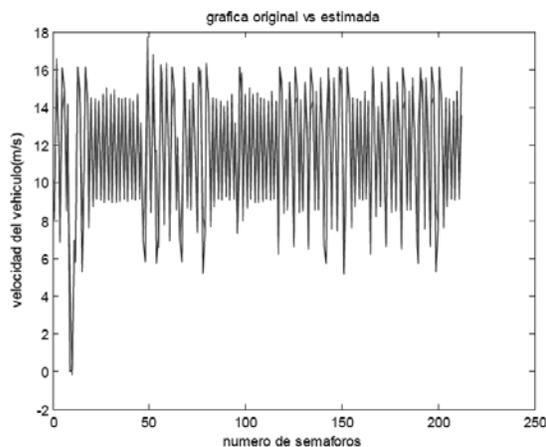


Figura 7. Serie original (azul) vs. Serie estimada (verde) por ANFIS para tres muestras anteriores

4.3 Prueba realizada con el método del algoritmo genético para dos muestras anteriores

Para el desarrollo de este algoritmo se utiliza:

- Número de generaciones: 200.
- Número de individuos: 20.
- Longitud del conjunto: 160.
- Valor de presión de selección: 2.
- Porcentaje de elitismo: 10%.
- Probabilidad de cruce: 0,8.
- Probabilidad de mutación: 0,04.

Lo cual arrojó una $\sum_0^k e^2(k) = 1681,3$ y un tiempo de procesamiento de 177,562 segundos. La gráfica de la serie original contra la predecida por este algoritmo se observa en la figura 9 y el error producido por la diferencia de éstas, se muestra en la figura 10.

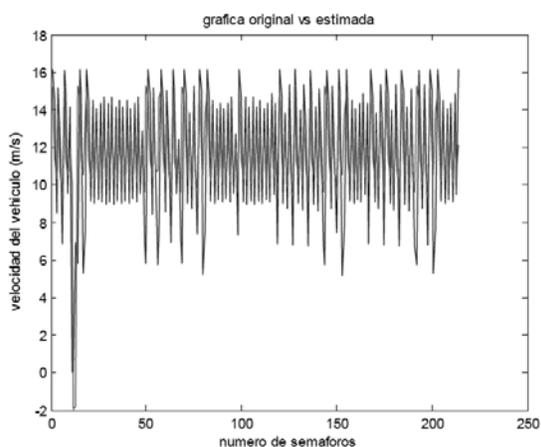


Figura 9. Serie original (rojo) vs. Serie estimada (azul) por algoritmos genéticos para dos muestras anteriores

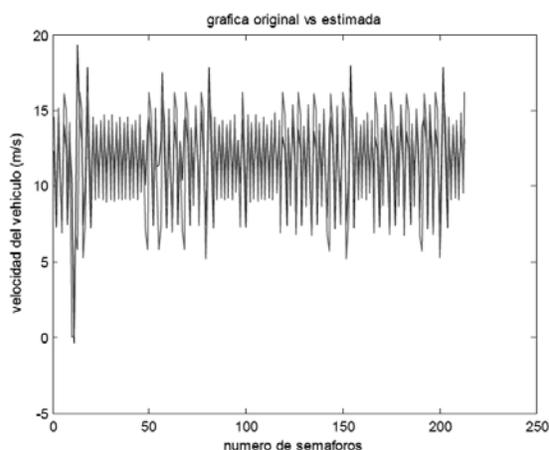


Figura 11. Serie original (rojo) vs. Serie estimada (azul) por algoritmos genéticos para tres muestras anteriores

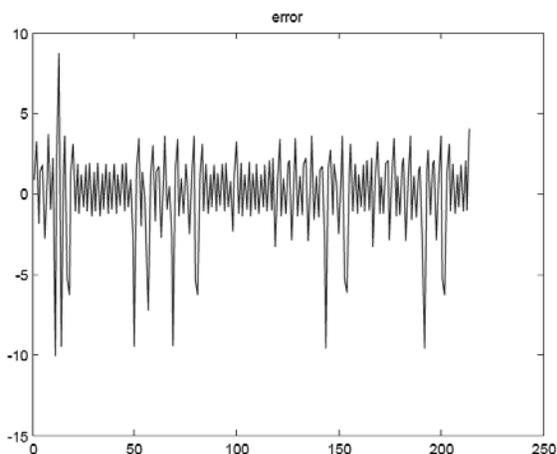


Figura 10. Diferencia entre la serie original y la pronosticada por algoritmos genéticos para dos muestras anteriores

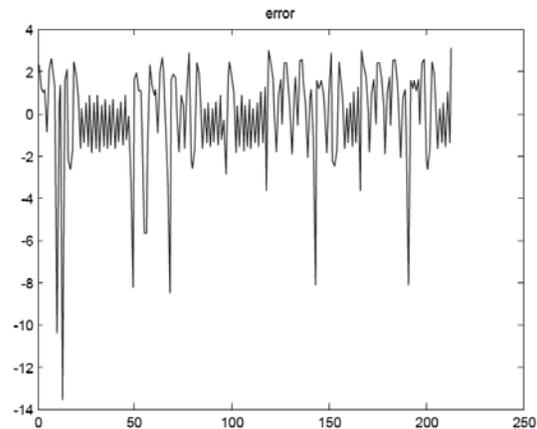


Figura 12. Diferencia entre la serie original y la pronosticada por algoritmos genéticos para tres muestras anteriores

4.4 Prueba realizada con el método del algoritmo genético para tres muestras anteriores

Usando los mismos valores de las variables del ítem anterior para tres muestras anteriores, se obtiene la figura 11 con $\sum_0^k e^2(k) = 1182,1$ y un tiempo de procesamiento de 864,3 segundos. En la figura 12 se grafica el error dado por este método.

5. Conclusiones

- Durante el desarrollo de las pruebas de los métodos, se determinó que la predicción basada en el método ANFIS ofrece una mejor representación de la señal original y esto se ve reflejado, a su vez, en la menor suma del error al cuadrado, sin contar que el tiempo de procesamiento es 138 veces menor que el utilizado por el algoritmo genético para el caso de dos muestras anteriores. Esto hace que ANFIS sea una mejor alternativa en la predicción de la velocidad de un flujo

vehicular, ya que se requiere una toma de decisión rápida y precisa, para, por ejemplo, cambiar los tiempos en los semáforos y evitar o reducir el tiempo de una congestión vehicular.

- Tomando tres muestras anteriores a la de la serie caótica original, se logra tener una mejor repre-

sentación de la serie de tiempo, aunque el costo computacional aumenta exponencialmente en el método de algoritmos genéticos.

- La sencilla manipulación de sus parámetros, convierte a los sistemas difusos en una excelente herramienta para predecir series de tiempo caóticas.

Referencias bibliográficas

- | | |
|--|--|
| <p>[1] L. Fillatre, D. Marakov y S. Vaton, <i>Forecasting Seasonal Traffic Flows</i>, Computer Science Department, ENST Bretagne, Brest, Paris, 2003.</p> <p>[2] S.H. Kellert, <i>In the Wake of Chaos: Unpredictable Order in Dynamical Systems</i>, The University of Chicago Press, 1993.</p> <p>[3] A. K. Palit and D. Popovic. <i>Computacional Intelligence in Time Series Forecasting Theory and Engineering Applications</i>, Springer, 2005.</p> <p>[4] J.S.R. Jang and C.T. Sun, "Predicting Chaotic Time Series with Fuzzy if-then Rules", <i>IEEE International Conference Fuzzy System</i>, 1993.</p> <p>[5] B.A Toledo, V. Muñoz, J. Rogan, and C. Tenreiro, "Modeling Traffic through a Sequence of Traffic Lights", <i>Physical Review</i>, 2004.</p> <p>[6] T. Takagi and M. Sugeno, "Fuzzy Identification of System and its Applications to Modeling and Control", <i>IEEE Transactions on Systems, Man, and Cybernetics</i>, Vol. 15, 1985.</p> <p>[7] J.S.R Jang, "ANFIS: Adaptive Network based Fuzzy Inference Systems", <i>IEEE Transactions on Systems, Man, and Cybernetics</i>, Vol. 23, 1993.</p> | <p>[8] T.P. Meyer and N.H. Packard, <i>Local Forecasting of High-Dimensional Chaotic Dynamics, Nonlinear modeling and forecasting</i>. Addison-Wesley, 1991.</p> <p>[9] M. Melanie, <i>An Introduction to Genetic Algorithms</i>, MIT Press, 1998.</p> <p>[10] J.S.R. Jang, C.T. Sun and E. Mizutani, <i>Neuro-Fuzzy and Soft Computing—A Computational Approach to Learning and Machine Intelligence</i>, Prentice Hall, 1997.</p> <p>[11] Z. Li, W. Halang and G. Chen, <i>Integration of Fuzzy Logic and Chaos Theory</i>, Springer, 2006.</p> <p>[12] S. Zak, <i>Systems and Control</i>, Oxford University Press, 2003.</p> <p>[13] C.R. Reeves and J.E. Rowe. <i>Genetic Algorithms: Principles and Perspectives Guide to GA Theory</i>, Kluwer Academic Publishers, 2002.</p> |
|--|--|