

## Capítulo 3

### Estudio de la capacidad aproximativa de las redes de módulos neuronales

En este capítulo se caracteriza el conjunto de sistemas que se puede aproximar utilizando los módulos neuronales y los operadores de combinación entre estos. Se verá que el tipo de sistemas que se puede aproximar cubre una clase muy amplia. En concreto, cubre todos aquellos sistemas cuya dinámica es esencialmente lineal, pero contiene no-linealidades en la entrada y/o salida, además de combinaciones más complejas entre este tipo de sistemas.

Una red neuronal modular se construye de forma similar a los diagramas de bloques clásicos en la descripción de sistemas dinámicos. Aquí, un módulo neuronal corresponde a un bloque y las combinaciones que se pueden realizar son las conexiones en serie y paralelo. La caracterización de los sistemas que se pueden aproximar con redes modulares se formaliza mediante los elementos con que éstas se construyen, que son los módulos neuronales, y las operaciones que se emplean para hacerlo, que son las conexiones en serie y paralelo.

La formalización de las conexiones en serie y paralelo corresponde a las operaciones de composición y suma de funciones, mientras que los módulos neuronales se han reducido a un conjunto elemental formado por sistemas lineales y el módulo UMBRALNM. Con estos elementos se pueden generar redes neuronales que aproximan, arbitrariamente bien, sistemas lineales y funciones no lineales, continuas y derivables. A partir de estos elementos y operaciones, observando las propiedades de los conjuntos que generan, se llega a caracterizar el tipo de sistemas que se puede aproximar con las redes de módulos neuronales.

En este capítulo se estudia el error máximo de aproximación que una de estas redes modulares presenta al comparar su comportamiento al del sistema que aproxima. Se establecen cotas del

error expresadas en función de los errores de aproximación de los módulos que componen la red.

Se concluye que, dado un sistema no lineal expresable como suma y composición de funciones no lineales, continuas y derivables, y sistemas dinámicos lineales, existe una aproximación neuronal cuya respuesta se aproxima a la del sistema con un grado arbitrario de precisión.

### 3.1 Formalización de los elementos básicos y los operadores

Se entiende por *elementos básicos* el conjunto de sistemas dinámicos más elemental, de entre los módulos neuronales, que permite generar un conjunto significativo de sistemas más complejos. En concreto, el conjunto de los elementos básicos es  $\Sigma_b = \{\sigma, \Sigma_L\}$ .

En las siguientes secciones se utiliza este conjunto de elementos básicos por su capacidad generadora y más adelante se valora la introducción de otros elementos del conjunto de sistemas definidos por los módulos neuronales del capítulo 2.

Este conjunto contiene funciones estáticas y dinámicas. Las primeras serán consideradas como un caso particular de las dinámicas para mantener la unicidad del tratamiento. Las definiciones de ambos elementos son:

- $\sigma(x(k))$ , la función sigmoideal. Corresponde a la función *umbral* $_{\alpha}$  del apartado 2.3.1. En realidad, esta función puede definirse de diferentes formas sin alterar los resultados que siguen. Las condiciones que sobre ella se imponen varían según los autores, pero en este contexto basta que cumpla las características de ser monótona creciente y acotada por ambos extremos.<sup>1</sup>
- $\Sigma_L$ , el conjunto formado por los sistemas lineales en tiempo discreto, de varias entradas y una salida y sin restricciones de orden. Está caracterizado por las matrices constantes  $A$ ,  $B$ ,  $C$  y  $D$  (matriz de estados, de entradas, de salidas y directa, respectivamente). Las ecuaciones dinámicas de  $\Sigma_L$  son

---

<sup>1</sup> En realidad, la definición de *umbral* $_{\alpha}(x)$  y la definición usual de  $\sigma(x)$  son diferentes porque la primera no realiza una suma ponderada de  $x$ . Sin embargo, éste no es más que un detalle técnico que no afecta a los resultados que siguen, pues es sencillo emular la suma ponderada a partir de la operación producto externo, definida más adelante.

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k)$$

$$y(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}u(k)$$

(si no se especifica lo contrario,  $\mathbf{x}$  es el vector de estado en el instante  $k$ , y  $u$  e  $y$  son las entradas y salidas en el instante  $k$ , respectivamente)

En particular, son elementos del conjunto  $\Sigma_b$ :

- *Función identidad, Id.* Se define a través de un elemento de  $\Sigma_L$ , con  $A = B = C = 0$  y  $D = 1$ . Su ecuación dinámica es  $y(k) = u(k)$ .
- *Función producto por constante, K.* Se define a través de un elemento de  $\Sigma_L$ , con  $A = B = C = 0$  y  $D = K$ . Su ecuación dinámica es  $y(k) = K \cdot u(k)$ .

Una red modular está formada mediante combinaciones en serie y paralelo de módulos neuronales. La combinación en serie es equivalente a la operación de composición de funciones y la combinación en paralelo es equivalente a la operación de suma de funciones. Por tanto, las operaciones permitidas sobre  $\Sigma_b$  son la suma y la composición, que se definen como es usual:

sean  $s_1, s_2 \in \Sigma_b$

- Suma:  $(s_1 + s_2)(x) = s_1(x) + s_2(x)$
- Composición:  $(s_1 \circ s_2)(x) = s_1(s_2(x))$

En este apartado también se considera la operación de producto externo y el operador retardo. La primera se define a partir de la composición de funciones y de la función producto por constante.

- Producto externo:  $(\lambda \cdot s_1)(x) = \lambda \cdot s_1(x)$

El operador retardo se define a partir de la composición de funciones y de un elemento de  $\Sigma_L$ , como se indica a continuación

- *Operador retardo.* Sea  $s \in \Sigma_b$  y  $n \in \mathbb{N}$ , se define el retardo  $s(x(k-n))$  como  $(s_n \circ s)$ , siendo  $s_n$  un sistema con

$$A = \begin{bmatrix} 0 & \cdots & 0 & 0 \\ 1 & & 0 & 0 \\ \vdots & \ddots & 0 & 0 \\ 0 & \cdots & 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad C = [0 \quad \cdots \quad 1] \quad D = 0$$

$A$  es una matriz de  $n \times n$ ,  $B$  y  $C$  vectores de  $n \times 1$  y  $1 \times n$ , respectivamente y la ecuación dinámica de  $s$  es  $y(k) = u(k - n)$ .

Los operadores de suma y composición y la operación de producto externo los agrupamos en dos conjuntos,  $\Theta = \{+, \circ\}$  y  $\Xi = \{+, \cdot\}$ . Finalmente, es necesario recordar que el *operador producto*, habitual entre otros tipos de funciones, no es definible sobre  $\Sigma_b$  porque no se utiliza esta combinación entre módulos neuronales en este trabajo.

### 3.2 Conjunto de elementos que generan $\Sigma_b$ y $\Theta$

En esta sección se caracterizan los elementos del conjunto generado por  $\Sigma_b$  bajo los operadores  $\Theta$ , al que denotaremos por  $\Sigma_{NM}$ . El conjunto generado por  $\Sigma_b$  bajo los operadores  $\Xi$  queda incluido dentro de  $\Sigma_{NM}$  porque, como ya se ha indicado anteriormente, la operación producto externo se realiza con elementos de  $\Sigma_b$  y el operador de composición. De esta manera, estudiando  $\Sigma_{NM}$  se estudian ambos conjuntos.

#### Definición 1:

$$\Sigma_{NM} = \{\Sigma_b, \Theta\}$$

Veamos qué propiedades algebraicas se pueden aplicar sobre  $\Sigma_{NM}$ .

#### 3.2.1 Álgebra de las redes de módulos neuronales

El conjunto de sistemas  $\Sigma_{NM}$ , generado a partir de  $\Sigma_b$  y  $\Theta$ , satisface:

- Elemento neutro respecto a la suma:  $0$
- Elemento simétrico respecto a la suma:  $s + s' = 0$ , con  $s' = -1 \cdot s$ .
- Elemento neutro respecto a la composición:  $Id$
- Se cumplen las propiedades asociativa y conmutativa respecto a la suma.

$$s_1 + (s_2 + s_3) = (s_1 + s_2) + s_3 \qquad s_1 + s_2 = s_2 + s_1$$

- Se cumple la propiedad asociativa respecto a la composición:

$$s_1 \circ (s_2 \circ s_3) = (s_1 \circ s_2) \circ s_3$$

- No se cumple la propiedad distributiva de la composición respecto a la suma:

$$s_1 \circ (s_2 + s_3) \neq s_1 \circ s_2 + s_1 \circ s_3$$

Con todo ello:

- $\Sigma_{NM}$  con la suma cumple características de grupo abeliano.
- $\Sigma_{NM}$  con la composición cumple características de grupo (no conmutativo).
- $\Sigma_{NM}$  con la suma y la composición no es un cuerpo. Para ello sería necesaria la existencia del elemento inverso que, en general, no se da en  $\Sigma_{NM}$ .

En este trabajo se tratan sistemas dinámicos causales. Por esta razón, no se ha considerado oportuno incluir en  $\Sigma_b$  una definición más genérica de sistema dinámico que permitiese a  $\Sigma_{NM}$  reunir características de cuerpo con los operadores de suma y composición.

### 3.2.2 Elementos de $\Sigma_{NM}$

En este apartado se caracterizan los elementos del conjunto  $\Sigma_{NM}$ . Este conjunto contiene las redes modulares que se pueden construir, combinando en serie y en paralelo, módulos neuronales.

Para ello, partiendo de los elementos básicos de  $\Sigma_b$ , se construye el conjunto  $NLN$ . Los componentes de  $NLN$  son aproximaciones de sistemas lineales con no-linealidades estáticas en sus entradas y salidas. Se verá que este conjunto es cerrado respecto a la suma y que, componiendo sus elementos, se obtienen aproximaciones de sistemas más complejos, a los que denominaremos  $NLN^2$ ,  $NLN^3$ , etc. La caracterización de los elementos del conjunto  $\Sigma_{NM}$  se realiza a través de la observación de las propiedades de  $NLN^i$  respecto a las operaciones de suma y composición.

La notación que se utiliza para los conjuntos  $NLN$  indica que sus elementos están compuestos por una función no lineal (entrada), un sistema lineal y otra función no lineal (salida). Esta notación es similar a la que se utiliza en los métodos de *modelado de redes con estructura de bloque*, como se verá en el apartado 3.2.3, para remarcar el paralelismo entre ambos tipos de modelos.

#### **Definición 2:**

Sea una función  $f \in C^r(\Omega)$ , donde  $\Omega$  es un compacto de  $\mathbb{R}^n$ .

Definimos  $\sigma$ -aproximación de  $f$  a la función  $f_\varepsilon = \sum_{j=1..N} \beta_j \sigma(\alpha_j x - \theta_j)$  tal que  $\|f - f_\varepsilon\| < \varepsilon$ .

$f_\varepsilon$  es la operación realizada por una red neuronal de tres capas (la última lineal). Los parámetros  $\alpha_j$  y  $\beta_j$  corresponden a los pesos de la red neuronal y  $\theta_j$  corresponde al sesgo (*bias*) de las neuronas de la capa intermedia. Se demuestra en [Hor-91] que  $f_\varepsilon$  puede aproximar arbitrariamente bien cualquier función continua de  $\mathbb{R}^n$  sobre  $\mathbb{R}$ , en el espacio  $L^p$ , bajo diferentes normas, en concreto, bajo la norma uniforme. Es decir, para una función  $f \in C^r(\Omega)$ , con soporte compacto en  $\mathbb{R}^n$ , existe  $f_\varepsilon$  que aproxima a  $f$  con error menor que una constante arbitraria  $\varepsilon$ .

Con ello, cualquier función de clase  $C^r(\Omega)$ , puede ser aproximada con elementos de  $\Sigma_b$ , de manera que  $\Sigma_{NM}$  contiene aproximaciones arbitrariamente precisas de todas ellas.

**Definición 3:**

Sean  $f_i, g, h$  (con  $i = 1..m$ )  $\sigma$ -aproximaciones de  $(m + 2)$  funciones continuas y con soporte compacto en  $\mathbb{R}^n$ ,

Sean  $A$  y  $C$  matrices constantes de  $n \times n$  y  $n \times l$  respectivamente,  $B = [b_1, \dots, b_m]$  vector constante y  $F = [f_1, \dots, f_m]^T$ .

Definimos *NLN* al conjunto de los sistemas que cumplen:

$$\begin{aligned} \mathbf{x}(k+1) &= A\mathbf{x} + BF(u) \\ y &= g(C\mathbf{x}) + h(u) \end{aligned} \tag{3.1}$$

que se pueden describir como:  $s = g \circ s_L \circ f + h$ , t.q.  $s_L \in \Sigma_L$ .

Este conjunto es interesante porque contiene los elementos dinámicos no lineales más sencillos del conjunto  $\Sigma_{NM}$ . Las combinaciones entre estos elementos generan el resto de  $\Sigma_{NM}$ . Veamos, pues, la caracterización de  $\Sigma_{NM}$  a partir de *NLN*.

**Proposición 1:**

$NLN \subset \Sigma_{NM}$ .

*Demostración:*

Sea  $s \in NLN$ , entonces existen  $F, g$  y  $h$  tales que:

$F = [f_1 \ \dots \ f_m]^T$  y  $f_i, g, h \in \Sigma_{NM}$  de forma que

$$\mathbf{x}(k+1) = A\mathbf{x} + BF(u)$$

$$y = g(C\mathbf{x}) + h(u)$$

representa a  $s$  en ecuaciones de espacio de estado.

Definimos un sistema lineal  $s_L \in \Sigma_{NM}$  tal como:

$$\mathbf{x}(k+1) = A\mathbf{x} + Bu$$

$$y = C\mathbf{x}$$

entonces  $s = g \circ s_L \circ f + h$ . Por tanto  $s \in \Sigma_{NM}$  y, en consecuencia,  $NLN \subset \Sigma_{NM}$ .

■

**Definición 4:**

$$NLN^i = NLN \circ \overset{(i)}{\dots} \circ NLN$$

**Proposición 2:**

Un sistema  $s \in NLN^n$  tiene la siguiente representación en ecuaciones de estado:

$$\mathbf{x}(k+1) = \begin{bmatrix} A_1 & 0 & \dots & 0 \\ 0 & A_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & A_n \end{bmatrix} \mathbf{x} + \begin{bmatrix} B_1 & B_2 & \dots & B_n \end{bmatrix} \begin{bmatrix} F_1(u) \\ F_2(\varphi_1(\mathbf{x}_1, u)) \\ \vdots \\ F_n(\varphi_{n-1}(\mathbf{x}_1, \dots, \mathbf{x}_{n-1}, u)) \end{bmatrix} \quad (3.2)$$

$$y = \varphi_n(\mathbf{x}_1, \dots, \mathbf{x}_n, u)$$

donde  $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]^T$  y cada componente,  $\mathbf{x}_i$ , es un vector que contiene variables de estado. La función  $\varphi_i(\mathbf{x}_1, \dots, \mathbf{x}_i, u)$  se define recursivamente según

$$\begin{aligned}
\varphi_1(\mathbf{x}_1, u) &= g_1(C_1 \mathbf{x}_1) + h_1(u) \\
\varphi_2(\mathbf{x}_1, \mathbf{x}_2, u) &= g_2(C_2 \mathbf{x}_2) + h_2(\varphi_1(\mathbf{x}_1, u)) \\
&\vdots \\
\varphi_n(\mathbf{x}_1, \dots, \mathbf{x}_n, u) &= g_n(C_n \mathbf{x}_n) + h_n(\varphi_{n-1}(\mathbf{x}_1, \dots, \mathbf{x}_{n-1}, u))
\end{aligned}$$

*Demostración:*

Sea el sistema  $s_i \in NLN$  descrito por

$$\begin{aligned}
\mathbf{x}_i(k+1) &= A_i \mathbf{x}_i + B_i F_i(u) \\
y_i &= g_i(C_i \mathbf{x}_i) + h_i(u)
\end{aligned}$$

Según la definición 4,  $s = s_n \circ s_{n-1} \circ \dots \circ s_1$ . Así que por simple sustitución  $s$  resulta

$$\begin{aligned}
\begin{bmatrix} \mathbf{x}_1(k+1) \\ \mathbf{x}_2(k+1) \\ \vdots \\ \mathbf{x}_n(k+1) \end{bmatrix} &= \begin{bmatrix} A_1 \mathbf{x}_1 \\ A_2 \mathbf{x}_2 \\ \vdots \\ A_n \mathbf{x}_n \end{bmatrix} + \begin{bmatrix} B_1 F_1(u) \\ B_2 F_2(g_1(C_1 \mathbf{x}_1) + h_1(u)) \\ \vdots \\ B_n F_n(g_{n-1}(C_{n-1} \mathbf{x}_{n-1}) + h_{n-1}(g_{n-2}(C_{n-2} \mathbf{x}_{n-2}) + \dots + h_1(u)) \dots) \end{bmatrix} \\
y &= g_n(C_n \mathbf{x}_n) + h_n(g_{n-1}(C_{n-1} \mathbf{x}_{n-1}) + \dots + h_1(u)) \dots
\end{aligned}$$

de manera que identificando términos se llega a la descripción propuesta. ■

**Lema 1:**

$$NLN^i \subset NLN^{i+1}$$

*Demostración:*

Por la definición 4, para dos elementos  $s_1 \in NLN^i$  y  $s_2 \in NLN$ ,  $s_1 \circ s_2 \in NLN^{i+1}$ . Puesto que el conjunto  $NLN$  tiene elemento neutro respecto a la composición, siempre es posible tomar  $s_2 = Id$  de forma que  $s_1 \circ s_2 \in NLN^i$ . Por tanto,  $NLN^i \subset NLN^{i+1}$ . ■



**Lema 2:**

$$NLN^{i+1} \not\subset NLN^i$$

*Demostración:*

Por el lema 1, si  $NLN^{i+1} \subset NLN^i$  entonces  $NLN^{i+1} = NLN^i$ , de manera que  $NLN = NLN^2 = \dots = NLN^n$ . Sin embargo, comparando las ecuaciones (3.1) y (3.2) que corresponden a un elemento genérico de  $NLN$  y de  $NLN^i$ , respectivamente, queda claro que no son equivalentes. En concreto, si nos fijamos en la parte no lineal de las ecuaciones de estado vemos que, en un elemento de  $NLN^i$ , la función de entrada,  $F$ , depende de la entrada y del estado  $\mathbf{x}$ , mientras que en  $NLN$  únicamente depende de la entrada. Por tanto,  $NLN^{i+1} \not\subset NLN^i$ .

■

Los resultados obtenidos hasta el momento se recapitulan seguidamente. Se han descrito los elementos generadores de  $\Sigma_{NM}$ , que son los sistemas discretos lineales de varias entradas y una salida (sin restricciones de orden) y aproximaciones arbitrariamente precisas de funciones estáticas no lineales con soporte compacto en  $\mathbb{R}^n$ . Se ha comprobado que la composición reiterada entre estos elementos da lugar a conjuntos más amplios, que se pueden organizar jerárquicamente (ver figura 3.1). También se ha caracterizado la forma concreta que tienen las ecuaciones de un elemento de  $NLN^i$ .

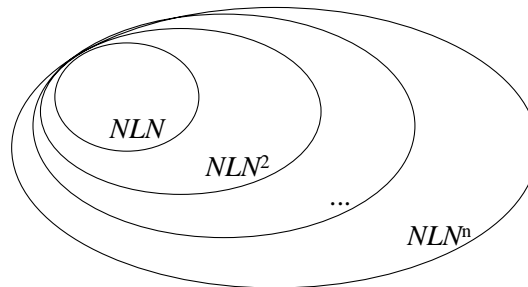


Figura 3.1: diagrama de las relaciones entre conjuntos  $NLN^i$

En lo que resta de sección se propone y demuestra que los elementos generados a partir de los conjuntos iniciales, los sistemas lineales y las  $\sigma$ -aproximaciones, con las operaciones suma y composición se encuentran en la unión de todos los conjuntos  $NLN^i$ .

**Lema 3:**

$NLN^n$  es cerrado respecto a la suma.

*Demostración:*

Sea  $s_1 \in NLN^n$  un sistema definido por

$$\mathbf{x}(k+1) = \begin{bmatrix} A_{11} & 0 & \cdots & 0 \\ 0 & A_{12} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & A_{1n} \end{bmatrix} \mathbf{x} + \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1n} \end{bmatrix} \begin{bmatrix} F_{11}(\varphi_0(u)) \\ F_{12}(\varphi_1(\mathbf{x}_1, u)) \\ \vdots \\ F_{1n}(\varphi_{n-1}(\mathbf{x}_1, \dots, \mathbf{x}_{n-1}, u)) \end{bmatrix}$$

$$y_1 = \varphi_n(\mathbf{x}_1, \dots, \mathbf{x}_n, u)$$

$$\varphi_l(\mathbf{x}_1, \dots, \mathbf{x}_l, u) = g_{1l}(C_{1l}\mathbf{x}_l) + h_{1l}(\varphi_{l-1}(\mathbf{x}_1, \dots, \mathbf{x}_{l-1}, u))$$

$$\varphi_0(u) = u$$

Sea  $s_2$  otro sistema del conjunto  $NLN^n$  definido por

$$\mathbf{z}(k+1) = \begin{bmatrix} A_{21} & 0 & \cdots & 0 \\ 0 & A_{22} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & A_{2n} \end{bmatrix} \mathbf{z} + \begin{bmatrix} B_{21} & B_{22} & \cdots & B_{2n} \end{bmatrix} \begin{bmatrix} F_{21}(\phi_0(u)) \\ F_{22}(\phi_1(\mathbf{z}_1, u)) \\ \vdots \\ F_{2n}(\phi_{n-1}(\mathbf{z}_1, \dots, \mathbf{z}_{n-1}, u)) \end{bmatrix}$$

$$y_2 = \phi_n(\mathbf{z}_1, \dots, \mathbf{z}_n, u)$$

$$\phi_l(\mathbf{z}_1, \dots, \mathbf{z}_l, u) = g_{2l}(C_{2l}\mathbf{z}_l) + h_{2l}(\phi_{l-1}(\mathbf{z}_1, \dots, \mathbf{z}_{l-1}, u))$$

$$\phi_0(u) = u$$

en estas definiciones los vectores  $\mathbf{x}$  y  $\mathbf{z}$  se pueden descomponer, respectivamente, en  $[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]^T$  y  $[\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n]^T$ , siendo las componentes  $\mathbf{x}_i$  y  $\mathbf{z}_i$  vectores que contienen variables de estado.

Las matrices  $A_{1i}$ ,  $A_{2i}$  y los vectores de  $B_{1i}$ ,  $B_{2i}$ ,  $C_{1i}$ ,  $C_{2i}$  ( $i = 1..n$ ) son constantes.

Tanto  $F_{1i}$  como  $F_{2i}$  son vectores de funciones y tienen las mismas dimensiones que  $B_{1i}$  y  $B_{2i}$ .

La suma de  $s_1$  y  $s_2$  es

$$\begin{aligned} y &= y_1 + y_2 = \varphi_n(\mathbf{x}_1, \dots, \mathbf{x}_n, u) + \phi_n(\mathbf{z}_1, \dots, \mathbf{z}_n, u) = \\ &= g_{1n}(C_{1n}\mathbf{x}_n) + h_{1n}(\varphi_{n-1}(\mathbf{x}_1, \dots, \mathbf{x}_{n-1}, u)) + g_{2n}(C_{2n}\mathbf{z}_n) + h_{2n}(\phi_{n-1}(\mathbf{z}_1, \dots, \mathbf{z}_{n-1}, u)) \end{aligned}$$

definimos  $\zeta_i = [\mathbf{x}_i, \mathbf{z}_i]$  de forma que  $\zeta = [\zeta_1, \zeta_2, \dots, \zeta_n]^T = [\mathbf{x}_1, \mathbf{z}_1, \mathbf{x}_2, \mathbf{z}_2, \dots, \mathbf{x}_n, \mathbf{z}_n]^T$ .

Esta definición permite expresar  $s$  como

$$\zeta(k+1) = \begin{bmatrix} A_1 & 0 & \dots & 0 \\ 0 & A_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & A_n \end{bmatrix} \zeta + \begin{bmatrix} B_1 & B_2 & \dots & B_n \end{bmatrix} \begin{bmatrix} F_1(\psi_0(u)) \\ F_2(\psi_1(\zeta_1, u)) \\ \vdots \\ F_n(\psi_{n-1}(\zeta_1, \dots, \zeta_{n-1}, u)) \end{bmatrix} \quad (3.3)$$

$$y = \psi_n(\zeta_1, \dots, \zeta_n, u)$$

$$\psi_l(\zeta_1, \dots, \zeta_l, u) = g_l(C_l \zeta_l) + h_l(\psi_{l-1}(\zeta_1, \dots, \zeta_{l-1}, u))$$

$$\psi_0(u) = u$$

En esta descripción de  $s$  las matrices  $A_i$  y  $C_i$ , los vectores  $B_i$  y  $F_i$  y las funciones  $g_i$  y  $h_i$  se definen de la siguiente manera:

$$A_i = \begin{bmatrix} A_{1i} & 0 \\ 0 & A_{2i} \end{bmatrix} \quad F_i(\psi_i(\zeta_1, \dots, \zeta_i, u)) = \begin{bmatrix} F_{1i}(\psi_i(\mathbf{x}_1, 0, \dots, \mathbf{x}_i, 0, u)) \\ F_{2i}(\psi_i(0, \mathbf{z}_1, \dots, 0, \mathbf{z}_i, u)) \end{bmatrix}$$

$$B_i = [B_{1i} \ B_{2i}] \quad C_i = \begin{bmatrix} C_{1i} & 0 \\ 0 & C_{2i} \end{bmatrix} \quad g_i(C_i \zeta_i) = g_{1i}(C_{1i} \mathbf{x}_i) + g_{2i}(C_{2i} \mathbf{z}_i)$$

$$h_i(\psi_i(\zeta_1, \dots, \zeta_i, u)) = h_{1i}(\psi_i(\mathbf{x}_1, 0, \dots, \mathbf{x}_i, 0, u)) + h_{2i}(\psi_i(0, \mathbf{z}_1, \dots, 0, \mathbf{z}_i, u))$$

Puesto que el sistema de la ecuación 3.3, según la proposición 2, tiene la forma de un elemento de  $NLN^n$ ,  $s \in NLN^n$  y, por lo tanto, la suma es cerrada para  $NLN^n$ .

■

### Proposición 3:

$$\Sigma_b \subset NLN \subset NLN^2 \subset \dots \subset NLN^i \subset \dots \subset \Sigma_{NM}$$

*Demostración:*

Se demuestra por inducción sobre  $i$ .

Por la proposición 1,  $NLN \subset \Sigma_{NM}$ .

Puesto que  $NLN^i = NLN \circ \overset{(i)}{\dots} \circ NLN$  (definición 4), entonces  $NLN^{i+1} = NLN^i \circ NLN$ .

Suponiendo, por hipótesis de inducción, que  $NLN^i \subset \Sigma_{NM}$ , la composición de dos elementos de  $\Sigma_{NM}$  es otro elemento del mismo conjunto. Por tanto,

$$NLN^{i+1} = NLN^i \circ NLN \subset \Sigma_{NM}$$

con lo cual  $\Sigma_b \subset NLN \subset NLN^2 \subset \dots \subset NLN^i \subset \dots \subset \Sigma_{NM}$ .

■

**Proposición 4:**

$$\Sigma_{NM} = \bigcup_{i=1}^{\infty} NLN^i$$

*Demostración:*

Se demuestra probando las dos siguientes implicaciones:

$$\textcircled{1} s \in \bigcup_{i=1}^{\infty} NLN^i \Rightarrow s \in \Sigma_{NM} \text{ es obvio a partir de la proposición 3.}$$

$$\textcircled{2} s \in \Sigma_{NM} \Rightarrow s \in \bigcup_{i=1}^{\infty} NLN^i$$

Se demuestra por inducción sobre el número de operaciones (sumas y composiciones) necesarias para crear  $s$  a partir de elementos de  $\Sigma_b$ . En esta demostración se utiliza una notación especial para indicar el número de operaciones aplicadas para crear un elemento. Así, un

sistema,  $s$ , que se ha creado a partir de la aplicación de  $n$  operaciones se denota por  $\overset{n}{s}$ .

Si  $s$  se genera con una operación  $\overbrace{s}^1 = s_1 + s_2$  o  $\overbrace{s}^1 = s_1 \circ s_2$ ,  $s_1, s_2 \in \Sigma_b$ . Es sencillo comprobar, a partir de la definición 3, que  $s \in NLN$ .

Si  $s$  se genera con  $n + 1$  operaciones (indistintamente sumas o composiciones) es posible expresar  $s$  como  $s = s_1 + s_2$  o  $s = s_1 \circ s_2$ . En el primer caso cualquiera de las siguientes opciones es posible

$$\begin{aligned} \overbrace{s}^{n+1} &= \overbrace{s_1}^0 + \overbrace{s_2}^n \\ \overbrace{s}^{n+1} &= \overbrace{s_1}^1 + \overbrace{s_2}^{n-1} \\ &\vdots \\ \overbrace{s}^{n+1} &= \overbrace{s_1}^n + \overbrace{s_2}^0 \end{aligned}$$

Para todas ellas existen enteros  $i$  y  $j$  de manera que  $s_1 \in NLN^i$  y  $s_2 \in NLN^j$ , por hipótesis de inducción. Puesto que la cualquiera de estos conjuntos es cerrado respecto a la suma, lema 3,  $s \in NLN^k$ ,  $k = \max(i, j)$ .

En el segundo caso, de forma análoga al anterior, las siguientes opciones son válidas

$$\begin{aligned} \overbrace{s}^{n+1} &= \overbrace{s_1}^0 \circ \overbrace{s_2}^n \\ \overbrace{s}^{n+1} &= \overbrace{s_1}^1 \circ \overbrace{s_2}^{n-1} \\ &\vdots \\ \overbrace{s}^{n+1} &= \overbrace{s_1}^n \circ \overbrace{s_2}^0 \end{aligned}$$

De nuevo, para todas ellas existen enteros  $i$  y  $j$  de manera que  $s_1 \in NLN^i$  y  $s_2 \in NLN^j$ , por hipótesis de inducción. La composición entre elementos de estos conjuntos está en  $NLN^{i+j}$ , según la definición 4. Por lo tanto, la implicación ② también es cierta y como conclusión

$$\Sigma_{NM} = \bigcup_{i=1}^{\infty} NLN^i$$

■

### 3.2.3 Discusión sobre el conjunto de sistemas $\Sigma_{NM}$

El conjunto  $\Sigma_{NM}$  representa un amplio espectro de sistemas no lineales. Observando los sistemas de  $NLN$ , se puede concluir que un elemento de  $\Sigma_{NM}$  está formado por subsistemas lineales en su dinámica y no lineales en las entradas y salidas. La evolución de las variables de estado en estos subsistemas, en ausencia de entradas, es esencialmente lineal. Sin embargo, el mapa entre entradas y salidas es claramente no lineal.

Un subconjunto de  $\Sigma_{NM}$  ha sido ya estudiado y caracterizado ampliamente (ver [Chen-95] y [Hab-90] para referencias). Se trata de los llamados modelos con estructura de bloque (*block-structured network models*). Estos modelos están formados por combinaciones de operadores  $L$  y  $N$ , definidos según:

$$L(x(t)) = y(t) = \int_0^{\infty} h(\tau)x(t-\tau)d\tau$$

$$N(x(t)) = y(t) = \sum_{n=0}^l a_n x^n(t)$$

donde  $x$  e  $y$  son las entradas y salidas de los operadores;  $l$  ( $2 \leq l < \infty$ ) es el orden del polinomio que caracteriza la no-linealidad y  $a_n$  son sus coeficientes;  $h(t)$  es la respuesta impulsional de la parte lineal. Esta definición es de tiempo continuo, pero su contrapartida discreta es directa.

Las combinaciones más conocidas en serie y paralelo de estos operadores son el modelo de Wiener (combinación  $LN$ ), el de Hammerstein (combinación  $NL$ ), el de Wiener-Hammerstein (combinación  $LNL$ ) y otras estructuras formadas a partir de conexiones en paralelo de diferentes tipos. Estos modelos se estudian desde los años 60 como estructuras válidas para modelar sistemas biológicos o circuitos electrónicos, entre otros tipos de aplicaciones.

El conjunto  $\Sigma_{NM}$ , por otro lado, todavía es más amplio porque admite modelos con conexiones en serie de los operadores formando modelos equiparables a  $LNLN$ ... De esta forma  $\Sigma_{NM}$  permite modelar sistemas compuestos por uno o más subsistemas  $LNL$ .

### 3.3 Capacidad aproximativa de los elementos de $\Sigma_{NM}$

En los apartados precedentes de este capítulo se han caracterizado los sistemas dinámicos que se pueden aproximar con redes modulares de módulos neuronales. En esta sección se estudia el error de aproximación. Es decir, se establece una cota a la norma de la diferencia entre la salida de un sistema no lineal y su aproximación neuronal en  $\Sigma_{NM}$ .

El problema, planteado formalmente en el siguiente apartado, consiste en estudiar la diferencia entre la respuesta de un sistema no lineal y su aproximación neuronal, en términos de elementos del conjunto  $\Sigma_{NM}$ .

### 3.3.1 Planteamiento del problema

Sean  $f_{i,j}$ ,  $g_i$  y  $h_i$  ( $i = 1, \dots, n$ ,  $j = 1, \dots, m_i$ ) funciones reales de clase  $C^r$  con soporte compacto en  $\mathbb{R}^n$  y  $\sigma f_{i,j}$ ,  $\sigma g_i$  y  $\sigma h_i$  sus  $\sigma$ -aproximaciones.

Sea  $A_i$  ( $i = 1, \dots, n$ ) una matriz constante de  $m_i \times m_i$ ,  $B_i = [b_1, \dots, b_{m_i}]$  y  $C_i = [c_1, \dots, c_{m_i}]^T$  vectores constantes y  $F_i = [f_{i,1}, \dots, f_{i,m_i}]^T$ .

Sea  $\sigma_s \in \Sigma_{NM}$  una red de módulos neuronales que aproxima a un sistema  $s$ .  $\sigma_s$  se define como

$$\sigma \mathbf{x}(k+1) = \begin{bmatrix} A_1 & 0 & \dots & 0 \\ 0 & A_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & A_n \end{bmatrix} \sigma \mathbf{x} + \begin{bmatrix} B_1 & B_2 & \dots & B_n \end{bmatrix} \begin{bmatrix} \sigma F_1(\sigma \varphi_0(\sigma \mathbf{x}, u)) \\ \sigma F_2(\sigma \varphi_1(\sigma \mathbf{x}, u)) \\ \vdots \\ \sigma F_n(\sigma \varphi_{n-1}(\sigma \mathbf{x}, u)) \end{bmatrix}$$

$$\sigma y = \sigma \varphi_n(\sigma \mathbf{x}, u)$$

$$\sigma \varphi_l(\sigma \mathbf{x}, u) = \sigma g_l(C_l \sigma \mathbf{x}_l) + \sigma h_l(\sigma \varphi_{l-1}(\sigma \mathbf{x}, u))$$

$$\sigma \varphi_0(\sigma \mathbf{x}, u) = u$$

donde  $\sigma \mathbf{x} = [\sigma \mathbf{x}_1, \sigma \mathbf{x}_2, \dots, \sigma \mathbf{x}_n]^T$  y cada componente,  $\sigma \mathbf{x}_i$ , es un vector que contiene variables de estado. El sistema  $s$  se define análogamente

$$\mathbf{x}(k+1) = \begin{bmatrix} A_1 & 0 & \dots & 0 \\ 0 & A_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & A_n \end{bmatrix} \mathbf{x} + \begin{bmatrix} B_1 & B_2 & \dots & B_n \end{bmatrix} \begin{bmatrix} F_1(\varphi_0(\mathbf{x}, u)) \\ F_2(\varphi_1(\mathbf{x}, u)) \\ \vdots \\ F_n(\varphi_{n-1}(\mathbf{x}, u)) \end{bmatrix}$$

$$y = \varphi_n(\mathbf{x}, u)$$

$$\varphi_l(\mathbf{x}, u) = g_l(C_l \mathbf{x}_l) + h_l(\varphi_{l-1}(\mathbf{x}, u))$$

$$\varphi_0(\mathbf{x}, u) = u$$

donde  $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]^T$  y las componentes  $\mathbf{x}_i$  son, paralelamente a la definición de  $\sigma \mathbf{x}_i$ , vectores que contienen variables de estado.

El objetivo de este apartado es calcular el valor de  $|\sigma y - y|$ , teniendo en cuenta que el estado inicial de ambos sistemas es  $\mathbf{x}_0$ .

**Observación 1:**

Recordando la definición 1, la norma de la diferencia entre una función real de clase  $C^r$  con soporte compacto en  $\mathbb{R}^n$  y su  $\sigma$ -aproximación es menor que una constante arbitraria  $\varepsilon$ , por tanto,

$$\|\sigma f_{i,j} - f_{i,j}\| < \varepsilon, \quad \|\sigma g_i - g_i\| < \varepsilon, \quad \|\sigma h_i - h_i\| < \varepsilon$$

**Lema 4:**

$\|\sigma \mathbf{x}(k) - \mathbf{x}(k)\| < \varepsilon_x$ , donde  $\varepsilon_x$  es una constante calculable a partir de  $\varepsilon$ , la cota entre una función real de clase  $C^r$ , con soporte compacto en  $\mathbb{R}^n$ , y su  $\sigma$ -aproximación.

*Demostración:*

En adelante, se denotarán  $A$ ,  $B$ ,  $C$  y  $F$  a las correspondientes matrices y vectores con componentes  $A_i$ ,  $B_i$ ,  $C_i$  y  $F_i$ .

La solución de la ecuación en diferencias para el sistema  $s$  (la de  $\sigma s$  es análoga) es

$$\mathbf{x}(k) = A^{k-k_0} \mathbf{x}_0 + \sum_{l=k_0}^{k-1} A^{k-l-1} BF(u(l))$$

por tanto

$$\|\sigma \mathbf{x}(k) - \mathbf{x}(k)\| = \left\| A^{k-k_0} (\sigma \mathbf{x}_0 - \mathbf{x}_0) + \sum_{l=k_0}^{k-1} A^{k-l-1} B(\sigma F(u(l)) - F(u(l))) \right\|$$

Puesto que el estado inicial de  $s$  y  $\sigma s$  es el mismo y dado que  $\sigma F$  es una  $\sigma$ -aproximación de  $F$  se puede acotar la diferencia inicial como sigue

$$\|\sigma \mathbf{x}(k) - \mathbf{x}(k)\| < \left\| \left( \sum_{l=k_0}^{k-1} A^{k-l-1} \right) B \varepsilon \right\| = \left\| (A^0 + A^1 + \dots + A^{k-k_0-1}) B \varepsilon \right\| \quad (3.4)$$



Si los valores propios de  $A$  (en valor absoluto) son menores que 1 la serie  $A^0 + A^1 + \dots + A^{k-k_0-1}$  converge en  $(Id - A)^{-1}$ , con  $k \rightarrow \infty$ , de manera que

$$\|\sigma \mathbf{x}(k) - \mathbf{x}(k)\| < \|(Id - A)^{-1} B \varepsilon\| = \varepsilon_x$$

Si algún valor propio de  $A$  (en valor absoluto) es mayor que 1 el término  $k$ -ésimo de la serie  $A^0 + A^1 + \dots + A^{k-k_0-1}$  converge en  $(Id - A)^{-1}(Id - A^{k+1})$  para valores finitos de  $k$ , así,

$$\|\sigma \mathbf{x}(k) - \mathbf{x}(k)\| < \|(Id - A)^{-1}(Id - A^{k+1})B \varepsilon\| = \varepsilon_x$$

■

En el caso de que la matriz de estado,  $A$ , que define la transición de estados, tenga valores propios de valor absoluto estrictamente menor que 1, la constante  $\varepsilon_x$  es independiente del tiempo. En el caso de que  $A$  tenga algún valor propio de valor absoluto mayor que 1, la constante  $\varepsilon_x$  depende del tiempo y, por lo tanto, sólo es posible acotar la norma para intervalos finitos de tiempo. El caso en que  $A$  tiene algún valor propio exactamente igual a 1 se comenta en el apartado 3.3.3.

### 3.3.2 Cálculo de una cota para $|\sigma y - y|$

#### Proposición 3:

$|\sigma y - y| < \varepsilon_y$ , donde  $\varepsilon_y$  es una constante calculable a partir de  $\varepsilon$ , como en el lema 4.

*Demostración:*

En el planteamiento del problema se dice que  $\sigma s \in \Sigma_{NM}$ , por tanto,  $\sigma s \in NLN^i$ . Se demuestra por inducción sobre  $i$ .

Para  $i = 1$ ,  $\sigma s \in NLN$ ,  $|\sigma y - y| < \varepsilon_1$

$$\begin{aligned} |\sigma y - y| &= |\sigma \varphi_1(\sigma \mathbf{x}, u) - \varphi_1(\mathbf{x}, u)| = |\sigma g_1(C_1 \sigma \mathbf{x}) + \sigma h_1(u) - g_1(C_1 \mathbf{x}) - h_1(u)| \leq \\ &\leq |\sigma g_1(C_1 \sigma \mathbf{x}) - g_1(C_1 \mathbf{x})| + |\sigma h_1(u) - h_1(u)| < |\sigma g_1(C_1 \sigma \mathbf{x}) - g_1(C_1 \mathbf{x})| + \varepsilon \end{aligned} \quad (3.5)$$

El primer término de este resultado es acotable

$$\begin{aligned} & \left| \sigma g_1(C_1 \sigma \mathbf{x}) - g_1(C_1 \mathbf{x}) \right| = \left| \sigma g_1(C_1 \sigma \mathbf{x}) - g_1(C_1 \sigma \mathbf{x}) + g_1(C_1 \sigma \mathbf{x}) - g_1(C_1 \mathbf{x}) \right| \leq \\ & \leq \left| \sigma g_1(C_1 \sigma \mathbf{x}) - g_1(C_1 \sigma \mathbf{x}) \right| + \left| g_1(C_1 \sigma \mathbf{x}) - g_1(C_1 \mathbf{x}) \right| < \varepsilon + \left| g_1(C_1 \sigma \mathbf{x}) - g_1(C_1 \mathbf{x}) \right| \end{aligned} \quad (3.6)$$

puesto que  $g_1$  es derivable se puede aplicar el Teorema de valor medio, de forma que

$$\left| g_1(C_1 \sigma \mathbf{x}) - g_1(C_1 \mathbf{x}) \right| \leq \left\| \nabla g_1 \Big|_{\bar{\mathbf{x}}} \right\| \cdot \left\| C_1 \sigma \mathbf{x} - C_1 \mathbf{x} \right\| = \left\| \nabla g_1 \Big|_{\bar{\mathbf{x}}} \right\| \cdot \left\| C_1 (\sigma \mathbf{x} - \mathbf{x}) \right\| \quad (3.7)$$

El valor  $\bar{\mathbf{x}}$  es un punto intermedio entre  $\mathbf{x}$  y  $\sigma \mathbf{x}$ , como presupone el citado teorema. Así, aplicando el lema 4 y recogiendo las cotas de las ecuaciones (3.5), (3.6) y (3.7) resulta que

$$\left| \sigma y - y \right| < \varepsilon + \varepsilon + \left| \nabla g_1 \Big|_{\bar{\mathbf{x}}} \right\| \left\| C_1 \varepsilon_x \right\| \Rightarrow \left| \sigma y - y \right| < 2\varepsilon + \varepsilon_g = \varepsilon_1$$

Para  $\sigma s \in NLN^{i-1}$ , si  $\left| \sigma y - y \right| < \varepsilon_{i-1}$  entonces  $\left| \sigma y - y \right| < \varepsilon_i$  con  $\sigma s \in NLN^i$

$$\begin{aligned} \left| \sigma y - y \right| &= \left| \sigma \varphi_i(\sigma \mathbf{x}, u) - \varphi_i(\mathbf{x}, u) \right| = \\ &= \left| \sigma g_i(C_i \sigma \mathbf{x}_i) + \sigma h_i(\sigma \varphi_{i-1}(\sigma \mathbf{x}, u)) - g_i(C_i \mathbf{x}_i) - h_i(\varphi_{i-1}(\mathbf{x}, u)) \right| \leq \\ &\leq \left| \sigma g_i(C_i \sigma \mathbf{x}_i) - g_i(C_i \mathbf{x}_i) \right| + \left| \sigma h_i(\sigma \varphi_{i-1}(\sigma \mathbf{x}, u)) - h_i(\varphi_{i-1}(\mathbf{x}, u)) \right| \end{aligned} \quad (3.8)$$

El primer término de este resultado, aplicando de nuevo el teorema del valor medio, es acotable y el segundo queda

$$\begin{aligned} & \left| \sigma h_i(\sigma \varphi_{i-1}(\sigma \mathbf{x}, u)) - h_i(\varphi_{i-1}(\mathbf{x}, u)) \right| = \\ &= \left| \sigma h_i(\sigma \varphi_{i-1}(\sigma \mathbf{x}, u)) - h_i(\sigma \varphi_{i-1}(\sigma \mathbf{x}, u)) + h_i(\sigma \varphi_{i-1}(\sigma \mathbf{x}, u)) - h_i(\varphi_{i-1}(\mathbf{x}, u)) \right| \leq \\ &\leq \left| \sigma h_i(\sigma \varphi_{i-1}(\sigma \mathbf{x}, u)) - h_i(\sigma \varphi_{i-1}(\sigma \mathbf{x}, u)) \right| + \left| h_i(\sigma \varphi_{i-1}(\sigma \mathbf{x}, u)) - h_i(\varphi_{i-1}(\mathbf{x}, u)) \right| < \\ &< \varepsilon + \left| \nabla h_i \Big|_{\bar{\varphi}} \right\| \cdot \left| \sigma \varphi_{i-1}(\sigma \mathbf{x}, u) - \varphi_{i-1}(\sigma \mathbf{x}, u) \right| \end{aligned} \quad (3.9)$$

Por la hipótesis de inducción  $\left| \sigma \varphi_{i-1}(\sigma \mathbf{x}, u) - \varphi_{i-1}(\sigma \mathbf{x}, u) \right|$  está acotado, por lo tanto, recogiendo de nuevo las cotas de las ecuaciones (3.8) y (3.9) el resultado que se persigue es

$$\left| \sigma y - y \right| < \varepsilon + \left| \nabla g_i \Big|_{\bar{\mathbf{x}}} \right\| \left\| C_i \varepsilon_x \right\| + \varepsilon + \varepsilon_\varphi \Rightarrow \left| \sigma y - y \right| < 2\varepsilon + \varepsilon_g + \varepsilon_\varphi = \varepsilon_i$$

Por todo ello  $|\sigma y - y| < \varepsilon_y = 2\varepsilon + \varepsilon_g + \varepsilon_\varphi$

■

### 3.3.3 *Discusión de los resultados*

Los resultados de esta sección demuestran que, dado un sistema no lineal que pueda expresarse como suma y composición de funciones estáticas no lineales y sistemas dinámicos lineales, existe una aproximación neuronal cuya respuesta se aproxima a la del sistema con un grado arbitrario de precisión. La aproximación neuronal es una red modular construida a partir de combinaciones en serie y paralelo (suma y composición) de módulos neuronales.

Este resultado es especialmente interesante por el hecho de que dos sistemas con la misma estructura y parámetros parecidos en los elementos que la componen, pero no idénticos, pueden tener respuestas completamente diferentes.

Las cotas que se establecen en esta sección son absolutas, válidas para períodos de tiempo finitos e infinitos cuando la dinámica del sistema que se aproxima es estable. En el caso de dinámicas estrictamente inestables (algún valor propio de la matriz de estados mayor que 1) las cotas son válidas para períodos de tiempo finitos. En sistemas marginalmente estables (algún valor propio igual a 1) es razonable que no se puedan establecer resultados concretos. Para comprender esto basta con tomar un sistema lineal de segundo orden marginalmente estable. Cualquier aproximación de éste, con valores propios de la matriz de estados diferentes de 1, será irremediamente estable o inestable. Es decir, los comportamientos cualitativos de las respuestas de ambos serán diferentes, por precisa que sea la aproximación.

## 3.4 *Discusión sobre las $\sigma$ -aproximaciones y los módulos neuronales*

En esta sección se estudia la utilidad de considerar el conjunto de módulos neuronales definidos en el capítulo 2 dentro del entorno de modelado de sistemas no lineales.

Hasta el momento se ha examinado el conjunto de sistemas que, solamente con dos de los módulos neuronales, se pueden generar aplicando los operadores de suma y composición. Sin embargo, en el capítulo 2 se describen 8 módulos neuronales más que podrían haber entrado a formar parte de  $\Sigma_b$  inicialmente. Se considera, seguidamente, la utilidad de incluir estos módulos en  $\Sigma_b$ .

### 3.4.1 Módulos estáticos

Las funciones (o más correctamente, familias de funciones) estáticas y no lineales aproximadas por módulos neuronales en este estudio son: SATURACIÓN, ZONA MUERTA, VALOR ABSOLUTO Y FRICCIÓN. Las tres primeras son funciones continuas no derivables y la última es discontinua.

La inclusión de los módulos neuronales que aproximan estas funciones en el conjunto  $\Sigma_b$  no es posible a priori porque no está demostrado que cumplan las características de ser  $\sigma$ -aproximaciones de ellas, con lo que dejarían de ser ciertos algunos de los resultados de este capítulo. Sin embargo, es fácil comprobar que la función que realiza cualquiera de ellos, por su construcción a partir de sigmoides, es localmente integrable en el sentido de Riemann. En este caso, Hornik (en [Hor-93]) demuestra que es posible construir  $\sigma$ -aproximaciones a partir de ellos.

Es evidente que la utilidad de los módulos neuronales no está en sustituir a las neuronas sencillas en la aproximación de funciones en general. Esto comportaría un aumento del coste computacional y ninguna ventaja adicional. Sin embargo, su inclusión en el conjunto  $\Sigma_b$  no altera los resultados presentados en este capítulo y aporta la posibilidad de aproximar ciertas familias de funciones con gran sencillez.

Es decir, que para las funciones de tipo saturación, zona muerta, valor absoluto y fricción existe una aproximación (el módulo neuronal) con menor cantidad de parámetros ajustables y con una tendencia natural a generalizar correctamente. Estas características permiten reducir el número de ejemplos e iteraciones necesarias para realizar el aprendizaje y, además, interpretar físicamente los parámetros obtenidos respecto a la función (o familia de funciones) que se aproxima.

Por estas razones es interesante utilizar los módulos neuronales correspondientes a las funciones no lineales estáticas citadas anteriormente.

### 3.4.2 Módulos dinámicos

Las funciones dinámicas no lineales (o familias de funciones, de nuevo) aproximadas por módulos neuronales en este estudio son: JUEGO, TOPE, LIMITADOR e HISTÉRESIS.

Al igual que sucede con los módulos neuronales estáticos, no está demostrado que las aproximaciones que realizan cumplan las características de ser  $\sigma$ -aproximaciones. Por esta razón no ha sido posible incluirlos en el estudio teórico. Es más, un sencillo análisis de

cualquiera de ellos permite comprobar que no forman parte de  $\Sigma_{NM}$ . Tomemos, por ejemplo, el módulo neuronal RELENM. La función que realiza se puede definir según

$$\tilde{f}(u(k)) = \text{sig}_s\left(w_s \left(\tilde{f}(u(k-1)) + \text{umbral}_{\alpha_1}(u(k)) + \text{umbral}_{\alpha_2}(u(k))\right)\right)$$

Esta función es dinámica, como ya se ha apuntado. Se puede, por tanto, escribir en forma de ecuaciones de estado. Para ello se notará a la suma de las funciones *umbral* por *h*, a la función sigmoideal simétrica (*sig*) por *g* y al estado del sistema por *x(k)*, quedando

$$\begin{aligned} x(k+1) &= g(w_s x(k) + w_s h(u(k))) \\ y(k) &= g(w_s x(k) + w_s h(u(k))) \end{aligned} \quad (3.10)$$

Al comparar las ecuaciones (3.10) y (3.2) (representación en el espacio de estados de un sistema  $s \in NLN^i$ ) se observa claramente que son incompatibles porque la función no lineal *g* se aplica sobre el estado del sistema y eso no es posible en los elementos de *NLN*. Por tanto, no disponemos de una base formal para poder asegurar que las capacidades aproximativas de los cuatro módulos dinámicos (JUEGONM, TOPENM, LIMITADORNM e HISTÉRESISNM) tienen cotas determinables.

Sin embargo, se han realizado estudios experimentales que confirman su utilidad en tareas de modelado e identificación. En efecto, se ha podido constatar, [Mor-96b], que, al igual que los módulos estáticos, los módulos que aproximan las funciones JUEGO, LIMITADOR e HISTÉRESIS superan en más de un orden de magnitud la velocidad de aprendizaje de las redes neuronales convencionales, generalizan con menor error y permiten la interpretación de sus parámetros.

<i>Problema</i>	<i>Tipo de red</i>	<i>aprendizaje OK</i>	<i>media</i>	<i>std</i>
Histéresis	NM	45/45/43	276.3	113.9
Histéresis	1-3-1	21/13/1	12947.1	994.7
Juego	NM	45/45/44/45	9.3	2.0
Juego	1-6-1	45/16/45/0	577.9	109.1
Limitador de velocidad	NM	45/45	3.4	0.8
Limitador de velocidad	1-8-10-1	24/0	645.1	650

Tabla 3.1: resumen de resultados del aprendizaje en módulos neuronales dinámicos

En [Mor-96b] se compara la velocidad de aprendizaje de los módulos neuronales frente a la de las redes convencionales. Éstas tienen una estructura de propagación directa de tres o cuatro capas, con retardos en las entradas y en las salidas cuando se modelan comportamientos dinámicos. Las pruebas consisten en el aprendizaje de una determinada función, como el juego de engranajes, por ejemplo, hasta su consecución con un error menor que un valor fijado de antemano.

En la tabla 3.1 se detallan los resúmenes de estos resultados. En la primera columna está el nombre de la función que se aproxima. La segunda columna describe el tipo de red, indicando con números la cantidad de neuronas de cada capa en las redes convencionales. La tercera columna muestra el número de redes que alcanzaron el valor prefijado de error mínimo para grupos de 45 ejecuciones que partían de pesos iniciales diferentes. Cada uno de estos grupos corresponde a una instancia concreta de la función a aproximar. Finalmente, la cuarta y quinta columna contienen las medias y las desviaciones estándar de las iteraciones necesarias para completar el aprendizaje, respectivamente.

Los resultados de la tabla 3.1 muestran que estos módulos neuronales son capaces de aprender con velocidades muy superiores a las redes convencionales (nótese que estas redes son las que tienen mejor comportamiento de entre un amplio conjunto probado).

Además, también se ha podido comprobar su buen funcionamiento cuando se hayan integrados en redes modulares, [Mor-97]. En esta presentación se compara la velocidad de aprendizaje de las redes de módulos neuronales y la de las redes convencionales. De nuevo, las últimas son redes de propagación directa, de varias capas y retardos en las entradas y salidas para permitir el comportamiento dinámico.

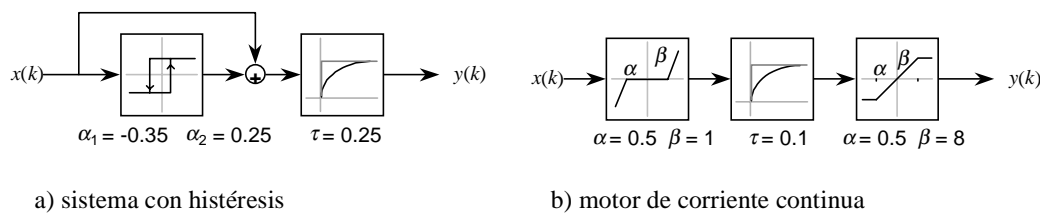


Fig. 3.2: diagramas de bloques de los sistemas de prueba en [Mor-97]

En este caso, las redes modulares incorporan la información estructural que se dispone del modelo, por lo que, inicialmente ya están compuestas por los módulos correctos, conectados adecuadamente. El aprendizaje tiene la finalidad de sintonizar los parámetros de cada módulo cuando están conectados formando una estructura más compleja.

En la figura 3.2 se detallan los sistemas que se utilizaron para realizar las pruebas. Se trata de un sistema simulado con histéresis y de un modelo simplificado de un motor de corriente continua. Ambos están formados por un sistema de primer orden y diversas no-linealidades.

Los resultados de las pruebas, que tienen características similares a las descritas anteriormente, se presentan en la tabla 3.2. De ellos se puede concluir que, en los casos comprobados, las redes de módulos neuronales aprenden más rápidamente, y para un mayor conjunto de pesos iniciales, que las redes convencionales.

<i>Problema</i>	<i>Tipo de red</i>	<i>Aprendizaje OK</i>	<i>Media</i>	<i>std</i>
Sistema con histéresis	NM	45	126.3	49.8
Sistema con histéresis	1-7-5-1	34	5663.5	1870.3
Motor de corriente continua	NM	45	980.3	256.6
Motor de corriente continua	1-15-1	41	3536.7	467.7

*Tabla 3.2: resumen de resultados del aprendizaje en redes de módulos neuronales*

Por todo ello, en aquellos casos en que se disponga de información previa que permita suponer la existencia de efectos no lineales dinámicos en el modelo del sistema, se incluirán los correspondientes módulos neuronales en el mecanismo de modelado, que se explica en el siguiente capítulo.