

# ENTRENAMIENTO DE REDES NEURONALES RECURRENTE PARA SISTEMAS DINÁMICOS TIPO NARMAX Y NOE

RUDY VELÁSQUEZ E., GONZALO ACUÑA L.

*Departamento de Ingeniería Informática, Universidad de Santiago de Chile, USACH  
Santiago, CHILE*

*E-mails: rudy.velasquez@gmail.com, gacuna@usach.cl*

**Abstract**— This work focuses on different training methods of a recurrent neural network acting as a non linear dynamical model of a system. Depending on how noise affects the system it is necessary to make a choice between a directed or semi-directed training method. A Matlab® program was developed which includes an adaptation of a first-order gradient algorithm to a second-order Levenberg-Marquardt back-propagation-through-time algorithm. A comparative analyses between the two already mentioned methods applied to recurrent neural networks acting as non-linear models of two different dynamic systems (NOE, NARMAX) was performed. Some statistical indexes like the Index of Agreement (IA), the Akaike Criterion (useful for evaluating the model complexity) and the Prediction Error Variance were used. Results verified the need of a good choice of the training method. In fact when an incorrect method was chosen the prediction error variance becomes greater than the noise variance, the Akaike index increases and the IA slightly decreases.

**Keywords**— Backpropagation Through Time, Recurrent Neural Networks, Non-linear Dynamic Systems.

**Resumen**— En este trabajo se lleva a cabo una comparación entre diferentes métodos de entrenamiento de redes neuronales recurrentes para sistemas dinámicos no lineales afectados de diferente manera por un ruido gaussiano de media cero. Dependiendo de la forma en que el ruido afecta al sistema es necesario escoger entre un entrenamiento dirigido o semi-dirigido. Un programa en Matlab® se desarrolló, el cual incluye una adaptación de un algoritmo de gradiente de primer orden para retropropagación en el tiempo en otro de segundo orden tipo Levenberg-Marquardt. Se utilizaron índices estadísticos como el Índice de Adecuación, el Criterio de Akaike (para evaluar la complejidad del modelo) y la Varianza del Error de Predicción. Los resultados revelan que una mala elección del método de entrenamiento deriva en que la Varianza del Error de Predicción es mayor que la varianza del ruido de los datos, el Criterio de Akaike crece al mismo tiempo que el IA levemente disminuye.

**Palabras-clave**— Retropropagación en el tiempo, Redes Neuronales Recurrentes, Sistemas Dinámicos no Lineales.

## 1 Introducción

La identificación de sistemas [Ljung, 1987] ha tenido gran relevancia en diversas áreas del conocimiento tales como la física, química, biología, economía, etc.; ya que los sistemas dinámicos —aquellos en los cuales el valor de la salida no depende sólo de los valores de sus entradas en ese mismo instante, sino que también de sus valores pasados— abundan en el medio ambiente y se requieren modelos para su análisis, predicción, simulación, diseño y control. Estos modelos necesitan simular el comportamiento real en los casos en que existe un conocimiento previo limitado de la estructura del sistema.

La mejor opción para modelar sistemas dinámicos no lineales es mediante redes neuronales recurrentes [Nerrand, 1993]. Para facilitar el aprendizaje de dichas redes, éstas pueden ser puestas bajo una *forma canónica* [Dreyfus, 1998], que corresponde a una representación mínima de estado compuesta por neuronas sin retardo o estáticas, en donde las recurrencias son solamente externas.

Recientemente, se han propuesto diversas alternativas para la identificación de estos modelos, entre ellos el uso de Wavelet en modelos polinomiales NARMAX (Non-linear Auto-Regressive Moving Average with eXogenous inputs) [Billings, 2005] y modelos NARX (Non-linear Auto-Regressive with eXogenous inputs) [Wei, 2006], aplicación a siste-

mas fuzzy del algoritmo BPTT (Backpropagation Through Time) [Werbos, 1990] mediante una optimización Quasi-Newton [Savran, 2007], y trabajos basados en LDDN (Layered Digital Dynamic Network) que permiten calcular de forma eficiente las derivadas de la red neuronal, tomando en cuenta los enfoques BPTT y RTRL (Real-Time Recurrent Learning) [De Jesus, 2007].

Una alternativa interesante desde el punto de vista de su simplicidad, es considerar los sistemas dinámicos no lineales modelados a través de estructuras del tipo NOE (Non-linear Output Error) [Ljung, 1987] y NARMAX [Leontaritis, 1987] que incluyan perturbaciones o ruido. Su identificación se basa en el aprendizaje semi-directo, a través de un algoritmo de gradiente descendente simple [Nerrand, 1993], el cual es modificado a un algoritmo de segundo orden como Levenberg-Marquardt, lo que permite obtener buenos resultados en un número menor de iteraciones. Para el entrenamiento se utiliza el enfoque de la *Retropropagación a través del Tiempo* (BPTT) [Werbos, 1990], el cual usa una arquitectura de red neuronal que es reproducida tantas veces según un horizonte de tiempo determinado. El ajuste de los pesos de la red considera todo el horizonte de predicción en donde para el cálculo del gradiente se toman en cuenta los errores según una ventana de tiempo definida dentro del horizonte.

Los resultados obtenidos muestran que se puede realizar una buena predicción de un sistema dinámico no lineal dentro de un horizonte de tiempo relati-

vamente pequeño y que el ruido que afecta el modelo puede ser correctamente identificado si se utiliza el tipo de entrenamiento adecuado. La selección del modelo neuronal se evalúa según un Índice de Adecuación (IA – Index of Agreement) [Willmott, 1981] y por el Criterio de Akaike (AIC) [Leontaritis, 1987].

La estructura de este documento incluye las siguientes secciones: una introducción a los sistemas dinámicos contemplados en el documento, uso redes neuronales para la identificación de este tipo de sistemas, resultados obtenidos aplicados en un proceso no lineal y las conclusiones finales.

## 2 Sistemas Dinámicos

Un sistema es un conjunto de unidades interrelacionadas que producen señales observables, las cuales se conocen como *salidas*. Las variables externas que son impuestas al sistema se denominan *entradas*, también se pueden considerar las *perturbaciones*, las cuales no pueden ser manipuladas.

En los sistemas dinámicos, el valor de la salida no depende sólo de los valores de sus entradas en ese mismo instante, sino que también de sus valores pasados. Un modelo representa matemáticamente el funcionamiento de estos sistemas, los cuales pueden ser de diferentes estructuras y diversas formulaciones matemáticas. Un punto a considerar es como las perturbaciones, o ruido, afectan al sistema.

### 2.1 Modelo con ruido a la salida NOE

El modelo NOE (Non-linear Output Error) representa un sistema afectado por un ruido a la salida de éste. Corresponde a la versión lineal del modelo OE (Output Error) [Ljung, 1987], se rige por:

$$\begin{cases} x_p(k) = f[x_p(k-1), \dots, x_p(k-N_y), \dots \\ \dots, u(k-1), u(k-2), \dots, u(k-N_u)] \\ y_p(k) = x_p(k) + w(k) \end{cases} \quad (1)$$

donde  $\{w(k)\}$  es una secuencia de variables aleatorias independientes de media cero y varianza  $\sigma^2$ . Su predictor [Ljung, 1987] está dado por:

$$\begin{aligned} y(k) &= f[y(k-1), \dots, y(k-N_y), \dots \\ &\dots, u(k-1), u(k-2), \dots, u(k-N_u)] \end{aligned} \quad (2)$$

### 2.2 Modelo NARMAX

El modelo NARMAX (Non-linear Auto-Regressive Moving Average with eXogenous inputs) corresponde a una extensión del modelo lineal ARMAX [Leontaritis, 1985]. El modelo obedece está dado por:

$$\begin{aligned} y_p(k) &= f[y_p(k-1), \dots, y_p(k-N_y), \dots \\ &\dots, u(k-1), \dots, u(k-N_u), \dots \\ &\dots, w(k-1), \dots, w(k-N_w)] + w(k) \end{aligned} \quad (3)$$

donde  $\{w(k)\}$  es una secuencia de variables aleatorias independientes de media cero y varianza  $\sigma^2$ . Su predictor [Leontaritis, 1985] está dado por:

$$\begin{aligned} y(k) &= \psi[y_p(k-1), \dots, y_p(k-N_y), \dots \\ &\dots, u(k-1), \dots, u(k-N_u), \dots \\ &\dots, e(k-1), \dots, e(k-N_e)] \end{aligned} \quad (4)$$

Teniendo en cuenta que:

$$e(k) = y_p(k) - y(k) \quad (5)$$

## 3 Redes Neuronales

Las redes neuronales son funciones universales de aproximación [Cybenko, 1989], por lo que pueden ser aplicadas a un gran número de problemas reales de alta complejidad. En este trabajo se consideran redes con una capa oculta, función de activación no lineal y neurona de salida lineal. Se pueden distinguir dos tipos de redes: las *redes estáticas*, en donde existen solamente conexiones hacia adelante, también conocidas como redes *feedforward*. Por otro lado, se encuentran las *redes recurrentes* o *dinámicas*, las cuales presentan conexiones recurrentes entre neuronas, conocidas también como *redes feedback*.

### 3.1 Redes Neuronales Recurrentes

Para facilitar el aprendizaje de estas redes, éstas pueden ser puestas bajo una *forma canónica* [Dreyfus, 1998], que corresponde a una representación mínima de estado compuesta por neuronas sin retardo o estáticas. El comportamiento de una red neuronal dinámica se puede escribir como (Figura 1):

$$\begin{aligned} S(k+1) &= \varphi[S(k), I(k); C] \\ Y(k) &= \psi[S(k), I(k); C] \end{aligned} \quad (6)$$

donde  $I(k)$  es el vector de entradas externas,  $S(k)$  el vector de las variables de estado,  $Y(k)$  el vector de salidas y  $S(k+1)$  es el vector de variables de estado en el instante  $k+1$ . Las funciones realizadas por la red neuronal estática son representadas por  $\varphi(\cdot)$  y  $\psi(\cdot)$ ,  $C$  corresponde a los coeficientes.

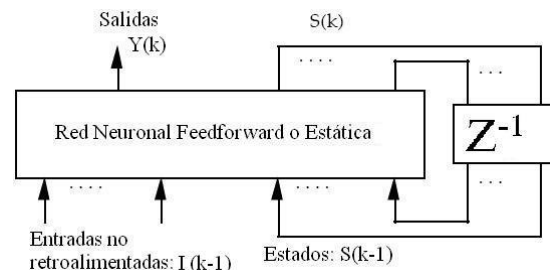


Figura 1. Forma canónica de una red neuronal dinámica

### 3.2 Predictores y esquemas de aprendizaje

Para estimar los coeficientes de la red neuronal, se de utilizar el algoritmo de aprendizaje adecuado. A con-

tinuación se muestran los predictores asociados a cada modelo y su esquema de aprendizaje.

### 3.2.1 Predictor asociado a un modelo NOE

La identificación del predictor NOE impone la utilización de una red neuronal dinámica. Dicho predictor puede ser realizado mediante una red totalmente conectada como lo muestra la figura 2.

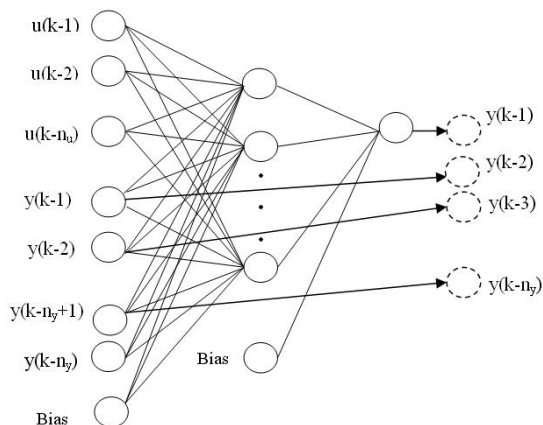


Figura 2. Red Neuronal Predictora NOE

El sistema de aprendizaje que utiliza este predictor es representado esquemáticamente en la figura 3; dicho enfoque se denomina “semi-dirigido”, ya que los valores del vector de estado son impuestos solamente al principio del entrenamiento, en la primera copia. Posteriormente, los valores del vector de estado son calculados por la red y desfasados en el tiempo para su uso. El gradiente se calcula mediante la retropropagación de los N errores que intervienen sobre las N copias en cascada.

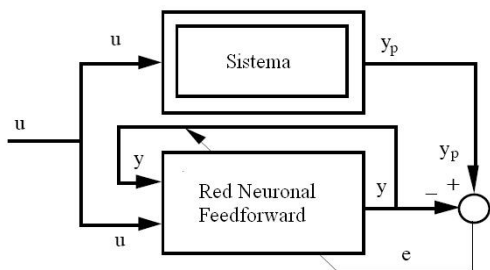


Figura 3. Aprendizaje semi-dirigido NOE

### 3.2.2 Predictor asociado a un modelo NARMAX

La identificación del predictor NARMAX sugiere el uso de una red neuronal recurrente. Dicho predictor puede ser realizado mediante una red totalmente conectada como lo muestra la figura 4.

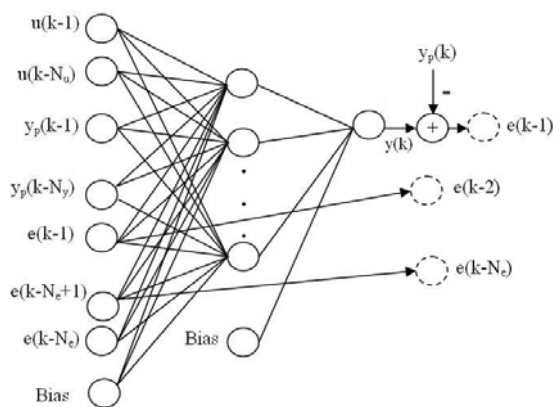


Figura 4. Red Neuronal Predictora NARMAX

El sistema de aprendizaje que utiliza este predictor es representado esquemáticamente en la figura 5 y se denomina “semi-dirigido”, tal como en 3.2.1.

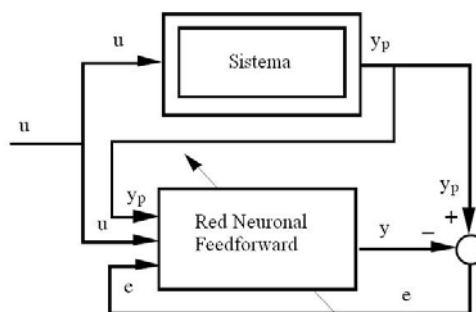


Figura 5. Aprendizaje semi-dirigido NARMAX

### 3.3 Consideraciones del algoritmo utilizado

El algoritmo utilizado para el aprendizaje semi-dirigido se basa en un trabajo anterior [Nerrand, 1993]. Dicho algoritmo se basa en la retropropagación a través del tiempo (BPTT) [Werbos, 1990], en donde se utiliza una red neuronal en cascada formando bloques para representar los diversos instantes de tiempo; la arquitectura de la red depende del modelo a identificar (Figura 3, Figura 5). El número total de bloques representa el horizonte y la ventana representa los bloques contemplados para el cálculo del gradiente (Figura 6).

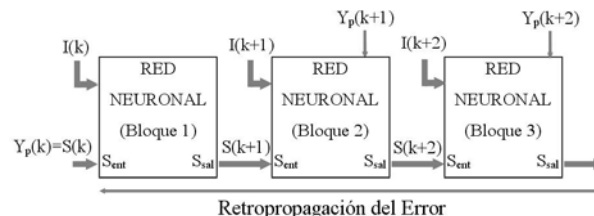


Figura 6. Red Neuronal en cascada, con horizonte 3 y ventana 2

El algoritmo fue escrito en lenguaje de programación MATLAB®, en donde se aprovechan ciertas funciones predefinidas en su Toolbox de Redes Neuronales. Se utilizan redes de una capa oculta

con función de transferencia logaritmo sigmoideal. La capa de salida corresponde a una neurona de activación lineal. Los datos de entrada son normalizados entre 0.1 y 0.9 para evitar la saturación de la sigmoideal.

Se realiza una adaptación a un algoritmo de segundo orden como Levenberg-Marquardt. Dicho algoritmo requiere el cálculo del Hessiano, que está formado por la matriz de derivadas de segundo orden de las salidas de las neuronas con respecto a los pesos. La matriz hessiana  $H$  puede ser aproximada según:

$$H = J^T J \quad (7)$$

donde  $J$  representa la matriz jacobiana. El gradiente  $g$  está definido por:

$$g = J^T e \quad (8)$$

donde  $e$  es el vector de errores de la red, por lo tanto si tenemos que  $e=I$  se obtendría la matriz jacobiana transpuesta o, al menos, una aproximación de ésta. Por lo que se aplica en cada bloque un error igual a 1 independientemente del número de la ventana. Esto se realiza ya que el algoritmo descrito en [Nerrand, 1993] sólo contempla el cálculo de gradientes y no de jacobianos. Así se obtienen los elementos necesarios para aplicar la modificación de pesos mediante Levenberg-Marquardt:

$$\theta_{k+1} = \theta_k - [J^T J + \mu I]^{-1} J^T e = \theta_k - [H + \mu I]^{-1} g \quad (9)$$

donde  $\mu$  es un escalar que representa el paso de modificación de gradiente mientras que  $I$  representa la matriz identidad.

### 3.4. Selección del mejor modelo

Para seleccionar el mejor modelo se recurrirán a tres índices de error, uno de ellos es el Índice de Adecuación (IA – del inglés Index of Agreement) [Willmott, 1981], en donde el modelo es aceptado si  $IA > 0.9$ :

$$IA = 1 - \frac{\sum_{i=1}^n (y_i - y_{pi})^2}{\sum_{i=1}^n (|y_i| + |y_{pi}|)^2} \quad (10)$$

Con:

$$y_i' = y_i - y_m$$

$$y_{pi}' = y_{pi} - y_m$$

donde  $y_i$  corresponde a los valores predichos,  $y_{pi}$  a los valores deseados,  $n$  al número total de datos y  $y_m$  al promedio de los valores predichos.

El otro índice corresponde al Criterio de Información de Akaike (AIC – Akaike's Information Criterion) [Leontaritis, 1987], el cual considera una penalización según la complejidad del modelo escogiéndose

el que presente el menor valor, lo cual es importante al comparar modelos NOE con modelos NARMAX, ya que éste último recibe un mayor número de entradas (ya que considera los errores pasados):

$$AIC = 2 * N * \log \left( \sum_{i=1}^n (y_i - y_{pi})^2 \right) + 4 * P \quad (11)$$

donde  $y_i$  corresponde a los valores predichos,  $y_{pi}$  a los valores deseados,  $n$  al número total de datos,  $N$  al número total de muestras y  $P$  al número de coeficientes o pesos de la red.

El tercer índice corresponde a la varianza del error de predicción, VEP.

## 4 Aplicación en un proceso no lineal

Se realizaron simulaciones en MATLAB tomando como referencia una ecuación no lineal de segundo orden, la cual es descrita en [Urbani, 1994]:

$$y_p(k) = \psi[y_p(k-1), y_p(k-2), u(k-1)]$$

$$\Downarrow$$

$$y_p(k) = 50 \tanh[2 * 10^{-3} (\psi_1 - \psi_2)] + 0.5 * u(k-1)$$

con

$$\psi_1 = \frac{(24 + y_p(k-1))}{3} y_p(k-1) \quad (12)$$

$$\psi_2 = 8 * \frac{u(k-1)^2}{1 + u(k-1)^2} y_p(k-2)$$

Se pretende entrenar una red neuronal dependiendo de cómo el ruido afecta al proceso estudiado, generando los modelos NOE y NARMAX a utilizar. Se analizan las consecuencias de utilizar un algoritmo incorrecto para la identificación de un modelo determinado. Para representar el ruido  $w$  se utilizan valores aleatorios de media cero que oscilan en el intervalo [-0.25, 0.25] y con una varianza de 0.0033. Para la medición de resultados se emplean el IA [Willmott, 1981], el Criterio de Akaike (AIC) [Leontaritis, 1987] y la VEP. Hipotéticamente este último valor debe ser similar a la varianza del ruido (VR) considerado en el proceso (al ser identificado correctamente).

### 4.1 Modelo NOE

Se procede a identificar un proceso NOE, el cual se rige por la siguiente ecuación:

$$x_p(k) = \psi[x_p(k-1), x_p(k-2), u(k-1)]$$

$$y_p(k) = x_p(k) + w(k) \quad (13)$$

#### 4.2.1 Hipótesis incorrecta: NARMAX

Al realizar la identificación suponiendo que el proceso se trata de un modelo NARMAX, se utiliza un aprendizaje semi-dirigido, tomando un horizonte entre 2 y 5 unidades de tiempo. Al ir aumentando el horizonte de tiempo los índices se van deteriorando (AIC=-10570.46 para h=2; AIC=-6763.43 para h=5), no se obtiene una identificación satisfactoria.

#### 4.2.2 Hipótesis correcta: NOE

Al realizar la identificación suponiendo que el proceso se trata de un modelo NOE (Figura 7), se utiliza un aprendizaje semi-dirigido, entre un horizonte de 2 y 5 unidades de tiempo. Al ir aumentando el tamaño del horizonte los índices van mejorando (AIC=-10783.02 para h=2; AIC=-11094.95 para h=6), hasta llegar a un horizonte de 6, en donde la VEP tiene un valor de 0.0036, valor muy similar a la varianza del ruido inducido (VR=0.0033). La identificación realizada es considerada satisfactoria.

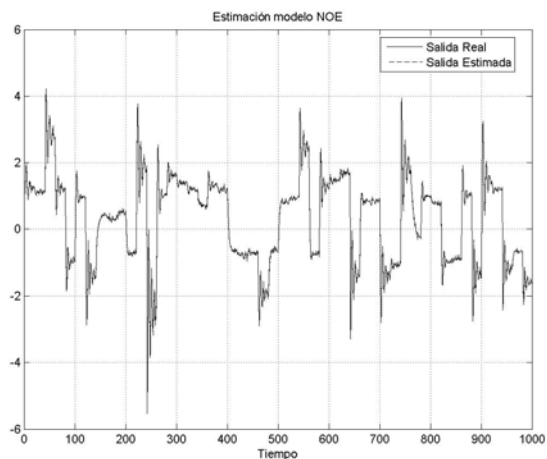


Figura 7. Predicción mediante Red Neuronal Recurrente identificada con algoritmo semi-dirigido para sistema NOE

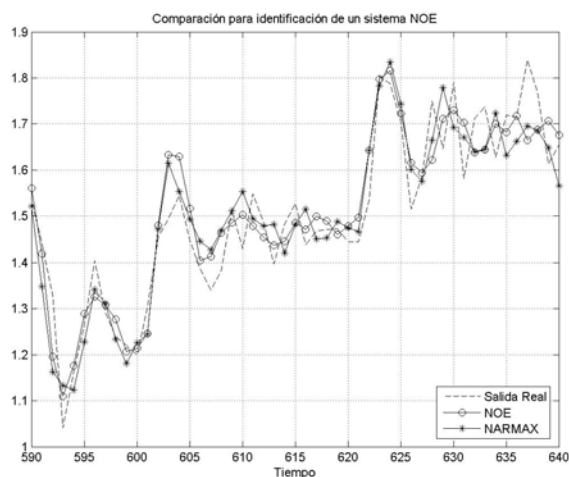


Figura 8. Comparación según las distintas hipótesis para el intervalo [580,640]

En la tabla 1 se ilustran los mejores resultados obtenidos para el modelo NOE:

Tabla 1. Mejores resultados obtenidos para un modelo NOE. Varianza de ruido, VR=0.0033

Hipótesis	h	AIC	IA	VEP
NOE	6	-11095.26	0.9995	0.0036
NAR-MAX	2	-10570.02	0.9993	0.0047

#### 4.2 Modelo NARMAX

Se procede a identificar un proceso NARMAX, el cual se rige por la siguiente ecuación:

$$y_p(k) = \psi[y_p(k-1), y_p(k-2), \dots, u(k-1), w(k-1)] + w(k) \quad (14)$$

#### 4.3.1 Hipótesis incorrecta: NOE

Al realizar la identificación suponiendo que el proceso se trata de un modelo NOE, se utiliza un aprendizaje semi-dirigido, con un horizonte de tiempo de longitud entre 2 y 5. Al ir incrementando el horizonte, los distintos índices de error van mejorando lentamente (AIC=-10350.53 para h=2; AIC=-10360.67 para h=5), pero el sistema aún no ha sido identificado satisfactoriamente, ya que el menor valor de la VEP corresponde a 0.0042.

#### 4.3.2 Hipótesis correcta: NARMAX

Al realizar la identificación suponiendo que el sistema corresponde a un modelo NARMAX (Figura 9), se utiliza un aprendizaje semi-dirigido, considerando un horizonte entre 2 y 6 unidades de tiempo. Los índices obtenidos en un horizonte de 2 son aceptables, ya que se obtiene un VEP igual a 0.0036 y un AIC de -11066.85, los cuales corresponden a los menores valores considerando la hipótesis anterior.

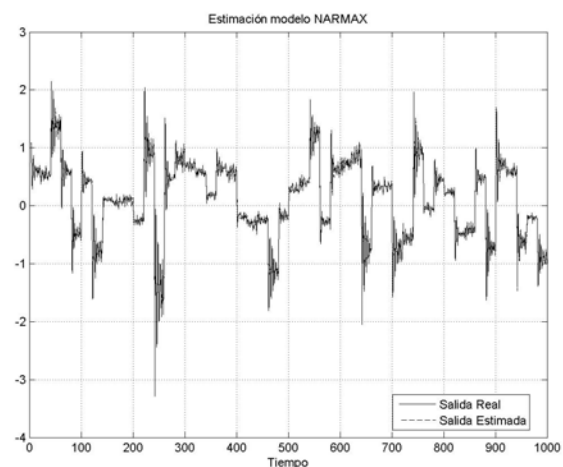


Figura 9. Predicción mediante Red Neuronal Recurrente identificada con algoritmo semi-dirigido para sistema NARMAX

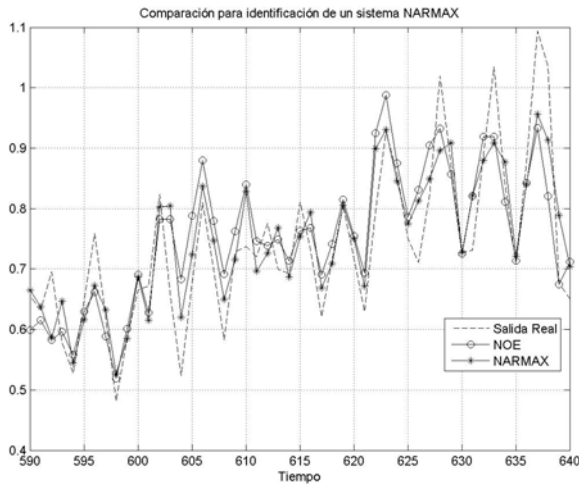


Figura 10 Comparación según las distintas hipótesis para el intervalo [590,640]

En la tabla 2 se ilustran los mejores resultados obtenidos para el modelo NARMAX:

Tabla 2. Mejores resultados obtenidos para un modelo NARMAX. Varianza de ruido, VR=0.0033

Hipótesis	h	AIC	IA	VEP
NOE	2	-10782.82	0.9977	0.0042
NARMAX	2	-11066.85	0.9981	0.0036

## 5 Conclusiones

Es importante elegir el algoritmo de entrenamiento adecuado para identificar un sistema dinámico en presencia de ruido. Según los resultados se aprecia que la mejor aproximación ocurre cuando la varianza del error de predicción es similar a la varianza del ruido que afecta al proceso. Si bien para la identificación de modelos NARMAX existe una cierta ventaja (la red tiene un mayor número de entradas ya que se toma en cuenta el retardo de los errores), se considera una compensación mediante el criterio de Akaike, ya que considera el número de pesos que comprende la red neuronal.

En general se obtuvieron buenos resultados en la identificación, ya que el IA obtenido generalmente superaba el valor de 0.9. La adaptación del algoritmo descrito en [Nerrand, 1993], a un algoritmo de segundo orden como Levenberg-Marquardt obtuvo resultados positivos en términos de convergencia y en tiempos de cómputo, aunque cabe destacar que los tiempos se incrementan notoriamente a medida que se aumenta el horizonte de predicción.

## Agradecimientos

Se agradece financiamiento parcial de proyecto FONDECYT 1040208 y DICYT 06-0719AL, Chile.

## Referencias Bibliográficas

- Billings, S.A. and Wei, H.L. (2005). *The Wavelet-NARMAX Representation: A Hybrid Model Structure Combining Polynomial Models with Multiresolution Wavelet Decompositions*. International Journal of Systems Science, Vol 36, No. 3, pp. 137-152.
- Cybenko, G. (1989). *Approximation by Superposition of a Sigmoidal Function*. Mathematics of Control. Signals and Systems, Vol 2, pp. 303-314.
- De Jesus, O. and Hagan, M.T. (2007). *Backpropagation Algorithms for a Broad Class of Dynamics Networks*. IEEE Transactions on Neural Networks, Vol. 18, No. 1.
- Dreyfus, G. and Idan, Y. (1998). *The Canonical Form of Discrete-Time Non-Linear Models*. Neural Computation. Vol 10, No. 1, pp. 133-164.
- Leontaritis, I.J. and Billings, S.A. (1985) *Input-Output Parametric Models for Non-Linear Systems; Part 1: Deterministic Non-Linear Systems; Part 2: Stochastic Non-Linear Systems*. International Journal of Control. Vol 45, pp. 303-344.
- Ljung, L. (1987). *System Identification: Theory of User*. Prentice Hall. Englewood Cliffs, New Jersey.
- Nerrand, O.; Roussel-Ragot, P.; Personnaz, L.; Dreyfus, G. and Marcos S. (1993). *Neural Networks and Non-Linear Adaptive Filtering: Unifying Concepts and New Algorithms*. Neural Computation, Vol. 5, No. 2, pp. 165-199.
- Savran, A. (2007). *An Adaptive Recurrent Fuzzy System for Nonlinear Identification*. Applied Soft Computing, Vol. 7, No. 2.
- Urbani, D.; Roussel-Ragot, P.; Personnaz, L. and Dreyfus G. (1994). *The Selection of Neural Models of Non-Linear Dynamical Systems by Statistical Test*. Neural Networks for Signal Processing IV, IEEE Press, pp. 229-337.
- Wei, H.L., Billings, S.A. and Balikhin, M.A. (2006). *Wavelet Based Non-Parametric NARX Models for Nonlinear Input-Output System Identification*. International Journal of Systems Science, Vol 37, No. 15.
- Werbos, P.J. (1990). *Backpropagation Through Time: What It Does and How to Do It*. IEEE, Vol 78, No. 10.
- Willmott, C.J. (1981). *On the Validation of Models*. Physical Geography, Vol. 2, pp. 184-194.