APRENDIZAJE EN REDES NEURONALES PARA IDENTIFICACIÓN Y CONTROL DE SISTEMAS DINÁMICOS NO LINEALES

Pedro J. Zufiria
Grupo de Redes Neuronales
Dpto. de Matemática Aplicada a las Tecnologías de la Información
E.T.S. Ingenieros de Telecomunicación, Universidad Politécnica de Madrid
Ciudad Universitaria s/n, 28040 Madrid

Abstract. En este trabajo se modela el aprendizaje en redes neuronales artificiales para la identificación y control de sistemas dinámicos no lineales. En concreto, se analiza el entrenamiento supervisado, fundamentado en la minimización de una función de coste, mediante procedimientos de descenso en la dirección del gradiente, utilizando como referencia la formulación propuesta en [1]. Con dicho marco de referencia, se estudia la retropropagación dinámica, y se proponen alternativas analíticas y computacionales a la misma.

1 Introducción

Entre los numerosos paradigmas neuronales existentes –según su estructura y funcionalidad– el Perceptrón Multicapa (PMC) es el más destacado. El PMC puede aproximar con una precisión dada cualquier función continua $f \in C(\mathcal{A}, \mathbb{R}^m)$ definida en un conjunto compacto $\mathcal{A} \subset \mathbb{R}^n$ mediante una estructura y entrenamiento adecuados [2]. Adicionalmente, existe un algoritmo de entrenamiento llamado de *Retropropagación (RP)* que es computacionalmente eficiente. En [3, 4] se presenta una forma extendida del algoritmo de RP, denominada *Retropropagación Dinámica*, que se formula analíticamente en [6].

Tras unificar las diferentes formulaciones –encuadrando la retropropagación dinámica en la formulación de [1]– y comentar sus capacidades, en este trabajo se propone y ejemplifica unas modificaciones analíticas y computacionales de este procedimiento de entrenamiento.

La estructura del artículo es la siguiente. La Sección 2 presenta el contexto de adaptación paramétrica continuo en tiempo. La sección 3 aborda la formulación discreta conducente al algoritmo de retropropagación dinámica y propone algunas modificaciones de índole analítica y computacional al mismo. La sección 4 ejemplifica los diferentes métodos, y las conclusiones se presentan en la sección 5.

2 Formulación continua en tiempo

2.1 Aprendizaje de puntos fijos

El punto de partida es una formulación continua en el tiempo en la que vamos a caracterizar el ajuste paramétrico de un sistema para que éste presente un determinado punto fijo.

Consideramos el sistema dinámico –caracterizado como un problema de valor inicial–:

$$\frac{dx}{dt} = f(x, w, t), \quad x(t_0) = x_0 \tag{1}$$

donde $w \in \mathbb{R}^m$ y $x \in X = (C^1([t_0, t_1], \mathbb{R}^n), \|\cdot\|_2)$. Como es sabido, los puntos fijos del sistema satisfacen la ecuación: $f(x^*, w, t) = 0, \quad \forall t \geq t_0$.

Se desea que dicho sistema tenga a x^d como punto fijo. Para ello, se define la función de error:

$$E(x^*, x^d) = (x^* - x^d)^2$$

y la dependencia del error con respecto de w –aplicando la regla de la cadena–:

$$\frac{\partial E}{\partial w} = \left(\frac{\partial E}{\partial x^*}\right)^t \frac{\partial x^*}{\partial w}$$

Para simpificar la notación definimos:

$$\frac{\partial x^*}{\partial w} = P, \quad \frac{\partial f}{\partial x^*} = L$$

y basados en el teorema de la función implícita, deducimos la expresión de las sensibilidades:

$$LP + \frac{\partial f}{\partial w} = 0$$

De manera que, bajo la suposición de que L es invertible:

$$\frac{\partial E}{\partial w} = -\left(\frac{\partial E}{\partial x^*}\right)^t L^{-1}\frac{\partial f}{\partial w} = -\left(\frac{\partial f}{\partial w}\right)^t (L^t)^{-1}\frac{\partial E}{\partial x^*} = \left(\frac{\partial f}{\partial w}\right)^t v$$

El cálculo de v se puede efectuar de forma directa o mediante un procedimiento alternativo como la resolución de:

$$\frac{dv}{dt} = L^t v + \frac{\partial E}{\partial x^*}$$

en donde v es el punto fijo asintóticamente estable –supuesto que L satisface las correspondientes condiciones– de dicho sistema. Esta formulación permite engarzar de forma natural con el problema abordado a continuación.

2.2 Aprendizaje de trayectorias

Consideramos ahora el mismo sistema dinámico (1) y deseamos que evolucione según una trayectoria preestablecida $x^d(t) \in X$. Para ello, definimos la función de error $E : \mathbb{R}^m \to X \to \mathbb{R}$:

$$E = \int_{t_0}^{t_1} e(x, x^d, t)dt$$

cuya dependencia en w es –aplicando la regla de Leibnitz–:

$$\frac{\partial E}{\partial w} = \int_{t_0}^{t_1} \frac{\partial e}{\partial w}dt = \int_{t_0}^{t_1} \left(\frac{\partial e}{\partial x}\right)^t \frac{\partial x}{\partial w}dt \tag{2}$$

Para simplificar la terminología definimos:

$$\frac{\partial x}{\partial w} = P(t), \quad \frac{\partial f}{\partial x} = L$$

y, basándonos en la ecuación diferencial del sistema obtenemos las sensibilidades:

$$\frac{\partial}{\partial w}\frac{dx}{dt} = \frac{d}{dt}\frac{\partial x}{\partial w} = \frac{dP(t)}{dt} = \frac{\partial f(x,w)}{\partial w} = LP(t) + \frac{\partial f}{\partial w} \tag{3}$$

donde $\frac{\partial f(x,w)}{\partial w}$ es la derivada (total) y $\frac{\partial f}{\partial w}$ representa la derivada parcial explícita. Si $x(t_0)$ no depende de w, entonces $P(t_0) = 0$. La ecuación (3) define la *perturbación regular* de (1). Finalmente, la expresión de $\frac{\partial E}{\partial w}$ se calcula resolviendo (3) y substituyendo en (2). Este procedimiento puede resultar computacionalmente muy costoso, por lo que existen enfoques alternativos para el cómputo de $\frac{\partial E}{\partial w}$, como se verá a continuación.

2.3 Ecuación adjunta y métodos variacionales

En esta sección consideramos una vía alternativa a la propuesta en la sección anterior para calcular $\frac{\partial E}{\partial w}$, término fundamental en el algoritmo de aprendizaje supervisado. Para ello, formulamos el problema como uno de optimización restringida, y definimos el variacional:

$$\mathcal{J} = \int_{t_0}^{t_1} \left[e(x, x^d, t) + v^t \left(f(x, w, t) - \frac{dx}{dt} \right) \right] dt = \int_{t_0}^{t_1} \mathcal{E}(x, \frac{dx}{dt}, w, x^d, t) dt$$

donde v son los multiplicadores de Lagrange.
La ecuación de Euler[1] es:

$$\frac{\partial \mathcal{E}}{\partial x} - \frac{d}{dt} \frac{\partial \mathcal{E}}{\partial \frac{dx}{dt}} = 0$$

que puede ser reescrita como:

$$\frac{dv}{dt} = -\frac{\partial f}{\partial x} v - \frac{\partial e}{\partial x}$$

proporcionando el llamado sistema adjunto –donde los multiplicadores de Lagrange v evaluados en el mínimo correspondiente al problema se pueden interpretar como el error propagado–. Este resultado es deducible también empleando resultados de la teoría de control óptimo de sistemas dinámicos. La condición de contorno fijando t_1 conduce a $\frac{\partial \mathcal{E}}{\partial \frac{dx}{dt}}(t_1) = 0$, osea $v(t_1) = 0$. Por lo tanto, se debe resolver la ecuación diferencial hacia atrás en el tiempo.
Asimismo, la relación entre las variaciones es $\delta \mathcal{J} = \left[\int_{t_0}^{t_1} v^t (\frac{\partial f}{\partial w}) dt \right] \delta w$, y conduce a:

$$\frac{\partial E}{\partial w} = \int_{t_0}^{t_1} v^t \frac{\partial f}{\partial w} dt \tag{4}$$

que proporciona otra interpretación de la ecuación de sensibilidad del error con respecto de w.

3 Identificación no lineal para el caso discreto en tiempo

Todos los resultados previos se pueden reformular para el caso discreto en tiempo de forma simplificada. En particular, el contexto de trabajo en espacios de Banach se reduce al espacio Euclídeo.
Consideramos una caracterización e/s de una planta no lineal invariante en el tiempo y discreta:

$$x_p(k+1) = f(x_p(k), x_p(k-1), \ldots, x_p(k-n+1), u(k), u(k-1), \ldots u(k-m+1)) \tag{5}$$

donde $U \in \mathbb{R}^m$ e $x_p \in \mathbb{R}$ representan la *entrada* y *salida* del sistema, respectivamente.
El problema de la identificación de sistemas consiste en estimar los parámetros que caracterizan al modelo, de forma que cuando la planta y el modelo tienen la misma entrada, u, la salida del modelo $\hat{x}_p$ aproxima la de la planta x_p en el sentido de minimizar un error cuadrático medio (ver la Figura 1).

[1] Una interpretación de esta ecuación desde el punto de vista variacional se obtiene desde una formulación de este problema restringido basada en multiplicadores de Lagrange [5]. Nótese que se satisface dicha ecuación para todo valor de w –incluido w^*–. La dependencia de $\mathcal{E}$ en w aparece solamente en la expresión de la restricción.

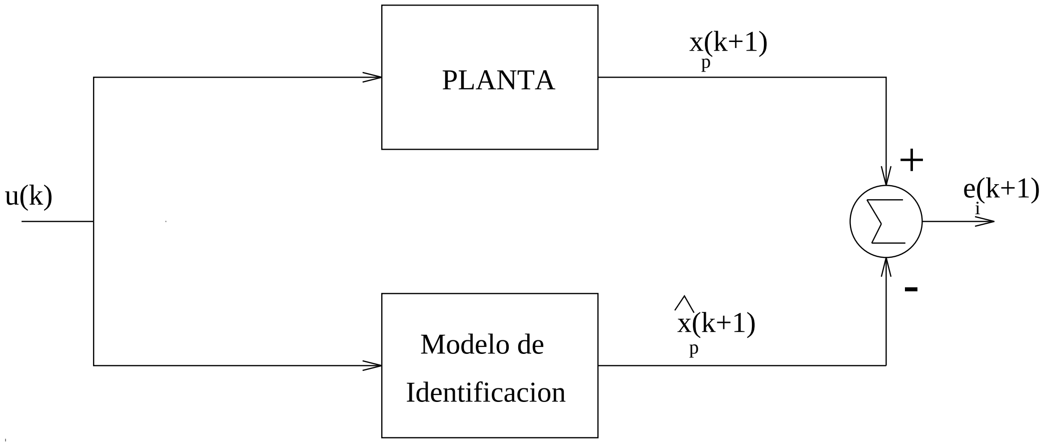

Figura 1: Planta y Modelo de Identificación.

Formalmente, el modelo empleado para realizar la identificación es:

$$x_p(k+1) = \mathcal{N}\left(x_p(k), x_p(k-1), \ldots, x_p(k-n+1), u(k), u(k-1), \ldots, u(k-m+1), \omega\right) \quad (6)$$

donde $\mathcal{N}$ es un PMC[2] con vector de pesos $\omega = (\omega_1, \ldots, \omega_{nw})$. La salida del modelo se denota por $\hat{x}_p$, cuando es interpretada como aproximación de la salida de la planta. La identificación del sistema en este caso consiste en estimar ω, según el criterio mencionado anteriormente.
El tipo de acoplo entre planta y modelo conduce a dos tipos diferentes de formulaciones denominadas *Modelo paralelo* y *Modelo serie-paralelo*, cuyos entrenamientos son drásticamente diferentes.
En el siguiente apartado se formula de forma sucinta el problema de identificación mediante redes neuronales en el contexto discreto.

3.1 Modelo paralelo. Aspectos analíticos de la retropropagación dinámica

Siguiendo a [6], la identificacion mediante el modelo paralelo es:

$$\hat{x}_p(k+1) = \mathcal{N}\left(\hat{x}_p(k), \hat{x}_p(k-1), \ldots, \hat{x}_p(k-n+1), u(k), u(k-1), \ldots, u(k-m+1), \omega\right) \quad (7)$$

Para su entrenamiento, se define una función de error en el intervalo $[k, \ldots, k+N-1]$ –conducente al ajuste de trayectoria según [1]–, donde el cálculo de las derivadas parciales precisa de los términos $\frac{\partial \hat{x}_p(k+j-i)}{\partial \omega_s}$ que pueden ser calculados mediante las ecuaciones de sensibilidad –obtenidas de (7)–.

$$\begin{aligned} E(\omega; k) &= \sum_{j=0}^{N-1} e^2(k+j) = \sum_{j=0}^{N-1} \left(x_p(k+j) - \hat{x}_p(k+j)\right)^2 \\ &= \sum_{j=0}^{N-1} \left(x_p(k+j) - \mathcal{N}(\hat{X}_p(k+j-1), U(k+j-1), \omega)\right)^2 \end{aligned} \quad (8)$$

[2]La elección de tal tipo de red se justifica debido a sus buenas capacidades de aproximación, pero el razonamiento que sigue es aplicable a otros tipos de redes.

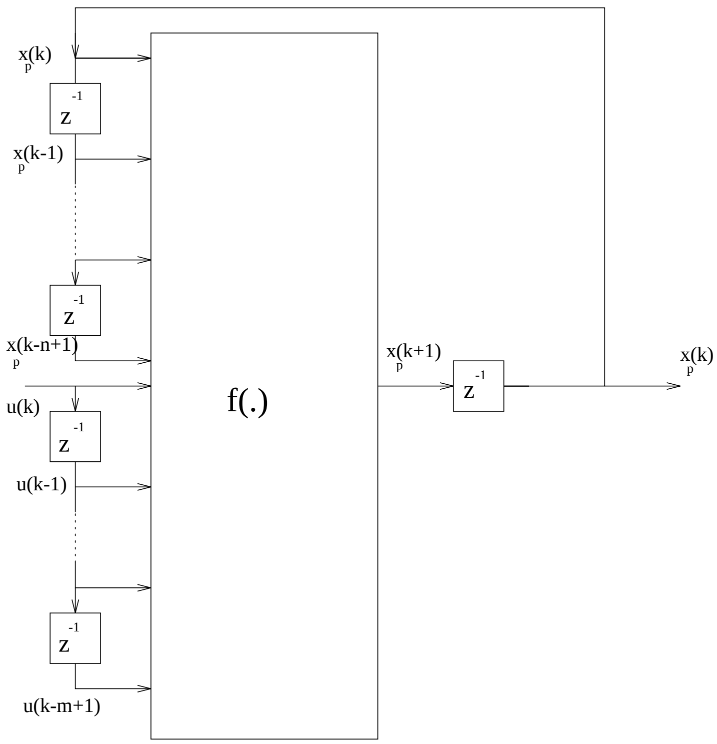

Figura 2: Caracterización de la Planta.

donde $\hat{X}_p(k+j-1) = [\hat{x}_p(k+j-1), \hat{x}_p(k+j-2), \ldots, \hat{x}_p(k+j-n)]$, $U(k+j-1) = [u(k+j-1), \ldots, u(k+j-m)]$, y $\omega = (\omega_1, \ldots, \omega_{nw})$ representa el vector de pesos de la red neuronal $\mathcal{N}$.
A continuación, se define una ley adaptativa de tipo descenso en la dirección del gradiente del tipo

$$\omega_{l+1} = \omega_l - \eta \frac{\partial E}{\partial \omega}|_{\omega_l} \tag{9}$$

para lo cual es preciso calcular las derivadas parciales:

$$\begin{aligned}
\frac{\partial E(\omega;k)}{\partial \omega_s} &= -2\sum_{j=0}^{N-1} e(k+j)\cdot \frac{\partial \hat{x}_p(k+j)}{\partial \omega_s} \\
&= -2\sum_{j=0}^{N-1} e(k+j)\cdot \left\{ \frac{\partial \mathcal{N}(\hat{X}_p(k+j-1), U(k+j-1), \omega)}{\partial \omega_s} + \right. \\
&+ \frac{\partial \mathcal{N}(\hat{X}_p(k+j-1), U(k+j-1), \omega)}{\partial \hat{x}_p(k+j-1)} \cdot \frac{\partial \hat{x}_p(k+j-1)}{\partial \omega_s} + \ldots \\
&+ \left. \frac{\partial \mathcal{N}(\hat{X}_p(k+j-1), \ldots, U(k+j-1), \omega)}{\partial \hat{x}_p(k+j-n)} \cdot \frac{\partial \hat{x}_p(k+j-n)}{\partial \omega_s} \right\}
\end{aligned} \tag{10}$$

donde $\frac{\partial \hat{x}_p(k+j-i)}{\partial \omega_s}$ se puede calcular de forma recurrente mediante las llamadas ecuaciones de sensibilidad –obtenidas de la aproximación neuronal de (7)–:

$$\begin{aligned}
\frac{\partial \hat{x}_p(k+1)}{\partial \omega_s} &= \frac{\partial \mathcal{N}(\hat{X}_p(k), U(k), \omega))}{\partial \omega_s} + \frac{\partial \mathcal{N}(\hat{X}_p(k), U(k), \omega)}{\partial \hat{x}_p(k)} \cdot \frac{\partial \hat{x}_p(k)}{\partial \omega_s} + \ldots \\
&+ \frac{\partial \mathcal{N}(\hat{X}_p(k), U(k), \omega)}{\partial \hat{x}_p(k-n+1)} \cdot \frac{\partial \hat{x}_p(k-n+1)}{\partial \omega_s}
\end{aligned} \tag{11}$$

con condiciones iniciales $\frac{\partial \hat{x}_p(0)}{\partial \omega_s} = \ldots = \frac{\partial \hat{x}_p(n-1)}{\partial \omega_s} = 0$.

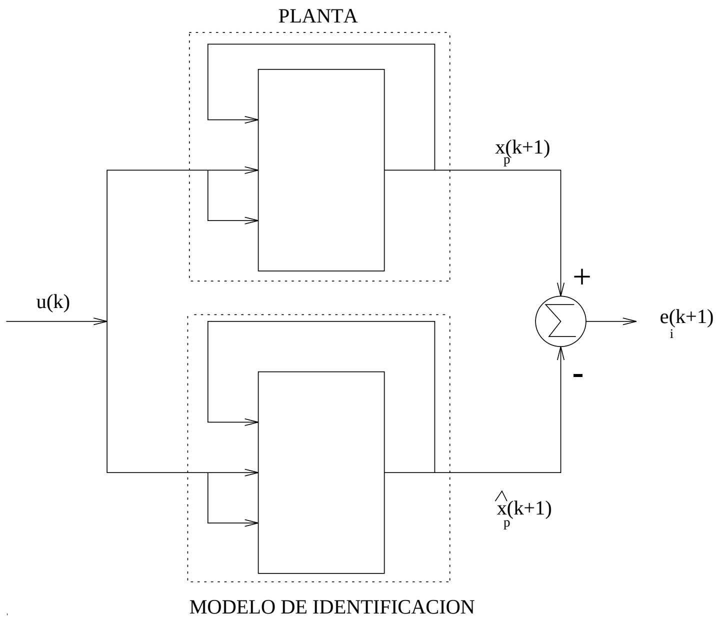

Figura 3: Modelo Paralelo.

En [3, 4] se propone una particularización de esta formulación mediante un algoritmo eficiente denominado de *retropropagación dinámica*, generalizando el clásico de RP para redes recurrentes multinivel. Dicho algoritmo calcula sensibilidades de forma muy eficiente y hay casos –cuando se satisfacen algunas condiciones de invertibilidad en los sistemas involucrados– en los que todas las computaciones se reducen al algoritmo estandar –estático– de retropropagación [3]. En general, no siempre se obtiene estabilidad en el modelo de identificación paralelo, aunque la planta a identificar sea estable. Por lo tanto, la convergencia del método de ajuste del PMC y la minimización del error de identificación tampoco están garantizadas.

3.1.1 Formulación variacional

Un procedimiento alternativo para el cálculo de $\frac{\partial E}{\partial \omega}$ se basa en el empleo del sistema adjunto según se ha visto en la formulación continua.
Para emplear resultados de teoría de control óptimo procedemos a redefinir el sistema en forma canónica, de manera que tenemos:

$$\hat{X}_p(k+1) = \begin{pmatrix} \hat{x}_p(k-n+2) \\ . \\ . \\ . \\ \hat{x}_p(k) \\ \mathcal{N}(\hat{X}_p(k), U_p(k), \omega) \end{pmatrix} = \mathcal{G}(\hat{X}_p(k), U_p(k), \omega) \qquad (12)$$

La ecuación adjunta viene definida por:

$$V(k+j-1) = \begin{pmatrix} 010\ldots & & \ldots 0 \\ & \ldots & \\ 000\ldots & & \ldots 1 \\ \multicolumn{3}{l}{\left(\frac{\partial \mathcal{N}(\hat{X}_p(k+j-1), U(k+j-1), \omega)}{\partial \hat{X}_p(k+j-1)}\right)^T} \end{pmatrix} V(k+j) + \frac{\partial e^2(k)}{\partial \hat{X}_p(k+j-1)}$$

con la condición $V(k+N) = 0$. Nótese que si se estudian las componentes de V por separado tenemos que se cumple la relación $V_i(k) = V_{i+1}(k+1), i = 1, \ldots, n-1$. Por otro lado, $V_n(k)$ depende de todas las componentes de $V(k+1)$. Por lo tanto, la evolución de $V_n(k) = v(k)$

caracteriza de forma completa temporalmente el multiplicador. Esto, nos permite escribir la siguiente ecuación de recurrencia:

$$v(k+j-1)=\sum_{l=0}^{n-1}\frac{\mathcal{N}(\hat{X}_p(k+j-1),U(k+j-1),\omega)}{\partial\hat{x}_p(k+j-1-l)}v(k+j+l)+\frac{\partial e^2(k+j-1)}{\partial\hat{x}_p(k+j-1-l)}$$

Por otro lado, se deduce que:

$$\frac{\partial e^2(k+j-1)}{\partial\omega}=-(\frac{\partial\mathcal{G}}{\partial\omega})^T V(k+j-1)$$

pudiendo ser reescrito para cada componente de ω como:

$$\frac{\partial e^2(k+j-1)}{\partial\omega_s}=-(0\ldots 0\frac{\partial\mathcal{N}}{\partial\omega_s})V(k+j-1)$$

Lo que conduce a:

$$\frac{\partial E}{\partial\omega_s}=-\sum_{j=0}^{N-1}\frac{\partial\mathcal{N}(\hat{X}_p(k+j-1),U(k+j-1),\omega)}{\partial\omega_s}^T v(k+j-1)$$

donde, una vez más, $v(k+j-1)=V_n(k+j-1)$. Así se obtiene una expresión necesaria para la ley de adaptación formalmente equivalente a (4).
Computacionalmente este enfoque es menos costoso al evitar el cálculo de las ecuaciones de sensibilidad para cada componente de w.

3.2 Modelo Serie-Paralelo

En el modelo serie-paralelo la salida verdadera de la planta se realimenta al modelo en lugar de su propia salida:

$$\hat{x}_p(k+1)=\mathcal{N}\left(x_p(k),x_p(k-1),\ldots,x_p(k-n+1),u(k),u(k-1),\ldots,u(k-m+1),\omega\right)\quad(13)$$

La función de error se calcula como:

$$\begin{aligned}E(\omega;k) &= \sum_{j=0}^{N-1}e^2(k+j)=\sum_{j=0}^{N-1}(x_p(k+j)-\hat{x}_p(k+j))^2\\ &= \sum_{j=0}^{N-1}(x_p(k+j)-\mathcal{N}(X_p(k+j-1),U(k+j-1),\omega))^2 \qquad (14)\end{aligned}$$

donde $X_p(k+j-1)=[x_p(k+j-1),x_p(k+j-2),\ldots,x_p(k+j-n)]$.

Calculando las derivadas parciales:

$$\begin{aligned}\frac{\partial E(\omega;k)}{\partial\omega_s} &= -2\sum_{j=0}^{N-1}e(k+j)\cdot\frac{\partial\hat{x}_p(k+j)}{\partial\omega_s}\\ &= -2\sum_{j=0}^{N-1}e(k+j)\cdot\left\{\frac{\partial\mathcal{N}(X_p(k+j-1),U(k+j-1),\omega)}{\partial\omega_s}+\right.\\ &+ \frac{\partial\mathcal{N}(X_p(k+j-1),U(k+j-1),\omega)}{\partial x_p(k+j-1)}\cdot\frac{\partial x_p(k+j-1)}{\partial\omega_s}+\ldots\\ &+ \left.\frac{\partial\mathcal{N}(X_p(k+j-1),U(k+j-1),\omega)}{\partial x_p(k+j-n)}\cdot\frac{\partial x_p(k+j-n)}{\partial\omega_s}\right\}\\ &= -2\sum_{j=0}^{N-1}e(k+j)\cdot\frac{\partial\mathcal{N}(X_p(k+j-1),U(k+j-1),\omega)}{\partial\omega_s} \qquad (15)\end{aligned}$$

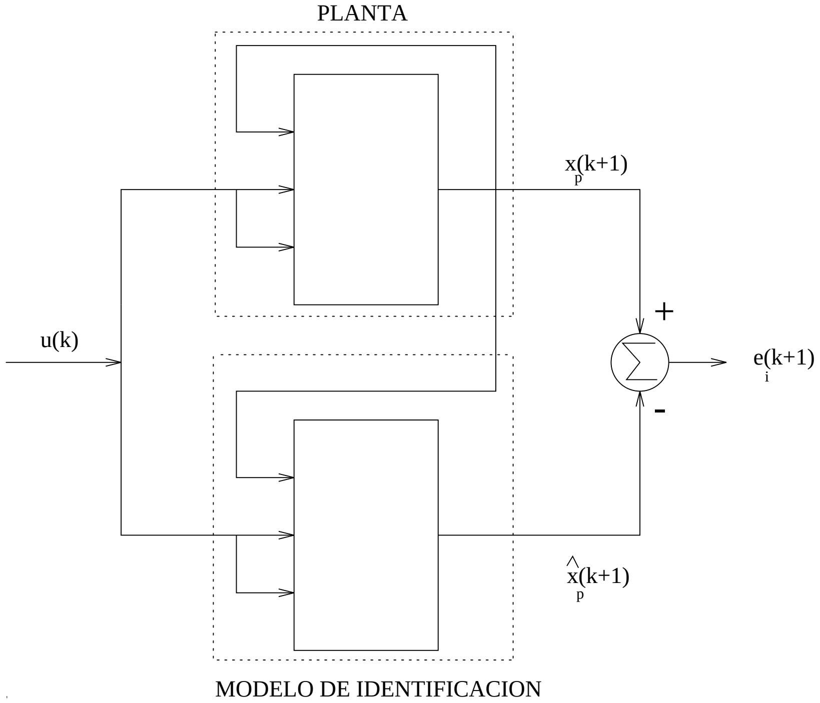

Figura 4: Modelo Serie-Paralelo.

debido a que $\frac{\partial x_p(k+j-i)}{\partial \omega_s} = 0, \; i = 1, \ldots, n$.
Como no hay lazo de realimentación, la identificación se realiza considerando el PMC como una red de flujo directo con el algoritmo usual de retropropagación para ajuste de pesos. Una vez que se ha entrenado el modelo, se puede sustituir por un modelo paralelo para la representación. En este caso, las derivadas parciales de la función de error se simplifican notablemente debido a que $\frac{\partial x_p(k+j-i)}{\partial \omega_s} = 0, \; i = 1, \ldots, n$. El aspecto más importante es que, como no hay lazo de realimentación en el modelo, se puede realizar la identificación considerando al PMC como red de flujo directo con el algoritmo de retropropagación usual para el ajuste de sus pesos. Esto supone una simplificación notable en el algoritmo de entrenamiento, y motiva la modificación propuesta a continuación.

3.3 Modificación de retropropagación dinámica

Partiendo de la función de error (8) y las derivadas parciales correspondientes en (10), se aproxima:

$$\frac{\partial E(\omega; k)}{\partial \omega_s} \approx -2 \sum_{j=0}^{N-1} e(k+j) \cdot \frac{\partial \mathcal{N}(\hat{X}_p(k+j-1), U(k+j-1), \omega)}{\partial \omega_s} \tag{16}$$

considerando que $\frac{\partial \hat{x}_p(k+j-i)}{\partial \omega_s} \approx 0, \; i = 1, \ldots n$. Esto equivale a considerar que los valores $\hat{x}_p(k+j-1)$ son fijos en el cómputo de las derivadas. Los valores de $\hat{x}_p(k+j-i), i = 1, \ldots n$, se deben actualizar de forma recurrente según (7), con condiciones iniciales $\hat{x}_p(0) = x(0), \ldots = \hat{x}_p(n-1) = x(n-1)$. Esta actualización de $\hat{x}_p(k+j-i), i = 1, \ldots n$, se puede realizar cada una o varias actualizaciones de ω, lo que permite definir dos escalas de tiempo para este algoritmo, con las consiguientes ventajas en el análisis y síntesis de su estabilidad. Algunos ejemplos ilustran propiedades de estabilidad superiores frente a los algoritmos ya existentes.

4 Ejemplo

Este ejemplo ilustra los tres modelos de identificación abordados en este trabajo. Consideramos la familia de sistemas dinámicos con condiciones iniciales:

$$x_p(k+1) = ax_p(k) + x_p(k-1), \quad x(0) = 1,\ x(1) = 1, \quad a \in \mathbb{R} \tag{17}$$

que deseamos identificar con el siguiente modelo de caracterización:

$$\hat{x}_p(k+1) = \omega \hat{x}_p(k) + \hat{x}_p(k-1) \tag{18}$$

Sea la función de coste –calculada considerando los instantes $k = 2$ y $k = 3$–:

$$E(\omega; 2) = \sum_{j=0}^{1} (x_p(2+j) - \hat{x}_p(2+j))^2 \tag{19}$$

Si se emplea el modelo serie-paralelo de identificación:

$$E(\omega; 2) = (1 + a - (\omega + 1))^2 + (1 + a(1+a) - (2\omega + 1))^2 \tag{20}$$

calculando la derivada con respecto de ω obtenemos la siguiente ley de actualización para ω (ecuación (9)):

$$\omega_{l+1} = \omega_l - \eta 2(1 + (1+a)^2(\omega_l - a) \tag{21}$$

que para $0 < \eta < \frac{1}{1+(1+a)^2}$ tiene $\omega^* = a$ como un punto fijo globalmente asintóticamente estable. Por lo tanto, el modelo serie-paralelo presenta unas propiedades de convergencia muy deseables.
Si se considera el modelo paralelo de identificación:

$$E(\omega; 2) = ((1+a) - (\omega + 1))^2 + (1 + (1+a) - ((\omega + 1)\omega + 1))^2 \tag{22}$$

calculando la derivada con respecto de ω la ecuación (9) toma la forma:

$$\omega_{l+1} = \omega_l - \eta(4\omega_l^3 + 6\omega_l^2 + 4(1 - a - a^2)\omega_l - 4a - 2a^2) \tag{23}$$

que para $0 < \eta < \frac{2}{8a^2+8a+4}$ tiene $\omega^* = a$ como un punto de equilibrio asintóticamente estable. Sin embargo, dicho sistema, para $a \notin A = (-\frac{1}{2} - \sqrt{2}, -\frac{1}{2} + \sqrt{2})$ –p.e., para $a = 1$– tiene dos puntos fijos adicionales ($\omega^* = -\frac{6+4a}{8} + \frac{\sqrt{4a^2+4a-7}}{4}$, $\omega^* = -\frac{6+4a}{8} - \frac{\sqrt{4a^2+4a-7}}{4}$), siendo el primero de ellos inestable para $0 < \eta$ pero el otro es asintóticamente estable para algunos valores de $0 < \eta$. Por lo tanto, el modelo paralelo con retropropagación dinámica puede converger a valores indeseables, siendo su eficiencia cuestionable.
Si consideramos el modelo propuesto en el apartado 3.3 –modificación de retropropagación dinámica–, tenemos:

$$E(\omega; 2) = (1 + a - (\omega + 1))^2 + (1 + a(1+a) - ((\varpi + 1)\omega + 1))^2 \tag{24}$$

donde ϖ debe considerarse como un valor fijo para el cálculo de la derivada con respecto de ω. Considerando que ϖ se actualiza cada paso de iteración –con el mismo valor de iteración que ω–, obtenemos la siguiente ley de actualización de ω:

$$\omega_{l+1} = \omega_l - \eta(2\omega_l^3 + 4\omega_l^2 + (4 - 2a - 2a^2)\omega_l - 4a - 2a^2) \tag{25}$$

que para $0 < \eta < \frac{2}{4a^2+6a+4}$ tiene $\omega^* = a$ como un punto fijo asintóticamente estable siempre que $a \in A' = (-2, 2)$, sin puntos fijos adicionales. Nótese que $A \subset A'$ por lo que las propiedades de convergencia en este caso son superiores a las de la retropropagación dinámica.

También es posible especular con las posibles escalas comparativas de tiempo entre la actualización de ω y ϖ. Por ejemplo, se puede estudiar el algoritmo que actualiza ω hasta su convergencia –con ϖ fijo– para actualizar ϖ a continuación, y repetir todo el proceso alternativamente. Las ecuaciones correspondientes son:

$$\omega_{l+1} = \omega_l - \eta 2(\omega_l(2 + 2\varpi + \varpi^2) - (2a + a^2 + a\varpi + a^2\varpi)) \tag{26}$$

que converge a $\omega^* = \frac{2a+a^2+a\varpi+a^2\varpi}{2+2\varpi+\varpi^2}$ siempre que $0 < \eta < \frac{1}{2+2\varpi+\varpi^2}$.
Consecuentemente, el sistema que define la evolución de ϖ es:

$$\varpi_{q+1} = \varpi_q - \eta' \frac{2a + a^2 + a\varpi_q + a^2\varpi_q}{2 + 2\varpi_q + \varpi_q^2} \tag{27}$$

que, curiosamente, tiene los mismos puntos fijos que (25).

5 Conclusiones

El aprendizaje o entrenamiento supervisado en redes neuronales artificiales para la identificación y control de sistemas dinámicos no lineales, ha sido modelado de forma analítica en el contexto de la minimización de funciones. Dicho entrenamiento está basado en procedimientos de descenso en la dirección del gradiente.
Un caso particular de dichas teçnicas de entrenamiento para sistemas discretos en tiempo es el algoritmo de retropropagación dinámica. En este trabajo se han propuesto alternativas analíticas y computacionales a dicho algoritmo; asimismo algunas de dichas alternativas han sido ilustradas mediante un ejemplo.

Agradecimientos
El autor agradece los valiosos comentarios de los miembros del *Grupo de Redes Neuronales* ($\mathcal{GRN}$) A. Vázquez y L. Rodríguez.
Este trabajo ha sido financiado parcialmente por el proyecto TAP94-0115 de la CICYT, y por la beca para doctores de IBERDROLA P94 0935 051.

Bibliografía

[1] P. Baldi. Gradient descent learning algorithm overview: A general dynamical systems perspective. *IEEE Trans. on Neur. Nets*, 6(1):182–195, January 1995.

[2] M. Stinchcombe K. Hornik and H. White. Multilayer feedforward networks are universal approximators. University of California, June 1988.

[3] K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Trans. on Neur. Nets*, 1(1):4–27, March 1990.

[4] K. S. Narendra and K. Parthasarathy. Gradient methods for the optimization of dynamical systems containing neural networks. *IEEE Trans. on Neur. Nets*, 2(2):252–262, March 1991.

[5] H. Sagan. *Introduction to the Calculus of Variations.* Dover, 1969.

[6] P. J. Zufiria. *Neural Networks for Chemical Engineers*, volume 6 of *Computer-Aided Chemical Engineering*, chapter An overview of dynamical system control using neural networks, pages 385–409. Elsevier, 1995.