

PRONÓSTICO DE LA SERIE DE MACKEY–GLASS USANDO MODELOS DE REGRESIÓN NO LINEAL

JUAN DAVID VELÁSQUEZ HENAO

Escuela de Sistemas. Facultad de Minas. Universidad Nacional de Colombia

Recibido para revisar 11 de Marzo de 2003, aceptado 6 de Agosto de 2003, versión final 9 de Octubre de 2003

RESUMEN: Tres técnicas de regresión no lineal son estudiadas y se compara su desempeño en el pronóstico de la serie de tiempo de Mackey–Glass. La primera de ellas es el modelo MARS, que se plantea como una alternativa a los modelos conexionistas. El segundo es el Sistema de Inferencia Difusa Adaptativo (ANFIS), y el tercero es la red neuronal de propagación hacia delante, ampliamente utilizada en aplicaciones de ingeniería y economía. Los resultados muestran claramente como ANFIS tiene un poder de aproximación superior a las otras técnicas evaluadas. Se propone un algoritmo para obtener modelos ANFIS con complejidad creciente.

PALABRAS CLAVE: MARS, ANFIS, redes neuronales, perceptrones multicapa, regresión no lineal

ABSTRACT: Three nonlinear regression techniques are studied and its performance forecasting the Mackey–Glass time series is compared. The first is the MARS model, which is an alternative to connectionist models. The second is the Adaptive Neuro Fuzzy Inference System (ANFIS), and the third is a feedforward neural network, very used in many applications in engineering and economy. Results show clearly how MARS has a approximation power superior respect to another evaluated techniques.

KEYWORDS: MARS, ANFIS, neural networks, multilayer perceptrons, nonlinear regression

1 INTRODUCCIÓN

La adquisición del conocimiento implícito en la información histórica del comportamiento de un sistema es uno de los mayores problemas existentes en Inteligencia Artificial. Ella puede ser formulada como un problema de regresión, en el que se pretende aproximar una función desconocida G , que mapea los valores de los parámetros del problema, representados por el conjunto X , hacia los valores de los parámetros que definen una instancia Y del espacio de soluciones.

Existen diferentes metodologías para construir la aproximación de la función G , denotada como F , dentro de las cuales, los modelos convencionales de regresión múltiple hacen parte de las técnicas más tradicionales.

En el caso particular de las series de tiempo, se pretende encontrar un modelo que permita pronosticar el valor de la variable de interés en el instante $t + D$ usando sus valores en los instantes $t, t - 1, \dots$, esto es:

$$x_{t+D} = F(x_t, x_{t-1}, \dots)$$

de tal forma que ella capture las características más relevantes de la dinámica que gobierna la evolución de x . La construcción de F no es una tarea trivial,

y se basa en la hipótesis de que la relación funcional entre las variables x_t, x_{t-1}, \dots puede ser aproximada por la combinación de un grupo de familias de funciones simples. Así las actividades de identificación y selección del modelo se convierte en un proceso dinámico en donde se adicionan y se quitan términos funcionales hasta que se construye una función F que aproxime con la suficiente calidad a la dinámica del proceso generador. La principal crítica que se ha realizado a esta forma de modelaje, corresponde a la suposición del conocimiento de la forma funcional de F , como por ejemplo, una combinación lineal, la cual es usualmente desconocida.

De esta forma, resulta ventajoso el uso de modelos de regresión no paramétricos y no lineales, especialmente en el sentido de no tener que suponer algún tipo de relación funcional entre las variables ya que ellos pueden considerarse aproximadores universales de funciones.

Dentro de este tipo de modelos, los perceptrones multicapa (PMC) y los sistemas adaptativos neurodifusos (ANFIS) han sido dos de los modelos más tradicionales en Inteligencia Artificial, mientras que MARS lo es dentro de los modelos estadísticos. Cada uno se basa en hipótesis diferentes respecto a la forma de construir la función F y los algoritmos para determinar los valores de los parámetros. En el caso partic-

ular de las series de tiempo, ellos han sido utilizados en diferentes aplicaciones, pero no existen evidencias que permitan clasificar alguno de ellos como superior a los otros.

Se presenta entonces, un análisis a nivel conceptual de MARS, los PMC y ANFIS, así como un análisis empírico de los resultados obtenidos al aplicar dichos modelos a la serie de Mackey-Glass. En este último punto, se comparan los resultados obtenidos con los reportados por Jang para el modelo ANFIS, y se presentan resultados para MARS y los PMC, que no aparecen en la literatura.

2 DESCRIPCIÓN DE LOS MODELOS

2.1 MARS – Multivariate Adaptive Regression Splines

Este es un modelo de regresión no lineal (Friedman, 1991), en donde una función multivariable es aproximada a través de un particionamiento recursivo del espacio de entrada, en donde cada partición se realiza sobre alguna de las variables independientes. En cada una de las regiones en que se va dividiendo el espacio se ajusta una función base de una variable, la cual usualmente es lineal, obteniéndose el modelo final como una combinación lineal de dichas funciones base. Tanto los puntos de partición como los parámetros de los modelos se obtienen a partir de los datos.

El modelo final obtenido es equivalente a:

$$f(x) = a_0 + \sum_{K_m=1} f_i(x_i) + \sum_{K_m=2} f_{ij}(x_i, x_j) + \dots$$

donde la función de aproximación es construida como la suma de una constante, más la suma de las funciones base de una variable, más la suma de todas las funciones base de dos variables, que representan todas las interacciones entre dos variables, y así sucesivamente.

La complejidad del modelo obtenido está controlada por el número máximo de regiones, M_{max} , en que se divide el espacio de entrada.

La selección final del modelo es realizada explorando modelos con diferentes valores de M_{max} , y seleccionando aquel que minimice el criterio de validación cruzada generalizada (GCV), el cual es una medida de ajuste a los datos y penalización por la complejidad del modelo:

$$GCV(M) = \frac{1}{N} \sum_{i=1}^N [y_i - f_M(x_i)]^2 / \left[1 - \frac{C(M)}{N} \right]^2$$

donde M es el número final de funciones base (o regiones en que se han dividido los datos), N es la cantidad de datos, y_i es el valor de la función que se desea aproximar, f_M es el modelo con M funciones base, x_i

es el vector de valores de las variables independientes en el punto i , y $C(M)$ es un factor de penalización por el aumento de la varianza debido al aumento de la complejidad del modelo.

Algunos investigadores han planteado que este modelo puede reemplazar a los modelos de redes neuronales artificiales, ya que el costo computacional de su construcción es bastante inferior comparativamente con estas últimas; el proceso de especificación del modelo es automático, y no tiene la problemática propia de los modelos de redes neuronales; adicionalmente realiza una selección automática de las variables relevantes.

2.2 Perceptrones Multicapa

Los perceptrones multicapa (PMC) son modelos matemáticos que imitan las redes de neuronas biológicas que existen en el cerebro. Están conformadas por elementos individuales de procesamiento relativamente simples o neuronas artificiales, los cuales se encuentran agrupados en capas. Cada una de las capas que conforman un PMC puede recibir entradas (capa de entrada), entregar salidas (capas de salida), o ser inaccesibles a las entradas y las salidas, con conexiones únicamente a otras capas (capas internas). En este tipo de red neuronal artificial, la señal se propaga hacia delante, sin que exista ningún tipo de realimentación entre las neuronas. Un PMC se caracteriza por el tipo de función de transferencia de las neuronas, la cantidad de capas de neuronas que posee, y el algoritmo de entrenamiento usado para su aprendizaje (Eberhart y Dobbins, 1990). Las tareas realizadas por este tipo de red neuronal artificial pueden agruparse en tres categorías (Masters, 1993): clasificación, reducción del ruido y pronóstico. Un perceptron multicapa típico con una capa de neuronas ocultas, función de activación sigmoidea en la capa oculta y transferencia lineal en la capa de salida se presenta en la Figura 1.

En general, las redes neuronales artificiales (RNA) son consideradas como aproximadores universales de funciones. Desde un punto de vista matemático, son independientes del modelo (Kosko, 1992), en el sentido de que no se requiere conocer el tipo de función que va a ser aproximada; esto implica, que no se requiere suponer ningún tipo de relación funcional entre las variables de entrada y la variable de salida, la cual es una ventaja sobre los modelos econométricos, ya que no se hace necesario el proceso de identificación.

Los modelos de RNA ya han sido utilizados en el modelamiento de series de tiempo. Westheider (1997) usa un PMC para pronosticar los retornos mensuales de un portafolio, para la cual, el número de las neuronas de la capa de entrada, de neuronas ocultas, y la existencia de neuronas adaptativas, es encontrada usando un algoritmo genético. Chenoweth (1996) usa un sistema compuesto por dos PMC, uno

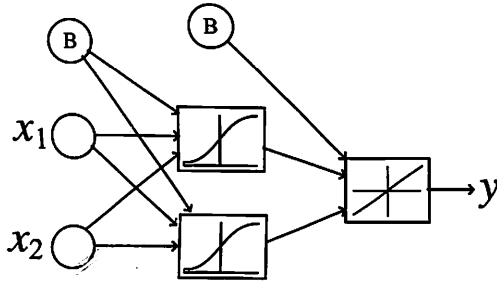


Figure 1: Perceptron multicapa con función de activación sigmoidea en la capa oculta y transferencia lineal en la de salida

para pronosticar retornos negativos del índice de S&P y otro para los positivos, y un sistema experto que da recomendaciones de compra/venta. Saad, Prokhorov y Wunsch (1998) comparan varios modelos de RNAs para el pronóstico de precios diarios de acciones. Zhang y Berardi (2001) emplean redes neuronales ensambladas para la predicción de tasas de intercambio. Hansen y Nelson. (1997) usan una RNA para pronosticar los ingresos por ventas para el estado de Utha. Doffner (1996) presenta un resumen sobre las principales arquitecturas de RNA utilizadas para el pronóstico de series de tiempo, así como de las ventajas que ellas pueden tener sobre los modelos tradicionales, basadas principalmente en su capacidad de modelar relaciones no lineales entre los datos. Sin embargo, una de las principales desventajas en el uso de modelos conexionistas es el mayor número de parámetros que tienen respecto a los modelos econométricos, haciendo su proceso de calibración más difícil y costoso computacionalmente.

2.3 ANFIS – Adaptive Neuro Fuzzy Inference System

Este modelo (Jang, 1993) es una red neuronal artificial que implementa un sistema difuso con su mecanismo de razonamiento, donde las reglas de inferencia son del tipo Takagi-Sugeno:

$$\text{IF } x \text{ IS } X \text{ AND } y \text{ IS } Y, \text{ THEN } z = f(x, y)$$

donde x y y son variables difusas, X y Y son conjuntos borrosos, y f es una función de x y y que aproxima el valor de z . Dicha función es usualmente una combinación lineal de las variables de entrada, cuyos coeficientes son estimados usando mínimos cuadrados. Cada capa de la red neuronal (véase la Figura 2) realiza un proceso específico en la inferencia de la salida del sistema:

Capa 1 : Entra x y calcula el valor de la función de

pertenencia

$$\mu_A(x) = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b}}$$

Capa 2 : Nodo Π – calcula la inferencia producto de cada regla

Capa 3 : Calcula el radio de activación de cada regla

$$\bar{w}_j = \frac{w_j}{\sum_{i=1}^N w_i}$$

Capa 4 : Función adaptativa $\bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i)$

Capa 5 : Salida del sistema: $\sum_i \bar{w}_i f_i$

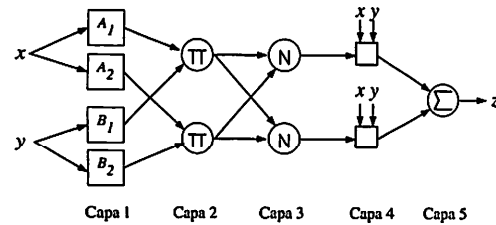


Figure 2: Arquitectura Típica de un Modelo ANFIS

Esta metodología se caracteriza por incorporar en un solo sistema, el conocimiento proveniente del experto, dado usualmente a través de reglas heurísticas, y el conocimiento oculto en los datos, obtenido a partir de un proceso de aprendizaje. Usualmente, el experto expresa las reglas heurísticas y define los conjuntos borrosos que representan los conceptos involucrados en las reglas. Posteriormente, las reglas pueden ser ajustadas a través de un proceso de optimización que busca minimizar alguna medida del error, en donde el vector de variables independientes es formado por los parámetros de las funciones de pertenencia y los coeficientes de los consecuentes de las reglas. En el caso particular en que los consecuentes de las reglas sean una combinación lineal de sus parámetros, puede realizarse su estimación óptima a través de mínimos cuadrados. Cuando solo está disponible la información histórica para construir el sistema, los parámetros son obtenidos por medio de un proceso de optimización.

ANFIS induce un particionamiento del espacio de entrada a través de los conjuntos borrosos en que se divide cada variable, asignando a cada partición una de las reglas difusas, y a diferencia de MARS, este particionamiento no es recursivo, tal como se muestra en la Figura 3. Sin embargo, esta forma de particionamiento, no es necesariamente la más adecuada, por lo que se han investigado otros esquemas de partición.

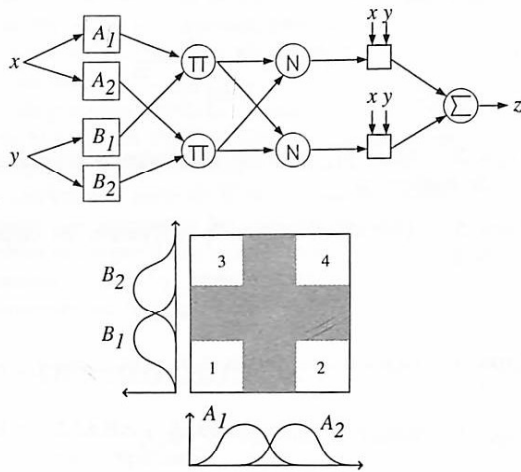


Figure 3: Partición del espacio de entrada realizado por ANFIS

La capacidad de ANFIS para aproximar funciones no lineales, está determinada por el número de conjuntos borrosos asignado a cada variable independiente; esto trae como desventaja que problemas con un número moderado de variables independientes, requieran una gran cantidad de información para realizar el entrenamiento.

Esto se debe a que un sistema simple, por ejemplo de 2 variables de entrada y dos conjuntos borrosos para cada variable, equivale a un modelo de 12 parámetros no lineales, que especifican la forma de los 4 conjuntos borrosos – tres por cada conjunto –, más otros 12 parámetros lineales, que provienen de 4 reglas difusas, donde el consecuente de cada regla tiene 3 parámetros. Se tiene entonces un total de 24 parámetros. Se ha demostrado para los perceptrones multicapa, que en el peor caso posible el número de datos debe ser al menos diez veces la cantidad de parámetros para lograr una generalización adecuada, aunque es de común aceptación usar al menos un factor de cinco. Extrapolando estos resultados para el sistema de ejemplo, se requerirían al menos 240 ejemplos en el peor caso, o al menos 120 para garantizar una generalización adecuada.

Si la relación entre las variables es altamente no lineal, donde no es suficiente este número de parámetros, puede aumentarse la capacidad de aproximación del modelo creando más conjuntos borrosos para las variables de entrada.

Para tener una idea de la capacidad no lineal de este modelo, puede compararse con un perceptron multicapa (PMC) con dos variables de entrada y una salida; aunque en la realidad ambos modelos son diferentes, esta comparación es válida en el sentido de extrapolar su capacidad no lineal como función del número de parámetros de cada uno de ellos. Con-

sidérese que este perceptron usa la función sigmoidea como activación en la capa de entrada y transferencia lineal en la capa de salida. De esta forma, puede suponerse que si se tienen H neuronas en la capa oculta, este PMC tendrá $3H$ parámetros no lineales que corresponden a los pesos de las conexiones entre la capa de entrada y la capa oculta, y $H + 1$ parámetros lineales que corresponden a los pesos de la capa oculta a la capa de salida, en total, $4H + 1$.

Así, un PMC con 5 neuronas en la capa oculta, tendrá un total de 21 parámetros, y uno con 6, un total de 25 parámetros. Es sabido que un PMC con esta cantidad de neuronas en la capa oculta, puede aprender relaciones bastante complejas entre variables, tal como lo evidencia el trabajo de Hwang (Hwang, Lay y Lippman, 1994).

De esta forma, el PMC considerado con 5 o 6 neuronas en la capa de entrada, será equivalente a un modelo ANFIS con dos conjuntos borrosos para cada una de las variables de entrada. Esto es, ambos sistemas tienen una capacidad similar para el modelamiento de sistemas no lineales. La principal consecuencia de esto, es que existe una región vacía sin modelos, en donde en un extremo está el modelo ANFIS mínimo en que cada variable tiene dos conjuntos borrosos, y la función de los consecuentes de las reglas, mientras que en el otro se encuentra el modelo de regresión lineal pura; en el caso de las redes neuronales, este espacio es ocupado por un perceptron con función lineal como activación de su salida, que es equivalente a un modelo de regresión lineal múltiple, luego el perceptron simple con una función no lineal de activación en la capa de salida, y posteriormente los PMC con 1, 2, o más neuronas en la capa oculta, en donde su capacidad de modelar problemas no lineales, es directamente proporcional a dicha cantidad.

3 LA SERIE DE MACKEY-GLASS

La serie de tiempo de Mackey-Glass es descrita por la ecuación diferencial:

$$\dot{x} = \frac{0.2x(t - \tau)}{1 + x^{10}(t - \tau)} - 0.1x(t)$$

cuyos puntos son obtenidos a partir de la integración numérica usando el método de Runge-Kutta de cuarto orden. Este problema ha sido utilizado por diferentes autores para el establecimiento de comparativos entre diferentes técnicas de pronóstico, y de modelos de regresión. El objetivo del experimento es usar los valores pasados y actual de esta serie de tiempo, esto es, $x[t - (D - 1) \cdot \Delta]$, $x[t - (D - 2) \cdot \Delta]$, ..., $x[t - \Delta]$ y $x[t]$, para pronosticar su valor futuro $x[t + P]$. Para este caso en particular, y con fines comparativos, se escogen los valores de $D = 4$ y $\Delta = P = 6$.

Los puntos fueron generados usando los siguientes parámetros: $h = 0.1$, $x[0] = 1.2$, $\tau = 19$, y $x[t] = 0$

para $t < 0$. La muestra de datos está formada por 500 puntos usados para la construcción de los distintos modelos, y 500 puntos para la validación de la serie. Previamente fueron eliminados los primeros 100 puntos, con el fin de eliminar los efectos de las condiciones iniciales. De esta forma, el primer conjunto de datos está formado por $x[100]$, $x[106]$, $x[112]$ y $x[118]$.

Como medida de error de ajuste de los modelos, algunos autores han usado el índice de error no dimensional (NDEI), que se define como la raíz del error cuadrático medio (RMSE) dividido por la desviación estándar de la serie, o la varianza relativa media que es igual a la raíz cuadrada del NDEI.

4 RESULTADOS OBTENIDOS

4.1 MARS

Para la selección final, se exploraron los modelos con M_{max} desde 1 hasta 28 funciones base, con un máximo de 5 interacciones entre variables. Tal como puede verse en la Figura 4, el punto de mínima para el GCV se presenta para el modelo de 24 funciones base.

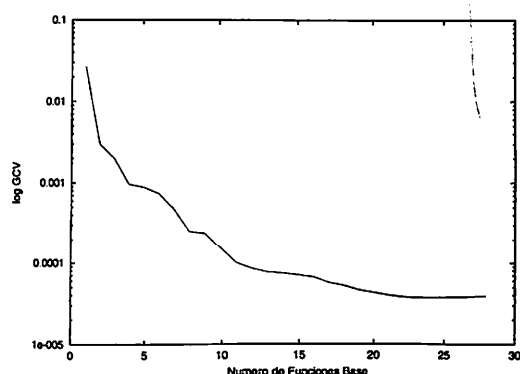


Figure 4: Validación Cruzada Generalizada vs. Cantidad de Funciones Base

La importancia de las variables en la disminución del error fue la siguiente:

Variable	-GCV	Importancia
$x[t-12]$	0.004	100.000
$x[t-18]$	0.001	50.044
$x[t]$	0.000351	26.437
$x[t-6]$	0.000161	16.160

Esto indica que la variable que mayor importancia tuvo en el particionamiento del espacio fue $x[t-12]$. La RMSE obtenida para el modelo con $M_{max} = 24$ para los datos de entrenamiento fue de 0.0069, y para los de validación de 0.0081. El modelo encontrado fue el siguiente:

$$\begin{aligned}
 BF1 &= \max(0, x[t-18] - 0.886); \\
 BF2 &= \max(0, 0.886 - x[t-18]); \\
 BF3 &= \max(0, x[t-12] - 0.848); \\
 BF4 &= \max(0, 0.848 - x[t-12]); \\
 BF5 &= \max(0, x[t] - 0.429); \\
 BF6 &= \max(0, x[t-18] - 0.860) * BF3; \\
 BF8 &= \max(0, x[t-12] - 1.156) * BF1; \\
 BF9 &= \max(0, 1.156 - x[t-12]) * BF1; \\
 BF10 &= \max(0, x[t-12] - 0.860) * BF2; \\
 BF11 &= \max(0, 0.860 - x[t-12]) * BF2; \\
 BF12 &= \max(0, x[t-6] - 0.518); \\
 BF13 &= \max(0, 0.518 - x[t-6]); \\
 BF14 &= \max(0, x[t-18] - 0.812) * BF5; \\
 BF15 &= \max(0, 0.812 - x[t-18]) * BF5; \\
 BF17 &= \max(0, 0.975 - x[t-18]) * BF12; \\
 BF19 &= \max(0, 0.627 - x[t-12]) * BF5; \\
 BF20 &= \max(0, x[t] - 1.254) * BF10; \\
 BF21 &= \max(0, 1.254 - x[t]) * BF10; \\
 BF22 &= \max(0, x[t-12] - 0.932) * BF2; \\
 BF24 &= \max(0, x[t-6] - 0.429) * BF14;
 \end{aligned}$$

$$x[t+6] = 1.030$$

$$\begin{aligned}
 &- 1.862 * BF1 - 0.106 * BF2 \\
 &- 1.515 * BF3 - 0.299 * BF4 \\
 &+ 0.282 * BF5 + 3.938 * BF6 \\
 &- 1.557 * BF8 + 3.138 * BF9 \\
 &+ 5.579 * BF10 - 0.536 * BF11 \\
 &+ 0.396 * BF12 - 0.372 * BF13 \\
 &- 0.878 * BF14 + 0.556 * BF15 \\
 &- 1.159 * BF17 - 0.596 * BF19 \\
 &+ 46.163 * BF20 - 7.551 * BF21 \\
 &- 2.549 * BF22 + 0.148 * BF24;
 \end{aligned}$$

4.2 Resultados obtenidos usando Perceptrones Multicapa (PMC)

El PMC usado tiene una capa oculta, 4 neuronas en la capa de entrada, una por cada una de las variables independientes consideradas, y una neurona en la capa de salida, que corresponde a la variable $x[t+6]$. Se usó la función sigmoidea para la activación de la capa oculta, y transferencia lineal para la capa de salida. Esta configuración de activación permite considerar como parámetros no lineales a los pesos de las conexiones entre la capa de entrada y la capa oculta, y como lineales a los pesos entre la capa oculta y la capa de salida, de tal forma que el modelo final puede ser comparado con el modelo ANFIS respecto a la cantidad y tipo de parámetros. Una ventaja adicional de este tipo de arquitectura, es que los parámetros lineales del PMC pueden estimarse como función de los pesos de la capa de entrada a la capa de salida, usando mínimos cuadrados.

Para el entrenamiento del PMC se usó el algoritmo de Quimiotactismo Bacteriano, con el que se obtuvieron los pesos de la capa de entrada a la capa oculta, mientras que los restantes fueron estimados de forma óptima usando mínimos cuadrados, tal como

ya se indicó.

Ya que la complejidad del PMC es controlado por la cantidad de neuronas en la capa oculta, se exploraron modelos desde una hasta 30 neuronas ocultas, seleccionándose el más simple con menor error de ajuste.

Puesto que el algoritmo de entrenamiento es sensible al punto inicial de arranque, se realizaron 20 optimizaciones para cada uno de los PMC considerados, y se seleccionaron los valores de los pesos con menor error de ajuste. Para cada una de las optimizaciones, los pesos de las conexiones fueron generados aleatoriamente.

Los PMC fueron entrenados usando los primeros 500 puntos de la serie para entrenamiento, y los restantes 500 para validación tal como ya se describió. Los errores, cantidad de parámetros y NDEI para cada uno de los PMC considerados se presentan en la Tabla 1.

Tal como puede observarse en la Figura 5, a partir de 17 neuronas en la capa oculta, no hay una disminución considerable de error, por lo que este se selecciona como el modelo final para el análisis.

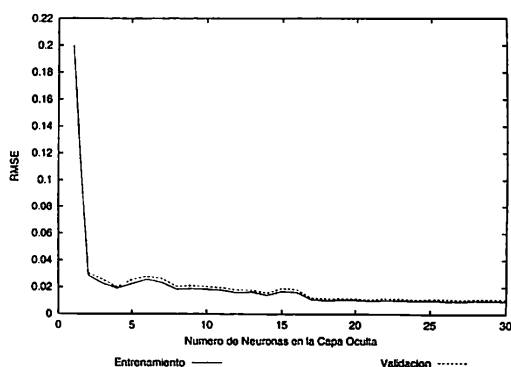


Figure 5: Variación del error cuadrático medio respecto al número de neuronas de la capa oculta

4.3 Resultados obtenidos usando ANFIS

En este modelo se asignaron dos conjuntos borrosos por cada una de las variables, obteniéndose un total de 16 reglas difusas, para un total de 104 parámetros, de los cuales 24 son no lineales y 80 lineales.

El proceso de entrenamiento fue realizado usando Regla Delta Generalizada, la cual fue aplicada a los parámetros de los conjuntos borrosos, mientras que los consecuentes de las reglas se estimaron en cada iteración por mínimos cuadrados.

Este caso es presentado por Jang como un ejemplo del uso de ANFIS en problemas de series de tiempo, reportando un RMSE de 0.0016 durante el entrenamiento, y de 0.0015 para la validación, obtenido después de 500 ciclos de entrenamiento.

En este experimento, la implementación del modelo fue realizada en Lenguaje C, mientras que la de Jang fue realizada en MATLAB. Para el primer ciclo de entrenamiento, el RMSE obtenido fue de 0.0018 – en donde los parámetros de las premisas fueron establecidos arbitrariamente, y los parámetros de las conclusiones fueron calculados usando mínimos cuadrados; valor inferior al reportado en la literatura. Al final de 200 ciclos de entrenamiento, se obtuvo un RMSE de 0.0008 durante la fase de entrenamiento y 0.0026 durante la validación.

5 ANÁLISIS DE LOS RESULTADOS

Un resumen de los resultados obtenidos se presenta en la Tabla 2. Puede observarse como MARS obtiene un desempeño superior al PMC seleccionado, usando un total de 22 coeficientes para las funciones base, y 13 puntos de partición del espacio así: 4 particiones para $x[t - 18]$, 6 para $x[t - 12]$, 2 para $x[t - 6]$ y 4 para $x[t]$. Estas particiones son más complejas que las realizadas por ANFIS, ya son realizadas de forma asimétrica. Esto hace que el modelo a pesar de usar funciones lineales por tramos presente un buen ajuste a los datos. Si se consideran tanto los coeficientes de las rectas, como los puntos de partición del espacio, es el modelo que usa el menor número de parámetros. Los valores pronosticados por MARS se presentan en la Figura 6 y sus errores en la Figura 7

Por su parte, los resultados obtenidos para el PMC fueron inferiores a los otros modelos utilizados. Un análisis detallado de los RMSE obtenidos para diferentes cantidades de neuronas en la capa oculta, indica que existe la posibilidad de obtener errores inferiores a los encontrados. Jang reporta un valor del NDEI de 0.02 para un PMC, pero no hay una descripción del PMC utilizado, y la referencia citada no está disponible.

Esto se evidencia en que a partir de 17 neuronas en la capa oculta, no hay una disminución importante del error, ni hay un aumento del error de validación. Ya que a medida que aumenta la cantidad de neuronas en la capa oculta aumenta la capacidad de aproximación del modelo, debería presentarse en algún punto el sobre-ajuste del modelo; esto es, la disminución progresiva del error de entrenamiento, con el aumento progresivo del error de validación, hecho que no se presenta en el modelo. Esto es debido a la dificultad que representa realizar la optimización de los parámetros no lineales – más de 68 –, manifestada por la complejidad de la superficie de error. Debe realizarse una búsqueda a mayor escala del punto de mínima para las redes con mayor cantidad de neuronas en la capa oculta. Este ejemplo muestra claramente la dificultad de entrenar PMC con un número alto de neuronas en la capa oculta. Igualmente, otro hecho en contra del PMC es que este no realiza un particionamiento del espacio, tal como

UNIVERSIDAD NACIONAL DE QUILMES
CENTRO DE INVESTIGACIONES
INSTITUTO DE BIBLIOTECAS
BIBLIOTECA MIPYAG

lo hacen las otras dos técnicas, e intenta ajustarse una sola función válida para todos los casos. Es interesante pensar en este punto, en un modelo ANFIS donde las conclusiones de las reglas son PMC. Los valores pronosticados por el PMC se presentan en la Figura 8 y sus errores en la Figura 9

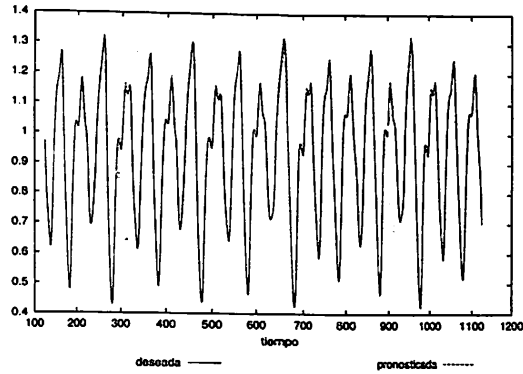


Figure 6: Valores deseados y pronosticados usando MARS

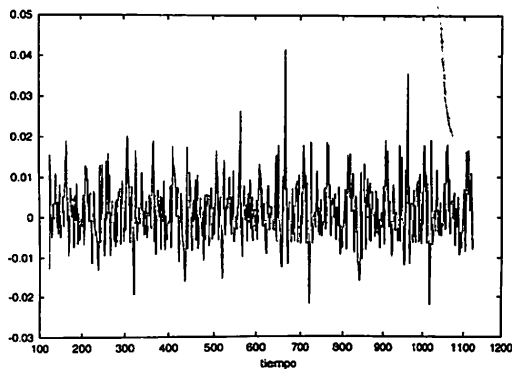


Figure 7: Errores de predicción usando MARS

6 PROPUESTA DE OBTENCIÓN DE MODELOS ANFIS CON COMPLEJIDAD CRECIENTE

En el caso del modelo ANFIS, se obtuvo un RMSE inferior al reportado, con un número inferior de iteraciones de la técnica de optimización. A diferencia de los resultados obtenidos por los otros modelos, se llegó a un punto de sobre-ajuste del modelo, tal como lo indica el valor del error de validación. Esto indica la posible existencia de modelos ANFIS con un número inferior de parámetros que puedan tener un desempeño adecuado. Los resultados obtenidos se presentan en la Figura 10 y la Figura 11.

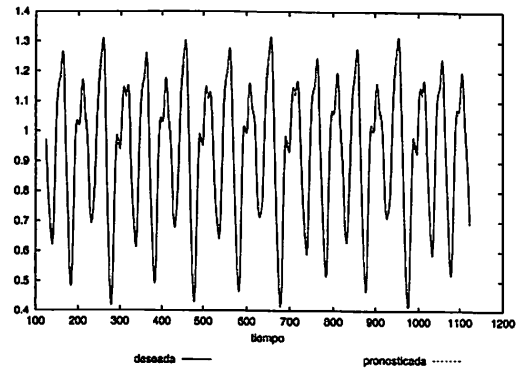


Figure 8: Valores deseados y pronosticados usando un PMC con 17 neuronas en la capa oculta

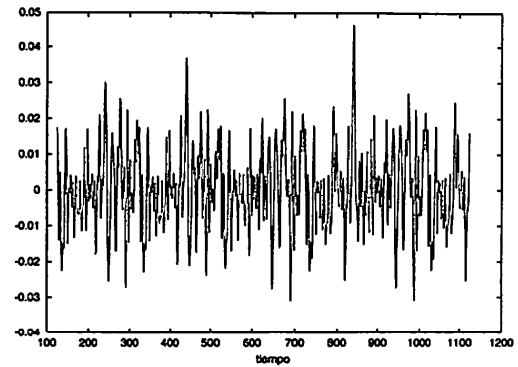


Figure 9: Errores de predicción usando un PMC con 17 neuronas en la capa oculta

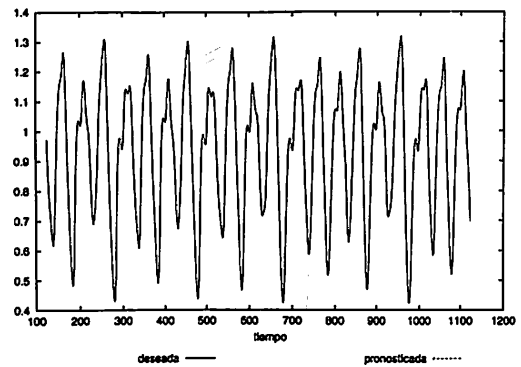


Figure 10: Valores deseados y pronosticados usando ANFIS

Table 1: Errores y Número de Parámetros para cada uno de los PMC considerados

Neuronas C. Oculta	RMSE Entre.	RMSE Val	NDEI	Parámetros Lineales	Parámetros no lineales	Total
1	0.1994	0.2001	0.8878	4	2	6
2	0.0286	0.0302	0.1339	8	3	11
3	0.0227	0.0256	0.1134	12	4	16
4	0.0190	0.0199	0.0882	16	5	21
5	0.0224	0.0252	0.1120	20	6	26
6	0.0257	0.0278	0.1232	24	7	31
7	0.0234	0.0264	0.1170	28	8	36
8	0.0185	0.0205	0.0908	32	9	41
9	0.0189	0.0210	0.0933	36	10	46
10	0.0184	0.0204	0.0903	40	11	51
11	0.0178	0.0195	0.0864	44	12	56
12	0.0159	0.0178	0.0791	48	13	61
13	0.0163	0.0178	0.0790	52	14	66
14	0.0140	0.0156	0.0691	56	15	71
15	0.0168	0.0190	0.0841	60	16	76
16	0.0162	0.0182	0.0807	64	17	81
17	0.0109	0.0121	0.0537	68	18	86
18	0.0101	0.0115	0.0511	72	19	91
19	0.0105	0.0117	0.0521	76	20	96
20	0.0105	0.0115	0.0508	80	21	101
21	0.0098	0.0110	0.0487	84	22	106
22	0.0103	0.0117	0.0520	88	23	111
23	0.0102	0.0115	0.0509	92	24	116
24	0.0098	0.0108	0.0480	96	25	121
25	0.0101	0.0110	0.0490	100	26	126
26	0.0096	0.0111	0.0495	104	27	131
27	0.0094	0.0106	0.0472	108	28	136
28	0.0099	0.0111	0.0493	112	29	141
29	0.0097	0.0111	0.0492	116	30	146
30	0.0097	0.0110	0.0488	120	31	151

Table 2: Resumen de los resultados obtenidos

Modelo	RMSE Entre.	RMSE Val.	NDEI Calculado	NDEI Reportado	Param. Lineales	Param. no lineales	Total
MARS	0.0069	0.0081	0.0357	—	23	0	23
PMC	0.0109	0.0121	0.0537	0.02	18	68	86
ANFIS	0.0008	0.0026	0.0113	0.007	80	24	104

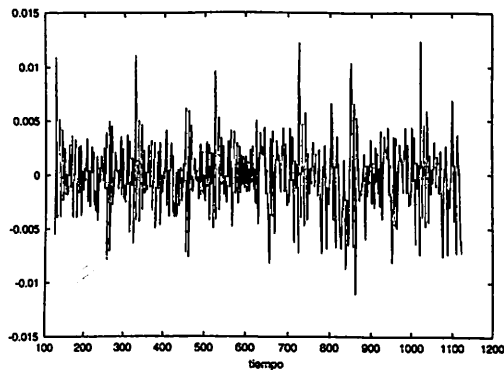


Figure 11: Errores de predicción usando ANFIS

Este modelo buscado, usa en las conclusiones de las reglas las mismas variables del modelo ANFIS usado hasta el momento, pero a diferencia del enfoque tradicional, se realiza un número menor de particiones del espacio, y por consiguiente se obtiene un número menor de reglas.

Formalmente el objetivo consiste en determinar si existe un modelo entre el caso lineal, y el modelo ANFIS usado en la experimentación. Para ello se pretende determinar si existe un conjunto $X_{opt} \subseteq X$ que particione el espacio de entrada en regiones, en donde para cada una de las cuales existe un modelo de regresión lineal, estimado sobre todas las variables del conjunto X . De esta forma, se llega a un conjunto de reglas donde los antecedentes están formados por las variables X_{opt} , y los consecuentes por las variables X .

Si se tiene el modelo ANFIS mínimo de dos variables y dos conjuntos por variables:

IF $x \in X_1$ AND $y \in Y_1$, THEN $z = a_1x + b_1y + c$
 IF $x \in X_1$ AND $y \in Y_2$, THEN $z = a_2x + b_2y + c$
 IF $x \in X_2$ AND $y \in Y_1$, THEN $z = a_3x + b_3y + c$
 IF $x \in X_2$ AND $y \in Y_2$, THEN $z = a_4x + b_4y + c$

Se pretende determinar si existe un modelo de la forma:

IF $x \in X_1$ THEN $z = a_1x + b_1y + c$
 IF $x \in X_2$ THEN $z = a_2x + b_2y + c$

o de la forma:

IF $y \in Y_1$ THEN $z = a_1x + b_1y + c$
 IF $y \in Y_2$ THEN $z = a_2x + b_2y + c$

que aproxime de una forma adecuada el sistema representado por los ejemplos disponibles. El algoritmo propuesto es el siguiente:

1. Estime un modelo de regresión lineal para las parejas de datos conocidos $(x^{(j)}, y^{(j)})$, con $j = 1, \dots, N$. Sea el error de ajuste de este modelo E_{opt} .

2. Sea $K = 1$ el número inicial de variables en los antecedentes del sistema de reglas.
3. Seleccione todos los subconjuntos de K variables de X , y construya para cada uno el sistema de reglas donde los antecedentes están formados por las variables seleccionadas en cada subconjunto y los consecuentes por las variables pertenecientes a X .
4. Entrene cada uno de los modelos del punto anterior y calcule el correspondiente error. Si hay un modelo con error inferior a E_{opt} , este es el nuevo modelo óptimo. Actualice el valor de E_{opt} . Si no existe tal modelo, se encontró el modelo óptimo y puede finalizarse el algoritmo.
5. $K = K + 1$
6. Si $K = 2^n$, prosiga con el proceso de modelado normal de ANFIS, en caso contrario vaya al paso 3.

El algoritmo propuesto realiza una búsqueda exhaustiva entre el modelo de regresión lineal múltiple y el modelo ANFIS mínimo. El proceso es detenido, ya sea porque no se pueden incluir más particiones, debido a que no hay suficientes datos, o porque el agregar una nueva partición no mejoró el error de aproximación, o finalmente, porque cada una de las variables fue dividida en dos conjuntos borrosos.

En este último caso, debería continuarse con la metodología clásica. El algoritmo descrito fue aplicado a los datos del experimento, obteniéndose los resultados presentados en la Tabla 3.

Puede observarse como el modelo 10, que usa como variables en las premisas a $x[t-12]$ y $x[t-18]$, tiene un desempeño superior a MARS – que tiene un error de entrenamiento de 0.0069 – y al PMC – 0.0109 –. Este modelo tiene un total de 12 parámetros no lineales y 10 no lineales, cantidad inferior a los otros modelos. Los pronósticos para este modelo se presentan en la Figura 13 y sus errores en la Figura 12. Los modelos 13, 14 no alcanzan un RMSE inferior al reportado por (Jang, 1993).

Ordenando los modelos por la magnitud del error y teniendo en cuenta las variables que forman las premisas de las reglas, se encuentra que el orden de importancia de las variables es: $x[t-12]$, $x[t-18]$, $x[t]$ y finalmente $x[t-6]$, que corresponde al mismo orden encontrado al usar MARS.

Table 3: Resultados obtenidos para modelos ANFIS con diferentes particiones del espacio de entrada

Modelo	Premisa	RMSE Entre.	RMSE Val.	NDEI Calculado	Param. Lineales	Param. no lineales	Total
1	$x(t)$	0.0609	0.0626	0.2777	6	10	16
2	$x(t-6)$	0.0544	0.0577	0.2559	6	10	16
3	$x(t-12)$	0.0137	0.0145	0.0644	6	10	16
4	$x(t-18)$	0.0284	0.0305	0.1352	6	10	16
5	$x(t)$ $x(t-6)$	0.0258	0.0274	0.1214	12	20	32
6	$x(t)$ $x(t-12)$	0.0089	0.0090	0.0398	12	20	32
7	$x(t)$ $x(t-18)$	0.0137	0.0157	0.0697	12	20	32
8	$x(t-6)$ $x(t-12)$	0.0096	0.0096	0.0428	12	20	32
9	$x(t-6)$ $x(t-18)$	0.0148	0.0178	0.0791	12	20	32
y10	$x(t-12)$ $x(t-18)$	0.0044	0.0042	0.0188	12	20	32
11	$x(t)$ $x(t-6)$ $x(t-12)$	0.0084	0.0097	0.0431	18	40	58
12	$x(t)$ $x(t-6)$ $x(t-18)$	0.0051	0.0076	0.0337	18	40	58
13	$x(t)$ $x(t-12)$ $x(t-18)$	0.0023	0.0034	0.0151	18	40	58
14	$x(t-6)$ $x(t-12)$ $x(t-18)$	0.0026	0.0034	0.0150	18	40	58
15	$x(t)$ $x(t-6)$ $x(t-12)$ $x(t-18)$	0.0008	0.0026	0.0113	24	80	104

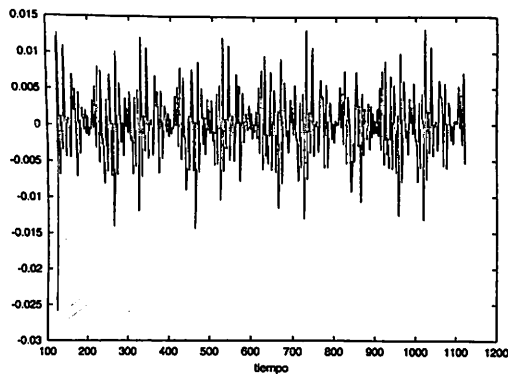


Figure 12: Errores de predicción usando el modelo 10 -ANFIS-

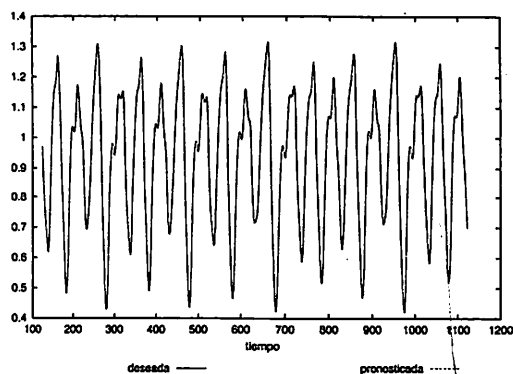


Figure 13: Valores deseados y pronosticados usando el modelo 10 -ANFIS-

7 CONCLUSIONES

Se han presentado las principales características de diferentes modelos utilizados, realizándose un comparativo respecto a sus algoritmos de ajuste y su capacidad de pronóstico de funciones no lineales, versus la cantidad de parámetros lineales y no lineales.

MARS a pesar de ajustar modelos lineales, tiene un desempeño superior al PMC, debido al particionamiento que realiza del espacio de variables independientes, con un número inferior de parámetros. A diferencia de sus dos competidores, esta metodología requiere unos pocos segundos para recorrer todo el espacio de modelos y seleccionar el mejor, mientras que los demás requieren al menos varios minutos.

El PMC, a diferencia de los otros dos modelos, intenta ajustar una única función no lineal a todo el espacio muestral, sin realizar ningún tipo de particionamiento del espacio. A pesar de ser una función no lineal, esta estrategia es poco exitosa, y lo coloca en el último lugar respecto a sus competidores.

Se han encontrado errores inferiores a los reportados por Jang para el modelo ANFIS, y se ha

mostrado que este puede alcanzar fácilmente el sobreentrenamiento. Se presenta una metodología para la obtención de modelos ANFIS con complejidad creciente, los cuales poseen requerimientos inferiores de información, los cuales pueden ser entrenados con los algoritmos convencionales de optimización, y que pueden ser usados en problemas, donde no hay suficientes datos para usar el particionamiento tradicional, en donde a todas las variables se les asigna, al menos, dos conjuntos borrosos.

REFERENCES

- Chenoweth, T. (1996), A neural network based system for predicting future returns of the S&P Stock Index, PhD thesis, Washington State University.
- Doffner, G. (1996), 'Neural network for time series processing', *Neural Networks World* (4), 447-468.
- Eberhart, R. C. y Dobbins, R. W. (1990), *Neural Network PC Tool: A Practical Guide*, Academic Press, Inc.
- Friedman, J. H. (1991), 'Multivariate adaptive regression splines', *Annals of Statistics* 19(1).
- Hansen, J. V. y Nelson, R. D. (1997), 'Neural network and time series methods: A synergic combination in state economic forecasts', *IEEE Trans on Neural Networks*.
- Hwang, J., Lay, S. y Lippman, A. (1994), 'Nonparametric multivariate density estimation: a comparative study'.
- Jang, J.-S. R. (1993), 'Anfis: Adaptive-network-based fuzzy inference system', *IEEE Transactions on Systems, Man, and Cybernetics* 23, 665-684.
- Kosko, B. (1992), *Neural Networks and Fuzzy Systems: A Dynamical Approach to Machine Intelligence*, Englewood Cliffs, NJ, Prentice-Hall.
- Masters, T. (1993), *Practical Neural Network Recipes in C++*, first edn, Academic Press, Inc.
- Saad, E., Prokhorov, D. y Wunsch, D. (1998), 'Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks', *IEEE Trans. on Neural Networks* 9, 1456-1470.
- Westheider, O. (1997), Predicting stock index returns by means of genetically engineered neural networks, PhD thesis, University of California.
- Zhang, G. P. y Berardi, V. L. (2001), *J. of Operational Research Society* 52, 652-664.