UNIVERSIDAD PONTIFICIA COMILLAS
MADRID

Escuela Técnica Superior de Ingenieros Industriales (ICAI)
Departamento de Electrotecnia y Sistemas

APLICACIÓN DE TÉCNICAS DE REDES NEURONALES ARTIFICIALES AL DIAGNÓSTICO DE PROCESOS INDUSTRIALES

ANTONIO MUÑOZ SAN ROQUE

Tesis doctoral

Madrid 1996

APLICACIÓN DE TÉCNICAS DE REDES NEURONALES ARTIFICIALES AL DIAGNÓSTICO DE PROCESOS INDUSTRIALES

Universidad Pontificia Comillas de Madrid

Colección de Tesis Doctorales : Nº 209/1996

UNIVERSIDAD PONTIFICIA COMILLAS
MADRID

Escuela Técnica Superior de Ingenieros Industriales (ICAI)
Departamento de Electrotecnia y Sistemas

APLICACIÓN DE TÉCNICAS DE REDES NEURONALES ARTIFICIALES AL DIAGNÓSTICO DE PROCESOS INDUSTRIALES

ANTONIO MUÑOZ SAN ROQUE

Tesis doctoral

Madrid 1996

Antonio Muñoz San Roque

UNIVERSIDAD PONTIFICIA COMILLAS
MADRID

La Tesis Doctoral de D.Antonio Muñoz San Roque

Titulada : “Aplicación de técnicas de redes neuronales artificiales al diagnóstico de procesos industriales”

Director Dr. D. Miguel Angel Sanz Bobi

Resumen de la tesis

La tesis doctoral que aquí se presenta se enmarca dentro de las áreas de trabajo de diagnóstico y mantenimiento de procesos industriales, y propone un nuevo sistema de detección de anomalías incipientes basado en el modelado conexionista del funcionamiento normal de los componentes. El sistema propuesto está especialmente dirigido a resolver el problema de la detección de anomalías en aquellos casos en los que no existe una completa base de datos de fallo, y en los que el modelado físico del comportamiento de los componentes resulta inviable.

La solución propuesta consiste en caracterizar el comportamiento normal de los componentes involucrados mediante la aplicación de técnicas de modelado de procesos dinámicos no lineales con aproximadores funcionales. Como aproximadores funcionales se propone utilizar Redes Neuronales Artificiales supervisadas, tales como el Perceptrón Multicapa y la red PRBFN (aportación original de esta tesis). Estas herramientas, además de ofrecer una elevada capacidad de representación, poseen una estructura modular que las hacen altamente paralelizables y realizables en "*hardware*".

Una vez que el modelo de funcionamiento normal de cada componente ha sido identificado y ajustado, la caracterización del comportamiento normal se consigue delimitando la región del espacio de entrada "conocida" por cada modelo (región de confianza) y estableciendo las cotas máximas admisibles de los residuos de la estimación. Para delimitar la región de confianza de cada modelo, se propone utilizar una red PRBFN para estimar la función de densidad probabilista según la cual se distribuye el vector de regresores en el conjunto de entrenamiento utilizado para el ajuste del modelo. Esta misma red puede ser utilizada posteriormente para estimar de forma local la varianza de los residuos en condiciones de funcionamiento normal, obteniendo de esta forma las cotas máximas admisibles de los errores de estimación, y por consiguiente las bandas de funcionamiento normal.

El sistema de detección de anomalías resultante será por lo tanto capaz de identificar por sí mismo aquellas condiciones de operación, que por su novedad, no pueden ser tratadas con un grado de fiabilidad adecuado. La estimación de la cota máxima de los residuos en función del punto de operación permite además ajustar la sensibilidad del sistema de detección a las características propias de cada punto de operación. Estas dos propiedades, unidas a la intrínseca capacidad de adaptación del sistema y a la versatibilidad de los modelos de funcionamiento normal, constituyen las principales prestaciones del sistema de detección de anomalías que se propone en esta tesis.

Agradecimientos

Quisiera expresar en estas líneas mi más sincero agradecimiento a todas aquellas personas que de una u otra forma han contribuido a la realización de esta tesis.

En primer lugar quisiera dirigirme a mis padres para expresarles mi más profundo sentir de agradecimiento por todos aquellos sacrificios que mi formación ha supuesto para ellos. Asimismo quisiera agradecerles a ellos y a mis hermanos todo el cariño y apoyo moral que siempre me han prestado.

A mi director de tesis, Miguel Angel Sanz Bobi, quisiera agradecerle la ayuda incondicional que siempre me ha prestado y el haber sabido guiar mi trabajo en los momentos más delicados.

A todos mis amigos y compañeros del IIT (cuya enumeración sería afortunadamente demasiado larga) quisiera agradecerles todo el apoyo y amistad que siempre me han brindado.

A Yolanda González quisiera agradecerle todo el apoyo y la colaboración que me ha prestado para la realización y presentación de este trabajo.

A mi inseparable amigo José Villar y a mi entrañable colaborador Thomas Czernichow quisiera darles el último empujón para que acaben sus tesis.

Al profesor Alain Germond quisiera agradecerle su hospitalidad durante mi estancia en su laboratorio de Lausanne.

Finalmente quisiera agradecer al personal técnico de las centrales Térmicas de Meirama y de Anllares, propiedad de Unión Eléctrica Fenosa S.A., y en especial a Agustín Gimeno, Luis Zarauza y José Alvarez, su siempre inestimable colaboración y la disponibilidad de los datos que han servido como ejemplo de esta tesis.

Este trabajo ha sido parcialmente financiado por la Comisión Interministerial de Ciencia y Tecnología (CICYT) dentro de su Programa para la Formación del Personal Investigador.

Índice general

Lista de los símbolos más importantes

Letras minúsculas

a_i	activación de la unidad radial *i* de una red RBFN
$\boldsymbol{d}$	vector de salidas reales, medidas o deseadas
$\boldsymbol{e}$	vector de errores de estimación de las salidas (residuos)
$e_{máx}$	cota máxima del error de estimación o residuo
f	aproximador funcional
g	función a aproximar
h	número de unidades ocultas, índice estructural y dimensión VC
n	dimensión del espacio de entrada
n_d	número de retardos de la salidas reales en el vector de regresores
n_e	número de retardos de los errores de estimación en el vector de regresores
n_u	número de retardos de las entradas exógenas en el vector de regresores
n_y	número de retardos de la salidas estimadas en el vector de regresores
m	dimensión del espacio de salida
p	función de densidad probabilista
$p_{mín}$	cota de extrapolación
$p_{\boldsymbol{x}}$	estimación de la fdp del vector de entradas
$p_{\boldsymbol{x},i}$	estimación local de la fdp del vector de entradas en la unidad *i*
q	dimensión del espacio de parámetros del aproximador
$\boldsymbol{r}_i$	vector representante de la unidad radial *i* de una red RBFN
$s_{e,i}^2$	estimación de la varianza residual local en la unidad *i*
$\boldsymbol{u}$	vector de entradas exógenas
v_i	peso de la capa de salida de una red RBFN asociado a la unidad radial *i*
$\boldsymbol{w}$	vector de parámetros del aproximador funcional
$\boldsymbol{x}$	vector de entradas o regresores del aproximador funcional
$\boldsymbol{y}$	vector de salidas estimadas

Letras mayúsculas

$\boldsymbol{D}$	dirección de búsqueda en la optimización paramétrica
F	familia de funciones de aproximación
H	número de funciones de aproximación
$Kmín$	número mínimo de iteraciones de optimización paramétrica
$Kmáx$	número máximo de iteraciones de optimización paramétrica
L	medida de discrepancia
N	número de ejemplos de entrenamiento
M	número de ejemplos de test

$\Re$	conjunto de los números reales
R	función de riesgo
R_{emp}	función empírica de riesgo
R_{entr}	error de estimación del conjunto de entrenamiento
R_{test}	error de estimación del conjunto de test
R_{valid}	error de estimación del conjunto de validación
S	conjunto de muestras
S_{entr}	conjunto de entrenamiento
S_{test}	conjunto de test
S_{valid}	conjunto de validación
V	tamaño de la ventana para el cálculo de la tendencia
$\boldsymbol{W}$	espacio de parámetros de un aproximador funcional
$\boldsymbol{X}$	espacio de entrada del modelo de funcionamiento normal

Letras griegas

α	longitud del paso de optimización paramétrica
β	tendencia del error de test
$\beta_{máx}$	cota máxima de la tendencia del error de test
$\boldsymbol{\varepsilon}$	vector de ruido superpuesto a la salida
μ_i	factor de escala de la unidad radial *i* de una red RBFN
σ_i	ancho de la unidad radial *i* de una red RBFN
ζ_i	sensibilidad de la salida *y* respecto de la variable de entrada x_i

Operaciones

cov(x,y)	covarianza de *x* e *y*
lím	función límite
m_x	media muestral de *x*
s_x	desviación típica muestral de *x*
$\boldsymbol{x}^T$	traspuesto de $\boldsymbol{x}$
z	operador de adelanto
∇R	vector gradiente de *R* respecto del vector de parámetros
$\nabla^2 R$	matriz hessiana de *R* respecto del vector de parámetros

Distribuciones estadísticas

N(m,s)	distribución normal de media *m* y desviación típica *s*
U([a,b])	distribución uniforme en el intervalo *[a,b]*

1. Introducción

1.1 Estado actual del problema

Una de las tareas de la ingeniería que está teniendo mayor auge en los últimos tiempos es la de aumentar la fiabilidad, disponibilidad y seguridad de los procesos industriales. Por proceso industrial puede entenderse tanto una máquina o equipo, como un conjunto de ellos, que realizan una función productiva industrial concreta.

Este auge se ha visto impulsado por la criticidad de ciertos procesos como son las líneas de montaje, las centrales eléctricas, las centrales químicas, etc., en los que la interrupción de la producción por alguna anomalía imprevista puede poner en peligro vidas humanas, además de conllevar enormes pérdidas económicas por falta de producción, costes de reparación, degradación de componentes y empeoramiento de la calidad de servicio.

A esta situación se suma la necesidad económica de obtener un mayor rendimiento de las instalaciones existentes como alternativa a la inversión necesaria para la construcción y puesta en marcha de otras nuevas. Este aprovechamiento más intensivo de los medios de producción puede propiciar la aparición de averías y acelerar la degradación de los componentes, si no se siguen adecuadas políticas de mantenimiento y de operación que mantengan por un lado el buen estado de los componentes y no les sometan por otro a condiciones de trabajo a las que no puedan hacer frente.

El establecimiento de estas políticas hace necesario conocer en todo momento el estado de salud de los componentes y sus condiciones de operación, de tal forma que se puedan detectar de forma incipiente las anomalías que pudieran producirse, localizar el origen de las mismas y actuar en consecuencia.

Por otro lado, la continua y rápida evolución tecnológica que afecta a los procesos de producción desde la revolución industrial ha complicado la tarea de conocer el estado de salud de los componentes involucrados. El volumen de información al que hay que hacer frente, así como la extraordinaria complejidad técnica de los procesos, ha hecho necesario automatizar los mecanismos de supervisión mediante la instalación de sistemas de seguimiento continuo y de sistemas de diagnóstico.

Los sistemas de seguimiento continuo están encargados de suministrar de forma automática valores de características particulares de los componentes que integran el proceso, a partir de señales procedentes de sensores instalados en los mismos. La información que suministran es la base del estudio del estado actual del proceso, pero puede además ser aprovechada para llevar un registro histórico de la vida de los componentes.

La instalación de un sistema de seguimiento continuo ha de verse precedida por un estudio previo en el que se identifique, para cada uno de los componentes que integran el proceso, el conjunto de variables físicas técnica y económicamente accesibles mediante la instalación de sensores, que sean representativas del estado del componente. Este estudio requiere un profundo conocimiento físico del proceso bajo estudio, tanto a nivel teórico como práctico. Una vez instalados los sensores seleccionados, se dotará al sistema de un sistema de adquisición de datos encargado de muestrear periódicamente las señales de salida de los sensores, y almacenarlas en un sistema de almacenamiento masivo (ordenador central).

El sistema de diagnóstico parte de la información suministrada por el sistema de seguimiento continuo y realiza cuatro tareas fundamentales (*[Patton & Chen,1991]*):

1.- La detección de anomalías (*"fault detection"*), que es la encargada de determinar la presencia de faltas en los componentes que integran el proceso. Podemos definir el término de falta o anomalía en un componente como el estado caracterizado por su inesperada incapacidad, ya sea total o parcial, para llevar a cabo la tarea que tenía encomendada (*[Pau, 1981]*).

2.- El aislamiento de anomalías (*"fault isolation"*), encargado de determinar el conjunto de componentes del proceso que se han visto afectados por las anomalías detectadas. El proceso industrial ha debido previamente ser descompuesto en un conjunto jerarquizado de componentes, que serán considerados como las unidades funcionales sobre las que se aplican las tareas de diagnóstico.

3.- La identificación de anomalías (*"fault identification"*), que trata de determinar las causas concretas de las anomalías detectadas.

4.- La corrección de anomalías (*"fault accomodation"*), que trata de corregir en la medida de lo posible las anomalías identificadas, bien mediante la reconfiguración del proceso, bien mediante la emisión de mensajes al usuario.

Estas cuatro tareas suelen realizarse en dos pasos, la detección y el aislamiento de anomalías en primer lugar, y la identificación y la corrección de las mismas en segundo. A modo de ejemplo podemos considerar la estructura del sistema de diagnóstico utilizado en las diversas aplicaciones desarrolladas por el equipo de trabajo al que pertenezco: el sistema experto de diagnóstico (*[Sanz Bobi, 1992]*, *[Sanz Bobi et al., 1993][Sanz Bobi et al., 1994-1], [Sanz Bobi et al., 1994-2]*). Su estructura queda representada en la Figura 1.1:

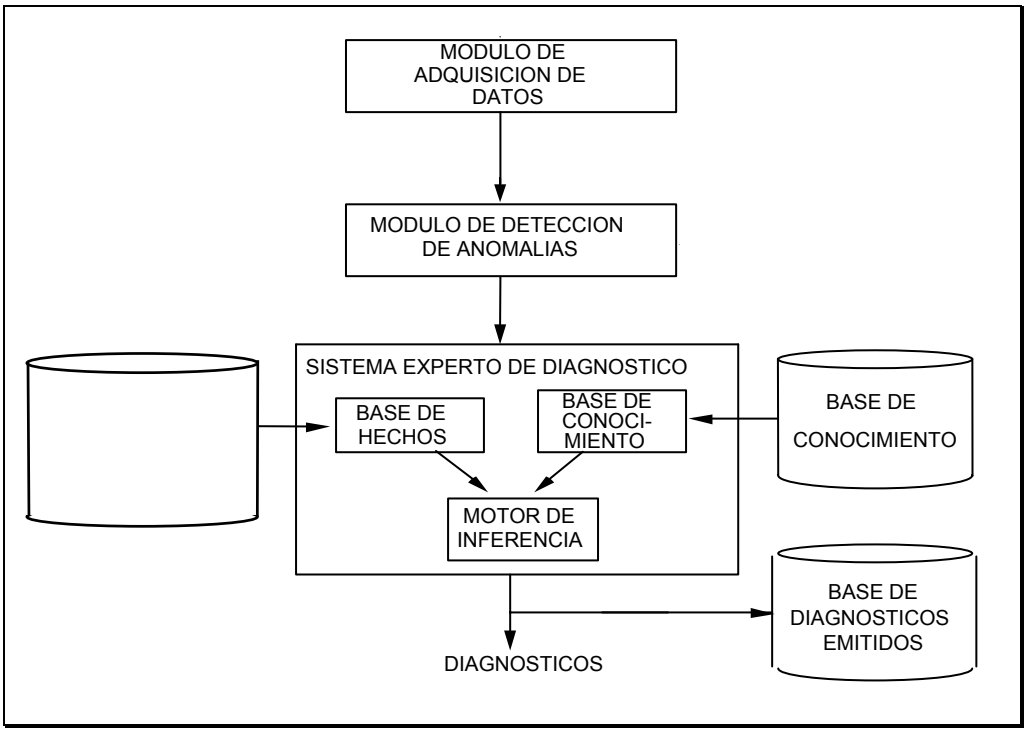

Figura 1.1: Estructura de un sistema experto de diagnóstico

Según la citada figura, los datos recogidos por el sistema de adquisición de datos pasan al módulo de detección de anomalías donde se realizan las tareas de detección y de aislamiento. En caso de observar alguna anomalía, el sistema de detección la comunica al sistema experto encargado de las tareas de identificación y de corrección.

Los tres componentes principales del sistema experto son la base de hechos, la base de conocimiento y el motor de inferencia.

La base de hechos contiene la descripción del entorno físico donde debe trabajar el sistema experto de diagnóstico. Ello incluye la enumeración y descripción de las características principales de los elementos que componen el proceso a monitorizar (componentes) y sus relaciones funcionales, así como la enumeración de las medidas que caracterizan su comportamiento ya sean continuas o no.

La base de conocimiento es el componente donde se encuentra representado en forma de reglas de producción tanto el conocimiento que posee el experto humano acerca de la experiencia del comportamiento del proceso como el propio conocimiento técnico del diseño y la configuración del mismo. En esta representación se encuentra una descripción de posibles causas de fallo relacionadas

con sus síntomas y posibles razones que las explican. El proceso de inferencia será iniciado por reglas capaces de interpretar las conclusiones extraídas por el detector de anomalías.

El motor de inferencia es el encargado de combinar la información incluida en la base de hechos acerca de los rasgos de las anomalías detectadas con el conocimiento incluido en la base de conocimiento con el propósito de investigar las causas de las anomalías detectadas. Normalmente el proceso de inferencia se basa en la comprobación de las hipótesis enunciadas en las reglas de conocimiento acudiendo para su verificación a los correspondientes valores almacenados en la base de hechos. Una vez que en una regla se verifican todas las hipótesis de su antecedente, el motor de inferencias procede al disparo o activación de sus consecuentes. Esto puede dar lugar a la activación de nuevas reglas o a la emisión de un diagnóstico.

Existen numerosas aplicaciones de los sistemas expertos al diagnóstico (*[Richardson, 1985], [Milne, 1987],[Tzafestas, 1989]*), pero aún así existen todavía cuellos de botella que limitan su aplicación. El más importante de ellos es la adquisición y representación del conocimiento. Por este motivo se ha comenzado ya a desarrollar sistemas automáticos de extracción de reglas basados en técnicas de redes neuronales (*[Sima, 1995]*), lógica borrosa (*[Wang, 1992]*) y algoritmos genéticos (*[Velasco, 1991]*) que completan el conjunto de reglas propuesto por los expertos con el análisis de los datos disponibles y amplían la capacidad de representación de las reglas tradicionales mediante el tratamiento de la incertidumbre.

La bibliografía revisada muestra una predominante aplicación de las Redes Neuronales Artificiales (RNA) a la tarea de la identificación de anomalías desde la perspectiva del reconocimiento de patrones (ver por ejemplo *[Dietz, 1988]*, *[Himmelblau et al., 1989]*, *[Sorsa et al., 1991]*, *[Barschforff, 1992], [Ranaweera, 1994]*). Otra línea de investigación abierta en este mismo campo es la que conjuga sistemas expertos de diagnóstico con Redes Neuronales Artificiales (*[Burattini & Tamburrini, 1991]*, *[Hudson & Cohen, 1991], [Yang et al., 1995]*). Estos sistemas tratan de sacar partido de la adaptabilidad, rapidez y robustez de estas estructuras de transformación de datos mediante su aplicación a un problema de tiempo real como es el diagnóstico.

En esta tesis nos vamos a centrar sin embargo en la aplicación de técnicas de RNA a las tareas de detección y aislamiento de anomalías. Estas dos tareas pueden considerarse como una sola si la detección se lleva a cabo a nivel de componente, ya que en este caso el conjunto de componentes afectados equivaldrá al conjunto de componentes en los que se ha detectado la presencia de una anomalía. Los métodos de detección de anomalías pueden dividirse según sus principios de operación en cuatro grandes categorías: el alcance de umbrales, los métodos basados en

redundancia física, los métodos basados en redundancia analítica, y los métodos basados en criterios estadísticos. A continuación se describe cada uno de ellos.

- **Alcance de umbrales**

La solución más extendida consiste simplemente en comprobar la permanencia de variables individuales dentro de unos límites preestablecidos o umbrales. En caso de sobrepaso de estos límites se activan de forma automática las alarmas correspondientes, quedando en manos del personal encargado la interpretación de las mismas. Esta solución, pese a su extrema sencillez, tiene las siguientes desventajas (*[Gertler, 1991]*): en primer lugar el establecimiento de los límites permitidos suele realizarse con criterios muy conservadores debido al amplio rango de variación que pueden tener las variables medidas. En segundo lugar la aparición de una falta simple en un componente puede provocar el que varias variables excedan sus límites permitidos, complicando la tarea de identificación de la anomalía. La primera consecuencia de la aplicación práctica de esta metodología es que las indisponibilidades se detectan una vez que sus efectos han causado daños importantes. Por otro lado deja en manos del usuario la difícil tarea de interpretar la secuencia de alarmas que se produce con el fin de emitir un diagnóstico sobre el estado de los equipos y poder tomar acciones correctoras. Las deficiencias descritas plantearon la necesidad de incorporar un conocimiento más profundo del proceso bajo estudio con vistas a realizar la acción del diagnóstico. Este conocimiento ha de incluir las ligaduras intrínsecas entre las distintas variables así como una perspectiva histórica de la evolución del proceso.

- **Redundancia física**

Este método consiste en duplicar físicamente componentes del proceso (como por ejemplo utilizar múltiples sensores para la misma medida, preferiblemente basados en principios distintos), con el fin de poder comprobar la consistencia de los elementos redundantes, y de esta forma realizar la detección de anomalías. El elevado coste que suele acarrear la duplicación física de componentes suele limitar sus campos de aplicación a sistemas críticos de seguridad.

- **Redundancia analítica**

La mayor parte de los trabajos que se están llevando a cabo en estos momentos en el área de detección de anomalías están basados en el principio de redundancia analítica, y son también conocidos bajo el nombre de métodos basados en modelos (*"model-based fault detection and isolation"*) (*[Chow & Willsky, 1984]*, *[Frank, 1990], [Gertler, 1991], [Patton & Chen, 1991]*).

Estos algoritmos constan básicamente de dos etapas (ver Figura 1.2): (a) la generación de residuos, y (b) su posterior interpretación por la lógica de detección. Los residuos son cantidades que miden la inconsistencia entre los valores actuales de

las variables y sus valores predichos por los modelos matemáticos de funcionamiento normal. Se calculan a partir de las variables medidas y son idealmente nulos en ausencia de anomalías. La lógica de detección es la encargada de analizar estadísticamente el grado de significación de estos residuos para decretar el estado de anomalía.

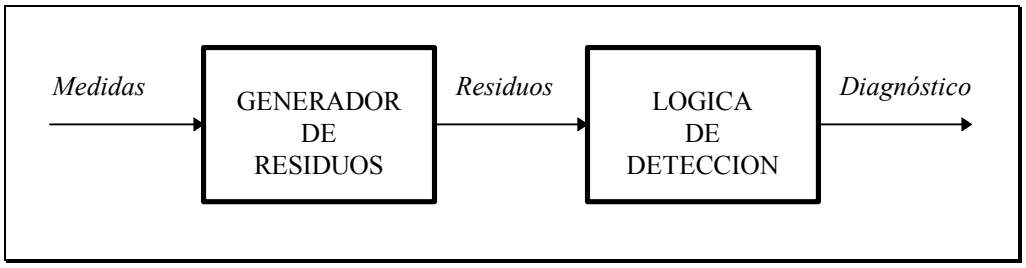

Figura 1.2: Estructura básica de un sistema de detección de anomalías basado en redundancia analitica (tomado de [Chow & Willsky, 1984])

Como generador de residuos suele utilizarse un modelo dinámico de funcionamiento normal que predice la evolución de una de las variables medidas (salida) en función de la evolución de otras variables medidas (entradas). El error de estimación puede entonces ser utilizado como residuo (ver Figura 1.3):

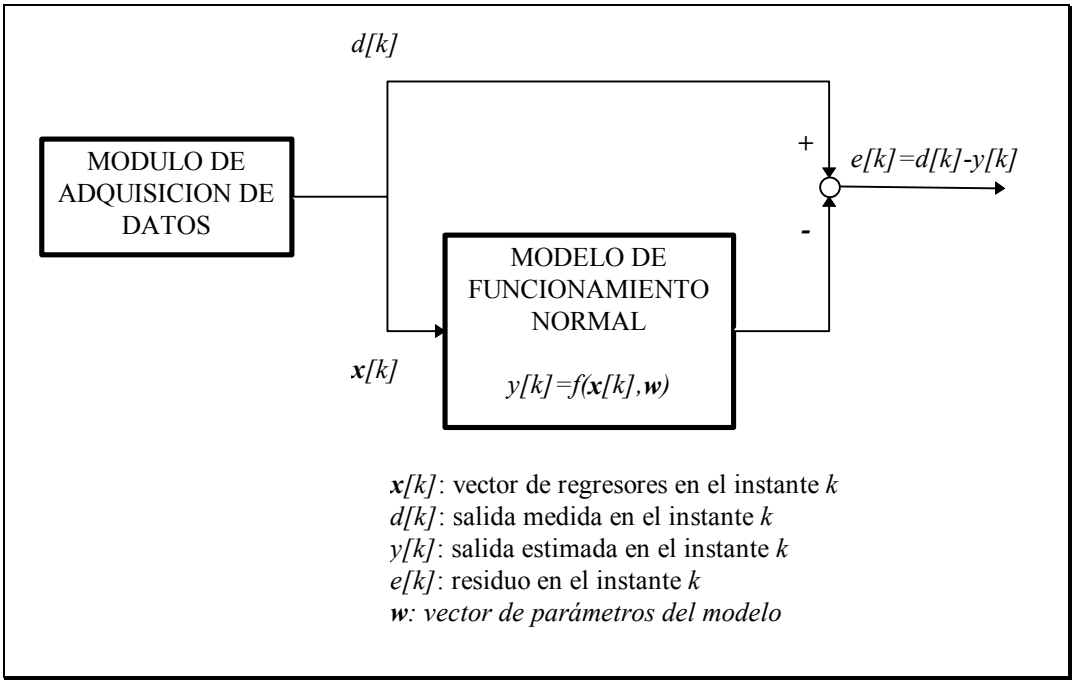

Figura 1.3: Generador de residuos basado en modelo de funcionamiento normal

La gran mayoría de los estudios realizados en redundancia analítica utilizan como modelo de funcionamiento normal un modelo lineal, ya sea en ecuaciones de estado o en términos de funciones de transferencia, con modelo aditivo de faltas. En estas condiciones es posible definir tres estrategias diferentes para la generación de los residuos (en *[Gertler, 1991]* puede encontrarse una buena introducción a estos métodos): las ecuaciones de paridad o relaciones de consistencia, el llamado observador de diagnóstico, y los filtros de Kalman.

En el caso general de procesos dinámicos no lineales pueden distinguirse básicamente dos tipos de modelos de funcionamiento normal: los modelos físicos y los modelos de caja negra. Los primeros aplican leyes físicas para ligar las variables que intervienen en el proceso. Los parámetros que intervienen en dichas ecuaciones tienen un significado físico, de tal forma que el conocimiento queda explícitamente reflejado en ellas. Sin embargo la obtención y utilización de estos modelos no siempre es posible. En primer lugar la complejidad de los procesos que hay que modelar hace que su modelado físico sea muy costoso e incluso una vez obtenido el modelo, el tiempo de cálculo necesario para su evaluación puede hacerlos inutilizables para una aplicación en tiempo real como es el diagnóstico. Por otro lado los modelos físicos requieren datos de diseño que conllevan dos problemas. En primer lugar resulta difícil y en ocasiones imposible extraer todos los datos necesarios de la documentación de diseño. En segundo lugar la experiencia demuestra que el comportamiento real de los componentes dista mucho de lo predicho por los datos de diseño.

Los modelos de caja negra están basados en criterios estadísticos capaces de modelar las relaciones existentes entre un conjunto de entradas y otro de salidas. Estas variables externas son variables físicas, pero el resto de variables y parámetros envueltos en el modelo pueden no tener significado físico. Los parámetros del modelo se ajustan a partir de un conjunto de medidas reales que caracterizan el comportamiento del proceso en condiciones de normalidad. Los modelos de caja negra basados en criterios estadísticos clásicos (*[Box-Jenkins, 1976]*) presuponen un comportamiento lineal del proceso, y se limitan a ajustar los parámetros de modelos lineales para dar forma a la relación de entrada y salida. Esta limitación plantea la necesidad de establecer una metodología clara de modelado que permita generar de forma automática una relación funcional, en general no lineal, que se adapte al conjunto de medidas representativo del comportamiento normal del proceso.

El resurgimiento de las técnicas conexionistas, y en concreto el de las Redes Neuronales Artificiales (RNA) supervisadas (*[Rumelhart et al., 1986b]*), trajo consigo el desarrollo de nuevos aproximadores funcionales no lineales con extraordinarias capacidades de representación y adaptación. Estas nuevas herramientas han servido como base para la extensión de la teoría clásica de identificación de sistemas (*[Ljung, 1987]*) al campo no lineal (*[Ljung & Sjöberg,*

1992], [Sjöberg, 1995]). Sin embargo, el desarrollo de las RNA tuvo la desgracia o la gran ventaja de tener su origen en la psicología. Este origen, escabroso para algunos y apasionante para otros, ha motivado el que la comunidad científica relacionada con la identificación de sistemas haya retrasado su primera cita con estas "nuevas" herramientas hasta hace pocos años. La espera ha dado lugar al nacimiento de una nueva terminología conexionista, cuyos conceptos ya tenían nombre en el área de la identificación de sistemas. Así, el profesor L. Ljung propone en *[Ljung & Sjöberg, 1992]* el siguiente diccionario:

Estructura del modelo	Red
Orden del modelo	Número de unidades ocultas
Estimación	Aprendizaje, entrenamiento
Datos de estimación	Conjunto de entrenamiento
Iteración	Ciclo de entrenamiento
Validación	Generalización
Datos de validación	Conjunto de generalización
Algoritmo recursivo del gradiente	Retropropagación
Sobre-ajuste	Sobre-entrenamiento

En ese mismo trabajo presenta las RNA, y más concretamente al perceptrón multicapa, como una estructura más a tener en cuenta (*"Just a particular model structure!"*), con capacidad de aproximación universal (*[Cybenko, 1989]*) y una estructura interna que permite su eficaz realización tanto a nivel *"software"* como *"hardware"*. Este mismo punto de vista es compartido por otros reconocidos investigadores (*[Barron, 1989]*). Sin embargo, ¿qué tienen de especial las RNA, a parte de sus ventajas de implantación, que les ha dado un protagonismo no compartido por otras estructuras de caja negra preexistentes? (*"Just a particular model structure?" [Ljung & Sjöberg, 1992]*). El profesor Ljung propone dos razones: la primera es que los sistemas que se suelen encontrar en la práctica presentan características de saturación para entradas elevadas. La propia estructura de las RNA incorpora este efecto de forma "natural", en contraposición por ejemplo a los aproximadores polinomiales. La segunda razón está relacionada con la redundancia interna de las RNA. Esta redundancia explica la relativa facilidad con que se maneja un elevado número de parámetros o pesos en el ajuste de RNA, cuando suele ser una tarea tan costosa con otro tipo de aproximadores. En efecto lo que ocurre es que la regularización[i] llevada a cabo de forma implícita o explícita hace que el número efectivo de parámetros libres sea mucho menor al número de parámetros existentes. A priori se desconoce el número de parámetros efectivos necesarios, basta con probar con un número sobrado y dejar que la regularización se encargue de anular el efecto de los parámetros sobrantes de una forma automática.

[i] La teoría de la regularización, que impone condiciones de "suavidad" a las funciones de aproximación, será tratada en el Capítulo 5.

En definitiva podemos concluir que las RNA se presentan como una herramienta de aproximación funcional a tener en cuenta en todo problema de identificación de sistemas con características no lineales. Estas características han multiplicado el número de aplicaciones de RNA a la detección y aislamiento de anomalías basados en modelos (*[Sorsa et al., 1991]*, *[Chow et al., 1993]*, *[Los Arcos et al., 1993]*, *[Ayoubi & Isermann, 1994]*, *[Renders et al., 1995]*, *[Polycarpou & Vemuri, 1995]*).

- **Métodos basados en criterios estadísticos**

Además de los métodos de detección de anomalías basados en modelos estadísticos de caja negra, existen otras filosofías de detección de anomalías que analizan de forma estadística los datos recogidos en continuo. Entre ellas caben destacar:

- El análisis de tendencias

Este método trata de detectar tendencias sistemáticas crecientes o decrecientes que alejan el comportamiento actual de determinadas variables características de su comportamiento esperado en condiciones de funcionamiento normal. A modo ilustrativo puede consultarse el trabajo dirigido por el director de esta tesis en *[Maturana, 1988]*.

- Reconocimiento de patrones

Los sistemas basados en reconocimiento de patrones tratan el problema de la detección de anomalías como un problema de clasificación de los vectores de características que definen el estado y la evolución del proceso, en el espacio de posibles faltas ("situación normal/situación anómala del tipo *x*"). De esta forma comparan la evolución presente del proceso con la evolución que típicamente muestran algunos tipos de fallos, con objeto de averiguar las posibles semejanzas y de ahí advertir de la posible anomalía. Estos métodos son especialmente recomendables cuando se puede construir, ya sea mediante simulación o mediante acceso a un registro histórico, un diccionario de fallos donde queden reflejados los síntomas típicos de determinados fallos. Existen varios métodos clásicos de clasificación, como el clasificador de Bayes (*[Fukunaga, 1972]*) o el de los k-ésimos vecinos más próximos (*"K-nearest neighbor algorithm"*, *[Dasarathy, 1991]*), pero el método de reconocimiento de patrones que actualmente está siendo más investigado en el campo del diagnóstico es el basado en Redes Neuronales Artificiales (ver por ejemplo *[Hoskins & Himmelblau, 1988]*, *[Watanabe et al., 1989], [Ebron et al., 1990], [Leonard & Kramer, 1991-1993]*). La mayoría de las aplicaciones de RNA al diagnóstico publicadas hasta la fecha siguen esta filosofía. Sin embargo estos métodos están dando paso a los métodos basados en modelos conexionistas de funcionamiento normal, debido fundamentalmente a la

dificultad encontrada a la hora de construir el diccionario de fallos, y a los avances realizados en la identificación de sistemas no lineales.

- Métodos de control de calidad

Estos métodos caracterizan de forma estadística el funcionamiento normal del proceso, y utilizan técnicas de control de calidad (*[Duncan, 1974], [Motgomery, 1985]*) como los cuadros de control de la media y de rangos para detectar desviaciones significativas. A modo de ejemplo puede consultarse la referencia *[Sanz Bobi et al., 1994-2]*.

Podemos pues concluir de esta exposición que las RNA se están aplicando al diagnóstico de procesos industriales tanto a nivel de detección y aislamiento como de identificación de anomalías. En el caso concreto de la detección de anomalías existen básicamente dos líneas de aplicación de RNA: los métodos basados en el reconocimiento de patrones, y los métodos basados en los modelos de funcionamiento normal (métodos basados en redundancia analítica). Los primeros tienen ya una larga historia de vida, pero padecen el problema de la necesidad del diccionario de fallos. Los segundos están ahora en pleno auge, gracias a los avances realizados en la identificación de sistemas no lineales con RNA.

1.2 Planteamiento de la tesis

En esta tesis nos vamos a centrar en la aplicación de técnicas de Redes Neuronales Artificiales al diagnóstico de procesos industriales, y más concretamente, a la detección de anomalías incipientes en procesos industriales.

En este entorno vamos a considerar a las RNA supervisadas como una estructura más de aproximación funcional, que nos permitirá modelar relaciones no lineales de entrada/salida a partir de un conjunto de muestras de esta relación.

Los conceptos de anomalía y de proceso industrial han quedado ya definidos en el apartado anterior. Falta pues tan sólo por definir el concepto de detección de anomalías incipientes, que se refiere a aquella pronta detección que trata de anticiparse a las posibles consecuencias negativas de las anomalías.

El objetivo final de la tesis será el proponer un sistema de detección de anomalías incipientes basado en técnicas conexionistas, especialmente diseñado para aquellos casos en los que se cumplen las siguientes condiciones (la experiencia demuestra que esta suele ser la situación más común en la mayoría de los procesos industriales):

- Existe (o cabe la posibilidad de instalar) un sistema de seguimiento continuo que proporciona periódicamente las medidas de un conjunto de variables suficientemente representativo del funcionamiento normal de los componentes a supervisar.

- No se dispone de una base de datos de fallo suficientemente rica para aplicar técnicas de detección de anomalías basadas en el reconocimiento de patrones de fallo, ni es viable su obtención mediante simulación.

- El modelado físico de los componentes del proceso resulta inviable, bien por la complejidad de los fenómenos físicos involucrados, bien por la ausencia de información técnica.

Bajo estas hipótesis, la vía más razonable que queda libre para resolver el problema de la detección de anomalías es caracterizar el funcionamiento normal de los componentes que integran el proceso, de forma tal que sea posible detectar las anomalías cuando se observe alguna desviación significativa del comportamiento actual al comportamiento patrón.

Los avances realizados en la identificación de sistemas con RNA abren un nuevo camino para esta caracterización: el modelado conexionista de procesos dinámicos no lineales. A partir de estas técnicas, esta tesis propone un sistema de detección de anomalías incipientes basado en el modelado conexionista del funcionamiento normal de los componentes involucrados en el proceso.

El sistema propuesto, además de predecir la evolución esperada de las variables de salida consideradas, será capaz de detectar la entrada del sistema en nuevos puntos de operación no contemplados por los modelos, y de estimar de forma local las cotas máximas de los residuos. Estas características disminuyen significativamente la ocurrencia de falsas alarmas y permiten ajustar la sensibilidad del sistema de detección a las características propias de los procesos subyacentes.

1.3 Organización de la exposición

El objetivo de esta exposición es presentar el sistema de detección de anomalías incipientes que se propone en esta tesis, y todos los aspectos necesarios para su aplicación a un problema real. No obstante, la descripción del sistema de anomalías propuesto no se realizará hasta el Capítulo 6, con el fin de presentar previamente todas las herramientas y técnicas que se requieren para la obtención de los modelos de funcionamiento normal. Se ha pretendido además que cada capítulo tenga entidad propia, con el fin de que sean apartados autocontenidos.

Como se muestra en la Figura 1.4, la tesis comenzará introduciendo el problema de la aproximación funcional con modelos de caja negra (Capítulo 2). En el Capítulo 3 se presentarán las técnicas necesarias para modelar procesos dinámicos no lineales con aproximadores funcionales. En los Capítulos 4 y 5 sustituiremos las cajas negras de los dos capítulos anteriores por Redes Neuronales Artificiales. El Capítulo 6 utilizará las técnicas de modelado presentadas en los capítulos anteriores para construir el sistema de detección de anomalías finalmente propuesto. El contenido de estos capítulos se detalla a continuación.

En el Capítulo 2 se tratará el tema de la aproximación funcional, considerando al aproximador funcional como una caja negra capaz de llevar a cabo una transformación ajustable de un vector de entradas en una salida escalar. Estos aproximadores tendrán dos niveles de adaptación: un nivel estructural relacionado con la capacidad de representación del aproximador, y por debajo del anterior, un nivel paramétrico que permite ajustar la relación funcional una vez fijada su estructura. En este mismo capítulo se presentará un esquema general de aproximación funcional que sistematizará el procedimiento de ajuste de los aproximadores, descomponienedo el problema en dos optimizaciones parciales: la optimización estructural, encagada de fijar la estructura interna del aproximador, y la optimización paramétrica, encargada de ajustar los parámetros del mismo. La teoría de aprendizaje establecerá los heurísticos que regirán la optimización estructural. La optimización paramétrica será tratada como un problema clásico de optimización no lineal sin restricciones, y en este sentido serán ensayados distintos algoritmos. Finalmente se describirá una herramienta de análisis de la influencia de las variables de entrada en la salida del aproximador, que jugará un papel decisivo en la identificación de sistemas no lineales: el Análisis Estadístico de Sensibilidades.

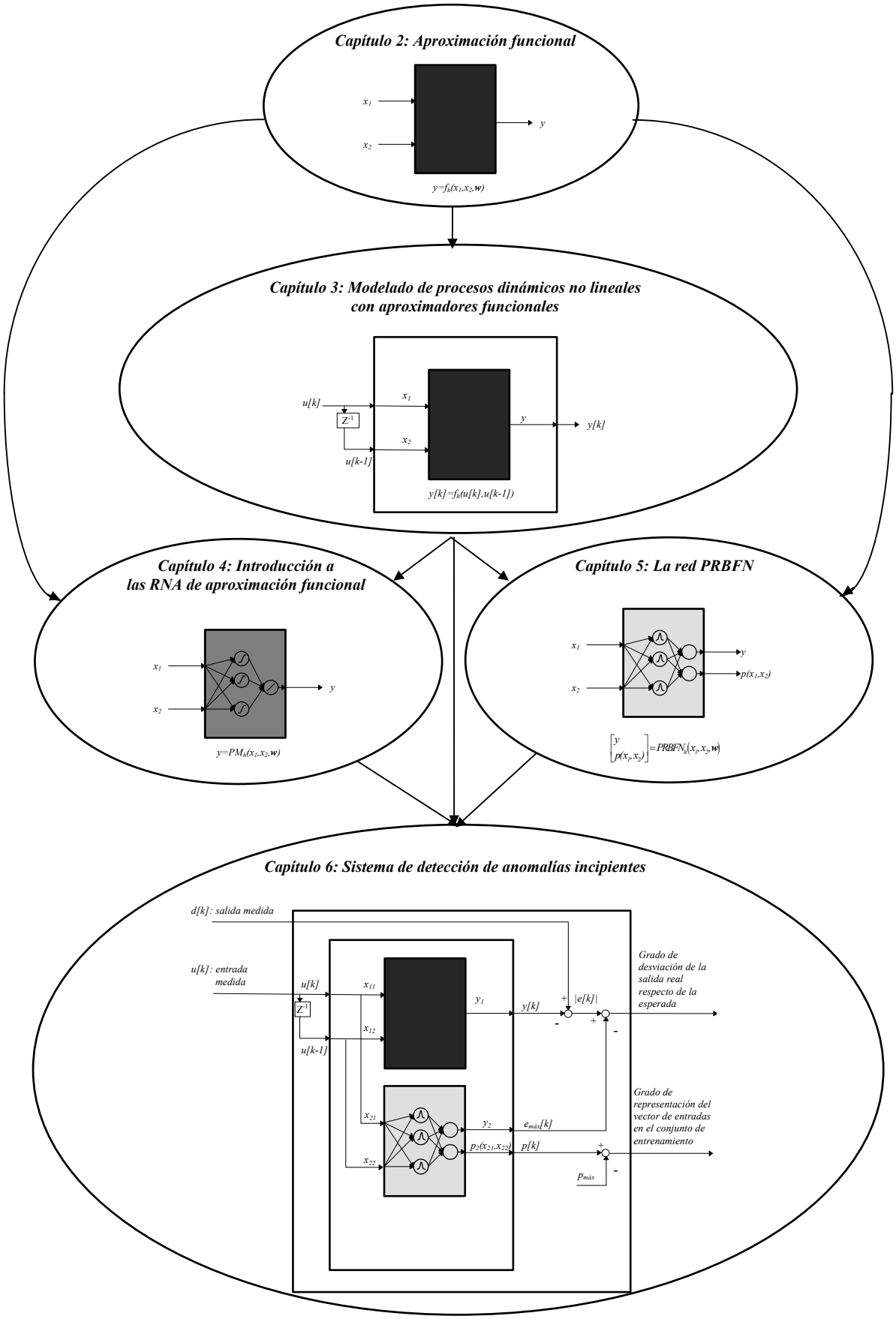

Figura 1.4: Organización de la exposición

En el Capítulo 3 se presentan las técnicas de modelado de procesos dinámicos no lineales con aproximadores funcionales, como una extensión directa de los modelos lineales de series temporales. Las técnicas propuestas no se limitan al caso conexionista, ya que consideran al aproximador funcional como una caja negra de estructura interna desconocida. Los modelos y las técnicas desarrollados en este capítulo permitirán aplicar métodos de detección de anomalías basados en redundancia analítica a procesos con características no lineales.

El Capítulo 4 ofrece una introducción al campo de las Redes Neuronales Artificiales (RNA), centrándose inmediatamente en las RNA supervisadas que pueden ser utilizadas como aproximadores funcionales. Dentro de estas estructuras destaca por su relevancia el Perceptrón Multicapa (PM), que será integrado al final de este capítulo dentro del esquema de aproximación funcional propuesto en el Capítulo 2.

El Capítulo 5 se centra en otro tipo de RNA supervisadas, que en lugar de realizar aproximaciones de forma global, las realizan de forma local: las redes de funciones base radiales (RBFN). Tras introducir las redes RBFN como resultado de la teoría de la regularización, se presentará la estructura GRNN (RNA de regresión generalizada). Esta estructura conexionista basada en criterios probabilistas servirá como punto de partida para la descripción de la familia de RNA que se proponen en esta tesis: las redes PRBFN. Estas redes, además de poder ser utilizadas como aproximadores funcionales, pueden ser utilizadas para estimar funciones de densidad probabilistas. Esta característica les otorgará un papel fundamental en el sistema de detección de anomalías propuesto.

El Capítulo 6 presenta el sistema de detección de anomalías incipientes propuesto en esta tesis. Este sistema está basado en el modelado conexionista del funcionamiento normal de los componentes que integran el proceso. Para este modelado se propone utilizar las técnicas vistas en el Capítulo 3, utilizando como aproximadores funcionales las RNA tratadas en los Capítulos 4 y 5. El ajuste de estos aproximadores podrá realizarse según el esquema de ajuste descrito en el Capítulo 2. Los modelos de funcionamiento normal así obtenidos se completan con la delimitación de sus regiones de confianza (regiones del espacio de entrada en las que ha sido posible caracterizar los residuos de los modelos), y con la estimación local de sus cotas máximas de error. Esta práctica reduce el riesgo de falsas alarmas y ajusta la sensibilidad del sistema de detección a las características propias de los procesos físicos subyacentes.

El Capítulo 7 recoge las conclusiones metodológicas y específicas más significativas del trabajo realizado, las aportaciones más relevantes y algunas líneas de futuros desarrollos que han quedado abiertas por el trabajo desarrollado.

A modo de apéndice se presentan los árboles de selección de datos. Esta estructura de almacenamiento y de selección de datos permite filtrar de forma secuencial un conjunto de muestras vectoriales para obtener un subconjunto actualizado de datos uniformemente repartidos en el espacio muestral. Esta herramienta ha sido diseñada para seleccionar los conjuntos de entrenamiento y de test de los modelos de funcionamiento normal, y para poder actualizar periódicamente los modelos ajustados con los nuevos datos que han sido incorporados a la base.

2. Aproximación funcional

El modelado de procesos estáticos a partir de un conjunto de muestras de la relación entrada/salida puede ser considerado como un problema de aproximación funcional, en el que se trata de hallar una expresión matemática que sea capaz de reproducir la relación inyectiva que existe entre las entradas y las salidas del proceso.

En este capítulo presentaremos un esquema general de aproximación funcional que pretende sistematizar el procedimiento de ajuste de aproximadores funcionales. Este procedimiento descompone el problema global del modelado de relaciones entrada/salida en dos optimizaciones parciales: la optimización estructural y la optimización paramétrica. La primera de ellas determinará la estructura del aproximador, mientras que la segunda ajustará sus parámetros.

En ambos casos tomaremos como criterio de optimalidad la capacidad de generalización del modelo resultante, es decir, su capacidad de aproximar ejemplos no tratados durante su etapa de ajuste o "aprendizaje".

Las estrategias de aprendizaje supervisado presentadas en este capítulo servirán para el entrenamiento de las redes neuronales que serán utilizadas como aproximadores funcionales para el modelado del comportamiento normal de los componentes.

2.1 Planteamiento del problema

Supongamos que tenemos ante nosotros un proceso estático (las salidas del proceso en el instante *t* sólo dependen de sus entradas en el mismo instante *t* y no de valores pasados) e invariante en el tiempo, que transforma un vector de entradas $\boldsymbol{x} \in \Re^n$ en un vector de salidas $\boldsymbol{d} \in \Re^m$, y que la única información disponible del proceso es un conjunto de muestras de la forma:

$$S = \{(\boldsymbol{x}[1], \boldsymbol{d}[1]), (\boldsymbol{x}[2], \boldsymbol{d}[2]), \ldots, (\boldsymbol{x}[N], \boldsymbol{d}[N])\}$$

Ecuación 2.1

El objetivo de la aproximación funcional es obtener una expresión matemática (modelo) que reproduzca lo más fielmente posible la transformación $\boldsymbol{x} \rightarrow \boldsymbol{d}$, esto es, que sea capaz de generar salidas correctas para vectores de entrada no contenidos en el conjunto de muestras *S*.

Podemos considerar el problema de la aproximación funcional como un problema de aprendizaje supervisado, en el que el ajuste y la adaptación del aproximador se interpreta como un proceso de aprendizaje, supervisado por el conjunto de muestras *S*.

Un modelo de aprendizaje supervisado aplicado al problema de aproximación funcional consta fundamentalmente de tres componentes (ver Figura 2.1):

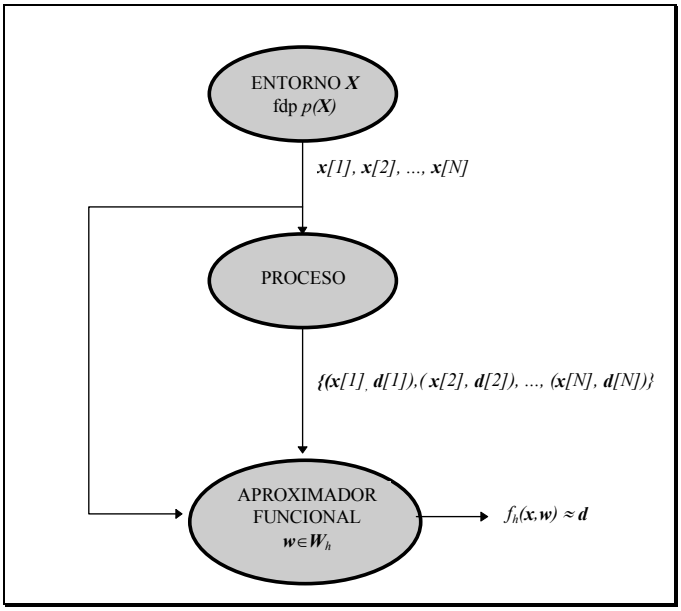

Figura 2.1: Modelo de aprendizaje supervisado para aproximación funcional

- **Entorno:** suministra el vector de entradas $\boldsymbol{x} \in \Re^n$ según una función de densidad probabilística (fdp) $p(\boldsymbol{x})$ determinada, invariante en el tiempo, y desconocida. En general, las entradas suministradas por el entorno no serán tomadas del espacio de entrada de una forma uniforme, sino que existirán regiones de este espacio más representadas que otras. La fdp $p(\boldsymbol{x})$ tiene como objetivo modelar este fenómeno.

- **Proceso:** proporciona la respuesta deseada $\boldsymbol{d} \in \Re^m$ para cada vector de entradas $\boldsymbol{x}$, de acuerdo con la fdp condicionada $p(\boldsymbol{d}/\boldsymbol{x})$ que es también invariante en el tiempo y desconocida. Este modelo probabilista permite concebir procesos en los que las salidas no quedan perfectamente determinadas por las entradas, sino que se admite un cierto nivel de incertidumbre. En el caso ideal de que las entradas del proceso determinen de forma unívoca las salidas, las fdp condicionadas tomarán la forma de funciones impulso unitarias (*[Oppenheim & Willsky, 83]*), también llamadas "deltas de Dirac", centradas en el correspondiente valor de las salidas (sólo se considera como posible un único vector de salida para cada vector de entrada). En el caso más general, si el vector de entradas $\boldsymbol{x}$ contiene todas las variables explicativas de la salida $\boldsymbol{d}$, se espera que para cada valor del vector de entradas del proceso, los posibles vectores de salida se concentren en torno a un único valor, con poca dispersión (esta dispersión será considerada como ruido). Esta situación da lugar a fdp condicionadas $p(\boldsymbol{d}/\boldsymbol{x})$ unimodales y de baja varianza.

  En esta tesis nos vamos a limitar a modelos de error de ruido blanco aditivo, de tal forma que supondremos que los vectores $\boldsymbol{x}$ y $\boldsymbol{d}$ están ligados por una función desconocida g tal que $\boldsymbol{d}=g(\boldsymbol{x})+\boldsymbol{\varepsilon}$, donde $\boldsymbol{\varepsilon}$ es un proceso de ruido blanco de media nula. Bajo estas suposiciones, la mejor estimación $\boldsymbol{y}$ de $\boldsymbol{d}$ que puede conseguirse es $\boldsymbol{y}=g(\boldsymbol{x})$.

- **Aproximador funcional:** es capaz de llevar a cabo un conjunto de transformaciones funcionales de entrada/salida descritas por:

$$F = \{ \boldsymbol{y}= f_h(\boldsymbol{x},\boldsymbol{w}) \;/\; \boldsymbol{x} \in \Re^n, \boldsymbol{y} \in \Re^m, \; \boldsymbol{w} \in \boldsymbol{W}_h \;, h=1,...,H \}$$

Ecuación 2.2

  donde $\boldsymbol{y}$ es la salida del sistema en respuesta al vector de entradas $\boldsymbol{x}$, h es un índice estructural que permite identificar la estructura interna del aproximador y $\boldsymbol{w}$ es el vector de parámetros libres que el aproximador funcional ha tomado del espacio de parámetros $\boldsymbol{W}_h$.

  De esta forma quedan definidos dos niveles de adaptación del aproximador: un nivel estructural determinado por el índice h, que determina la estructura

interna del aproximador y por tanto su capacidad de representación, y un nivel paramétrico gobernado por el vector de parámetros $\boldsymbol{w}$, que da forma a la función f_h una vez determinado h.

Así por ejemplo, si consideramos el aproximador polinomial definido de $\Re$ en $\Re$ por:

$$y = f_h(x,\boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_h x^h$$

Ecuación 2.3

podremos asociar el índice estructural h con el grado del polinomio aproximador, el vector de parámetros libres $\boldsymbol{w}$ con los $(h+1)$ coeficientes del polinomio, y los espacios paramétricos $\boldsymbol{W}_h$ con $\Re^{h+1}$.

El problema del aprendizaje es el de seleccionar, de un conjunto dado de transformaciones de entrada/salida, aquella función $f_{h^*}(\boldsymbol{x},\boldsymbol{w}^*)$ que aproxima de alguna forma óptima (en sentido estadístico) el vector de respuestas deseadas $\boldsymbol{d}$. Para ello habrá que determinar la estructura óptima del aproximador funcional (dada por el índice estructural h^*), y el óptimo vector de parámetros $\boldsymbol{w}^*$ para esta estructura.

Para llevar a cabo este proceso de optimización es necesario establecer una función de error de la aproximación que nos permita evaluar la bondad del aproximador. La función de error ideal que se desearía poder evaluar en todo problema de aproximación funcional es la función de "riesgo"[i]. Esta función queda definida de la siguiente manera:

> "Sea $L(\boldsymbol{d}, f_h(\boldsymbol{x},\boldsymbol{w}))$ una medida de la discrepancia entre el vector de respuestas deseadas $\boldsymbol{d}$, correspondientes al vector de entradas $\boldsymbol{x}$, y la salida $f_h(\boldsymbol{x},\boldsymbol{w})$ del aproximador funcional. El valor esperado de la discrepancia queda definido por la función de riesgo (*[Vapnik, 1992]*):
>
> $$R(h,\boldsymbol{w}) = \int L(\boldsymbol{d}, f_h(\boldsymbol{x},\boldsymbol{w}))\ d\big(p(\boldsymbol{x},\boldsymbol{d})\big)$$
>
> *Ecuación 2.4*
>
> donde $p(\boldsymbol{x},\boldsymbol{d})$ es al fdp conjunta del vector de entradas $\boldsymbol{x}$ y de la salida deseada $\boldsymbol{d}$, y la integral ha sido tomada en el sentido de Rieman-Stieltjes."

[i] El término "función de riesgo" en el ámbito de la teoría de aprendizaje no ha de ser confundido con el concepto de "riesgo" tratado en seguridad y fiabilidad.

Como medidas de discrepancia más habituales podemos citar el error cuadrático ($L(\mathbf{d}, f_h(\mathbf{x},\mathbf{w})) = \|\mathbf{d} - f_h(\mathbf{x},\mathbf{w})\|^2$) y el error absoluto ($L(\mathbf{d}, f_h(\mathbf{x},\mathbf{w})) = |\mathbf{d} - f_h(\mathbf{x},\mathbf{w})|$) El primero de ellos es una función continua y derivable del vector de parámetros $\mathbf{w}$ (siempre que f_h lo sea) y penaliza los errores grandes. El error absoluto sin embargo no es derivable en el origen, lo que limita su aplicación a algoritmos de aprendizaje que no están basados en el gradiente del error respecto de los parámetros.

Bajo estas condiciones, el objetivo del aprendizaje es minimizar la función de riesgo $R(h,\mathbf{w})$ sobre el conjunto de funciones $f_h(\mathbf{x},\mathbf{w})$, con $\mathbf{w} \in \mathbf{W}_h$. Desgraciadamente, la evaluación de la función de riesgo es complicada por el hecho de ser desconocida la fdp real del proceso:

$$p(\mathbf{x},\mathbf{d}) = p(\mathbf{d}/\mathbf{x})\, p(\mathbf{x}).$$

Ecuación 2.5

donde $p(\mathbf{d}/\mathbf{x})$ es la fdp de la salida deseada condicionada al valor de las entradas $\mathbf{x}$, y $p(\mathbf{x})$ es la fdp de las entradas.

En la mayoría de los problemas prácticos de aproximación funcional, toda la información disponible se encuentra contenida en un conjunto de N ejemplos de entrenamiento, independientes e idénticamente distribuidos:

$$S_{entr}=\{(\mathbf{x}[1], \mathbf{d}[1]),(\mathbf{x}[2], \mathbf{d}[2]), ..., (\mathbf{x}[N], \mathbf{d}[N])\}$$

Ecuación 2.6

al que llamaremos conjunto de entrenamiento. Por lo tanto tendremos que estimar la función de riesgo a partir de las discrepancias cometidas sobre este conjunto. Para ello se puede utilizar el principio de inducción de minimización empírica del riesgo (*[Vapnik, 1982]*), que no requiere la estimación de las fdp.

2.2 Principio de minimización empírica del riesgo

La idea básica de este método es utilizar el conjunto de entrenamiento *{(**x**[1],**d**[1]),(**x**[2],**d**[2]), ..., (**x**[N],**d**[N])}* para construir la función empírica de riesgo (*[Vapnik, 1982]*):

$$R_{emp}(h,\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} L(\mathbf{d}[i], f_h(\mathbf{x}[i],\mathbf{w}))$$

Ecuación 2.7

que no depende explícitamente de la fdp desconocida $p(\mathbf{x},\mathbf{d})$. En contraposición con la función de riesgo original ($R(h,\mathbf{w})$), la función de riesgo empírica ($R_{emp}(h,\mathbf{w})$) puede ser minimizada, al menos en teoría, respecto del vector de parámetros $\mathbf{w}$ para cada valor de h (en lo que sigue supondremos dado el valor de h, por lo que obviaremos su notación).

Sean $\mathbf{w}_{emp}$ y $f(\mathbf{x}, \mathbf{w}_{emp})$ el vector de parámetros y la relación funcional correspondiente que minimizan $R_{emp}(\mathbf{w})$. De forma análoga, sean $\mathbf{w}_0$ y $f(\mathbf{x}, \mathbf{w}_0)$ el vector de parámetros y la relación funcional correspondiente que minimizan $R(\mathbf{w})$.

Los dos vectores $\mathbf{w}_{emp}$ y $\mathbf{w}_0$ pertenecen al espacio de parámetros $\mathbf{W}$. Hemos de considerar ahora bajo qué condiciones la solución aproximada $f(\mathbf{x}, \mathbf{w}_{emp})$ "es parecida" a la solución deseada $f(\mathbf{x}, \mathbf{w}_0)$, entendiendo este parecido como la proximidad de $R(\mathbf{w}_{emp})$ y $R(\mathbf{w}_0)$.

Para cada vector fijo $\mathbf{w}=\mathbf{w}^*$, la función de riesgo $R(\mathbf{w}^*)$:

$$R(\mathbf{w}^*) = \int L(\mathbf{d}, f(\mathbf{x},\mathbf{w}^*))\; d\big(p(\mathbf{x},\mathbf{d})\big)$$

Ecuación 2.8

determina la esperanza matemática de la variable aleatoria definida por:

$$z_{\mathbf{w}^*} = L(\mathbf{d}, f(\mathbf{x},\mathbf{w}^*))$$

Ecuación 2.9

Por otro lado, la función empírica de riesgo $R_{emp}(\mathbf{w}^*)$ es la media aritmética de un conjunto de muestras de la variable aleatoria $z_{\mathbf{w}^*}$. De acuerdo con los teoremas de la teoría de probabilidad, cuando el tamaño N del conjunto de muestras tiende a infinito, la media aritmética de estas muestras converge a la esperanza matemática de

la variable aleatoria asociada. Esta observación justifica teóricamente el uso de la función empírica de riesgo $R_{emp}(\boldsymbol{w})$, en lugar de la función de riesgo $R(\boldsymbol{w})$:

$$\lim_{N\to\infty} R_{emp}(\boldsymbol{w}^*) = R(\boldsymbol{w}^*)$$

Ecuación 2.10

Hay que señalar sin embargo que el hecho de que la media aritmética de z_{w^*} converja a su valor esperado no prueba que el vector $\boldsymbol{w}_{emp}$, que minimiza $R_{emp}(\boldsymbol{w})$, minimice también $R(\boldsymbol{w})$.

Para asegurar la convergencia de $\boldsymbol{w}_{emp}$ a $\boldsymbol{w}_0$, hemos de imponer ciertas condiciones a $R_{emp}(\boldsymbol{w})$, dando lugar al principio de minimización empírica de riesgo:

> "En lugar de la función de riesgo $R(\boldsymbol{w})$, constrúyase la función empírica de riesgo:
>
> $$R_{emp}(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N} L(\boldsymbol{d}[i], f(\boldsymbol{x}[i], \boldsymbol{w}))$$
>
> *Ecuación 2.11*
>
> en base al conjunto de muestras independientes e idénticamente distribuidas $\{(\boldsymbol{x}[1], \boldsymbol{d}[1]), (\boldsymbol{x}[2], \boldsymbol{d}[2]), \ldots, (\boldsymbol{x}[N], \boldsymbol{d}[N])\}$.
>
> Sea $\boldsymbol{w}_{emp}$ el vector de parámetros que minimiza la función empírica de riesgo $R_{emp}(\boldsymbol{w})$ sobre el espacio de parámetros $\boldsymbol{W}$.
>
> Entonces $R(\boldsymbol{w}_{emp})$ converge en probabilidad al mínimo valor de $R(\boldsymbol{w})$ cuando el tamaño N del conjunto de muestras tiende a infinito, si $R_{emp}(\boldsymbol{w})$ converge uniformemente a $R(\boldsymbol{w})$."

Diremos que $R_{emp}(\boldsymbol{w})$ converge uniformemente a $R(\boldsymbol{w})$ si para todo $\boldsymbol{w}\in\boldsymbol{W}$ y para todo real $\varepsilon>0$ se cumple la condición:

$$\lim_{N\to\infty} prob\left(\sup_{\boldsymbol{w}\in\boldsymbol{W}} \left| R(\boldsymbol{w}) - R_{emp}(\boldsymbol{w}) \right| > \varepsilon \right) = 0$$

Ecuación 2.12

Esta condición es una condición necesaria y suficiente para la consistencia del principio de minimización empírica del riesgo.

La dimensión VC

La teoría de convergencia uniforme de $R_{emp}(\boldsymbol{w})$ a $R(\boldsymbol{w})$ establece límites en el ratio de convergencia basados en la llamada "Dimensión de Vapnik-Chervonenkis", que notaremos dimensión VC (*[Vapnik and Chervonenkis, 1971]*). La dimensión VC es una medida de la capacidad de representación de la familia de funciones realizables por el aproximador funcional.

Para simplificar esta discusión vamos a considerar el caso de clasificación binaria de patrones, en el que la respuesta (unidimensional) deseada toma los valores binarios {0,1}.

Sea F la familia de dicotomías (o funciones de clasificación binarias) que pueden ser realizadas por el aproximador funcional:

$$F = \left\{ f(\boldsymbol{x}, \boldsymbol{w}) / \boldsymbol{w} \in \boldsymbol{W},\ \ f: \Re^n \rightarrow \{0,1\} \right\}$$

Ecuación 2.13

Sea S_x el conjunto de las N muestras de entrenamiento proyectadas en el espacio n-dimensional de entrada $\boldsymbol{X}$:

$$S_x = \left\{ \boldsymbol{x}[i] \in \boldsymbol{X},\ \ i = 1,\ldots, N \right\}$$

Ecuación 2.14

Una dicotomía llevada a cabo por el aproximador funcional divide el conjunto S_x en dos subconjuntos S_0 y S_1, tales que:

$$si\ \ \boldsymbol{x} \in S_x\ \ entonces\ f(\boldsymbol{x}, \boldsymbol{w}) = \begin{cases} 0\ si\ x \in S_0 \\ 1\ si\ x \in S_1 \end{cases}$$

Ecuación 2.15

Sea $\Delta_F(S_x)$ el número de dicotomías distintas realizables por el aproximador funcional, y $\Delta_F(l)$ el máximo de $\Delta_F(S_x)$ sobre todo S_x con $|S_x|=l$, donde $|S_x|$ es la cardinalidad de S_x (para S_x conjunto finito, $|S_x|$ es el número de elementos de S_x).

Se dice que el conjunto S_x queda "desagregado" (del inglés "shattered") por F si $\Delta_F(S_x)=2^{|S|}$, esto es, si todas las dicotomías posibles de S_x pueden ser inducidas por funciones de F.

Podemos entonces definir la dimensión VC como (*[Blumer et al, 1989], [Baum and Haussler, 1989], [Vapnik and Chervonenkis, 1971]*):

> "La dimensión VC de la familia de dicotomías F es la máxima cardinalidad de cualquier conjunto de puntos S que pueda ser desmenuzado por F".

En otras palabras, la dimensión VC de F es el máximo N tal que $\Delta_F(N)=2^N$. La dimensión VC de la familia de funciones $\{f(\boldsymbol{x},\boldsymbol{w}),\ \boldsymbol{w}\in\boldsymbol{W}\}$ es el máximo número de ejemplos de entrenamiento que pueden ser aprendidos sin error para todos los posibles etiquetados binarios de los mismos.

Ratios de convergencia uniforme

En el caso de clasificación binaria de patrones, la función de discrepancia puede tomar sólo dos valores:

$$L(d, f(\boldsymbol{x},\boldsymbol{w})) = \begin{cases} 0 & si\ f(\boldsymbol{x},\boldsymbol{w}) = d \\ 1 & en\ otro\ caso \end{cases}$$

Ecuación 2.16

Bajo estas condiciones la función de riesgo $R(\boldsymbol{w})$ y la función empírica de riesgo $R_{emp}(\boldsymbol{w})$ toman las siguientes interpretaciones:

- $R(\boldsymbol{w})$ es la probabilidad media de error de clasificación
- $R_{emp}(\boldsymbol{w})$ es el error de entrenamiento, o frecuencia relativa de error de clasificación sobre el conjunto de entrenamiento.

Según la ley de los grandes números (*[Leon-García, 1994]*), la frecuencia empírica de ocurrencia de un suceso converge a la probabilidad del mismo cuando el número de ensayos (independientes e idénticamente distribuidos) tienden a infinito.

En nuestro caso: para todo $\boldsymbol{w}\in\boldsymbol{W}$ y para todo real $\varepsilon>0$ (*[Vapnik, 1982]*):

$$\lim_{N\to\infty} prob\left(\left|R(\boldsymbol{w}) - R_{emp}(\boldsymbol{w})\right| > \varepsilon\right) = 0$$

Ecuación 2.17

siendo N el tamaño del conjunto de entrenamiento.
Hay que señalar nuevamente que la propiedad expresada por la Ecuación 2.17 no implica que el vector de parámetros $\boldsymbol{w}_{emp}$ que minimiza el error de entrenamiento $R_{emp}(\boldsymbol{w})$ vaya también a minimizar la probabilidad media de error de clasificación $R(\boldsymbol{w})$.

Para un conjunto de entrenamiento de tamaño N suficientemente grande, la proximidad requerida entre $R(\mathbf{w})$ y $R_{emp}(\mathbf{w})$ queda garantizada si se cumple una condición aún más estricta que la anterior, y que requiere que para todo real $\varepsilon>0$ se cumpla:

$$\lim_{N\to\infty} \; prob\left(\sup_{\mathbf{w}\in\mathbf{W}} \left| R(\mathbf{w}) - R_{emp}(\mathbf{w}) \right| > \varepsilon \right) = 0$$

Ecuación 2.18

Cuando se cumple la condición expresada en la Ecuación 2.18, se habla de convergencia uniforme de la frecuencia relativa de errores de entrenamiento a su probabilidad media.

La dimensión VC proporciona una cota para el ratio de convergencia uniforme: dado un conjunto de funciones de clasificación $\{f(\mathbf{x},\mathbf{w}),\ \mathbf{w}\in\mathbf{W}\}$ con una dimensión VC h, se cumple:

$$prob\left(\sup_{\mathbf{w}\in\mathbf{W}} \left| R(\mathbf{w}) - R_{emp}(\mathbf{w}) \right| > \varepsilon \right) < \left(\frac{2eN}{h} \right)^{h} exp(-\varepsilon^2 N)$$

Ecuación 2.19

donde N es el tamaño del conjunto de entrenamiento y e es la base de los logaritmos naturales.

El objetivo es hacer que el segundo miembro de la Ecuación 2.19 tienda a cero cuando N tiende a infinito. El factor $exp(-\varepsilon^2 N)$ contribuye a este propósito de forma exponencial, mientras que $(2eN/h)^h$ es un factor de crecimiento que crece con N. Mientras este crecimiento no sea demasiado rápido, el segundo miembro de la ecuación tenderá a cero cuando N tiende a infinito. De hecho esta condición se cumple para todo valor finito de la dimensión VC (h). En otras palabras, un valor finito de h es condición necesaria y suficiente para la convergencia uniforme de la frecuencia relativa de errores de entrenamiento a su probabilidad media, y por tanto para la convergencia del minimizador del error de entrenamiento ($\mathbf{w}_{emp}$) al minimizador de la probabilidad media de error de clasificación.

Este resultado justifica la utilización de la función empírica de riesgo, pero es una justificación asintótica. Para conjuntos de entrenamiento de tamaño N finito, podemos extraer las siguientes conclusiones (*[Vapnik, 1982][Vapnik, 1992]*): dado un conjunto de funciones de clasificación $\{f(\mathbf{x},\mathbf{w}),\ \mathbf{w}\in\mathbf{W}\}$ con una dimensión VC h, se cumple, con probabilidad $(1-\alpha)$ y para todo $\mathbf{w}\in\mathbf{W}$:

1.- En general:

$$\left|R(\boldsymbol{w}) - R_{emp}(\boldsymbol{w})\right| < \varepsilon_1(N, h, \alpha, R_{emp})$$

Ecuación 2.20

siendo:

$$\varepsilon_1(N, h, \alpha, R_{emp}) = 2\ \varepsilon_0^2(N, h, \alpha)\left(1 + \sqrt{1 + \frac{R_{emp}(\boldsymbol{w})}{\varepsilon_0^2(N, h, \alpha)}}\right)$$

Ecuación 2.21

$$\varepsilon_0(N, h, \alpha) = \sqrt{\frac{h}{N}\left(ln\left(\frac{2N}{h}\right) + 1\right) - \frac{1}{N} ln\,\alpha}$$

Ecuación 2.22

2.- Para errores de entrenamiento pequeños, se obtiene la cota aproximada:

$$\left|R(\boldsymbol{w}) - R_{emp}(\boldsymbol{w})\right| < 4\ \varepsilon_0^2(N, h, \alpha)$$

Ecuación 2.23

que resulta ser una cota muy precisa en la mayoría de los casos de interés.

3.- Para errores de entrenamiento cercanos a la unidad, se obtiene la cota aproximada:

$$\left|R(\boldsymbol{w}) - R_{emp}(\boldsymbol{w})\right| < \varepsilon_0(N, h, \alpha)$$

Ecuación 2.24

La no coincidencia de los minimizadores de la función empírica de riesgo y de la función de riesgo global obliga a supervisar el proceso de aprendizaje mediante la estimación de la capacidad de generalización del aproximador. Para ello haremos uso de una técnica estadística conocida bajo el nombre de validación cruzada (*"cross-validation"*, *[Stone, 1974][Jansen et al., 1988]*), que estima la capacidad de generalización del aproximador utilizando un conjunto adicional de datos.

Siguiendo esta técnica, dividiremos el conjunto S para formar tres conjuntos disjuntos de muestras de la relación de entrada/salida, independientes e idénticamente distribuidos (una proporción típica de estos conjuntos es: 60%, 30% y 10%). El primero de ellos será el conjunto de entrenamiento:

$$S_{entr}=\{(\mathbf{x}[1],\mathbf{d}[1]), (\mathbf{x}[2],\mathbf{d}[2]), ..., (\mathbf{x}[N], \mathbf{d}[N])\}$$

Ecuación 2.25

que será utilizado para dirigir el ajuste del vector de parámetros $\boldsymbol{w}$ del aproximador durante la optimización paramétrica, mediante la minimización de la función de error de entrenamiento (R_{entr}).

El segundo de ellos es el conjunto de test:

$$S_{test}=\{(\mathbf{x'}[1],\mathbf{d'}[1]),(\mathbf{x'}[2],\mathbf{d'}[2]), ..., (\mathbf{x'}[M],\mathbf{d'}[M])\})$$

Ecuación 2.26

utilizado para estimar la capacidad de generalización de los aproximadores. Como este conjunto de datos no ha sido utilizado para dirigir la búsqueda de los parámetros del aproximador, se pueden utilizar estos datos "frescos" para estimar el comportamiento del aproximador frente a datos no conocidos. De esta forma, estimaremos la función de riesgo global mediante el error de test (R_{test}).

El tercero es el conjunto de validación (S_{valid}), utilizado para dar el visto bueno al óptimo aproximador hallado.

2.3 Optimización estructural

La solución propuesta en esta tesis descompone el problema del aprendizaje en dos optimizaciones parciales: la optimización estructural, que determina la estructura óptima del aproximador funcional (dada por el índice estructural h^*) y por tanto su capacidad de representación, y la optimización paramétrica, que determina para cada estructura considerada de índice h el óptimo vector de parámetros $\boldsymbol{w}^* \in \boldsymbol{W}_h$.

De esta forma el optimizador estructural irá proponiendo una serie de estructuras candidatas según una estrategia determinada, encargará la evaluación de estos aproximadores al optimizador paramétrico, y detendrá la optimización estructural cuando se cumplan ciertas condiciones de finalización.

La minimización estructural del riesgo tiene pues como objetivo determinar la estructura óptima del aproximador, en base a su capacidad de generalización. Como se ha visto en el apartado anterior, evaluaremos la capacidad de generalización del aproximador mediante el error de test (R_{test}).

Para establecer una estrategia de búsqueda de la estructura del aproximador, es necesario conocer cómo se comporta la función de riesgo frente a la complejidad estructural. Para ello haremos uso de los resultados obtenidos en el caso de clasificación binaria, y los extrapolaremos al caso más general de aproximación funcional.

En el caso de clasificación binaria, el error de entrenamiento (R_{entr}) es la frecuencia relativa de errores cometidos por el aproximador funcional sobre el conjunto de ejemplos utilizados para el entrenamiento (conjunto de entrenamiento).

De forma similar, el error de generalización o error de test (R_{test}) se define como la frecuencia relativa de errores cometidos por el aproximador funcional sobre un conjunto de ejemplos no utilizados para el ajuste del aproximador, que denominaremos conjunto de test.

Sean R_{entr} y R_{test} los errores de entrenamiento y de test respectivamente, suponiendo que los conjuntos de entrenamiento y de test han sido tomados según la misma fdp. Sea h la dimensión VC de la familia de funciones de clasificación $\{f(\boldsymbol{x},\boldsymbol{w}),\ \boldsymbol{w} \in \boldsymbol{W}\}$ realizables por el aproximador funcional, respecto del espacio de entrada $\boldsymbol{X}$.

Podemos entonces establecer (retomando la Ecuación 2.20) que con probabilidad $(1-\alpha)$, para un número de ejemplos de entrenamiento $N>h$, y simultáneamente para

todas las funciones de clasificación $f(\mathbf{x},\mathbf{w})$, el error de generalización se encuentra en el intervalo:

$$\left|R_{test}(\mathbf{w}) - R_{entr}(\mathbf{w})\right| < \varepsilon_1(N, h, \alpha, \nu)$$

Ecuación 2.27

donde el intervalo de confianza ε_1 ha quedado definido en la Ecuación 2.21.

Para un número fijo de ejemplos de entrenamiento, el mínimo error de entrenamiento decrece monótonamente según aumenta la dimensión VC, mientras que el intervalo de confianza ε_1 crece de forma monótona. La cota superior del error de generalización, a la que notaremos $R_{máx}$, pasa pues por un mínimo global, como queda representado en la Figura 2.2:

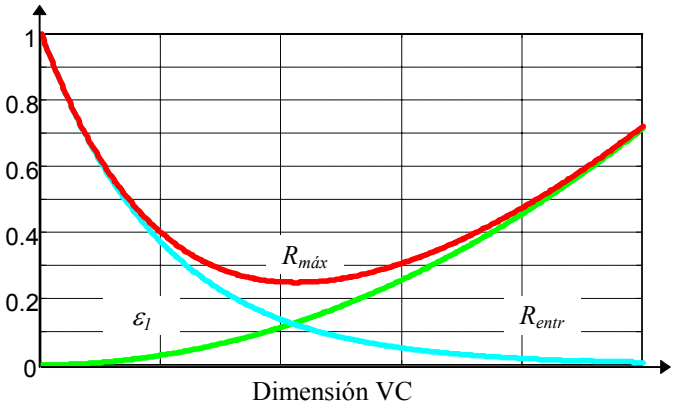

Figura 2.2: Cota superior del errror de generalización en función de la dimensión VC

Antes de que la cota superior del error de generalización alcance su mínimo, el problema del aprendizaje está sobredeterminado en el sentido que la capacidad de representación del aproximador funcional es demasiado baja dada la complejidad del conjunto de entrenamiento. Una vez sobrepasado ese mínimo, el problema estará sobreparametrizado en el sentido que la capacidad del aproximador funcional es demasiado alta para la complejidad del conjunto de entrenamiento.

De este resultado se desprende una importante conclusión que condicionará toda estrategia de aprendizaje supervisado:

> "A medida que la capacidad de representación funcional del aproximador funcional va creciendo, la capacidad de generalización del mencionado sistema, ajustado sobre el mismo conjunto de entrenamiento, se hace más incierta."

Este resultado nos lleva a establecer como estrategia de optimización estructural el tomar como punto de partida estructuras sencillas, e ir aumentando la complejidad del aproximador hasta alcanzar un mínimo en su capacidad de generalización.

Para ello, formaremos una familia anidada de funciones de aproximación:

$$\{ \mathbf{y} = f_h(\mathbf{x},\mathbf{w}) / \mathbf{x} \in \Re^n,\ \mathbf{y} \in \Re^m,\ w \in \mathbf{W}_h\ ,\ h=1,...,H \}$$

de tal forma que:

$$f_1 \subset f_2 \subset ... \subset f_H$$

donde el símbolo $\subset$ ha de interpretarse como "contenido en", en el sentido que por ejemplo la función f_2 es capaz de realizar todas las transformaciones de entrada/salida realizables por f_1. De esta forma obtenemos una ordenación creciente con h de la capacidad de representación de los aproximadores.

El optimizador estructural se limita a proponer como estructuras candidatas la secuencia de funciones $f_1, f_2, ...,$ hasta detectar que el error de test devuelto por el optimizador paramétrico ha alcanzado un mínimo o se ha estabilizado. De esta forma se trata de evitar la sobreparametrización del aproximador, alcanzando el mínimo de la cota de riesgo garantizado mostrado en la Figura 2.2.

Existen varias formas de construir familias anidadas de aproximadores, controlando su capacidad de representación. Entre las más habituales cabe destacar: (*[Vapnik, 1992] [Guyon et al., 1992]*):

- **Control directo de la estructura:** en muchos casos, la propia estructura del aproximador permite definir un orden del mismo ligado directamente a su capacidad de representación. Por ejemplo, si tratamos de aproximar una onda periódica por una serie de senos y cosenos de frecuencias múltiplos de la fundamental (serie de Fourier), el número de armónicos considerado controlará directamente la capacidad de representación del aproximador.

- **Control basado en el debilitamiento de los parámetros ("weight decay"):** el aproximador funcional tiene una estructura fija y el control de su capacidad se ejerce variando la norma euclídea del vector de parámetros $\boldsymbol{w}$. Se considera la familia de clasificadores definida por:

$$\left\{ f_h(\boldsymbol{x}, \boldsymbol{w}),\ \ \|\boldsymbol{w}\| < c_h \right\} \ \ con \qquad h = 1,\ldots, H$$

$$donde \ \ \|\boldsymbol{w}\| = \sum_j w_j^2 \quad y \quad c_1 < c_2 <\ldots< c_H$$

Ecuación 2.28

La minimización del riesgo empírico de cada función f_h, se realiza minimizando la función de coste aumentada:

$$R_h\left(\boldsymbol{w}, \lambda_h\right) = \frac{1}{N} \sum_{i=1}^{N} L\left(\boldsymbol{d}[i], f_h(\boldsymbol{x}[i], \boldsymbol{w})\right) + \lambda_h \ \|\boldsymbol{w}\|^2$$

Ecuación 2.29

donde la función de discrepancia L suele ser una función de coste cuadrática y λ_h es el parámetro de regularización.

La secuencia $c_1<c_2<...<c_H$ se obtiene utilizando una secuencia decreciente de parámetros de regularización $\lambda_1>\lambda_2>...>\lambda_H$.

Este método está directamente relacionado con la teoría de la regularización (*[Moody, 1992]*).

- **Preprocesamiento:** otro forma de controlar la capacidad del aproximador funcional es reducir la dimensión del espacio de entrada, lo que tiene el efecto de reducir el número necesario de parámetros del sistema.

Esta reducción de la dimensión suele realizarse mediante el uso de un método de extracción de características tal como el análisis de componentes principales (PCA, del inglés *"Principal Component Analysis"*) (*[Johnson and Wichern, 1982]*). El PCA está basado en el estudio de los autovectores de la matriz de correlación de los vectores de entrada del conjunto de entrenamiento, de dimensión n, que son aproximados por una combinación lineal de los h autovectores asociados a los mayores autovalores de la mencionada matriz, con $h<n$. La estructura anidada de familias de aproximadores se obtiene variando la dimensión reducida h.

2.4 Optimización paramétrica

Una vez propuesta por el minimizador estructural una estructura determinada del aproximador, sea $\boldsymbol{y}=f(\boldsymbol{x},\boldsymbol{w})$, la optimización paramétrica tiene como objetivo hallar el vector de parámetros óptimo $\boldsymbol{w}^*$, que mejor asemeja la función aproximadora f a la función subyacente g. Como en el caso de minimización estructural, la función objetivo utilizada es la capacidad de generalización del aproximador, estimada según el método de validación cruzada.

Cuando el optimizador estructural pide al optimizador paramétrico que evalúe una estructura determinada, éste aplica un método iterativo para minimizar el error de entrenamiento:

$$R_{entr.}(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N} L(\boldsymbol{d}[i], f(\boldsymbol{x}[i], \boldsymbol{w}))$$

Ecuación 2.30

y en cada iteración del proceso de optimización paramétrica se evalúa la capacidad de generalización del aproximador mediante el error de test dado por:

$$R_{test}(\boldsymbol{w}) = \frac{1}{M}\sum_{i=1}^{M} L(\boldsymbol{d}'[i], f(\boldsymbol{x}'[i], \boldsymbol{w}))$$

Ecuación 2.31

El optimizador paramétrico detiene el proceso de optimización paramétrica cuando el error de test ha alcanzado un mínimo o se ha estabilizado, y devuelve como vector de parámetros óptimo aquél que minimiza R_{test} (que no tiene por qué coincidir con el vector de parámetros que minimiza el error de entrenamiento).

2.4.1 Efectos negativos del sobre-entrenamiento

El proceso de minimización del error de entrenamiento puede ser visto como un proceso de ajuste de curvas. Bajo esta perspectiva, la capacidad de generalización se traduce en una buena interpolación de los datos de entrenamiento. Esto significa que el aproximador funcional será capaz de reproducir correctamente salidas del proceso correspodientes a entradas no contempladas en el conjunto de entrenamiento. Para ello es necesario suponer que la función $g(\boldsymbol{x})$ es una función "suave", de tal forma que la interpolación de los datos de entrenamiento sea una buena aproximación de la función subyacente.

Cuando el aproximador funcional tiene una capacidad de representación sobrada, es decir, cuando el problema está sobreparametrizado, la minimización del error de entrenamiento puede acabar siendo un proceso de memorización en el que la capacidad de generalización se va degenerando poco a poco, dando lugar al fenómeno conocido como sobre-entrenamiento ("overtraining"). Este fenómeno es particularmente habitual cuando las muestras del conjunto de entrenamiento son ruidosas, es decir, cuando el vector de salidas deseadas viene dado por $\boldsymbol{d}[i]=g(\boldsymbol{x}[i])+\boldsymbol{\varepsilon}[i]$, donde $\boldsymbol{\varepsilon}[i]$ es un proceso de ruido blanco. Esta situación es muy común en aplicaciones prácticas, donde los procesos de muestreo no son perfectos. Suele ser habitual también que la salida del proceso se vea ligeramente influenciada por toda una combinación de variables aleatorias no medidas, lo que suele traducirse en una señal ruidosa superpuesta en la salida.

Bajo estas condiciones, una buena aproximación funcional de $g(\boldsymbol{x})$ será capaz de filtrar el ruido superpuesto a los datos de entrenamiento, pero si el aproximador tiene la capacidad suficiente y se prolonga el proceso de minimización del error de entrenamiento, el aproximador acabará memorizando también el ruido.

Para ilustrar este fenómeno tomemos como ejemplo la aproximación funcional de la función $g(x)=exp(-x)$, en el intervalo $x\in[0;4]$. Para medir la capacidad de generalización del aproximador $y=f(x,\boldsymbol{w})$ a ajustar, generaremos un conjunto de test compuesto por $M=400$ muestras de la forma:

$$S_{test} = \{(x'[i],d'[i]) / i=0,1,...399 \ ; \ x'[i]=i/100 \ ; \ d'[i]=exp(-x'[i])\}$$

Ecuación 2.32

y utilizaremos como medida inversa de la capacidad de generalización del aproximador, el error cuadrático medio de test:

$$R_{test}(\boldsymbol{w}) = \frac{1}{M}\sum_{i=1}^{M}(d'[i]-f(x'[i],\boldsymbol{w}))^2$$

Ecuación 2.33

Como conjunto de entrenamiento utilizado para el ajuste del vector de parámetros $\boldsymbol{w}$, generaremos $N=20$ muestras ruidosas de la relación $g(x)$, según:

$$S_{entr} = \{(x[i],d[i]) / i=0,1,...19; \ x[i]=i/5; \ d[i]=exp(-x[i])+\varepsilon[i] \ ; \ \varepsilon[i]\in N(0, \sigma=0.1)\}$$

Ecuación 2.34

siendo la función de error a minimizar durante el ajuste de los parámetros $\boldsymbol{w}$, el error cuadrático medio de entrenamiento:

$$R_{entr}(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N}(d[i] - f(x[i], \boldsymbol{w}))^2$$

Ecuación 2.35

La minimización paramétrica del vector $\boldsymbol{w}$ consiste en aplicar un algoritmo iterativo de optimización que vaya modificando el vector de parámetros $\boldsymbol{w}$ de tal forma que R_{entr} vaya disminuyendo de forma monótona. Tras cada modificación de parámetros se evalúa la capacidad de generalización del aproximador resultante, a través de R_{test}. La evolución de estos errores durante 200 ciclos de minimización de R_{entr}, utilizando una red neuronal como aproximador, queda mostrada en la Figura 2.3:

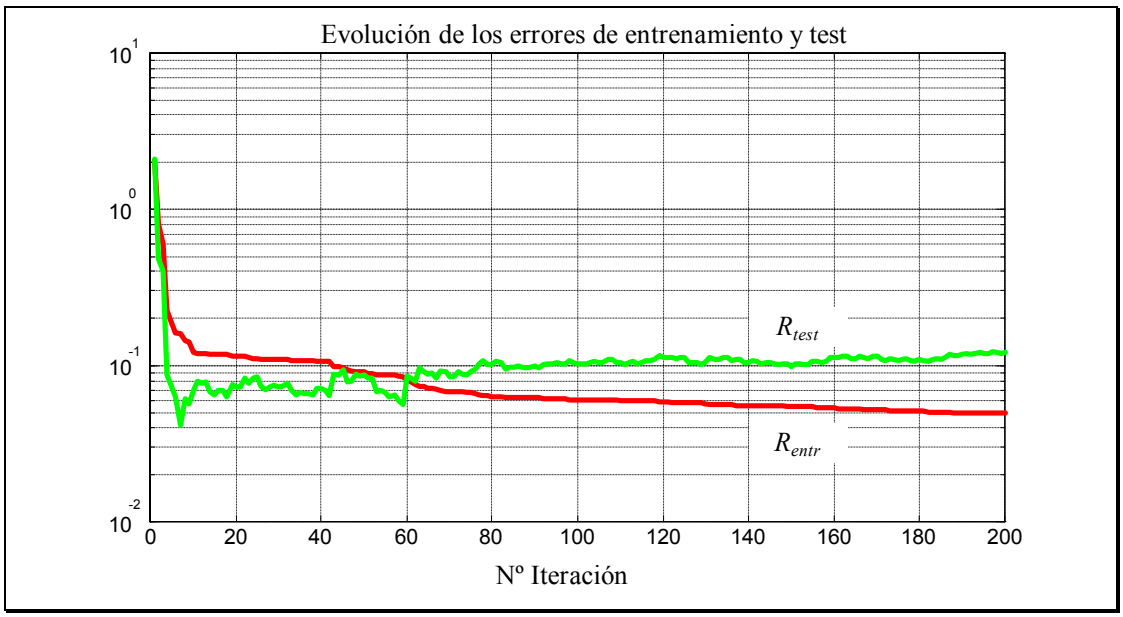

Figura 2.3: Efecto del sobre-entrenamiento con ejemplos ruidosos: evolución de los errores de entrenamiento y test durante la minimización de R_{entr} . Ejemplo exponencial.

Como cabía esperar, R_{entr} decrece de forma monótona según avanza su minimización. Por el contrario, R_{test} tiene un comportamiento no monótono, presentando un mínimo al cabo de 7 iteraciones. Tras superar este mínimo, el error de entrenamiento sigue disminuyendo, mientras que el de test comienza a aumentar en promedio. Para ilustrar las causas de este comportamiento comparemos las estimaciones del conjunto de test tras 20 y 200 iteraciones:

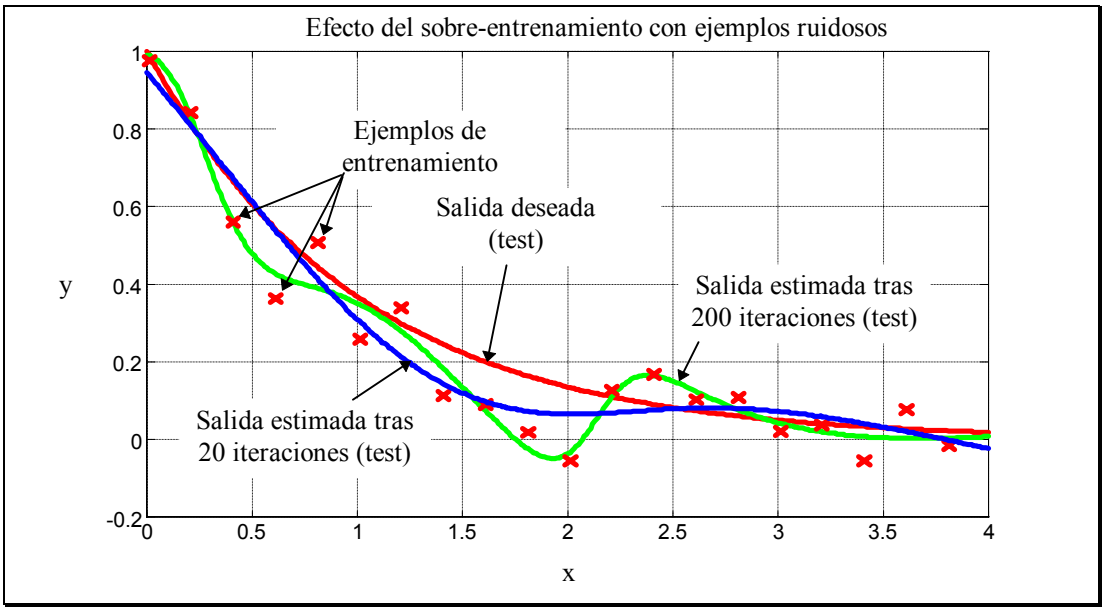

Figura 2.4:Efecto del sobre-entrenamiento con ejemplos ruidosos: estimación del conjunto de test tras 20 y 200 iteraciones de minimización de R_{entr}. Ejemplo exponencial

Como puede apreciarse en la Figura 2.4, la diferencia fundamental entre las estimaciones realizadas tras 20 y 200 ciclos de minimización de R_{entr} son debidas a la capacidad sobrada de representación del aproximador, que le permite memorizar el ruido de la serie de entrenamiento. Tras 20 iteraciones de optimización paramétrica, la interpolación de los datos de entrenamiento realizada por el aproximador resulta ser una función "suave" que consigue filtrar en parte el ruido de la serie de entrenamiento original. A medida que avanza la minimización de R_{entr}, como el aproximador tiene capacidad suficiente, la función interpoladora se va deformando poco a poco atraída por los datos ruidosos de entrenamiento. Este proceso podría desembocar en una función de interpolación que ajustase a la perfección todos los ejemplos de entrenamiento, a costa de introducir ondulaciones que empeorasen cada vez más la capacidad de generalización del aproximador.

Para impedir este fenómeno de sobre-entrenamiento, utilizaremos nuevamente el método de validación cruzada, evaluando en cada paso de la minimización de R_{entr} el error de test R_{test}. Este método permite por un lado guardar en memoria el óptimo vector de parámetros $\boldsymbol{w}^*$ (que es aquél que minimiza R_{test}) hallado a lo largo de la minimización de R_{test}. Por otro lado permite establecer criterios de finalización de la minimización paramétrica, basados en esta medida de la capacidad de generalización del aproximador.

2.4.2 Criterios de finalización de la optimización paramétrica

Sean $R_{entr}[k]$ y $R_{test}[k]$ los errores de entrenamiento y test cometidos en la iteración nºk de la optimización paramétrica. Los criterios de finalización de la optimización paramétrica propuestos en esta tesis para evitar el fenómeno de sobre-entrenamiento son los siguientes:

- **Número mínimo de iteraciones ($K_{mín}$):** Se establece un número mínimo de iteraciones que han de realizarse sin importar la evolución de R_{test}. El objetivo de este periodo de la optimización es dar tiempo a la búsqueda paramétrica para que sitúe al vector de parámetros en regiones aceptables. Esto evita detener la optimización de forma prematura, pasando por alto los extraños comportamientos de R_{test} que suelen darse al comienzo de esta etapa. El establecimiento del número mínimo de iteraciones dependerá de la eficacia del algoritmo de optimización utilizado y de la complejidad del problema a resolver. Para algoritmos de optimización como los quasi-Newton de baja memoria, un número razonable es 10.

- **Número máximo de iteraciones ($K_{máx}$):** Se establece un número máximo de iteraciones para asegurar una duración aceptable del proceso de minimización. El establecimiento de este número máximo de iteraciones depende también de la eficacia del algoritmo de optimización utilizado y de la complejidad del problema a resolver. Para algoritmos de optimización como los quasi-Newton de baja memoria, un número razonable es 400.

- **Cota máxima de la tendencia del error de test ($\beta_{máx}$):** En cada iteración se evalúa la tendencia del error de test para detener la optimización paramétrica si este valor supera una cota preestablecida. Para medir la tendencia del error de test en la iteración k, se define un tamaño de ventana V en el que se evalúa la pendiente de la recta de regresión de los V logaritmos de R_{test} correspondientes a las iteraciones k, k-1, ..., k-V+1. Al haber tomado logaritmos, esta tendencia se convierte en una medida del factor de crecimiento del error de test.

La pendiente β de la recta de regresión $y=\alpha+\beta x$, de V muestras $\{(x[i],y[i])$, con $i=0,1,...,V-1\}$ es una estimación "filtrada" de la derivada media de y respecto de x, en el intervalo de x considerado. Se tiene (*[Peña, 1986]*):

$$\beta = \frac{cov(x,y)}{s_x^2} = \frac{\sum_{i=0}^{V-1}(y[i]-m_y)(x[i]-m_x)}{\sum_{i=0}^{V-1}(x[i]-m_x)^2}$$

Ecuación 2.36

siendo *cov(x,y)* la covarianza de las series *x* e *y*, s_x^2 la varianza de *x*, y m_x y m_y las medias de *x* e *y*.

Si consideramos *x(i)=i*, con *i=0,1,..., V-1* (*V* muestras igualmente espaciadas en el eje de abcisas), resulta entonces:

$$\beta = \frac{12}{V(V^2-1)} \sum_{i=0}^{V-1} \left(i - \frac{V-1}{2} \right) \left(y[i] - m_y \right)$$

Ecuación 2.37

Sea $LR_{test}[k] = log_{10}(R_{test}[k])$. Tomaremos como medida de la tendencia del error de test, en la iteración nº*k* de la optimización paramétrica, el valor:

$$\beta[k] = \frac{12}{V(V^2-1)} \sum_{i=0}^{V-1} \left(i - \frac{V-1}{2} \right) \left(LR_{test}[k-V+i+1] - m_{LR_{test}} \right)$$

Ecuación 2.38

Valores negativos de $\beta[k]$ indicarán un decrecimiento en el error de test, y por tanto, la conveniencia de proseguir la optimización paramétrica. Al comienzo de la optimización, la tendencia de R_{test} tomará valores negativos significativos, que poco a poco se irán haciendo menos negativos, hasta llegar a cero o incluso hacerse positivos. Valores positivos de $\beta[k]$ indican un crecimiento medio del error de test, y por tanto, el momento de dar por finalizada la optimización paramétrica. Por ello detendremos la optimización si se cumple:

$$\beta[k] > \beta_{max}$$

Ecuación 2.39

Valores típicos a utilizar con métodos de optimización como los quasi-Newton de baja memoria son *V=10* y $\beta_{máx}$*=-0.001*.

Una vez alcanzada la cota $\beta_{máx}$, se detendría la optimización paramétrica y se retomaría el vector de parámetros $\boldsymbol{w}^*$ que ha minimizado R_{test} durante esta búsqueda.

Retomando el ejemplo utilizado para ilustrar el fenómeno de sobre-entrenamiento, la evolución de la tendencia del error de test con *V*=10 durante las 50 primeras iteraciones de la optimización paramétrica queda:

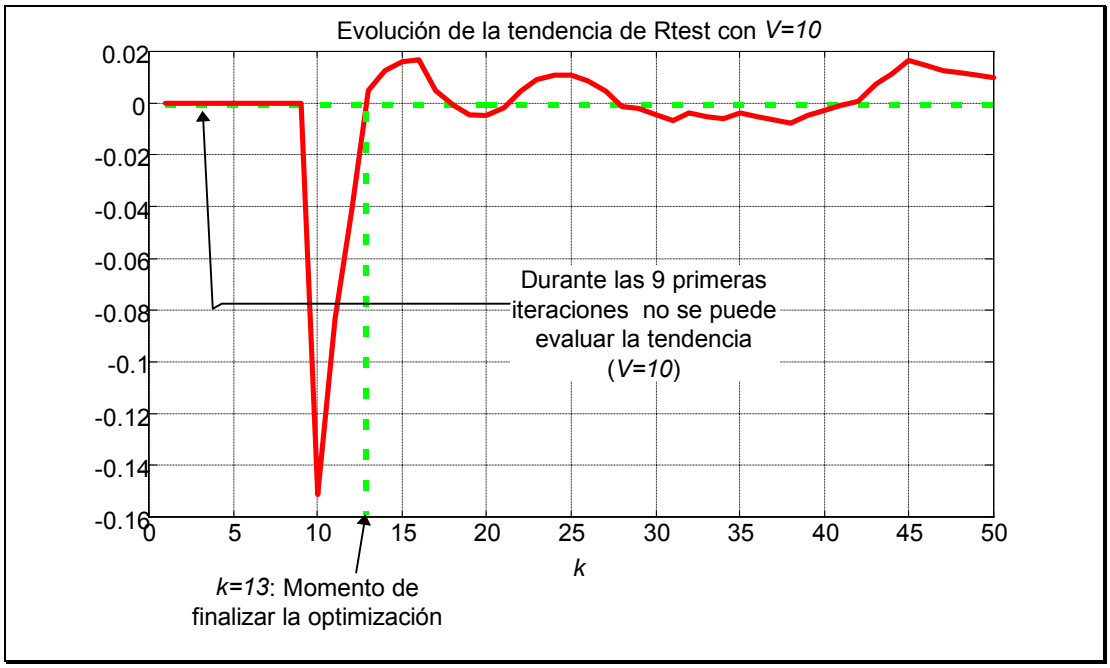

Figura 2.5: Evolución de la tendencia del error de test con V=10 durante las 50 primeras iteraciones de la optimización paramétrica. Ejemplo exponencial.

de tal forma que detendríamos la optimización paramétrica a las 13 iteraciones, y retomaríamos el vector de parámetros de la séptima iteración, que es el que minimizaba R_{test}.

Un inconveniente de este criterio de finalización es que puede detener la optimización paramétrica de forma prematura, si se diese el caso que el error de test llega a estabilizarse durante un número de iteraciones superior a V, anulando su tendencia, antes de volver a disminuir. La práctica muestra que son raras estas situaciones, y que en la gran mayoría de los casos, la evolución del error de test tiene un comportamiento como el mostrado en la Figura 2.3 (*[Hecht-Nielsen, 1990], [Sjöberg, 1995]*).

Aumentando el tamaño V de la ventana se previene la finalización prematura, pero se prolonga la etapa de minimización paramétrica.

Además de estos criterios de finalización, pueden establecerse cotas mínimas en el error de entrenamiento, en el error de test, y en el módulo del gradiente del error de entrenamiento respecto de los parámetros (*[Kramer et al., 1989]*). En particular será necesario detener la optimización paramétrica si se alcanza un mínimo local (o global) durante la minimización del error de entrenamiento (correspondiente a un gradiente nulo). En este caso será imposible seguir minimizando el error de

entrenamiento con algoritmos de búsqueda local, como los utilizados (para más detalles consultar el apartado siguiente).

2.4.3 Minimización del error de entrenamiento

Como hemos visto en los apartados anteriores, la optimización paramétrica requiere de un algoritmo iterativo que minimice el error de entrenamiento R_{entr} respecto del vector de parámetros $\boldsymbol{w} \in \Re^q$. Tras cada iteración de este proceso de búsqueda se evaluará el error de test R_{test}, con el fin de detener la optimización y retomar el óptimo vector de parámetros hallado, $\boldsymbol{w}^*$.

La función de coste a minimizar en esta etapa es el error de entrenamiento R_{entr}, cometido por el aproximador no lineal $\boldsymbol{y}=f(\boldsymbol{x},\boldsymbol{w})$ (con $\boldsymbol{x} \in \Re^n$, $\boldsymbol{y} \in \Re^m$, $\boldsymbol{w} \in \Re^q$), sobre el conjunto de ejemplos $S_{entr}=\{(\boldsymbol{x}[1], \boldsymbol{d}[1]),(\boldsymbol{x}[2], \boldsymbol{d}[2]), ..., (\boldsymbol{x}[N], \boldsymbol{d}[N])\}$. Como medida de discrepancia utilizaremos el error cuadrático medio de tal forma que el error de entrenamiento quedará definido por:

$$R(\boldsymbol{w}) = R_{entr}(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{m}\left(d_j[i] - y_j[i]\right)^2$$

Ecuación 2.40

siendo $\boldsymbol{y}[i]=f(\boldsymbol{x}[i],\boldsymbol{w})$ la salida estimada por el aproximador funcional para el ejemplo de entrenamiento i.

El problema de la minimización de $R(\boldsymbol{w})$, respecto del vector de parámetros $\boldsymbol{w}$ en $\Re^q$ es un problema de optimización no lineal sin restricciones (hemos tomado como espacio de parámetros $\boldsymbol{W}=\Re^q$), cuya resolución dependerá de las características de la función de coste $R: \Re^q \rightarrow \Re$, y de la información que de ella se pueda conseguir.

Pasemos a continuación a introducir las distintas estrategias de optimización no lineal sin restricciones (apartado a), para dar posteriormente paso a los algoritmos "clásicos" de optimización basados en el gradiente (apartado b) y al método de optimización más popular en el campo conexionista: la Regla Delta (apartado c).

a) Optimización no lineal sin restricciones

La literatura ha tratado el tema de la optimización no lineal sin restricciones de una forma muy extensa, y sigue siendo el campo de trabajo de muchos investigadores. Como tratados de optimización especialmente relevantes podemos citar *[Luenberger,1984]* y *[Gill et al., 1981]*. Una buena introducción teórica y práctica a estas técnicas se encuentra contenida en *[Nocedal, 1992]*.

De una forma general podemos distinguir tres tipos de algoritmos:

- **Métodos de búsqueda directa**: Cuando la función R es una función no derivable, o cuando no resulta práctico calcular el gradiente de R respecto del vector de parámetros $\boldsymbol{w}$:

$$\nabla R = \frac{\partial R}{\partial \boldsymbol{w}}$$

Ecuación 2.41

la única información disponible durante la optimización es el valor de la propia función R para cada vector de parámetros $\boldsymbol{w}$. Por ello, los algoritmos de optimización aplicables en este caso se reducen a algoritmos de búsqueda directa como el método de "Hook and Jeeves" (*[Hooke and Jeeves, 1961]*), o los más recientes algoritmos genéticos (*[Goldberg, 1989]*).

Estos algoritmos son en general métodos muy robustos capaces de obtener soluciones aceptables con funciones objetivo de mal comportamiento. Cuando la dimensión p del espacio de búsqueda comienza a aumentar, su eficacia disminuye considerablemente ya que el número de evaluaciones de la función objetivo se dispara. Una estrategia interesante es utilizar uno de estos algoritmos en la primera etapa de la optimización para hallar un buen punto inicial.

Otra solución posible es estimar el gradiente ∇R a partir de la función R, y aplicar métodos de optimización basados en el gradiente. El problema de esta solución es nuevamente su ineficacia cuando la dimensión del espacio de búsqueda es elevada. Si para estimar el gradiente utilizamos la expresión (*[Gill et al., 1981]*):

$$\nabla R_i(\boldsymbol{w}) = \frac{\partial R(\boldsymbol{w})}{\partial w_i} \approx \frac{R(\boldsymbol{w} + \delta\ \boldsymbol{u}^i) - R(\boldsymbol{w})}{\delta} \quad ;\ \delta > 0$$

Ecuación 2.42

donde $\boldsymbol{u}^i$ es un vector q-dimensional cuya única componente no nula es su componente i, que es igual a la unidad, el número de evaluaciones de la función R, necesarias para estimar el gradiente ∇R en $\boldsymbol{w}$, será $(q+1)$.

Si por el contrario utilizamos una estimación más precisa del gradiente (*[Gill et al., 1981]*), dada por:

$$\nabla R_i(\boldsymbol{w}) \approx \frac{R(\boldsymbol{w}+\delta \boldsymbol{u}^i) - R(\boldsymbol{w}-\delta \boldsymbol{u}^i)}{2\delta} \quad ; \ \delta > 0$$

Ecuación 2.43

el número de evaluaciones de R necesario para estimar $\nabla R(\boldsymbol{w})$ será $2q$. La precisión requerida en la estimación del gradiente por los métodos basados en el gradiente suele ser elevada, sobre todo en los métodos del tipo quasi-Newton, como veremos en apartados posteriores.

- **Métodos basados en el gradiente**: estos métodos, además de utilizar la información suministrada por la evaluación de $R(\boldsymbol{w})$, utilizan la información dada por $\nabla R(\boldsymbol{w})$ para ir descendiendo por la hipersuperficie de error $R(\boldsymbol{w})$. Su formulación general queda expresada de forma iterativa por:

$$\boldsymbol{w}[k+1] = \boldsymbol{w}[k] + \alpha[k]\boldsymbol{D}[k]$$

Ecuación 2.44

Partiendo del vector de parámetros $\boldsymbol{w}[k]$, hallado en la iteración k, el vector de parámetros $\boldsymbol{w}[k+1]$ de la iteración $(k+1)$ se obtiene dando un paso de longitud $\alpha[k]$ en la dirección de búsqueda $\boldsymbol{D}[k]$.

El valor de la longitud del paso $\alpha[k]$ se obtendrá bien mediante la aplicación de un heurístico, bien como resultado de la minimización unidimensional de $R(\boldsymbol{w}[k]+\alpha[k]\boldsymbol{D}[k])$, respecto de $\alpha[k]$, una vez establecida la dirección de búsqueda $\boldsymbol{D}[k]$.

Los distintos métodos de optimización basados en el gradiente se diferencian en la forma de obtener la dirección de búsqueda $\boldsymbol{D}[k]$, a partir de la información de primer orden contenida en el gradiente $\nabla R\,[k]$, y en algunos casos, de la estimación de información de segundo orden contenida en la serie de gradientes $\{\nabla R\,[k]\}$.

Podemos destacar entre estos métodos:

- el método de descenso del gradiente:

$$\boldsymbol{D}[k] = - \nabla R\,[k]$$

Ecuación 2.45

- los métodos de gradiente conjugado, que tienen la forma general:

$$\boldsymbol{D}[k] = - \nabla R\,[k] + \gamma[k]\,\boldsymbol{D}[k-1]$$

Ecuación 2.46

- y los métodos quasi-Newton , cuya expresión general es:

$$\nabla R[k] = - \boldsymbol{B}[k]\ \nabla R[k]$$

Ecuación 2.47

siendo $\boldsymbol{B}$ una estimación del inverso de la matriz hessiana de R dada por:

$$\nabla^2 R = \left[\frac{\partial^2 R}{\partial w_i \partial w_j} \right]$$

Ecuación 2.48

Los métodos quasi-Newton tienen el inconveniente de requerir el almacenamiento de la matriz $\boldsymbol{B}$, que es de dimensión ($q \times q$). Para problemas de dimensiones elevadas, se han desarrollado métodos quasi-Newton de baja memoria (*[Gill et al., 1981], [Luenberger, 1984]*) que eliminan este problema. Un trabajo muy interesante en este sentido queda recogido en *[Liu & Nocedal, 1989]*, donde se presenta una versión de "memoria reducida" del método quasi-Newton BFGS (Broyden-Fletcher-Goldfarb-Shanno).

Los métodos basados en el gradiente son métodos de búsqueda local que reducen el problema global de minimización q-dimensional a sucesivas minimizaciones unidimensionales. Esto les permite tratar problemas de elevada dimensión de una forma eficaz, sin que se dispare el número de evaluaciones de la función R y de su gradiente $\boldsymbol{g}$. Además de ser algoritmos muy eficaces, las aportaciones de distintos investigadores han hecho de ellos algoritmos muy robustos. Estas razones les hacen ser la estrella de la optimización no lineal de funciones derivables, y serán los métodos utilizados en esta tesis.
En el apartado siguiente se presentará un estudio comparativo de una selección de algoritmos de optimización no lineal sin restricciones basados en el gradiente, pero

es importante tener en cuenta que existe todo un abanico de ellos como los no mencionados métodos de Gauss-Newton y de Levenberg-Marquadt (*[Gill et al., 1981]*.

- **Métodos basados en el hessiano:** estos métodos, además de utilizar el gradiente de *R*, utilizan la matriz hessiana ***H*** de *R*. Están encabezados por el método de Newton, cuyo algoritmo iterativo queda también definido por la Ecuación 2.44, siendo la dirección de búsqueda en este caso:

$$\boldsymbol{D}[k] = -\left(\nabla^2 R\,[k]\right)^{-1} \nabla R\,[k]$$

Ecuación 2.49

Este método, además de requerir el cálculo y almacenamiento de la matriz $\nabla^2 R$, requiere el cálculo de su inversa. Esta circunstancia les hace ser en general métodos poco robustos, aplicables a funciones objetivo con buen comportamiento y en espacios de búsqueda de dimensión reducida.

b) Métodos de optimización no lineal sin restricciones basados en el gradiente

Como vimos en el apartado anterior, podemos plantear el problema de la minimización del error de entrenamiento como un problema de optimización no lineal sin restricciones, en el que se trata de minimizar de forma iterativa la función $R(\boldsymbol{w})$, de la que además se conoce el gradiente $\nabla R\,(\boldsymbol{w})$.

Los métodos de optimización no lineal sin restricciones basados en el gradiente toman la forma general expresada anteriormente:

$$\boldsymbol{w}[k+1] = \boldsymbol{w}[k] + \alpha[k]\boldsymbol{D}[k]$$

Ecuación 2.44

donde $\boldsymbol{D}[k]$ es la dirección de búsqueda, que ha de ser una dirección de descenso y por tanto cumplir la condición:

$$\nabla R[k]\ \boldsymbol{D}[k] < 0$$

Ecuación 2.50

y $\alpha[k] > O$ la longitud del paso.

Determinación de la longitud del paso:

El establecimiento de la longitud del paso puede realizarse de dos formas distintas: bien mediante un heurístico (siendo el más sencillo de todos el dejar constante la longitud del paso, como hacen *[Kuan et al, 1993]*), bien mediante la minimización unidimensional de $R(\mathbf{w}[k] + \alpha[k]\,\mathbf{D}[k])$, respecto de $\alpha[k]$, una vez establecida la dirección de búsqueda $\mathbf{D}[k]$.

Cuando $\alpha[k]$ resulta ser la solución de un problema de minimización unidimensional, se suele imponer como criterio de finalización de esta búsqueda unidimensional el cumplimiento de las condiciones de Wolfe (*[Liu & Nocedal, 1989]*):

$$\begin{cases} R(\mathbf{w}[k] + \alpha[k]\mathbf{D}[k]) \le R(\mathbf{w}[k]) + \beta_1 \alpha[k] \nabla R(\mathbf{w}[k])^T \mathbf{D}[k] \\ \left| \nabla R(\mathbf{w}[k] + \alpha[k]\mathbf{D}[k])^T \mathbf{D}[k] \right| \le -\beta_2 \nabla R(\mathbf{w}[k])^T \mathbf{D}[k] \end{cases}$$

Ecuación 2.51

con $0<\beta_1<\beta_2$. La primera condición asegura una reducción significativa de la función de coste, mientras que la segunda evita dar pasos demasiado cortos. Valores típicos a utilizar con estas condiciones son $\beta_1=0.0001$ y $\beta_2=0.9$

El método de minimización unidimensional utilizado en esta tesis, para la búsqueda de $\alpha[k]$, es un método de ajuste cúbico con salvaguarda (*[Bertsekas, 1979], [Gill et all, 1981]*).

Determinación de la dirección de búsqueda

Veamos a continuación las direcciones de búsqueda propuestas por algunos métodos de optimización no lineal sin restricciones:

sean: $\mathbf{g}_k = \nabla R(\mathbf{w}[k])$
$\mathbf{p}_k = \mathbf{w}[k] - \mathbf{w}[k-1]$
$\mathbf{q}_k = \nabla R(\mathbf{w}[k]) - \nabla R(\mathbf{w}[k-1])$

1) Método del descenso del gradiente (Steepest Descent)

$$\mathbf{D}[k] = -\mathbf{g}_k$$

Ecuación 2.52

2) Método de gradiente conjugado versión Polack-Ribiere (*[Bertsekas, 1979]*)

$$\boldsymbol{D}[k] = -\boldsymbol{g}_k + \frac{(\boldsymbol{g}_k^T \boldsymbol{q}_k)}{(\boldsymbol{g}_{k-1}^T \boldsymbol{g}_{k-1})} \boldsymbol{d}_{k-1}$$

Ecuación 2.53

3) Método de gradiente conjugado versión Fletcher-Reeves (*[Bertsekas, 1979]*)

$$\boldsymbol{D}[k] = -\boldsymbol{g}_k + \frac{(\boldsymbol{g}_k^T \boldsymbol{g}_k)}{(\boldsymbol{g}_{k-1}^T \boldsymbol{g}_{k-1})} \boldsymbol{d}_{k-1}$$

Ecuación 2.54

4) Método Quasi-Newton de "memoria reducida" versión 0 (LMQN0) (*[Gill et all, 1981]*)

$$\boldsymbol{D}[k] = -\boldsymbol{g}_k + \frac{1}{(\boldsymbol{q}_k^T \boldsymbol{p}_k)} (\boldsymbol{p}_k^T \boldsymbol{g}_k \boldsymbol{q}_k + \boldsymbol{q}_k^T \boldsymbol{g}_k \boldsymbol{p}_k) - \frac{(\boldsymbol{p}_k^T \boldsymbol{g}_k)}{(\boldsymbol{q}_k^T \boldsymbol{p}_k)} (1 + \frac{(\boldsymbol{q}_k^T \boldsymbol{q}_k)}{(\boldsymbol{q}_k^T \boldsymbol{p}_k)}) \boldsymbol{p}_k$$

Ecuación 2.55

5) Método Quasi-Newton de "memoria reducida" versión 1 (LMQN1) (*[Luenberger, 1984]*)

$$\boldsymbol{D}[k] = -\boldsymbol{g}_k + \frac{(\boldsymbol{q}_k^T \boldsymbol{g}_k)}{(\boldsymbol{p}_k^T \boldsymbol{q}_k)} \boldsymbol{p}(k)$$

Ecuación 2.56

6) Método Quasi-Newton autoescalado (SCQN) (*[Luenberger, 1984]*)

$$\begin{cases} \boldsymbol{D}[k] = -\boldsymbol{B}_k \boldsymbol{g}_k \\ \boldsymbol{B}_k = \left(\boldsymbol{B}_{k-1} - \frac{(\boldsymbol{B}_{k-1} \boldsymbol{q}_k \boldsymbol{q}_k^T \boldsymbol{B}_{k-1})}{(\boldsymbol{q}_k^T \boldsymbol{B}_{k-1} \boldsymbol{q}_k)} \right) \frac{(\boldsymbol{p}_k^T \boldsymbol{q}_k)}{(\boldsymbol{q}_k^T \boldsymbol{B}_{k-1} \boldsymbol{q}_k)} + \frac{(\boldsymbol{p}_k \boldsymbol{p}_k^T)}{(\boldsymbol{p}_k^T \boldsymbol{q}_k)} \end{cases}$$

Ecuación 2.57

7) Método Quasi-Newton autoescalado con reinicio (SCQNR) (*[Luenberger, 1984]*)

$$\begin{cases} \boldsymbol{D}[k] = -\boldsymbol{B}_k\, \boldsymbol{g}_k \\ \boldsymbol{B}_k = \left(\boldsymbol{B}_{k-1} - \dfrac{(\boldsymbol{B}_{k-1}\, \boldsymbol{q}_k\, \boldsymbol{q}_k^T\, \boldsymbol{B}_{k-1})}{(\boldsymbol{q}_k^T\, \boldsymbol{B}_{k-1}\, \boldsymbol{q}_k)} \right) \dfrac{(\boldsymbol{p}_k^T \boldsymbol{q}_k)}{(\boldsymbol{q}_k^T\, \boldsymbol{B}_{k-1}\, \boldsymbol{q}_k)} + \dfrac{(\boldsymbol{p}_k \boldsymbol{p}_k^T)}{(\boldsymbol{p}_k^T\, \boldsymbol{q}_k)} & \text{si k no es multiplo de q} \\ \boldsymbol{B}_k = \boldsymbol{I} \quad \text{(matriz identidad)} & \text{si k es multiplo de q} \end{cases}$$

Ecuación 2.58

siendo q la dimensión de $\boldsymbol{w}$.

Para evaluar los distintos métodos propuestos tomemos como función de coste a evaluar la clásica función de Rosenbrock (*[Gill et all, 1981]*), cuya expresión es:

$$R(\boldsymbol{w}) = 100(w_2 - w_1^2)^2 + (1 - w_1)^2$$

Ecuación 2.59

Esta función, también conocida como la "función plátano", tiene un único mínimo situado en el punto $(1,1)^T$ al que se accede a través de un valle en forma de arco. La forma de este valle permite poner a prueba la bondad de la dirección de búsqueda propuesta por los distintos métodos de optimización, así como la efectividad de los métodos de búsqueda unidimensional.

Todos los ensayos toman como punto inicial el $\boldsymbol{w}[0]=(-1.2,1.0)^T$ y darán por finalizada la optimización cuando se alcance un punto $\boldsymbol{w}[k]$ tal que:

$$\begin{cases} \left\| \boldsymbol{w}[k] - \boldsymbol{w}[k-1] \right\|^2 < 1.e^{-3} \\ \left\| \nabla R(\boldsymbol{w}[k]) \right\|^2 < 1.e^{-7} \end{cases}$$

Ecuación 2.60

Los métodos de optimización sin restricciones ensayados han sido programados en lenguaje ANSI C, y ejecutados en un ordenador tipo *PC DX33*.

En todos los casos se ha utilizado para el establecimiento de $\alpha[k]$ una búsqueda unidimensional por ajuste cúbico y salvaguarda que se da por finalizada cuando se cumple alguno de los siguientes criterios de finalización:

$$\begin{cases} \left| \left[\frac{\partial R(\mathbf{w}[k] + \alpha\, \mathbf{d}[k])}{\partial \alpha} \right]_{\alpha=\alpha_k} \right| \leq 0.005 \left| \left[\frac{\partial R(\mathbf{w}[k] + \alpha\, \mathbf{d}[k])}{\partial \alpha} \right]_{\alpha=0} \right| \\ \left| \alpha[k] - \alpha[k]^{opt} \right| < 1.e^{-3} \quad siendo: \left| \left[\frac{\partial R(\mathbf{w}[k] + \alpha\, \mathbf{d}[k])}{\partial \alpha} \right]_{\alpha=\alpha(k)^{opt}} \right| = 0 \end{cases}$$

Ecuación 2.61

El siguiente cuadro recoge los resultados obtenidos en la optimización de la función de Rosenbrock con los anteriores algoritmos. Se han hecho dos tipos de ensayos: en el primero de ellos la función de evaluación de la función de coste suministraba al mismo tiempo el valor "exacto" del gradiente $\nabla R(\mathbf{w}[k])$, mientras que en el segundo ensayo se estimaba numéricamente el valor del vector gradiente según la Ecuación 2.43.

En cada caso han sido recogidos el número de iteraciones empleadas hasta llegar a la solución, el tiempo medio empleado en cada iteración y el tiempo total empleado en la búsqueda.

	Gradiente exacto			**Gradiente aproximado**		
Método	**Nº de Iterac.**	**T. por It. (ms.)**	**T. Total (ms.)**	**Nº de Iterac.**	**T. por It. (ms.)**	**T. total (ms.)**
1. Descenso grad.	3421	0.389	1329.67	2179	0.731	1593.40
2. Polack Ribiere	20	0.605	12.09	27	1.140	30.77
3. Fletcher Reeves	62	0.496	30.77	72	1.160	83.52
4. LMQN0	20	0.605	12.09	21	1.570	32.97
5. LMQN1	20	0.550	10.99	21	1.465	30.77
6. SCQN	23	0.763	17.58	71	1.083	76.92
7. SCQNR	38	0.636	24.18	40	1.319	52.75

Tabla 2.1: Estudio comparativo de diversos métodos de optimización no lineal sin restricciones: los resultados han sido obtenidos mediante la ejecución, en un ordenador tipo PC DX33, de un programa codificado en lenguaje C. La función de coste a minimizar es la función de Rosenbrok, tomando como punto inicial el $\mathbf{w}[0]=(-1.2,1.0)^T$. Para cada método se presenta el número total de iteraciones utilizado en la búsqueda, el tiempo medio por iteración, y el tiempo total empleado.

La primera conclusión que puede extraerse del examen de la Tabla 2.1 es el buen comportamiento generalizado de los métodos de solución (salvo el descenso del gradiente) cuando el gradiente es calculado de forma "exacta".

Hay que señalar que los criterios de finalización impuestos a la búsqueda unidimensional no son muy rígidos, habiendo llegado a un compromiso entre la exactitud de esta búsqueda y el tiempo empleado en esta etapa. Sin embargo los métodos ensayados siguen siendo robustos frente a la posible inexactitud de α.

La aproximación numérica del gradiente implica un incremento en el tiempo empleado en cada iteración. Esto es debido a que su estimación requiere *2q* evaluaciones de la función objetivo, al haber utilizado la estimación centrada (Ecuación 2.43) en lugar de la estimación sesgada (Ecuación 2.42), que sólo hubiese requerido *(q+1)* evaluaciones de la función objetivo, a costa de perder precisión en la estimación.

La pérdida de precisión en el gradiente tiene efectos que varían según el método de optimización utilizado. Así por ejemplo, al utilizar el gradiente aproximado el método del descenso del gradiente mejora considerablemente su comportamiento, en lo referente a número de iteraciones. Esto es debido a que la dirección del gradiente resulta nefasta con esta función de coste, y por tanto, al optimizar el valor de α, se sigue un camino en zigzag a pasos muy cortos.

El resto de los algoritmos aumentan el número de iteraciones necesarias al introducir el gradiente aproximado. Este efecto es especialmente relevante en los métodos de gradiente conjugado (3 y 4) y en el Quasi-Newton (6). Esto es debido a la gran importancia que dan estos métodos a la información contenida en este vector. Los métodos Quasi-Newton de "memoria reducida" (4 y 5) y el Quasi-Newton con reinicio (7) se muestran más robustos frente a la imprecisión del gradiente. Así por ejemplo el método 7 supera al 6 en el caso de gradiente aproximado, siendo perjudicial la reinicialización de la matriz H en el caso de gradiente exacto.

Otra consideración a tener en cuenta es la superioridad del método del gradiente conjugado versión Polack-Ribiere frente a la de Fletcher-Reeves, pese a la similitud de su formulación.

Como conclusión final, cabe resaltar las buenas cualidades de los métodos LMQN1 y Polack-Ribiere en cuanto a simplicidad de su formulación, bajos requerimientos de almacenamiento y efectividad.

Por estas razones tomaremos como método de optimización no lineal sin restricciones el método quasi-Newton de "memoria reducida" LMQN1. La Figura 2.6 ilustra el comportamiento de este algoritmo de optimización durante la

minimización de la función de Rosenbrok. En ella aparecen las curvas de nivel de esta función de coste, el punto inicial de la búsqueda ($\boldsymbol{w}[0]=(-1.2,1.0)^T$), la solución alcanzada ($\boldsymbol{w}^*=(1.0,1.0)^T$), y el camino seguido durante la búsqueda. Las direcciones de búsqueda propuestas por el método LMQN1 se adaptan perfectamente a la hipersuperficie de error, minimizando el número de iteraciones necesarias.

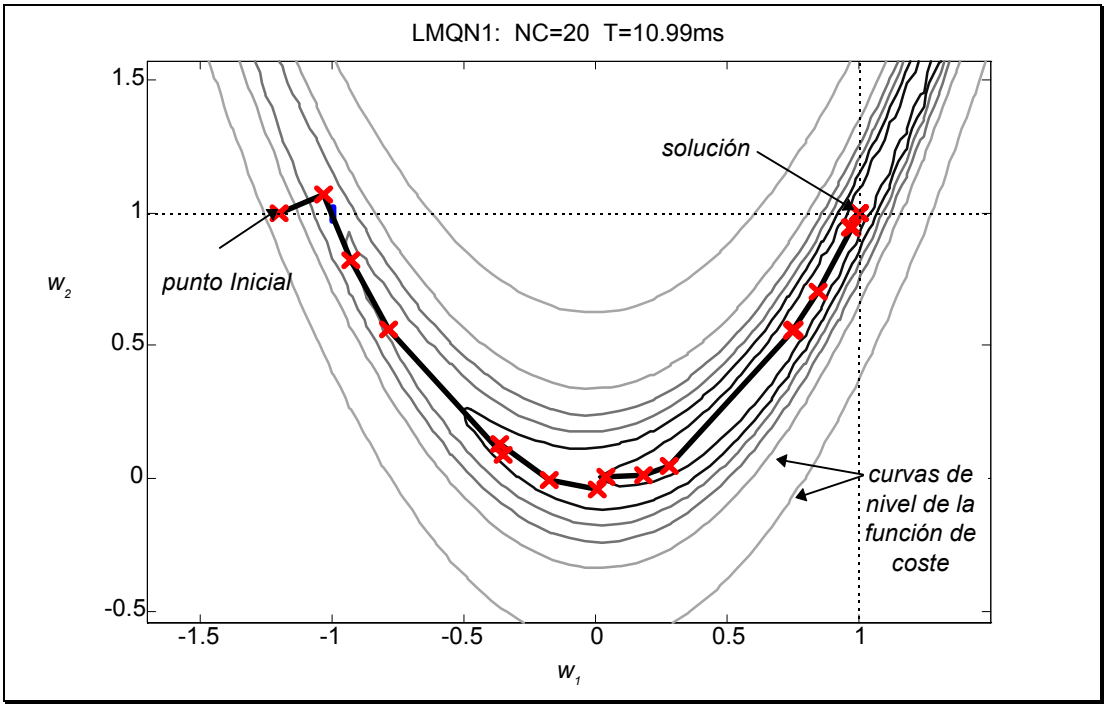

Figura 2.6: Comportamiento del algoritmo LMQN1 en la minimización de la función de Rosenbrok

Esta característica de los métodos de optimización que determinan la dirección de búsqueda a partir de valores pasados del gradiente contrasta con el comportamiento del método del descenso del gradiente.

La dirección de descenso dada por el gradiente con funciones de coste como la de Rosenbrok, tiene el problema que nada más comenzar a avanzar en esa dirección, la función de coste comienza a aumentar. Por esta razón, el camino seguido desde el punto inicial hasta la solución se lleva a cabo a pasos demasiado cortos. Este fenómeno queda ilustrado en la Figura 2.7:

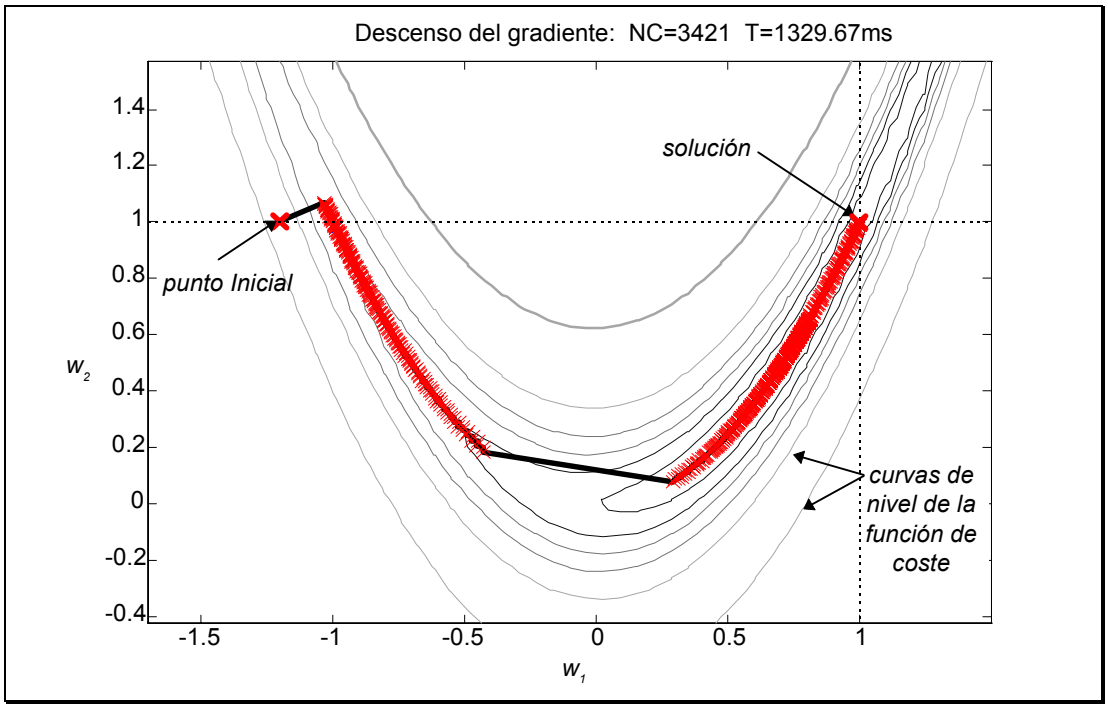

Figura 2.7: Comportamiento del algoritmo de descenso del gradiente en la minimización de la función de Rosenbrok

Cabe preguntarse entonces si la minimización unidimensional, realizada para determinar el valor de la longitud del paso, es realmente beneficiosa en el caso de tomar como dirección de búsqueda la del gradiente.

En el campo de las redes neuronales artificiales, el método de minimización del error de entrenamiento más utilizado es una variante del método del descenso del gradiente conocida como la Regla Delta[i] (*[Rumelhart et al., 1986]*). El éxito de este método tiene su origen en la sencillez de su formulación, ya que no realiza la optimización unidimensional para la determinación de la longitud del paso.

[i] Este método es también conocido en el mundo conexionista bajo el nombre de retropropagación ("backpropagation"), por extensión del algoritmo de cálculo del gradiente en Perceptrones Multicapa.

c) La Regla Delta

La regla Delta (*[Rumelhart et al., 1986b]*) es simplemente un descenso del gradiente con longitud de paso constante, de tal forma que el incremento que se da al vector de parámetros en cada iteración es proporcional al gradiente de la función de coste.

Esta sencilla regla suele modificarse en la práctica añadiendo un término proporcional al incremento del vector de parámetros del paso anterior, de tal forma que se obtiene:

$$\Delta \boldsymbol{w}[k] = \boldsymbol{w}[k+1] - \boldsymbol{w}[k] = -\alpha \nabla R(\boldsymbol{w}[k]) + \eta \, \Delta \boldsymbol{w}[k-1]$$

Ecuación 2.62

La constante α recibe el nombre de ratio de aprendizaje, ya que está directamente relacionada con la rapidez y estabilidad de este proceso de búsqueda. Valores elevados de α harán que la búsqueda avance a grandes pasos, pudiendo producir comportamientos oscilatorios. Valores reducidos de α evitarán las oscilaciones pero harán más lenta la búsqueda.

La constante η recibe el nombre de momento de inercia y su objetivo es filtrar las posibles oscilaciones mencionadas en el párrafo anterior. Gracias a este efecto se pueden utilizar valores de α más elevados, aumentando de esta forma la rapidez de la optimización. Otro efecto que se consigue al introducir el momento de inercia es el de escapar a mínimos locales de la función de error. Valores típicos a utilizar son α*=0.25* y η*=0.9*.

En el caso concreto de la aproximación funcional a partir de un conjunto de entrenamiento, existen dos variantes de revisión de los parámetros: la revisión por época[i] y la revisión por caso. En la revisión por época (o versión "batch"), el valor de los parámetros no se modifica hasta que no han sido utilizados todos los ejemplos de entrenamiento para el cálculo del gradiente, que en este caso será el gradiente real.

En la revisión por caso (o versión "online"), el vector de parámetros se modifica tras cada presentación de ejemplo de entrenamiento. Los ejemplos pueden ser tomados de forma secuencial (siempre en el mismo orden), o aleatoria (para evitar posibles comportamientos cíclicos). Si descomponemos el vector gradiente de la forma:

[i] Una época equivale a la presentación de cada uno de los ejemplos del conjunto de entrenamiento al aproximador funcional, para la modificación de sus parámetros.

$$\begin{cases} \nabla R(\boldsymbol{w}) = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} \nabla R^{[i]}(\boldsymbol{w}) \\ \nabla R^{[i]}(\boldsymbol{w}) = -2\displaystyle\sum_{j=1}^{m} \left(\boldsymbol{d}_j[i] - f_j(\boldsymbol{x}[i], \boldsymbol{w})\right) \dfrac{\partial f_j(\boldsymbol{x}[i], \boldsymbol{w})}{\partial \boldsymbol{w}} \end{cases}$$

Ecuación 2.63

podemos utilizar los vectores $\nabla R^{[i]}(\boldsymbol{w})$ para modificar el vector de parámetros $\boldsymbol{w}$ con cada ejemplo de entrenamiento. Para ello, si en la iteración *j* de la época *k*, se toma el ejemplo de entrenamiento de índice *i(j)*, el incremento a dar al vector de parámetros será:

$$\Delta \boldsymbol{w}[j] = -\alpha \nabla R^{[i(j)]}(\boldsymbol{w}[j]) + \eta \Delta \boldsymbol{w}[j-1]$$

Ecuación 2.64

Hay que señalar que aunque los ejemplos de entrenamiento sean tomados de forma secuencial, la revisión por caso no equivale en modo alguno a la revisión por época. Esto es debido a que al irse modificando los parámetros tras cada presentación de ejemplo, la media de los gradientes individuales no iguala al gradiente total (como en la Ecuación 2.63). La práctica demuestra sin embargo que la revisión por caso arroja mejores resultados que la revisión por época, aunque su fundamentación matemática sea algo más oscura.

El principal problema de este método de optimización es la elección de los valores del ratio de aprendizaje y del momento de inercia, siendo el primero de ellos el más crítico. Los valores típicos de estos parámetros que han sido propuestos previamente no son más que puras indicaciones de carácter general, y pueden ser utilizados al comienzo de la optimización. Para conseguir resultados aceptables será sin embargo necesario modificar de forma adaptativa el ratio de aprendizaje, con el fin de acelerar la convergencia. Podemos citar cuatro reglas generales para guiar este proceso de adaptación (*[Jacobs, 1988]*):

1.- Cada componente del vector de parámetros $\boldsymbol{w}$ debería llevar asociado su propio ratio de aprendizaje

2.- Cada ratio de aprendizaje debería poder variar de una iteración a otra.

3.- Cuando la derivada de la función de coste respecto de un parámetro (dR/dw_i) mantiene su signo durante varias iteraciones consecutivas, el ratio de aprendizaje asociado a ese parámetro debería ser incrementado.

4.- Cuando la derivada de la función de coste respecto de un parámetro (dR/dw_i) alterna su signo durante varias iteraciones consecutivas, el ratio de aprendizaje asociado a ese parámetro debería ser decrementado.

A continuación vamos a presentar una regla de adaptación del ratio de aprendizaje que es conocida bajo el nombre de "Regla Delta-Bar-Delta". Esta regla cumple las cuatro pautas generales enunciadas anteriormente, estableciendo un mecanismo automático de adaptación para cada uno de los ratios de aprendizaje (uno por cada componente del vector de parámetros $\boldsymbol{w}$): sean $\kappa \geq 0$ y $1 > \lambda \geq 0$ dos parámetros de control de la adaptación *y* demos nombre a la derivada parcial de la función de coste respecto del parámetro *i* en la iteración *k*:

$$\nabla R_i[k] = \frac{\partial R(\boldsymbol{w}[k])}{\partial w_i[k]}$$

Ecuación 2.65

y a la "derivada acumulada móvil":

$$S_i[k] = (1-\xi)\ \nabla R_i[k-1] \ + \ \xi\, S_i[k-1]$$

Ecuación 2.66

siendo $1 > \xi > 0$ otra constante. Podemos entonces utilizar como regla de adaptación del ratio de aprendizaje, asociado al parámetro w_i, la "Regla Delta-Bar-Delta" dada por:

$$\Delta\alpha_i[k] = \begin{cases} \kappa & si \quad S_i[k-1]\nabla R_i[k] > 0 \\ -\lambda\alpha_i(k) & si \quad S_i[k-1]\ \nabla R_i[k] < 0 \\ 0 & en\ otro\ caso \end{cases}$$

Ecuación 2.67

Los resultados obtenidos con esta regla llegan a ser sorprendentes en algunos casos, a costa siempre de aumentar la capacidad de almacenamiento requerida. Es necesario señalar por otro lado que el mero hecho de dotar a cada parámetro de un ratio de aprendizaje distinto modifica la dirección de búsqueda, que ya no será la del gradiente.

d) Selección del método de optimización

La selección del método de optimización a utilizar para la minimización del error de entrenamiento ha de tener en cuenta la dimensión y complejidad del problema a resolver. La dimensión del problema está asociada con el número de parámetros a optimizar, mientras que su complejidad vendrá determinada por el tiempo requerido para la evaluación de la función de coste y de su gradiente (sin hacer referencia a la hipersuperficie de error).

Los "modernos" algoritmos de optimización del tipo quasi-Newton de "memoria reducida" (*[Shanno, 1990]*), permiten ya resolver problemas de optimización no lineal de elevada dimensión.

Sin embargo, cuando el número de ejemplos de entrenamiento es muy elevado, o la evaluación de la salida del aproximador funcional y de sus derivadas es muy costosa, las ventajas de la revisión por caso de los parámetros comienzan a pesar favorablemente.

Como regla general podemos decir que los algoritmos del tipo quasi-Newton de "memoria reducida" serán los más eficientes cuando la complejidad del problema sea baja o media (hasta varios millares de ejemplos de entrenamiento), pero que su eficiencia comenzará a debilitarse cuando el tiempo requerido para la evaluación de la función de coste y de sus derivadas comience a aumentar.

En estos casos sería necesario diseñar nuevas estrategias de revisión de los parámetros, que no requiriesen la evaluación del aproximador con cada uno de los ejemplos de entrenamiento en cada iteración. La utilización de métodos como la Regla Delta solucionan este problema, modificando el vector de parámetros tras cada presentación de ejemplo. Otra posible solución es estimar la función de coste y su gradiente a partir de un subconjunto de ejemplos, seleccionados de forma aleatoria o mediante la aplicación de un algoritmo de agrupamiento ("clustering") (*[Hartigan, 1975]*).

e) Escalado

Una buena técnica a seguir en todo problema de optimización es la de *escalar* el problema. La forma más habitual de escalar un problema es introducir factores de escala en cada uno de los parámetros a optimizar, de forma tal que se igualen aproximadamente las segundas derivadas de la función de coste respecto de los nuevos parámetros (*[Luenberger, 1984]*).

En nuestro caso la matriz hessiana de la función de coste depende fuertemente del vector de parámetros a optimizar, pero podemos escalar parcialmente el problema sin más que estandarizar todas las entradas y salidas del aproximador. La estandarización consiste en aplicar una transformación lineal a las señales de entrada y de salida del aproximador, de forma tal que tengan media nula y desviación típica unidad. Para estimar la media y desviación típica de las señales originales, utilizaremos el conjunto de entrenamiento. Con estas medias (m_x) y desviaciones típicas (s_x) muestrales obtendremos las señales estandarizadas (x') a partir de las originales (x), de la forma:

$$x' = \frac{x - m_x}{s_x}$$

Ecuación 2.68

Los resultados obtenidos con esta simple modificación resultan en algunos casos asombrosos, acelerando más de 10 veces el proceso de aprendizaje.

Para ilustrar este efecto, comparemos la evolución del aprendizaje de dos aproximadores con la misma estructura que tratan de aproximar la misma función $y=x^2$, partiendo del mismo vector inicial de parámetros, siguiendo la misma ley de aprendizaje (descenso del gradiente), pero aplicando el escalado sólo al conjunto de entrenamiento de uno de ellos. Los resultados obtenidos se muestran en la siguiente figura:

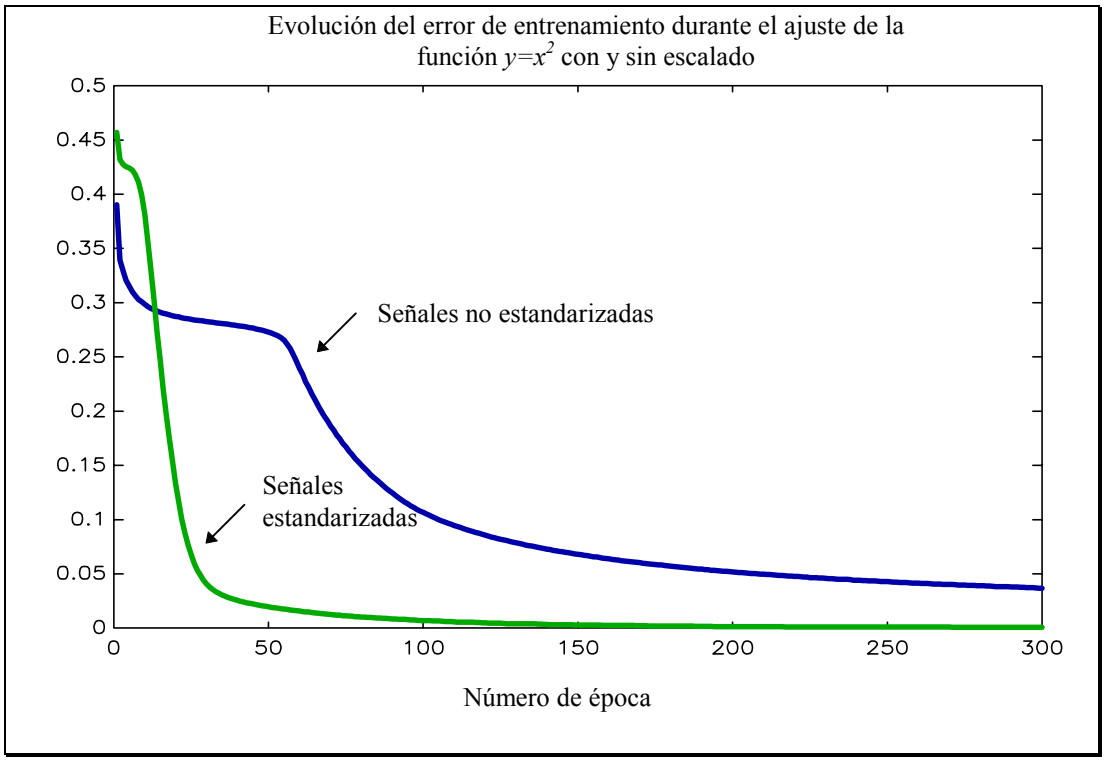

Figura 2.8: Evolución del error de entrenamiento durante el ajuste de la función $y=x^2$ con y sin escalado. El aproximador funcional utilizado es un Perceptrón Multicapa.

En lo que sigue supondremos que todos los conjuntos de datos han sido previamente estandarizados, utilizando como coeficientes de esta transformación lineal los estimados con el conjunto de entrenamiento.

f) Mínimos locales

Los métodos de minimización basados en el gradiente son métodos de búsqueda local que quedarán atrapados en todo aquel punto de la hipersuperficie de error en el que el gradiente sea nulo. Cuando se utiliza un aproximador no lineal en los parámetros, la función de coste utilizada podrá tener en general varios mínimos locales. Al aplicar un algoritmo de optimización local, dependiendo del punto de partida seleccionado (vector inicial de parámetros), se llegará a uno u otro mínimo. En cualquier caso será imposible asegurar que la solución hallada es la óptima (correspondiente a un mínimo global).

Ante esta situación caben tres posibles posturas:

- Seleccionar un buen punto inicial de tal forma que se espera que la minimización acabe en una solución aceptable. Es la solución más rápida, pero sólo podrá ser aplicada cuando se tenga el conocimiento suficiente para determinar un buen vector de parámetros iniciales.

- Probar varios puntos iniciales y quedarse con la mejor solución hallada. Esta solución requiere llevar a cabo múltiples procesos de optimización, lo que en muchos casos es inviable.

- Aplicar un método de búsqueda global que conjugue las búsquedas locales con exploraciones aleatorias. Un conocido método de búsqueda global es el llamado "simulated annealing", que dota a la búsqueda local de cierta componente aleatoria que hace tender a uno la probabilidad de hallar el óptimo global, al aumentar el número de iteraciones. El método toma su nombre del proceso metalúrgico en el que un metal es llevado casi a tempertura de fusión y luego es dejado enfriar lentamente para que la energía total del metal alcance finalmente un mínimo global. A continuación se detallan los pasos de una versión simplificada del algoritmo que aparece en *[Hecht-Nielsen, 1989]* (versiones más sofisticadas pueden encontrarse en *[Dekkers & Aarts, 1991]* y *[Maren et al., 1990]*):

1.- Inicializar aleatoriamente el vector de parámetros $\boldsymbol{w}$.

2.- Tomar un valor de T elevado, i.e., tal que:

$$exp(-\Delta/T) \geq 0.999 \qquad \forall \Delta = R(\boldsymbol{w}^{nuevo}) - R(\boldsymbol{w}^{viejo})$$

3.- Actualizar el vector de pesos $\boldsymbol{w}$ según:

$$\boldsymbol{w}^{nuevo} = LS(\boldsymbol{w}^{viejo}) \qquad \text{si } u > t$$

$$\boldsymbol{w}^{nuevo} = RS(\boldsymbol{w}^{viejo}) \qquad \text{si } u \leq t$$

con: t constante en el intervalo $[0,1[$.

u variable aleatoria distribuída según una uniforme $U[0,1[$.

$LS()$ búsqueda local en dirección de descenso: $\Delta<0$.

$RS()$ vector de variables aleatorias distribuídas uniformemente en el espacio de $\boldsymbol{w}$.

4.- Si $\Delta<0$, entonces $\boldsymbol{w}^{viejo} = \boldsymbol{w}^{nuevo}$. Ir a paso 6.

5.- Si $\Delta\geq 0$, entonces $\boldsymbol{w}^{viejo} = \boldsymbol{w}^{nuevo}$ con probabilidad $exp(-\Delta/T)$.

6.- Si K_1 actualizaciones propicias de $\boldsymbol{w}$, o K_2 actualizaciones de $\boldsymbol{w}$ desde la última actualización de T, entonces $T=\alpha T$ (con K_1 de un orden de magnitud por debajo de K_2 y con $\alpha_{típico} \in [0.8;0.999]$).

7.- Si $R(\boldsymbol{w})$ no ha decrecido más de ε en las últimas K_3 ($K_3>>K_2$) iteraciones, fin, en caso contrario ir a paso 3.

En las aplicaciones prácticas presentadas en esta tesis, la solución adoptada será la de elegir cuidadosamente el vector de parámetros iniciales, y comprobar tras el ajuste del aproximador que la solución hallada es aceptable. La estructura conexionista que será tomada como aproximador funcional permitirá partir de un buen punto inicial de la hipersuperficie de error, con lo que el problema de los mínimos locales tendrá poca importancia.

2.5 Estudio de la influencia de las variables de entrada mediante el Análisis Estadístico de Sensibilidades (AES)[i]

Una vez ajustado el aproximador funcional, es posible analizar la influencia que cada una de las variables de entrada tiene en la salida del mismo, con el fin de identificar aquellas variables no relevantes que no aportan ninguna información a la salida. La eliminación de estas variables disminuye la complejidad del aproximador, aumentando su capacidad de generalización.

Este análisis permite al mismo tiempo adentrarse en la forma de operar del aproximador, y extraer información del proceso subyacente que se está modelando.

Limitándonos por simplicidad de la exposición al caso unidimensional de salida ($m=1$), y suponiendo que los conjuntos de datos de entrada/salida han sido estandarizados, es posible comparar la influencia que cada una de las variables de entrada x_i tiene en la salida y, analizando las sensibilidades:

$$\varsigma_i = \frac{\partial y}{\partial x_i}$$

Ecuación 2.69

En el caso particular de aproximador funcional lineal:

$$y = \sum_{i=1}^{n} w_i x_i$$

Ecuación 2.70

las sensibilidades serán constantes en todo el espacio de entrada, e iguales a los coeficientes de regresión lineal:

$$\varsigma_i = w_i$$

Ecuación 2.71

[i] Estos estudios fueron realizados en la "École Polytechnique Fédérale de Lausanne" en estrecha colaboración con mi colega Thomas Czernichow. El estudio completo puede encontrarse en *[Muñoz & Czernichow, 1995]*.

Si normalizamos las sensibilidades dividiendo por su valor máximo, obtendremos una medida relativa de la influencia de cada variable. De esta forma, cuanto menor sea el valor absoluto de estos ratios, menor será la influencia de la variable de entrada correspondiente, pudiendo eliminar aquellas variables que tienen una influencia relativa despreciable.

Cuando la relación entrada/salida no es lineal, las sensibilidades dejan de ser constantes y dependen del vector de entradas $\boldsymbol{x}$. El único recurso que queda entonces para cuantificar la importancia de cada una de las variables de entrada, es el análisis de las distribuciones estadísticas de las sensibilidades. Para ello podemos utilizar el conjunto de entrenamiento para evaluar el vector de sensibilidades con cada uno de los ejemplos y construir una matriz donde cada fila corresponda a un ejemplo y cada columna a una variable de entrada. Esta matriz puede ser filtrada eliminando las sensibilidades correspondientes a ejemplos que dan un error de aproximación superior a dos veces la desviación típica del mismo.

Las variables de entrada no relevantes en la salida tendrán una distribución de sensibilidades centrada en el origen y de pequeña varianza. Las variables relevantes generarán sensibilidades no nulas en distintas regiones del espacio de entrada.

Para realizar el análisis de sensibilidades se proponen tres herramientas de análisis complementarias:

- Los histogramas de las sensibilidades: dan la información global más completa que se puede obtener sin hacer un análisis local de sensibilidades. En ellos podrán ser identificadas las variables no relevantes cuando aparezcan distribuciones "picudas" centradas en el origen (correspondientes idealmente a funciones de densidad probabilista deltas de Dirac).

- Los gráficos (media, desviación típica) de las sensibilidades: las variables no relevantes aparecerán centradas en el origen de estos gráficos, mientras que las más relevantes tenderán a alejarse de este punto.

- Los centiles del 95% normalizados del valor absoluto de las sensibilidades: para calcularlos se toma el valor absoluto de la matriz filtrada de sensibilidades, se calcula el centil del 95% de cada columna y se normalizan dividiendo por el máximo. Esta medida relativa de la importancia de cada variable de entrada servirá para rechazar aquellas variables que el 95% de los casos producen sensibilidades despreciables.

Ejemplo:

Sea *{z[k]}* un proceso aleatorio independiente e idénticamente distribuido según una N(0,1), del que depende la salida del proceso a modelar (*d[k]*) de la forma:

$$d[k] = 0.3z[k-6] - 0.6z[k-4] + 0.5z[k-1] + 0.3z[k-6]^2 - 0.2z[k-4]^2 + \varepsilon[k]$$

Ecuación 2.72

siendo *{ε[k]}* una serie de ruido blanco distribuido según una N(0,0.05)

Podemos plantear este ejemplo de modelado como un problema de aproximación funcional, definiendo como vector de variables de entrada el vector ***x*** dado por:

$$x_i[k] = z[k-i+1] \quad con \;\; i = 1,\ldots,10$$

Ecuación 2.73

de tal forma que podemos expresar:

$$d = 0.3x_7 - 0.6x_5 + 0.5x_2 + 0.3x_7{}^2 - 0.2x_5{}^2 + \varepsilon$$

Ecuación 2.74

El vector de entradas considerado es de dimansión 10, de forma tal que además de contener las variables relevantes (x_2, x_5 y x_7), contiene otras siete variables no influyentes. El objetivo de este experimento es ilustrar la aplicación del AES para verificar que el aproximador ajustado utiliza las variables pertinentes.

Generando una serie de 5000 muestras de la forma *(**x**[k],d[k])*, se dedicaron 3000 al conjunto de entrenamiento, 1000 para el de test y otras 1000 para el de validación. Se ajustó como aproximador funcional una red neuronal PRBFN tipo II (ver Capítulo 5) y fue calculada la matriz de sensibilidades.

La Figura 2.9 muestra los histogramas de las sensibilidades así obtenidas. En esta figura puede ya comprobarse cómo las sensibilidades de las variables irrelevantes han quedado distribuidas en un estrecho intervalo centrado en el origen. Las sensibilidades de las variables relevantes salen significativamente fuera de esta región, poniendo de manifiesto su importancia.

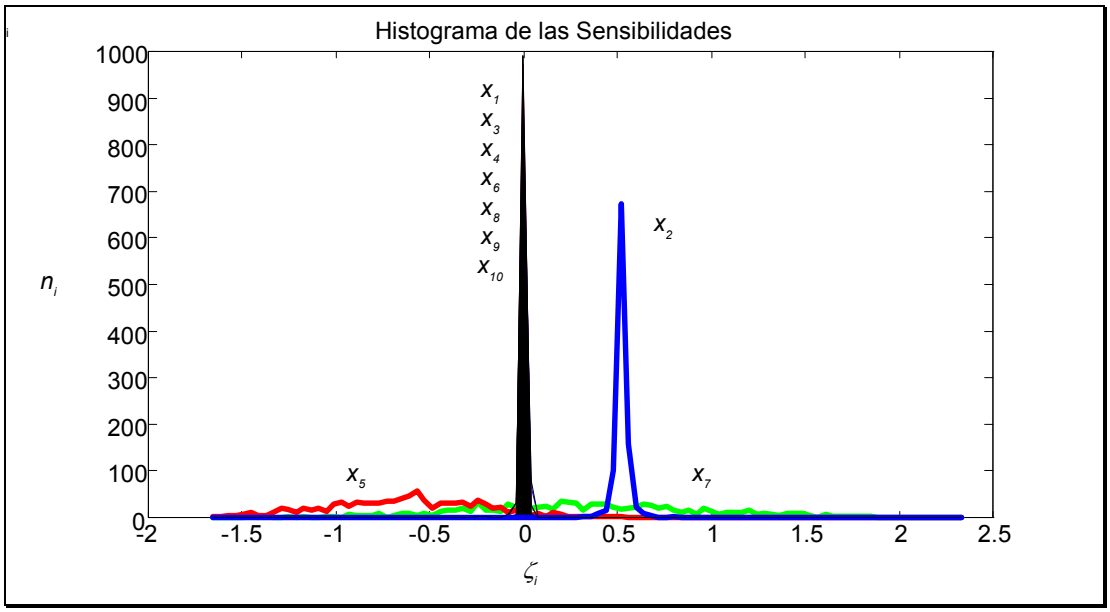

Figura 2.9: Histogramas de las Sensibilidades obtenidas en la aproximación de la función: $d=0.3x_7-0.6x_5+0.5x_2+0.3x_7^2-0.2x_5^2+\varepsilon$

La Figura 2.10 muestra el gráfico (media, desviación típica) de las sensibilidades:

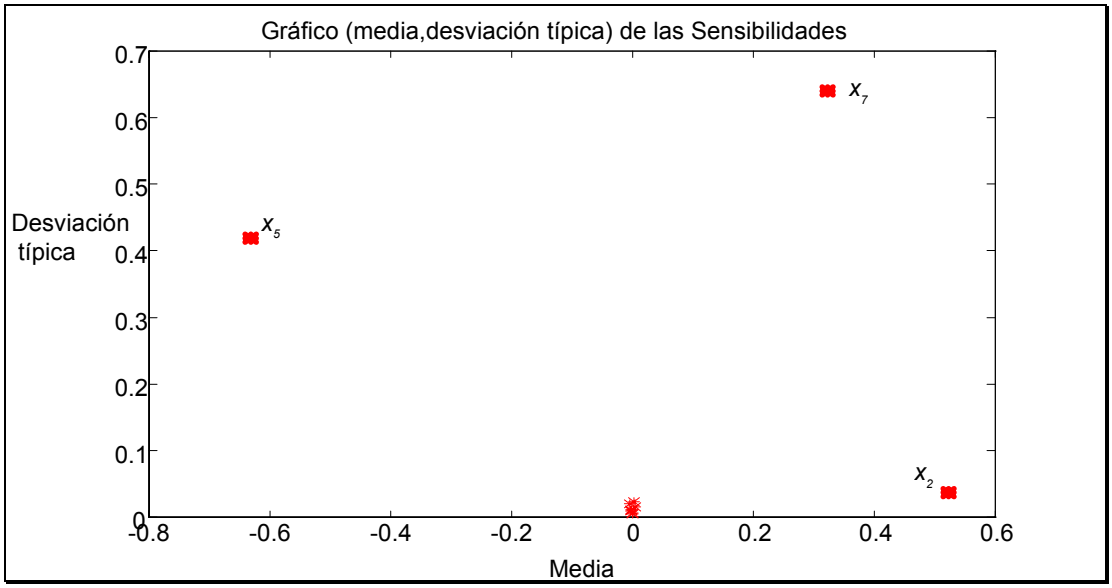

Figura 2.10: Gráfico (media, desviación típica) de las Sensibilidades obtenidas en la aproximación de la función: $d=0.3x_7-0.6x_5+0.5x_2+0.3x_7^2-0.2x_5^2+\varepsilon$

donde puede comprobarse cómo las variables irrelevantes han quedado localizadas en el origen del gráfico, mientras que las relevantes se distancian del mismo.

Para obtener una medida cuantitativa de la importancia relativa de las variables de entrada, calculemos los centiles del 95% normalizados del valor absoluto de las sensibilidades:

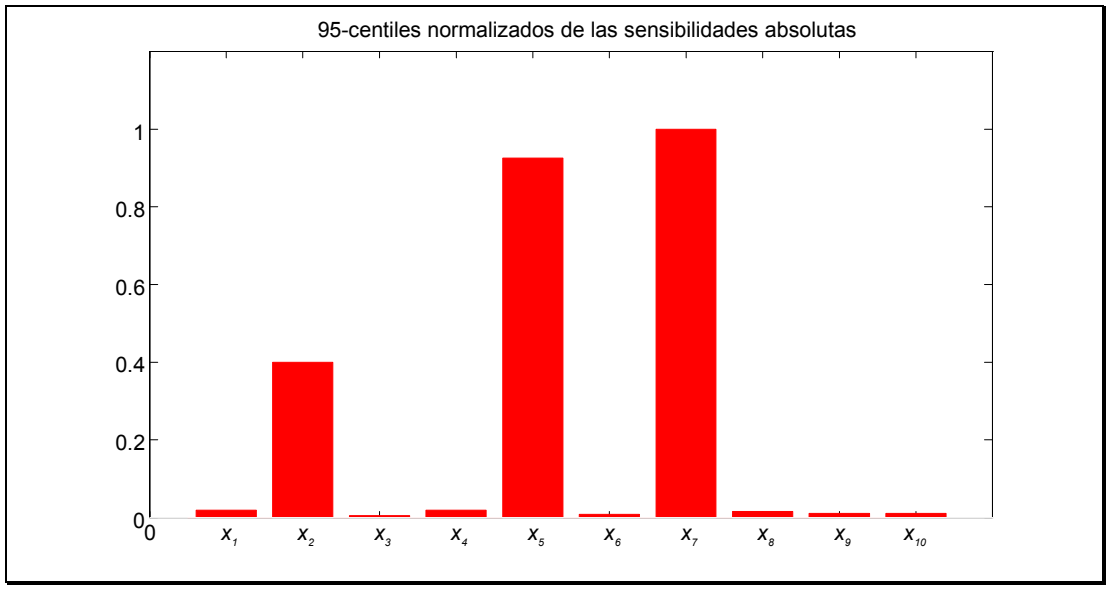

Figura 2.11: 95-centiles normalizados de las Sensibilidades obtenidas en la aproximación de la función: $d=0.3x_7-0.6x_5+0.5x_2+0.3x_7^2-0.2x_5^2+\varepsilon$

De los resultados obtenidos podemos concluir que el Análisis Estadístico de Sensibilidades ha permitido identificar el subconjunto de variables de entrada que realmente influyen en la salida del modelo. Esta identificación puede ser utilizada para reducir la complejidad del modelo mediante la eliminación de las variables de entrada no relevantes.

2.6 Esquema general de aproximación funcional

A continuación vamos a presentar un esquema general de aproximación funcional que pretende sistematizar el procedimiento propuesto para el ajuste de aproximadores funcionales. Este procedimiento aparece representado en lenguaje gráfico ANA (*[Cuadra, 1992]*) en la Figura 2.12.

En esta figura puede apreciarse cómo el punto de partida del proceso de ajuste de un aproximador funcional es un conjunto *S* de muestras de la relación entrada/salida que se quiere aproximar:

$$S = \{ (\mathbf{x}[1], \mathbf{d}[1]), (\mathbf{x}[2], \mathbf{d}[2]), \ldots, (\mathbf{x}[N], \mathbf{d}[N]) / \mathbf{x}(i) \in \Re^n , \mathbf{d}(i) \in \Re^m \}$$

Ecuación 2.75

donde implícitamente se ha seleccionado un conjunto inicial de *n* variables como variables de entrada, que son las componentes del vector $\boldsymbol{x}$.

Junto con este conjunto de muestras (*S*) y esta selección inicial de variables de entrada (representada por *n*) ha de proponerse además una familia anidada de funciones de aproximación de la que se seleccionará el aproximador funcional. Esta familia tendrá la forma general[i]:

$$f = \{ \mathbf{y} = f_h(\mathbf{x},\mathbf{w}) / \mathbf{x} \in \Re^n, \ \mathbf{y} \in \Re^m, \ \mathbf{w} \in \mathbf{W}_h \ , h=1,\ldots,H \}$$
$$\text{con: } f_1 \subset f_2 \subset \ldots \subset f_H$$

Ecuación 2.76

Partiendo de estas tres entradas, el procedimiento de ajuste de aproximadores funcionales ha de cumplir tres tareas fundamentales:

Tarea 1: Determinar si la familia *f* de funciones de aproximación tiene las características adecuadas para modelar la relación de entrada/salida descrita por los datos del conjunto *S*. En caso negativo será necesario proponer un nueva familia de

[i] Cada una de estas funciones de aproximación, además de obtener el vector de salidas estimadas $\boldsymbol{y}$ en función del vector de entradas $\boldsymbol{x}$, deberá obtener el valor de las derivadas de sus salidas respecto de sus parámetros ($d\boldsymbol{y}/d\boldsymbol{w}$) y el valor de las derivadas de sus salidas respecto de sus entradas ($d\boldsymbol{y}/d\boldsymbol{x}$). Estas derivadas permitirán aplicar algoritmos de optimización basados en el gradiente durante la optimización paramétrica, y el Análisis Estadístico de Sensibilidades para la selección de las variables de entrada. Por otro lado, cada función de aproximación deberá ser capaz de inicializar convenientemente su vector de parámetros en base al conjunto de entrenamiento suministrado.

funciones de aproximación. Este procedimiento de búsqueda se dará por finalizado cuando se encuentre una familia adecuada de funciones de aproximación a la que notaremos f^*.

Tarea 2: Determinar si el conjunto inicial de n variables de entrada contiene al menos todas las variables explicativas necesarias para la estimación de la salida del proceso. En caso afirmativo será conveniente filtrar este conjunto inicial eliminando las componentes del vector de entradas $\boldsymbol{x}$ que no son relevantes. En caso negativo será necesario buscar otras variables de entrada que aporten la información necesaria. Representaremos por n^* el conjunto final de variables de entrada al que hemos llegado con este procedimiento.

Tarea 3: Ajustar la estructura y el vector de parámetros del aproximador funcional que será tomado de la familia de funciones de aproximación f^* con el conjunto de las n^* variables de entrada seleccionadas. Para ello será necesario realizar las dos optimizaciones parciales contempladas en apartados anteriores: la optimización estructural para la determinación de su estructura (dada por el índice h^*) y la optimización paramétrica para la determinación de su vector de parámetros (al que notaremos $\boldsymbol{w}^*$).

El procedimiento propuesto para llevar a cabo estas tareas consta básicamente de los siguientes pasos (ver Figura 2.12):

Paso 1: Selección de los datos de entrenamiento, test y validación: del conjunto inicial S se extraerán el conjunto de entrenamiento (S_{entr}, del orden del 60% de S), el conjunto de test (S_{test}, del orden del 30% de S), y el conjunto de validación (S_{valid}, del orden del 10% de S). Cuando el conjunto inicial de datos es demasiado extenso y redundante pueden utilizarse los Arboles de Selección de Datos presentados en el Apéndice A para extraer un subconjunto de dimensión apropiada. Esta práctica será tratada más en detalle en el Capítulo 6.

Paso 2: Optimización estructural: este procedimiento, representado en la Figura 2.13, toma como entradas los conjuntos de entrenamiento y test, la selección inicial de variables de entrada y la familia de funciones de aproximación y tiene como objetivo ajustar la estructura y los parámetros del aproximador funcional (dados por h^* y $\boldsymbol{w}^*$ respectivamente). Este ajuste se realiza mediante dos optimizaciones parciales: la optimización estructural propiamente dicha, y la optimización paramétrica.

La optimización estructural se ha planteado como un procedimiento de búsqueda en el que el optimizador estructural irá proponiendo como estructuras candidatas las $f_h(\boldsymbol{x},\boldsymbol{w})$ en orden creciente de complejidad, hasta que el error de test devuelto por el optimizador paramétrico se estabilice o comience a aumentar.

El optimizador paramétrico (ver Figura 2.14) es el encargado de evaluar cada estructura propuesta por el optimizador estructural. Para ello aplicará un algoritmo iterativo de minimización del error de entrenamiento (R_{entr}) y en cada iteración evaluará el error de test (R_{test}). Como medida de error de aproximación se propone utilizar el error cuadrático medio dado por:

$$R = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{m}(d_j[i] - y_j(\mathbf{x}[i], \mathbf{w}))^2$$

Ecuación 2.77

de tal forma que podamos calcular el gradiente de la función de error respecto de los parámetros según:

$$\begin{cases} \nabla R(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}\nabla R^{[i]}(\mathbf{w}) \\ \nabla R^{[i]}(\mathbf{w}) = -2\sum_{j=1}^{m}\left(d_j[i] - y_j(\mathbf{x}[i], \mathbf{w})\right)\frac{\partial y_j(\mathbf{x}[i], \mathbf{w})}{\partial \mathbf{w}} \end{cases}$$

Ecuación 2.78

Cuando la tendencia del error de test sobrepase una cota predeterminada, el optimizador paramétrico detendrá su proceso de optimización, y devolverá al optimizador estructural el mínimo error de test conseguido con el correspondiente vector de parámetros y el error de entrenamiento.

Paso 3: Análisis Estadístico de Sensibilidades: una vez ajustado el aproximador, el Análisis Estadístico de Sensibilidades permitirá identificar las posibles variables de entrada que no tengan ninguna influencia significativa sobre la estimación. La eliminación de estas variables no influyentes supondrá un nuevo proceso de optimización estructural pero permitirá reducir la complejidad del aproximador aumentando su capacidad de generalización.

Paso 4: Validación: el aproximador finalmente ajustado es evaluado sobre el conjunto de validación. Si el resultado es aceptable, se congelan los parámetros del aproximador y se da por concluida esta etapa. En caso contrario será necesario comprobar que todas las entradas que influyen en la salida han sido incluidas en el modelo, y/o probar nuevas familias de aproximadores.

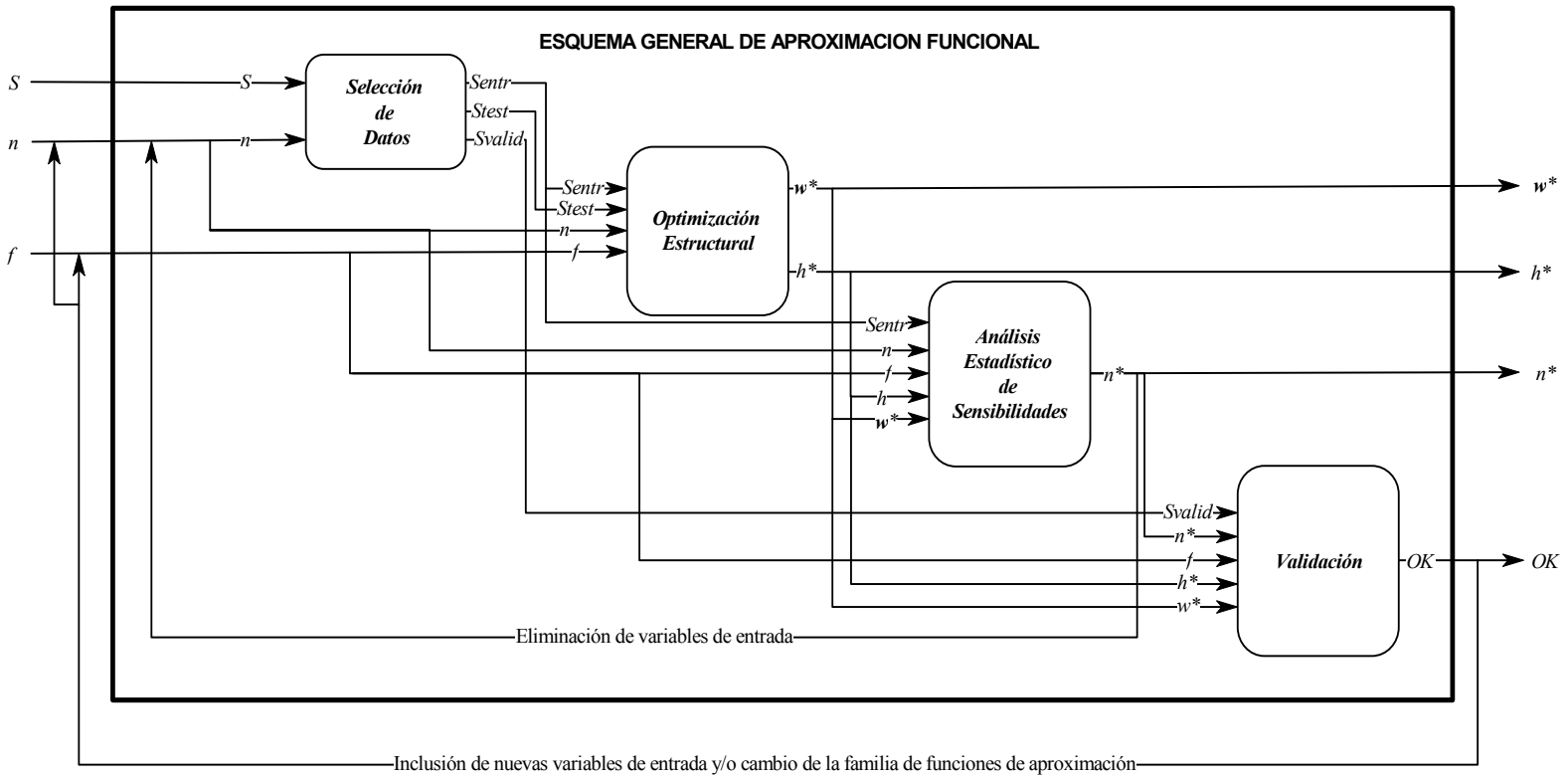

Figura 2.12: Representación ANA del esquema general de aproximación funcional

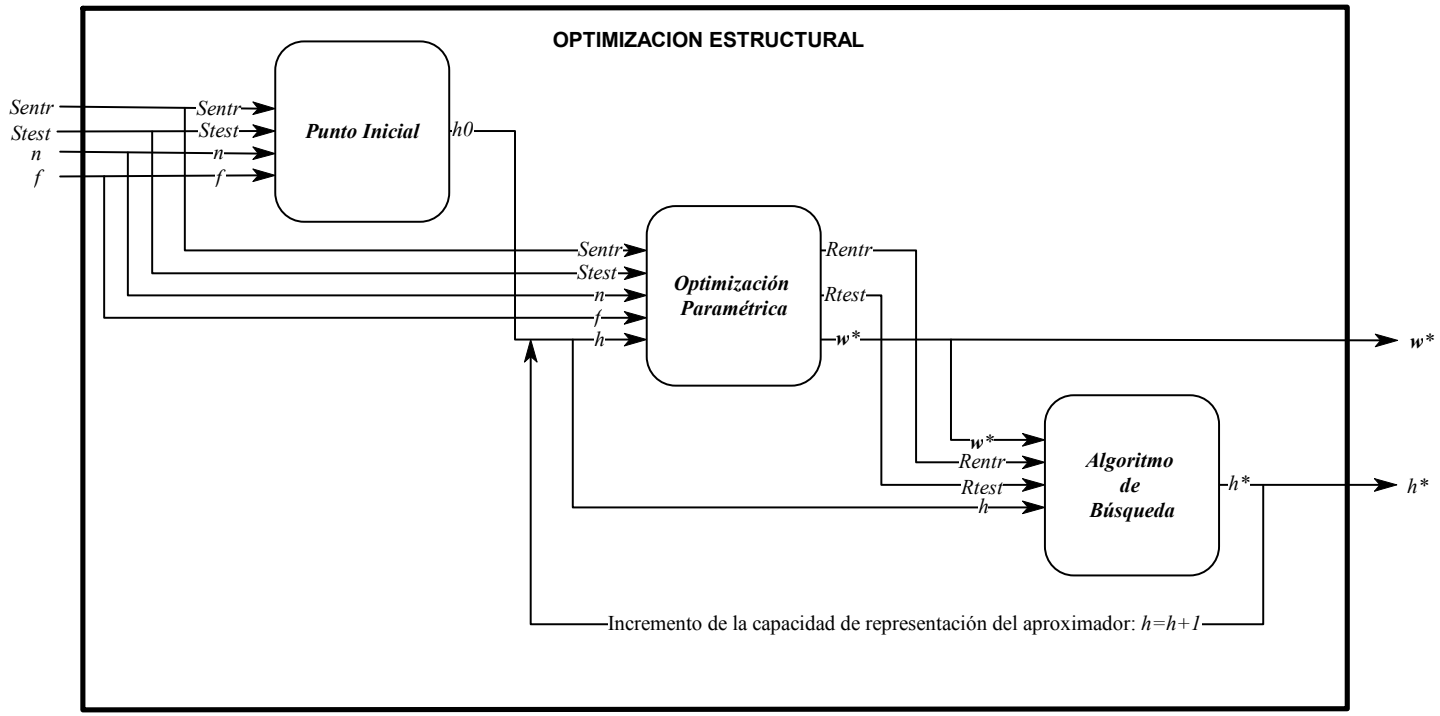

Figura 2.13: Representación ANA de la Optimización Estructural

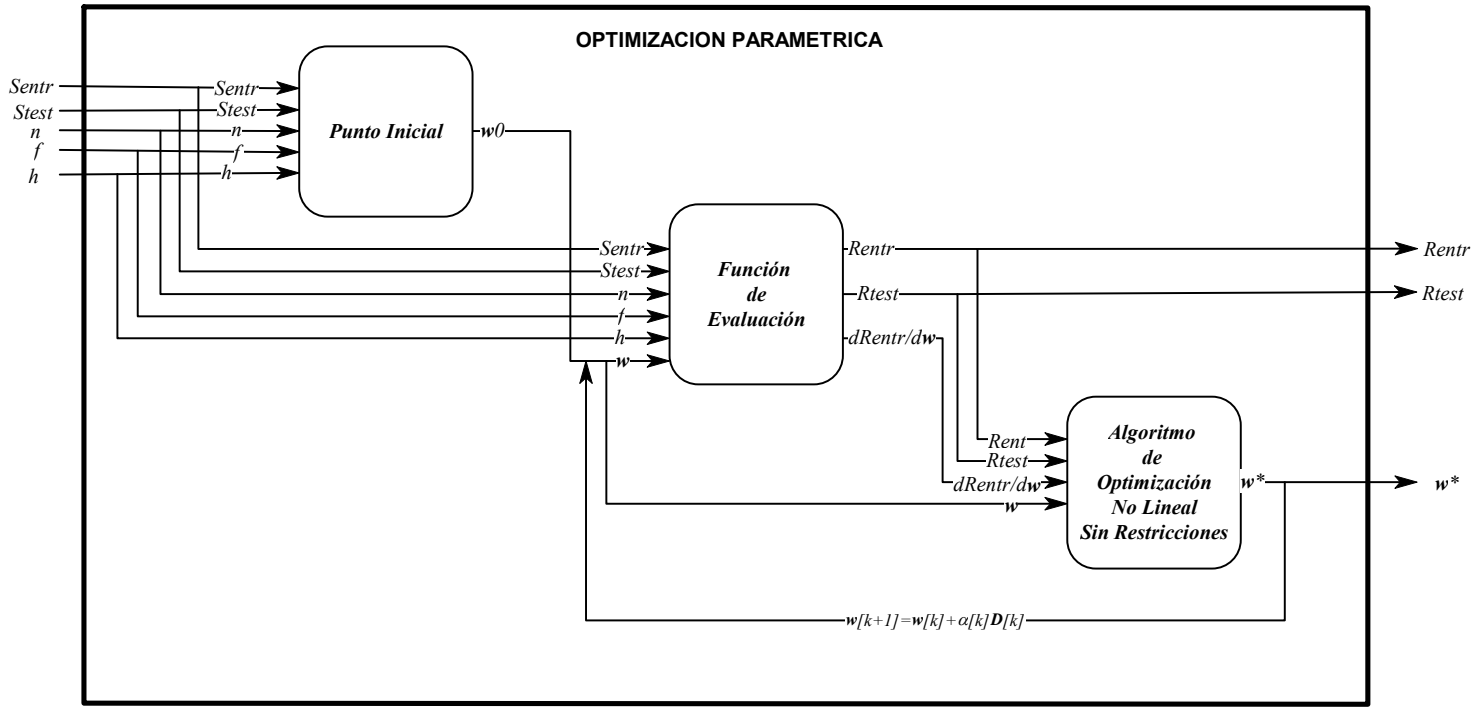

Figura 2.14: Representación ANA de la Optimización Paramétrica

3. Modelado de procesos dinámicos no lineales con aproximadores funcionales

En este capítulo se va a tratar el tema del modelado de procesos dinámicos no lineales. Las estructuras de modelado no lineal que serán presentadas son una extensión de los modelos lineales de series temporales, en los que se sustituye la transformación lineal por un aproximador funcional en general no lineal.

Los modelos presentados permitirán predecir el comportamiento normal de los componentes a supervisar, de tal forma que la detección de anomalías estará basada en el análisis de las desviaciones entre el comportamiento predicho y el observado.

3.1 Introducción

El principal problema de la identificación de sistemas es determinar la estructura adecuada del modelo que posteriormente será ajustado. El ajuste del modelo una vez que la estructura ha sido fijada (estimación paramétrica) suele ser un problema de menor complejidad, para el que existe toda una teoría clásica bien establecida (teoría de optimización *[Luenberger, 1984]*).

La regla de oro de la identificación de sistemas es "no estimar aquello que ya se conoce", lo que equivale a incluir en la estructura elegida todo el conocimiento a priori y físico que se tiene del proceso. Se suelen distinguir tres niveles de conocimiento a priori (*[Sjöberg, 1995]*):

- **Modelos de caja blanca:** este caso corresponde a situaciones en las que se conoce perfectamente el modelo, por conocimiento a priori y aplicación de leyes físicas.

- **Modelos de caja gris:** en este caso se tiene un conocimiento físico del proceso, pero varios de sus parámetros han de ser determinados a partir de las observaciones. Pueden distinguirse dos situaciones:

  - *Modelado físico:* la estructura del modelo puede establecerse aplicando leyes físicas, y los parámetros del modelo se ajustan a partir de las observaciones.

  - *Modelado semi-físico:* se utilizan leyes físicas para sugerir combinaciones no lineales de variables de entrada. Estas combinaciones forman las variables de entrada de modelos de caja negra.

- **Modelos de caja negra:** no se utiliza ningún tipo de conocimiento físico para establecer la estructura del modelo. Los modelos de caja negra están basados en criterios estadísticos capaces de modelar las relaciones existentes entre un conjunto de entradas y otro de salidas. Estas variables externas son variables físicas, pero el resto de variables y parámetros envueltos en el modelo pueden no tener significado físico. Los parámetros del modelo se ajustan a partir de un conjunto de medidas reales que caracterizan el comportamiento del proceso a modelar. El punto de partida será una familia de estructuras muy flexibles cuyas prestaciones habrán sido ya probadas en el pasado.

En este capítulo nos vamos a centrar en los modelos semi-físicos y en los modelos de caja negra. En ambos casos, la pieza fundamental del modelo será un aproximador funcional al que se le suministrarán evoluciones temporales de sus entradas. De esta forma, trataremos el problema del modelado de sistemas dinámicos como un problema de predicción de series temporales.

3.2 El caso lineal

La teoría de identificación de sistemas en el caso lineal (*[Box & Jenkins,1976], [Ljung, 1987]*) es una teoría bien conocida que suministra toda una serie de modelos y métodos de ajuste de parámetros para el modelado de procesos dinámicos lineales.

La descripción básica de un sistema lineal SISO[i] ("Single Input Single Output": una única entrada exógena *u*, y una única salida *d*), sujeto a perturbaciones aleatorias de tipo aditivo, queda representada por (*[Ljung, 1987]*):

$$d[k] = B(z)u[k] + C(z)\varepsilon[k]$$

Ecuación 3.1

Donde $\{\varepsilon[k]\}$ es una serie de variables aleatorias independientes, de media nula y varianza constante (ruido estacionario), y *z* es el operador de adelanto, de tal forma que para toda señal *u[k]*:

$$z^{j}u[k] = u[k-j]$$

Ecuación 3.2

Los polinomios *B* y *C* tienen la forma:

$$B(z) = \sum_{k=0}^{\infty} b_k z^{-k}; \qquad C(z) = 1 + \sum_{k=1}^{\infty} c_k z^{-k};$$

Ecuación 3.3

La característica común de los estimadores lineales que tratan de modelar este tipo de procesos, es que la salida estimada en el instante *k* (*y[k]*), se obtiene multiplicando un vector de coeficientes constantes (*vector de parámetros* del modelo: $\boldsymbol{w}$) por el *vector de entradas* (o *vector de regresores*: $\boldsymbol{x}$) disponibles en el instante *k*:

$$y[k] = \boldsymbol{w}^T \boldsymbol{x}[k]$$

Ecuación 3.4

[i] Nos restringimos en esta introducción a sistemas lineales SISO, aunque los modelos no lineales que serán presentados posteriormente serán del tipo MIMO ("Multiple Inputs Multiple Outputs": múltiples entradas y múltiples salidas). De igual forma sólo son tratados los sistemas lineales estacionarios.

El vector de entradas o regresores $\boldsymbol{x}$ puede constar de los siguientes elementos[i]:

- Valores presentes y pasados de otras señales externas que influyen en la salida:

$$\boldsymbol{u}[k], \ldots, \boldsymbol{u}[k-n_u] \in \Re^n$$

- Valores pasados de las salidas reales del proceso (medidas):

$$\boldsymbol{d}[k-1], \ldots, \boldsymbol{d}[k-n_d] \in \Re^m$$

- Valores pasados de las salidas estimadas, que dependen de los parámetros del modelo:

$$\boldsymbol{y}[k-1], \ldots, \boldsymbol{y}[k-n_y] \in \Re^m$$

- Valores pasados de los errores de estimación ($\boldsymbol{e}=\boldsymbol{d}-\boldsymbol{y}$), que dependen de los parámetros del modelo:

$$\boldsymbol{e}[k-1], \ldots, \boldsymbol{e}[k-n_e] \in \Re^m$$

A continuación vamos a dar un breve repaso a la estructura de estos modelos, para pasar a continuación a su extensión al caso no lineal.

3.2.1 Modelo de respuesta impulsional finita (FIR)

El sistema lineal más sencillo queda descrito por el modelo de respuesta impulsional finita (FIR):

$$d[k] = \boldsymbol{B}(z)u[k] + \varepsilon[k]$$

Ecuación 3.5

El estimador correspondiente viene dado por:

$$y[k] = b_0 u[k] + \ldots + b_{nu} u[k - n_u]$$

Ecuación 3.6

[i] Supondremos que tanto las salidas del proceso como las entradas externas tienen media nula, para no tener que incluir en el modelo términos constantes.

siendo su vector de parámetros:

$$\boldsymbol{w}=[b_0,\ldots,b_{nu}]^T$$

Ecuación 3.7

y su vector de entradas:

$$\boldsymbol{x}[k]=\left[u(k),\ldots,u(k-n_u)\right]^T$$

Ecuación 3.8

Al aumentar el orden n_u, este modelo permite ajustar la mayoría de los procesos lineales más habituales. No permite sin embargo modelar las características del ruido[i].

3.2.2 Modelo de error de salida (OE)

Una variante del modelo FIR es el modelo de error de salida (OE), en el que se incluyen como entradas valores pasados de la salida estimada:

$$d[k]=\frac{\boldsymbol{B}(z)}{\boldsymbol{A}(z)}u[k]+\varepsilon[k]$$

Ecuación 3.9

El estimador correspondiente es:

$$y[k]=a_1y[k-1]+\ldots+a_{ny}y\left[k-n_y\right]+b_0u[k]+\ldots+b_{nu}u\left[k-n_u\right]$$

Ecuación 3.10

siendo su vector de parámetros:

$$\boldsymbol{w}=[\,a_1,\ldots,a_{ny}\,,b_0,\ldots,b_{nu}]^T$$

Ecuación 3.11

[i] Por lo tanto el modelo sólo será perfecto si el ruido superpuesto a la salida del proceso es un ruido blanco, lo que equivale a decir que $H(z)=1$ en la Ecuación 3.1.

y su vector de entradas:

$$\boldsymbol{x}[k] = \left[y[k-1], \ldots, y\left[k-n_y\right], u[k], \ldots, u[k-r]\right]^T$$

Ecuación 3.12

Al igual que el modelo FIR, esta estructura permite modelar la mayoría de los procesos lineales, pero no la característica del ruido. Su ventaja frente al FIR es que requiere menos parámetros. Su gran desventaja es que es un modelo recurrente, por lo que el ajuste del vector de parámetros es más complicado.

3.2.3 Modelo autoregresivo con entradas exógenas (ARX)

Una forma de evitar la recurrencia del modelo de error de salida, es suministrar como entrada la salida real del proceso, en lugar de la salida estimada. Esto da lugar al modelo autoregresivo con entradas exógenas ARX:

$$\boldsymbol{A}(z)d[k] = \boldsymbol{B}(z)u[k] + \varepsilon[k]$$

Ecuación 3.13

El estimador resultante es:

$$y[k] = a_1 d[k-1] + \ldots + a_{nd} d\left[k-n_d\right] + b_0 u[k] + \ldots + b_{nu} u\left[k-n_u\right]$$

Ecuación 3.14

siendo su vector de parámetros:

$$\boldsymbol{w} = [\, a_0, \ldots, a_{nd}, b_0, \ldots, b_{nu}]^T$$

Ecuación 3.15

y su vector de entradas:

$$\boldsymbol{x}[k] = \left[d[k-1], \ldots, d\left[k-n_d\right], u[k], \ldots, u\left[k-n_u\right]\right]^T$$

Ecuación 3.16

Este modelo permite ajustar todo proceso lineal corrompido por un ruido aditivo (*[Ljung & Wahlberg, 1992]*), a costa de aumentar n_d y n_u. Su ventaja frente al FIR es que requiere menos parámetros, y al ser un modelo no recurrente, la estimación de sus parámetros es más sencilla que en el caso de modelos de error de salida.

3.2.4 Modelo autoregresivo de media móvil con entradas exógenas (ARMAX)

Es frecuente en la predicción de series temporales que el análisis de correlaciones residuales muestre un alto contenido de información en la serie del error de predicción. Esta información, enmascarada bajo forma de ruido coloreado, puede ser realimentada al estimador de tal forma que su utilización mejore considerablemente la calidad de la estimación, sin aumentar de forma desproporcionada el número de parámetros a estimar.

La forma general de un proceso ARMAX viene dada por:

$$\boldsymbol{A}(z)d[k] = \boldsymbol{B}(z)u[k] + \boldsymbol{C}(z)\varepsilon[k]$$

Ecuación 3.17

El estimador utilizado es:

$$\begin{aligned} y[k] = a_1 d[k-1] + \ldots + a_{nd} d[k-n_d] + \\ b_0 u[k] \quad + \ldots + b_{nu} u[k-n_u] + \\ c_1 e[k-1] + \ldots + c_{ne} e[k-n_e] \end{aligned}$$

Ecuación 3.18

siendo *e[k]* el error de estimación en el instante *k* dado por:

$$e[k] = d[k] - y[k]$$

Ecuación 3.19

Las componentes asociadas a valores pasados de las salidas reciben el nombre de autoregresivas, mientras que las asociadas a errores de estimación son llamadas de media móvil.

Su vector de parámetros resulta ser pues:

$$\boldsymbol{w} = [\, a_1,\ldots,a_{nd}\,,\, b_0,\ldots,b_{nu}\,,\, c_1,\ldots,c_{ne}]^T$$

Ecuación 3.20

y su vector de entradas:

$$\boldsymbol{x}[k] = \left[d[k-1], \ldots, d[k-n_d], u[k], \ldots, u[k-n_u], e[k-1], \ldots, e[k-n_e] \right]^T$$

Ecuación 3.21

Esta estructura permite modelar todos los procesos lineales corrompidos por un ruido aditivo, con un modelo mucho más compacto que el modelo ARX (en el caso de ruido coloreado). Su principal problema es que es un modelo recurrente, lo que complica la estimación de sus parámetros.

3.2.5 Validación de modelos lineales

Una vez que se ha propuesto una estructura determinada de modelo, el proceso de estimación de parámetros selecciona el "mejor" modelo dentro de esta estructura. La validación del modelo ajustado trata de determinar si el óptimo modelo así obtenido es lo suficientemente bueno.

Si se ha detectado que el modelo es inadecuado, sería conveniente indagar las razones de esta inaptitud, para de esta forma poder sugerir las modificaciones oportunas.

Existen básicamente dos formas de validar un modelo. La primera de ellas, conocida bajo el nombre de validación cruzada, analiza el comportamiento del modelo sobre una base de datos no utilizados para el ajuste de sus parámetros ("conjunto de validación"). Si el modelo cumple los requisitos preestablecidos con este conjunto de datos, el modelo se da por bueno.

La segunda de ellas está basada en el análisis de los residuos[i]. Al ajustar un modelo lineal al proceso descrito por Ecuación 3.1, se ha supuesto que $\{\varepsilon[k]\}$ es una serie de variables aleatorias independientes de media nula y varianza constante (ruido estacionario), distribuida según una función de densidad probabilista determinada p_ε, típicamente normal. La validación del modelo ha de comprobar que la serie de errores de estimación:

$$e[k] = d[k] - y[k]$$

Ecuación 3.22

[i] El análisis de los residuos puede también aplicarse al conjunto de validación.

cumple las hipótesis establecidas, es decir, que es una realización de una serie de variables aleatorias independientes e idénticamente distribuidas según p_ε, de media nula y varianza constante. Este procedimiento aparece representado de forma gráfica en la Figura 3.1.

El procedimiento de validación comienza con la inspección visual de la serie de residuos, lo que suele realizarse sobre un gráfico en el que se dibujan los límites $\pm 2s_e$ y $\pm 3s_e$, siendo s_e la desviación típica muestral de los residuos.

Seguidamente se comprueba la hipótesis de independencia. Para ello existe toda una serie de tests estadísticos basados en el análisis de la función de autocorrelación simple (FAS) de la serie de errores de estimación:

$$FAS_e^N(i) = \frac{\sum_{k=1}^{N-i}\left(e[k]-m_e\right)\left(e[k+i]-m_e\right)}{\sum_{k=1}^{N}\left(e[k]-m_e\right)^2} \quad siendo\ m_e = \frac{1}{N}\sum_{k=1}^{N} e[k]$$

Ecuación 3.23

de tal forma que si *{e[t]}* es una serie de ruido blanco, $\{FAS_e^N(i)\}$ se distribuye asintóticamente según una normal de media nula y desviación típica $\frac{1}{\sqrt{N}}$ (*[Wei, 1990]*).

Podemos pues establecer el intervalo de confianza del 95% en $\frac{\pm 2}{\sqrt{N}}$, de tal forma que si todos los coeficientes $FAS_e^N(i)$ (con $i>0$) están dentro de este intervalo, aceptaremos la hipótesis de independencia de los residuos. Al ser un intervalo de confianza del 95%, es de esperar que uno de cada veinte coeficientes de autocorrelación se salga fuera de estos límites. Este sobrepaso será admisible para retardos elevados, pero no para los retardos bajos, para los que la varianza está sobreestimada.

Otro test de independencia es el dado por el estadístico de Ljung y Box (*[Peña, 1986]*):

$$Q_{N,M} = N(N+2)\sum_{i=1}^{M}\frac{\left(FAS_e^N(i)\right)^2}{N-i}$$

Ecuación 3.24

que en el caso de ruido blanco se distribuye asintóticamente según una $\chi^2(M-2)$ de Pearson, pudiendo pues comprobar la independencia de los residuos con un grado de confianza α, si se cumple:

$$Q_{N,M} < \chi^2_{\alpha}(M-2)$$

Ecuación 3.25

siendo $\chi^2_{\alpha}(M-2)$ el nivel α de la distribución $\chi^2(M-2)$.

Posteriormente se comprueba la hipótesis de media nula de los residuos (esta hipótesis está garantizada en los casos FIR y ARX con ajuste de parámetros por mínimos cuadrados). Para ello, se calcula la media de los N residuos:

$$m_e = \frac{1}{N}\sum_{k=1}^{N} e[k]$$

Ecuación 3.26

se estima su varianza mediante:

$$s_e^2 = \frac{1}{q}\sum_{k=1}^{N} (e[k] - m_e)^2$$

Ecuación 3.27

siendo q el número de parámetros del modelo, y se rechaza la hipótesis de media nula si el estadístico:

$$\frac{m_e}{s_e / \sqrt{N}}$$

Ecuación 3.28

es significativamente grande comparado con una distribución N(0,1) (*[Peña, 1986]*).

El contraste de la media ha de realizarse después de comprobar que los residuos son incorrelados, para asegurar que s_e^2 es un estimador razonable de la varianza.

La estabilidad de la varianza se comprueba analizando la evolución temporal de los residuos. En caso de duda pueden estimarse las varianzas sobre distintos intervalos temporales y aplicar un test de razón de verosimilitudes (*[Peña, 1986]*).

Una vez comprobadas las hipótesis anteriores (independencia de los residuos, media nula y varianza constante), falta aún por comprobar el ajuste de la distribución de

{e[k]} a la supuesta p_ε. Bajo la hipótesis de normalidad de los residuos, justificada teóricamente por el teorema central del límite, los estimadores mínimo-cuadráticos coinciden con los máximo-verosímiles y son estimadores eficientes. Para comprobar la hipótesis de normalidad existen varios test estadísticos, pudiendo destacar el test de la χ^2 de Pearson (*[Peña, 1986]*).

El último paso de la validación del modelo es comprobar la independencia entre los residuos *{e[k]}* y las entradas exógenas *{u[k]}* (incluidas o no en el modelo). Este test trata de asegurar que toda la información relevante disponible ha sido utilizada por el modelo. Para más detalles consultar *[Ljung, 1987]* y *[Peña, 1986]*.

El cumplimiento de las hipótesis anteriores asegura la idoneidad del modelo desde un punto de vista teórico (y lineal). Sin embargo hay que asegurar también el buen comportamiento práctico del modelo, comprobando su funcionalidad desde el punto de vista de los objetivos que ha de cumplir. En el caso concreto de la detección de anomalías basada en el modelado del funcionamiento normal de los componentes, el objetivo del modelo es poner de manifiesto posibles comportamientos anómalos, utilizando como indicador los errores de predicción. Cuanto más sensible a las faltas sea el modelo, y cuantas menos falsas alarmas produzca, tanto mejor será desde un punto de vista práctico.

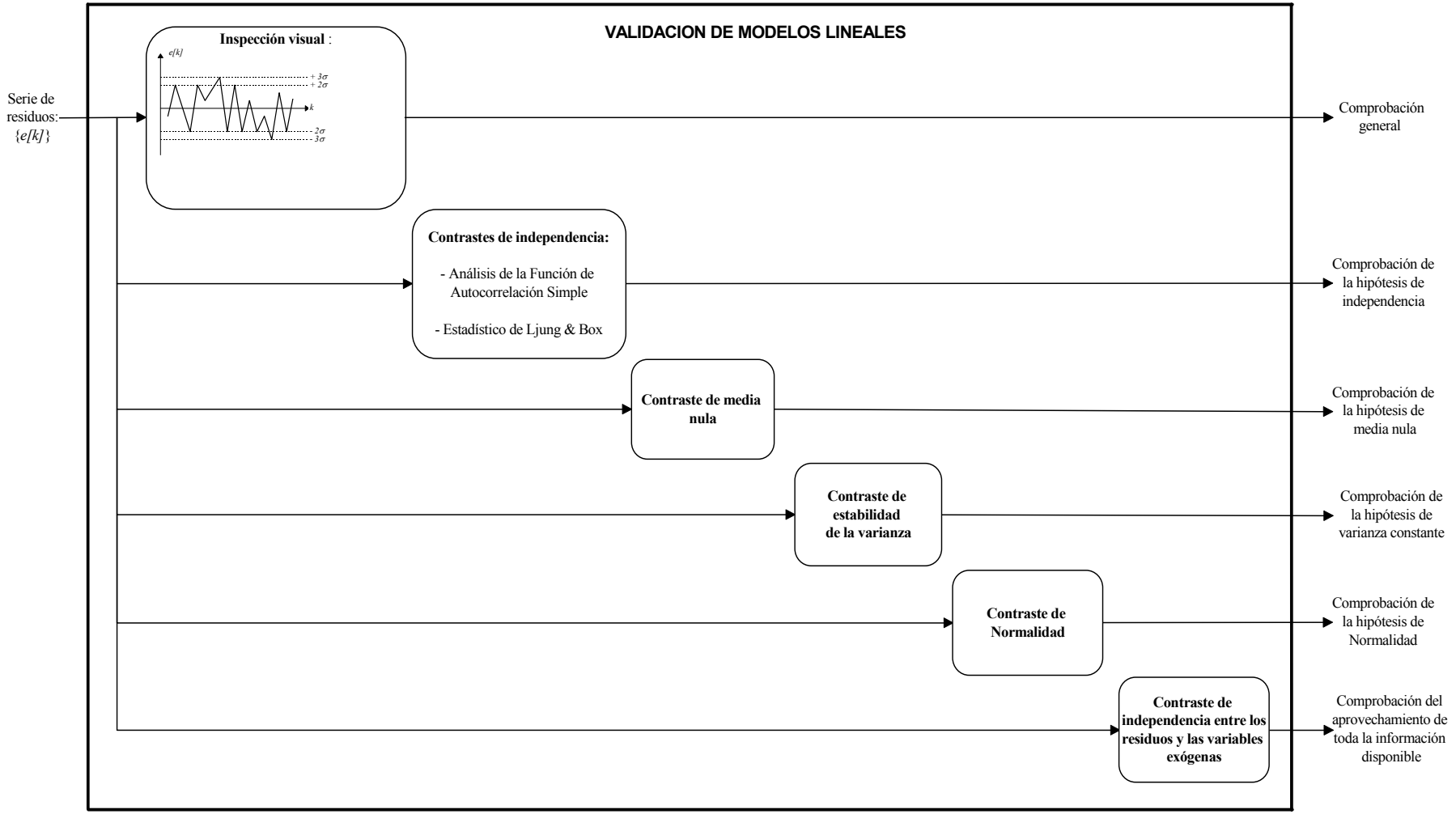

Figura 3.1: Validación de modelos lineales

3.3 El caso no lineal

Los sistemas que ahora vamos a considerar son una extensión de los sistemas lineales, y quedan descritos por:

$$\boldsymbol{d}[k] = g(\boldsymbol{d}^{\{k-1\}}, \boldsymbol{u}^{\{k\}}, \boldsymbol{\varepsilon}^{\{k-1\}}) + \boldsymbol{\varepsilon}[k]$$

Ecuación 3.29

donde:

- *g:* es una función no lineal.

- $\boldsymbol{d}[k] \in \Re^m$ es el valor de las salidas del sistema en el instante *k*.

- $\boldsymbol{d}^{\{k-1\}} = [\boldsymbol{d}[k-1], \boldsymbol{d}[k-2], ...]^T$ es un vector que contiene el valor de las salidas del proceso en los instantes *k-1* y anteriores.

- $\boldsymbol{u}[k] \in \Re^n$ es el valor de las entradas exógenas del sistema en el instante *k*.

- $\boldsymbol{u}^{\{k\}} = [\boldsymbol{u}[k], \boldsymbol{u}[k-1], ...]^T$ es un vector que contiene el valor de las entradas exógenas en los instantes *k* y anteriores.

- $\boldsymbol{\varepsilon}[k] \in \Re^m$ es la realización de un proceso de ruido blanco estacionario superpuesto a la salida del sistema en el instante *k*.

- $\boldsymbol{\varepsilon}^{\{k-1\}} = [\boldsymbol{\varepsilon}[k-1], \boldsymbol{\varepsilon}[k-2], ...]^T$ es un vector que contiene las realizaciones del ruido blanco superpuesto a la salida en los instantes *k-1* y anteriores.

Para modelar este tipo de sistemas, vamos a ampliar los modelos lineales vistos en el apartado anterior, extendiéndolos al caso no lineal mediante la sustitución de la transformación lineal por un aproximador funcional no lineal:

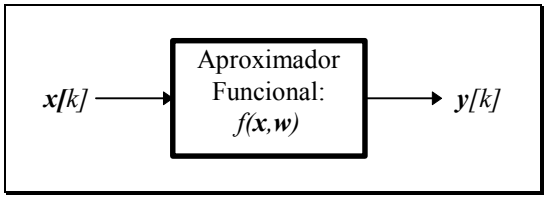

Figura 3.2: Extensión de los modelos lineales al caso no lineal

de tal forma que obtendremos una estimación de la salida del proceso de la forma:

$$\boldsymbol{y}[k] = f(\boldsymbol{x}[k], \boldsymbol{w})$$

Ecuación 3.30

Bajo esta perspectiva, el vector de salidas estimadas en el instante k ($\boldsymbol{y}[k] \in \Re^m$)[i] se obtendrá a partir de toda la información relevante disponible en el mismo instante k, contenida en el vector de entradas $\boldsymbol{x}[k]$. Este vector estará formado por valores de las variables exógenas $\boldsymbol{u} \in \Re^p$ en los instantes k y anteriores (estamos suponiendo que conocemos el valor de estas variables en el mismo instante de la estimación, lo que se justifica en el ámbito del diagnóstico, donde compararemos las salidas estimadas $\boldsymbol{y}[k]$ con las salidas reales $\boldsymbol{d}[k]$), y por los valores de las salidas reales $\boldsymbol{d} \in \Re^m$, de las salidas estimadas $\boldsymbol{y} \in \Re^m$ y de los errores de estimación $\boldsymbol{e}=(\boldsymbol{d}-\boldsymbol{y}) \in \Re^m$ en instantes anteriores.

Como vimos en el Capítulo 2, para ajustar el vector de parámetros mediante la minimización de una función de riesgo basada en el error de estimación (como el error cuadrático medio), bastará con ser capaces de calcular las derivadas de las salidas del modelo respecto a cada uno de sus parámetros: $d\boldsymbol{y}/d\boldsymbol{w}$.

Cuando ninguna de las componentes del vector de entradas del aproximador funcional depende del vector de parámetros $\boldsymbol{w}$ del modelo (no existen realimentaciones), el valor de estas derivadas vendrá determinado por las derivadas de la propia función j del aproximador.

Sin embargo, cuando algunas de las entradas del aproximador funcional depende del vector de parámetros, debido a una realimentación de las salidas del modelo (como será el caso de los modelos NOE y NARMAX, extensión al campo no lineal de los modelos OE y ARMAX), el valor de estas derivadas en lazo abierto se verá modificado por la influencia de los parámetros en las entradas recurrentes. Por ello, cuando presentemos los modelos NOE y NARMAX, habrán de deducirse las expresiones analíticas de estas derivadas, para poder adaptarnos al esquema general de aproximación funcional presentado en el Capítulo 2.

Pasemos a continuación a presentar la estructura de los modelos no lineales propuestos. En los esquemas que aparecen a continuación, se han incluido unos módulos de estandarización o tipificación de las señales externas, que tienen como objetivo aplicar una transformación lineal a dichas variables de tal forma que las señales tratadas por los aproximadores funcionales tengan media nula y desviación típica unidad. Estas transformaciones lineales aceleran el proceso de ajuste de los parámetros y permiten aplicar el análisis estadístico de sensibilidades.

[i] Se especifica explícitamente la dependencia de $\boldsymbol{y}$ con $\boldsymbol{w}$ (y no con $\boldsymbol{x}$) para subrayar las posibles recurrencias que aparecerán en los modelos no lineales que serán introducidos.

3.3.1 El modelo NFIR

El sistema no lineal más sencillo (NFIR) es la versión no lineal del sistema de respuesta impulsional finita (FIR). Queda descrito por:

$$\boldsymbol{d}[k] = g(\boldsymbol{u}^{\{k\}}) + \boldsymbol{\varepsilon}[k]$$

Ecuación 3.31

El estimador correspondiente viene dado por (ver Figura 3.3):

$$\boldsymbol{y}[k] = f(\boldsymbol{u}[k], \ldots, \boldsymbol{u}[k - n_u], \boldsymbol{w})$$

Ecuación 3.32

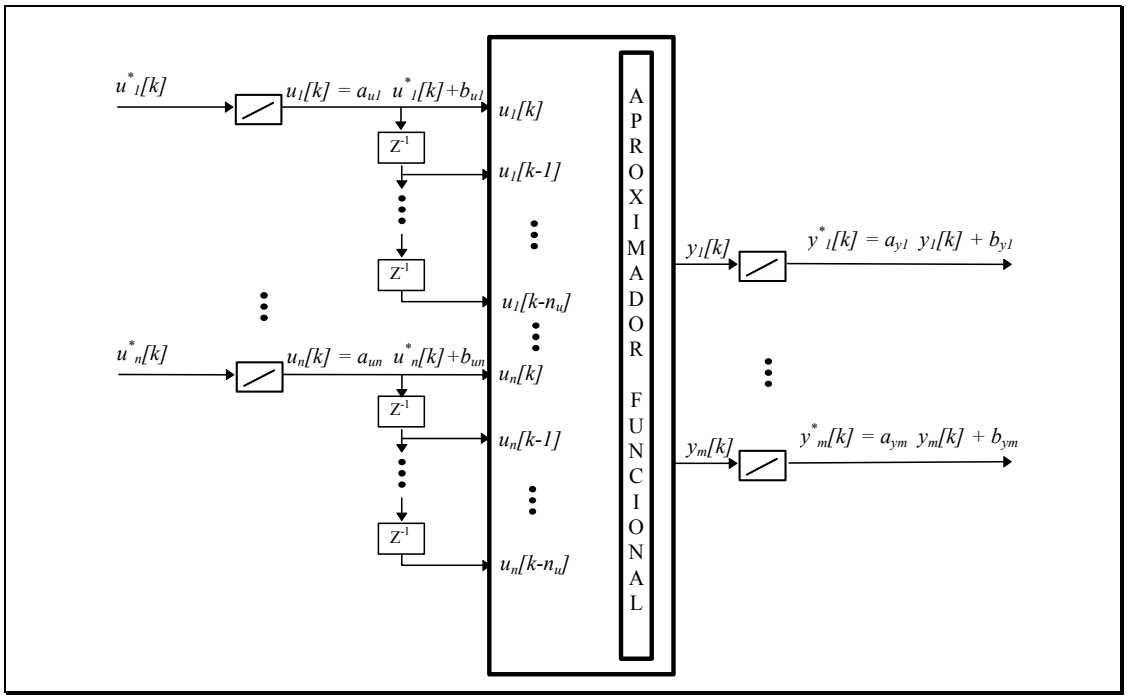

Figura 3.3: Estructura del modelo NFIR

Al no existir conexiones recurrentes, las derivadas de las salidas estimadas respecto de los parámetros del modelo vendrán dadas directamente por las derivadas de las salidas del aproximador funcional respecto de sus parámetros:

$$\frac{\partial y_i[k]}{\partial w} = \frac{\partial f_i}{\partial w}$$

Ecuación 3.33

3.3.2 El modelo NOE

Un sistema no lineal autoregresivo con entradas exógenas (NARX) queda descrito por:

$$\boldsymbol{d}[k] = g(\boldsymbol{d}^{\{k-1\}}, \boldsymbol{u}^{\{k\}}) + \boldsymbol{\varepsilon}[k]$$

Ecuación 3.34

y admite como primer estimador el modelo no lineal de error de salida (NOE):

$$\boldsymbol{y}[k] = f(\boldsymbol{y}[k-1], \ldots, \boldsymbol{y}[k-n_y], \boldsymbol{u}[k], \ldots, \boldsymbol{u}[k-n_u])$$

Ecuación 3.35

Al tratarse de un modelo recurrente, como queda ilustrado en la Figura 3.4, la expresión analítica de las derivadas ***dy/dw*** ya no depende sólo de las derivadas del aproximador funcional *f* respecto de sus parámetros.

La idea general es calcular las derivadas suponiendo constantes las entradas del aproximador (lazo abierto), e incluir posteriormente el efecto de la realimentación calculando la derivada de cada salida respecto de cada entrada recurrente y multiplicando estos valores por las derivadas de las entradas recurrentes respecto de los pesos (estas últimas derivadas habrán sido calculadas en pasos anteriores). Este procedimiento queda formalizado en la Ecuación 3.36:

$$\frac{\partial y_i[k]}{\partial w} = \left.\frac{\partial y_i[k]}{\partial w}\right|_{\substack{u^{\{k\}} \\ y^{\{k-1\}}}} + \sum_{j=1}^{m}\sum_{h=1}^{ny_j} \left.\frac{\partial y_i[k]}{\partial y_j[k-h]}\right|_{\substack{u^{\{k\}} \\ w}} \frac{\partial y_j[k-h]}{\partial w}$$

Ecuación 3.36

Los términos $\left.\frac{\partial y_i[k])}{\partial w}\right|_{\substack{u^{\{k\}} \\ y^{k-1}}} = \frac{\partial f_i}{\partial w}$ son las derivadas de las salidas del aproximador funcional respecto de sus parámetros, manteniendo constantes todas las entradas, mientras que $\left.\frac{\partial y_i[k]}{\partial y_j[k-h]}\right|_{\substack{u^{\{k\}} \\ w}} = \frac{\partial f_i}{\partial x}$ son las derivadas de las salidas respecto de las entradas recurrentes, manteniendo constantes las entradas exógenas y los parámetros.

La Ecuación 3.36 ofrece un algoritmo recurrente para la evaluación de las derivadas en el modelo NOE. Los primeros términos de la serie de derivadas, $\partial y_i[k]/\partial w$, se inicializan a cero para $k=-1, ...,-n_{yj}$ para que el cálculo del gradiente parta siempre del mismo punto inicial.

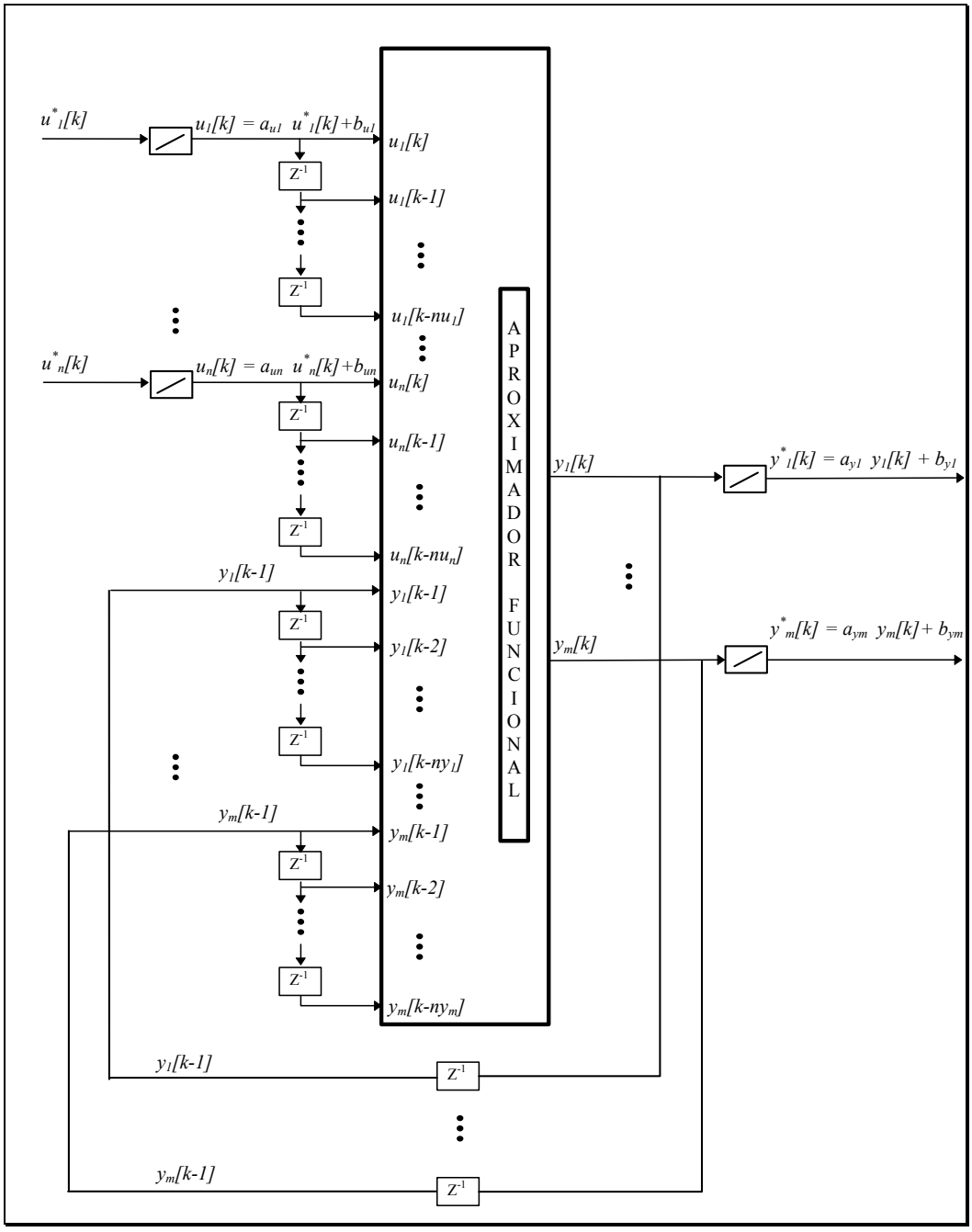

Figura 3.4: Estructura del modelo NOE

3.3.3 El modelo NARX

Como se ha mencionado en el apartado anterior, un sistema NARX queda descrito por:

$$\boldsymbol{d}[k] = g(\boldsymbol{d}^{\{k-1\}}, \boldsymbol{u}^{\{k\}}) + \boldsymbol{\varepsilon}[k]$$

Ecuación 3.37

El estimador NARX utilizado en este caso es:

$$\boldsymbol{y}[k] = f(\boldsymbol{d}[k-1],\ldots,\boldsymbol{d}[k-n_d],\boldsymbol{u}[k],\ldots,\boldsymbol{u}[k-n_u])$$

Ecuación 3.38

donde se han sustituido las entradas recurrentes que utilizaba el modelo NOE (las salidas estimadas) por los valores reales o medidos de las salidas (ver Figura 3.5). De esta forma se evitan las recurrencias, simplificando el cálculo de las derivadas de las salidas estimadas respecto de los parámetros del aproximador::

$$\frac{\partial y_i[k]}{\partial w} = \frac{\partial f_i}{\partial w}$$

Ecuación 3.39

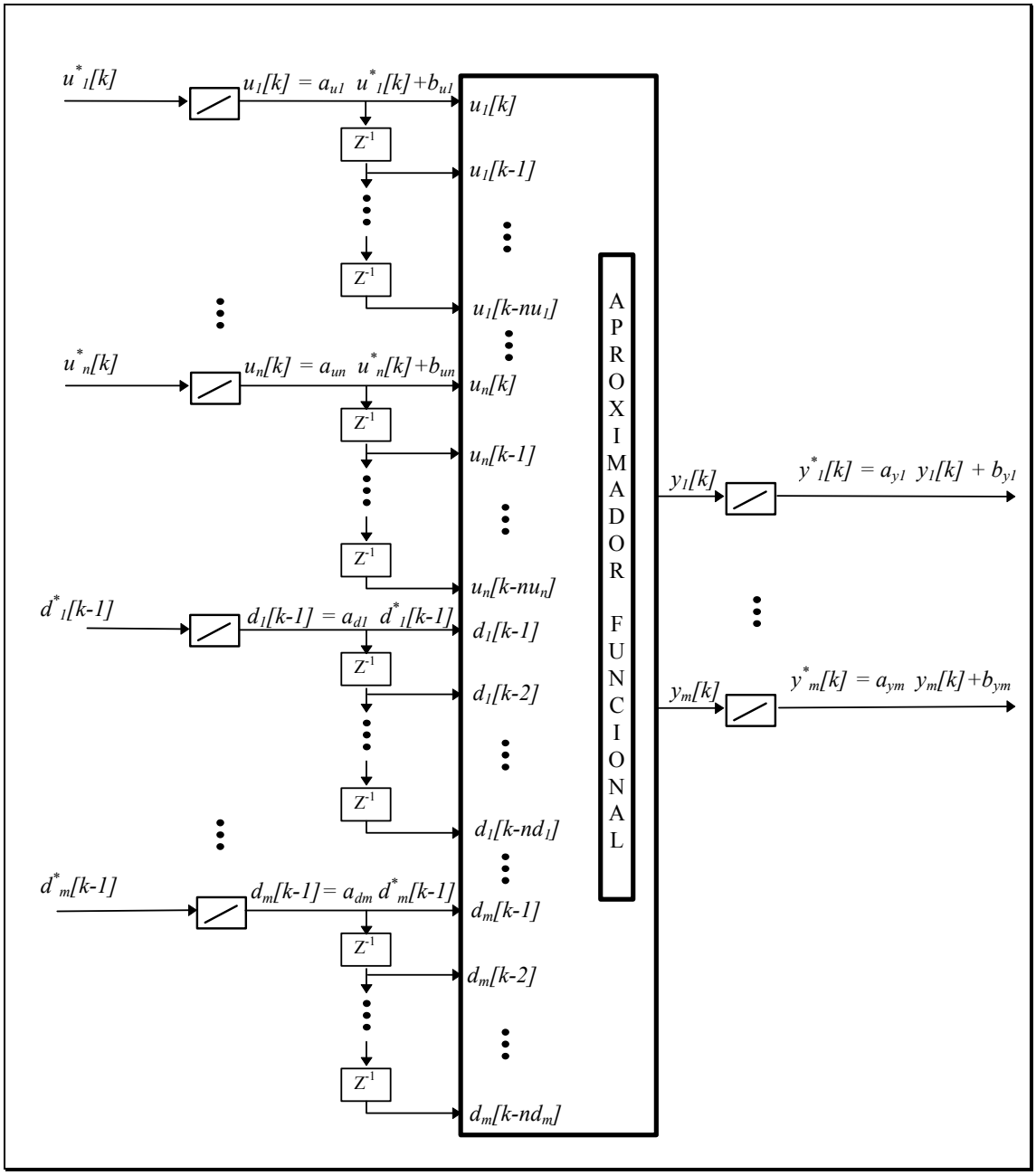

Figura 3.5: Estructura del modelo NARX

3.3.4 El modelo NARMAX

Un sistema no lineal autoregresivo de media móvil con entradas exógenas (NARMAX) queda descrito por:

$$\boldsymbol{d}[k] = g(\boldsymbol{d}^{\{k-1\}}, \boldsymbol{u}^{\{k\}}, \boldsymbol{\varepsilon}^{\{k-1\}}) + \boldsymbol{\varepsilon}[k]$$

Ecuación 3.40

y admite como estimador NARMAX el modelo:

$$\boldsymbol{y}[k] = f(\boldsymbol{d}[k-1], \ldots, \boldsymbol{d}[k-n_d], \boldsymbol{u}[k], \ldots, \boldsymbol{u}[k-n_u], \boldsymbol{e}[k-1], \ldots, \boldsymbol{e}[k-n_e])$$

Ecuación 3.41

Al tratarse de un modelo recurrente (ver Figura 3.6), la expresión analítica de las derivadas de las salidas estimadas respecto de los parámetros del aproximador (*dy/dw*) ya no depende sólo de las derivadas del aproximador funcional *f*. Podemos obtener la expresión analítica de estas derivadas de una forma similar a como se hizo para el modelo NOE:

$$\frac{\partial y_i[k]}{\partial w} = \left.\frac{\partial y_i[k]}{\partial w}\right|_{\substack{u^{\{k\}} \\ e^{\{k-1\}}}} + \sum_{j=1}^{m}\sum_{h=1}^{ne_j} \left.\frac{\partial y_i[k]}{\partial e_j[k-h]}\right|_{\substack{u^{\{k\}} \\ w}} \frac{\partial e_j[k-h]}{\partial w}$$

Ecuación 3.42

Los términos $\left.\frac{\partial y_i[k]}{\partial w}\right|_{\substack{u^{\{k\}} \\ e^{\{k-1\}}}} = \frac{\partial f_i}{\partial w}$ son las derivadas de las salidas del aproximador funcional respecto de sus parámetros manteniendo constantes las entradas (lazo abierto), mientras que $\left.\frac{\partial y_i[k]}{\partial e_j[k-h]}\right|_{\substack{u^{\{k\}} \\ w}} = \frac{\partial f_i}{\partial x}$ son las derivadas de las salidas respecto de las entradas recurrentes, manteniendo constantes las entradas externas y los parámetros del aproximador.

Sustituyendo la expresión del error resulta:

$$\begin{aligned}\frac{\partial e_j[k-h]}{\partial w} &= \frac{\partial\left(a_{ej}\left(d_j^*[k-h]-y_j^*[k-h]\right)\right)}{\partial w}\\ &= \frac{\partial\left(a_{ej}\left(d_j^*[k-h]-\left(a_{yj}\,y_j[k-h]+\,b_{yj}\right)\right)\right)}{\partial w}\\ &= -a_{ej}a_{yj}\frac{\partial y_j[k-h]}{\partial w}\end{aligned}$$

Ecuación 3.43

y sustituyendo en la Ecuación 3.42 queda finalmente:

$$\frac{\partial y_i[k]}{\partial w} = \frac{\partial y_i[k]}{\partial w}\bigg|_{\substack{u^{\{k\}}\\ e^{k\{-1\}}}} - \sum_{j=1}^{m}\sum_{h=1}^{ne_j}\frac{\partial y_i[k]}{\partial e_j[k-h]}\bigg|_{\substack{u^k\\ w}} a_{ei}a_{yi}\frac{\partial y_j[k-h]}{\partial w}$$

Ecuación 3.44

Nos encontramos nuevamente ante un algoritmo recursivo, en el que para calcular las derivadas de las salidas respecto de los parámetros del aproximador en un instante de tiempo determinado, es necesario conocer el valor de estas derivadas en instantes anteriores. Los primeros términos de la serie serán inicializados a cero, para que el cálculo del gradiente parta siempre del mismo punto inicial.

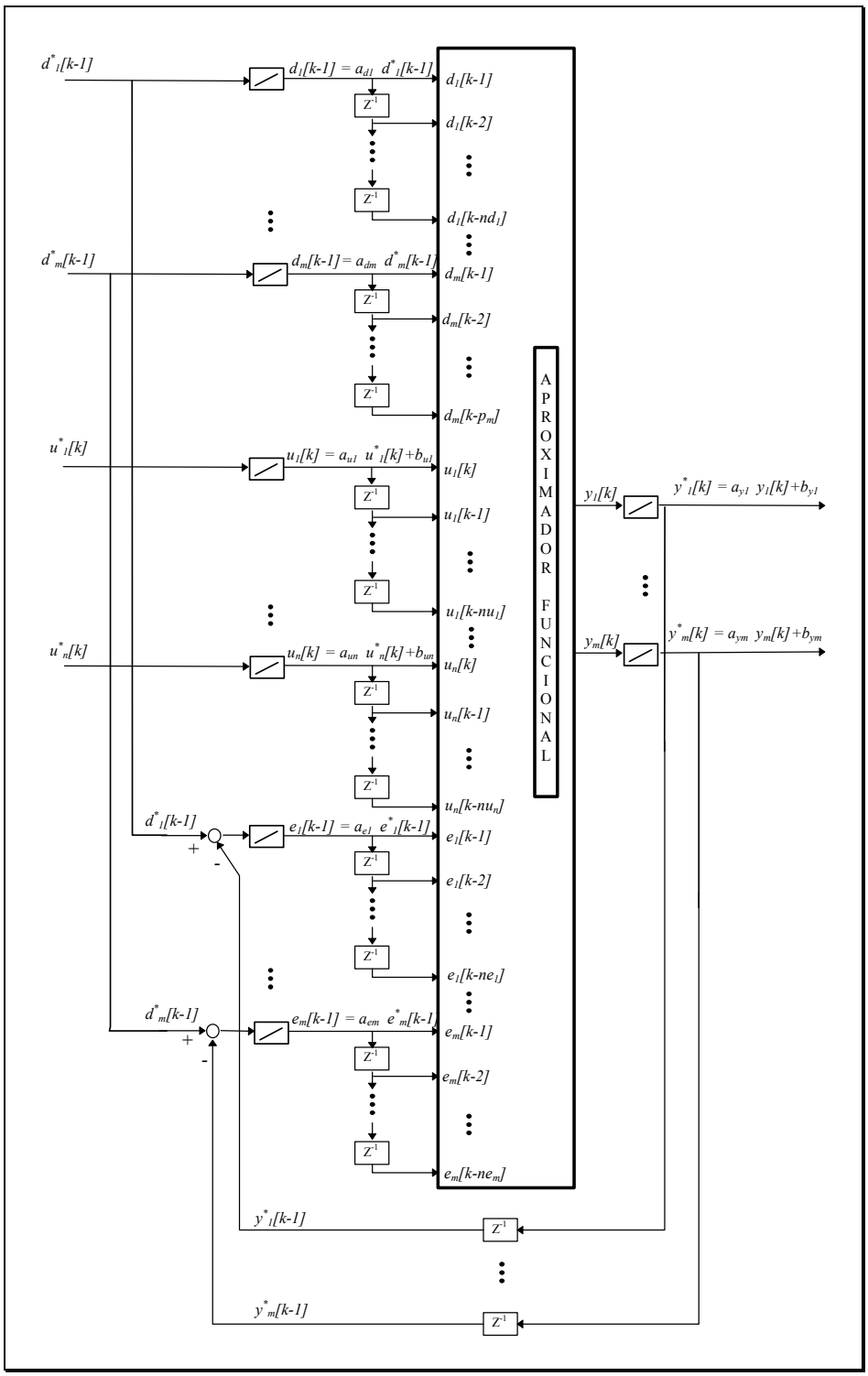

Figura 3.6: Estructura del modelo NARMAX

3.4 Selección del modelo

El modelado de procesos dinámicos no lineales aplicado al caso concreto de la detección de anomalías por análisis de residuos, tiene como objetivo principal modelar el comportamiento considerado como normal del componente bajo estudio, de tal forma que cualquier desviación significativa de la salida estimada de la real, será considerada como resultado de una situación anómala.

Para poder evaluar cuantitativamente esta desviación, es necesario disponer de las medidas de las variables que han sido seleccionadas como salidas de los modelos de funcionamiento normal, con el fin de compararlas con los valores estimados.

La disponibilidad de estas medidas hace posible que se puedan utilizar los modelos NARX y NARMAX, que requieren las medidas de valores pasados de las variables que se están estimando. Esta posibilidad permite limitar el conjunto de modelos candidatos no lineales a los modelos NARX y NARMAX, evitando las recurrencias del modelo NOE y tratando el modelo NFIR como un caso particular del NARX.

3.4.1 Modelos a ensayar

El proceso de selección del modelo seguirá la máxima de "probar primero lo más sencillo". Siguiendo esta pauta, el proceso de selección comenzará ajustando un modelo lineal del tipo ARX[i], cuyo orden se establecerá a partir del análisis de las funciones de correlación simple y de correlación parcial. La validación del modelo resultante determinará la necesidad o no de ajustar un modelo ARMAX para modelar el ruido. Si los resultados obtenidos con los modelos lineales no son suficientemente buenos para nuestros propósitos (lo que será determinado por validación cruzada o la detección de no linealidades), pasaremos a los modelos no lineales. En cualquier caso el estudio lineal habrá servido para determinar el punto de partida en cuanto a variables de entrada a tener en cuenta, desde un punto de vista meramente cualitativo.

El modelo NARX será el primer candidato a ensayar tras haber rechazado los modelos lineales. En el caso lineal, los modelos ARX eran capaces de modelar cualquier sistema lineal con tal de hacer $n_d, n_u \to \infty$. En el caso no lineal, los modelos NARX jugarán el mismo papel (*[Sjoberg, 1995]*), siempre que el aproximador funcional utilizado tenga la capacidad de representación funcional requerida.

[i] Supondremos que la serie es estacionaria, o ha sido hecha estacionaria

Las ventajas del modelo NARX sobre otros modelos no lineales son:

- Es un modelo general, que puede describir cualquier sistema no lineal representado por la Ecuación 3.29.

- Es un modelo no recurrente, en el que el vector de entradas del aproximador funcional (el regresor $\boldsymbol{x}$) no depende de sus parámetros $\boldsymbol{w}$. Esta característica hace que el ajuste de los parámetros del modelo NARX sea mucho menos costoso que en el caso de modelos recurrentes.

Sin embargo presenta también las siguientes desventajas:

- Una buena aproximación del sistema a modelar y de las características del ruido puede requerir un elevado número de valores pasados de las entradas exógenas y de las salidas reales.

- Al no realimentar las salidas estimadas, ha de modelar conjuntamente la dinámica del sistema y la del ruido.

Para corregir estos últimos defectos en aquellos casos en los que el ruido de la serie tiene un protagonismo relevante (esta situación puede detectarse mediante la validación del modelo NARX), es necesario realimentar las salidas estimadas de tal forma que el aproximador funcional admita también como entradas los errores de estimación. De esta forma pasamos al modelo NARMAX.

Las ventajas del modelo NARMAX son:

- Es un modelo general, que puede describir cualquier sistema no lineal representado por la Ecuación 3.29.

- Al crear una representación más compacta, requiere menos parámetros que el modelo NARX para modelar las características del ruido.

Su principal desventaja:

- Al ser un modelo recurrente, la estimación de los parámetros de su aproximador es muy costosa.

Una posible manera de acelerar el ajuste paramétrico de la estructura NARMAX (Figura 3.6) es simplificar su estructura desligando las dinámicas del sistema y del ruido. Si además suponemos un modelo lineal de ruido, el modelo NARMAX resultante sería equivalente a un NARX al que se le añade una combinación lineal de errores pasados de estimación (*[Chen & Billings, 1989]*), resultando el estimador:

$$\boldsymbol{y}[k] = f(\boldsymbol{d}^{\{k-1\}}, \boldsymbol{u}^{\{k\}}) + \boldsymbol{C}(z)\boldsymbol{e}[k]$$

Ecuación 3.45

El ajuste de este modelo simplificado resulta mucho más sencillo que el del modelo NARMAX completo, ya que las conexiones recurrentes no atraviesan al aproximador funcional no lineal.

3.4.2 Selección de las variables de entrada

La selección de las variables de entrada en el caso lineal es un tema muy estudiado en la literatura estadística (*[Hocking, 1976]*, *[Thompson, 1978]*, *[Linhart & Zucchini, 1986]*. La idea general de estos algoritmos de selección de variables es evaluar el crecimiento o decrecimiento de una medida de la bondad del modelo lineal ajustado al incluir o excluir un subconjunto de variables de entrada.

A modo de ejemplo, podemos citar los métodos de selección hacia delante y eliminación hacia atrás ("forward selection" y "backward elimination"). El primero de ellos va incluyendo las variables de entrada (distintos retardos de las señales disponibles) de acuerdo con la máxima correlación parcial entre la salida a estimar y las variables que todavía no han sido incluidas. A cada paso, se realiza un test estadístico para estimar el nivel de significación de la inclusión de la nueva variable. El proceso de selección se detiene cuando la adición de una nueva variable no es significativa, estadísticamente hablando. El segundo de ellos sigue el camino opuesto, eliminando una variable a cada paso, y estimando la significación de esta poda. Los dos algoritmos arrojan resultados diferentes.

Otros métodos de selección parecidos son el ránking hacia delante y hacia atrás ("forward and backward ranking") y la regresión paso a paso ("stepwise regression"). La metodología de Box y Jenkins (*[Box & Jenkins, 1976]*) es en sí misma también un método de selección de variables.

En nuestro caso basaremos la selección de las variables de entrada de los modelos no lineales en el Análisis Estadístico de Sensibilidades AES presentado en el Capítulo 2 (*[Muñoz & Czernichow, 1995]*).

Cuando la componente lineal del sistema a modelar sea importante, podremos utilizar como punto de partida para el ajuste de los modelos no lineales el estudio lineal que le ha precedido. De este análisis tomaremos el conjunto inicial de variables de entrada del aproximador funcional. Sin embargo, cuando la componente lineal sea débil, el análisis lineal puedrá resultar engañoso y será preferible

establecer el conjunto inicial de variables de entrada por criterios heurísticos basados en el conocimiento experto. Una vez que el modelo no lineal ha sido ajustado, aplicaremos el AES para eliminar del modelo todas aquellas variables de entrada que no son relevantes para predecir la salida. El modelo será entonces reajustado y validado.

En el caso de rechazo por validación cruzada, podremos incrementar el orden del modelo incluyendo nuevos retardos de las variables de entrada, volver a ajustarlo y aplicar de nuevo el AES. Este proceso se repite hasta que la validación resulte positiva, o hasta alcanzar un orden máximo preestablecido.

3.5 Validación de modelos no lineales

La validación de modelos no lineales suele realizarse por validación cruzada. Para ello es necesario disponer de un conjunto de datos suficientemente representativo, formado por datos no utilizados durante el ajuste de los parámetros del aproximador. El ensayo del modelo sobre este conjunto de datos permitirá darlo por bueno o rechazarlo. Esta decisión puede estar basada en una medida de la precisión de la aproximación (como es el error cuadrático medio utilizado como función de coste en el ajuste de los parámetros), o en otro tipo de medidas de carácter más práctico que evalúen el cumplimiento de los objetivos finales del modelo (como sería la eficacia de la detección de anomalías en nuestro caso).

Este método de validación tiene la gran ventaja de no haber tenido que hacer hipótesis sobre las distribuciones probalísticas de las variables aleatorias involucradas (*[Ljung & Sjöberg, 1992]*). Su desventaja más palpable es tener que dejar en reserva un conjunto de datos, y no utilizarlo para la estimación de los parámetros. Esta desventaja desaparece en el caso de disponer de un conjunto de entrenamiento suficientemente rico, como suele ser el caso en la obtención de modelos de comportamiento normal en el ámbito del diagnóstico.

Existen otros métodos de validación de modelos no lineales, basados en el análisis de las correlaciones existentes entre el error de estimación $\boldsymbol{e}[k]$ y la información disponible a la hora de predecir $\boldsymbol{y}[k]$. Si se detecta una correlación significativa (realizando un test estadístico) entre el error de estimación y alguna función de las variables disponibles para la predicción ($\boldsymbol{d}^{\{k-1\}}$, $\boldsymbol{u}^{\{k\}}$, $\boldsymbol{e}^{\{k-1\}}$), entonces podremos concluir que el error de predicción no es un ruido blanco y que el modelo no es el mejor posible (*[Sjöberg, 1995]*).

Una vez que hayan sido introducidos los aproximadores funcionales neuronales, se presentarán en el Capítulo 5 dos ejemplos de aplicación de estas técnicas al modelado de procesos dinámicos no lineales.

4. Introducción a las Redes Neuronales Artificiales de aproximación funcional

Este capítulo es una introducción al campo de las Redes Neuronales Artificiales (RNA), y más concretamente, a las RNA supervisadas que serán utilizadas como aproximadores funcionales.

No pretende ser una revisión exhaustiva de estructuras conexionistas, sino una breve introducción a este nuevo campo del tratamiento de la información y una presentación clara y concisa del modelo conexionista supervisado que mayor aplicación está teniendo en nuestros días: el perceptrón multicapa (*"Multilayer Perceptron"*, PM).

El capítulo siguiente será una continuación de éste, donde se presentará una nueva familia de RNA supervisadas derivadas de las llamadas redes de funciones base radiales (*"Radial Basis Function Network"*, RBFN).

4.1 Origen de las Redes Neuronales Artificiales

Las Redes Neuronales Artificiales (RNA) surgieron de la observación del funcionamiento del cerebro y de su comparación con la forma de trabajar de los ordenadores digitales. Los elementos estructurales constituyentes del cerebro, las neuronas biológicas (*[Ramón y Cajal, 1911]*), son unos seis órdenes de magnitud más lentas que las puertas lógicas de silicio, ofreciendo tiempos de respuesta del orden del milisegundo frente a los vertiginosos tiempos de respuesta del orden del nanosegundo que tienen ciertos dispositivos digitales actuales.

Esta relativa lentitud no impide al cerebro realizar ciertas funciones habituales, como son las tareas de reconocimiento, percepción y control motriz, con una eficacia y rapidez fuera del alcance del mejor ordenador digital. Esta supremacía le viene dada por la alta complejidad de su estructura, formada por un elevado número de células nerviosas (neuronas) masivamente interconectadas y funcionando en paralelo. Se estima que el número de neuronas en el córtex humano es del orden de diez millares de millones (10^{10}), y que el número de sinapsis o conexiones es del orden de 60 billones (60 10^{12}). La alta complejidad de esta estructura biológica no sería operativa si no fuese además eficiente. La eficiencia energética del cerebro es aproximadamente de 10^{-16} julios por operación por segundo, mientras que los mejores ordenadores no superan los 10^{-6} julios por operación por segundo (*[Faggin, 1991]*).

Al nacer, el cerebro ya tiene gran parte de su estructura formada, y lo que es más importante, está dotado de los mecanismos necesarios para construirse sus propias reglas a través de lo que conocemos como "experiencia". Este proceso de aprendizaje se extiende a lo largo de los años, aunque el desarrollo más espectacular tiene lugar durante los dos primeros años de vida, durante los cuales se forman alrededor de un millón de sinapsis por segundo. La adaptación del sistema nervioso a su entorno sigue dos mecanismos distintos en un cerebro adulto: la creación de nuevas conexiones sinápticas y la modificación de las conexiones existentes.

Podemos pues concluir que el cerebro humano es una estructura compleja, no lineal y paralela de tratamiento de información que almacena su conocimiento en las conexiones que ligan a sus elementos de proceso (neuronas), y que es capaz de adaptarse a su entorno. Las RNA están inspiradas de la estructura del cerebro, y fueron concebidas para resolver cierto tipo de problemas especialmente mal resueltos por las técnicas de programación tradicionales. Formalmente, computación neuronal es la disciplina tecnológica que trata sistemas paralelos y adaptativos de procesamiento de información distribuida, que desarrollan sus capacidades bajo su exposición a un entorno de información.

Esta definición muestra claramente el paralelismo entre la estructura cerebral biológica y las RNA. Este paralelismo ha motivado a muchos investigadores de diversos campos de la ciencia a profundizar en la interpretación biológica de las RNA, proponiendo nuevas estructuras conexionistas y estrategias de aprendizaje directamente inspiradas de la modelización del cerebro. Estos estudios han dado resultados interesantes en el campo de las RNA, pero desde un punto de vista subjetivo, lo más importante de ellos es el conocimiento que directa o indirectamente están aportando para esclarecer los complejos procesos de aprendizaje, percepción y gestión del conocimiento que tienen lugar en el cerebro humano.

Una definición más concreta de RNA es la propuesta por Robert Hecht Nielsen en *[Hecht-Nielsen, 1990]*, y que dice así:

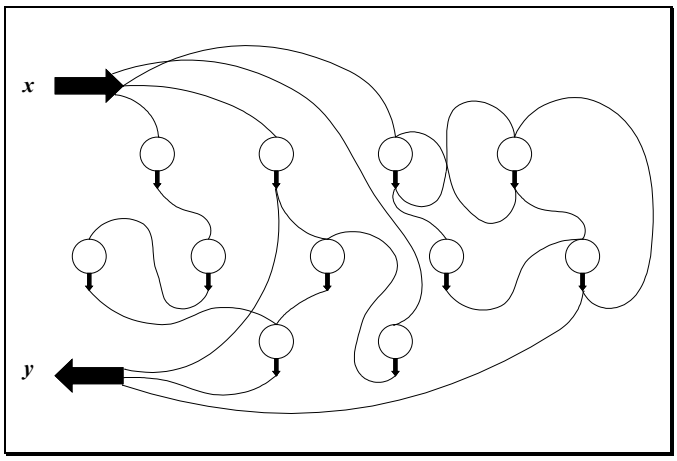

Figura 4.1: Estructura general de una RNA

"Una RNA es una estructura de procesamiento paralelo de información distribuida, bajo forma de grafo orientado (ver Figura 4.1), con las siguientes subdefiniciones y restricciones:

- Los nodos del grafo son llamados *elementos de proceso* o *neuronas*.
- Las uniones del grafo son llamadas *conexiones* (unidireccionales e instantáneas).
- Se admite cualquier número de conexiones de entrada.
- Cada elemento de proceso sólo puede tener una señal de salida, que puede ramificarse en cualquier número de conexiones de salida.
- Los elementos de proceso pueden tener *memoria local*.

- Cada elemento de proceso tiene una *función de transferencia* que puede utilizar y alterar la memoria local, puede usar las señales de entrada, y produce la señal de salida. La función de transferencia puede actuar de forma continua o discreta en el tiempo. Si actúa de forma discreta, existirá una entrada de estímulo que active la aplicación de la función de transferencia proveniente de la unidad de control de la RNA".

La figura siguiente muestra el esquema general de un elemento de proceso:

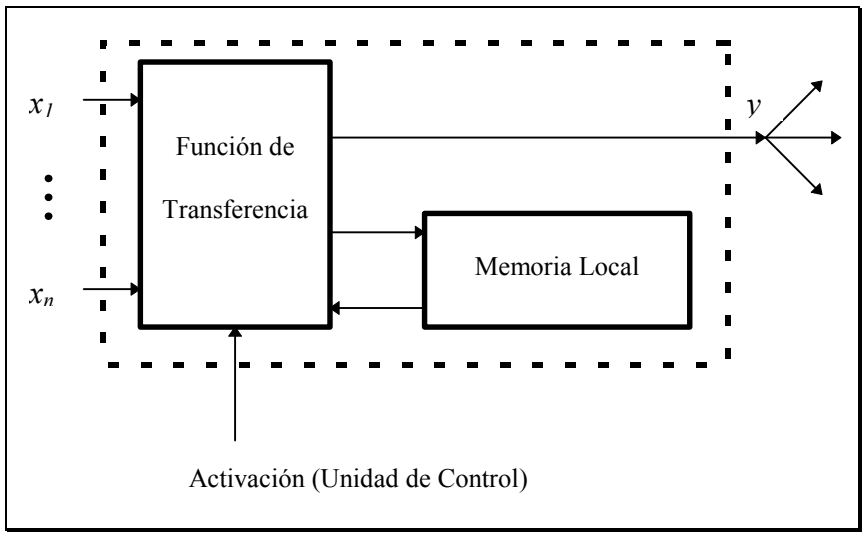

Figura 4.2: Estructura de un elemento de proceso

Podemos pues concluir que las RNA son estructuras adaptativas de procesamiento de información inspiradas en la estructura cerebral, donde el procesamiento se lleva a cabo mediante la interconexión de elementos de proceso muy sencillos a los que llamaremos neuronas. Esta arquitectura da lugar a estructuras altamente paralelizables donde el flujo de información no sigue un camino secuencial, sino que se distribuye a través de las conexiones de los elementos de proceso donde la información es tratada.

Durante la fase de aprendizaje, la RNA se expone a un entorno de información para que pueda adaptar sus pesos (parámetros libres que dan forma a las funciones de transferencia de sus elementos de proceso) y posiblemente también su estructura. Durante la fase de evaluación, los pesos de la red se mantienen fijos y la RNA se limita a tratar la información de entrada que se le suministra.

4.2 Breve recorrido histórico

Los comienzos

- S. XIX Se plantea a nivel teórico la posibilidad de modelar la fisiología del cerebro, teniendo como objetivo la creación de un modelo capaz de reproducir procesos del razonamiento humano.

- 1943 Warren Mc Culloch y Walter Pitts publican la formalización de la descripción de una neurona artificial que sigue siendo hoy en día elemento constitutivo de las RNA más complejas. No contemplaron sin embargo aplicaciones prácticas.

- 1949 Donald Hebb publica "La organización del comportamiento", donde afirma que el condicionamiento psicológico es una propiedad de las neuronas a nivel individual. Propone una ley de aprendizaje que le permitió explicar cualitativamente algunos ejemplos experimentales de carácter psicológico.

- 1951 Marvin Minsky construye el primer neurocomputador ("Snark"). Resultó un éxito desde el punto de vista técnico, pues ajustaba automáticamente sus pesos, pero no llegó a darle ninguna aplicación práctica.

Primeros éxitos

- 1957-1958 Frank Rosenblatt, Charles Whightman y otros, construyen el primer neurocomputador ("Mark I Perceptron") con aplicación práctica al reconocimiento de patrones. Rosenblatt publica "Principios de Neurodinámica".

- 1960 Bernard Widrow y Marcian E. Hoff desarrollan la ADALINE y la dotan de una potente ley de aprendizaje. Su aplicación práctica fue extensa. Widrow funda también la "Memistor Corporation".

Etapa de crisis

La falta de rigor analítico que se produjo al basar todos los trabajos en la mera experimentación, unido a un entusiasmo descontrolado, fueron el origen de esta etapa.

- 1969 Minsky y Papert publican "Perceptrons", donde se prueba que un perceptrón no puede resolver problemas linealmente no separables como es el caso de la función XOR. Llevan a cabo una verdadera campaña anti RNA.

Momentos de silencio

- 1967-1982 La investigación en este campo en los EEUU se ve acallada por el libro "Perceptrons". Sin embargo se consolidan importantes pilares de las RNA bajo los nombres de proceso adaptativo de señales, reconocimiento de patrones y modelado biológico. Pertenecen a esta época Shun-ichi Amari, James Anderson, Kunihiko Fukushima, Stephen Grossberg, Harry Klopf, Teuvo Kohonen y David Willshaw.

El despegue

- Principios de los 80 .- Se produce el resurgimiento de la computación neuronal en EEUU, en parte financiado por la "Defense Advanced Research Projects Agency" (DARPA).
Por otro lado Hopfield, físico de gran reputación, se dedica entre 1983 y 1986 a relanzar la computación neuronal mediante publicaciones y conferencias.

- 1986 Publicación de "PDP Books" (Parallel Distributed Processing, Vol. I and II) editados por David Rumelhart y James Mc Clelland que supone un verdadero acontecimiento por la presentación del método de retropropagación (*"backpropagation"*).

- 1987 IEEE International Conference on Neural Networks con 1700 participantes (San Diego)
Se crea la International Neural Networks Society (INNS)

- 1988 Publicación de la revista "Neural Networks" por el INNS.

- 1989 Publicación de la revista "Neural Computation"

- 1990 Publicación de la revista "Transactions on Neural Networks" por el IEEE.

4.3 Ventajas de las RNA

Las RNA deben su capacidad de procesamiento de información a su estructura distribuida y paralela (la información queda almacenada en los elementos de proceso de la red de forma no centralizada) y a su capacidad de aprendizaje y por tanto de generalización (en contraposición con la memorización).

Estas dos capacidades de procesamiento hacen que las RNA sean capaces de resolver cierto tipo de problemas muy complejos (tanto por su tamaño como por su estructura) que hasta el momento no habían quedado resueltos de forma satisfactoria.

Hay que dejar claro que las RNA no son una nueva metodología de ingeniería para la resolución global de problemas. Son herramientas de tratamiento de información que pueden integrarse fácilmente en arquitecturas modulares, para resolver de forma muy eficaz aquellas subtareas precisas del problema global que mejor se adaptan a sus capacidades. Entre estas subtareas caben citar procedimientos de reconocimiento de patrones, de aproximación funcional y de memorias asociativas.

Las propiedades o características de las RNA que suelen ser más útiles son:

- No linealidad: las neuronas son elementos de proceso generalmente no lineales. La interconexión de estos elementos genera estructuras de transformación de datos donde este carácter no lineal queda distribuido a lo largo y ancho de la red. Esta característica permite modelar procesos intrínsecamente no lineales (como por ejemplo el reconocimiento del discurso hablado) pero complica también los métodos de análisis de las estructuras resultantes, impidiendo la aplicación de técnicas de análisis bien establecidas como son las de los sistemas lineales.

- Modelado de relaciones de entrada/salida: un paradigma de aprendizaje especialmente extendido y útil para los objetivos de esta tesis es el llamado aprendizaje supervisado. En este tipo de aprendizaje se dispone de un conjunto de muestras de la relación entrada/salida a modelar, formado por pares (entradas, salidas deseadas), que permite optimizar los pesos de la red de tal forma que se espera que la relación de entrada/salida generada sea capaz de reproducir casos no representados en el conjunto de datos original. Esta capacidad de generalización se obtiene utilizando estructuras de aproximación funcional con capacidad de representación universal, y estrategias de aprendizaje como las descritas en el Capítulo 2, que cuidan de un modo especial el no sobrepasar el límite del sobre-entrenamiento.

- Adaptabilidad: las RNA son por definición estructuras adaptativas capaces de ajustar sus pesos, y por tanto su función de transferencia, a cambios en el entorno.

Esta característica las hace particularmente útiles en el tratamiento de procesos no estacionarios, donde pueden diseñarse estrategias de aprendizaje en tiempo real para que el modelo conexionista se vaya adaptando de forma continua a los cambios del proceso en cuestión. En el diseño de sistemas adaptativos hay que tener siempre en cuenta las constantes de tiempo del sistema de adaptación y las del sistema bajo estudio: un sistema adaptativo con constantes de tiempo demasiado bajas puede responder demasiado rápido de tal forma que no sea capaz de ignorar perturbaciones espúreas y se haga inestable. En el caso concreto del diagnóstico basado en modelos conexionistas de funcionamiento normal esta capacidad de adaptación permitirá al sistema de diagnóstico el irse acomodando al lento proceso de envejecimiento de los componentes. Otros campos de aplicación relevantes son el reconocimiento adaptativo de patrones, el procesamiento adaptativo de señales y el control adaptativo.

- Respuesta evidencial: en el ámbito de aprendizaje supervisado (para aproximación funcional y clasificación), una RNA puede, además de estimar la salida deseada, dar una medida de la fiabilidad de la estimación. Esta información puede ser utilizada para rechazar patrones de entrada, completando de esta forma el proceso de estimación.

- Tolerancia frente a fallos: una red neuronal realizada en *"hardware"* tiene la capacidad de seguir respondiendo de forma no catastrófica cuando parte de su estructura está dañada. Esto es debido al tratamiento distribuido de la información y a la redundancia implícita en su estructura.

- Realización en VLSI: la naturaleza masivamente paralela de las RNA las hace potencialmente eficaces para la realización de ciertas tareas complejas. Esta misma característica, junto con la uniformidad de su estructura, las hace especialmente adecuadas para su construcción en tecnología de integración VLSI (*"Very Large Scale Integration"*). Esto permite construir estructuras extremadamente complejas (en cuanto a su tamaño) que de otro modo serían inviables.

4.4 Principales estructuras conexionistas

Dado el enorme abanico de estructuras conexionistas existentes, nos vamos a limitar en este apartado a enumerar las más relevantes al enfoque de esta tesis, haciendo un especial hincapié en sus respectivos campos de aplicación (*[Maren et al, 1990]*).

Para cada una de ellas se especificará sus creadores o desarrolladores más decisivos, su fecha de aparición o desarrollo, las referencias bibliográficas más relevantes, una breve descripción de su estructura, sus aplicaciones, y sus ventajas e inconvenientes.

Para obtener una visión general de las redes neuronales se recomienda la consulta de los artículos de introducción *[Lippmann, 1987]* y *[Hush & Horn, 1992]*. El primero de ellos es una referencia clásica pero un poco anticuada, mientras que el segundo presenta una visión más actualizada.

Como libros de consulta de carácter general se recomienda *[Hecht-Nielsen, 1989]* y *[Haykin, 1994]*. Este último es sin lugar a dudas uno de los mejores tratados que se han escrito sobre redes neuronales artificiales, no por sus aportaciones originales, sino por la riqueza de su índice de materias, y por la claridad y seriedad de su exposición.

Desde un punto de vista práctico, cabe mencionar el paquete de simulación de redes neuronales de Matlab™, desarrollado por la compañía Math Works, Inc. (*[Demuth & Beale, 1992]*). Este paquete permite obtener, de una forma muy sencilla, una primera experiencia práctica en este campo, y viene dotado de una serie de programas de demostración que ilustran gráficamente las capacidades de esta herramienta.

A un nivel más avanzado existen varias herramientas de simulación de RNA de dominio público, generalmente desarrolladas en el seno de universidades europeas y americanas (*[Luzzy & Dengel, 1993]*). Entre estas herramientas cabe destacar la desarrollada por la universidad de Stuttgart ("Stuttgart Neural Net Simulator") que está disponible via *ftp* anónimo y funciona bajo entornos *X Windows*.

Pasemos a continuación a presentar las estructuras conexionistas más relevantes, comenzando por las estructuras aplicables a problemas de aproximación funcional y de reconocimiento de patrones, para terminar con las estructuras de hetero y auto asociación.

4.4.1 ADALINE/MADALINE (*"Many Adaptive LINear Elements"*)

Creadores: B. Widrow y M.E. Hoff
Fecha: 1960
Referencias: *[Widrow & Hoff, 1960], [Widrow & Winter, 1988]*

Estructura:

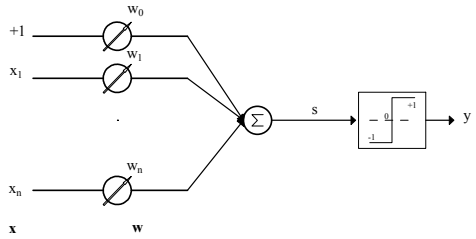

Figura 4.3: Estructura ADALINE

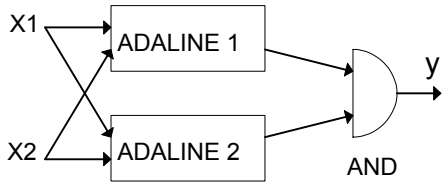

Figura 4.4: Ejemplo de estructura MADALINE

Aplicaciones: - Filtrado adaptativo de señales.
- Ecualización adaptativa
- Reconocimiento de patrones

Ventajas: Su gran sencillez y homogeneidad les hacen fácilmente realizables en tecnología VLSI.

Desventajas: Sólo son capaces de resolver problemas de clasificación linealmente separables y llevar a cabo transformaciones lineales.

4.4.2 Perceptrón Multicapa: PM (*"Multilayer Perceptron"*)

Creadores: P.J. Werbos
D. Parker
D. Rumelhart
Fecha: 1974-1986
Referencias: *[Werbos, 1974] [Werbos, 1988] [Werbos, 1989] [Parker, 1985][Parker, 1987] [Rumelhart et al., 1986a] [Rumelhart et al., 1986b]*

Estructura:

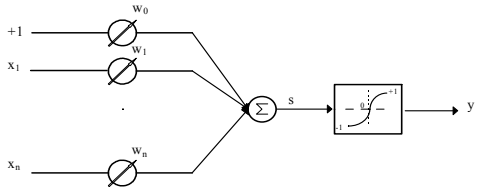

Figura 4.5: Estructura del Perceptrón

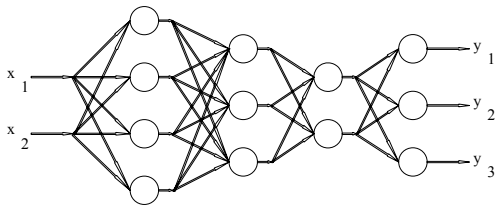

Figura 4.6: Estructura del Perceptrón Multicapa

Aplicaciones:
- Aproximación funcional
- Reconocimiento de patrones
- Filtrado de señales
- Eliminación de ruido
- Segmentación de imágenes y señales
- Control adaptativo
- Compresión de datos
- etc.

Ventajas: Capacidad de representación funcional universal. Gran rapidez de procesamiento. Genera buenas representaciones internas de las características de los datos de entrada. Ampliamente estudiada. Es la estructura conexionista que más se ha aplicado en la práctica.

Desventajas: Tiempo de aprendizaje elevado para estructuras complejas.

4.4.3 Red de funciones base radiales: RBFN (*"Radial Basis Function Network"*)

Creadores: M.J.D. Powell
D.S. Broomhead y D. Lowe
Fecha: 1985-1988
Referencias: *[Powell, 1985]*
[Broomhead & Lowe, 1988]

Estructura:

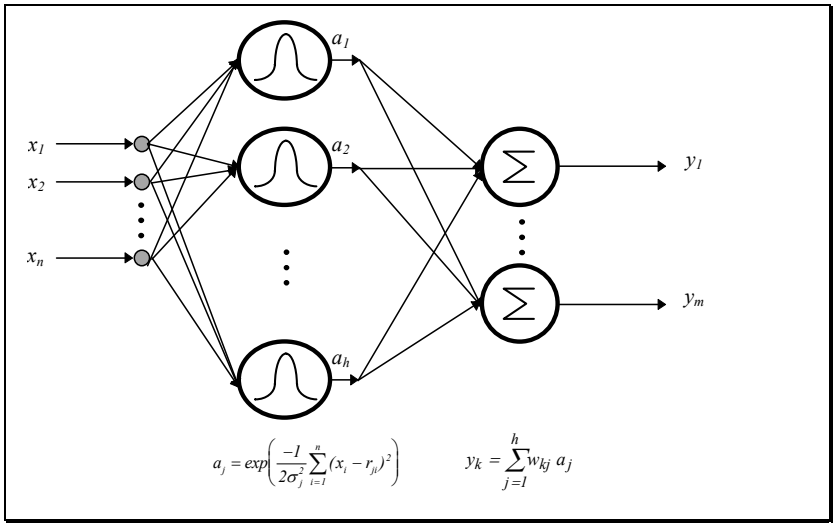

Figura 4.7: Estructura de la red RBFN

Aplicaciones: - Aproximación funcional
- Reconocimiento de patrones

Ventajas: Capacidad de representación funcional universal. La estructura de esta red tiene una interpretación matemática directa, lo que permite realizar una buena inicialización de los pesos de la red, y extraer conocimiento de las estructuras ajustadas. La buena inicialización de los pesos acelera el proceso de aprendizaje. Además de estimar la salida deseada, esta estructura permite estimar de forma simultánea la función de densidad probabilista del vector de entradas.

Desventajas: El procesamiento realizado es algo más complejo que en el caso del perceptrón multicapa.

4.4.4 Red de ligaduras funcionales ("*Functional-link network*")

Creadores: Y.H. Pao
Fecha: 1988
Referencias: *[Pao, 1989]*

Estructura:

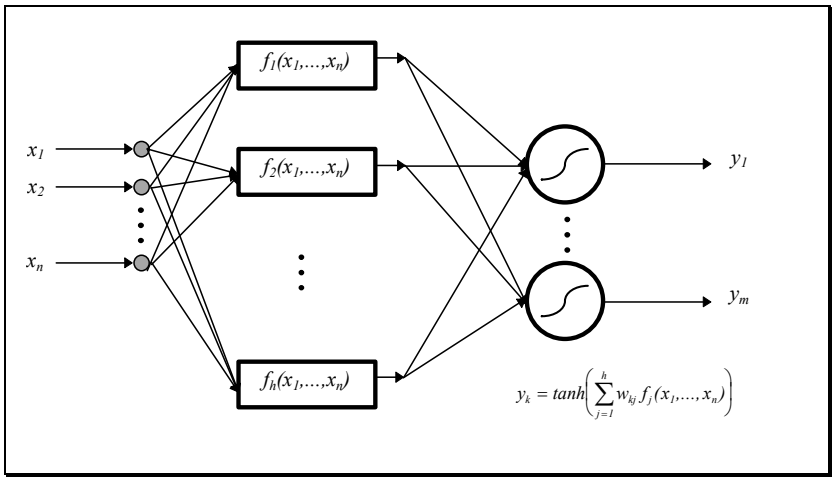

Figura 4.8: Estructura de la red de ligaduras funcionales

El principio de esta red está basado en la aproximación mediante la superposición de funciones no lineales de las variables de entrada. Un ejemplo típico que permite resolver el problema de la paridad binaria utiliza como funciones de ligadura el conjunto: $\{f_1(\mathbf{x})=x_1, f_2(\mathbf{x})=x_2, f_3(\mathbf{x})=x_1\, x_2\}$.

Aplicaciones: - Aproximación funcional
- Reconocimiento de patrones

Ventajas: Sólo consta de dos capas de elementos de proceso
Entrenamiento muy rápido

Desventajas: Es difícil identificar las funciones de las ligaduras.

4.4.5 Perceptrón Multicapa recurrente

Creadores: Almeida
Pineda
Fecha: 1987
Referencias: *[Almeida, 1987][Almeida, 1988]*
[Pineda, 1987][Pineda, 1988]

Estructura:

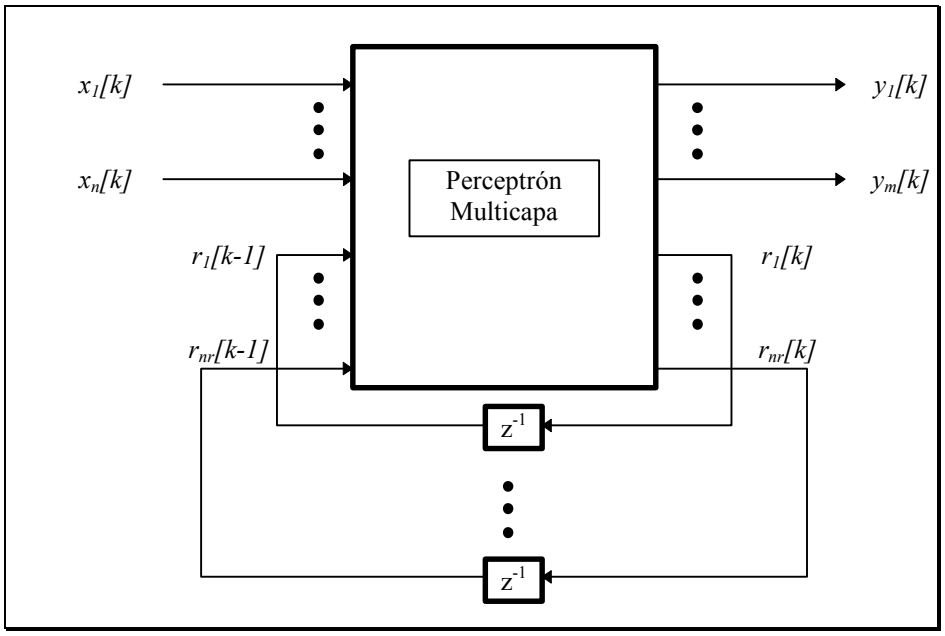

Figura 4.9: Conexión recurrente de un Perceptrón Multicapa

Esta estructura recurrente permite estimar los valores de las salidas ***y*** en el instante *k*, a partir de los valores de las entradas externas ***x*** en el mismo instante *k*, y de los valores de lo que podríamos llamar variables de estado internas ***r*** en el instante *(k-1)*, entre las que podría haberse incluído algunas de las variables de salida.

Aplicaciones: - Control
- Reconocimiento del habla
- Predicción de secuencias

Ventajas: Capaz de tratar información temporal

Desventajas: Estructuras muy complicadas. El aprendizaje puede resultar muy difícil.

4.4.6 Red de Retardos Temporales: TDNN (*"Time Delay Neural Network"*)

Creadores: D.W. Tank y J.J. Hopfield
K.J. Lang y G.E. Hinton
Fecha: 1987
Referencias: *[Tank & Hopfield, 1987]*
[Lang & Hinton, 1988]

Estructura:

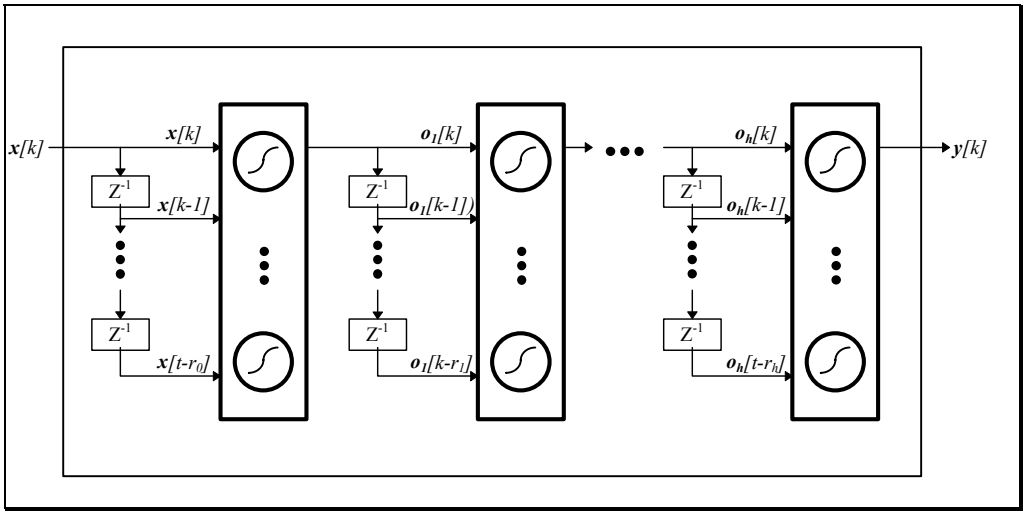

Figura 4.10: Estructura TDNN

Esta estructura está formada por capas de perceptrones totalmente interconectados (todos los perceptrones de una capa admiten el mismo conjunto de entradas), a los que se ha dotado de copias retrasadas de sus entradas.

Aplicaciones: Reconocimiento del habla

Ventajas: Obtiene tasas de reconocimiento similares a las obtenidas con los métodos tradicionales, pero con una rapidez de proceso mucho mayor.

Desventajas: La ventana temporal es de tamaño fijo.

4.4.7 Teoría de Resonancia Adaptativa: ART (*"Adaptive Resonance Theory"*)

Creadores: G. Carpenter & S. Grossberg
Fecha: 1983
Referencias: *[Carpenter & Grossberg, 1987]*

Estructura:

El objetivo de esta red es ser capaz de aprender nuevos patrones de entrada (plasticidad), reteniendo los patrones ya aprendidos (estabilidad). Para llevar a cabo esta tarea, Carpenter y Grossberg se vieron obligados a crear una estructura muy compleja compuesta por tres subsistemas: el subsistema atencional, el subsistema orientativo y el control de ganancia atencional.

El subsistema atencional lleva a cabo el reconocimiento de patrones, y está estructurado como una memoria asociativa bidireccional (ver Figura 4.11). El subsistema orientativo es invocado cuando se descubre un nuevo patrón. Bajo estas circunstancias, este subsistema inhibe el proceso de comparación con categorías de patrones ya aprendidos y crea una nueva categoría para dar respuesta al nuevo tipo de patrón de entrada. El control de ganancia atencional es el encargado de estabilizar todo el sistema.

Aplicaciones: Reconocimiento de patrones

Ventajas: Esta estructura es capaz de aprender nuevos patrones, creando nuevas categorías.

Desventajas: Las categorías obtenidas son muy sensibles a los parámetros del aprendizaje.

4.4.8 Memorias Asociativas Bidireccionales

Creadores: B. Kosko
Fecha: 1987
Referencias: *[Kosko, 1988]*

Estructura:

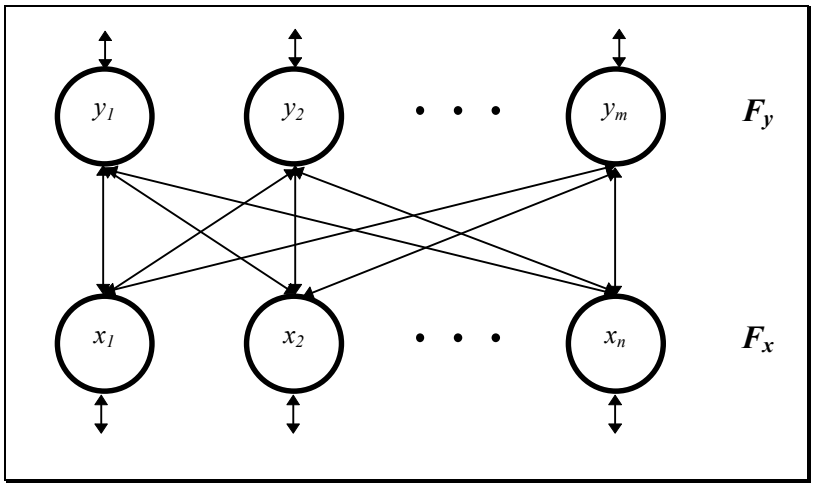

Figura 4.11: Estructura de una Memoria Asociativa Bidireccional

Esta estructura está formada por dos capas de elementos de proceso, conectadas de forma bilateral. Los elementos de proceso son perceptrones con función de activación en escalón: toman su vector de entradas, lo multiplican escalarmente por su vector de pesos, y dan como salida el valor "+1" si el resultado anterior es positivo, y "-1" en caso contrario. La red consta pues de dos matrices de conexiones, una asociada a la capa de entrada F_x, y otra a la capa de salida F_y siendo una la traspuesta de la otra.

La evaluación de la red consiste en inicializar la capa de entrada con un vector $\boldsymbol{x}$, e ir evaluando consecutivamente ambas capas hasta alcanzar un estado estable. En ese momento quedan almacenados en la capa F_y los valores de salida.

Aplicaciones: Memorias heteroasociativas

Ventajas: Arquitectura, dinámica y ley de aprendizaje sencillas. Su estabilidad está probada.

Desventajas: Baja capacidad de almacenamiento
La recuperación de datos no tiene mucha precisión.

4.4.9 Red de Hopfield

Creadores: J. Hopfield
Fecha: 1982
Referencias: *[Hopfield, 1982]*

Estructura:

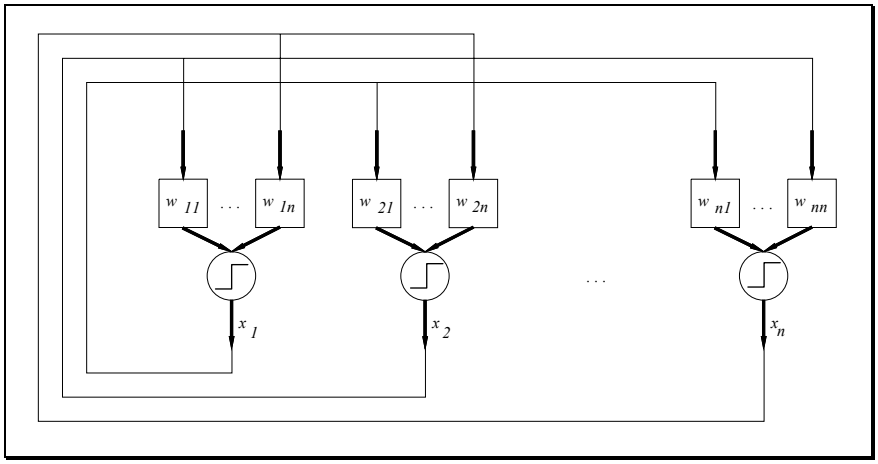

Figura 4.12: Estructura de la red de Hopfield

El principio de funcionamiento de esta red es inicializar el vector de estados $\boldsymbol{x}$ con el vector de entradas suministrado, y dejar converger la red hasta que se alcance un estado estable en el que la evaluación de la red no altere el vector de estados $\boldsymbol{x}$. El aprendizaje consiste en dar forma a la función de energía de la red de tal manera que admita como mínimos locales los estados estables que se requieran.

Aplicaciones: - Autoasociación
- Optimización

Ventajas: Arquitectura muy sencilla, fácilmente realizable en tecnología VLSI.
La estabilidad de su dinámica ha sido probada.

Desventajas: Baja capacidad de almacenamiento.
Aparecen estados estables espúreos

4.4.10 Máquina de Boltzmann

Creadores: D. Ackley, G. Hinton, T. Sejnowski
Fecha: 1984
Referencias: *[Hinton et al., 1984][Ackley et al., 1985]*

Estructura:

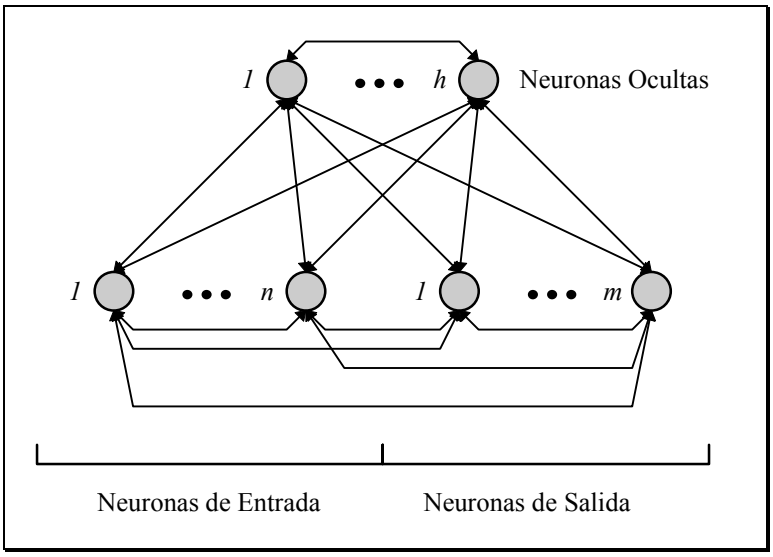

Figura 4.13: Estructura de la máquina de Boltzmann

La máquina de Boltzmann está formada por un conjunto de unidades completamente interconectadas de forma bidireccional. Cada unidad está siempre en uno de los dos estados posibles ("ON/OFF"), y adopta uno de estos dos estados según una función probabilista de los estados de sus unidades vecinas y de los pesos que rigen las conexiones que les unen. Esta matriz de conexiones es una matriz simétrica. El ajuste de esta matriz de conexiones puede realizarse de forma supervisada minimizando una función de energía (método directamente relacionado con el algoritmo de recocido simulado).

Aplicaciones: - Reconocimiento de patrones
- Optimización

Ventajas: Son capaces de formar representaciones óptimas de las características de los patrones de entrada.
La optimización está basada en el descenso por una superficie de energía.

Desventajas: Tiempos de aprendizaje muy elevados.

4.4.11 Red de cuantización vectorial: LVQ (*"Learning Vector Quantization"*)

Creadores: T. Kohonen
Fecha: 1981
Referencias: *[Kohonen, 1988]*

Estructura:

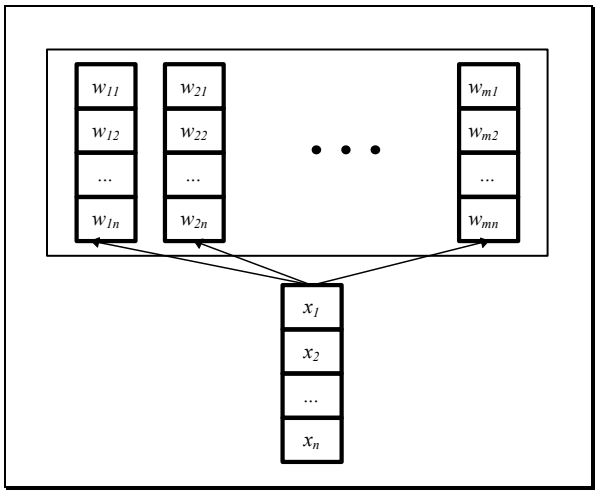

Figura 4.14: Estructura de la red de cuantización vectorial

Esta estructura está formada por un conjunto de *m* elementos de proceso que se limitan a almacenar un vector $\boldsymbol{w_i}$ de igual dimensión que el vector de entrada $\boldsymbol{x}$. Cuando se presenta un vector de entrada, la red ofrece como salida el vector $\boldsymbol{w}$ más cercano al vector $\boldsymbol{x}$. El objetivo del aprendizaje es distribuir los vectores $\boldsymbol{w_i}$ en el espacio de entrada, de forma tal que se minimice el error de reconstrucción de los vectores $\boldsymbol{x}$ a partir de los $\boldsymbol{w_i}$. Esto se consigue cuando los vectores $\boldsymbol{w_i}$ están distribuídos en el espacio de entrada según la misma función de densidad probabilista que los vectores de entrada.

Aplicaciones: - Autoasociación (recuperación de datos incompletos o ruidosos)
- Compresión de datos

Ventajas: Crea una representación vectorial distribuída según la misma función de densidad probabilística que los datos originales.

Desventajas: No hay una metodología clara para la selección del número de representantes y de los parámetros de aprendizaje.

4.4.12 Mapas auto-organizativos de Kohonen

Creadores: T. Kohonen
Fecha: 1981
Referencias: *[Kohonen, 1990]*

Estructura:

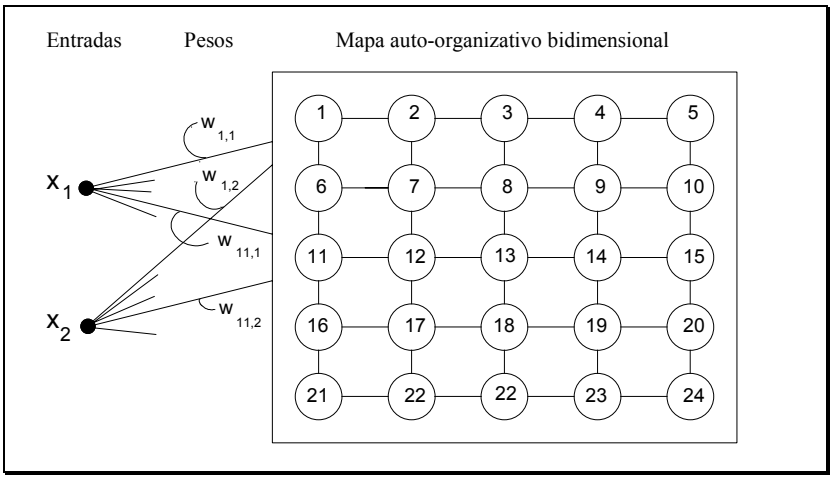

Figura 4.15: Estructura de un Mapa Auto-organizativo de Kohonen

Los mapas auto-organizativos de Kohonen son similares a las redes de cuantización vectorial, pero en este caso las neuronas están ordenadas topológicamente. Frente a la presentación de un patrón n-dimensional de entrada, compiten lateralmente hasta que sólo una de ellas queda activada. Esta relación no lineal (vector de entrada/unidad vencedora) se fragua mediante aprendizaje no supervisado en la etapa de entrenamiento. El objetivo de este aprendizaje es que patrones de entrada con características parecidas queden asociados a neuronas topológicamente cercanas, o dicho de otra forma, se trata de sintonizar de una forma topológicamente ordenada, las neuronas de la red con las características de los patrones de entrada.

Aplicaciones: - Agrupación y representación de datos
- Compresión de datos
- Optimización

Ventajas: Crea una representación vectorial, ordenada topológicamente, de los datos de entrada.

Desventajas: No hay una metodología clara para la selección del número de representantes y de los parámetros de aprendizaje.

4.5 RNA Supervisadas de Aproximación Funcional.

En el apartado anterior hemos dado un rápido repaso a las estructuras conexionistas más importantes, señalando sus principales campos de aplicación. En este apartado nos vamos a centrar en un subconjunto de ellas, que son las RNA supervisadas no recurrentes, diseñadas para resolver problemas de aproximación funcional.

En el Capítulo 2 establecimos un modelo de aprendizaje supervisado, y las estrategias de aprendizaje necesarias para el ajuste de aproximadores funcionales, capaces de llevar cabo un conjunto de transformaciones de entrada/salida de la forma $\boldsymbol{y}=f(\boldsymbol{x},\boldsymbol{w})$.

Llegados a este punto vamos a olvidarnos del símil biológico y a dar un giro al concepto hasta aquí introducido de RNA, centrándonos en los modelos conexionistas supervisados de aproximación funcional no recurrentes. Vamos a considerar a estos modelos como aproximadores funcionales (o herramientas de interpolación multidimensional no lineal), que llevan a cabo una transformación de un vector de entradas externas $\boldsymbol{x}\in\Re^n$ en un vector de salidas externas $\boldsymbol{y}\in\Re^m$:

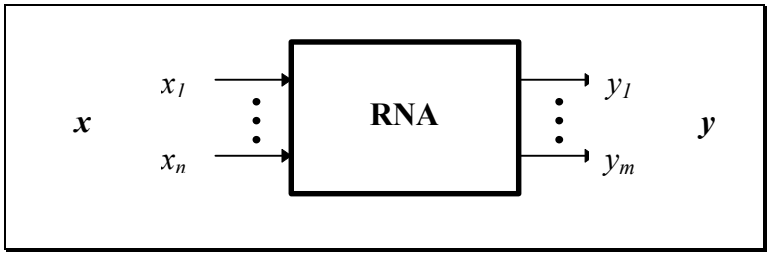

Figura 4.16: RNA como aproximador funcional

Una vez que se ha fijado la estructura interna de la RNA, la relación funcional llevada a cabo por el aproximador resultante dependerá en general de su vector de parámetros $\boldsymbol{w}$ (llamado vector de *pesos* en el ámbito conexionista), de tal forma que desde un punto de vista externo la relación entrada/salida queda definida por:

$$\boldsymbol{y} = f(\boldsymbol{x},\boldsymbol{w}).$$

Ecuación 4.1

Los algoritmos de aprendizaje supervisado vistos en el Capítulo 2 establecen estrategias de búsqueda del óptimo vector de pesos $\boldsymbol{w}$, partiendo de un conjunto de muestras de la relación de entrada/salida deseada. El cálculo de las derivadas de las

salidas respecto a cada uno de los parámetros ($\partial y_i/\partial x_j$) permitirá utilizar algoritmos de optimización basados en el gradiente que agilicen este proceso de búsqueda.

El carácter no recurrente de estos modelos obliga a que la salida $\boldsymbol{y}$ en el instante k no dependa más que de las entradas $\boldsymbol{x}$ y de los parámetros $\boldsymbol{w}$ del aproximador en el mismo instante k . Esto impide la existencia de conexiones internas recurrentes que inducirían una dependencia de la salida en valores pasados de las entradas y de los parámetros. En el Capítulo 3 fueron introducidas arquitecturas recurrentes especialmente recomendables para el modelado de procesos dinámicos. Estas arquitecturas pueden utilizar como aproximador funcional los modelos introducidos en este capítulo.

La estructura interna de estos aproximadores funcionales determina la expresión de la función $f(\boldsymbol{x},\boldsymbol{w})$ y por tanto de las derivadas $\partial y_i/\partial x_j$. La utilización de una RNA como aproximador funcional obliga a que esta transformación se realice mediante la interconexión de elementos de proceso muy sencillos, de tal forma que el procesamiento resultante queda descompuesto en subtareas altamente paralelizables. La información funcional queda almacenada de forma distribuida en los pesos de los elementos de proceso. El tratamiento de la información de entrada se realiza también de forma distribuida, en contraposición con los algoritmos secuenciales a los que estamos acostumbrados.

Las dos familias de RNA supervisadas que mayor aceptación y desarrollo han tenido en los últimos diez años han sido el Perceptrón Multicapa (PM: "Multilayer Perceptron") y las redes de funciones base radiales (RBFN: "Radial Basis Function Network").

En este capítulo nos vamos a centrar en el Perceptrón Multicapa, dejando las redes RBFN para el Capítulo 5. El origen del PM se remonta a los trabajos de Rosenblatt sobre el "Perceptrón" (*[Rosenblatt, 1962]*) y a los trabajos de Widrow sobre la estructura "MADALINE" (*[Widrow, 1962]*).

4.5.1 El Perceptrón

El Perceptrón es un elemento de proceso no lineal cuya estructura se muestra en la Figura 4.17:

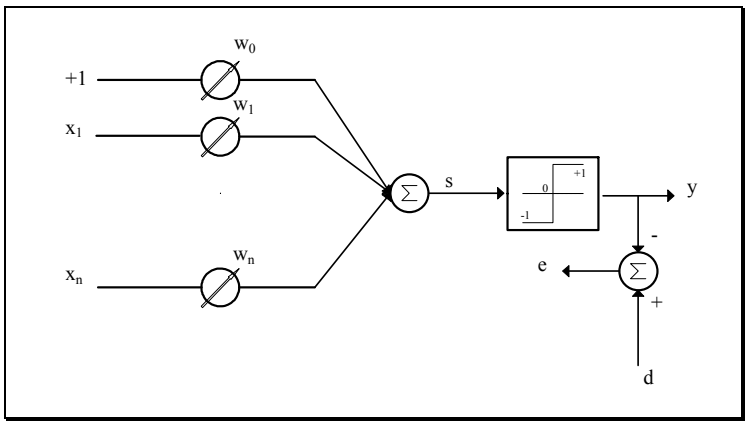

Figura 4.17: Estructura del Perceptrón

El vector de entradas externas $\boldsymbol{x} \in \Re^n$ se amplia con una entrada interna constantemente conectada al valor +1 y se multiplica escalarmente por el vector de pesos $\boldsymbol{w} \in \Re^{n+1}$ para obtener la activación *s* del perceptrón dada por:

$$s = w_0 + \sum_{i=1}^{n} w_i x_i$$

Ecuación 4.2

Este nivel de activación determina el valor binario de la salida *y*, que será 1 si la activación es positiva y -1 en otro caso. El resultado de esta transformación es una dicotomía del espacio de entrada $\boldsymbol{X}$, que ha quedado dividido en dos regiones por el hiperplano $s=0$. Si por ejemplo consideramos el espacio de entrada $\boldsymbol{X} \equiv \Re^2$, la frontera entre las dos regiones del espacio de entrada que darán salida distinta vendrá dada por:

$$s = 0 \Leftrightarrow x_2 = -\frac{w_1}{w_2} x_1 - \frac{w_0}{w_2}$$

Ecuación 4.3

que es la ecuación de una recta de pendiente ($-w_1/w_2$) y de ordenada en el origen ($-w_0/w_2$). La inclusión de la entrada umbral conectada a +1 permite obtener fronteras de decisión que no pasen por el origen.

Este elemento de proceso podrá pues resolver problemas de clasificación linealmente separables, como las funciones lógicas "AND" y "OR" (ver Figura 4.18), en los que se puede encontrar un hiperplano (recta en el caso bidimensional) que separe el espacio de entrada en dos regiones, una asociada con cada clase. Por el contrario, si el problema de partida no es separable linealmente, como es el caso del "XOR", el perceptrón no podrá resolverlo.

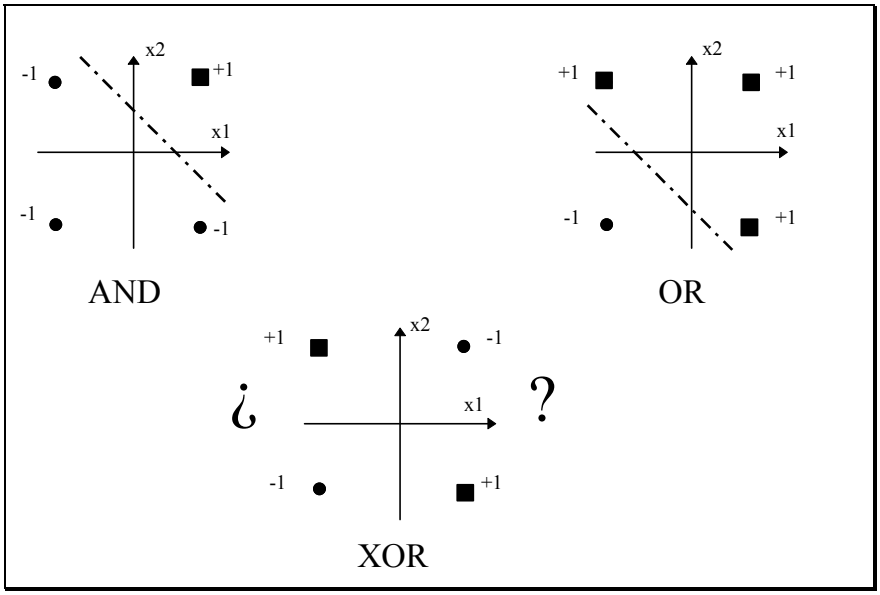

Figura 4.18: Separabilidad lineal

La ley de aprendizaje de Rosenblatt (*[Rosenblatt, 1962]*) establece un algoritmo iterativo para la determinación del vector de pesos $\boldsymbol{w}$. Este algoritmo es un algoritmo de aprendizaje supervisado basado en la corrección del error de clasificación sobre un conjunto de entrenamiento de tamaño N, formado por pares de entradas y salida deseada $\{(\boldsymbol{x}(k),d(k)), k=1,...,N\}$. Esta ley viene dada por:

$$\boldsymbol{w}[k+1] = \boldsymbol{w}[k] + \frac{1}{2}\alpha\, e[k]\,\boldsymbol{x}[k]$$

Ecuación 4.4

donde $e[k]=d[k]-y[k]$ es el error cometido con el *k-imo* ejemplo presentado al perceptrón durante el entrenamiento.

Este algoritmo es capaz de separar cualquier conjunto de entrenamiento linealmente separable. Sin embargo, si no se da la condición de separabilidad lineal, el algoritmo puede entrar en una dinámica oscilante en torno a errores no pequeños. El valor de α no afecta a la estabilidad del mismo, pero sí a la rapidez de convergencia. Rosenblatt solía tomar $\alpha=1$.

4.5.2 La estructura MADALINE

La estructura MADALINE fue en su origen una combinación no lineal y estática de combinadores lineales adaptativos, a los que se les añadió una función escalón a su salida. Los elementos de proceso resultantes fueron bautizados por Widrow con el nombre de ADALINE (*"ADAptive LINear Element"*) y toman la forma mostrada en la Figura 4.19:

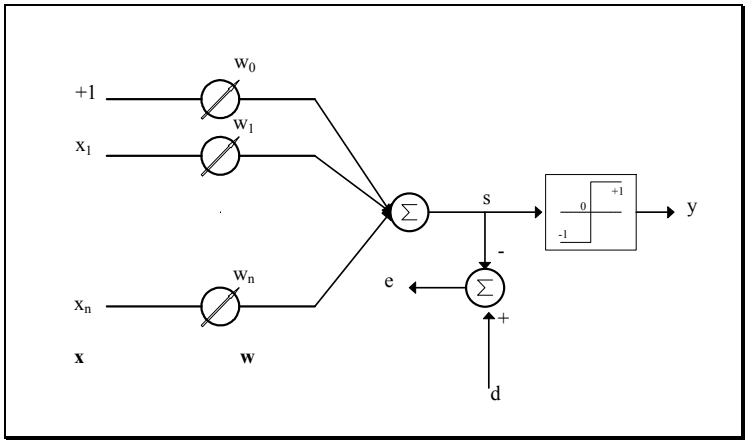

Figura 4.19: Estructura del ADALINE

Su estructura es idéntica a la del perceptrón, salvo que la señal de error que se considera para el aprendizaje se calcula a partir de la salida del ADALINE antes de pasar por la función escalón (*e[k]=d[k]-s[k]*). Esta consideración hace que el error *e[k]* sea una función continua y derivable del vector de pesos ***w***, pudiendo aplicar estrategias de aprendizaje basadas en el descenso del gradiente. Además, si se considera el error cuadrático medio como medida del error de clasificación, la hipersuperficie de error resultante es un paraboloide que, salvo degeneraciones, tiene un único punto mínimo, como muestra la Figura 4.20.

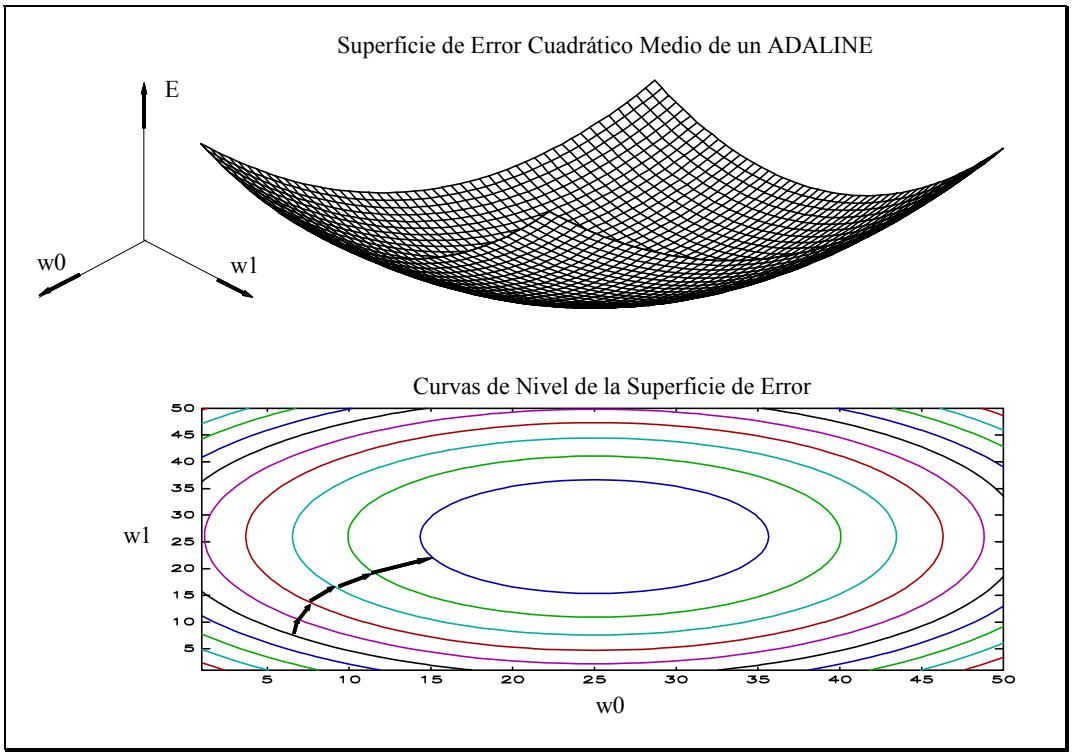

Figura 4.20: Superficie y curvas de nivel del error cuadrático medio de un ADALINE

Las leyes de aprendizaje utilizadas en este caso (*[Widrow and Lehr, 1990]*) incluyen el algoritmo α-*LMS* o "Regla Delta de Widrow-Hoff":

$$\boldsymbol{w}[k+1] = \boldsymbol{w}[k] + \alpha \frac{e[k]\,\boldsymbol{x}[k]}{\|\boldsymbol{x}[k]\|^2}$$

Ecuación 4.5

y reglas basadas en el descenso del gradiente, como el algoritmo μ-*LMS*:

$$\boldsymbol{w}[k+1] = \boldsymbol{w}[k] + 2\,\mu\, e[k]\,\boldsymbol{x}[k]$$

Ecuación 4.6

Estas estrategias de aprendizaje son más robustas que las del perceptrón en el caso de problemas no separables linealmente, llegando a alcanzar subóptimos.

La combinación de estos elementos con lógicas no adaptativas sencillas, como la mostrada en la Figura 4.21, dieron origen a la primera estructura MADALINE (*"Many ADALINE's"*), que ya era capaz de resolver problemas linealmente no separables, si los ADALINE se combinaban adecuadamente.

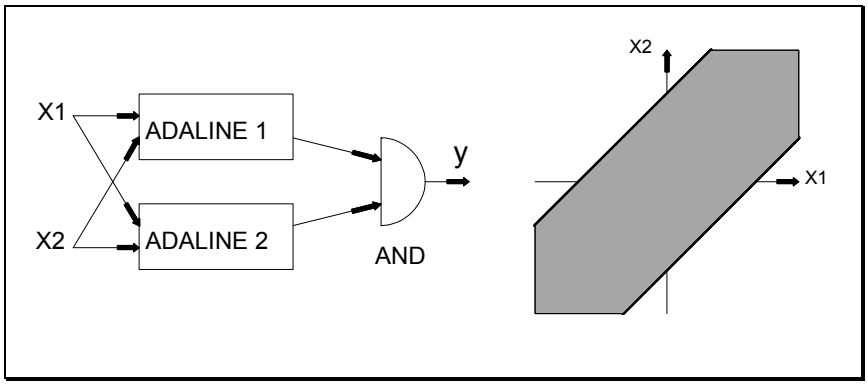

Figura 4.21: Ejemplo de MADALINE

Hoff desarrolló en los años 60 toda una serie de leyes de aprendizaje para esta estructura (*[Hoff, 1962]*), pero hubo que esperar hasta finales de los años ochenta para disponer de algoritmos de aprendizaje que permitiesen adaptar redes multicapa de ADALINE, donde fueron sustituidos los elementos fijos de la MADALINE original por elementos adaptativos (*[Widrow et al., 1987],[Widrow and Winter, 1988], [Winter, 1989]*).

Tanto el perceptrón como la MADALINE fueron inicialmente diseñados para resolver problemas de clasificación. Las conclusiones de estas primeras experiencias fueron que para ser capaces de resolver problemas linealmente no separables, era necesario conectar en cascada varias capas de estos elementos de proceso no lineales. Surgió entonces el problema de cómo entrenar estas estructuras multicapa, donde no se disponía de salidas deseadas para las neuronas de las capas ocultas. El primer paso hacia la solución consistió en sustituir la función de activación en escalón por una función sigmoidal ($\varphi(s)=1/(1+exp(-s))$), o hiperbólica ($\varphi(s)=tanh(s)$), que aproximase a la función escalón, pero que fuese derivable. Esta derivabilidad permitió aplicar algoritmos de aprendizaje basados en el descenso del gradiente, dando lugar al éxito del Perceptrón Multicapa y del algoritmo de retropropagación del error.

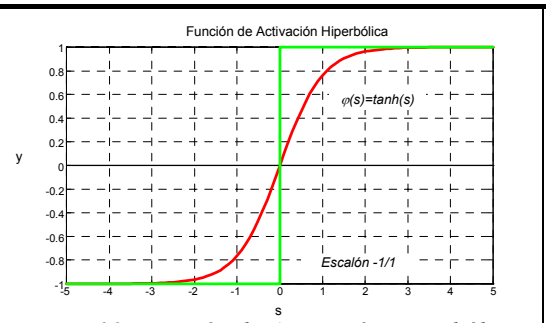

Figura 4.22: Función de Activación Hiperbólica

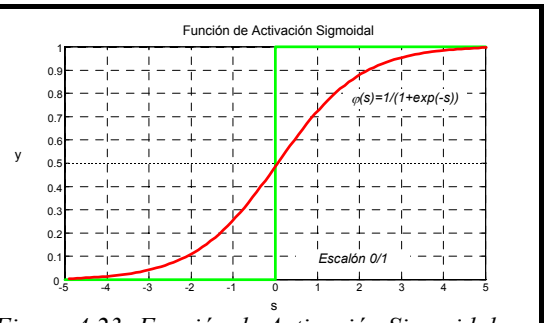

Figura 4.23: Función de Activación Sigmoidal

4.5.3 El Perceptrón Multicapa (PM)

El Perceptrón Multicapa (PM: *"Multilayer Perceptron"*) es la estructura conexionista que impulsó la investigación en el campo de las RNA allá por los años 80, y que por lo tanto, mayor número de aplicaciones ha tenido en la práctica.

Su historia ha sido muy controvertida, ya que su origen ha sido adjudicado a diversos investigadores: Paul Werbos en 1974 (*[Werbos, 1974]*), David Parker en 1984/85 (*[Parker, 1985]*) y David Rumelhart, Ronald Williams y otros miembros del grupo "PDP" en 1985 (*[Rumelhart et al., 1986 a y b]*).

A pesar de estas controversias, nadie duda de quién hizo de esta estructura conexionista una herramienta de aplicación práctica ni de quién la divulgó: antes de los trabajos de D. Rumelhart y del resto de los miembros del *"PDP Group"*, esta estructura era materia obscura que no despertaba el más mínimo interés. Hoy en día se ha convertido en la estrella por excelencia de la computación neuronal.

a) Estructura del PM

El PM es una estructura jerárquica de capas de elementos de proceso (neuronas) totalmente interconectadas: los elementos de proceso de una capa admiten como entradas las salidas de los elementos de proceso de la capa anterior, pero no se admiten interconexiones dentro de una misma capa.

La estructura del PM es "alimentada hacia delante" (*"feedforward"*), de tal modo que en fase de operación (no de aprendizaje), la salida de una neurona no puede ser realimentada sobre sí misma, directa o indirectamente. Esto implica que la salida actual de la red no puede influir en salidas futuras.

Se distinguen tres tipos de capas (ver Figura 4.24):

- Capa de entrada: está formada por *n* unidades (siendo *n* el número de entradas externas) que se limitan a distribuir las señales externas de entrada a la capa siguiente. No tiene ninguna funcionalidad matemática, pero sí está físicamente presente en las realizaciones *"hardware"*.

- Capas ocultas: están formadas por elementos de proceso que no tienen contacto físico con el exterior. El número de capas ocultas de un PM es variable, pudiendo ser incluso nulo.

- Capa de salida: está formada por *m* elementos de proceso (siendo *m* el número de salidas externas) cuyas salidas constituyen el vector de salidas externas del PM. Sólo existe una capa de salida en cada PM.

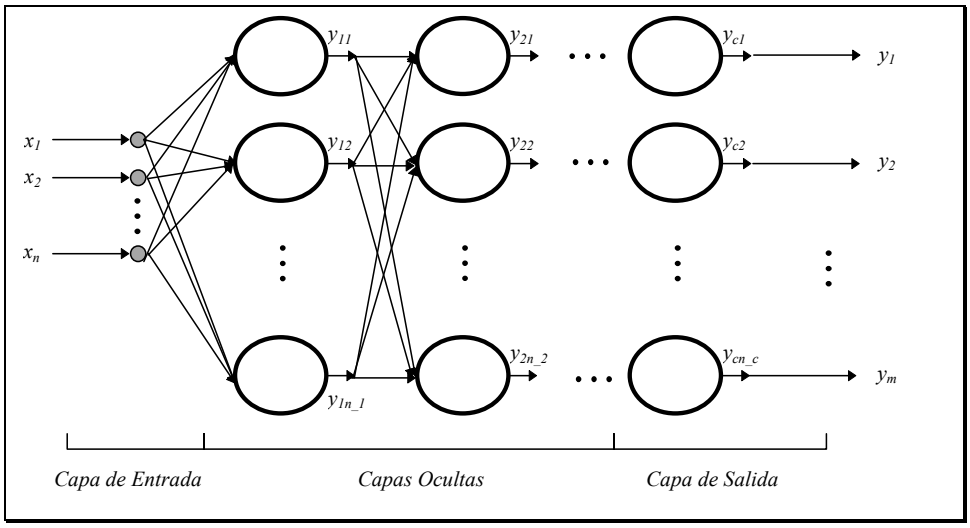

Figura 4.24: Estructura del Perceptrón Multicapa

Todos los elementos de proceso de las capas ocultas y la capa de salida son del tipo Perceptrón, aunque el número de entradas y la función de activación puede variar de uno a otro. En la Figura 4.25 se ha reproducido la estructura interna de estos elementos de proceso, con la notación que vamos a utilizar:

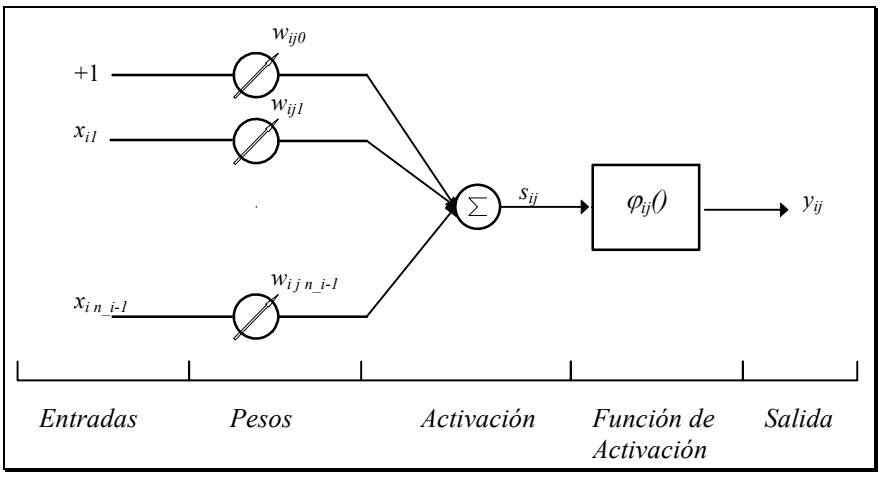

Figura 4.25: Estructura de un Perceptrón

- Denotaremos *c* al número de capas de elementos de proceso del PM, de las cuales las (*c-1*) primeras serán ocultas y la número *c* será la de salida. Utilizaremos el subíndice *i* (*i=1,...,c*) para referirnos al número de capa.

- La capa número *i* estará formada por *n_i* perceptrones, que serán identificados mediante dos subíndices: el perceptrón (*ij*) estará localizado en la capa *nºi*, en la posición *nºj* dentro de esa capa (*j=1,...,n_i*).

- La activación del elemento de proceso (*ij*) será notada s_{ij}.

- La función de activación del elemento de proceso (*ij*) será notada $\varphi_{ij}()$.

- La salida del elemento de proceso (*ij*) será notada y_{ij}.

- El elemento de proceso (*ij*) estará conectado a todos los elementos de proceso de la capa anterior si *i>1* o a todas las señales de entrada si *i=1*, además de a la señal de polarización de valor +1. De esta forma x_{ijk} (*k=0,...,n_i-1*), con *k>0,* será la entrada del elemento de proceso (*ij*), conectada a la salida del elemento de proceso (*i-1 k*) si *i>1* ó a la entrada externa *k* si *i=1*. En cualquier caso, x_{ij0} estará conectado a la señal de polarización de valor +1. Se desprende que *n_0=n*, el número de entradas externas, y *n_c=m*, el número de salidas externas.

- Los pesos del elemento de proceso (*ij*) serán notados w_{ijk}. Su numeración es análoga a la de las entradas x_{ijk}.

De esta forma, la función de transferencia del elemento de proceso (*ij*) queda definida de una forma homogénea por:

$$y_{ij} = \varphi_{ij}\left(\sum_{k=0}^{n\_i-1} w_{ijk}\, x_{ijk} \right)$$

Ecuación 4.7

Una vez definida la notación utilizada, la estructura del PM quedará unívocamente definida al precisar el vector $(n\_0,\ n\_1,\ ...,\ n\_c)^T$, que contiene el número de elementos de proceso de cada capa, incluida la de entrada, y las funciones de activación de cada elemento de proceso.

Entre las funciones de activación derivables más comunes podemos citar:

- Función Hiperbólica: $\varphi(s)=tanh(s)$ (ver Figura 4.22)

- Función Sigmoidal: $\varphi(s)=1/(1+exp(-s))$ (ver Figura 4.23)

- Función Identidad: $\varphi(s)=s$

Una elección generalmente apropiada de funciones de activación para aproximación funcional, es dotar a los elementos de proceso de las posibles capas ocultas de funciones de activación hiperbólicas, y a los de la capa de salida de funciones identidad. De esta forma se dispone de la capacidad de generar salidas no limitadas a un rango concreto, como el [0,1] en el caso de la sigmoidal, no siendo pues necesario el escalar previamente los valores de salida para este fin.

Incluso con salidas escaladas, la utilización de funciones de activación acotadas en la capa de salida obliga a incluir en el conjunto de entrenamiento los máximos valores de salida esperados.

Por ejemplo, si el rango de la salida en el conjunto de entrenamiento es [0,100] y todas nuestras funciones de activación son sigmoidales, el paso previo al aprendizaje sería la normalización de las salidas al intervalo [0,1]. Una vez escalado el conjunto de entrenamiento, pasaríamos a la fase de aprendizaje, donde la actualización de los pesos de la capa de salida obligaría a que las salidas extremas de nuestro rango se obtuviesen en las zonas saturadas de la sigmoidal. En fase de operación, nuestra red sería incapaz de generar salidas equivalentes mayores de 100 o menores de 0, incluso frente a entradas alarmantes.

Esta elección no es apropiada sin embargo en el caso de salidas binarias, donde la elección más oportuna de funciones de activación para los elementos de proceso de la capa de salida parece ser la del tipo sigmoidal (ajustando sus límites a los de los valores binarios) durante el aprendizaje, pudiendo luego ser sustituídas por funciones escalón para la fase de evaluación.

b) Capacidad de Representación del PM

Las primeras investigaciones que trataron de probar la capacidad de representación funcional universal del PM fueron debidas a Robert Hecht-Nielsen, quien aplicó una versión ampliada del teorema de superposición de Kolmogorov a esta estructura conexionista (*[Hecht-Nielsen, 1987]*). En 1989 se publicó por vez primera una demostración rigurosa de la capacidad de representación funcional del PM (*[Cybenco,1989]*), en la que se probaba que un PM de una única capa oculta era capaz de aproximar uniformemente cualquier función continua de soporte el hipercubo unitario. Estos trabajos fueron realizados en el año 1988 (y publicados un año después). En 1989 fueron publicados de forma independiente otros dos artículos donde se demostraba la capacidad de aproximación funcional del PM: *[Funahashi, 1989]* y *[Hornik et al., 1989]*. De cualquier forma el teorema de aproximación universal dice así:

> "Sea $\varphi(.)$ una función continua no constante, acotada y monótonamente creciente. Sea I_n el hipercubo unitario n-dimensional $[0,1]^n$. Sea $C(I_n)$ el espacio de funciones continuas definidas en I_n .
>
> Entonces, dada cualquier función $g \in C(I_n)$ y cualquier $\varepsilon>0$, existe un entero h y unos conjuntos de constantes reales α_i, θ_i y w_i donde $i=1,...,h$ y $j=1,...,n$ tales que la función f definida por:
>
> $$f(x_1,\ldots,x_n) = \sum_{i=1}^{h} \alpha_i \ \varphi\left(\sum_{j=1}^{n} w_{ij} x_j - \theta_i\right)$$
>
> *Ecuación 4.8*
>
> es una realización aproximada de la función $g(.)$, de tal modo que:
>
> $$\left|f(x_1,\ldots,x_n) - g(x_1,\ldots,x_n)\right| < \varepsilon$$
>
> *Ecuación 4.9*
>
> para todo $(x_1,...,x_n) \in I_n$."

Este teorema puede aplicarse directamente a PM de n entradas y una salida, con una única capa oculta compuesta por h neuronas de función de activación sigmoidal o hiperbólica (ambas cumplen las condiciones impuestas a $\varphi()$), y función de activación identidad en la capa de salida. La Ecuación 4.8 da la función de

transferencia del correspondiente PM, donde las constantes w_{ij} y θ_i se refieren a los pesos de la capa oculta y las α_i a los de la capa de salida.

Aunque el teorema haga mención de una única capa oculta, puede ser conveniente en algunos casos el aumentar el número de capas, ya que de otra forma el número de elementos de proceso ocultos (h) sería impracticable.

Cotas del error de aproximación

El teorema de aproximación universal impulsó a varios investigadores a profundizar en el estudio de PM de una única capa oculta. Este fue el caso de Barron, quien se interesó por las propiedades de aproximación del PM de una única capa oculta con funciones de activación sigmoidales, y neurona lineal de salida.

En sus trabajos (*[Barron, 1991], [Barron, 1992]*), Barron estableció la siguiente cota para la función de riesgo total R, definida como la esperanza matemática del error cuadrático de aproximación de la función *g()* por el aproximador *f()*:

$$R < O\left(\frac{C_f^2}{h}\right) + O\left(\frac{h\,n}{N} log(N)\right)$$

Ecuación 4.10

donde *O()* significa "del orden de", y C_f es el momento absoluto de primer orden de la distribución de la magnitud de la transformada de Fourier de la función f, que es una medida de su "regularidad".

El resto de parámetros que intervienen en la Ecuación 4.10 dependen de la estructura del PM y de su entrenamiento:

- n: número de unidades de entrada del PM.

- h: número de unidades ocultas del PM.

- N: número de ejemplos del conjunto de entrenamiento.

Como vimos en el Capítulo 2, la función de riesgo R es una medida de la capacidad de generalización del aproximador funcional. La Ecuación 4.10 muestra cómo R es un compromiso entre la complejidad del PM, determinada por el número de unidades ocultas h, y cantidad de información utilizada para ajustar cada peso, medida por el ratio *(hn)/N*. La conclusión más importante de este resultado es que no se necesitan

conjuntos de entrenamiento de tamaño exponencial para conseguir buenas estimaciones con PM. Como regla general aproximada podemos establecer que el tamaño del conjunto de entrenamiento ha de ser mayor que el ratio entre el número de parámetros libres (pesos) del PM y el error de estimación.

Existen también estudios que tratan de acotar el número de unidades ocultas de PM, entre los que cabe destacar los trabajos de Shih-Chi y Yih-Fang Huang (*[Huang, 1991]*). Estos trabajos no establecen una estrategia clara para la selección práctica del número de unidades ocultas de un PM, debiendo seguir en la mayoría de los casos un procedimiento de prueba y error.

c) Algoritmo de Retropropagación y cálculo de derivadas

El algoritmo de retropropagación (*[Rumelhart et al., 1986b]*) es un algoritmo de cálculo del gradiente de la función de error cuadrático medio respecto de los pesos del PM. Vamos a introducir este algoritmo por la importancia histórica que tuvo en su momento, y por la eficacia que lo caracteriza.

Posteriormente serán calculadas las derivadas de las salidas externas del PM respecto de sus pesos y de sus entradas externas, para ajustarnos al modelo de aproximador funcional definido en el Capítulo 2. El cálculo del gradiente de la función de error respecto de los pesos, realizado a partir de las derivadas de las salidas, es un procedimiento más homogéneo y flexible que el algoritmo de retropropagación convencional, que supone ya una función de error determinada (el error cuadrático medio). Por otro lado, cuando el PM sea integrado dentro de una estructura recurrente, será necesario calcular las derivadas de sus salidas externas respecto de cada una de sus entradas. Este cálculo puede realizarse de forma simultánea al cálculo de las derivadas de las salidas respecto de los pesos, aprovechando resultados intermedios comunes.

Algoritmo de Retropropagación (*"Backpropagation"*)

Dado un conjunto de entrenamiento de la forma $\{(\mathbf{x}[1],\mathbf{d}[1]),...,(\mathbf{x}[N],\mathbf{d}[N])\}$, con $\mathbf{x}\in\Re^n$ y $\mathbf{d}\in\Re^m$, y un PM que lleva a cabo la transformación $\mathbf{y}=f(\mathbf{x},\mathbf{w})$, podemos definir el error cuadrático medio de aproximación del conjunto de entrenamiento como:

$$R(\mathbf{w}) = \frac{1}{N}\sum_{t=1}^{N}\sum_{a=1}^{m}(d_a[t]-y_a[t])^2 = \frac{1}{N}\sum_{t=1}^{N}e[t]$$

Ecuación 4.11

donde ha quedado definida la señal de error cuadrático correspondiente al ejemplo *t* como :

$$e[t] = \sum_{a=1}^{m} (d_a[t] - y_a[t])^2$$

Ecuación 4.12

Derivando la Ecuación 4.11 respecto del peso *k* de la neurona (*ij*) se obtiene:

$$\frac{\partial R(\mathbf{w})}{\partial w_{ijk}} = \frac{1}{N} \sum_{t=1}^{N} \frac{\partial e[t]}{\partial w_{ijk}}$$

Ecuación 4.13

Luego basta con calcular las derivadas de *e[t]* respecto de los pesos, para cada ejemplo de entrenamiento, y hacer la media de las derivadas así obtenidas. De ahora en adelante obviaremos el número de ejemplo *t*, entendiendo que el PM ha sido evaluado con el ejemplo *t*. Por la regla de la cadena obtenemos:

$$\frac{\partial e}{\partial w_{ijk}} = \frac{\partial e}{\partial s_{ij}} \frac{\partial s_{ij}}{\partial w_{ijk}}$$

Ecuación 4.14

El segundo factor de la ecuación anterior, que corresponde a la derivada de la activación de la neurona (*ij*) respecto a una de sus entradas, viene dado por:

$$\frac{\partial s_{ij}}{\partial w_{ijk}} = x_{ijk}$$

Ecuación 4.15

mientras que el primer factor es algo más complicado y depende de si la capa *i* es la capa de salida, o una capa oculta. A este factor ($\partial e/\partial s_{ij}$) le llamaremos señal retropropagada de error en la neurona (*ij*) y lo deniminaremos ξ_{ij}, de tal forma que resulta:

$$\frac{\partial e}{\partial w_{ijk}} = \xi_{ij}\, x_{ij}$$

Ecuación 4.16

Si la capa *i* es la capa de salida (*i*=*c*):

$$\xi_{ij} = \frac{\partial \sum_{a=1}^{m} (d_a - y_a)^2}{\partial s_{ij}} = -2\,(d_j - y_j)\;\varphi'_{ij}\,(s_{ij})$$

Ecuación 4.17

En otro caso podemos escribir:

$$\xi_{ij} = \sum_{h=1}^{n\_(i+1)} \frac{\partial e}{\partial s_{(i+1)h}} \frac{\partial s_{(i+1)h}}{\partial s_{ij}} = \sum_{h=1}^{n\_(i+1)} \xi_{(i+1)h}\;\varphi'_{ij}\,(s_{ij})\,w_{(i+1)hj}$$

Ecuación 4.18

Resumiendo queda pues:

Algoritmo de Retropropagación

- Función de error cuadrático medio:

$$R(\mathbf{w}) = \frac{1}{N}\sum_{t=1}^{N}\sum_{a=1}^{m}(d_a[t] - y_a[t])^2 = \frac{1}{N}\sum_{t=1}^{N} e[t]$$

Ecuación 4.19

- Para cada ejemplo:

$$\frac{\partial e}{\partial w_{ijk}} = \xi_{ij}\ x_{ij}$$

Ecuación 4.20

- Para la capa de salida ($i=c$):

$$\xi_{ij} = -2\ (d_j - y_j)\ \varphi'_{ij}\ (s_{ij})$$

Ecuación 4.21

- En las capas ocultas ($i=1,...,c-1$):

$$\xi_{ij} == \sum_{h=1}^{n\_(i+1)} \xi_{(i+1)h}\ \varphi'_{ij}\ (s_{ij})\ w_{(i+1)hj}$$

Ecuación 4.22

Este algoritmo permite calcular las derivadas de la función de error cuadrático medio respecto de cada peso del PM. El cálculo comienza evaluando la señal de error retropropagada (ξ_{ij}) y las correspondientes derivadas ($\partial e/\partial w_{ijk}$) en la capa de salida, y va retrocediendo hasta llegar a la primera capa oculta. Capa unidad ha de almacenar la señal de error retropropagada (un valor por unidad), para que unidades de capas inferiores puedan acceder a ellas.

Cálculo de las derivadas de las salidas respecto de los pesos:

La derivada de la salida externa *a* (*a=1,...,m*) del PM, respecto del peso w_{ijk}, del elemento de proceso (*ij*), viene dada por la regla de la cadena:

$$\frac{\partial y_a}{\partial w_{ijk}} = \frac{\partial y_a}{\partial y_{ij}} \frac{\partial y_{ij}}{\partial w_{ijk}}$$

Ecuación 4.23

El segundo factor de la ecuación anterior viene dado por:

$$\frac{\partial y_{ij}}{\partial w_{ijk}} = \varphi'_{ij}(s_{ij})\ x_{ijk}$$

Ecuación 4.24

mientras que el primero es algo más complicado y depende de si la capa *i* es la capa de salida o una capa oculta.

Si la capa *i* es la capa de salida (*i=c*):

$$\frac{\partial y_a}{\partial y_{ij}} = \delta_{aj} = \begin{cases} 1 & si\ a = j \\ 0 & si\ a \neq j \end{cases}$$

Ecuación 4.25

En otro caso podemos escribir:

$$\frac{\partial y_a}{\partial y_{ij}} = \sum_{h=1}^{n_{(i+1)}} \frac{\partial y_a}{\partial y_{(i+1)h}} \frac{\partial y_{(i+1)h}}{\partial y_{ij}}$$

Ecuación 4.26

donde además se cumple:

$$\frac{\partial y_{(i+1)h}}{\partial y_{ij}} = \frac{\partial y_{(i+1)h}}{\partial x_{(i+1)hj}} = \varphi'_{(i+1)h}(s_{(i+1)h})\ w_{(i+1)hj}$$

Ecuación 4.27

Resumiendo pues tenemos:

Derivadas de las salidas respecto de los pesos del PM

- Para la capa de salida ($i=c$):

$$\begin{cases} \dfrac{\partial y_a}{\partial w_{cjk}} = \delta_{aj}\ \varphi'_{cj}\ (s_{cj})\ x_{cjk} \\ \dfrac{\partial y_a}{\partial y_{cj}} = \delta_{aj} \quad siendo\ \delta_{aj} = \begin{cases} 1 & si\ a = j \\ 0 & si\ a \neq j \end{cases} \end{cases}$$

Ecuación 4.28

- Para las capas ocultas: ($i=1,...,c-1$)

$$\begin{cases} \dfrac{\partial y_a}{\partial w_{ijk}} = \dfrac{\partial y_a}{\partial y_{ij}}\ \varphi'_{ij}\ (s_{ij})\ x_{ijk} \\ \dfrac{\partial y_a}{\partial y_{ij}} = \displaystyle\sum_{h=1}^{n_{(i+1)}} \dfrac{\partial y_a}{\partial y_{(i+1)h}}\ \varphi'_{(i+1)h}\ (s_{(i+1)h})\ w_{(i+1)hj} \end{cases}$$

Ecuación 4.29

El algoritmo de cálculo de las derivadas es pues un algoritmo recursivo, en el que se comienza calculando las derivadas en la capa de salida, y se va retrocediendo hasta llegar a la primera capa oculta. Cada elemento de proceso ha de almacenar el vector de las derivadas de cada una de las salidas externas respecto de su salida, para poder retropropagárselo a la capa anterior.

Cálculo de las derivadas de las salidas respecto de las entradas:

De forma análoga podemos calcular las derivadas de las salidas externas respecto de las entradas de cada elemento de proceso: la derivada de la salida externa *a* (*a=1,...,m*) del PM, respecto de la entrada x_{ijk} del elemento de proceso (*ij*), viene dada por:

Derivadas de las salidas respecto de las entradas del PM

- Para la capa de salida (*i=c*):

$$\begin{cases} \dfrac{\partial y_a}{\partial x_{cjk}} = \delta_{aj}\ \varphi'_{cj}(s_{cj})\ w_{cjk} \\ \dfrac{\partial y_a}{\partial y_{cj}} = \delta_{aj} = \begin{cases} 1 & \text{si a = j} \\ 0 & \text{si a} \neq \text{j} \end{cases} \end{cases}$$

Ecuación 4.30

- Para las capas ocultas: (*i=1,...,c-1*)

$$\begin{cases} \dfrac{\partial y_a}{\partial x_{ijk}} = \dfrac{\partial y_a}{\partial y_{ij}}\ \varphi'_{ij}(s_{ij})\ w_{ijk} \\ \dfrac{\partial y_a}{\partial y_{ij}} = \displaystyle\sum_{h=1}^{n_{(i+1)}} \dfrac{\partial y_a}{\partial y_{(i+1)h}}\ \varphi'_{(i+1)h}(s_{(i+1)h})\ w_{(i+1)hj} \end{cases}$$

Ecuación 4.31

Estas derivadas pueden utilizarse para calcular las derivadas de las salidas externas del PM (y_a, *a=1,...,m*), respecto de sus entradas externas (x_b, *b=1,...,n*):

$$\frac{\partial y_a}{\partial x_b} = \sum_{j=1}^{n_1} \frac{\partial y_a}{\partial x_{1jb}}$$

Ecuación 4.32

d) Inicialización de los pesos

Importancia de una buena inicialización de pesos

Antes de aplicar un algoritmo de optimización para el ajuste de los pesos del PM, es necesario inicializar estos parámetros libres de tal forma que el proceso de búsqueda parta del mejor punto posible de la superficie de error. La práctica más habitual cuando no existe ninguna información previa, es inicializar los pesos del PM según una distribución uniforme en un pequeño margen de valores cercanos a cero.

Una mala inicialización de los pesos puede llevar, en el caso del PM, al fenómeno conocido como "saturación prematura" (*[Lee et al. 1991]*). Este fenómeno ocurre cuando el vector de pesos inicial queda localizado en una zona plana de la superficie de error (*[Rush et al., 1992]*), correspondiente a un punto de trabajo en el que las neuronas están trabajando en la zona de saturación de sus funciones de activación. Bajo estas condiciones, un algoritmo de optimización basado únicamente en la información dada por el gradiente de la función de error en ese punto y en ese instante, irá descendiendo parsimoniosamente por esta superficie hasta llegar a una región de mayor pendiente, donde la optimización se acelera.

Por otro lado, cada elemento de proceso de un PM ha de desempeñar una tarea concreta a la hora de la aproximación. Para determinar qué tarea le corresponde a cada cual, cada unidad sólo dispone de sus señales de entrada y de la señal de error que se retropropaga (en el caso de retropropagación). Por si esto fuera poco, todas estas señales están variando durante el aprendizaje, por lo que el objetivo resulta difícil de seguir. De aquí nace el llamado *efecto rebaño* (*[Fahlman & Lebiere, 1990]*), causado por la falta de comunicación entre unidades de una misma capa, que desemboca en una mala distribución de las tareas de cada unidad.

Supongamos que los elementos de proceso de una capa oculta han de realizar dos tareas y que cada uno de ellos es capaz de realizar una u otra indistintamente. Al no haber comunicación entre ellos, cada uno ha de decidirse por una u otra individualmente partiendo de las mismas señales de error y de las mismas entradas. Lo único que diferencia a un elemento de otro es su conjunto de pesos iniciales. Por lo tanto, si los pesos son muy parecidos, o si una tarea provoca una señal de error más fuerte o más coherente que la otra, todos los elementos de proceso se decidirán por la misma tarea, no ocupándose de la otra hasta que la primera ha quedado satifecha de forma redundante. Este posible intercambio de tareas queda reflejado en la hipersuperficie de error bajo forma de simetrías. Esta característica puede ser utilizada para reducir el espacio de búsqueda del óptimo valor de los pesos, como propone Robert Hecht-Nielsen (*[Hecht-Nielsen, 1990]*).

Podemos pues concluir que la inicialización de los pesos del PM tiene consecuencias decisivas en la etapa de entrenamiento (*[Guo & Saul, 1991]*), y que por lo tanto, requiere una atención especial. Una buena elección de pesos iniciales tiene como objetivos fundamentales el acelerar el proceso de aprendizaje y el encaminar la búsqueda del óptimo valor de los pesos a un óptimo global.

Para ello, la inicialización de los pesos ha de evitar en primer lugar el situarse en una zona saturada de la hipersuperficie de error, de tal forma que no se desvirtúe la información dada por el gradiente. Por otro lado, la inicialización ha de dotar a cada elemento de proceso de una "personalidad propia", de tal forma que se eviten los problemas asociados con el efecto rebaño. Esta asignación inicial de tareas no es una decisión cerrada que determine absolutamente el resultado final, ya que el aprendizaje tomará lo bueno de esta decisión inicial e irá desplazando lo malo hasta llegar a una distribución óptima.

Funcionamiento interno de un PM

Para ser capaces de realizar una distribución inicial de tareas, hemos de tener claro qué tipos de tareas puede realizar un elemento de proceso dentro de un perceptrón multicapa.

Limitándonos en principio a perceptrones multicapa de una capa oculta y de una salida, la relación funcional entre la entrada y la salida puede expresarse como:

$$y = \varphi_{21}\left(w_{210} + \sum_{j=1}^{h} w_{21j}\ \varphi_{1j}\left(w_{1j0} + \sum_{k=1}^{n} w_{1jk}\ x_k \right) \right)$$

Ecuación 4.33

Tomando además como ejemplo el caso de una sola entrada x, y con función de activación identidad en la salida, queda:

$$y = w_{210} + \sum_{j=1}^{h} w_{21j}\ \varphi_{1j}\left(w_{1j0} + w_{1j1}\, x \right)$$

Ecuación 4.34

Si utilizamos en las unidades ocultas funciones de activación tangentes hiperbólicas, que son aproximadamente lineales en el dominio [-1,1] con pendiente unidad, y se saturan según aumenta la magnitud del argumento, cada sumando de la ecuación anterior no es más que una función lineal de x en un pequeño intervalo de x,

acompañada de zonas de saturación. La longitud de cada intervalo de linealidad es inversamente proporcional a w_{1j1}, mientras que su localización queda determinada por la posición de su centro en $(-w_{1j0}/w_{1j1})$. La pendiente de la relación lineal viene dada por el producto $(w_{21j}\ w_{1j1})$.

Durante el entrenamiento la red aprende a aproximar la función deseada *g(x)* por superposición de estos elementos, desplazando los centros de los intervalos de linealidad, regulando sus longitudes, y ajustando las pendientes.

Es importante señalar que una unidad oculta sólo puede realizar una relación funcional monótona, pues su función de activación es estrictamente creciente. Por esta razón el número de zonas de monotonía de la función a aproximar, *g(x)*, establece un mínimo en el número de unidades ocultas necesarias para conseguir cierto nivel de precisión. Ilustremos estas ideas con la aproximación de la función cúbica:

$$g(x) = 2\,x^3 - x$$

Ecuación 4.35

con distintas estructuras de red. Para ello generaremos un conjunto de entrenamiento según la Ecuación 4.35 y entrenaremos tres PM, con distinto número de unidades ocultas en su única capa intermedia. Las figuras siguientes muestran valores normalizados al intervalo [-1,1] tanto para las entradas como para las salidas. Esta normalización es un forma de escalado del problema, como se vió en el Capítulo 2.

La Figura 4.26 corresponde a la aproximación de la función cúbica con un PM de estructura (1,3,1). Las tres primeras gráficas muestran la función realizada por cada unidad oculta; en ellas puede verse cómo cada unidad se centra en una zona determinada de la aproximación, generando las tres zonas de monotonía necesarias:

- La primera unidad (de índice *11*) se encarga de la zona central. Su salida se verá multiplicada por un peso negativo en la capa de salida.
- La segunda unidad (de índice *12*) se encarga de la zona derecha. Su salida se verá multiplicada por un peso positivo en la capa de salida.
- La tercera unidad (de índice *13*) se encarga de la zona izquierda. Su salida se verá multiplicada por un peso negativo en la capa de salida.

de tal forma que la suma total aproxima a *g(x)* con un buen grado de precisión (apenas pueden distinguirse en la última gráfica de la Figura 4.26 la función *g(x)* y la salida de la red).

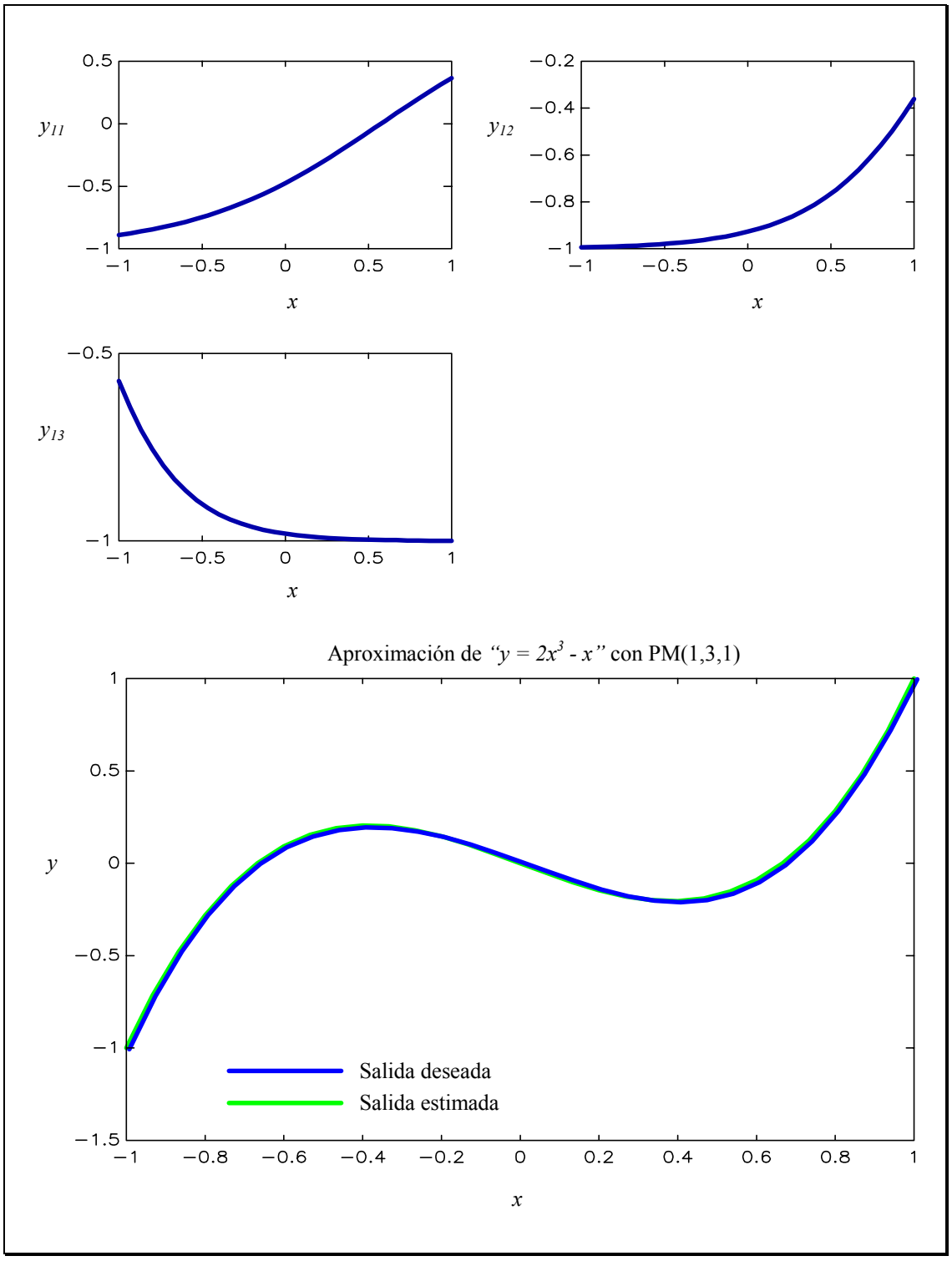

Figura 4.26: Distribución de tareas en la aproximación de una función cúbica con PM(1,3,1)

La Figura 4.27 muestra el resultado de la aproximación con tan sólo dos unidades ocultas. Con esta estructura (1,2,1), sólo podemos conseguir una relación funcional con dos zonas de monotonía, siendo pues imposible la aproximación precisa de la cúbica, al constar ésta de tres zonas de monotonía. La aplicación de la ley de aprendizaje ha dado con un conjunto óptimo de pesos, que aprovecha al máximo las posibilidades de esta estructura de PM.

La Figura 4.28 muestra los resultados de la aproximación con estructura (1,4,1), no apreciándose una mejora significativa respecto a la aproximación con el PM (1,3,1), estructura suficientemente capacitada para tal fin.

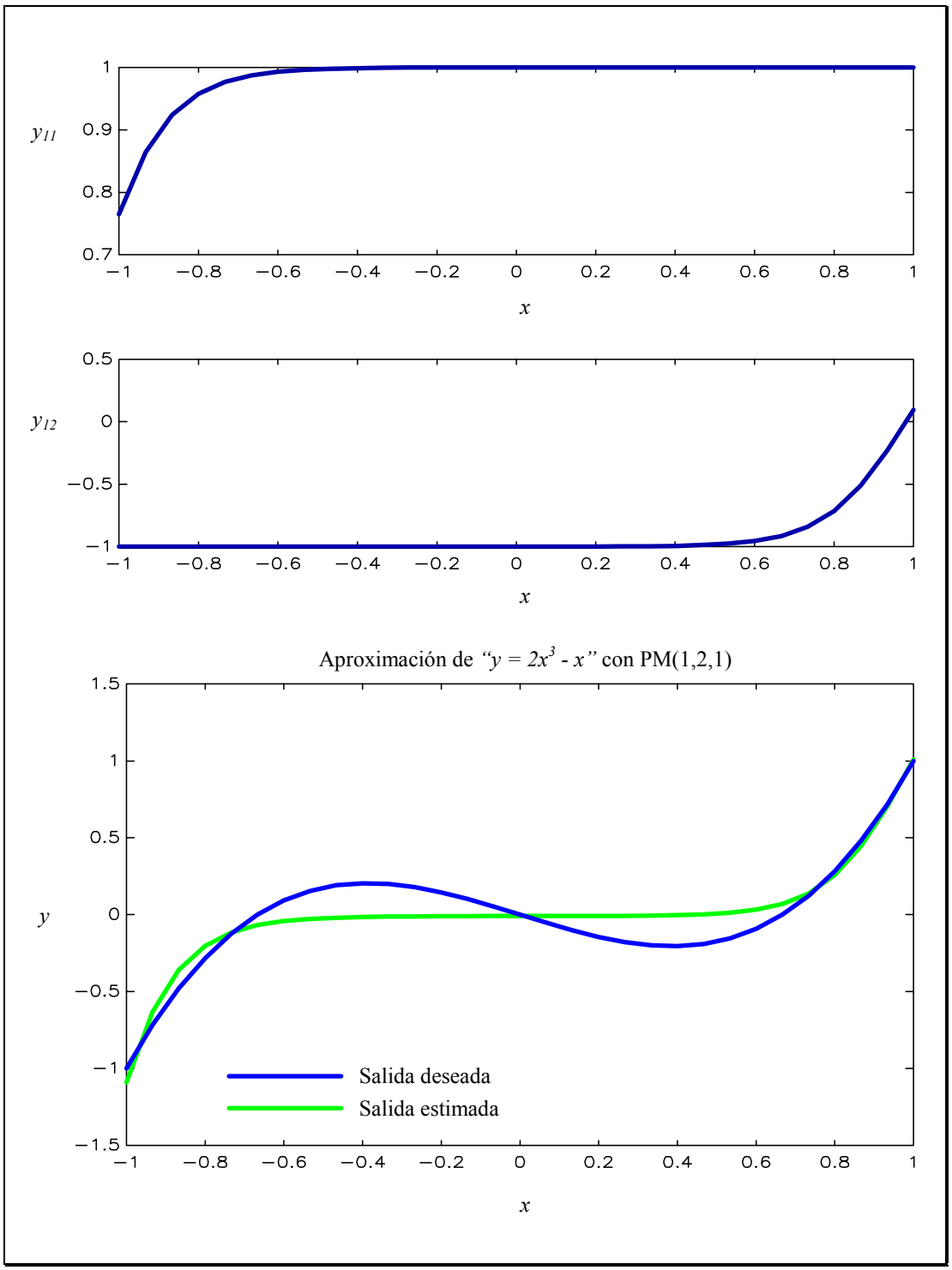

Figura 4.27: Distribución de tareas en la aproximación de una función cúbica con PM(1,2,1)

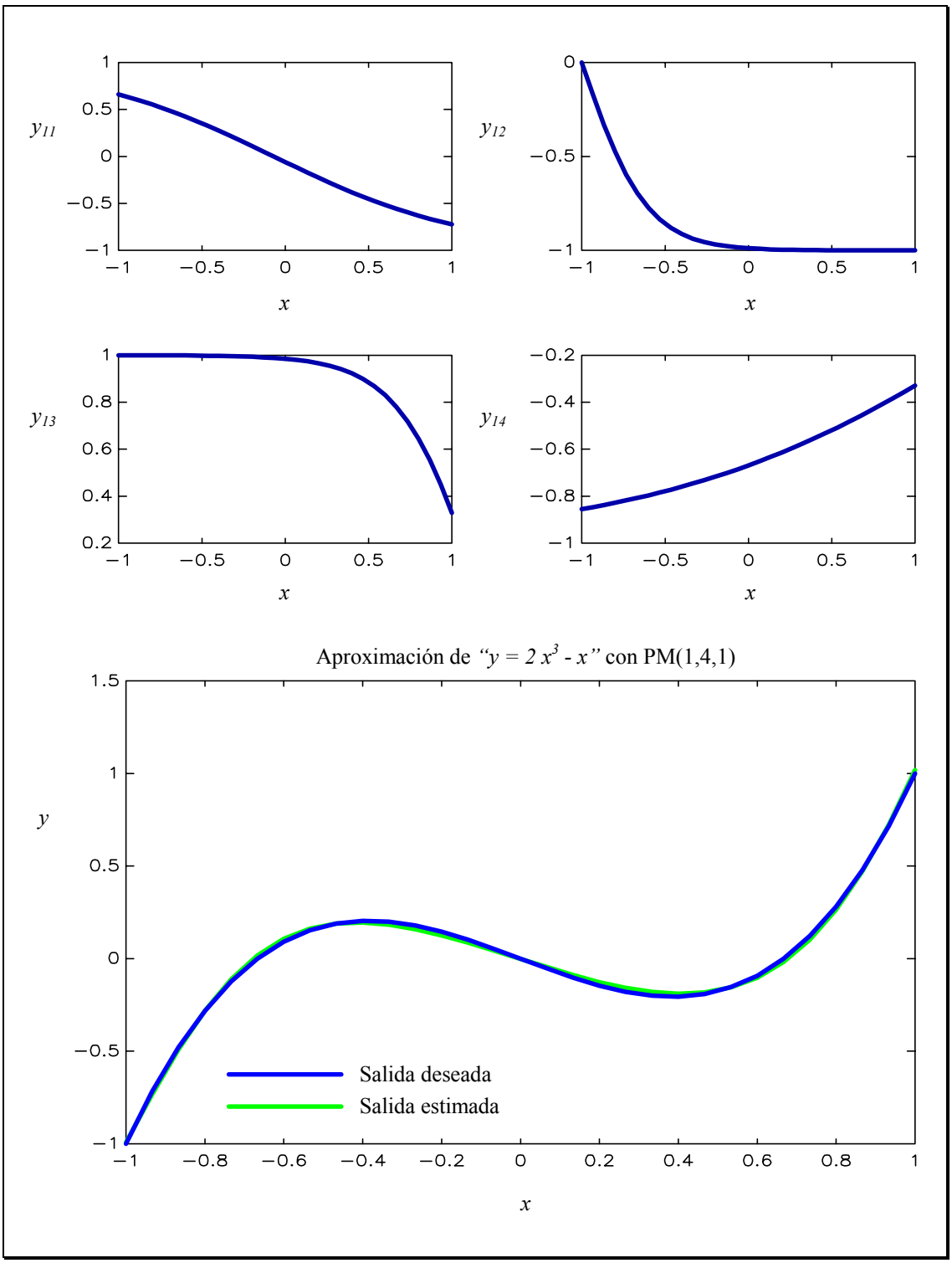

Figura 4.28: Distribución de tareas en la aproximación de una función cúbica con PM(1,4,1)

Algoritmo de inicialización de los pesos de un PM

Una vez esbozada la distribución de tareas de los elementos de proceso de un perceptrón bicapa (una capa oculta y otra de salida), pasemos a detallar un algoritmo de inicialización de los pesos de las neuronas ocultas de un PM con una única capa oculta, en el que se supone que las entradas han sido normalizadas al rango [-1;1].

El procedimiento para perceptrones bicapa con *n* entradas externas a la red (resultando perceptrones con *(n+1)* pesos en la capa oculta, al añadir la entrada umbral conectada al valor *+1*) y *h* elementos de proceso en la capa oculta, consta de los siguientes pasos (*[Nguyen & Widrow 1990]*):

- Generación de vectores de pesos asociados a las entradas externas, de direcciones uniformemente distribuidas:

$$w_{1jk} \text{ según una distribución } U[-1;1]^{n+1} \text{ para } j=1,...,h \text{ y } k=1,...,n$$

- Ajuste del tamaño de las regiones de aproximación, de tal forma que:

$$\sqrt{\sum_{k=1}^{n} w_{1jk}^2} = h^{\frac{1}{n}} \quad \text{para } j=1,...,h$$

- Localización uniformemente distribuida de los pesos umbrales (conectados al valor *+1*) en las regiones de aproximación:

$$w_{1j0} \text{ segun } U\left[-\sqrt{\sum_{k=1}^{n} w_{1jk}^2}\,;\sqrt{\sum_{k=1}^{n} w_{1jk}^2}\right] \text{ para } j = 1,\ldots,h$$

Con esta inicialización de los pesos de la capa oculta, se asegura una distinción de los elementos de proceso de la capa oculta, facilitando las labores del aprendizaje.

e) Entrenamiento del PM

La forma más habitual de entrenar PM es utilizar el algoritmo de retropropagación para evaluar el gradiente del error cuadrático medio de aproximación respecto de los pesos, y aplicar la Regla Delta con revisión por caso y momento de inercia. Esta estrategia de aprendizaje es conocida generalmente bajo el nombre de retropropagación, por extensión del método de cálculo del gradiente. Esta estrategia suele conllevar tiempos de aprendizaje muy elevados, por lo que es conveniente tener en cuenta los siguientes puntos:

- La utilización de funciones de activación impares ($\varphi(-s) = -\varphi(s)$), como la función tangente hiperbólica, suele ser más conveniente que el uso de funciones de activación como la sigmoidal.

- Cuando la capa de salida tiene funciones de activación saturadas (como la sigmoidal o las funciones hiperbólicas), los valores deseados de salida de los conjuntos de entrenamiento han de alejarse ligeramente de las zonas de saturación, con el fin de evitar que los pesos del PM se vayan a infinito. Así por ejemplo, si utilizamos funciones de activación tangentes hiperbólicas en la capa de salida (de rango [-1;1]), deberemos restringir las salidas deseadas a un intervalo del tipo [-0.95, 0.95].

- Todas las neuronas de un PM deberían llevar un mismo ritmo de entrenamiento. Para ello, y debido a la atenuación que sufre la señal retropropagada de error al ir pasando por las funciones de activación, es necesario utilizar un ratio de aprendizaje α_i distinto en cada capa, de tal forma que las primeras capas tengan un ratio mayor que las más próximas a la salida. Esto puede realizarse aplicando la Regla Delta-Bar-Delta presentada en el Capítulo 2.

- Suele ser más efectiva la revisión por caso que la revisión por época. Sin embargo, la revisión por caso es más difícil de paralelizar.

- La toma de ejemplos de entrenamiento en la revisión por caso debe ser aleatoria. Esto evita posibles comportamientos periódicos del aprendizaje y acelera este proceso de búsqueda.

Aunque la Regla Delta es el método de aprendizaje más utilizado en el campo conexionista, la experiencia demuestra que cuando el tiempo requerido para la evaluación de la función de error (teniendo en cuenta todos los ejemplos) no es excesivamente elevado, se obtienen mejores resultados aplicando algoritmos de optimización más poteentes, como los quasi-Newton de baja memoria. En las aplicaciones presentadas en esta tesis, los PM serán entrenados con el algoritmo LMQN1 presentado en el Capítulo 2.

5. La estructura PRBFN: un aproximador funcional basado en criterios probabilistas capaz de estimar funciones de densidad

Este capítulo introduce una nueva estructura de red neuronal basada en criterios probabilistas, que además de estimar las variables de salida contempladas, puede ser utilizada para estimar la función de densidad conjunta de su vector de entradas.

La estructura RBFN (*"Radial Basis Function Network"*) probabilista, a la que notaremos PRBFN, es además una estructura de fácil interpretación, lo que permite por un lado realizar una inicialización efectiva de los pesos, y extraer por otro información relevante del modelo finalmente ajustado.

Estas razones justifican su utilización en el ámbito del diagnóstico basado en modelos, donde además de poder ser utilizada como aproximador funcional en los modelos de funcionamiento normal, permitirá delimitar la región del espacio de entrada representada en el conjunto de entrenamiento.

5.1 Introducción

Dentro del campo de la aproximación funcional con redes neuronales supervisadas, el Perceptrón Multicapa (PM) ha sido la estrella indiscutible desde hace ya casi más de una década.

Muchos investigadores sin embargo han girado su atención hacia nuevas estructuras conexionistas inspiradas en técnicas matemáticas de interpolación, entre la que destaca de una forma muy significativa las redes de funciones base radiales ("Radial Basis Function Networks": RBFN) (*[Broomhead & Lowe, 1988], [Poggio & Girosi, 1989]*, *[Moody & Darken, 1988]*, *[Specht, 1990]*, *[Park & Sandberg, 1991]*, *[Chen et al., 1991], [Leonard et al., 1992], [Xu et al., 1994], [Webb, 1994],[Orr, 1995]*).

Las aplicaciones de estas estructuras a distintos campos de la técnica han sido multiples (*[Renals & Rohwer, 1989], [Kardirkamanathan et al, 1991], [Sanner & Slotine, 1992]*, *[Leonard & Kramer, 1993], [Damitha et al., 1993], [Chen et al., 1993], [Billings & Fung, 1995]*), así como los resultados teóricos que se han obtenido (*[Poggio & Girosi, 1989], [Girosi & Poggio, 1990], [Girosi et al, 1995], [Park & Sandberg, 1991]*).

Por otro lado se han relacionado las redes RBFN con el estimador de funciones de densidad probabilista de Parzen (*[Parzen, 1962]*), dando lugar a una variante probabilista de las redes RBFN conocida bajo el nombre de GRNN ("General Regression Neural Network") (*[Specht, 1990],[Specht, 1991]*). Esta red será el origen de las estructuras PRBFN propuestas en esta tesis, también relacionadas con las redes RBFN con conexiones laterales o normalizadas mencionadas en (*[Moody & Darken, 1988, 1989]*) y las estructuras conexionistas propuestas desde el campo de la lógica borrosa (*[Jang & Sun, 1993], [Wienholt, 1993]*).

En este capítulo daremos una breve introducción a la estructura RBFN, presentándola como un resultado directo de la teoría de la regularización. Posteriormente presentaremos la red GRNN, como paso previo a la descripción de las estructuras PRBFN.

5.2 Redes de funciones base radiales: RBFN

5.2.1 Aproximación funcional y regularización

Supongamos que queremos estimar con el aproximador *f* una cierta función *g*, de la que hemos obtenido, mediante muestreo aleatorio en presencia de ruido, un conjunto de muestras de la forma:

$$S = \{(\mathbf{x}[i], d[i]) \in \Re^n \times \Re,\ i = 1, \ldots, N$$

Ecuación 5.1

El problema de la estimación de *g* a partir de *S* es un problema mal enunciado, ya que existen infinitas soluciones. Para elegir una, hemos de imponer ciertas restricciones basadas en el conocimiento a priori que se tiene de la función que se quiere aproximar. La restricción más habitual es suponer que la función *g* es una función "suave", en el sentido que dos entradas similares han de generar salidas similares.

La teoría de la regularización (*[Girosi et al., 1995]*) contempla esta hipótesis definiendo una "función de suavidad" $\phi[f]$ que decrece con la lisura del aproximador funcional *f*. La función de error a minimizar durante el ajuste del aproximador se ve entonces modificada de la forma:

$$R = \sum_{i=1}^{N} (d[i] - f(\mathbf{x}[i])^2 + \lambda \phi[f]$$

Ecuación 5.2

de tal forma que se tienen en cuenta de forma simultánea los dos criterios considerados: el ajuste del aproximador al conjunto de datos *S*, y la suavidad de la función resultante. El parámetro de regularización $\lambda > 0$ controla el compromiso entre estos dos criterios, y suele ser establecido mediante validación cruzada.

Se demuestra que para una amplia gama de funciones ϕ, los minimizadores de la función de error definida en la Ecuación 5.2 tienen la misma forma general (*[Wahba, 1990], [Girosi et al., 1995]*).

Para llegar a este resultado es necesario expresar matemáticamente el concepto de "suavidad" de una función. En concreto, diremos que una función es más suave que otra, si su contenido espectral en altas frecuencias es más bajo. El contenido espectral en altas frecuencias de una función *f* puede ser medido utilizando un filtro

paso alto para medir la potencia (norma L_2) de alta frecuencia de la función *f*. De esta forma podemos definir funciones de suavidad de la forma:

$$\phi[f] = \int_{\Re^n} \frac{\left|\widetilde{f}(\boldsymbol{s})\right|^2}{\widetilde{G}(\boldsymbol{s})} d\boldsymbol{s}$$

Ecuación 5.3

siendo $\widetilde{f}(\boldsymbol{s})$ la transformada de Fourier de $f(\boldsymbol{x})$ y $\widetilde{G}(\boldsymbol{s})$ una función positiva que tiende a cero cuando $|\boldsymbol{s}| \to \infty$, de tal forma que $1/\widetilde{G}(\boldsymbol{s})$ es la función de transferencia del filtro paso alto.

Para una clase bien definida de funciones *G* (*[Madych & Nelson, 1990]*), la función ϕ así definida es una seminorma, con un núcleo (espacio nulo) Λ de dimensión finita. Si suponemos además que $\widetilde{G}(\boldsymbol{s})$ es simétrica, de tal forma que su antitransformada *G* sea real, es posible demostrar que la función que minimiza la Ecuación 5.2 tiene la forma:

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} v_i G(\boldsymbol{x} - \boldsymbol{x}[i]) + \sum_{j=1}^{K} \gamma_j \Psi_j(\boldsymbol{x})$$

Ecuación 5.4

siendo $\{\Psi j\}_{j=1,\ldots,K}$ una base del núcleo *K*-dimensional Λ de ϕ, que en la mayoría de los casos es un conjunto de polinomios, por lo que se conoce al segundo término de la Ecuación 5.4 bajo el nombre de término polinomial.

Los coeficientes v_i y γ_j dependen de los datos, y satisfacen el sistema lineal:

$$\begin{cases} (\boldsymbol{G} + \lambda \boldsymbol{I})\boldsymbol{v} + \boldsymbol{\Psi}^T \boldsymbol{\gamma} = \boldsymbol{d} \\ \boldsymbol{\Psi} \boldsymbol{v} = 0 \end{cases}$$

Ecuación 5.5

siendo ***I*** la matriz identidad, y

$$(\boldsymbol{d})_i = d[i], \quad (\boldsymbol{v})_i = v_i, \quad (\boldsymbol{\gamma})_i = \gamma_i$$
$$(\boldsymbol{G})_{ik} = G(\boldsymbol{x}[i] - \boldsymbol{x}[k]), \quad (\boldsymbol{\Psi})_{ji} = \Psi_j(\boldsymbol{x}[i])$$

Ecuación 5.6

La mayoría de los estabilizadores utilizados en aproximación funcional tienen simetría radial, es decir, satisfacen la ecuación:

$$\phi[f(\mathbf{x})] = \phi[f(\mathbf{Rx})]$$

Ecuación 5.7

para cualquier matriz de rotación $\mathbf{R}$. Estos estabilizadores invariantes a la rotación llevan asociados funciones base radiales: $G(\|\mathbf{x}\|)$, dando lugar a la técnica de aproximación funcional conocida bajo el nombre de funciones base radiales ("Radial Basis Functions") (*[Powell, 1985]*).

Si tomamos como estabilizador la función de suavidad:

$$\phi[f] = \int_{\Re^n} exp\left(\frac{\|\mathbf{s}\|^2}{\sigma^2}\right) \left|\tilde{f}(\mathbf{s})\right|^2 d\mathbf{s}$$

Ecuación 5.8

siendo σ una constante real positiva, obtenemos como función de transferencia del filtro:

$$\tilde{G}(\mathbf{s}) = exp\left(-\frac{\|s\|^2}{\sigma^2}\right)$$

Ecuación 5.9

$\phi[f]$ resulta ser en este caso una norma, y por lo tanto, su núcleo sólo contiene el elemento nulo, por lo que el término polinomial de la Ecuación 5.4 desaparece. La función base resulta ser entonces la función gaussiana:

$$G(\|\mathbf{x} - \mathbf{x}(i)\|) = exp(-\frac{\|\mathbf{x} - \mathbf{x}(i)\|^2}{\sigma^2})$$

Ecuación 5.10

siendo finalmente la expresión del aproximador:

$$f(\mathbf{x}) = \sum_{i=1}^{N} v_i \, exp\left(-\frac{\|\mathbf{x} - \mathbf{x}(i)\|^2}{\sigma^2}\right)$$

Ecuación 5.11

5.2.2 Estructuración conexionista: la red RBFN

En el apartado anterior hemos visto cómo la teoría de la regularización ofrece la justificación matemática para el uso de técnicas de aproximación funcional basadas en funciones base radiales (*[Powell, 1985]*). Partiendo de este esquema de aproximación surgieron las redes de funciones base radiales ("Radial Basis Function Networks": RBFN) (*[Broomhead & Lowe, 1988], [Poggio & Girosi, 1989]*, *[Moody & Darken, 1989]*), donde se limita el número de unidades radiales a una constante *h* independiente del número de muestras del conjunto *S*, y se introduce un factor de escala distinto en cada unidad. El aproximador resultante viene dado por:

$$f(\boldsymbol{x}) = \sum_{i=1}^{h} v_i \, exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{r}_i\|^2}{\sigma_i^2} \right)$$

Ecuación 5.12

donde $\boldsymbol{r}_i \in \Re^n$ son los centros o representantes de las unidades radiales, que ya no tienen por qué coincidir con las muestras, y σ_i son los anchos de las mismas. La estructura conexionista resultante, conocida bajo el nombre de red RBFN, consta de dos capas de elementos de proceso (sin tener en cuenta la capa de entrada):

- la capa oculta, compuesta por *h* unidades radiales totalmente conectadas al vector de entradas. Esta capa transforma el espacio *n*-dimensional de entrada en otro espacio *h*-dimensional, donde cada componente está relacionada con una medida de la distancia del vector de entradas al vector centro de cada unidad.

- la capa de salida, compuesta por una unidad de salida (la extensión al caso multidimensional es inmediata), que realiza una suma ponderada de las salidas de las unidades radiales.

La estructura resultante aparece en la Figura 5.1:

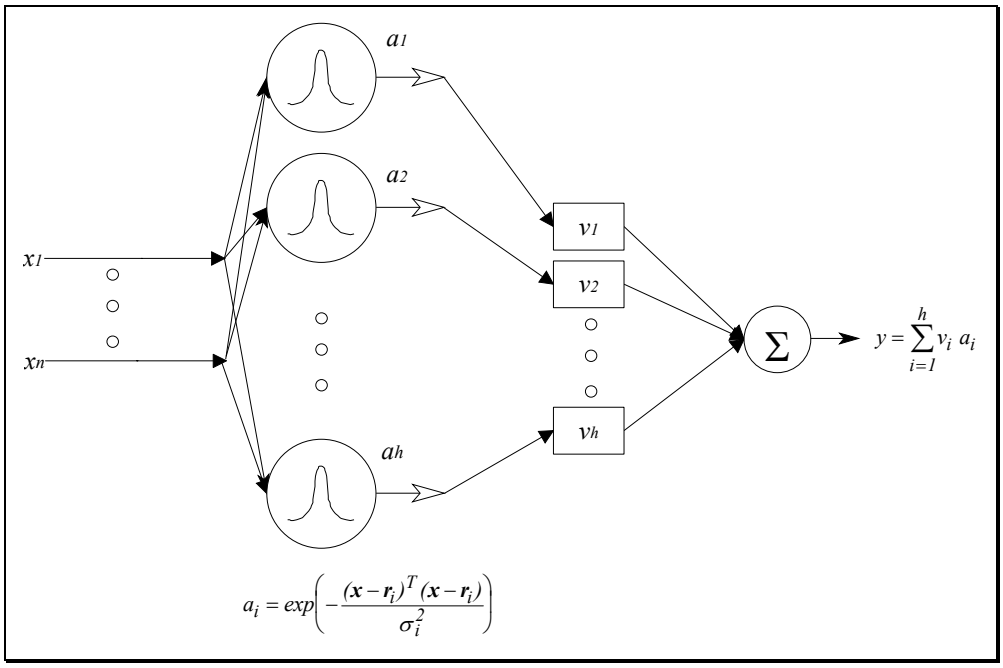

Figura 5.1: Estructura de la red RBFN

5.2.3 Entrenamiento

El vector $\boldsymbol{w}$ de parámetros libres de la red RBFN está compuesto por tres tipos de elementos: los vectores $\boldsymbol{r}_i \in \Re^n$ que son los centros de las unidades radiales, los anchos σ_i de dichas unidades, y los pesos v_j de la unidad de salida.

El entrenamiento de una red RBFN deberá por tanto localizar los vectores centrales de las unidades radiales en el espacio de entrada, ajustar los soportes de las unidades radiales actuando sobre sus anchos, y determinar por último el valor de los pesos de la capa de salida.

El método de entrenamiento más extendido (*[Moody & Darken, 1989], [Leonard & Kramer, 1993]*) consta de los siguientes pasos:

1.- Ubicación no supervisada de los h vectores centrales de las unidades radiales ($\boldsymbol{r}_i$) mediante la aplicación de un algoritmo de agrupamiento ("clustering") sobre el conjunto de entrenamiento. El algoritmo más utilizado es el algoritmo *k-means* (*[Hartigan, 1975]*).

2.- Determinación de los h "anchos" de las unidades radiales (σ_i) mediante la aplicación de un heurístico ("p-nearest-neighbor") que asegura el ligero solapamiento de los soportes de las unidades radiales. Para ello se toma como valor de σ_i la media de las distancias euclídeas de $\boldsymbol{r}_i$ a los p centros más cercanos, siendo habitual el tomar $p=2$.

3.- El valor de los pesos de la capa de salida (v_i) se obtiene mediante regresión lineal, una vez que han sido fijados los restantes parámetros.

De esta forma se obtiene un algoritmo de aprendizaje de convergencia polinomial, mucho más rápido que el equivalente PM entrenado según la regla Delta.

La literatura recoge todo un abanico de estrategias de aprendizaje para las redes RBFN, que tratan de solventar las limitaciones introducidas al ajustar los centros y los anchos de las unidades radiales de una forma no supervisada (*[Chen et al., 1991], [Musavi et al., 1992],[Wienholt, 1993],[Bianchini & Frasconi, 1995],[Orr, 1995]*).

Entre estas estrategias cabe destacar la expuesta en *[Chen et al., 1991]*, donde se realiza la selección de los centros de las unidades radiales de una forma secuencial, basada en el método de mínimos cuadrados ortogonales (cada centro seleccionado maximiza el incremento de la varianza explicada de la variable de salida).

Otra posible alternativa es utilizar el algoritmo descrito anteriormente para inicializar todos los parámetros o pesos de la red, y aplicar posteriormente un algoritmo de optimización, como los descritos en el Capítulo 2, para ajustar de forma supervisada cada uno de los parámetros de la red. La gran ventaja de este procedimiento es que el punto de partida es ya muy bueno, y en general bastarán con unas pocas iteraciones para alcanzar una solución aceptable.

5.2.4 Propiedades

a) Capacidad de aproximación universal

Al igual que en el caso del PM, se ha probado la capacidad de aproximación universal de las redes RBFN (*[Park & Sandberg, 1991]*) en el espacio de funciones absolutamente integrables.

b) Propiedad de óptima aproximación

Se ha probado también la propiedad de óptima aproximación de las redes RBFN (*[Girosi & Poggio, 1990]*), que asegura la existencia de la estructura que minimiza el

error de aproximación. Esta propiedad no la cumple sin embargo el Perceptrón Multicapa.

c) El problema de la dimensionalidad

Los aproximadores de tipo RBFN padecen el grave problema del incremento exponencial del número de unidades radiales requerido al aumentar la dimensión del espacio de entrada. El origen del problema radica en la construcción de la hipersuperficie de aproximación a partir de hiperesferas, que dan la misma importancia a todas las dimensiones. Este problema puede suavizarse utilizando redes RBFN generalizadas (*[Poggio & Girosi, 1990]*), donde los soportes de las unidades radiales dejan de ser hiperesferas para convertirse en hiperelipses, o redes RBFN normalizadas.

5.2.5 Comparación de las redes RBFN y el PM

Tanto el PM como la red RBFN son ejemplos de redes neuronales multicapa del tipo "feedforward", con capacidad de aproximación universal. Existen sin embargo ciertas diferencias significativas en su modo de operar que hay que tener en cuenta:

1.- Las redes RBFN sólo tienen una capa oculta, mientras que los PM pueden tener varias.

2.- Todos los elementos de proceso de un PM tienen la misma estructura (aunque su función de activación sea distinta), mientras que las unidades radiales de las redes RBFN son completamente distintas a sus unidades de salida.

3.- La función de activación de las unidades radiales de las redes RBFN requiere el cómputo de la distancia euclídea entre el vector de entradas y el vector representante de la unidad. En el PM sin embargo sólo se requiere el cálculo del producto escalar entre el vector de entradas de la unidad y su vector de pesos.

4.- El PM realiza aproximaciones globales de las transformaciones de entrada/salida. Son capaces por lo tanto de generalizar en regiones del espacio de entrada no representadas en el conjunto de entrenamiento. Las redes RBFN sin embargo, al utilizar no linealidades decrecientes de forma exponencial, realizan aproximaciones locales. Esta propiedad acelera su proceso de aprendizaje, pero puede hacer que el número requerido de unidades radiales tenga que ser muy elevado. Su comportamiento en regiones del espacio no representadas en el conjunto de entrenamiento es muy malo, por lo que suele ser imprescindible añadir una medida

de extrapolación que refleje el grado de representación que el vector de entradas tiene en el conjunto de entrenamiento.

5.2.6 La red RBFN normalizada

Una variante de las redes RBFN aparecida en *[Moody & Darken, 1988]*, modifica la expresión del aproximador de la Ecuación 5.12, dividiendo por la suma de las activaciones de las unidades radiales:

$$f(\boldsymbol{x}) = \frac{\sum_{i=1}^{h} v_i \, exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{r}_i\|^2}{\sigma_i^2} \right)}{\sum_{i=1}^{h} exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{r}_i\|^2}{\sigma_i^2} \right)}$$

Ecuación 5.13

La estructuración conexionista de este aproximador es conocida bajo los nombres de red RBFN normalizada y red RBFN con conexiones laterales. En el siguiente apartado daremos una interpretación probabilista de una variante de esta estructura, conocida bajo el nombre de GRNN ("General Regression Neural Network").

5.3 Origen de las estructuras PRBFN: la red GRNN

La estructura de la red PRBFN puede ser interpretada en términos probabilistas como una herramienta de regresión. Siguiendo este enfoque se presenta a continuación la estructura de la red GRNN ("General Regression Neural Network") introducida por Donald F. Specht (*[Specht, 91]*) que servirá como punto de partida para el desarrollo de la estructura definitiva.

5.3.1 Regresión generalizada

Sea $p_r(\boldsymbol{X},D)$ la función de densidad conjunta del vector n-dimensional de variables aleatorias $\boldsymbol{X}$ y la variable escalar aleatoria D. Sea $\boldsymbol{x}$ una medida particular de $\boldsymbol{X}$. La esperanza condicionada de D dado $\boldsymbol{x}$ viene dada por:

$$E\left(\frac{D}{\boldsymbol{x}}\right) = \frac{\int_{-\infty}^{+\infty} z \cdot p_r(\boldsymbol{x}, z) \cdot dz}{\int_{-\infty}^{+\infty} p_r(\boldsymbol{x}, z) \cdot dz}$$

Ecuación 5.14

Cuando la función de densidad $p_r(\boldsymbol{X},D)$ no es conocida, puede ser estimada partiendo de un conjunto de observaciones de $(\boldsymbol{X},D)$. En nuestro caso haremos uso de una clase de estimadores consistentes propuestos por Parzen (*[Parzen, 62]*) y ampliados al caso multidimensional por Cacoullos (*[Cacoullos, 66]*), que suponen una función de densidad continua y con primeras derivadas "pequeñas". El estimador de probabilidad $p(\boldsymbol{x},d)$ parte de un conjunto de N muestras $\{(\boldsymbol{x}[i],d[i]),\ i=1,...,N\}$ de las variables aleatorias $(\boldsymbol{X},D)$ y queda definido por:

$$p(\boldsymbol{x}, d) = \frac{1}{(2\pi)^{(n+1)/2}\,\sigma^{(n+1)}} \cdot \frac{1}{N} \sum_{i=1}^{N} \left(exp\left(-\frac{(\boldsymbol{x} - \boldsymbol{x}[i])^T (\boldsymbol{x} - \boldsymbol{x}[i])}{2\sigma^2} \right) \cdot exp\left(-\frac{(d - d[i])^2}{2\sigma^2} \right) \right)$$

Ecuación 5.15

La interpretación geométrica de la *Ecuación 5.15* pasa por centrar en cada muestra un campo de probabilidad de radio σ de tal forma que la función de densidad estimada es suma de estas probabilidades.

Sustituyendo la función de densidad estimada *(Ecuación 5.15)* en la expresión de la esperanza condicionada *(Ecuación 5.14)* y realizando las integraciones se obtiene la expresión del valor estimado de *d* en el punto $\boldsymbol{x}$ ($y(\boldsymbol{x})$):

$$y(\boldsymbol{x}) = \frac{\sum_{i=1}^{N} d[i] \cdot exp\left(-\frac{(\boldsymbol{x} - \boldsymbol{x}[i])^T (\boldsymbol{x} - \boldsymbol{x}[i])}{2\sigma^2} \right)}{\sum_{i=1}^{N} exp\left(-\frac{(\boldsymbol{x} - \boldsymbol{x}[i])^T (\boldsymbol{x} - \boldsymbol{x}[i])}{2\sigma^2} \right)}$$

Ecuación 5.16

Se demuestra que el estimador de función de densidad dado por *(Ecuación 5.15)* es consistente (convergente asintóticamente) si la función $\sigma=\sigma(N)$ es función decreciente de *N* tal que:

$$\begin{cases} lim_{N\to\infty}(\sigma(N)) = 0 \\ lim_{N\to\infty}\left(N\sigma^n(N)\right) = \infty \end{cases}$$

Ecuación 5.17

La estimación $y(\boldsymbol{x})$ puede ser vista como la media ponderada de todos los valores observados *d[i]*, donde cada valor queda ponderado de forma exponencial según su distancia euclídea a $\boldsymbol{x}[i]$.

La *Figura* 5.2 muestra un ejemplo de estimación de la función de densidad de una población según el estimador de Parzen con $\sigma=0.1$. La población en este caso está compuesta por un conjunto de puntos del plano de los cuales la mitad se distribuye según una gausiana bidimensional de media *(0.25,0.25)* y matriz de covarianzas $[0.01\boldsymbol{I_2}]$ (siendo $\boldsymbol{I_2}$ la matriz identidad de orden 2), y el resto según una gausiana bidimensional de media *(0.75,0.75)* y misma matriz de covarianzas.

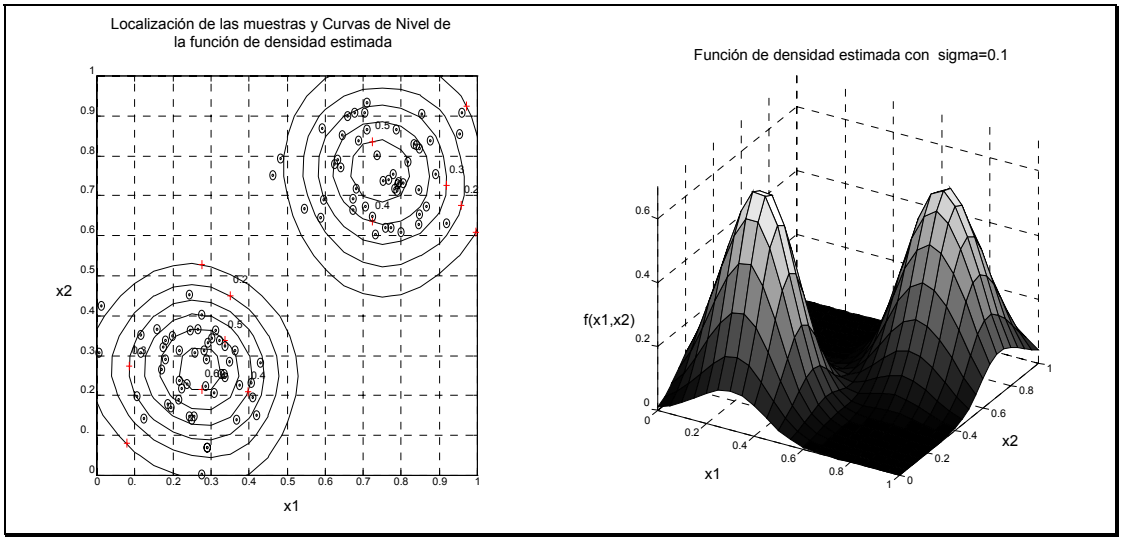

Figura 5.2: Ejemplo de estimación de la función de densidad con $\sigma = 0.1$

Al aumentar el parámetro σ, se suaviza la función de densidad estimada de tal forma que en el límite tiende a convertirse en una gausiana multidimensional de matriz de covarianzas $\sigma^2\boldsymbol{I}$. Esta tendencia queda reflejada en la *Figura* 5.3. Cuando σ alcanza valores muy elevados, *y* toma el valor de la media de las muestras *d[i]*.

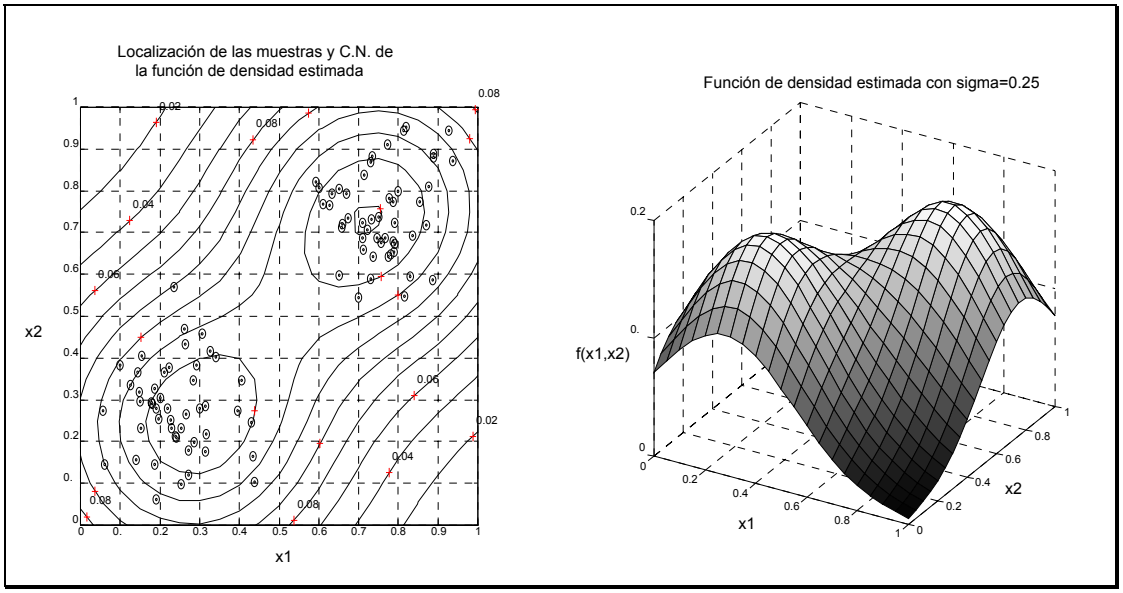

Figura 5.3: Ejemplo de estimación de la función de densidad con $\sigma = 0.25$

Al disminuir el valor de σ, la función de densidad estimada puede tomar formas no gausianas, pero valores excesivamente pequeños pueden hacer que observaciones extrañas que hayan quedado incluídas en las muestras iniciales influyan excesivamente en la estimación. Esto es debido a que cuando σ tiende a cero, *y(**x**)* toma el valor de la muestra más cercana, sin tener en cuenta más valores (ver *Figura* 5.4 , donde es necesario tener en cuenta los efectos del muestreo en la representación).

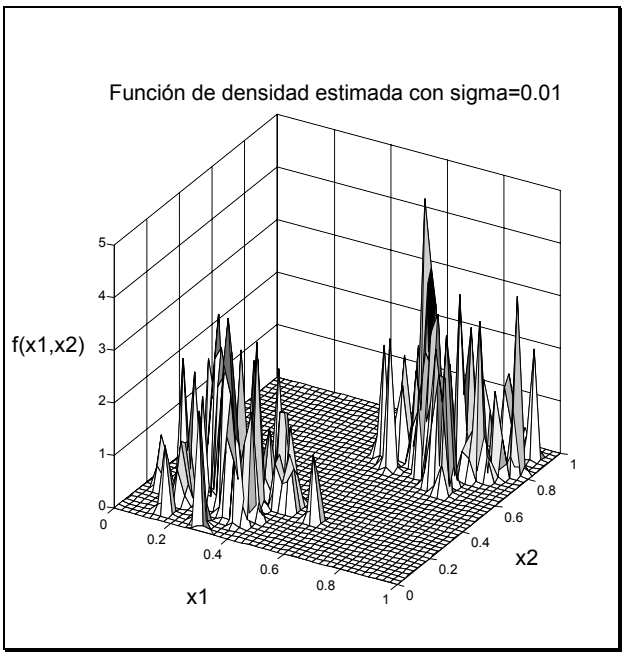

Figura 5.4 : Ejemplo de estimación de la función de densidad con σ = 0.01

Con valores intermedios de σ (como es el caso de la *Figura* 5.2), varias muestras entran en consideración, dando mayor peso a las más cercanas. Si la función de densidad original no es conocida, no es posible calcular un valor óptimo de σ dadas *N* observaciones, siendo entonces necesario encontrar de forma empírica un valor adecuado de σ. Esta búsqueda se facilita cuando la función de densidad estimada es utilizada como un medio de regresión, ya que surge entonces un medio natural de evaluar la bondad de σ, como es el error cuadrático medio de la estimación de *d[i]* por *y(**x**[i])*.

Este parámetro puede ser entonces determinado por el método de validación cruzada ("cross-validation"), que consiste en calcular, para distintos valores de σ, el error cuadrático medio cometido en la estimación de *d[k]* al dejar fuera la muestra *k* :

$$Err(\sigma) = \frac{1}{N}\sum_{k=1}^{N}\left(\frac{\sum_{\substack{i=1\\ i\neq k}}^{N} d[i]\cdot exp\left(-\frac{(\mathbf{x}[k]-\mathbf{x}[i])^T(\mathbf{x}[k]-\mathbf{x}[i])}{2\sigma^2}\right)}{\sum_{\substack{i=1\\ i\neq k}}^{N} exp\left(-\frac{(\mathbf{x}[k]-\mathbf{x}[i])^T(\mathbf{x}[k]-\mathbf{x}[i])}{2\sigma^2}\right)} - d[k]\right)^2$$

Ecuación 5.18

y quedarse con el óptimo valor de σ hallado.

Típicamente la curva *Err(σ)* presenta un "amplio" margen de valores de σ con errores cercanos al error óptimo, no siendo pues difícil escoger un buen valor sin un elevado número de pruebas.

5.3.2 Estandarización de las entradas

La utilización de la distancia euclídea como medida de proximidad entre vectores obliga a realizar un paso previo de preprocesamiento de las variables de entrada, de tal forma que todas ellas tengan aproximadamente mismo rango o varianza similar.

Esta necesidad viene impuesta por la utilización en la estimación de la función de densidad de un "*kernel"* que tiene el mismo ancho en cada dimensión y se reforzará cuando sea necesario aplicar algoritmos de agrupamiento al conjunto de observaciones, con vistas a reducir el número de datos que intervienen en la estimación. La normalización exacta no es necesaria, no teniendo que revisar los parámetros de escalado cada vez que añadamos una nueva observación al conjunto de datos.

La transformación más extendida es la llamada estandarización (ver Capítulo 2), que consiste en aplicar una transformación lineal a los datos de tal forma que cada una de las *n* variables tenga media muestral nula y varianza muestral unidad (*[Kaufman & Rousseeuw, 1990]*).

5.3.3 Agrupamiento y reajuste dinámico

En ciertos problemas en los que el número de observaciones es reducido, puede ser conveniente utilizar todos los datos disponibles a la hora de estimar una nueva salida según la *Ecuación 5.16* .

Sin embargo, en muchas aplicaciones reales, el número de observaciones disponibles puede ser tan elevado que no resulte práctico asignar un nuevo nodo a cada dato. Por esta razón es necesario aplicar algoritmos de agrupamiento ("clustering") que seleccionen un número adecuado de *representantes* de las observaciones según su distribución en el espacio de entrada, de tal forma que la estimación dada por la *Ecuación 5.16* no precise de un número de nodos excesivamente elevado.

La idea general de los algoritmos de agrupamiento es dividir el conjunto total de las muestras en subconjuntos tales que la distancia entre muestras del mismo subconjunto sea mínima y la distancia entre muestras de subconjuntos distintos sea máxima. A cada subconjunto o grupo se le asigna un *centro* como representante (vector $\boldsymbol{r_i}$) de las muestras asociadas.

Entre los métodos de agrupamiento caben destacar por su importancia la *Cuantización Vectorial* (*[Burrascano, 1991]*) y las distintas variantes del algoritmo *"K-means"* (*[Hartigan, 1975]*). El algoritmo de agrupamiento desarrollado en este trabajo es un algoritmo mixto que será descrito en siguientes apartados.

Sea cual fuere el algoritmo de agrupamiento utilizado, asignaremos a cada nodo una nueva variable K_i que representa el número de muestras que han quedado representadas por el centro del grupo *i*. La *Ecuación 5.16* puede entonces convertirse en:

$$y(\boldsymbol{x}) = \frac{\sum_{i=1}^{h} A_i \cdot exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{r_i})^T(\boldsymbol{x}-\boldsymbol{r_i})}{2\sigma^2}\right)}{\sum_{i=1}^{h} B_i \cdot exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{r_i})^T(\boldsymbol{x}-\boldsymbol{r_i})}{2\sigma^2}\right)}$$

Ecuación 5.19

con:

$$\begin{cases} A_i(k) = A_i(k-1) + d(j) \\ B_i(k) = B_i(k-1) + 1 \end{cases}$$

Ecuación 5.20

y siendo:

- $A_i[k]$ y $B_i[k]$ el valor de los parámetros del grupo *i* tras haber sido procesadas *k* observaciones, que se actualizan cada vez que un dato de entrenamiento (de índice *j*) queda asociado al grupo *i*.

- $h<N$ el número de "clusters"

La definición recursiva de los parámetros $A_i[k]$ y $B_i[k]$ permite incluir una *función de olvido* que permita actualizar de forma dinámica el modelo de regresión obtenido con los datos más recientes. Esta propiedad es aconsejable cuando se trata de modelar sistemas de características que cambian con el tiempo, como los procesos de envejecimiento. De esta forma podemos modificar la *Ecuación 5.20* para obtener:

$$\begin{cases} A_i[k] = \dfrac{\tau - 1}{\tau} A_i[k-1] + \dfrac{1}{\tau} d[k] \\ B_i[k] = \dfrac{\tau - 1}{\tau} B_i[k-1] + \dfrac{1}{\tau} \end{cases} \quad \text{si la nueva muestra queda asociada al "cluster " } i$$

$$\begin{cases} A_i[k] = \dfrac{\tau - 1}{\tau} A_i[k-1] \\ B_i[k] = \dfrac{\tau - 1}{\tau} B_i[k-1] \end{cases} \quad \text{si la nueva muestra queda asociada a un "cluster " } \neq i$$

Ecuación 5.21

donde el parámetro τ puede ser interpretado como la constante de tiempo (medida en número de muestras y no en tiempo de muestreo) de una función exponencial.

En la práctica suele ser aconsejable también imponer un umbral mínimo a B_i, de tal forma que cuando pase un intervalo de tiempo suficientemente amplio sin que un determinado grupo se haya visto actualizado por una nueva muestra, se elimine el nodo correspondiente y sea sustituido por uno nuevo.

Sin embargo esta práctica no es recomendable cuando se tratan sistemas con distintos modos de operación, ya que no conviene en estos casos olvidar la información concerniente a modos de operación distintos al actual. Para ello puede definirse un radio de olvido selectivo ρ, de tal forma que la *Ecuación 5.21* sea aplicable en cada paso a aquellos centros de grupo que distan menos de ρ del centro del grupo i.

5.3.4 Estructuración conexionista

Dada la estrategia de estimación expresada por la *Ecuación 5.16* o la *Ecuación 5.19* , podemos plantear una estructura conexionista que materialice la transformación del espacio de entradas en el de salida. Esta estructura fue bautizada con el nombre de GRNN (*"General Regression Neural Network"*) por Donald F. Specht (*[Specht, 1991]*) y queda representada en la *Figura 5.5*:

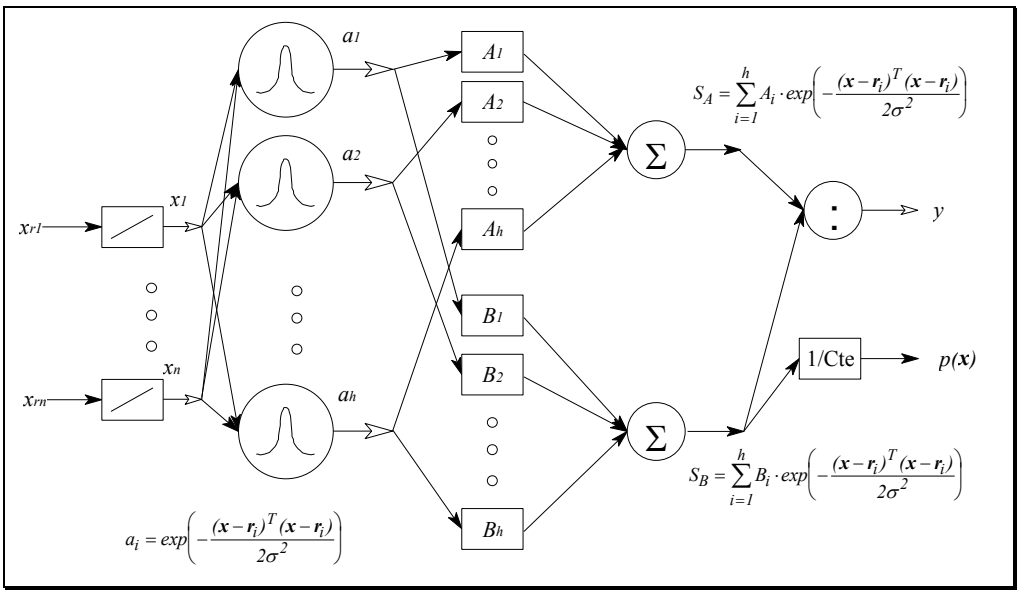

Figura 5.5: Estructura de la red GRNN

La red está compuesta por cuatro capas de elementos de proceso por los que las señales avanzan en sentido directo ("feedforward"):

- La ***capa de entrada*** está formada por elementos de proceso que realizan una transformación lineal de sus entradas. En esta capa se lleva a cabo la normalización de los datos aplicando un proceso de estandarización o de normalización del rango de cada una de las variables. Los parámetros de la transformación habrán sido predefinidos en una etapa previa de análisis del conjunto de entrenamiento. El resultado de la transformación son las variables normalizadas x_j que se distribuyen en la capa siguiente.

- La segunda capa es la ***capa de unidades radiales***. Cada unidad radial almacena un vector *n*-dimensional correspondiente al centro del grupo que representa y que será llamado representante de la unidad (al representante de la unidad radial *i* lo notaremos $\mathbf{r_i}$). Todas las unidades tienen en este caso el mismo ancho σ, por lo que la activación de la unidad radial *i* vendrá dada por:

$$a_i(\mathbf{x}) = exp\left(-\frac{(\mathbf{x}-\mathbf{r}_i)^T(\mathbf{x}-\mathbf{r}_i)}{2\sigma^2}\right)$$

Ecuación 5.22

siendo pues una función Gaussiana que en el caso unidimensional, con $\sigma=1$ y $r=0$, toma la forma de la *Figura* 5.6:

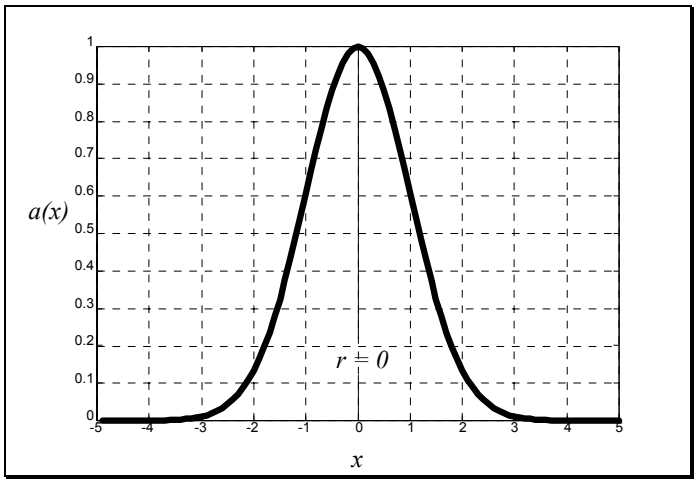

Figura 5.6: Función de activación Gaussiana

- Las salidas de la capa de unidades radiales estimulan a las ***unidades sumatorias*** a través de las conexiones ponderadas por los coeficientes A_i y B_i, de tal forma que a la salida de la primera unidad (de tipo "A") se obtiene:

$$S_A = y \cdot p(\mathbf{x}) \cdot Cte = \sum_{i=1}^{h} A_i \cdot exp\left(-\frac{(\mathbf{x}-\mathbf{r}_i)^T (\mathbf{x}-\mathbf{r}_i)}{2\sigma^2} \right)$$

Ecuación 5.23

mientras que la salida de la segunda unidad (de tipo "B") queda expresada por:

$$S_B = p(\mathbf{x}) \cdot Cte = \sum_{i=1}^{h} B_i \cdot exp\left(-\frac{(\mathbf{x}-\mathbf{r}_i)^T (\mathbf{x}-\mathbf{r}_i)}{2\sigma^2} \right)$$

Ecuación 5.24

siendo:

$$Cte = (2\pi)^{\frac{n}{2}} \cdot \sigma^n \cdot h$$

Ecuación 5.25

- De esta forma se obtiene en las unidades de la capa de salida el valor estimado de la salida como cociente de las salidas de las unidades sumatorias de la capa anterior ($y=S_A/S_B$), y la estimación de la función de densidad de $\boldsymbol{x}$, como señal proporcional a S_B ($p(\boldsymbol{x}) = S_B/Cte$).

En el caso general de que la salida fuese *m*-dimensional, bastaría con incluir *m* unidades sumatorias del tipo "A" con sus correspondientes coeficientes, y *m* unidades de estimación de los valores de salida, manteniendo una unidad sumatoria del tipo "B" y una unidad estimadora de la función de densidad de las entradas.

5.3.5 Ejemplo de aplicación de la red GRNN

En este apartado se presenta un ejemplo de aproximación funcional especialmente escogido para desvelar las ventajas de la estructura de la red PRBFN frente a la GRNN.

En concreto se trata de aproximar la función $d=g(x_1,x_2)=sen(x_1)$ partiendo de un conjunto de entrenamiento, formado por (9x9) muestras uniformemente distribuídas en la región ([0,8]x[0,8]) como muestra la Figura 5.7, y de un conjunto de test formado por (17x17) muestras uniformemente distribuídas en la misma región (ver Figura 5.8).

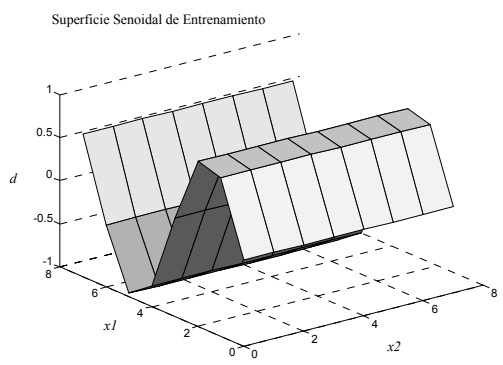

Figura 5.7: Superficie senoidal de entrenamiento

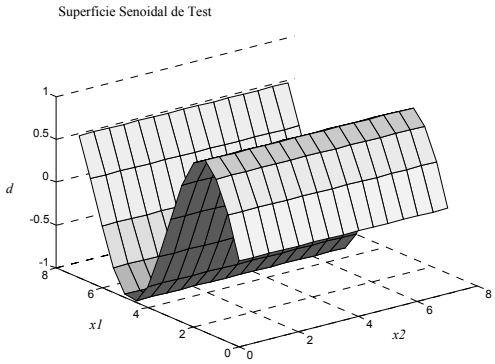

Figura 5.8: Superficie senoidal de test

Este método de selección de los conjuntos de entrenamiento y test (conjunto de test mucho más rico en información que el de entrenamiento) convierten al error de estimación del conjunto de test en una medida efectiva de la capacidad de generalización de las distintas estructuras conexionistas, entrenadas para minimizar el error de estimación del conjunto de entrenamiento.

La red GRNN utilizada en este caso para aproximar la función senoidal fue una red de 2 unidades de entrada, 25 unidades radiales determinadas por un algoritmo de agrupamiento, y una unidad de salida. Las variables A_i y B_i se determinaron según la *Ecuación 5.20* , quedando definida la salida de la red por la *Ecuación 5.19* .

El único grado de libertad que queda por determinar es el ancho σ de las unidades radiales. Para ello emplearemos en este caso un método computacionalmente costoso pero ilustrativo de la influencia de σ en la estimación final, como es el de calcular el error de estimación del conjunto de test con distintos valores de σ y quedarse con el valor óptimo. La *Figura 5.9* muestra los resultados así obtenidos:

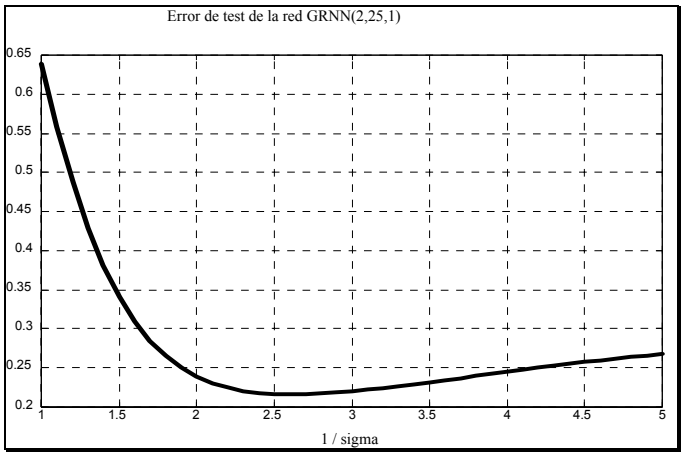

Figura 5.9: Error cuadrático medio cometido por la red GRNN(2,25,1) en la estimación del conjunto de test de la superficie senoidal como función de 1/σ

De la figura anterior se desprende un valor óptimo de *(1/σ) = 2.6* . Con el correspondiente valor de σ, se obtiene una estimación de la superficie senoidal de test como la mostrada en la *Figura 5.10*:

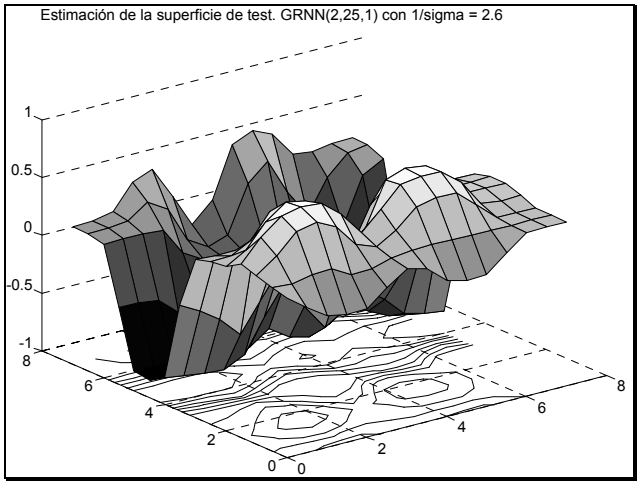

Figura 5.10: Superficie senoidal de test estimada por la red GRNN(2,25,1) con 1/σ=2.6

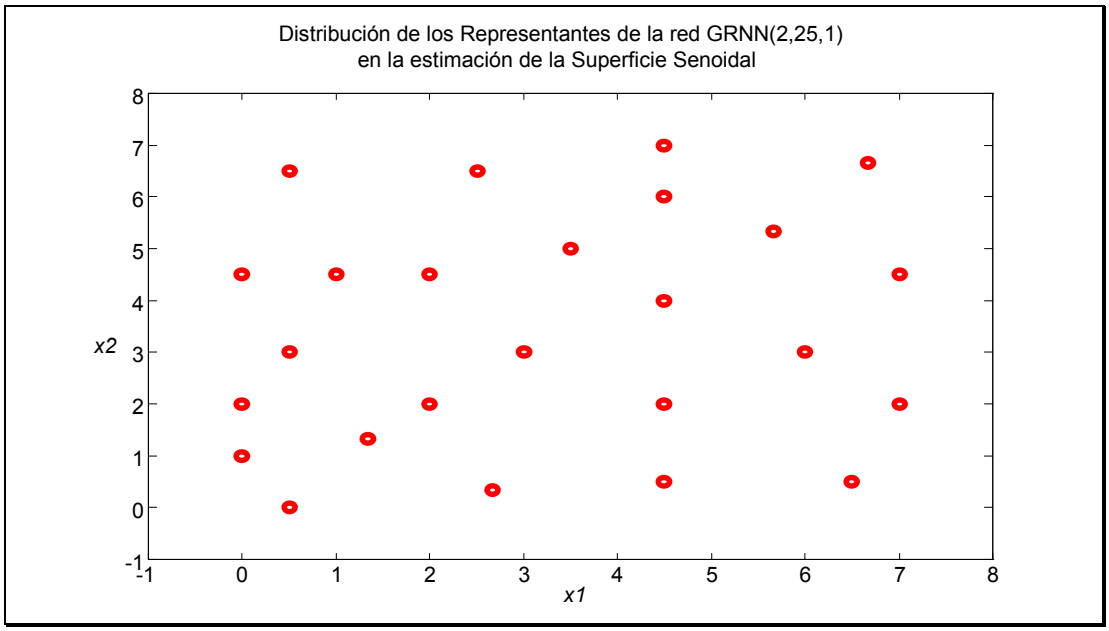

Figura 5.11: Distribución de los Representantes de las Unidades Radiales de la red GRNN(2,25,1) en la aproximación de la Superficie Senoidal.

Como ha quedado reflejado en la *Figura 5.10*, la superficie estimada dista mucho de la original. La inmovilidad de los representantes de las unidades radiales (situados en el espacio de entrada por un algoritmo de agrupamiento, tal como muestra la *Figura 5.11*), asociada con la rigidez impuesta al considerar un único ancho σ en toda la red, hacen necesario elevar el número de unidades radiales para poder cubrir el espacio de entrada de una forma más densa.

Con un número reducido de unidades radiales, como es el caso del ejemplo expuesto, la influencia del parámetro σ en el error final de estimación toma formas como la mostrada en la *Figura 5.9*. En ella aparece efectivamente una zona relativamente amplia de valores adecuados de σ, pero esta misma figura muestra también los efectos perjudiciales de un valor de σ demasiado pequeño.

Al ir aumentando el número de unidades radiales, el espacio de entrada va quedando muestreado por los representantes de forma cada vez más fina, de tal forma que la interpolación ponderada realizada a la hora de la estimación de un nuevo punto, va tomando en consideración puntos cada vez más próximos en el espacio de entrada, pudiendo de esta forma ir moldeando los detalles de la relación de entrada/salida.

Como conclusión final de la estructura GRNN podemos decir que esta estructura basa su capacidad de aproximación funcional en la densidad del recubrimiento del

espacio de entrada por parte de los representantes de las unidades radiales. Al no existir más que un ancho σ para todas las unidades, serán las regiones más "conflictivas" de la relación de entrada/salida las que impongan una cota superior a este parámetro, cota que resultará sobradamente exigente en regiones más "suaves" de la transformación, y que obligará a aumentar el número de unidades radiales.

Por otro lado, uno de los principales atractivos de esta estructura, como es la determinación en un sólo paso por el conjunto de datos de los principales parámetros de la red, se ve enturbiado por la determinación más o menos empírica del parámetro σ.

En los apartados siguientes dotaremos a esta estructura de una mayor flexibilidad, lo que permitirá un mejor aprovechamiento de sus unidades radiales, a costa siempre de tener que aplicar algoritmos de optimización para la determinación de sus parámetros.

5.4 Las estructuras PRBFN

Hemos visto al comienzo de este capítulo cómo la red RBFN que resulta de la teoría de la regularización era capaz de reconstruir la hipersuperficie de aproximación como una suma de funciones base radiales. El rasgo más característico de este tipo de aproximaciones es su caracter de localidad: las unidades radiales se distribuyen entre los vectores de entrada del conjunto de entrenamiento y se especializan en su zona de acción, interfiriendo muy levemente unas con otras. Esta clara delimitación de la zona de influencia de cada unidad radial facilita el proceso de aprendizaje (una determinada unidad radial sólo intervendrá en la aproximación de un subconjunto determinado de datos de entrenamiento), pero limita la capacidad de generalización del aproximador a las regiones del espacio de entrada que han quedado representadas en el conjunto de entrenamiento (las salidas de la red tenderán a anularse según nos alejemos de los centros de las unidades radiales). Este tipo de comportamiento es a mi modo de ver muy recomendable para tareas de reconocimiento de patrones, en las que se trata de delimitar las regiones del espacio de entrada correspondientes a cada patrón, dando al mismo tiempo por desconocido todo vector de entrada que no entre en una de estas regiones.

Para realizar tareas de aproximación funcional es en mi opinión más conveniente utilizar modelos que en lugar de reconstruir la hipersuperficie de aproximación como suma de funciones base radiales, realicen interpolaciones locales a partir de una selección de muestras de entrada/salida representativas de la relación funcional que se quiere aproximar. Esta es la forma de operar de las redes RBFN con conexiones laterales y de las redes GRNN, en las que la salida estimada se obtiene como una media ponderada de las salidas esperadas en cada unidad radial (pesos de las unidades de salida), utilizando como factores de ponderación una medida de la distancia del vector de entradas actual a los vectores centros de las unidades radiales correspondientes. De esta forma se mantienen las ventajas de la aproximación local, pero se mejoran las capacidades de generalización.

La principal ventaja de la red GRNN frente a la red RBFN con conexiones laterales es su directa interpretación en términos probabilistas[i], lo que permite realizar una buena inicialización de sus parámetros, extraer conocimiento de los modelos ajustados, y estimar funciones de densidad probabilista. Sus principales desventajas son sin embargo la rigidez de su estructura y la poca flexibilidad de sus leyes de aprendizaje, que obligan a utilizar un elevado número de unidades radiales para conseguir una precisión aceptable.

[i] Existe una analogía muy interesante entre la red GRNN y la red RBFN con conexiones laterales: si bien la red GRNN puede interpretarse en términos de probabilidad clásica, la red RBFN con conexiones laterales admite una interpretación análoga en términos de lógica borrosa (*[Jang & Sun, 1993]*, *[Wienholt, 1993]*).

Estas razones dieron lugar al desarrollo de dos variantes de la red GRNN originales de esta tesis: las estructuras RBFN probabilistas a las que notaremos estructuras PRBFN tipo I y tipo II. Estas variantes solucionan la rigidez de la estructura GRNN, aumentando su flexibilidad mediante la inclusión de nuevos parámetros que permiten amoldar los soportes de las unidades radiales a las características de los datos de entrenamiento. Este aumento de flexibilidad de las unidades radiales, que se traduce en un decrecimiento del número de unidades ocultas requerido, se ve acompañado por la propuesta de nuevas estrategias de aprendizaje necesarias para el ajuste de sus parámetros.

Las redes PRBFN, además de poder ser utilizadas como aproximadores funcionales para el modelado de procesos dinámicos según lo descrito en los Capítulos 2 y 3, pueden ser además utilizadas para estimar funciones de densidad probabilista. Esta capacidad jugará un papel fundamental en el sistema de diagnóstico propuesto en el Capítulo 6.

5.4.1 La red PRBFN tipo I

a) Definición

La estructura PRBFN tipo I resulta de asignar a cada unidad radial un ancho propio σ_i, de tal forma que si definimos el factor de escala:

$$\mu_i = \frac{1}{\sqrt{2}\cdot\sigma_i} > 0$$

Ecuación 5.26

podemos modificar la expresión del estimador de función de densidad (Ecuación 5.15) para obtener:

$$p(\mathbf{x}, y) = \frac{1}{\pi^{(n+1)/2}}\cdot\frac{1}{h}\sum_{i=1}^{h}\left[\mu_i^n\cdot exp\left(-\mu_i^2\cdot(\mathbf{x}-\mathbf{r_i})^T\cdot(\mathbf{x}-\mathbf{r_i})\right)\cdot\mu_i\cdot exp\left(-\mu_i^2\cdot(y-v_i)^2\right)\right]$$

$$= \frac{1}{\pi^{(n+1)/2}}\cdot\frac{1}{h}\sum_{i=1}^{h}\left[a_i\cdot\mu_i\cdot exp\left(-\mu_i^2\cdot(y-v_i)^2\right)\right]$$

Ecuación 5.27

donde ya ha sido modificada la antigua expresión de la activación de las unidades radiales (a_i) de la Ecuación 5.22 , convirtiéndose en:

$$a_i(\boldsymbol{x}) = \mu_i{}^n \cdot exp\left(-\mu_i^2 \cdot (\boldsymbol{x} - \boldsymbol{r_i})^T \cdot (\boldsymbol{x} - \boldsymbol{r_i})\right)$$

Ecuación 5.28

de tal forma que la nueva estimación de la salida deseada d, dado el vector de entradas $\boldsymbol{x}$ vendrá dada por:

$$y(\boldsymbol{x}) = E\left(\frac{D}{\boldsymbol{x}}\right) = \frac{\int_{-\infty}^{+\infty} z \cdot p(\boldsymbol{x}, z) \cdot dz}{\int_{-\infty}^{+\infty} p(\boldsymbol{x}, z) \cdot dz} = \frac{\frac{1}{\pi^{(n+1)/2}} \cdot \frac{1}{h} \cdot \sum_{i=1}^{h} \left[a_i \cdot \mu_i \cdot \int_{-\infty}^{+\infty} \left[z \cdot exp\left(-\mu_i^2 \cdot (z - v_i)^2\right)\right] \cdot dz \right]}{\frac{1}{\pi^{n/2}} \cdot \frac{1}{h} \cdot \sum_{i=1}^{h} a_i}$$

Ecuación 5.29

de donde resulta:

$$y(\boldsymbol{x}) = \frac{\sum_{i=1}^{h} a_i \cdot v_i}{\sum_{i=1}^{h} a_i}$$

Ecuación 5.30

al ser:

$$\int_{-\infty}^{+\infty} \left[z \cdot exp\left(-\mu_i^2 \cdot (z - v_i)^2\right)\right] \cdot dz = v_i \cdot \frac{\sqrt{\pi}}{\mu_i}$$

Ecuación 5.31

De igual forma se obtiene la estimación de la función de densidad del vector de entradas $\boldsymbol{x}$, dada por la expresión:

$$p(\boldsymbol{x}) = \frac{1}{\pi^{n/2}} \cdot \frac{1}{h} \cdot \sum_{i=1}^{h} a_i$$

Ecuación 5.32

b) Estructuración conexionista

El esquema de aproximación funcional y de funciones de densidad establecido por el conjunto de ecuaciones {Ecuación 5.27, Ecuación 5.28, Ecuación 5.30} puede ser estructurado bajo forma de red neuronal (a la que llamaremos red PRBFN tipo I), con cuatro capas de elementos de proceso:

1.- La capa de entrada, formada por n elementos de proceso lineales que realizan la estandarización de las n variables externas de entrada $\boldsymbol{x}_r$, produciendo las variables estandarizadas $\boldsymbol{x}$:

$$x_i = c_{i0} + c_{i1} x_{ri}$$

Ecuación 5.33

Los coeficientes c_{ij} de la estandarización se calculan a partir de los ejemplos de entrenamiento, como se vió en el Capítulo 2.

2.- La capa de unidades radiales, formada por h unidades de función de transferencia:

$$a_i(\boldsymbol{x}) = |\mu_i|^n \cdot exp\left(-\mu_i^2 \cdot (\boldsymbol{x} - \boldsymbol{r_i})^T \cdot (\boldsymbol{x} - \boldsymbol{r_i})\right)$$

Ecuación 5.34

donde cada unidad ha de almacenar su vector centro o representante $\boldsymbol{r}_i \in \Re^n$ que indica el centro del soporte de la unidad radial, y su factor de escala $\mu_i \in \Re$, que regula el radio del mencionado soporte. Se ha incluido el valor absoluto de los factores de escala para permitir valores negativos de estos parámetros, de tal forma que no haya que aplicar métodos de optimización con restricciones durante la etapa de aprendizaje.

3.- La capa de unidades sumatorias, donde se calculan los valores:

$$S_A = \sum_{i=1}^{h} v_i a_i$$

$$S_B = \sum_{i=1}^{h} a_i$$

Ecuación 5.35

Los pesos v_i están ligados a la esperanza de la salida en la región del espacio de entrada cubierta por la unidad radial i.

4.- La capa de salida, donde se calcula el valor estimador de la salida:

$$y = \frac{S_A}{S_B}$$

Ecuación 5.36

y el valor estimado de la función de densidad del vector de entradas:

$$p(\boldsymbol{x}) = \frac{1}{\pi^{n/2}} \frac{1}{h} S_B$$

Ecuación 5.37

La Figura 5.12 muestra la estructura resultante:

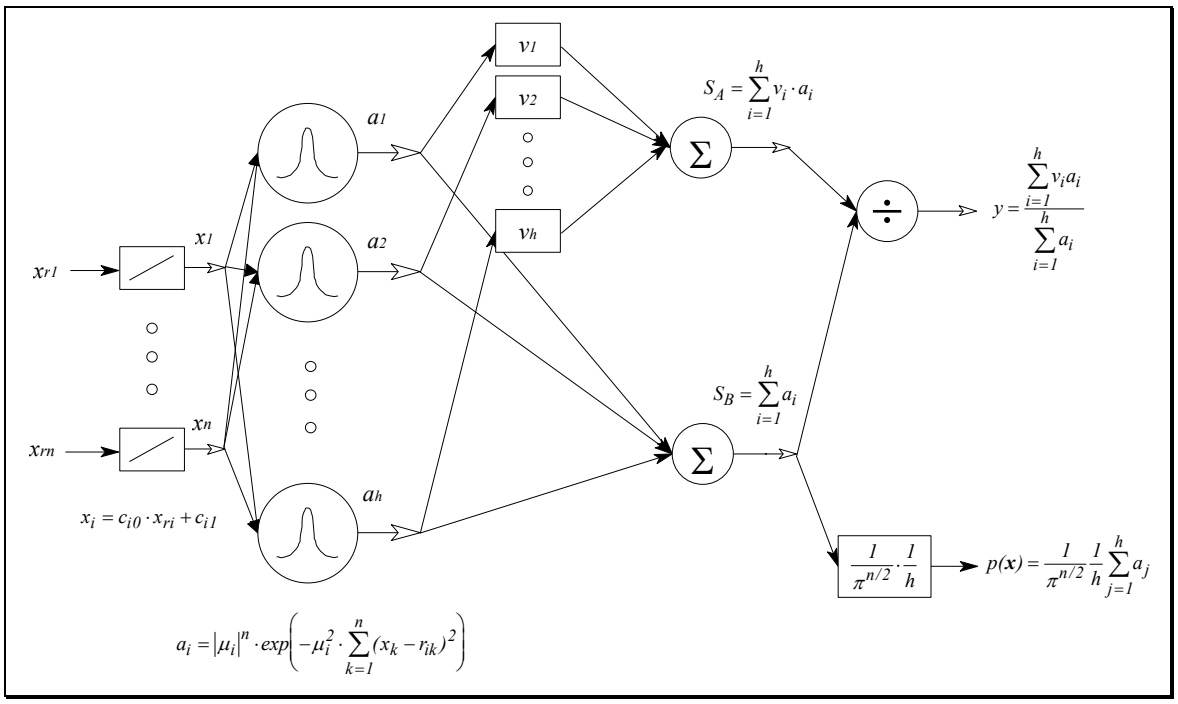

Figura 5.12: Estructura de la red PRBFN tipo I

c) Cálculo de derivadas

Para ajustarnos al esquema de ajuste de aproximadores funcionales presentado en el Capítulo 2, hemos de disponer de las derivadas de las salidas de la red respecto de sus parámetros y de sus entradas.

Cálculo de las derivadas de la salida estimada respecto de los parámetros:

Las derivadas de la salida estimada *y* respecto de los tres tipos de parámetros de la red PRBFN tipo I vienen dadas por:

$$\begin{cases} \dfrac{\partial y}{\partial r_{ik}} = \dfrac{\partial y}{\partial a_i}\dfrac{\partial a_i}{\partial r_{ik}} = \dfrac{v_i - y}{\sum_{j=1}^{h} a_j} 2a_i \mu_i^2 (x_k - r_{ik}) \\ \dfrac{\partial y}{\partial \mu_i} = \dfrac{\partial y}{\partial a_i}\dfrac{\partial a_i}{\partial \mu_i} = \dfrac{v_i - y}{\sum_{j=1}^{h} a_j}\left(n|\mu_i|^{n-1} sgn(\mu_i) exp\left(-\mu_i^2 \sum_{k=1}^{n} (x_k - r_{ik})^2 \right) - 2a_i \mu_i \sum_{k=1}^{n} (x_k - r_{ik})^2 \right) \\ \quad = \dfrac{v_i - y}{\sum_{j=1}^{h} a_j}\dfrac{a_i}{\mu_i}\left(n + 2(-\mu_i^2 \sum_{k=1}^{n} (x_k - r_{ik})^2) \right) \quad si\ \mu_i \neq 0 \\ \dfrac{\partial y}{\partial v_i} = \dfrac{a_i}{\sum_{j=1}^{h} a_j} \end{cases}$$

Ecuación 5.38

donde se ha utilizado la función signo definida por:

$$sgn(x) = \begin{cases} 1 & si \quad x > 0 \\ 0 & si \quad x = 0 \\ -1 & si \quad x < 0 \end{cases}$$

Ecuación 5.39

Cálculo de las derivadas de la salida estimada respecto de las entradas:

Las derivadas de la salida estimada *y* respecto de las entradas estandarizadas vienen dadas por:

$$\frac{\partial y}{\partial x_k} = \sum_{i=1}^{h} \frac{\partial y}{\partial a_i} \frac{\partial a_i}{\partial x_k} = 2 \sum_{i=1}^{h} \frac{a_i}{\sum_{j=1}^{h} a_j} \mu_i^2 (x_k - r_{ik})(y - v_i)$$

Ecuación 5.40

Si suponemos que r_{jk} es una medida del valor esperado de x_k en el soporte de la unidad nºj, y que v_j es una medida del valor esperado de la salida y en la misma región del espacio de entrada, podemos interpretar la esperanza de la derivada de la salida respecto de la entrada dada por la ecuación anterior como una medida de correlación no lineal entre las variables x_k e y. Esta medida de correlación se obtiene como una media de las correlaciones locales en cada unidad, ponderada por la activación normalizada de cada una de ellas Esta consideración está directamente ligada al Análisis Estadístico de Sensibilidades presentado en el Capítulo 2.

5.4.2 La red PRBFN tipo II

a) Definición

La estructura PRBFN tipo II resulta de asignar a cada unidad radial, y en cada dimensión, de un factor de escala $\mu_{ik}>0$ propio, de tal forma que podemos modificar la expresión del estimador de función de densidad (Ecuación 5.15) para obtener:

$$p(\mathbf{x}, y) = \frac{1}{\pi^{(n+1)/2}} \cdot \frac{1}{h} \sum_{i=1}^{h} \left[\left(\prod_{k=1}^{n} \mu_{ik} \right) \cdot exp\left(-\sum_{k=1}^{n} \mu_{ik}^2 \cdot (x_k - r_{ik})^2 \right) \cdot \mu_{io} \cdot exp\left(-\mu_{io}^2 \cdot (y - v_i)^2 \right) \right]$$

$$= \frac{1}{\pi^{(n+1)/2}} \cdot \frac{1}{h} \sum_{i=1}^{h} \left[a_i \cdot \mu_{io} \cdot exp\left(-\mu_{io}^2 \cdot (y - v_i)^2 \right) \right]$$

Ecuación 5.41

donde ya ha sido modificada la antigua expresión de la activación de las unidades radiales (a_i) de la Ecuación 5.22, convirtiéndose en:

$$a_i(\boldsymbol{x}) = \left(\prod_{k=1}^{n} \mu_{ik} \right) \cdot exp\left(-\sum_{k=1}^{n} \mu_{ik}^2 \cdot (x_k - r_{ik}) \right)$$

Ecuación 5.42

de tal forma que la nueva estimación de la salida deseada d, dado el vector de entradas $\boldsymbol{x}$ ($y(\boldsymbol{x})$) vendrá dada, como en el caso anterior, por:

$$y(\boldsymbol{x}) = \frac{\sum_{i=1}^{h} a_i \cdot v_i}{\sum_{i=1}^{h} a_i}$$

Ecuación 5.43

De igual forma se obtiene la estimación de la función de densidad del vector de entradas $\boldsymbol{x}$, dada por la expresión:

$$p(\boldsymbol{x}) = \frac{1}{\pi^{n/2}} \cdot \frac{1}{h} \cdot \sum_{i=1}^{h} a_i$$

Ecuación 5.44

b) Estructuración conexionista

La estructuración conexionista del aproximador PRBFN tipo II es idéntica a la red PRBFN tipo I, salvo en la expresión de la función de transferencia de las unidades radiales ($a_i(\boldsymbol{x})$). De esta forma obtenemos la red PRBFN tipo II formada por cuatro capas de elementos de proceso:

1.- La capa de entrada, formada por n elementos de proceso lineales que realizan la estandarización de las n variables externas de entrada $\boldsymbol{x}_r$, produciendo las variables estandarizadas $\boldsymbol{x}$:

$$x_i = c_{i0} + c_{i1} x_{ri}$$

Ecuación 5.45

Los coeficientes c_{ij} de la estandarización se calculan a partir de los ejemplos de entrenamiento, como se vió en el Capítulo 2.

2.- La capa de unidades radiales[i], formada por h unidades de función de transferencia:

$$a_i(\boldsymbol{x}) = \left(\prod_{k=1}^{n} |\mu_{ik}| \right) \cdot exp\left(-\sum_{k=1}^{n} \mu_{ik}^2 \cdot (x - r_{ik})^2 \right)$$

Ecuación 5.46

donde cada unidad ha de almacenar su vector centro o representante $\boldsymbol{r}_i \in \Re^n$ que indica el centro del soporte de la unidad radial, y su vector de factores de escala $\boldsymbol{\mu}_i \in \Re^n$, que regula el radio del mencionado soporte. Se ha incluido el valor absoluto de los factores de escala para permitir valores negativos de estos parámetros, de tal forma que no haya que aplicar métodos de optimización con restricciones durante la etapa de aprendizaje.

3.- La capa de unidades sumatorias, donde se calculan los valores:

$$S_A = \sum_{i=1}^{h} v_i a_i$$

$$S_B = \sum_{i=1}^{h} a_i$$

Ecuación 5.47

Los pesos v_i están ligados a la esperanza de la salida en la región del espacio de entrada cubierta por la unidad radial nºi.

4.- La capa de salida, donde se calcula el valor estimador de la salida:

$$y = \frac{S_A}{S_B}$$

Ecuación 5.48

y el valor estimado de la función de densidad del vector de entradas:

[i] A pesar de que las unidades ocultas de la red PRBFN tipo II no llevan a cabo una transformación radial (no son simétricas en las distintas dimensiones del espacio de entrada), seguiremos llamando a estas unidades por el nombre de unidades radiales.

$$p(\boldsymbol{x}) = \frac{1}{\pi^{n/2}} \frac{1}{h} S_B$$

Ecuación 5.49

La Figura 5.12 muestra la estructura resultante:

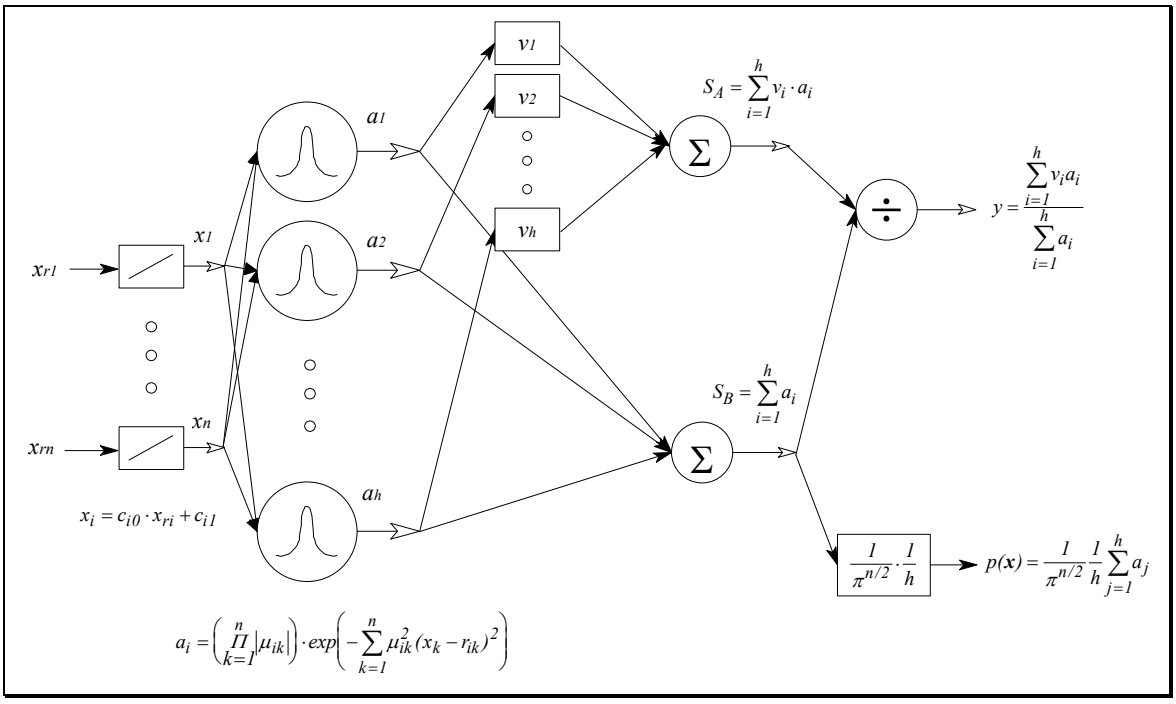

Figura 5.13: Estructura de la red PRBFN tipo I

c) Cálculo de derivadas

Cálculo de las derivadas de la salida estimada respecto de los parámetros:

Las derivadas de la salida estimada *y* respecto de los tres tipos de parámetros de la red PRBFN tipo II vienen dadas por:

$$\begin{cases} \dfrac{\partial y}{\partial r_{ik}} = \dfrac{\partial y}{\partial a_i}\dfrac{\partial a_i}{\partial r_{ik}} = \dfrac{v_i - y}{\sum_{j=1}^{h} a_j} 2a_i \mu_{ik}^2 (x_k - r_{ik}) \\ \dfrac{\partial y}{\partial \mu_{ik}} = \dfrac{\partial y}{\partial a_i}\dfrac{\partial a_i}{\partial \mu_{ik}} = \dfrac{v_i - y}{\sum_{j=1}^{h} a_j} \left(\left(\prod_{\substack{l=1 \\ l \neq k}}^{n} |\mu_{il}| \right) sgn(\mu_{ik}) exp\left(-\sum_{k=1}^{n} \mu_{ik}^2 (x_k - r_{ik})^2 \right) - 2a_i \mu_{ik} (x_k - r_{ik})^2 \right) \\ \quad = \dfrac{v_i - y}{\sum_{j=1}^{h} a_j} a_i \left(\dfrac{1}{\mu_{ik}} - 2\mu_{ik} (x_k - r_{ik})^2 \right) \qquad si \quad \mu_{ik} \neq 0 \\ \dfrac{\partial y}{\partial v_i} = \dfrac{a_i}{\sum_{j=1}^{h} a_j} \end{cases}$$

Ecuación 5.50

Cálculo de las derivadas de la salida estimada respecto de las entradas:

Las derivadas de la salida estimada *y* respecto de las entradas estandarizadas vienen dadas por:

$$\frac{\partial y}{\partial x_k} = \sum_{i=1}^{h} \frac{\partial y}{\partial a_i}\frac{\partial a_i}{\partial x_k} = 2\sum_{i=1}^{h} \frac{a_i}{\sum_{j=1}^{h} a_j} \mu_i^2 (x_k - r_{ik})(y - v_i)$$

Ecuación 5.51

5.4.3 Aprendizaje

Las estructuras PRBFN pueden ser utilizadas de dos formas diferentes: como aproximadores funcionales, y/o como estimadores de funciones de densidad probabilista. La diferencia radica en la definición de la función de error utilizada para el ajuste de los parámetros. Si esta función de error sólo contempla errores de estimación (como es el caso del error cuadrático medio definido en el Capítulo 2), los pesos de la red serán ajustados sin tener en cuenta la bondad de la aproximación de la fdp de los datos de entrada (*p(x)*), asegurando tan sólo que la fdp conjunta estimada induce una esperanza condicionada *E(y/x)* que aproxima de forma correcta los datos de entrenamiento.

Para que la fdp estimada (*p(x)*) represente la distribución de los vectores de entrada del conjunto de entrenamiento, es necesario incluir en la función de coste un término de tal forma que se maximice la verosimilitud logarítmica de las muestras de entrenamiento (*[Hasselblad, 1966], [Traven, 1991], [Hernoth & Clark, 1995]*).

a) Aproximación funcional con redes PRBFN

Si la red PRBFN va a ser utilizada como aproximador funcional, sin tener en cuenta su capacidad de aproximación de la fdp del vector de entradas, podemos encajar esta estructura en el esquema general de aproximación funcional definido en el Capítulo 2, controlando la complejidad del modelo mediante el número de unidades radiales.

De esta forma, para cada estructura propuesta y después de haber inicializado sus parámetros libres (pesos), aplicaremos un algoritmo de optimización para minimizar el error cuadrático de entrenamiento (R_{entr}), al mismo tiempo que evaluamos la capacidad de generalización del modelo resultante mediante la evaluación del error cuadrático de test (R_{test}).

Veamos a continuación cómo pueden inicializarse de una forma efectiva los pesos de las estructuras PRBFN:

Inicialización de los pesos

El vector de pesos de las redes PRBFN consta de tres tipos de elementos: los vectores centro de las unidades radiales ($\boldsymbol{r}_i$), los factores de escala (μ_i ó μ_{ik}), y los pesos de las unidades sumatorias (v_i).

- Inicialización de los centros de las unidades radiales:

La inicialización de los vectores centros de las unidades radiales se realiza mediante la aplicación de un algoritmo de agrupamiento ("clustering") sobre los datos de entrada del conjunto de entrenamiento estandarizado ($\{\boldsymbol{x}[i] \in \Re^n,\ i=1,...,N\}$). El objetivo de este procedimiento es seleccionar un conjunto de *h* representantes de los datos de entrada, de tal forma que las unidades radiales cubran la región del espacio de entrada que contiene datos de entrenamiento.

Los algoritmos de agrupamiento del tipo "K-means" y de cuantización vectorial (*[Hartigan, 1975], [Kohonen, 1988]*) tratan de minimizar el error de reconstrucción definido como la suma de las distancias de los datos de entrenamiento a su representante más cercano. Los representantes seleccionados por estos métodos quedarán localizados en as regiones del espacio de entrada que contienen más datos, pudiendo despreciar regiones poco representadas.

Frente a un caso real de modelado de un proceso físico con distintos puntos de operación, es de esperar que el conjunto de entrenamiento contenga muchos datos en cada uno de estos puntos, y comparativamente pocos en las regiones de paso de uno a otro. Estos datos que describen el paso de un punto de operación a otro contienen sin embargo una información muy valiosa, que podría ser despreciada al aplicar algoritmos de agrupamiento como los mencionados anteriormente.

El algoritmo de agrupamiento propuesto en esta tesis tiene como objetivo crear la representación más homogénea posible de los datos, prestando una especial atención a la diversidad de los representantes. Por ello se ha optado por un algoritmo de agrupamiento en dos pasos, al que llamaremos "Algoritmo Mixto de Agrupamiento" (AMA), en el que en el primer paso se escogen los *h* representantes más distantes entre sí (aplicando el algoritmo "leader" *[Hartigan, 1975]*), y en el segundo se centran estos representantes en los centros de gravedad de sus regiones de influencia (mediante la aplicación del algoritmo "K-means").

El algoritmo AMA consta pues de los siguientes pasos:

Etapa I: algoritmo "leader"

> Paso 1.- Se inicializa el primer representante tomando aleatoriamente un dato de entrada:
>
> $$\boldsymbol{r}_1 = \boldsymbol{x}[j]$$
>
> *Ecuación 5.52*
>
> con *j* entero aleatorio del intervalo *[1; N]*
> Paso 2.- Para *i=2,...,h*, se toma como vector representante *i* el dato de entrada más lejano a los representantes ya seleccionados:

$$\boldsymbol{r}_i = \boldsymbol{x}[k_i]$$

Ecuación 5.53

siendo:

$$k_i = \underset{j=1,\ldots,N}{arg\,max}\left(\min_{l=1,\ldots,i-1}\left(\left\|\boldsymbol{x}[j]-\boldsymbol{r}_l\right\|\right)\right)$$

Ecuación 5.54

Etapa II: algoritmo "K-means"

Paso 3.- Se determina la zona de influencia de cada uno de los representantes, como el conjunto de datos de entrada de entrenamiento tales que el representante más cercano a cada uno de ellos es el representante en cuestión:

$$Z_i = \left\{\boldsymbol{x}\left[k_{i1}\right], \boldsymbol{x}\left[k_{i2}\right], \ldots, \boldsymbol{x}\left[k_{in_i}\right]\right\}$$

Ecuación 5.55

cumpliéndose:

$$i = \underset{l=1,\ldots,h}{arg\,min}\left(\left(\left\|\boldsymbol{x}\left[k_{ij}\right]-\boldsymbol{r}_l\right\|\right)\right) \quad \forall j = 1,\ldots,n_i$$

Ecuación 5.56

y se localizan los representantes en el centro de gravedad de su zona de influencia:

$$\boldsymbol{r}_i = \frac{1}{n_i}\sum_{j=1}^{n_i}\boldsymbol{x}\left[k_{ij}\right]$$

Ecuación 5.57

Este paso se repite un número predeterminado de veces (tomaremos 10 repeticiones en los ejemplos presentados), o hasta que ningún representante varía en más de una cota predeterminada.

Para ilustrar el funcionamiento del algoritmo AMA, tomemos como ejemplo la selección de 7 representantes de una población bidimensional ($n=2$) que ha sido generada tomando 100 muestras según una distribución normal de media (0.25,0.25)

y matriz diagonal de covarianzas $0.01\mathbf{I}$, y 100 muestras según una distribución normal de media (0.75,0.75) y misma matriz diagonal de covarianzas $0.01\mathbf{I}$.

La Figura 5.14 muestra la localización de las muestras y los representantes obtenidos tras la primera y segunda etapa de aplicación del AMA:

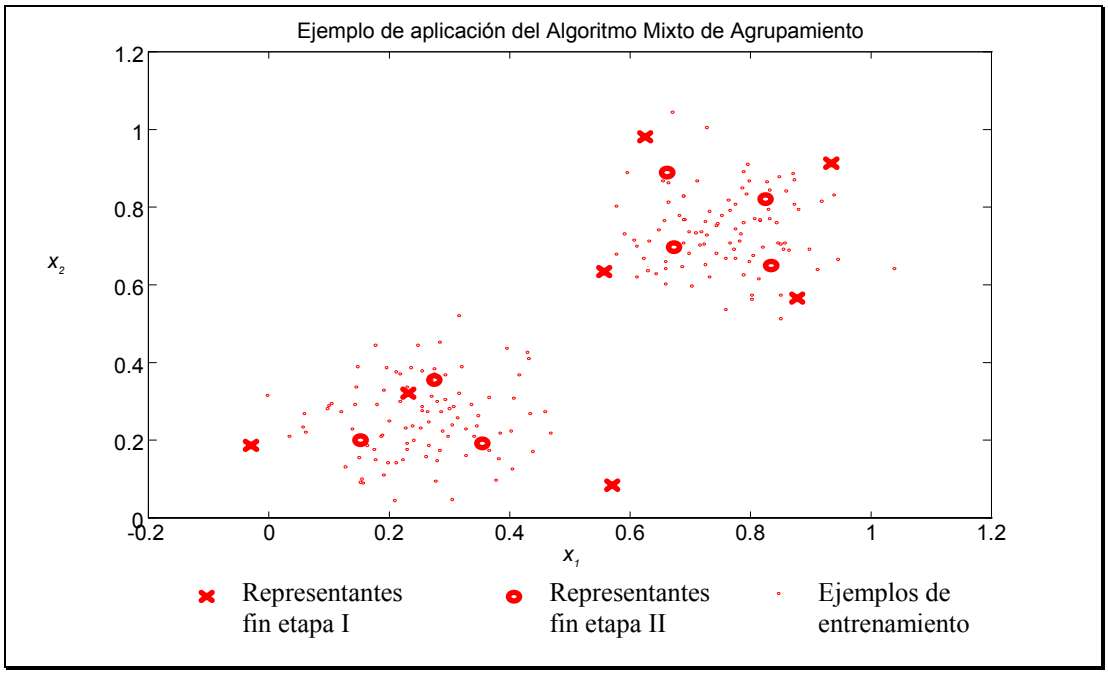

Figura 5.14: Ejemplo de aplicación del Algoritmo Mixto de Agrupamiento (AMA)

Como puede apreciarse en el ejemplo anterior, los representantes obtenidos tras la primera etapa del AMA se han centrado en muestras muy distantes entre sí. La segunda etapa del AMA realiza una especie de filtrado paso bajo de la localización de los representantes, situándolos en los centros de gravedad de sus zonas de influencia.

- Inicialización de los factores de escala de las unidades radiales:

Los factores de escala de la red PRBFN tipo I (μ_i) se obtienen con el heurístico "p-nearest-neighbor". En nuestro caso tomaremos $p=2$, de tal forma que si d_{i1} y d_{i2} son las distancias del representante $\boldsymbol{r}_i$ a sus dos representantes más cercanos, resulta:

$$\mu_i = \frac{1}{\sqrt{2}\,\frac{d_{i1}+d_{i2}}{2}} = \frac{\sqrt{2}}{d_{i1}+d_{i2}}$$

Ecuación 5.58

Los fatores de escala de la red PRBFN tipo II se inicializan como los de la red PRBFN tipo I, sin más que igualar todos los de una misma unidad ($\mu_{i1}=\mu_{i2}=...=\mu_{in}$).

Esta inicialización asegura un ligero solapamiento de los soportes de las unidades radiales, cubriendo aquella región del espacio de entrada donde existen suficientes datos de entrenamiento.

- Inicialización de los pesos de las unidades sumatorias:

Los pesos de las unidades sumatorias (ν_i) están ligados con el valor esperado de la salida en el soporte de la unidad radial i. Por esta razón inicializaremos estos parámetros igualándolos al valor deseado de la salida correspondiente al vector de entradas más próximo al representante de cada unidad radial:

$$\nu_i = d[k_i]$$

Ecuación 5.59

siendo:

$$k_i = \underset{k=1,\ldots,N}{arg\,min}\left(\left(\|\boldsymbol{x}[k]-\boldsymbol{r}_i\|\right)\right)$$

Ecuación 5.60

Una inicialización más robusta podría conseguirse realizando una media ponderada de las salidas deseadas en la zona de influencia de cada unidad radial.

b) Estimación de funciones de densidad con redes PRBFN

Cuando la red PRBFN va a ser utilizada para estimar la fdp de una población muestral de la forma $\{\boldsymbol{x}[i]\in\Re^n,\ i=1,...,N\}$, mediante la estimación dada por la

Ecuación 5.37, los centros y factores de escala de las unidades radiales pueden ajustarse maximizando la verosimilitud logarítmica ("log-likelihood") (*[Hasselblad, 1966], [Traven, 1991], [Hernoth & Clark, 1995]*):

$$V = \frac{1}{N}\sum_{i=1}^{N} log\big(p(\mathbf{x}[i])\big)$$

Ecuación 5.61

En este caso sólo se utiliza parte de la estructura de la red, no interviniendo en la estimación de la fdp los pesos de las unidades sumatorias.

La maximización de V, o minimización de $-V$, puede realizarse aplicando un algoritmo de optimización basado en el gradiente, como los descritos en el Capítulo 2. Para ello es necesario calcular las derivadas de V con respecto a cada uno de los parámetros a optimizar, lo que es inmediato al considerar:

$$\frac{\partial V}{\partial w} = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{p(\mathbf{x}[i])} \frac{\partial p(\mathbf{x}[i])}{\partial w}$$

Ecuación 5.62

representando en este caso w cada uno de los parámetros libres de la red.

La inicialización de los centros y factores de escala de las unidades radiales puede realizarse del mismo modo que en el caso de aproximación funcional, aunque en este caso concreto podrían utilizarse directamente algoritmos clásicos de agrupamiento, como el "K-means" o la cuantización vectorial.

Ejemplo nº1

Tomemos como primer ejemplo la estimación de la función de densidad de una población unidimensional uniformemente distribuida en el intervalo *[-1,1]*. Para ello consideremos el conjunto de muestras generado según:

$$x[i]=-1+0.01[i-1] \quad \text{con} \quad i=1,...,201$$

Ecuación 5.63

y ajustemos tres redes PRBFN tipo I con distinto número de unidades ocultas, para estimar la fdp de x. Los pasos seguidos en este caso durante el ajuste de cada uno de los estimadores han sido:

1. Inicialización de los centros de las unidades mediante la aplicación del algoritmo de agrupamiento "*K*-means".

2. Inicialización de los factores de escala de las unidades radiales mediante el heurístico "*p*-nearest-neighbor" con $p=2$.

3. Ajuste final de los centros y factores de escala de las unidades radiales mediante la maximización de la verosimilitud logarítmica del conjunto de entrenamiento, aplicando el algoritmo de optimización LMQN1 descrito en el Capítulo 2.

La figura siguiente muestra los resultados obtenidos con 4, 6 y 8 unidades ocultas:

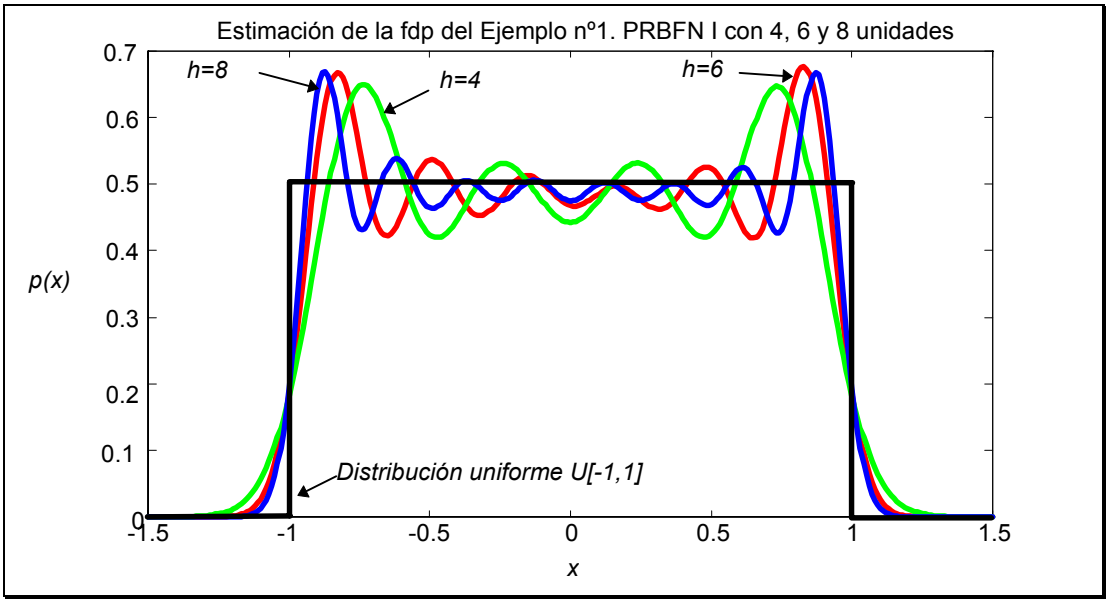

Figura 5.15: Estimación de la fdp del Ejemplo nº1

Las estimaciones obtenidas muestran cómo se ha reconstruido la distribución uniforme de la población original (igual a 0.5 para $x \in [-1,1]$ y 0 fuera de este intervalo) como una suma de h funciones gaussianas.

Ejemplo nº2

Como segundo ejemplo generemos un conjunto de 200 muestras según:

$$z[i] = z[i-1] + 1 / i \qquad \text{con } z(0)=0$$

$$x[i] = 2\, z[i] / max(z) - 1$$

y entrenemos distintas estructuras PRBFN tipo I (variando el número de unidades radiales) para estimar la fdp de la variable *x*.

El ajuste de las distintas estructuras ensayadas ha sido realizado siguiendo los mismos pasos que en el ejemplo anterior.

Los resultados obtenidos se muestran en la Figura 5.16, donde puede verse nuevamente cómo la fdp muestral (obtenida a partir del histograma de las muestras) ha sido reconstruida como una suma de funciones gausianas.

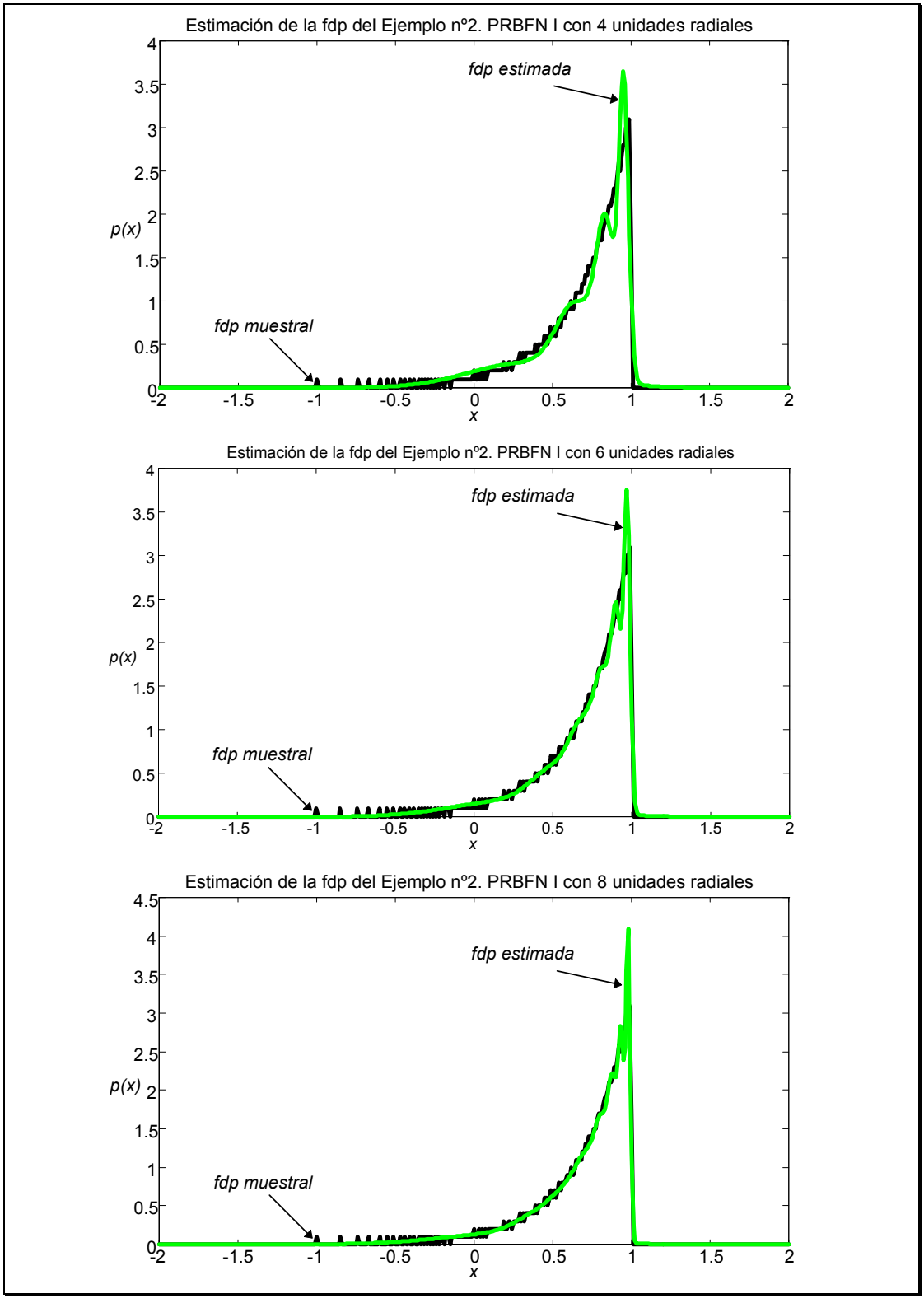

Figura 5.16: Estimación de la fdp del Ejemplo nº2

c) Aproximación funcional y estimación de la fdp del vector de entradas con una red PRBFN

En muchos problemas de aproximación funcional interesa tener una estimación de la fdp de las muestras del vector de entradas que han sido utilizadas para el ajuste del aproximador. Esta información permite detectar en fase de operación la presentación de casos a la entrada no representados en el conjunto de entrenamiento, para los cuales se desconoce el funcionamiento del aproximador funcional.

En el caso concreto de la detección de anomalías mediante modelos de funcionamiento normal, se supone que todos los modos de funcionamiento normal han quedado descritos o representados en el conjunto de entrenamiento, por lo que la detección de un vector de entradas desconocido (que da lugar a una fdp significativamente reducida) podrá interpretarse como una situación desconocida, posiblemente producida por un fallo en un elemento anterior del proceso.

Para obtener ambas estimaciones (la salida estimada del proceso y la estimación de la fdp del vector de entradas del modelo), caben dos posibilidades. La primera de ellas es utilizar dos redes PRBFN, una entrenada utilizando la función de error cuadrático medio de estimación como función de error a minimizar, y otra utilizando el criterio de máxima verosimilitud logarítmica. La segunda posibilidad es utilizar una única estructura PRBFN para realizar las dos estimaciones. En este caso la función de error a minimizar durante el ajuste del modelo ha de contemplar los dos criterios de optimalidad de forma simultánea: minimizar el error de estimación de la salida del proceso, y maximizar la función de verosimilitud logarítmica de los vectores de entrada del conjunto de entrenamiento.

La solución más sencilla a este problema sería ajustar los centros y factores de escala de las unidades radiales maximizando la función de verosimilitud logarítmica, y una vez resuelta la estimación de la fdp del vector de entradas, ajustar los pesos de las unidades sumatorias minimizando el error cuadrático de estimación. Esto es posble gracias a que los pesos de las unidades sumatorias no intervienen en la expresión de la función de verosimilitud logarítmica, por lo que el problema global puede ser descompuesto en dos subproblemas. Esta solución aparece ya en *[Traven, 1991]*, donde se asegura que una buena estimación de la salida deseada (que minimice la función de riesgo introducida en el Capítulo 2) requiere una buena estimación de la fdp del vector de entradas. Sin embargo la experiencia adquirida en los trabajos realizados para esta tesis indica que la afirmación anterior no es correcta, ya que al no ajustar los parámetros de las unidades radiales utilizando como criterio de optimalidad la precisión de la aproximación, se pierde capacidad de aproximación (como se verá en ejemplos posteriores). La ventaja de esta estrategia es que los pesos de las unidades sumatorias pueden ser calculados mediante regresión lineal, una vez que han sido fijados los parámetros de las unidades radiales.

Como alternativa a la estrategia anterior cabe la posibilidad de contemplar de forma simultánea estos dos criterios utilizando como función de error a minimizar una suma ponderada de los dos términos:

$$RV = \frac{1}{N}\left(\sum_{i=1}^{N}\left(d[i] - y(\boldsymbol{x}[i])\right)^2\right) - \lambda \frac{1}{N}\sum_{i=1}^{N} log\left(p(\boldsymbol{x}[i])\right)$$

Ecuación 5.64

siendo λ el factor de ponderación que controla el compromiso entre los dos criterios.

Bajo esta perspectiva, los pesos de las unidades sumatorias se ajustarían para minimizar el primer término de *RV* (ya que no intervienen en el segundo), mientras que los parámetros de las unidades radiales se ajustarían teniendo en cuenta los dos criterios.

Un problema importante de este método es la determinación del factor de ponderación λ. Una posible solución es entrenar la red PRBFN utilizando como función de error el error cuadrático medio (*R*), evaluar entonces los dos términos de la Ecuación 5.64, y reajustar los pesos de la red utilizando *RV* como función de error a minimizar con un valor de λ tal que el segundo término de *RV* represente un porcentaje predeterminado del primero.

Los siguientes ejemplos permitirán evaluar el efecto de la inclusión del término de verosimilitud logarítmica en la función de error a minimizar.

Ejemplo nº1:

Se trata de aproximar la función:

$$d(x) = 2x^3 - x$$

Ecuación 5.65

El primer conjunto de entrenamiento ha sido generado según la Ecuación 5.65, añadiendo un ruido blanco gaussiano de media nula y desviación típica 0.03, con *x* uniformemente distribuida en el intervalo *[-1;1]*.

Con este conjunto de entrenamiento se han entrenado distintas estructuras PRBFN con 4, 6 y 8 unidades radiales, minimizando la función de coste:

$$RV(\mathbf{w}) = \frac{1}{N}\left(\sum_{i=1}^{N}(d[i] - y(x[i])^2 - \lambda\sum_{i=1}^{N} log(p(x[i]))\right)$$

Ecuación 5.66

donde el primer término corresponde al error de estimación de la salida deseada y el segundo a la estimación de máxima verosimilitud logarítmica de la fdp de las entradas. El objetivo de este experimento es estudiar el efecto del segundo término de la función de coste sobre la estimación de $p(\mathbf{x})$. Para ello entrenaremos las mismas estructuras PRBFN utilizando los valores $\lambda=0.0$ y $\lambda=0.1$ Los resultados obtenidos se muestran a continuación:

1.- Error de estimación de la salida deseada:

Como puede apreciarse en la Tabla 5.1, las estructuras entrenadas con $\lambda=0$ (sin tener en cuenta el término de verosimilitud logarítmica) obtienen de forma sistemática una aproximación más precisa de la salida:

	std(d-y) $\lambda=0.0$	***std(d-y)*** $\lambda=0.1$
h=4	0.0296	0.0311
h=6	0.0294	0.0306
h=8	0.0285	0.0292

Tabla 5.1: Desviación típica de los errores de estimación

Esta pérdida de precisión es sin embargo poco significativa en este caso concreto, en el que las muestras cubren de forma uniforme la región del espacio de entrada representada en el conjunto de entrenamiento (ver Figura 5.17).

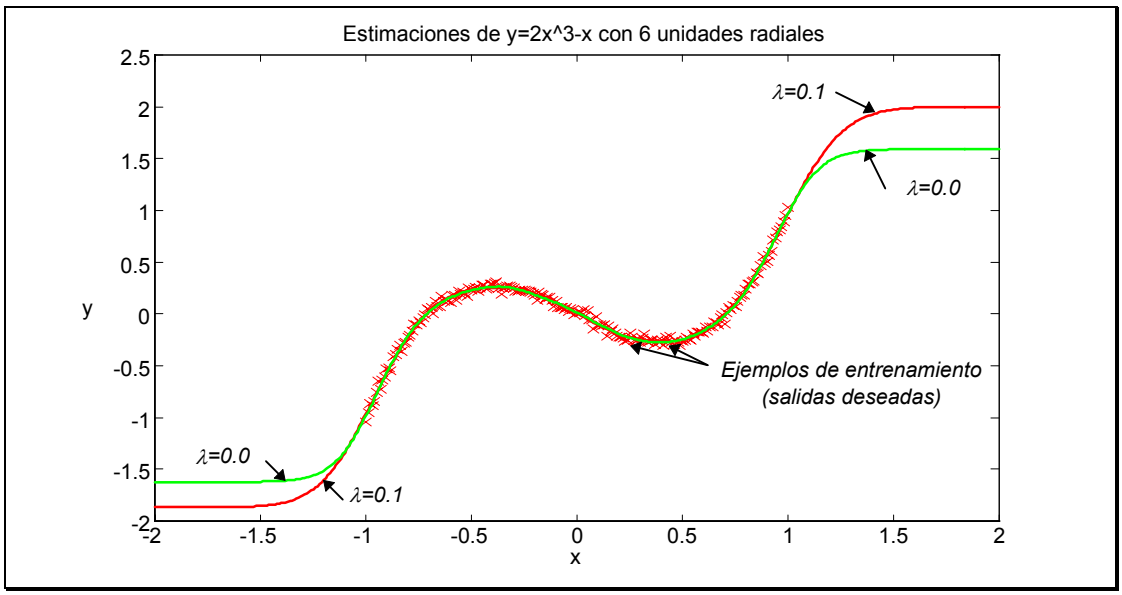

Figura 5.17: Estimaciones de la salida deseada obtenidas con 6 unidades radiales, con λ=0.0 y λ=0.1

2.- Estimaciones de la fdp de la variable de entrada: *p(x)*

Como puede apreciarse en la Figura 5.18, las estimaciones de la fdp de la variable de entrada (que debiera ser $p(x)=0.5$ para $x \in [-1;1]$ y nula en el resto de la recta real) realizadas por la red PRBFN sin tener en cuenta el término de máxima verosimilitud logarítmica, están muy desvirtuadas. Sin embargo al incluir el término de máxima verosimilitud se obtiene una buena estimación de *p(x)*, sin dañar apreciablemente la estimación de la salida deseada.

Hay que señalar que las estimaciones de la fdp de *x* difieren de las presentadas en el apartado b)) a causa de la inclusión del primer término en la función de error. Las unidades se han distribuido por parejas para poder dar forma a la relación de entrada/salida, dando lugar a estimaciones *p(x)* que presentan un número de "jorobas" igual a la mitad del número de unidades radiales.

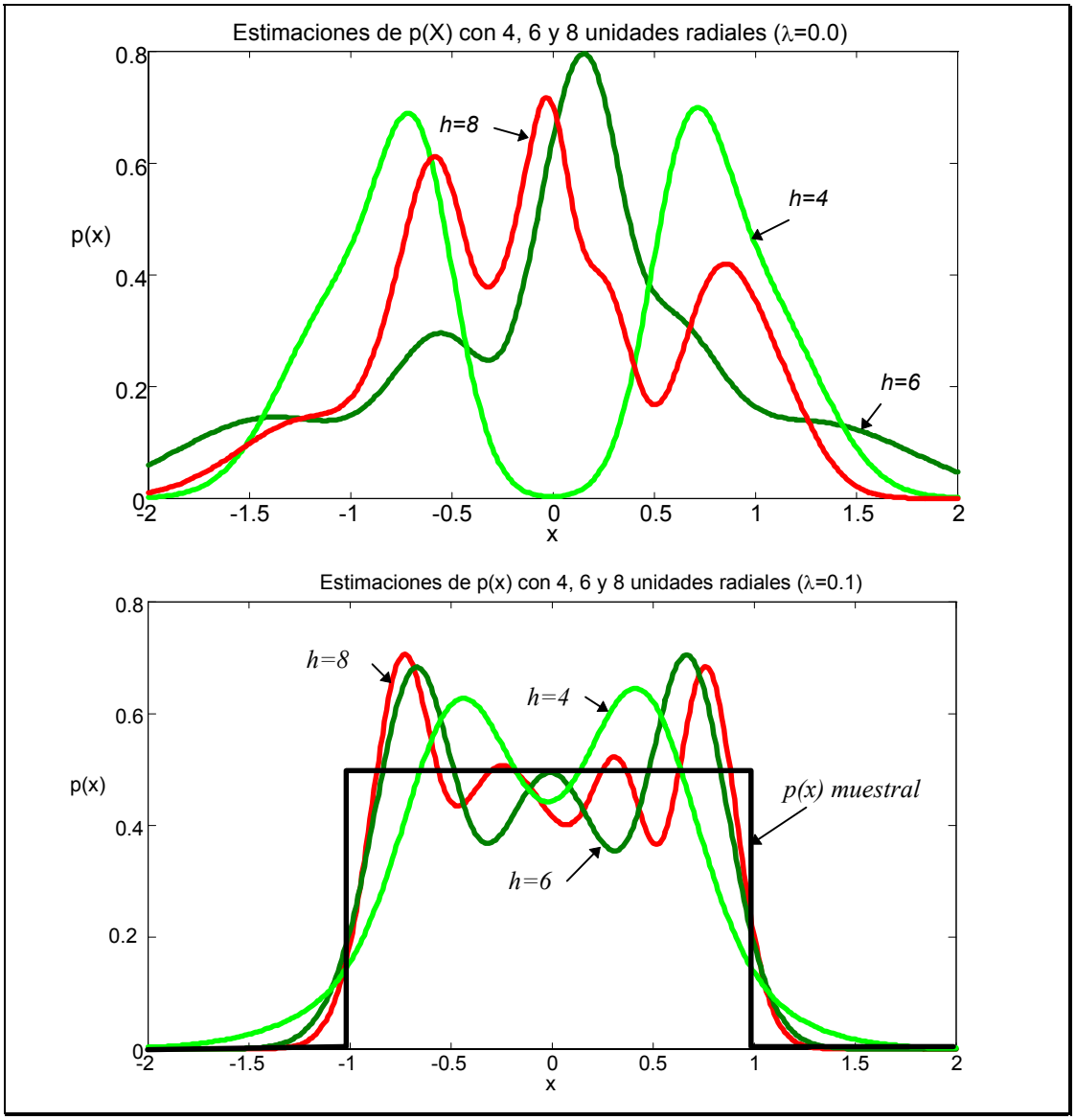

Figura 5.18: Estimación de p(x) con PRBFN de 4, 6 y 8 unidades radiales entrenadas con λ=0.1 y λ=0.0

Ejemplo nº2

Repitamos ahora el mismo experimento pero tomando ahora las muestras de la variable de entrada *x* de la forma:

$$z[i] = z[i-1] + 1 / i \qquad \text{con } z(0)=0$$
$$x[i] = 2\, z[i] / \text{máx}(z) - 1$$

Ecuación 5.67

Los resultados obtenidos se muestran a continuación:

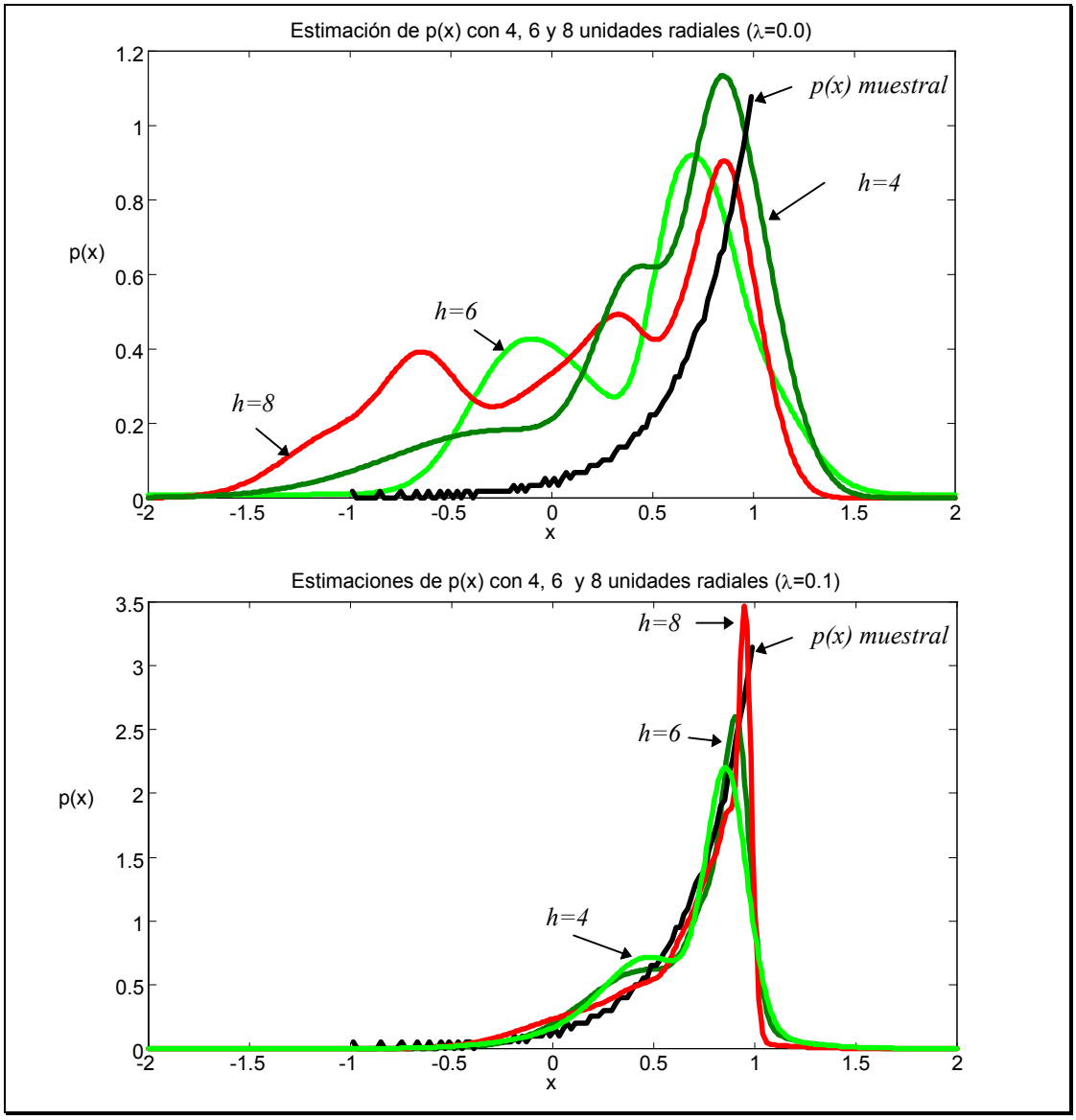

Figura 5.19: Estimación de p(x) con PRBFN de 4, 6 y 8 unidades radiales entrenadas con λ=0.1 y λ=0.0

Nuevamente, la estimación de la fdp de la variable de entrada *x* mejora considerablemente al incluir el término de verosimilitud logarítmica en la función de error. Resulta pues necesario tener en cuenta este término en la función de error si se pretende disponer de una medida de la representación de los vectores de entrada en el conjunto de entrenamiento.

Veamos ahora cómo influye la inclusión de este término en la estimación de la salida deseada:

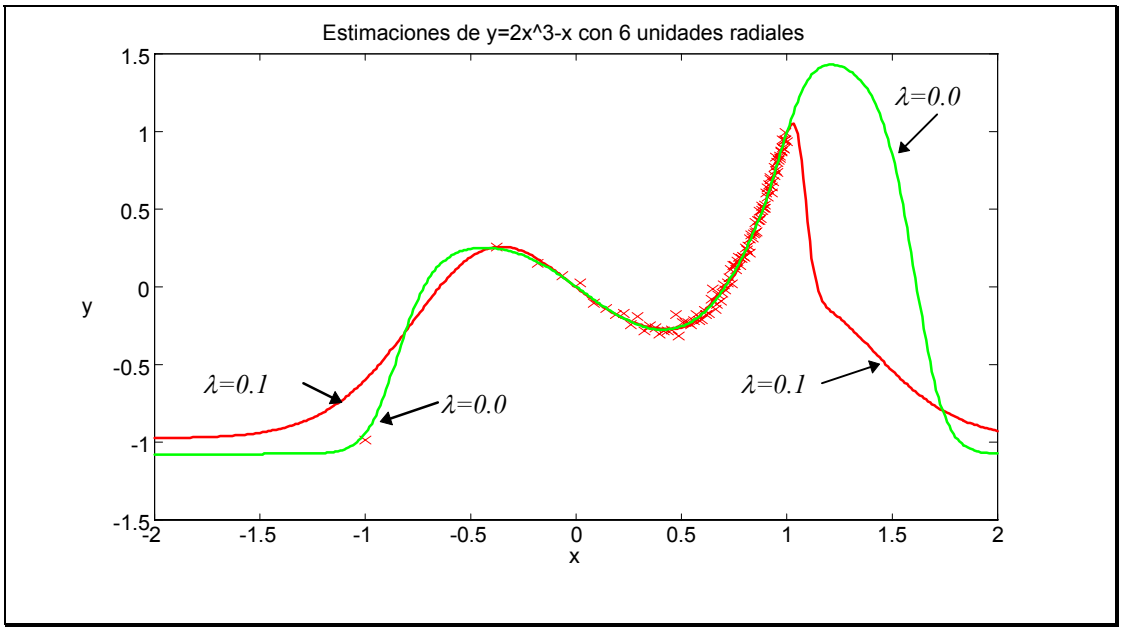

Figura 5.20: Estimaciones de la salida deseada obtenidas con PRBFN de 6 unidades radiales entrenadas con λ=0.1 y λ=0.0

La Figura 5.20 muestra claramente cómo la inclusión del término de verosimilitud en la función de error ha perjudicado la estimación de la salida deseada en la región del espacio de entrada menos representada en el conjunto de entrenamiento.

Como resultado de estos experimentos podemos concluir:

- Es necesario incluir el término de verosimilitud logarítmica en la función de error a minimizar si se pretende estimar con cierta precisión la fdp del vector de entradas.

- En aquellas situaciones en las que el espacio de entrada se encuentra desigualmente representado en el conjunto de entrenamiento (existen zonas muy poco representadas), la inclusión del término de verosimilitud logarítmica en la función de coste puede perjudicar de forma significativa la precisión de la estimación.

El causante de este efecto negativo es el conflicto que se establece entre los dos términos de la función de error: la minimización del error cuadrático medio de la

estimación de la salida tiende a situar unidades radiales en las zonas menos representadas (una vez que las regiones más representadas ya han quedado cubiertas), mientras que la maximización de la verosimilitud logarítmica tiende a expulsar las unidades radiales de estas regiones. El mismo efecto perjudicial puede darse si para realizar una buena estimación en la frontera de la región del espacio de entrada representada en el conjunto de entrenamiento es necesario que los centros de las unidades radiales se salgan fuera de ella. Esta disposición de los centros de las unidades radiales es incompatible con una buena estimación de la fdp del vector de entradas.

La cuestión que se plantea entonces es decidir si merece la pena utilizar dos redes distintas, una para estimar la salida del proceso y otra para la fdp de su vector de entradas, cada una con una función de error distinta, o bien utilizar una única red PRBFN encargada de estimar ambas señales, utilizando la función mixta de error RV.

En el sistema de detección de anomalías que se presentará en el Capítulo 6, se optará por la utilización de dos redes distintas, una utilizada como aproximador funcional para el modelo de funcionamiento normal, y otra como estimador de la fdp del vector de entradas. Esta alternativa deja abierto el camino para utilizar distintos tipos de aproximadores funcionales, y desliga el problema de la estimación de la salida del modelo del problema de la estimación de la fdp de x.

5.5 Ejemplos de aplicación

En este apartado serán presentados dos ejemplos de modelado de procesos dinámicos no lineales donde se utilizan redes neuronales PRBFN como aproximadores funcionales.

Los ejemplos seleccionados corresponden a la predicción de dos series temporales sintéticas que ya han sido utilizadas por otros autores para evaluar el ajuste de modelos no lineales. La primera de ellas es una serie temporal del tipo NARX conocida bajo el nombre de "serie de Chen" (*[Chen et al, 1990]*), mientras que la segunda es una serie bilineal de estructura NARMAX (*[Kuan et al., 1993]*).

Estos ejemplos permitirán ilustrar el procedimiento de modelado de procesos dinámicos que se propuso en el Capítulo 3, así como las capacidades de aproximación de las redes tipo PRBFN. En concreto serán tratados los siguientes aspectos:

- Métodos de análisis lineal:
  - Ajuste de modelos lineales
  - Análisis de correlogramas
  - Análisis de residuos

- Métodos de análisis no lineal:
  - Ajuste de modelos no lineales
  - Análisis Estadístico de Sensibilidades
  - Validación cruzada

- Aplicación de la red PRBFN
  - Como aproximador funcional
  - Como estimador de funciones de densidad probabilista

Como se indica en este último punto, se ha aprovechado la ocasión para ilustrar la utilización de la red PRBFN como estimador de funciones de densidad probabilista. Estas estimaciones permitirán delimitar los *dominios de validez*[i] de los modelos ajustados, sin más que estimar las fdp del vector de regresores de cada modelo. Un valor significativamente pequeño de la fdp del vector de regresores (sea $p(\boldsymbol{x}) < p_{mín}$) indicará que el vector de entradas presentado al aproximador funcional no estaba representado en el conjunto de entrenamiento, y que por lo tanto se desconoce el comportamiento del modelo en esas circunstancias.

[i] Por dominio de validez de un modelo entendemos aquella región de su espacio de entrada en la que es posible caracterizar sus residuos a partir de los datos utilizados para su ajuste.

5.5.1 Ejemplo nº1: Serie de Chen

Tomemos como primer ejemplo la predicción de una serie temporal con estructura NARX propuesta por *[Chen et al, 1990]* y estudiada posteriormente por *[Burrows & Niranjan, 1993]*. El proceso queda descrito por:

$$d[t] = \left(0.8 - 0.5\exp(-d^2[t-1])\right)d[t-1] - \left(0.3 + 0.9\exp(-d^2[t-1])\right)d[t-2] + u[t-1] + 0.2u[t-2] + 0.1u[t-1]u[t-2] + \varepsilon[t]$$

Ecuación 5.68

siendo *{ε[t]}* una serie de ruido blanco gaussiano de media nula y desviación típica 0.2, y *{u[t]}* una serie independiente e idénticamente distribuida según una uniforme de media nula y varianza unidad.

El conjunto de entrenamiento (utilizado para construir la función de error a minimizar durante el ajuste de los parámetros), el conjunto de test (utilizado para estimar la capacidad de generalización de los modelos y detener la minimización paramétrica) y el conjunto de validación (utilizado para evaluar los modelos ajustados) han sido generados según la Ecuación 5.68 dedicando 500 muestras a cada uno de ellos.

El estudio de la serie temporal comienza con el análisis lineal de los datos disponibles. En la Figura 5.21 aparecen la función de autocorrelación simple y la función de autocorrelación parcial de la variable de salida *d*, y la función de correlación cruzada entre las variables *d* (salida) y *u* (entrada). Las funciones de autocorrelación de la variable de salida sugieren un comportamiento autoregresivo de segundo orden, mientras que la función de autocorrelación cruzada sugiere una influencia de orden no superior a 6 para la variable exógena *u*. Estas razones nos inclinan a ensayar un modelo ARX(2,6). Como segundo modelo lineal ensayaremos un modelo ARX(2,2), ya que es el modelo que incluye todos los regresores que intevienen en la Ecuación 5.68 (*d[t-1]*, *d[t-2]*, *u[t-1]* y *u[t-2]*). Los modelos resultantes son:

- Modelo ARX(2,6):

$$y[t] = 0.78d[t-1] - 0.45d[t-2] + u[t-1] + 0.49u[t-2] - 0.025u[t-3] - 0.12u[t-4] - 0.14u[t-5] - 0.024u[t-6]$$

- Modelo ARX(2,2):

$$y[t] = 0.85d[t-1] - 0.55d[t-2] + 1.00u[t-1] + 0.42u[t-2]$$

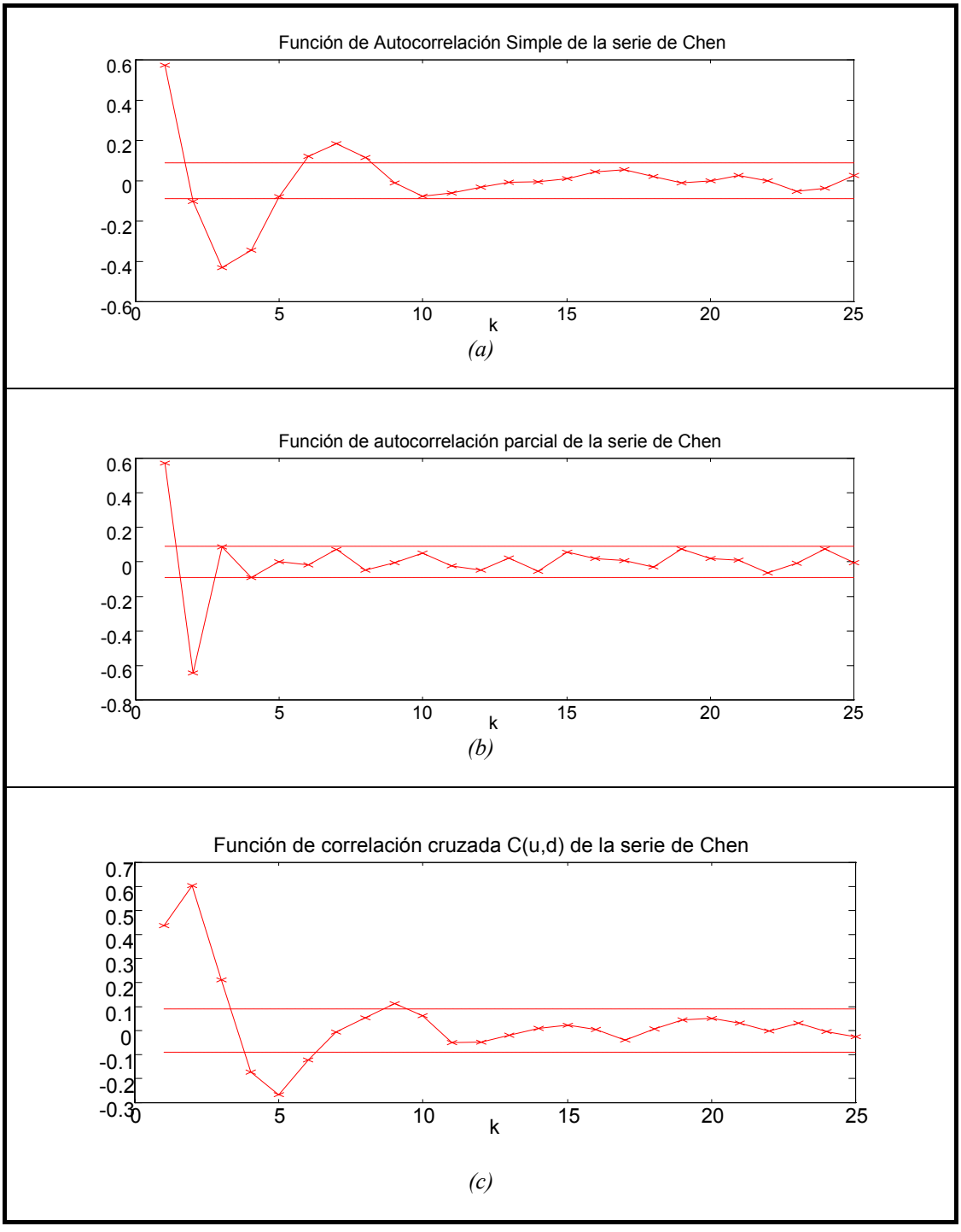

Figura 5.21: Análisis lineal de la serie de Chen:

(a) Función de Autocorrelación Simple (b) Función de Autocorrelación Parcial
(c) Función de Correlación Cruzada C(u,d)

Para analizar los modelos ensayados se ha utilizado el conjunto de validación, y para cada uno de ellos se ha calculado

- La desviación típica del error de estimación del conjunto de entrenamiento: $std(e_{entren})$
- La desviación típica del error de estimación del conjunto de validación: $std(e_{valid})$
- El estadístico de Ljung y Box para comprobar la independencia de los residuos de entrenamiento (Q_{lim}=35.17): Q_{entren}
- El estadístico de Ljung y Box para comprobar la independencia de los residuos de validación (Q_{lim}=35.17): Q_{valid}

La siguiente tabla recoge los resultados obtenidos:

	$std(e_{entren})$	$std(e_{valid})$	Q_{entren} (Q_{lim}=35.17)	Q_{valid} (Q_{lim}=35.17)
ARX(2,6)	0.61	0.59	32.48	33.98
ARX(2,2)	0.63	0.58	36.62	46.58

Tabla 5.1: Análisis de los modelos lineales ensayados sobre la serie de Chen

Aunque el análisis de independencia de los residuos sea favorable en el caso del modelo ARX(2,6), los errores de estimación obtenidos tienen una desviación típica muy superior a la desviación típica del ruido de la serie original ($std(\varepsilon)=0.2$). Esta característica revela la incapacidad de modelar esta serie con modelos lineales ensayados.

Una vez descartados los modelos lineales, pasemos al caso no lineal. En primer lugar ajustaremos un estimador de la fdp $p(u[t-1],u[t-2],d[t-1],d[t-2])$ que nos permita disponer de una medida de extrapolación de los modelos ajustados con el mismo conjunto de entrenamiento. Utilizaremos como aproximador una red PRBFN tipo I con 10 unidades radiales y la ajustaremos maximizando la verosimilitud logarítmica de los vectores de entrada del conjunto de entrenamiento.

Para calcular la *cota de extrapolación*, estimaremos la función de distribución $p(u[t-1],u[t-2],d[t-1],d[t-2])$ sobre el conjunto de entrenamiento (ver Figura 5.22) y tomaremos como cota de extrapolación el valor de la fdp correspondiente a una función de distribución del 5%. El valor obtenido en este caso es $p_{mín}=10^{-3.5}$, de tal forma que consideraremos desconocida aquella región del espacio de entrada que arroje un valor estimado de $p(u[t-1],u[t-2],d[t-1],d[t-2]) < p_{mín}$.

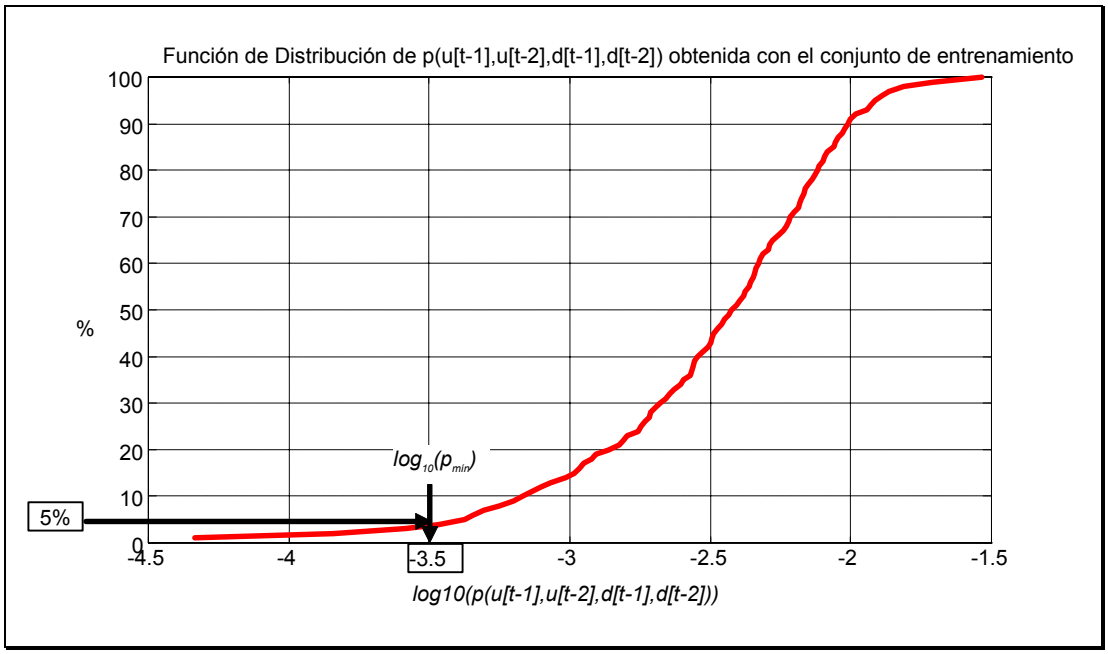

Figura 5.22: Función de distribución de p(u[t-1],u[t-2],d[t-1],d[t-2]) estimada con el conjunto de entrenamiento de la serie de Chen. Determinación de la cota de extrapolación

Para ilustrar el proceso de selección de variables de entrada con el Análisis Estadístico de Sensibilidades, comenzaremos ajustando un modelo NARX(4,4) (4 retardos de la entrada exógena *u* y 4 retardos de la salida deseada *d*), utilizando como aproximador una red PRBFN tipo II con 15 unidades radiales. Los resultados obtenidos sobre parte de los conjuntos de entrenamiento y validación se muestran en la Figura 5.23 y en la Figura 5.24 respectivamente.

Como puede apreciarse en estas figuras, la estimación de la salida deseada e incluso la del ruido de la serie son muy aceptables. El test de Ljung y Box muestra sin embargo cierto grado de dependencia en los residuos de validación.

Es importante resaltar cómo la red de estimación de *p(u[t-1],u[t-2],d[t-1],d[t-2])* permite identificar vectores de entrada poco representados en el conjunto de entrenamiento, para los que se desconoce en principio el nivel de precisión de la estimación de la salida del modelo de funcionamiento normal. En la serie de validación mostrada en la Figura 5.24 ocurre este fenómeno en dos ocasiones, sin deterioro significativo de la precisión de la estimación.

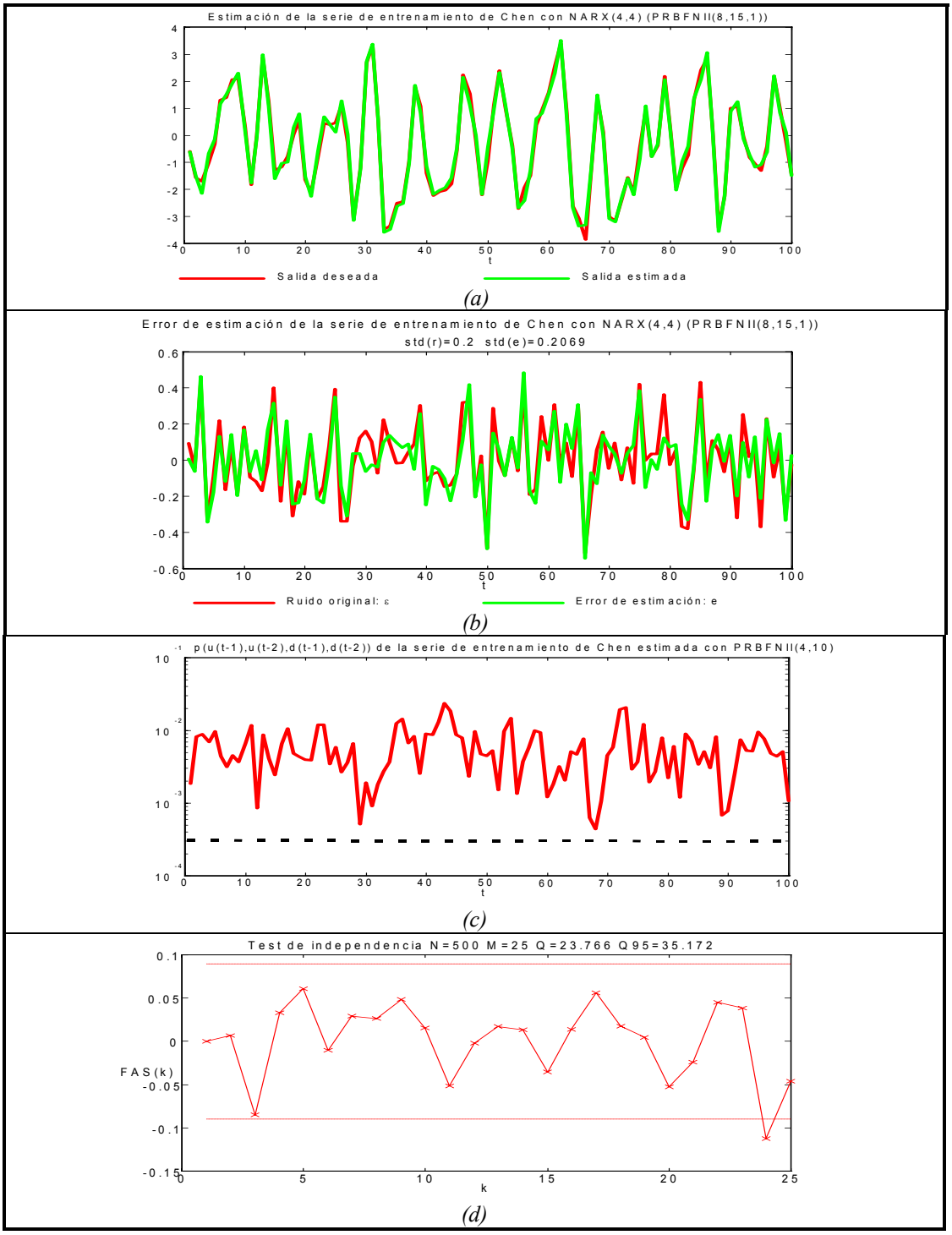

Figura 5.23: Estimación de la serie de Chen con el modelo NARX(4,4) utilizando como aproximador una red PRBFN II con 15 unidades radiales
(a) Estimación de la serie de entrenamiento (b) Error de estimación de la serie de entrenamiento
(c) fdp de (u[t-1], u[t-2], d[t-1], d[t-2]) estimada para la serie de entrenamiento (red PRBFN(4,10))
(d) Test de independencia de los residuos

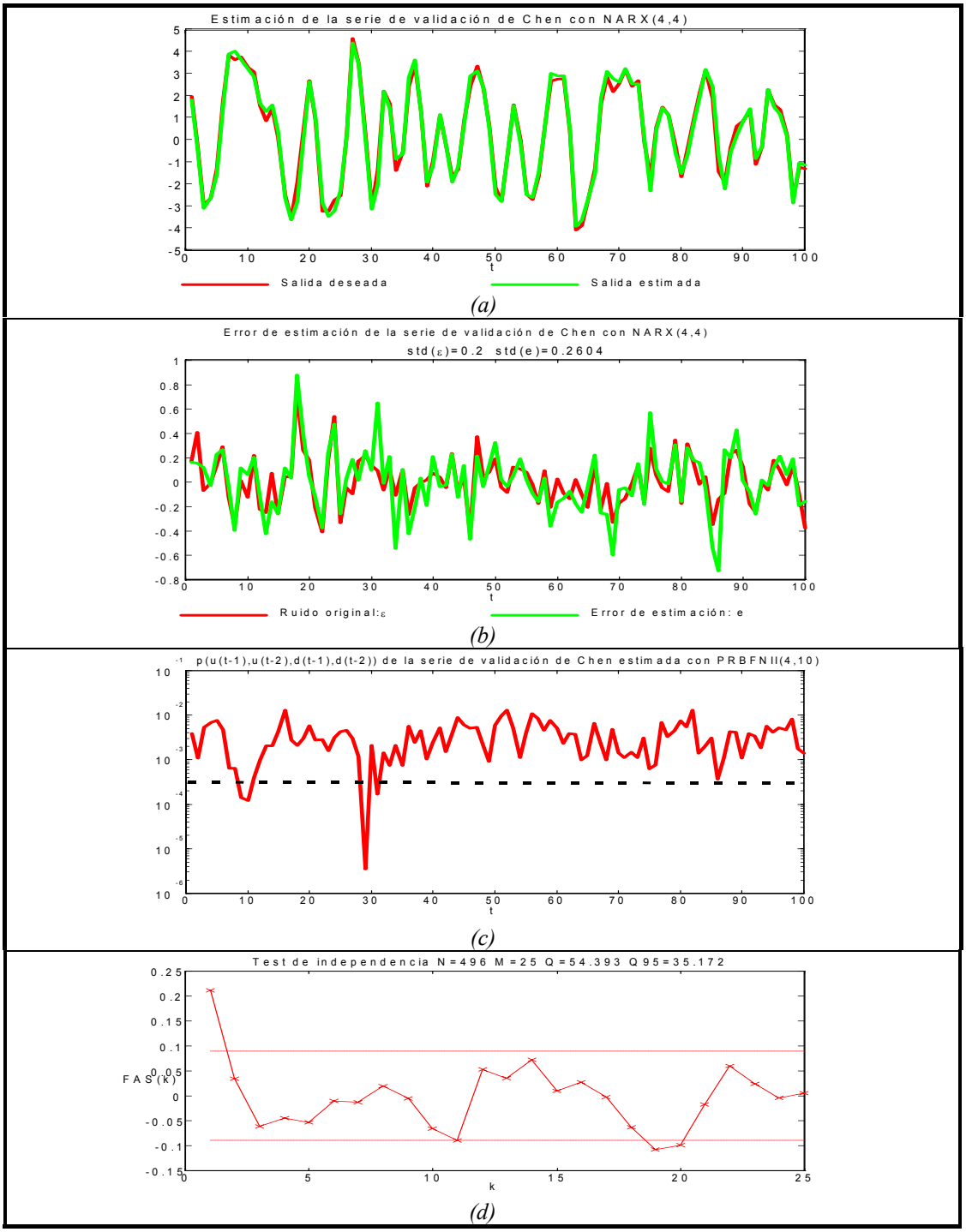

Figura 5.24: Estimación de la serie de Chen con el modelo NARX(4,4) utilizando como aproximador una red PRBFN II con 15 unidades radiales

(a) Estimación de la serie de validación (b) Error de estimación de la serie de validación
(c) fdp de (u[t-1], u[t-2], d[t-1], d[t-2]) estimada para la serie de validación (red PRBFN(4,10))
(d) Test de independencia de los residuos

Para tratar de mejorar el modelo eliminando las variables de entrada irrelevantes, apliquemos el Análisis Estadístico de Sensibilidades al modelo previamente ajustado:

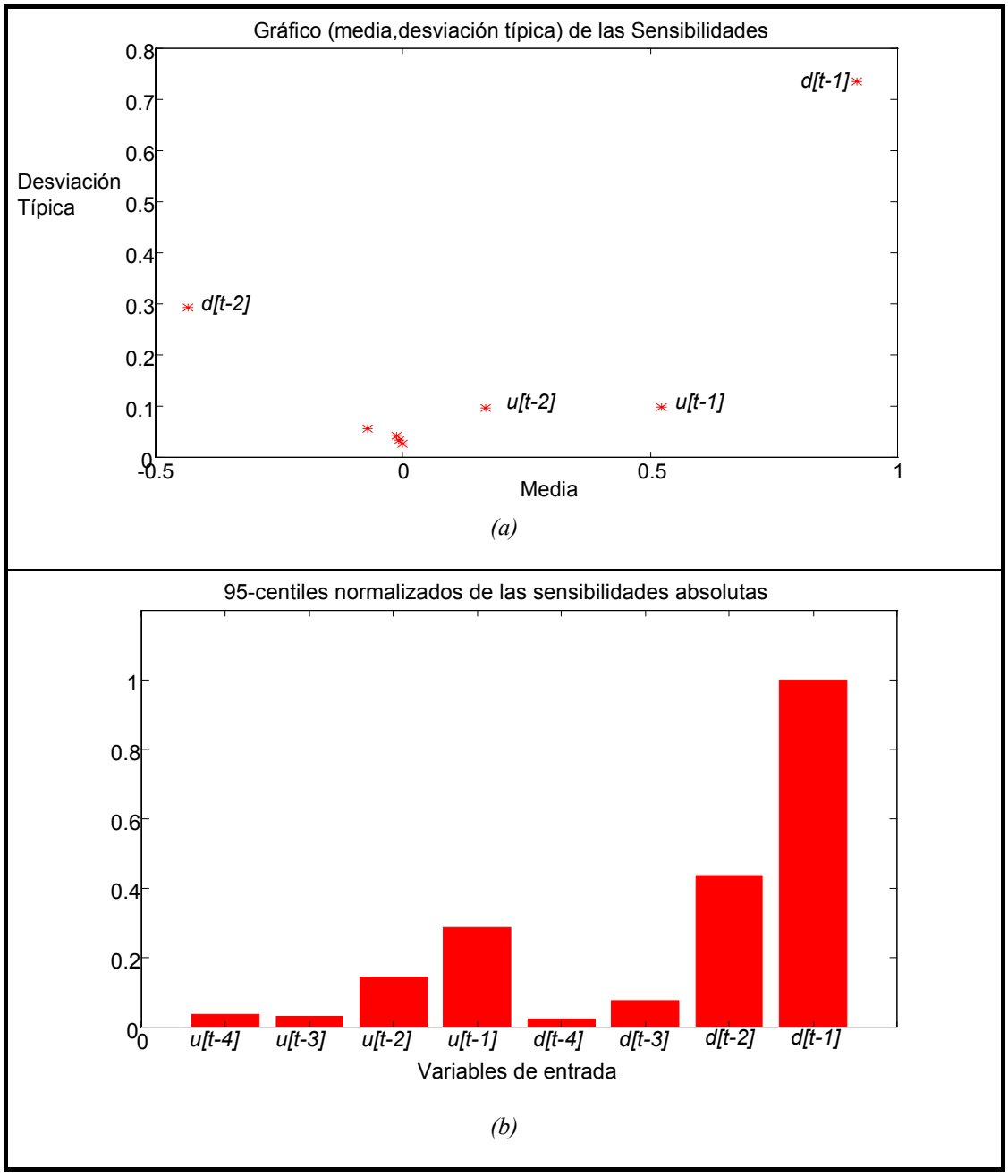

Figura 5.25: Estimación de la serie de Chen con el modelo NARX(4,4) utilizando como aproximador una red PRBFN II con 15 unidades radiales. Aplicación del Análisis Estadístico de Sensibilidades.

(a) Gráfico (media, desviación típica) de las sensibilidades
(b) 95-centiles normalizados de las sensibilidades absolutas

Como puede comprobarse en la Figura 5.25, las variables *u[t-1], u[t-2], d[t-1] y d[t-2]* se muestran efectivamente como las variables de entrada significativamente influyentes en la salida.

Eliminando del modelo las variables de entrada irrelevantes (*u[t-3], u[t-4], d[t-3] y d[t-4]*), ajustemos un modelo NARX(2,2) utilizando nuevamente como aproximador una red PRBFN tipo II con 15 unidades radiales. Los resultados de la estimación de la serie de entrenamiento y validación se ilustran en la Figura 5.26 y en la Figura 5.27 respectivamente.

Como era de esperar, la simplificación del modelo ha mejorado significativamente su capacidad de generalización. Por un lado se ha conseguido reducir la varianza del error de validación. Por otro lado se ha reducido también el valor del estadístico de Ljung y Box para la serie de error de validación, dando un resultado positivo en el test de independencia.

La Tabla 5.2 recoge los resultados obtenidos con los modelos ensayados:

	$std(e_{entren})$	$std(e_{valid})$	Q_{entren} (Q_{lim}=35.17)	Q_{valid} (Q_{lim}=35.17)
ARX(2,2)	0.63	0.58	36.62	46.58
ARX(2,6)	0.61	0.59	32.48	33.98
NARX(4,4)	0.2069	0.2604	23.766	54.393
NARX(2,2)	0.2006	0.2477	35.831	27.979

Tabla 5.2: Análisis de los modelos ensayados sobre la serie de Chen

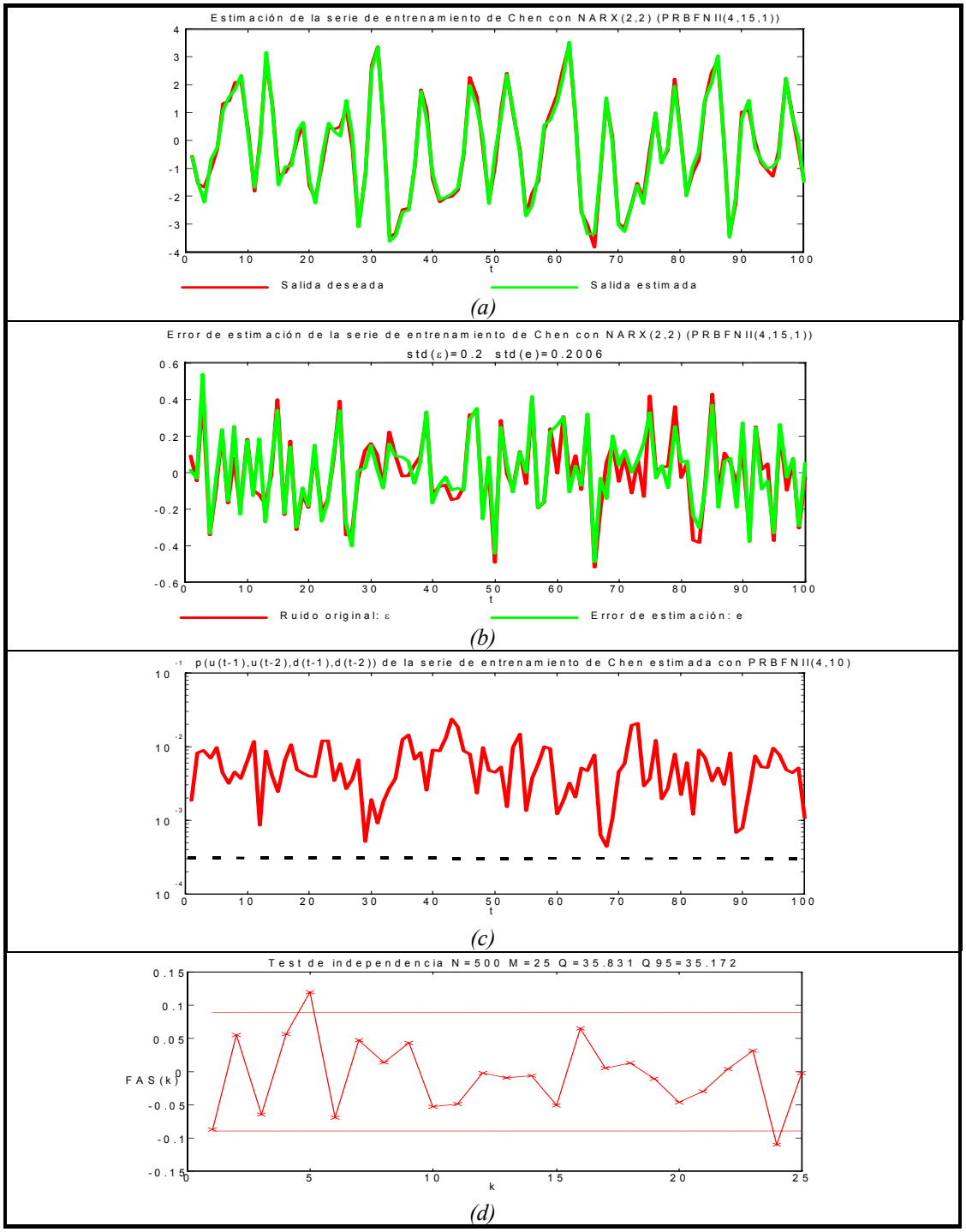

Figura 5.26: Estimación de la serie de Chen con el modelo NARX(2,2) utilizando como aproximador una red PRBFN II con 15 unidades radiales

(a) Estimación de la serie de entrenamiento (b) Error de estimación de la serie de entrenamiento (c) fdp de (u[t-1], u[t-2], d[t-1], d[t-2]) estimada para la serie de entrenamiento (red PRBFN(4,10)) (d) Test de independencia de los residuos

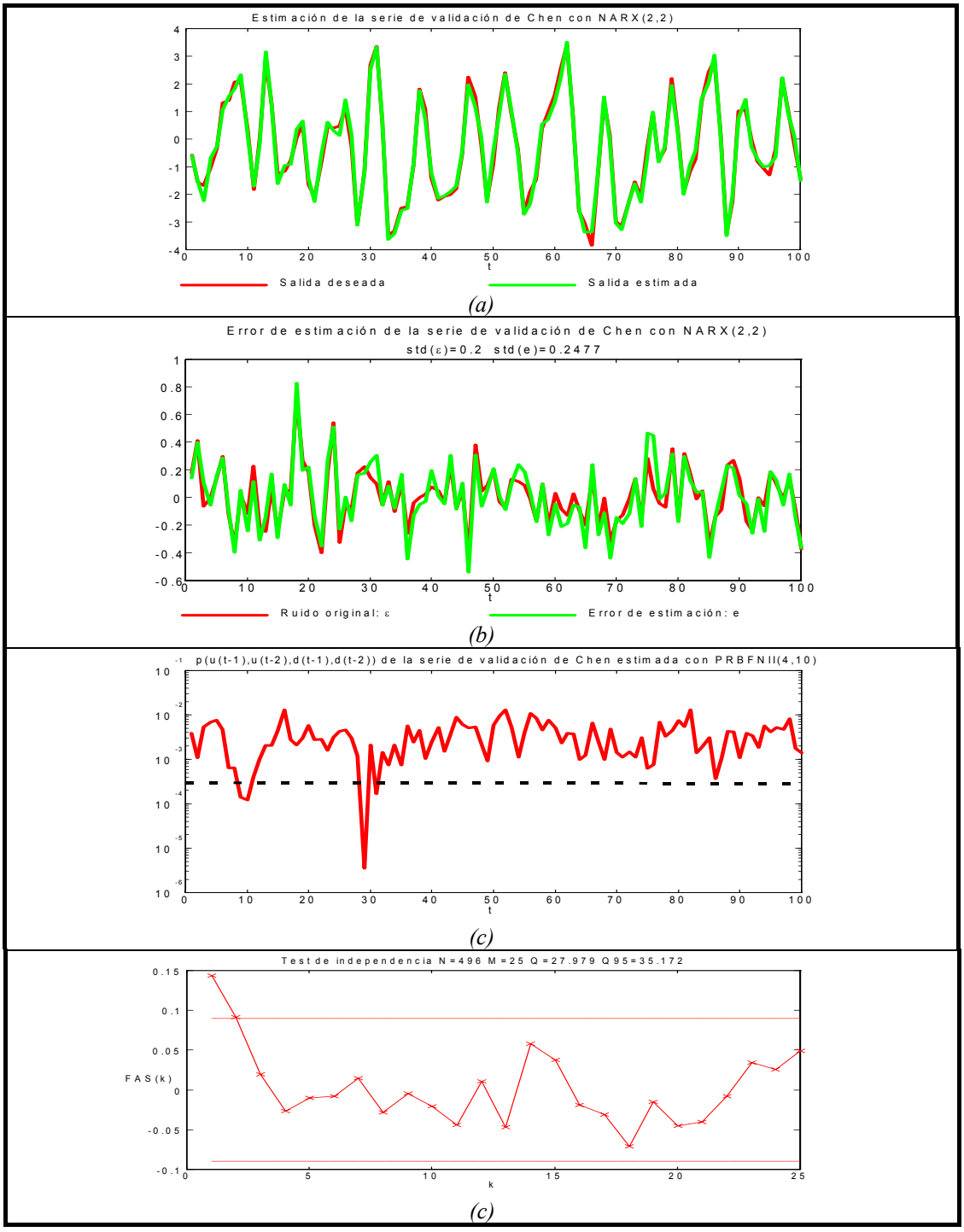

Figura 5.27: Estimación de la serie de Chen con el modelo NARX(2,2) utilizando como aproximador una red PRBFN II con 15 unidades radiales

(a) Estimación de la serie de validación (b) Error de estimación de la serie de validación
(c) fdp de (u[t-1], u[t-2], d[t-1], d[t-2]) estimada para la serie de validación (red PRBFN(4,10))
(d) Test de independencia de los residuos

5.5.2 Ejemplo nº2: Serie Bilineal

Con este segundo ejemplo vamos a ilustrar el proceso de modelado de un sistema con características NARMAX, correspondiente a la serie temporal conocida bajo el nombre de serie bilineal (*[Kuan et al., 1993]*) definida por:

$$d[t] = 0.4d[t-1] - 0.3d[t-2] + 0.5d[t-1]\varepsilon[t-1] + \varepsilon[t]$$

Ecuación 5.69

siendo $\varepsilon[t]$ un ruido blanco gaussiano distribuido según una N(0,1)

La dependencia no lineal de $d[t]$ con $\varepsilon[t-1]$ hace necesario, para la estimación de esta serie, el ajuste de un modelo NARMA con al menos dos retardos autoregresivos (AR) y uno de media móvil (MA). Para ilustrar la aplicación del modelo NARMA, generaremos una serie de 1500 elementos según la Ecuación 5.69 y dedicaremos 500 para entrenamiento, 500 para test y 500 para validación.

De esta forma utilizaremos el conjunto de entrenamiento para formar la función de error a minimizar durante el aprendizaje, el conjunto de test para detener la optimización paramétrica y el conjunto de validación para analizar la bondad de los modelos finalmente ajustados.

Para poder delimitar el dominio de validez de los modelos ajustados, se ha entrenado una red PRBFN tipo I con 5 unidades radiales para estimar la función de densidad probabilista de la variable aleatoria vectorial (*d[t-1],d[t-2]*), que formará parte del vector de entradas de los modelos. Esta señal (*p(d[t-1],d[t-2])*) permitirá detectar muestras no representadas en el conjunto de entrenamiento, para las cuales las estimaciones no serán fiables. Como cota inferior de *p(d[t-1],d[t-2])* podemos tomar el 5-centil de la distribución de la fdp *p(d[t-1],d[t-2]* estimada sobre el conjunto de entrenamiento, obteniendo un valor aproximado de $10^{-2.5}$ (ver Figura 5.28). De esta forma diremos que el modelo ajustado con el conjunto de entrenamiento está extrapolando cuando se le presente un ejemplo para el cual se cumpla que la fdp $p(d[t-1],d[t-2]) < 10^{-2.5}$.

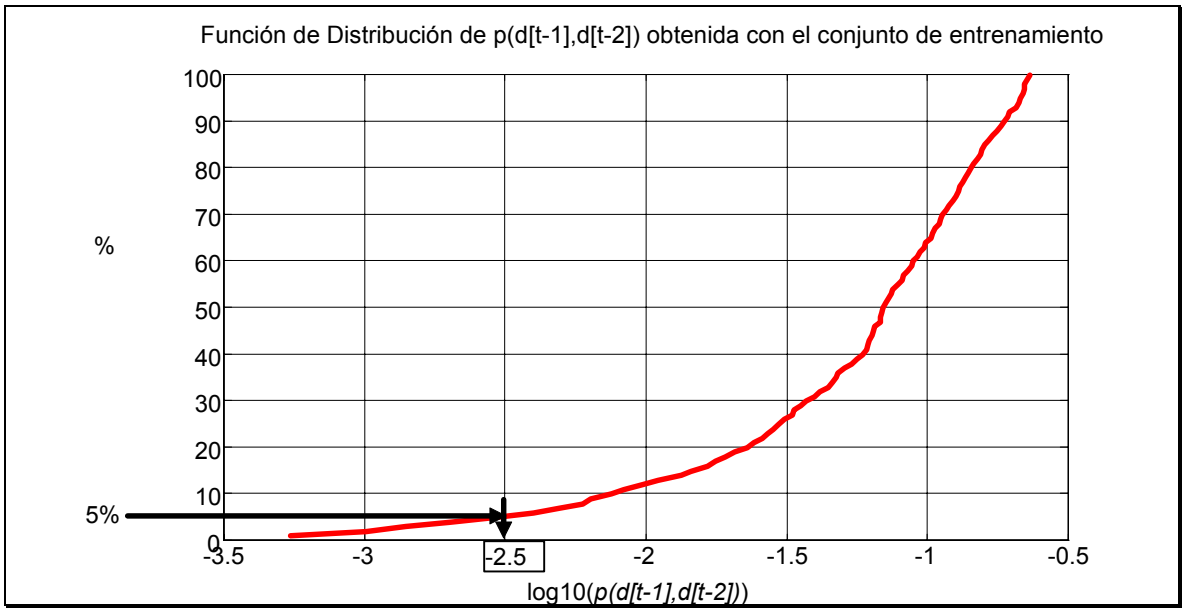

Figura 5.28: Función de distribución de (p(d[t-1],d[t-2]) estimada con el conjunto de entrenamiento de la serie bilineal. Determinación de la cota inferior de extrapolación.

La Figura 5.29 muestra las curvas de nivel de la fdp estimada y la localización en el plano *(d[t-1],d[t-2])* de los ejemplos de entrenamiento *(a)* y de validación *(b)*. En ellas queda patente la existencia en ambos conjuntos de ejemplos "extraños" ("outliers") que han quedado poco representados en el conjunto de entrenamiento. Veremos a continuación cómo se comportan los distintos modelos ajustados con estos ejemplos.

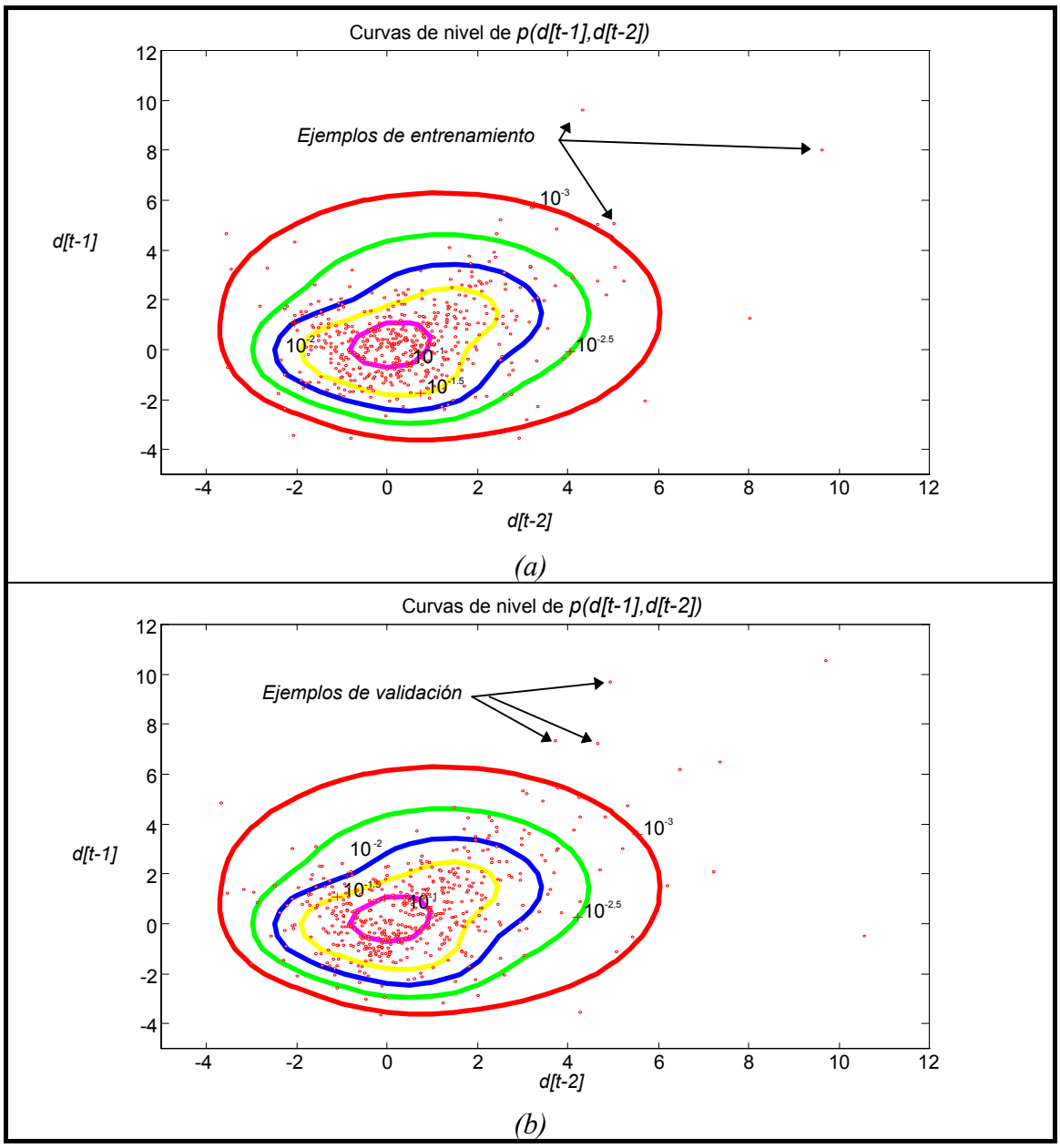

Figura 5.29: Serie bilineal. Curvas de nivel de la fdp p(d[t-1],d[t-2]) estimada con PRBFN I
(a) localicación de los ejemplos de entrenamiento (b) localización de los ejemplos de validación

El proceso de modelado comienza con el análisis lineal de la serie bilineal. El primer paso es analizar las funciones de autocorrelación simple y parcial de la serie, con el objetivo de determinar la estructura de los modelos lineales a ensayar. Estas funciones se representan en la Figura 5.30:

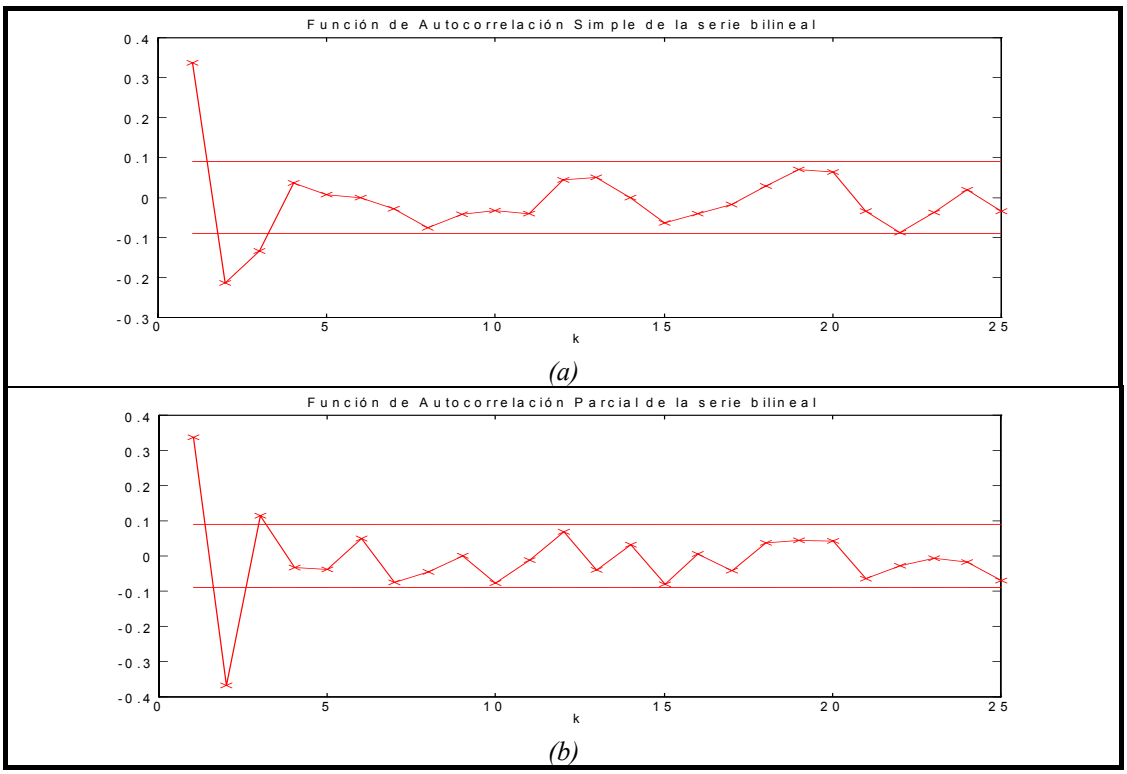

Figura 5.30: Funciones de Autocorrelación Simple (a) y Parcial (b) de la serie bilineal

La fuerte componente autoregresiva de la serie se ha visto reflejada en la función de autocorrelación parcial, donde los dos primeros coeficientes muestran un valor significativamente elevado. El análisis simultáneo de las dos funciones de correlación apunta hacia un modelo lineal del tipo ARMA(2,2). Los modelos lineales ajustados con el conjunto de entrenamiento son por tanto:

- Un modelo auroregresivo de orden 2: AR(2), resultando:

$$y[t]=0.461\ d[t-1]-0.370\ d[t-2]$$

- Un modelo autoregresivo de media móvil de orden (2,2): ARMA(2,2):

$$y[t]=0.146\ d[t-1]-0.322\ d[t-2]+0.572\ e[t-1]+0.087\ e[t-2]$$

Para analizar los modelos ensayados se ha utilizado el conjunto de validación, y para cada uno de ellos se ha calculado

- La desviación típica del error de estimación del conjunto de entrenamiento: $std(e_{entren})$
- La desviación típica del error de estimación del conjunto de validación: $std(e_{valid})$
- El estadístico de Ljung y Box para comprobar la independencia de los residuos de entrenamiento (Q_{lim}=35.17): Q_{entren}
- El estadístico de Ljung y Box para comprobar la independencia de los residuos de validación (Q_{lim}=35.17): Q_{valid}

La siguiente tabla recoge los resultados obtenidos:

	$std(e_{entren})$	$std(e_{valid})$	Q_{entren} (Q_{lim}=35.17)	Q_{valid} (Q_{lim}=35.17)
AR(2)	1.34	1.50	28.71	30.28
ARMA(2,2)	1.34	1.50	21.57	28.57

Tabla 5.3: Análisis de los modelos lineales ensayados sobre la serie bilineal

Como era de esperar, los modelos lineales arrojan unos errores de estimación de varianza muy superior a la del ruido de la serie original (de varianza unitaria). El origen de estas discrepancias es la componente no lineal de la serie bilineal (*0.5d[t-1]ε[t-1]*) que entorpece la aplicación del análisis lineal.

Es importante resaltar que aunque los residuos no estén significativamente autocorrelados entre sí desde un punto de vista lineal (todos los valores del estadístico de Ljung y Box han quedado por debajo de su cota), sí que existe una relación no lineal entre ellos. Esta observación pone de manifiesto la dificultad de la detección de no linealidades, y por tanto de la validación de modelos lineales.

La utilización de herramientas de validación basadas en criterios de correlación lineal son en muchos casos insensibles a relaciones no lineales entre las variables involucradas. Esta circunstancia favorece la utilización de técnicas de validación cruzada, y del intuitivo arte de la prueba y error.

Comencemos el análisis no lineal de la serie ajustando un modelo NARMA (4,4) (4 entradas AR y 4 entradas MA) para ilustrar la selección de variables de entrada mediante el Análisis Estadístico de Sensibilidades.

El aproximador funcional utilizado en este caso es una red PRBFN tipo II con 15 unidades radiales. Los resultados obtenidos sobre los conjuntos de entrenamiento y validación se muestran el la Figura 5.31 y en la Figura 5.32 respectivamente.

Al disponer en este caso de la serie de ruido blanco $\varepsilon[t]$, podemos comparar el error de estimación de los modelos ensayados con el ruido de la serie, que deberían coincidir en el caso ideal. Para ello se han dibujado las estimaciones y los errores de estimación correspondientes a una ventana de 100 muestras para cada una de las series. En ellas puede apreciarse la coincidencia entre los errores de estimación y las series de ruido, salvo en ciertas regiones donde el análisis de la fdp *p(d[t-1],d[t-2])* muestra que son zonas de extrapolación, poco representadas en el conjunto de entrenamiento. La desviación típica del error de estimación de la serie de entrenamiento es en este caso de 1.048 (frente a la varianza unitaria de la serie de ruido blanco), mientras que la desviación típica del error de estimación de la serie de validación es de 1.368.

Los test de independencia de los residuos prueban la falta de correlación en el caso de los residuos de entrenamiento, mientras que en el caso de validación se presenta una débil autocorrelación para los retardos de orden 1 y 4.

Figura 5.31: Estimación de la serie bilineal con el modelo NARMA(4,4) utilizando como aproximador una red PRBFN II con 15 unidades radiales

(a) Estimación de la serie de entrenamiento (b) Error de estimación de la serie de entrenamiento (c) fdp de (d[t-1],d[t-2]) estimada para la serie de entrenamiento (red PRBFN I con 5 unidades radiales) (d) Test de independencia de los residuos

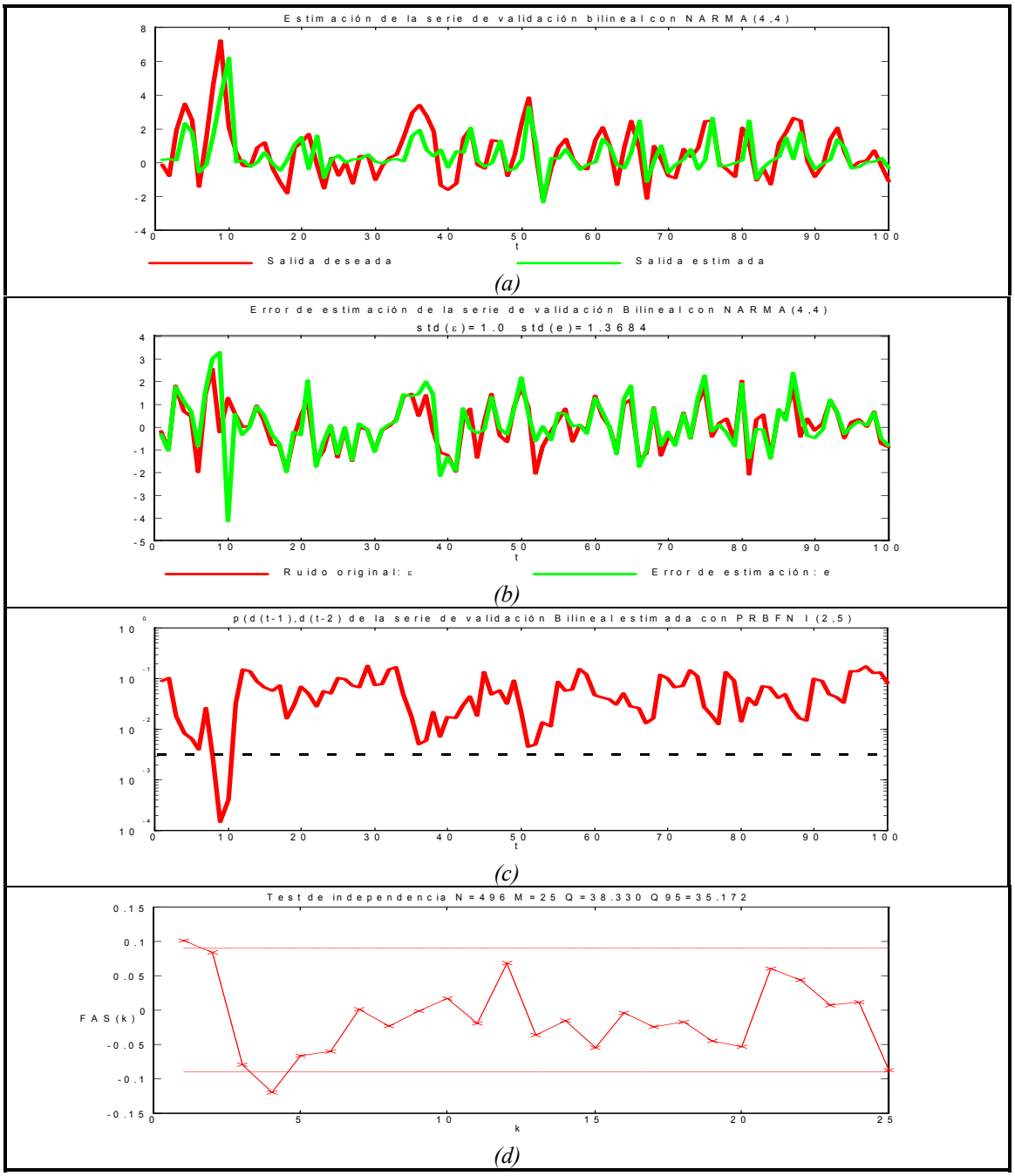

Figura 5.32: Estimación de la serie bilineal con el modelo NARMA(4,4) utilizando como aproximador una red PRBFN II con 15 unidades radiales

(a) Estimación de la serie de validación (b) Error de estimación de la serie de validación (c) fdp de (d[t-1],d[t-2]) estimada para la serie de validación (red PRBFN I con 5 unidades radiales) (d) Test de independencia de los residuos

Para tratar de mejorar el modelo eliminando aquellas variables de entrada poco relevantes, apliquemos el Análisis Estadístico de Sensibilidades (AES) al modelo ajustado:

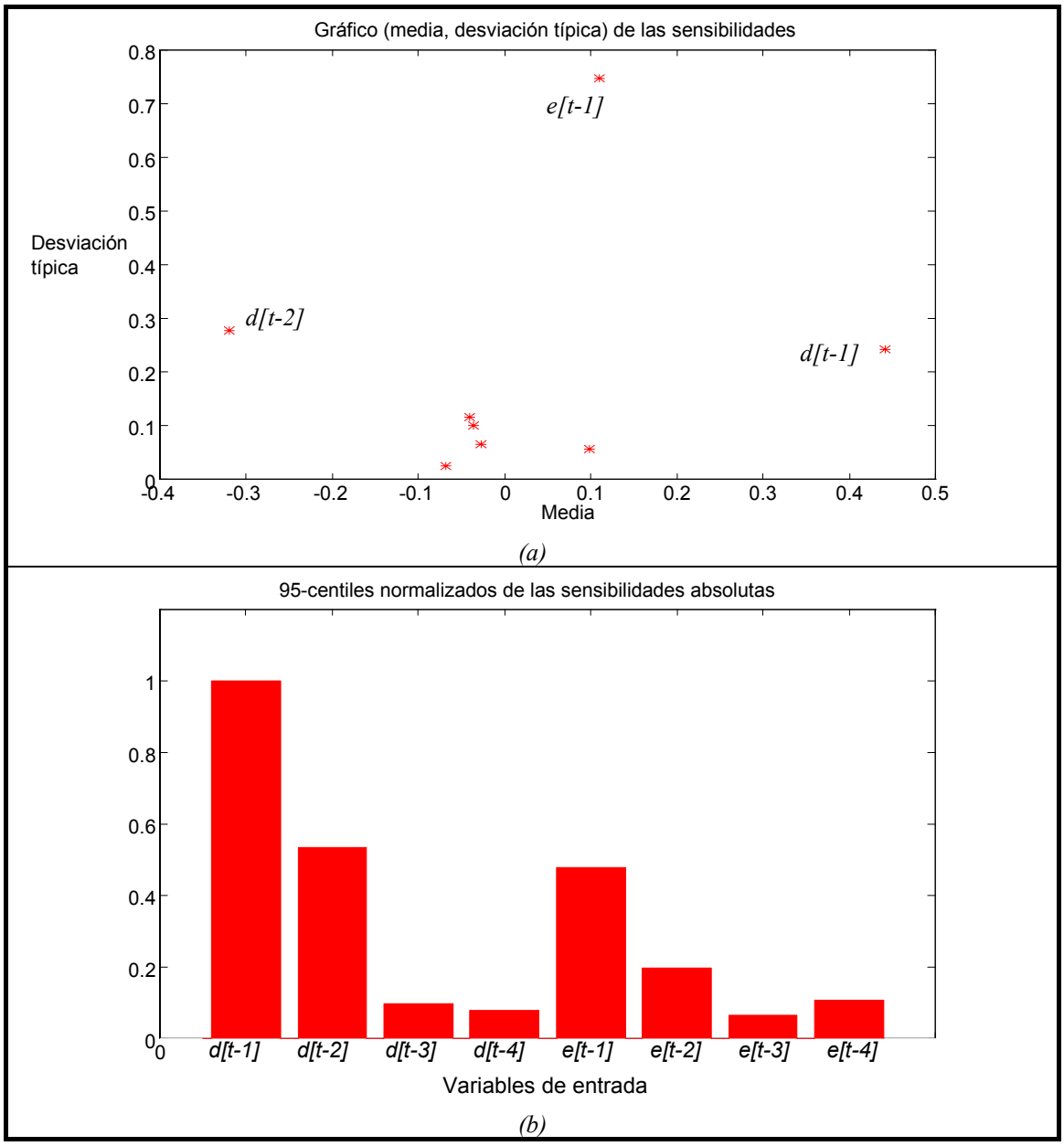

Figura 5.33: Estimación de la serie bilineal con el modelo NARMA(4,4) utilizando como aproximador una red PRBFN II con 15 unidades radiales. Aplicación del Análisis Estadístico de Sensibilidades

(a) Gráfico (media, desviación típica) de las sensibilidades
(b) 95-centiles normalizados de las sensibilidades absolutas

El análisis de las distribuciones estadísticas de las sensibilidades muestra una clara influencia de las variables de entrada *d[t-1]*, *d[t-2]* y *e[t-1]*, que son las que realmente aparecen en el modelo original. Estas distribuciones revelan también una ligera influencia en el modelo ajustado de la variable *e[t-2]*.

Para reducir la complejidad del modelo eliminemos las variables de entrada poco significativas y rejustemos un modelo NARMA(2,1). Los resultados obtenidos con este nuevo modelo se muestran en la Figura 5.34 y en la Figura 5.35.

Las desviaciones típicas de los errores de entrenamiento y validación se han reducido respectivamente a 1.001 y 1.295. El análisis de independencia de los residuos es ahora también favorable para la serie de validación. La reducción de la complejidad del modelo ha aumentado pues su capacidad de generalización, mejorando la calidad del modelo.

La Tabla 5.4 recoge los resultados obtenidos con los modelos ensayados:

	$std(e_{entren})$	$std(e_{valid})$	Q_{entren} (Q_{lim}=35.17)	Q_{valid} (Q_{lim}=35.17)
AR(2)	1.34	1.50	28.71	30.28
ARMA(2,1)	1.34	1.50	21.57	28.57
NARMA(4,4)	1.048	1.368	19.44	38.33
NARMA(2,1)	1.001	1.295	18.33	34.42

Tabla 5.4: Resultados obtenidos con los modelos ajustados sobre la serie bilineal

Como conclusiones de este experimento podemos resaltar:

- La ventaja de disponer de una medida de la extrapolación del aproximador. Esta información permite identificar ejemplos poco representados en el conjunto de entrenamiento, para los cuales la estimación puede no ser fiable. Como veremos en el Capítulo 6, esta información será vital en el campo del diagnóstico.

- La dificultad de detectar las no linealidades de los procesos: la insensibilidad de los métodos de análisis lineal a comportamientos no lineales hace necesaria la aplicación de métodos de validación cruzada.

- La utilidad del Análisis Estadístico de Sensibilidades como herramienta de selección de las variables de entrada de los modelos. La reducción de la complejidad de los modelos trae consigo una mejora en la capacidad de generalización de los mismos.

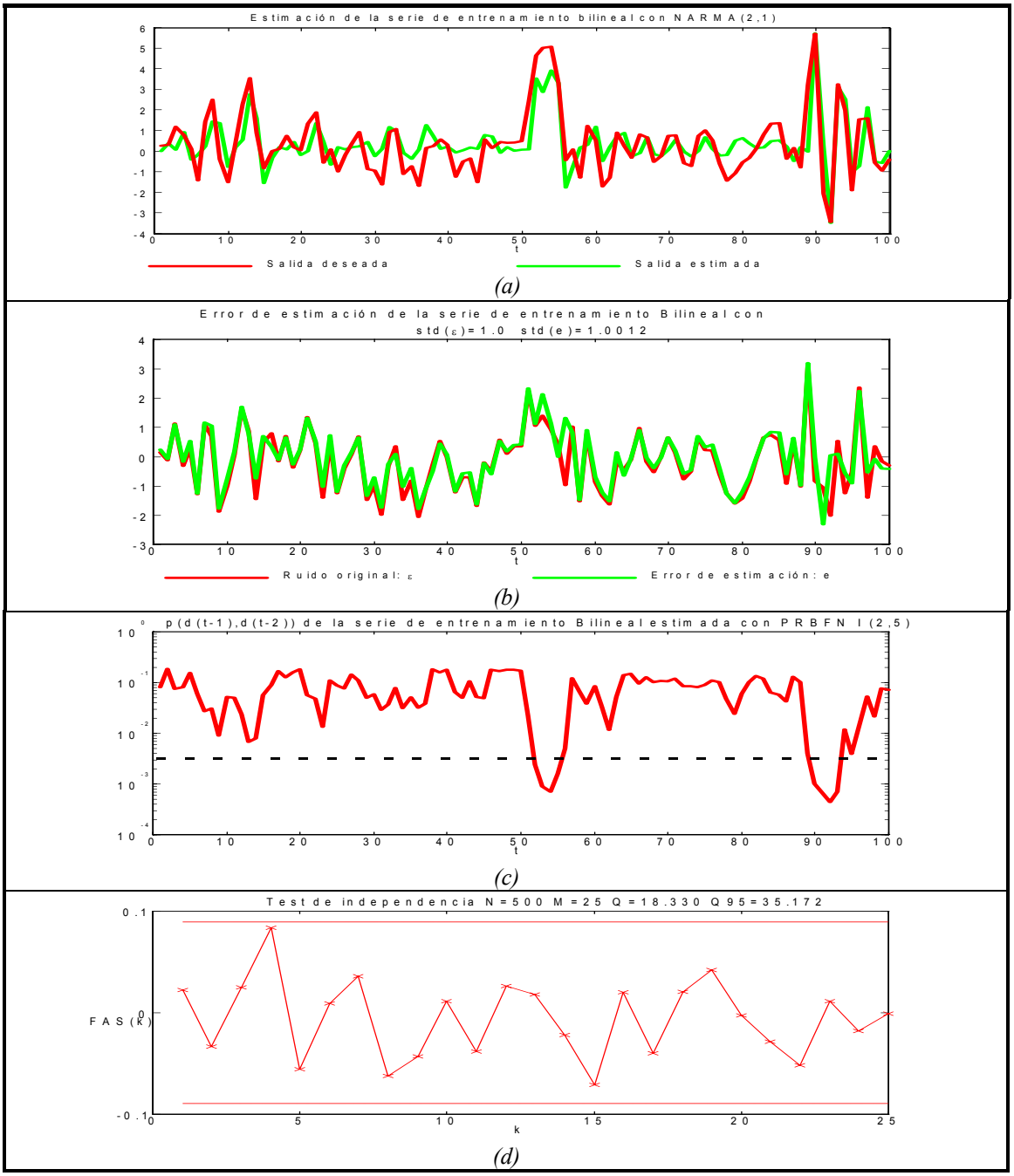

Figura 5.34: Estimación de la serie bilineal con el modelo NARMA(2,1) utilizando como aproximador una red PRBFN II con 15 unidades radiales

(a) Estimación de la serie de entrenamiento (b) Error de estimación de la serie de entrenamiento (c) fdp de (d[t-1],d[t-2]) estimada para la serie de entrenamiento (red PRBFN I con 5 unidades radiales) (d) Test de independencia de los residuos

Figura 5.35: Estimación de la serie bilineal con el modelo NARMA(2,1) utilizando como aproximador una red PRBFN II con 15 unidades radiales

(a) Estimación de la serie de validación (b) Error de estimación de la serie de validación
(c) fdp de (d[t-1],d[t-2]) estimada para la serie de validación (red PRBFN I con 5 unidades radiales)
(d) Test de independencia de los residuos

6. Sistema de detección de anomalías incipientes basado en el modelado conexionista del funcionamiento normal de los componentes

Este capítulo presenta el sistema que se propone en esta tesis para la detección de anomalías incipientes en procesos industriales complejos. El sistema propuesto está basado en el modelado conexionista del funcionamiento normal de los componentes que integran el proceso a supervisar, de tal forma que la detección se lleva a cabo mediante el análisis de los residuos obtenidos al comparar los comportamientos medidos y predichos.

Una vez presentado el sistema, se expondrán dos aplicaciones del mismo a casos reales de diagnóstico: la primera de ellas tratará la supervisión del funcionamiento del condensador de una central térmica. La sencillez del modelo de funcionamiento normal aplicado en este caso permitirá ilustrar la capacidad del procedimiento de detección de anomalías. La segunda aplicación corresponde al diagnóstico de la química del agua de una Central Térmica. En este caso el modelo de funcionamiento normal es ya un modelo dinámico que ilustrará las técnicas de modelado presentadas en el Capítulo 3. Por último referir al lector al trabajo *[Muñoz et al., 1995-2]* donde se describe una nueva aplicación de este sistema a la supervisión de la llama de la caldera de una Central Térmica. El prototipo de esta aplicación fue instalado en el año 1993 en una Central Térmica.

6.1 Introducción

Este capítulo constituye el núcleo central de esta tesis y presenta el sistema propuesto para la detección de anomalías incipientes en procesos industriales, basado en el modelado conexionista del funcionamiento normal de los componentes.

Como fue señalado en el Capítulo 1, existe todo un abanico de métodos de detección de anomalías basados en principios de operación completamente distintos. La elección de uno u otro para el presente sistema se ha visto condicionada por la experiencia adquirida en distintos proyectos de investigación, que han puesto de relieve las siguientes restricciones, que definirán el campo de aplicación del sistema propuesto:

1. No suele existir una base de datos de fallo donde aparezcan representadas todas y cada una de las anomalías que se quieren detectar (*[Muñoz et al., 1995-1]*).

2. El modelado físico de los distintos componentes que integran el proceso (que sería el modelado deseable) suele resultar inviable por dos razones fundamentales: (1) la complejidad de los fenómenos físicos involucrados y (2) la ausencia de una documentación completa del proceso donde aparezcan todos y cada uno de los parámetros requeridos por el modelo físisco (*[Sanz Bobi et al., 1994-1]*).

La primera restricción descarta de forma directa todos los métodos de detección de anomalías que están basados en la caracterización del comportamiento del proceso bajo las distintas situaciones anómalas, ya sea mediante modelos de fallos o sistemas de detección basados en el reconocimiento de patrones de fallo. Estos métodos son sólo viables si se dispone de una completa base de datos de fallo, que podría ser obtenida mediante ensayos (posiblemente destructivos) o mediante simulación.

La única vía que queda libre es pues caracterizar el comportamiento normal de los componentes que integran el proceso, de forma tal que sea posible detectar anomalías cuando se observe alguna desviación significativa del comportamiento actual al comportamiento patrón.

El método más extendido para caracterizar el comportamiento normal de cada componente es el llamado diagnóstico basado en modelos (*[Isermann, 1984], [Gertler, 1988]*). Según este enfoque, cada componente queda descrito por un modelo cuantitativo o cualitativo (*[De Kleer & Brown, 1984], [Forbus, 1984], [Kuipers, 1986]*) que predice la evolución de sus variables de salida en función de la evolución de sus variables de entrada, en condiciones de funcionamiento normal. El

análisis de los residuos obtenidos al comparar las evoluciones medidas y estimadas de las variables de salida permitirá detectar las anomalías que pudieran producirse.

Al enfrentarnos a situaciones que sufren la segunda restricción expuesta anteriormente, la obtención de modelos físicos de funcionamiento normal se ve imposibilitada. Esto no quiere decir que sea imposible introducir ningún tipo de conocimiento a priori de tipo físico a la hora de plantear los modelos, sino que habrá que utilizar modelos de caja negra en los que la selección de las variables de entrada habrá resultado de consideraciones físicas (modelos semi-físicos de caja gris introducidos en el Capítulo 3). El ajuste de estos modelos de caja negra (ver Capítulos 2 y 3) requiere disponer de una base de datos donde quede representado el funcionamiento normal de cada componente bajo todas las condiciones de operación del proceso. Esta base de datos podrá extraerse de la base de datos histórica del proceso (si la hubiere), o bien recogida mediante un plan de medida al comienzo de la implantación del sistema de diagnóstico. El sistema propuesto permitirá además detectar nuevas condiciones de operación que podrán ser añadidas a esta base de datos de funcionamiento normal si no correspondiesen a situaciones anómalas.

El sistema propuesto está pues orientada a resolver la tarea de detección de anomalías en aquellos casos en los que no se dispone de datos de fallo, y en las que el modelado puramente físico resulta inviable. La solución aportada puede encuadrarse dentro de los métodos basados en redundancia analítica (*[Chow & Willsky, 1984], [Frank, 1990], [Patton & Chen, 1991], [Gertler, 1991]*), utilizando como generador de residuos y lógica de detección modelos conexionistas del funcionamiento normal de los componentes (ver Figura 6.1).

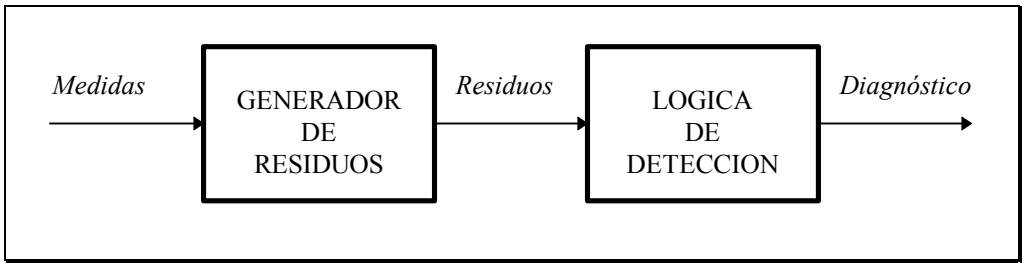

Figura 6.1: Estructura básica de un sistema de detección de anomalías basado en redundancia analítica (tomado de [Chow & Willsky, 1984])

El primer paso a dar en el desarrollo de todo sistema de diagnóstico consiste en representar el proceso industrial como un conjunto jerarquizado de unidades funcionales o componentes. El nivel de anidamiento de estos componentes dependerá en general de la complejidad del proceso y del nivel de detalle al que se quiera llegar con la detección. Cada uno de estos componentes llevará asociado su

propio sistema de detección de anomalías, facilitando de esta forma la tarea de aislamiento de los componentes afectados.

El único requisito imprescindible para la aplicación del sistema de detección de anomalías propuesto, es disponer de un sistema de seguimiento continuo que proporcione de forma periódica las medidas de las variables representativas del estado del componente en cuestión.

La existencia de redundancias implícitas entre estas variables es la base de la metodología. Por variables representativas del estado del componente entenderemos un conjunto de variables $\{X_1, X_2, ..., X_{n'+1}\}$ tal que en condiciones de funcionamiento normal del componente sería posible definir un conjunto de *relaciones de paridad* de la forma:

$$G_i(X_1, X_2, \ldots, X_{n'+1}) = 0; \quad i = 1, \ldots, m$$

Ecuación 6.1

que expresan ligaduras entre distintos retardos de las variables de estado X_j, y tal que en presencia de alguna de las anomalías que se quieren detectar, al menos una de las relaciones anteriores deja de cumplirse.

Las relaciones G_i no tienen por qué conocerse de forma analítica, sino simplemente es necesario conocer de forma cualitativa el conjunto de variables que están relacionadas entre sí.

En el caso de procesos estáticos, las relaciones G_i expresarán ligaduras entre valores de las variables correspondientes al mismo instante temporal, mientras que en el caso de procesos dinámicos, las ligaduras involucrarán distintos retardos de las variables.

El sistema propuesto trata cada una de las relaciones G_i como un proceso dinámico (o estático si fuese el caso), para lo que es necesario seleccionar una de las variables como variable de salida del proceso (sea $X_{n'+1}$), e identificar la estructura del modelo que permitirá comprobar el cumplimiento o el incumplimiento de la relación en cuestión.

Supondremos por lo tanto que cada relación G_i ($i=1,...,m$) puede ser puesta de la forma:

$$d_i[k] = g_i(d_i^{\{k-1\}}, \boldsymbol{u}_i^{\{k\}}, \varepsilon_i^{\{k-1\}}) + \varepsilon_i[k]$$

Ecuación 6.2

donde $d_i[k] \in \Re$ es el valor de la variable ($X_{n'+1}$) tomada como salida de la relación i en el instante k, $d_i^{\{k-1\}}$ es un vector que contiene el valor de la salida d_i en los instantes *(k-1)* y anteriores, $\boldsymbol{u}_i^{\{k\}}$ es un vector que contiene los valores de las n' restantes variables de la relación i que no han sido tomadas como salidas (serán consideradas como variables exógenas del proceso) en los instantes k y anteriores, y $\{\varepsilon_i[k]\}$ es un proceso de ruido blanco que se ha añadido para tener en cuenta fenómenos de tipo aleatorio.

Esta formulación permitirá utilizar como residuos $e_i[k]$ los errores de estimación:

$$e_i[k] = d_i[k] - y_i[k]; \quad i = 1,\ldots,m$$

Ecuación 6.3

siendo $y_i[k]$ la estimación de $d_i[k]$ dada por el modelo no lineal:

$$y_i[k] = f_i(d_i^{\{k-1\}}, \boldsymbol{u}_i^{\{k\}}, e_i^{\{k-1\}})$$

Ecuación 6.4

Llamaremos *modelo de funcionamiento normal* del componente al conjunto de estimadores definidos por la Ecuación 6.4 para $i=1,...,m$.

Supongamos que la aparición de una anomalía determinada en el componente en cuestión provoca el incumplimiento de la relación G_i. Este incumplimento se traducirá en un incremento significativo del valor absoluto del residuo e_i, cuya evolución temporal comenzará a presentar cierto comportamiento determinista impropio de un ruido blanco aleatorio. Las dos misiones fundamentales del sistema de detección de anomalías serán por tanto la de generar los residuos e_i (i=1,...,m) y la de activar el estado de anomalía del componente en cuestión, para lo que será necesario especificar lo que se entiende por valor significativamente elevado del residuo.

6.2 Estructura del sistema de detección de anomalías

6.2.1 Descripción general

Supongamos que el modelo de funcionamiento normal del componente en cuestión consta tan sólo de una única salida (el estado del componente queda determinado por una única relación de paridad). En este caso, el modelo de funcionamiento normal tendrá la expresión general:

$$y[k] = f(d^{\{k-1\}}, \boldsymbol{u}^{\{k\}}, e^{\{k-1\}}) = f(\boldsymbol{x}[k])$$

Ecuación 6.5

donde en el vector de entradas en el instante k del aproximador funcional ($\boldsymbol{x}[k] \in \Re^n$) se han incluido los retardos apropiados de las distintas variables que puedan conformar el modelo identificado.

A partir de este modelo y de la medida actual de la salida ($d[k]$) se obtendrá el valor actual del residuo $e[k]$ dado por:

$$e[k] = d[k] - y[k]$$

Ecuación 6.6

Cuando el valor absoluto de este residuo sea significativamente elevado, siendo fiable la predicción $y[k]$, el sistema emitirá el diagnóstico de anomalía en el componente correspondiente.

De la anterior afirmación quedan dos conceptos por definir:

- ¿Cuándo consideraremos fiable la predicción $y[k]$?
- ¿Qué entendemos por residuo significativamente elevado?

Para responder a la primera pregunta es necesario recordar que el modelo de funcionamiento normal $y[k]=f(\boldsymbol{x}[k])$ es un modelo de caja negra que resulta del ajuste de un aproximador funcional, a partir de las muestras de entrada/salida contenidas en el conjunto de entrenamiento. La estimación será por lo tanto "fiable" en aquella región del espacio de entrada $\boldsymbol{X} \subset \Re^n$ que queda representada por las muestras del conjunto de entrenamiento. Esta región del espacio de entrada del

modelo será llamada *región de confianza del dominio del modelo*, o simplemente *región de confianza*.

El método propuesto para delimitar la región de confianza del modelo de funcionamiento normal consiste en ajustar una red PRBFN para obtener una estimación de la función de densidad probabilista (fdp) según la cual se distribuye el vector de entradas $\boldsymbol{x}$ en el conjunto de entrenamiento. Sea $p_x[k]$ la fdp estimada para el vector $\boldsymbol{x}[k]$. Cuanto mayor sea $p_x[k]$, mejor representado habrá quedado el entorno del punto $\boldsymbol{x}[k]$ en el conjunto de entrenamiento, y mejor caracterizado habrá quedado el residuo correspondiente $e[k]$ en condiciones de funcionamiento normal en ese mismo entorno. Cuando $p_x[k]$ quede por debajo de una *cota de extrapolación* preestablecida, sea $p_{mín}$, daremos por desconocida esa región del espacio de entrada y no podremos utilizar el residuo $e[k]$ para activar el estado de anomalía del componente. Sí podremos sin embargo advertir de la situación detectada, para que el sistema de identificación de anomalías analice las causas de esta situación.

Es importante hacer notar que si el vector de entradas $\boldsymbol{x}$ sólo contiene retardos de las variables exógenas del proceso dinámico (variables $\boldsymbol{u}$), la salida de la región de confianza puede corresponder a la entrada del sistema en un nuevo punto de operación hasta ahora desconocido, que en el caso de corresponder a situación de funcionamiento normal, debería ser incluido en la base de datos. La otra posibilidad es que una de las variables exógenas del modelo del componente en cuestión se esté saliendo de su rango de funcionamiento normal por funcionamiento anómalo de otro componente.

Si por el contrario el modelo contiene alguna componente autorregresiva o de media móvil, la salida del vector $\boldsymbol{x}$ de su región de confianza puede ser además debida a un valor anómalo de la salida del proceso en instantes anteriores, causado por un fallo en el propio componente. Bajo estas condiciones, la región de confianza del modelo puede ser interpretada como la región de funcionamiento normal del componente, donde se establecen los rangos de funcionamiento normal de las componentes no exógenas del vector $\boldsymbol{x}$, bajo las distintas condiciones de operación expresadas por las componentes exógenas de $\boldsymbol{x}$. Estas disquisiciones serán ampliadas en el apartado 6.3.

La delimitación de la región de confianza tal como se ha planteado realiza una bipartición del espacio de entrada en "conocido/desconocido". Esta decisión tan rígida podría relajarse en la frontera de la región de confianza sin más que definir una transición continua de lo "conocido" a lo "desconocido" en función de la fdp estimada del vector de entradas $p(\boldsymbol{x})$. Esta definición podría realizarse en términos de lógica borrosa definiendo el conjunto borroso "conocido" a partir de la estimación $p(\boldsymbol{x})$ (*[Zadeh, 1965], [Zadeh, 1983], [Yager & Zadeh, 1992]*). De esta forma los vectores de entrada que generasen una $p(\boldsymbol{x})$ alta serían considerados como "conocidos" en grado 1, grado que iría disminuyendo hasta 0 a medida que

disminuyese $p(\boldsymbol{x})$. Es importante aclarar en este punto que un grado de pertenencia 1 al conjunto borroso "conocido" nada tiene que ver con una probabilidad 1 del vector de entradas $\boldsymbol{x}$, de igual forma que un grado de pertenencia nulo al conjunto borroso "conocido" no implica probabilidad nula.

Para dar respuesta a la segunda pregunta planteada, se propone en esta tesis obtener una estimación de la desviación típica del residuo $e[k]$, sea $s_e[k]$, en función del vector de entradas $\boldsymbol{x}[k]$. Suponiendo una distribución de $e[k]$ localmente normal[i] (en condiciones de funcionamiento normal), diremos que el residuo es significativamente elevado, con un nivel de confianza del 95%, si su valor absoluto sobrepasa la *cota máxima del residuo* definida por:

$$e_{max}[k] = 2 \cdot s_e[k]$$

Ecuación 6.7

Este criterio está tomado de la teoría de control de procesos (*[Duncan, 1974], [Montgomery, 1985]*), donde suele suponerse que la variable bajo estudio tiene una distribución normal de media y varianza constantes. Al utilizar en nuestro caso, como especial aportación, una estimación de la desviación típica del residuo en función del vector de entradas, se establece de forma automática un umbral distinto para cada condición de operación. De esta forma el sistema de detección de anomalías se ajusta de forma automática a las características propias de cada punto de operación.

Como ocurría con el modelo de funcionamiento normal, la estimación de la desviación típica residual $s_e[k]$ sólo será fiable en la región de confianza del modelo, ya que fuera de ella se desconoce en principio la distribución del residuo $e[k]$.

Para aumentar la robustez de la detección es necesario tener en cuenta la evolución temporal del residuo de tal forma que se identifiquen comportamientos deterministas. Para ello se puede filtrar el criterio anterior utilizando una media móvil del error de estimación en lugar del valor instantáneo, o imponer como condición de detección el que se cumpla el criterio anterior al menos dos veces en las tres últimas muestras (*[Duncan, 1974], [Montgomery, 1985]*). Esta última estrategia será el procedimiento utilizado y justifica el haber tomado un nivel de confianza del 95% en lugar del 99%.

El sistema de detección de anomalías de cada componente supervisado tomará pues la estructura representada en la Figura 6.2:

[i] Entenderemos por "distribución localmente normal" del residuo, una distribución normal de media y varianza constantes en el entorno de cada punto $\boldsymbol{x}$, pudiendo variar el valor de estos momentos de un punto a otro.

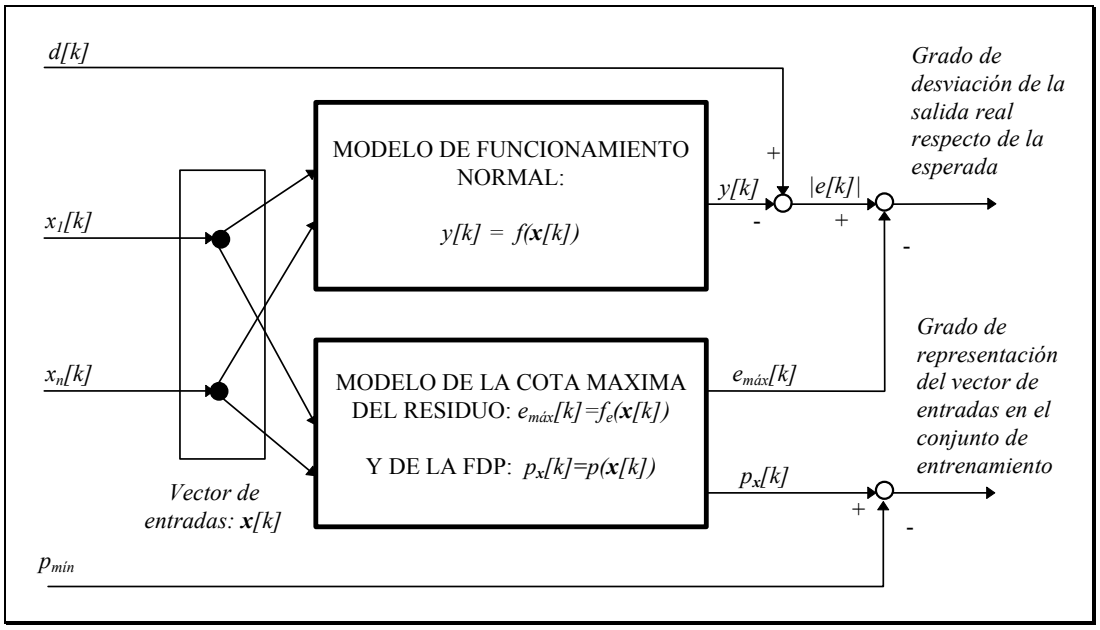

Figura 6.2: Estructura del sistema de detección de anomalías

donde:

- $d[k]$ es la medida en el instante k de la variable tomada como salida del modelo de funcionamiento normal.
- $\mathbf{x}[k]=[x_1[k], ...,x_n[k]]^T$ es el vector de regresores (o entradas del aproximador funcional) del modelo de funcionamiento normal en el instante k.
- $y[k]=f(\mathbf{x}[k])$ es la salida estimada por el modelo de funcionamiento normal en el instante k.
- $p_{\mathbf{x}}[k]=p(\mathbf{x}[k])$ es la fdp estimada del vector de entradas $\mathbf{x}[k]$.
- $e_{máx}[k]=f_e(\mathbf{x}[k])$ es la cota máxima del residuo en el instante k.
- $p_{mín}$ es la cota de extrapolación.
- $e[k]=d[k]-y[k]$ es el valor del residuo o error de estimación en el instante k.

A continuación se presenta una descripción más exhaustiva de cada uno de los componentes de este sistema.

6.2.2 Modelo de funcionamiento normal

Una vez descompuesto el proceso en un conjunto jerarquizado de componentes (físicos o inmateriales), cada componente a diagnosticar será tratado por separado para la tarea de detección de anomalías (no para la tarea de aislamiento). El siguiente paso a dar a la hora de desarrollar el sistema de detección de anomalías de cada componente, será el de identificar y ajustar su modelo de funcionamiento normal.

El conocimiento experto permitirá determinar un punto de partida para el número de relaciones de paridad necesarias para caracterizar el comportamiento normal del componente bajo estudio, y para el conjunto de variables que interviene en cada una de ellas. Esta determinación se verá gravemente limitada en la práctica por el conjunto de variables medidas o medibles del proceso. Una vez determinado para cada relación de paridad un conjunto candidato de variables ligadas, será necesario seleccionar una de las variables como variable de salida del proceso correspondiente. Esta decisión estará nuevamente basada en conocimiento experto.

Supongamos de ahora en adelante que el componente en cuestión lleva asociada una única relación de paridad. La extensión al caso de múltiples relaciones es inmediato, ya que cada una de ellas puede ser tratada por separado.

Llegados a este punto en el que ya se ha determinado la variable de salida del proceso a modelar y el conjunto de variables exógenas disponibles, la obtención del modelo de funcionamiento normal se reduce al procedimiento de modelado de procesos dinámicos no lineales con aproximadores funcionales descrito en el Capítulo 3. Los aproximadores funcionales que se proponen en esta tesis son redes neuronales artificiales supervisadas, como el Perceptrón Multicapa (Capítulo 4) y la red PRBFN (Capítulo 5).

Un punto fundamental a la hora de obtener el modelo de funcionamiento normal es la selección de los datos utilizados para el ajuste del aproximador funcional. El primer paso a dar en esta selección será extraer de la base de datos histórica disponible, los periodos de tiempo correspondientes al funcionamiento normal del componente. Esta selección deberá ser supervisada por el personal de operación del proceso, ya que la inclusión de datos correspondientes a algún tipo de funcionamiento anómalo insensibilizaría el sistema de detección frente a este tipo de comportamientos.

Durante el procedimiento de identificación de la estructura del modelo de funcionamiento normal, será necesario ensayar distintos tipos de modelos, con distintos conjuntos de regresores o entradas (modelo estático, NFIR, NARMAX, ...). Dado un conjunto de regresores determinado (y por tanto un espacio de entrada determinado), es posible muestrear la base de datos de funcionamiento normal de

forma tal que se obtenga un subconjunto de datos repartidos homogéneamente en el espacio de entrada. Esta tarea es fundamental en aquellos casos en los que el proceso presenta distintos puntos de operación en los que el sistema suele mantenerse en régimen permanente (como por ejemplo la baja y plena carga en una central eléctrica), y que quedan "sobrerepresentados" en la base de datos disponible, en comparación con los datos correspondientes a la transición de un punto de operación a otro, que contienen un alto grado de información sobre la dinámica del sistema.

Para realizar esta selección se propone en esta tesis la utilización de los "árboles de selección de datos", que es una estructura de selección y almacenamiento de información que ha sido especialmente diseñada para este fin. El objetivo de los árboles de selección de datos es proporcionar un método eficaz de mantenimiento dinámico de una base de datos de tamaño predefinido, formada por N datos representativos del espacio muestral. Es en definitiva una herramienta de *muestreo selectivo* que permitirá seleccionar de forma secuencial un conjunto de datos representativo de la población, entendiendo por representativo el que cubran de forma uniforme u homogénea el espacio muestral. La eficacia de este método radica en su capacidad para revisar la base de datos sin necesidad de utilizar datos no contenidos en ella, y en la rapidez de posicionamiento de un nuevo dato en la base. Por mantenimiento dinámico de la base de datos se entiende la capacidad de ir renovando los datos seleccionados con datos más recientes, según van siendo leídos.

Estas características permiten utilizar los árboles de selección de datos para la extracción de los conjuntos de entrenamiento y test de los modelos a ajustar. Esta misma estructura permitirá ir reajustando los modelos con datos de funcionamiento normal más recientes que puedan además contener la descripción de nuevos puntos de operación, dotando de esta forma al sistema de detección de anomalías de la capacidad de adaptación a nuevos modos de funcionamiento y a los naturales procesos de envejecimiento de los componentes.

Una vez identificado, ajustado y validado el modelo de funcionamiento normal, el sistema de detección de anomalías será ya capaz de generar los residuos que serán analizados posteriormente.

6.2.3 Modelo de la fdp del vector de entradas

Una vez obtenido el modelo de funcionamiento normal del componente, retomaremos el conjunto de entrenamiento utilizado para su ajuste para obtener una estimación de la función de densidad probabilista (fdp) según la cual se distribuye el vector de entradas $\boldsymbol{x} \in \Re^n$ en el mencionado conjunto. Este modelo permitirá establecer la región de confianza del modelo de funcionamiento normal, que equivale

a la región del espacio de entrada en la que nuestro modelo es válido, entendiendo por válido la capacidad de poder caracterizar estadísticamente los residuos de estimación en esa región.

Como modelo de fdp se propone utilizar una red neuronal PRBFN tal como se describe en el cuarto apartado del Capítulo 5. Si el modelo de funcionamiento normal hubiese sido realizado utilizando una red PRBFN como aproximador funcional, cabría la posibilidad de utilizar esta misma estructura como estimador de la fdp, si para el ajuste de los parámetros se utilizase la función mixta de error definida por la Ecuación 5.64. No obstante se recomienda utilizar una nueva red PRBFN para este propósito, por dos razones fundamentales:

- La inclusión del término de máxima verosimilitud logarítmica en la función de error puede reducir la precisión del modelo de funcionamiento normal, perjudicando la tarea de detección de anomalías que será tanto más incipiente cuanto más preciso sea el modelo.

- Como veremos más adelante, esta segunda red utilizada para estimar la fdp $p(\mathbf{x})$ será utilizada simultáneamente para estimar la cota máxima de los residuos.

En caso de utilizar una red PRBFN del tipo I, la estimación de la fdp del vector de entradas admitiría como expresión:

$$\begin{cases} p_x[k] = p(\mathbf{x}[k]) = \dfrac{1}{\pi^{n/2}} \dfrac{1}{h} \displaystyle\sum_{i=1}^{h} a_i[k] \\ a_i[k] = |\mu_i|^n \cdot exp\left(-\mu_i^2 \cdot (\mathbf{x}[k] - \mathbf{r}_i)^T \cdot (\mathbf{x}[k] - \mathbf{r}_i)\right) \end{cases}$$

Ecuación 6.8

siendo h el número de unidades radiales de la red (ver Capítulo 5).

Partiendo del conjunto de entrenamiento utilizado para ajustar el modelo de funcionamiento normal ($S_{entr}=\{(\mathbf{x}[k],d[k])$ con $\mathbf{x}[k] \in \Re^n,\ d[k] \in \Re,\ k=1,...,N\}$), los centros $\mathbf{r}_i$ y factores de escala μ_i de la red se obtendrían maximizando la verosimilitud logarítmica:

$$V = \frac{1}{N} \sum_{k=1}^{N} log(p_x[k])$$

Ecuación 6.9

Una vez ajustados estos parámetros, estaremos en condiciones de establecer el valor de la cota de extrapolación $p_{mín}$. El método propuesto para este fin es el de seleccionar un grado de confianza α (en torno al 98%) y tomar como cota de extrapolación el valor $p_{mín}$ tal que $p_x[k] > p_{mín}$ en el 98% de los datos de entrenamiento (centil del $(100-\alpha\%)$ de las fdp estimadas para el conjunto de entrenamiento).

6.2.4 Modelo de la cota máxima de los residuos

Una vez obtenidos el modelo de funcionamiento normal del componente y la región de confianza del mismo, es posible caracterizar el comportamiento de los residuos en condiciones de funcionamiento normal.

Esta caracterización será válida en el interior de la región de confianza, ya se que se realizará a partir del mismo conjunto de entrenamiento que se utilizó para el ajuste del modelo de funcionamiento normal.

La solución propuesta está inspirada en *[Leonard et al., 1992]* y consiste en obtener una estimación de la desviación típica $s_e[k]$ del residuo $e[k]$ en función del vector de entradas $\boldsymbol{x}[k]$. Suponiendo una distribución localmente normal de los residuos, tomaremos como cota máxima del residuo el valor $2s_e[k]$, correspondiente a un nivel de confianza del 95%. Esta dependencia de la varianza del residuo con el vector de entradas permite tener en cuenta diferencias significativas del proceso subyacente en los distintos puntos de operación. Esta situación suele darse por ejemplo cuando el componente ha sido diseñado para trabajar de forma estable en un determinado punto de operación, y luego es utilizado en régimen de funcionamiento normal en otros puntos de operación muy distintos que pueden resultar más inestables.

Para estimar la varianza residual $s_e^2[k]$, retomaremos la red PRBFN utilizada para estimar la fdp $p_x[k]$. De la Ecuación 6.8 se desprende que la fdp $p_x[k]=p(\boldsymbol{x}[k])$ se obtiene como suma normalizada de h distribuciones normales centradas en los representantes de las unidades radiales de la red PRBFN:

$$\begin{cases} p_{\boldsymbol{x}}[k] = p(\boldsymbol{x}[k]) = \dfrac{1}{h}\displaystyle\sum_{i=1}^{h} p_{\boldsymbol{x},i}[k] \\ p_{\boldsymbol{x},i}[k] = \dfrac{1}{\pi^{n/2}}\, a_i[k] \end{cases}$$

Ecuación 6.10

de tal forma que podemos definir la varianza residual local en la unidad *i* ($s^2_{e,i}$) como la esperanza matemática del cuadrado del residuo dada la distribución $p_{\mathbf{x},i}$. Estas esperanzas pueden ser estimadas a partir de los residuos obtenidos con el conjunto de entrenamiento de la forma:

$$s^2_{e,i} = \frac{\sum_{k=1}^{N} p_{\mathbf{x},i}[k] \cdot e^2[k]}{\sum_{k=1}^{N} p_{\mathbf{x},i}[k]} = \frac{\sum_{k=1}^{N} a_i[k] \cdot e^2[k]}{\sum_{k=1}^{N} a_i[k]}$$

Ecuación 6.11

Esta estimación de la varianza residual local en la unidad *i* puede ser interpretada también como una media ponderada de los errores cuadráticos de estimación del conjunto de entrenamiento, donde el factor de ponderación asociado al ejemplo *k* es una medida del grado de pertenencia del vector de entradas del ejemplo *k* a la unidad radial *i*.

La estimación de la varianza residual se obtendrá entonces por regresión generalizada (ver el apartado 3.1 del Capítulo 5) de la forma:

$$s^2_e[k] = \frac{\sum_{i=1}^{h} a_i[k] \cdot s^2_{e,i}}{\sum_{i=1}^{h} a_i[k]}$$

Ecuación 6.12

De las dos ecuaciones anteriores se desprende que la varianza residual estimada puede ser calculada utilizando la misma red PRBFN utilizada para estimar la fdp del vector de entradas. Para ello basta con igualar los hasta ahora inutilizados pesos de la capa de salida de la mencionada red (parámetros v_i) a las varianzas locales:

$$v_i = s^2_{e,i}$$

Ecuación 6.13

de tal forma que la misma estructura será capaz de generar como salidas la estimación de la función de densidad del vector de entradas y la varianza residual estimada:

$$\begin{cases} p[k] = \dfrac{1}{\pi^{n/2}} \dfrac{1}{h} \displaystyle\sum_{i=1}^{h} a_i[k] \\ s_e^2[k] = \dfrac{\displaystyle\sum_{i=1}^{h} a_i[k] \cdot v_i}{\displaystyle\sum_{i=1}^{h} a_i[k]} \end{cases}$$

Ecuación 6.14

Como cota máxima del residuo admisible en condiciones de funcionamiento normal tomaremos el valor correspondiente a un nivel de confianza del 95%:

$$e_{max} = 2 \cdot s_e = 2\sqrt{s_e^2}$$

Ecuación 6.15

y llamaremos *banda de funcionamiento normal* del componente al intervalo de valores de salida definido por : $y[k] \pm e_{máx}[k]$.

A modo de ilustración retomemos el ejemplo introducido en el apartado 5.4.3 del Capítulo 5 y añadamos a la salida del proceso una señal de ruido $\varepsilon[k]$ de la forma:

$$d[k] = 2x^3[k] - x[k] + \varepsilon[k]$$

Ecuación 6.16

Generemos un conjunto de entrenamiento tomando muestras uniformemente repartidas en el intervalo *[-1,1]* ($x[k]=-1+0.01(k-1)$ con $k=1,...,201$) y hagamos que el ruido tenga una varianza creciente con x:

$$\varepsilon[k] = \left(0.045 + 0.04x[k]\right) \cdot \eta[k] \quad con \quad \eta[k] \in N(0,1)$$

Ecuación 6.17

Ajustemos una primera red PRBFN con 6 unidades radiales al conjunto de datos así generado *({(x[k],d[k]), k=1,..,201}* para obtener el modelo $y[k]=f(x[k])$.

Ajustemos a continuación los centros y factores de escala de una segunda red PRBFN con 6 unidades radiales para obtener una estimación de la fdp $p_x[k]=p(x[k])$.

Con la serie de residuos *{e[k]=d[k]-y[k]}* calculemos el valor de los pesos de la capa de salida de esta segunda red según la Ecuación 6.13 y la Ecuación 6.11. De esta forma ya tendremos ajustados el estimador de la salida del proceso, el estimador de la fdp del vector de entradas, y el estimador de la varianza residual.

En la Figura 6.3.a aparece representada la estimación *y[k]* junto con las bandas de funcionamiento normal correspondientes a unos niveles de confianza del 95% y 99% de la estimación, superpuestas a las muestras del conjunto de entrenamiento. Estas bandas han sido obtenidas a partir de la estimación de la varianza residual s_e^2 como $(y[k] \pm 3s_e[k])$ la del 99%, e $(y[k] \pm 2s_e[k])$ la del 95%.

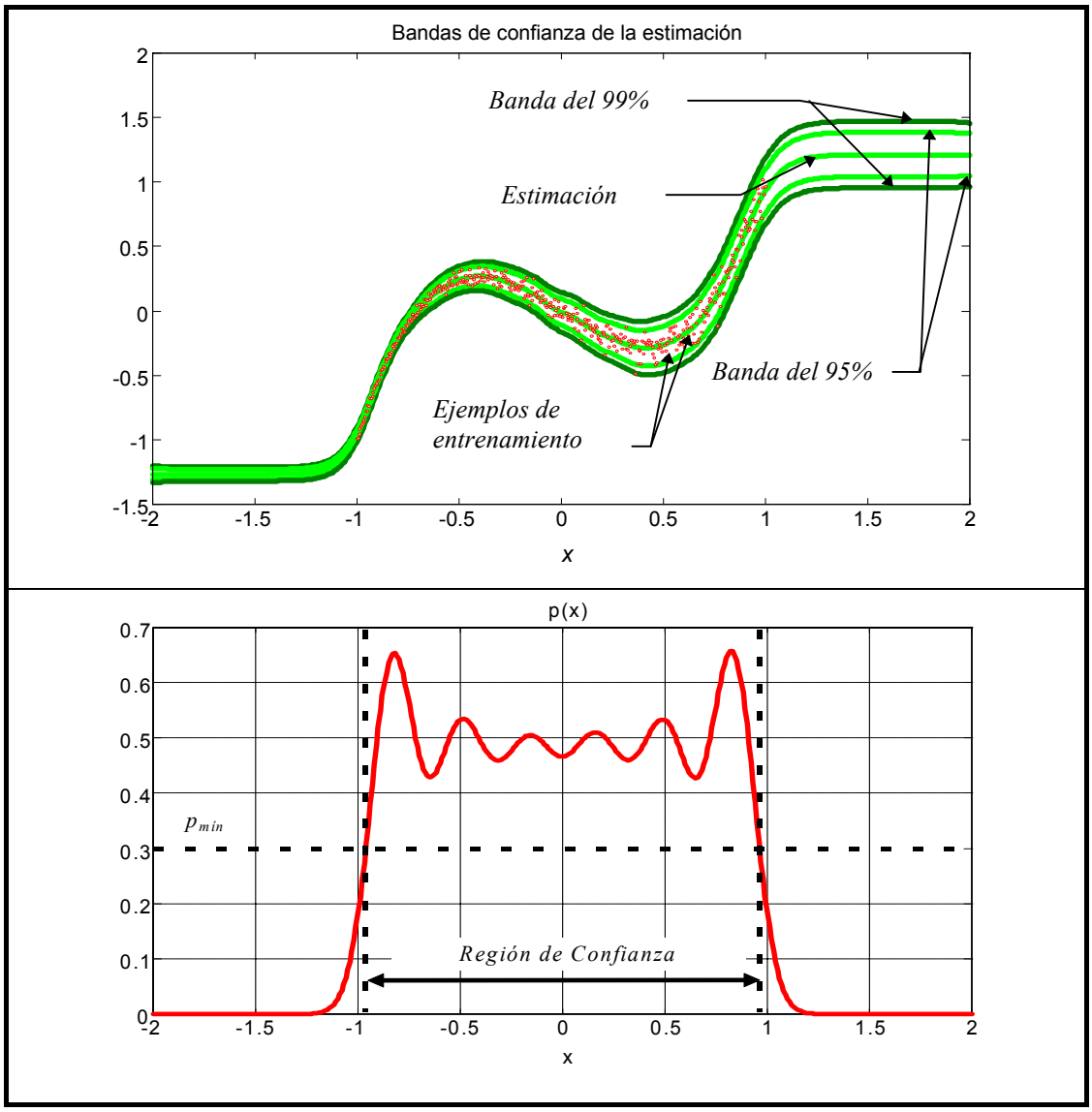

Figura 6.3: (a) Bandas de confianza de la estimación de una cúbica con ruido de varianza creciente superpuesto y (b) Estimación de la fdp de x

Como era de esperar, las bandas de funcionamiento normal de la estimación se ajustan fielmente a la creciente dispersión de los datos del conjunto de entrenamiento, caracterizando de forma precisa el comportamiento esperado de los residuos de la estimación.

En la Figura 6.3.b aparece la estimación de la fdp *p(x)* obtenida con la segunda red PRBFN. Esta estimación permite establecer la región de confianza de los modelos ajustados. Para ello basta con calcular el centil del *2%* de la distribución obtenida al evaluar la fdp *p(x)* sobre el conjunto de entrenamiento, obteniendo una cota de extrapolación en este caso igual a $p_{mín}=0.3$. De esta forma se obtiene como región de confianza el intervalo: $x \in [-0.97, 0.97]$.

6.3 Lógica de detección de anomalías

La misión fundamental del sistema de detección de anomalías, como parte integrante del sistema de diagnóstico, es la de "despertar" al sistema de identificación para que analice las causas de las anomalías que ha detectado.

El sistema de detección presentado en este capítulo basa su decisión en dos criterios complementarios que permiten distinguir toda una serie de situaciones. Estos criterios están basados en el grado de representación del vector de entradas en el conjunto de entrenamiento y en el grado de desviación de la salida real respecto de la esperada.

Según el primer criterio, consideraremos desconocido todo vector de entradas $\boldsymbol{x}[k]$ que genere una fdp $p_x[k]$ inferior a la cota de extrapolación $p_{mín}$.

Según el segundo criterio, consideraremos anómalo todo valor de la salida $d[k]$ que salga fuera de la banda de funcionamiento normal ($y[k] \pm e_{máx}[k]$) al menos dos veces en las tres últimas muestras ($y[k] \pm e_{máx}[k]$).

De esta forma podremos asegurar con un elevado grado de certeza que el funcionamiento del componente en cuestión es normal cuando se cumplan de forma simultánea las condiciones de vector de entradas conocido y de salida normal.

De igual forma podremos asegurar con un elevado grado de certeza que el componente tiene un comportamiento anómalo cuando siendo el vector de entradas conocido, su salida es anómala.

La interpretación de las dos situaciones restantes, correspondientes a un vector de entradas desconocido, es algo más complicada y depende de la constitución interna del vector de entradas $\boldsymbol{x}$. En cualquiera de estos casos el sistema de detección deberá avisar al sistema de identificación que será el encargado de analizar las causas de la situación detectada en base a la información que le llegue de otros componentes del proceso y de su evolución temporal.

Consideremos en primer lugar aquellos casos en los que el vector de entradas no tiene componentes autorregresivas ni de media móvil (sólo valores de las entradas exógenas). Bajo estas circunstancias, la salida del vector de entradas de la región de confianza puede interpretarse como la entrada del proceso en unas nuevas condiciones de operación hasta ahora desconocidas. Estas condiciones podrían corresponder a un nuevo punto de operación normal del proceso, en cuyo caso deberían ser incluidas en la base de datos de funcionamiento normal para que se las

aprenda el modelo de funcionamiento normal. La otra posibilidad es que estas nuevas condiciones de operación tengan como origen el fallo de otro componente. En este caso los datos no deberían ser incluidos en la base de datos de funcionamiento normal para poder seguir detectando esta situación. En ambos casos el sistema de detección será incapaz de determinar con un elevado grado de certeza si el componente está funcionando de forma correcta o no, aunque la permanencia de la salida real en la banda de funcionamiento normal del modelo apoyará la hipótesis de funcionamiento normal.

Consideremos ahora aquellos casos en los que el vector de entradas $\boldsymbol{x}$ tiene además componentes autorregresivas o de media móvil. Bajo estas circunstancias, además de la casuística presentada anteriormente, cabe la posibilidad de que la salida del vector de entradas de la región de confianza sea debida a un fallo interno del propio componente. De esta forma las situaciones que pueden dar pie a un vector de entradas desconocido en un modelo con entradas autorregresivas o de media móvil son:

- Nueva condición de operación desconocida pero correspondiente a funcionamiento normal: estas situaciones de normalidad que no han sido contempladas en el conjunto de entrenamiento han de incluirse en la base de datos de funcionamiento normal para su posterior inclusión en el modelo. A medida que todas las condiciones de funcionamiento normal vayan quedado representadas en el conjunto de entrenamiento, este tipo de situaciones tenderá a desaparecer.

- Fallo en un componente externo que se transmite por una entrada exógena: el fallo de un componente externo cuya salida forma parte de las entradas exógenas del modelo de funcionamiento normal del componente en cuestión puede provocar la salida del vector de entradas de la región de confianza. Estos datos no han de ser incluidos en la base de datos de funcionamiento normal.

- Fallo en el propio componente que se transmite por una entrada autorregresiva o de media móvil: de forma análoga al caso anterior, el fallo en el componente en cuestión puede provocar la salida del vector de entradas de la región de confianza. Estos datos tampoco han de ser incluidos en la base de datos de funcionamiento normal, ya que su inclusión insensibilizaría al sistema frente a este tipo de anomalías.

Toda la casuística que se ha presentado suele ignorarse en los métodos clásicos de detección de anomalías basados en modelos por no disponer de una medida del grado de representación del vector de entradas en el conjunto de datos utilizado para el ajuste del modelo.

De la misma forma, las bandas de funcionamiento normal que se obtienen con los métodos clásicos suelen ser mucho más anchas y por lo tanto menos sensibles por dos razones fundamentales: (1) los modelos utilizados suelen ser menos precisos y (2) la cota máxima de los residuos suele ser constante e independiente del punto de trabajo.

Las principales aportaciones del método de detección aquí propuesto frente a los métodos clásicos son pues:

- Capacidad de predicción del comportamiento de sistemas complejos (no lineales) gracias a la flexibilidad de los modelos conexionistas.

- Capacidad de estimación local de la cota de los residuos en función del punto de trabajo.

- Capacidad de cuantificación de la fiabilidad de la predicción y de detección de entradas desconocidas mediante la estimación de la fdp del vector de entradas.

Estas características hacen de la aplicación de este sistema un método muy preciso y autoexplicativo para el modelado de un comportamiento y por tanto con gran capacidad de detección.

6.4 Adaptación del sistema de detección de anomalías

Una característica importante de todo sistema de diagnóstico que pretenda ser útil a la industria, es su capacidad de adaptación a nuevas condiciones de operación y a los inevitables procesos de envejecimiento de los componentes involucrados.

En el caso concreto de la detección de anomalías basada en modelos de funcionamiento normal, esta capacidad puede conseguirse sin más que adoptar una estrategia de reajuste periódico de los modelos utilizados. Para ello es necesario ir renovando de forma continua la base de datos de funcionamiento normal utilizada para el reajuste de los modelos, e informar al usuario de los reajustes realizados. Esta puesta al día ha de realizarse prestando una especial atención a la no inclusión de datos correspondientes a situaciones anómalas.

La solución propuesta en esta tesis para la adaptación del sistema de detección de anomalías se basa en la utilización de los árboles de selección de datos (Apéndice A). Esta estructura permite ir actualizando de forma secuencial una base de datos de tamaño predefinido atendiendo a dos criterios de optimalidad simultáneos: por un lado el espacio muestral ha de quedar representado de la forma más homogénea posible, y por otro ha de darse prioridad a las muestras más recientes. De esta forma se consigue que todos los puntos de operación por los que ha pasado el sistema queden representados en la base de datos y que esta representación sea lo más actual posible.

Para evitar la inclusión de datos correspondientes a situaciones anómalas es indispensable que la renovación de la base de datos se realize tras haber rechazado los correspondientes a situaciones anómalas. El propio sistema de detección de anomalías será capaz de rechazar aquellas muestras que hayan activado el estado de anomalía (muestras cuyo vector de entradas pertenece a la región de confianza pero que generan un residuo significativamente elevado). Quedan pendientes sin embargo las muestras cuyos vectores de entrada salen fuera de la región de confianza. El sistema de detección de anomalías será incapaz de discernir si estas muestras corresponden a nuevos puntos de operación considerados normales, en cuyo caso es de vital importancia el incluirlas, o si por el contrario se trata de una situación anómala. Esta determinación puede realizarse *a posteriori* por el operario o por el sistema experto de diagnóstico una vez que se hayan analizado las consecuencias. Para ello será necesario dotar al sistema de gestión de datos de una agenda donde quedarán almacenados los datos diarios de cada componente, hasta que se les dé el visto bueno para pasar por los árboles de selección de datos. Esta intervención externa se irá haciendo menos necesaria a medida que el propio sistema de detección de anomalías vaya conociendo nuevos puntos de operación.

6.5 Aplicación a la detección de anomalías en el condensador de una Central Térmica[i]

6.5.1 Introducción

El proceso a supervisar en este caso es el funcionamiento del condensador de una central térmica. El condensador es el componente del circuito de agua-vapor de la central térmica encargado de condensar el vapor de salida de las turbinas de baja presión para convertirlo en agua de alimentación (*[Ramírez Vázquez, 1980] [Wilson et al., 1991]*). Este proceso de condensación es un proceso de intercambio de calor entre el vapor a condensar y el agua de circulación que se enfría en la torre de refrigeración (ver Figura 6.4).

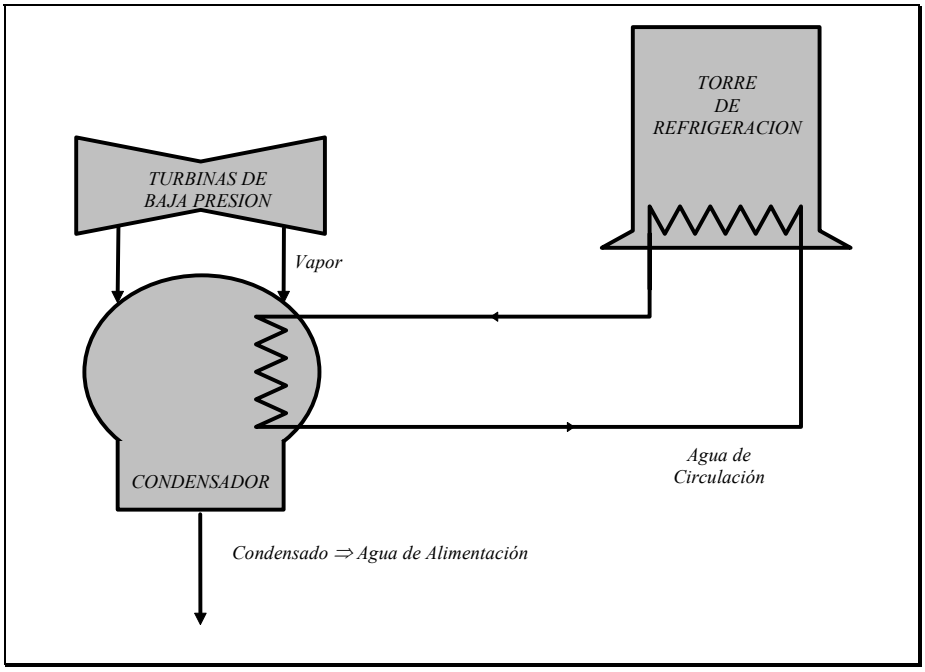

Figura 6.4: Esquema simplificado del proceso de condensación

Además de recuperar el vapor condensado para utilizarlo como agua de alimentación de las calderas, el condensador es un elemento clave a la hora de maximizar el rendimiento termodinámico del ciclo. Gracias al vacío creado en su interior permite llevar hasta el máximo la expansión de vapor en las turbinas, aumentando la caída térmica en la expansión y disminuyendo la presión de escape del vapor.

[i] Parte de este trabajo puede encontrarse publicado en la referencia *[Sanz Bobi et al., 1994-1]*.

6.5.2 Selección de las variables del modelo de funcionamiento normal y recogida de datos

La presión de vacío resulta ser pues un indicador fundamental del rendimiento del condensador y será la variable que tomaremos como variable de salida del modelo.

Como posibles variables explicativas disponemos en este caso del siguiente conjunto de medidas provenientes de los sensores instalados:

1. Potencia generada en la central
2. Caudal de condensado
3. Presión de vacío del condensador
4. Nivel del pozo caliente del condensador
5. Presión de descarga de las bombas de agua de circulación
6. Temperatura de entrada del agua de circulación
7. Temperatura de salida del agua de circulación

Bajo condiciones normales de funcionamiento, las estrategias de operación de la central establecen una serie de restricciones y ligaduras entre las variables de tal forma que para un flujo constante de agua de circulación, la presión de vacío del condensador es sólo función del punto de operación de la central (potencia generada) y de la temperatura de entrada del agua de circulación (*[Wilson et al., 1991]*).

Es importante aclarar que el modelo así resultante:

$$y = f(PG , TAC)$$

Ecuación 6.18

siendo:
- y: estimación de la presión de vacío (PV) del condensador (mbar)
- PG: potencia generada (MW)
- TAC: temperatura de entrada del agua de circulación (ºC)

no es un modelo físico que relacione las características termodinámicas de los elementos involucrados, sino un modelo del funcionamiento en lazo cerrado del condensador, que lleva implícito las acciones de control determinadas por las estrategias de operación. Utilizaremos este modelo como modelo de funcionamiento normal del condensador.

Para seleccionar los conjuntos de entrenamiento y test se han escogido 9 días de funcionamiento normal y se han dedicado 6 para entrenamiento y 3 para test. Como conjunto de validación se han ensayado los modelos sobre otros dos días de

funcionamiento normal, y como veremos posteriormente, sobre dos días de funcionamiento anómalo. Los datos recogidos se muestran a continuación:

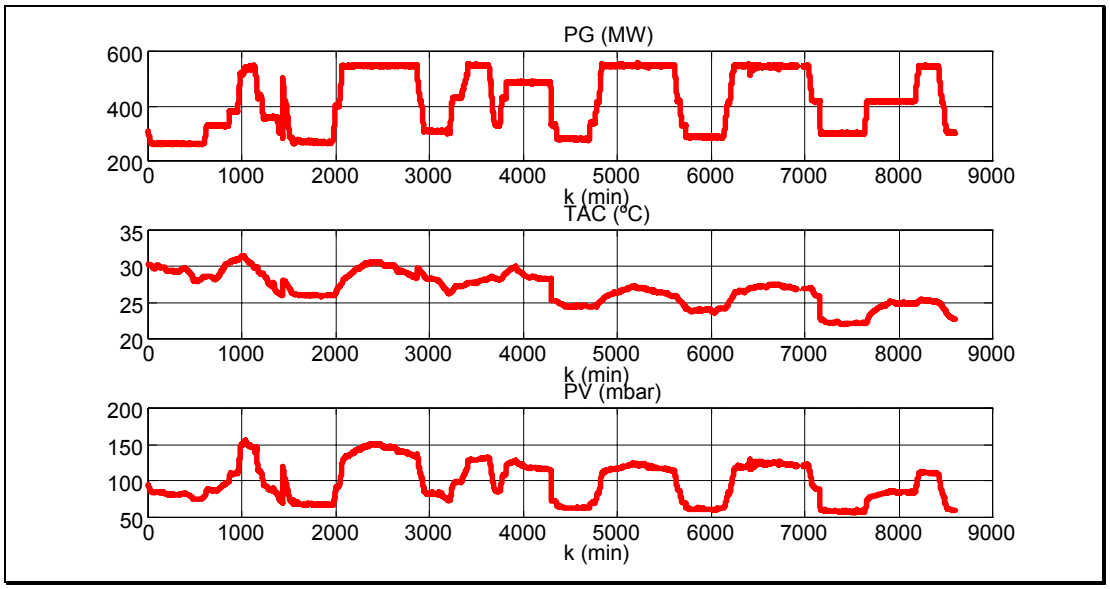

Figura 6.5: Modelo del Condensador: Datos correspondientes al conjunto de Entrenamiento (PG:Potencia Generada, TAC:Temperatura de entrada del Agua de Circulación, PV:Presión de Vacío)

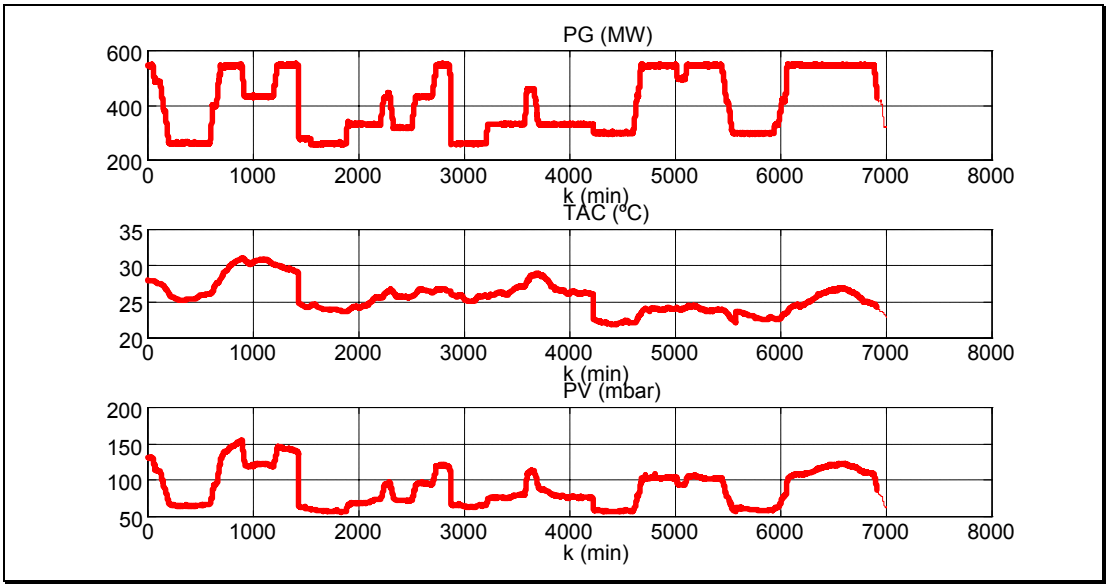

Figura 6.6: Modelo del Condensador: Datos correspondientes a los conjuntos de Test (3 primeros días) y Validación (dos últimos días) (PG:Potencia Generada, TAC:Temperatura de entrada del Agua de Circulación, PV:Presión de Vacío)

El ajuste de los parámetros de los modelos se ha realizado utilizando conjuntos reducidos de entrenamiento y test, obtenidos mediante la aplicación de los árboles de selección de datos a los conjuntos completos (ver Apéndice A). La siguiente figura muestra la selección de datos realizada para el conjunto de entrenamiento y para el conjunto de test (en ambos casos han sido seleccionados 400 datos):

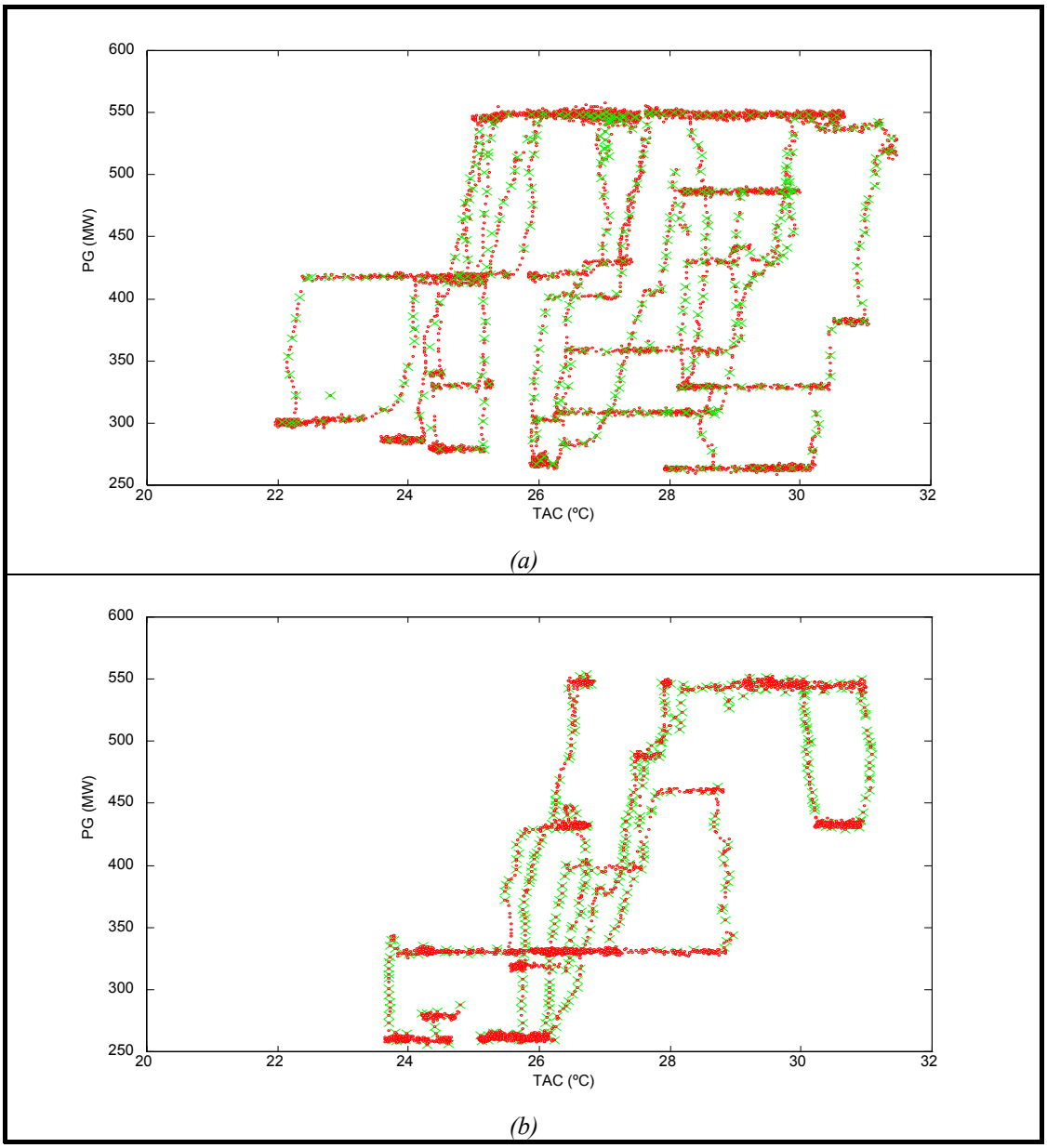

Figura 6.7: Localización en el plano (TAC,PG) de los conjuntos de entrenamiento (a) y test (b), completos (puntos) y reducidos (cruces) mediante los árboles de selección de datos

6.5.3 Ajuste del sistema de detección de anomalías

Una vez escogidas las variables del modelo de funcionamiento normal (la potencia generada *PG* y la temperatura de entrada del agua de circulación *TAC* como entradas y la presión de vacío *PV* como salida) y los conjuntos de entrenamiento, test y validación, es necesario ajustar los modelos que componen el sistema de detección de anomalías.

La Figura 6.8 muestra la estructura del sistema de detección de anomalías introducida anteriormente en este mismo capítulo, aplicada al caso concreto del condensador:

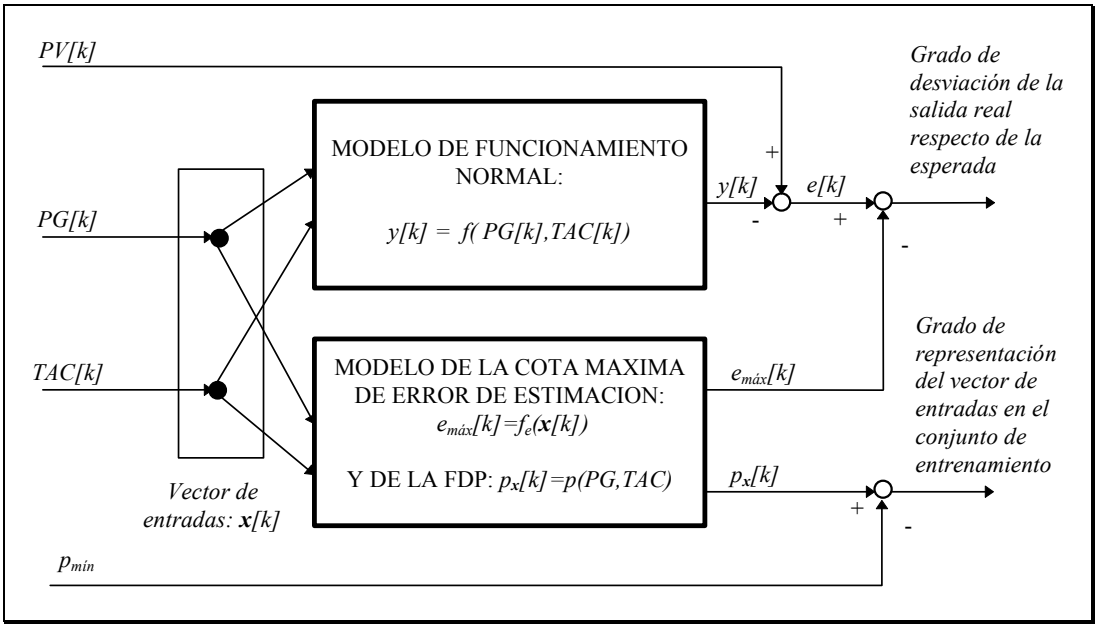

Figura 6.8: Sistema de detección de anomalías del condensador

El sistema de seguimiento continuo suministra de forma periódica (con un periodo de un minuto en este caso) las medidas de las variables de entrada y de salida de los modelos de funcionamiento normal.

Con estas medidas se evalúa en primer lugar el estimador de la fdp *p(PG,TAC)* y de la cota máxima de error de estimación $e_{máx}(PG,TAC)$. Si el valor estimado de *p(PG,TAC)* está por encima de la cota de extrapolación ($p_{mín}$), se considera que el modelo de funcionamiento normal del condensador ha sido entrenado con suficientes datos de esa región del espacio de entrada (nos encontramos dentro de la región de

confianza del estimador), y que el error de estimación de la presión de vacío deberá quedar por lo tanto por debajo de $e_{máx}(PG,TAC)$.

a) Modelo de funcionamiento normal: estimación de la Presión de Vacío

El primer modelo a ajustar es el modelo de funcionamiento normal, que en este caso concreto queda descrito por la Ecuación 6.18. Como aproximador funcional ajustaremos una red PRBFN tipo II. Para ello llevaremos a cabo las dos optimizaciones parciales presentadas en el Capítulo 2: la optimización estructural para determinar el número de unidades radiales de la red, y la optimización paramétrica para el ajuste de sus parámetros.

La Figura 6.10 muestra la evolución durante el aprendizaje del número de unidades radiales de la red, así como la evolución de los errores de entrenamiento y de test. El optimizador estructural va proponiendo sucesivamente estructuras cada vez más complejas al optimizador paramétrico, hasta que detecta que el error de test ha pasado por un mínimo. El optimizador paramétrico utiliza en este caso un método quasi-Newton de "memoria reducida" para minimizar de forma iterativa el error de entrenamiento, y se da por finalizado cuando la tendencia del error de test sobrepasa una cota máxima (ver Capítulo 2).

La estructura óptima así obtenida consta de 10 unidades radiales repartidas en el espacio de entrada como se muestra en la Figura 6.9:

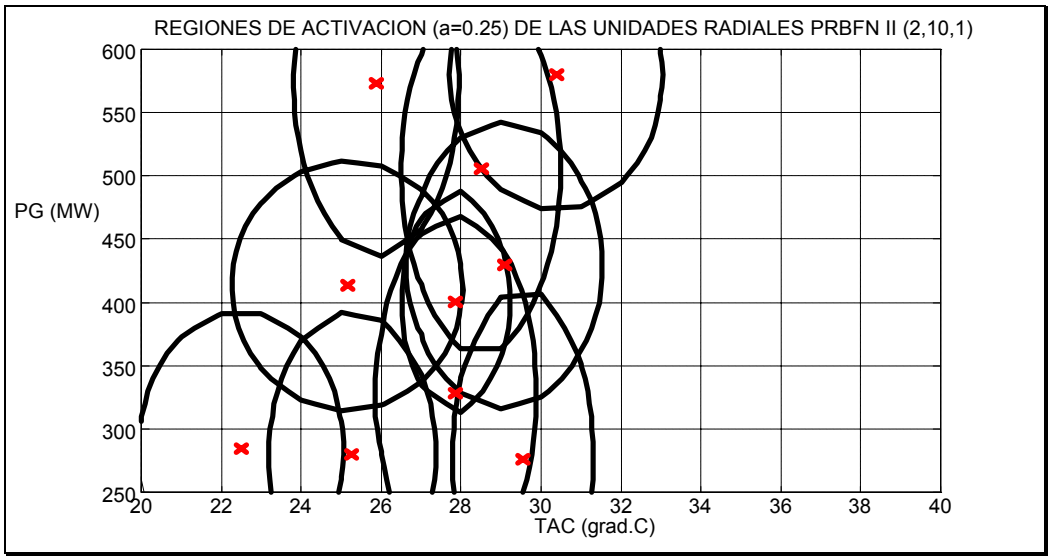

Figura 6.9: Regiones de activación de las unidades radiales de la red de estimación de la presión de vacío del condensador

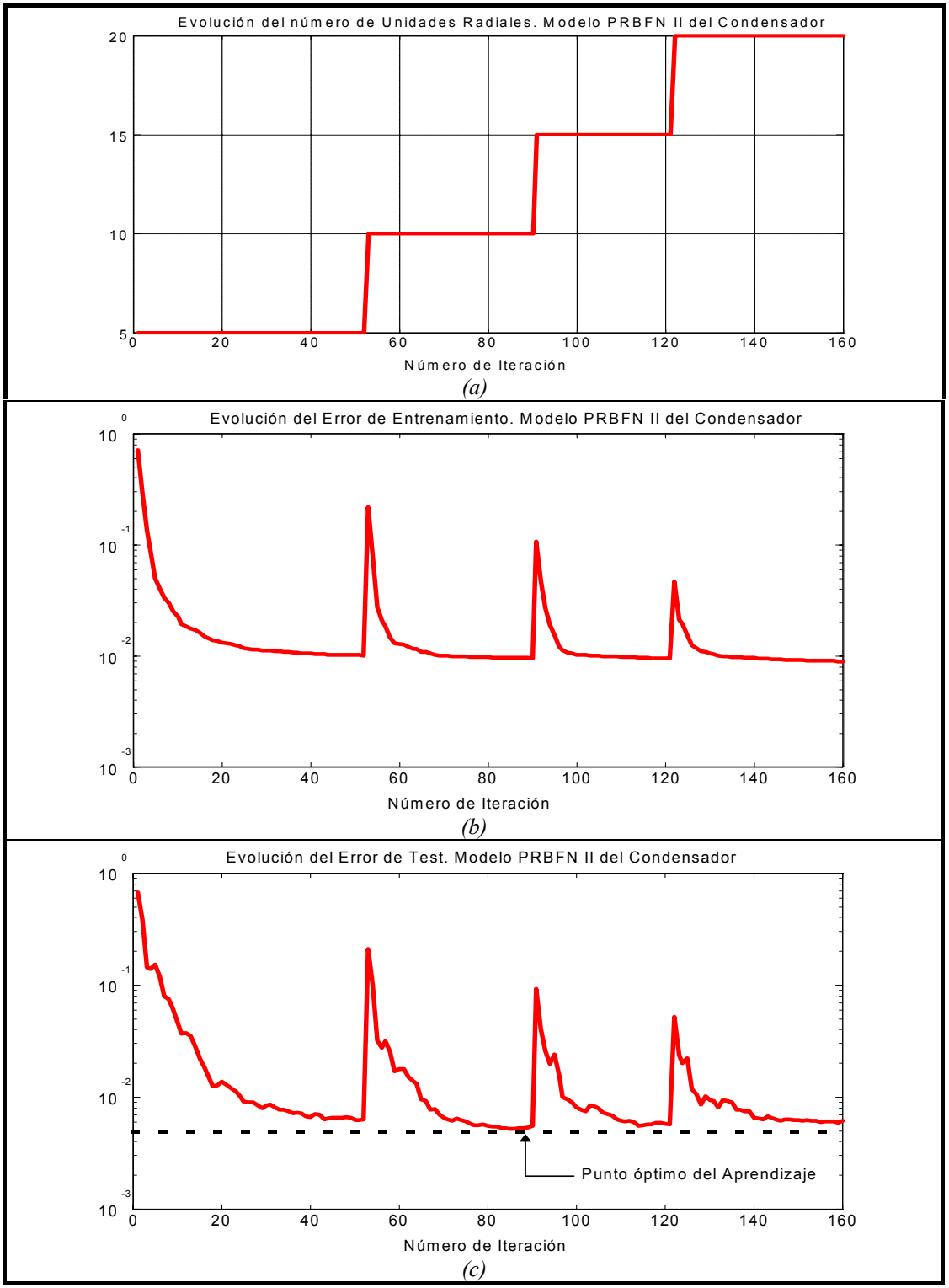

Figura 6.10: Ajuste del estimador de la presión de vacío: (a) Evolución del número de unidades radiales (b) Evolución del error de entrenamiento (c) Evolución del error de test

La presión de vacío estimada por la red PRBFN tipo II, a partir de la potencia generada y de la temperatura de entrada del agua de circulación, se muestra en la Figura 6.11:

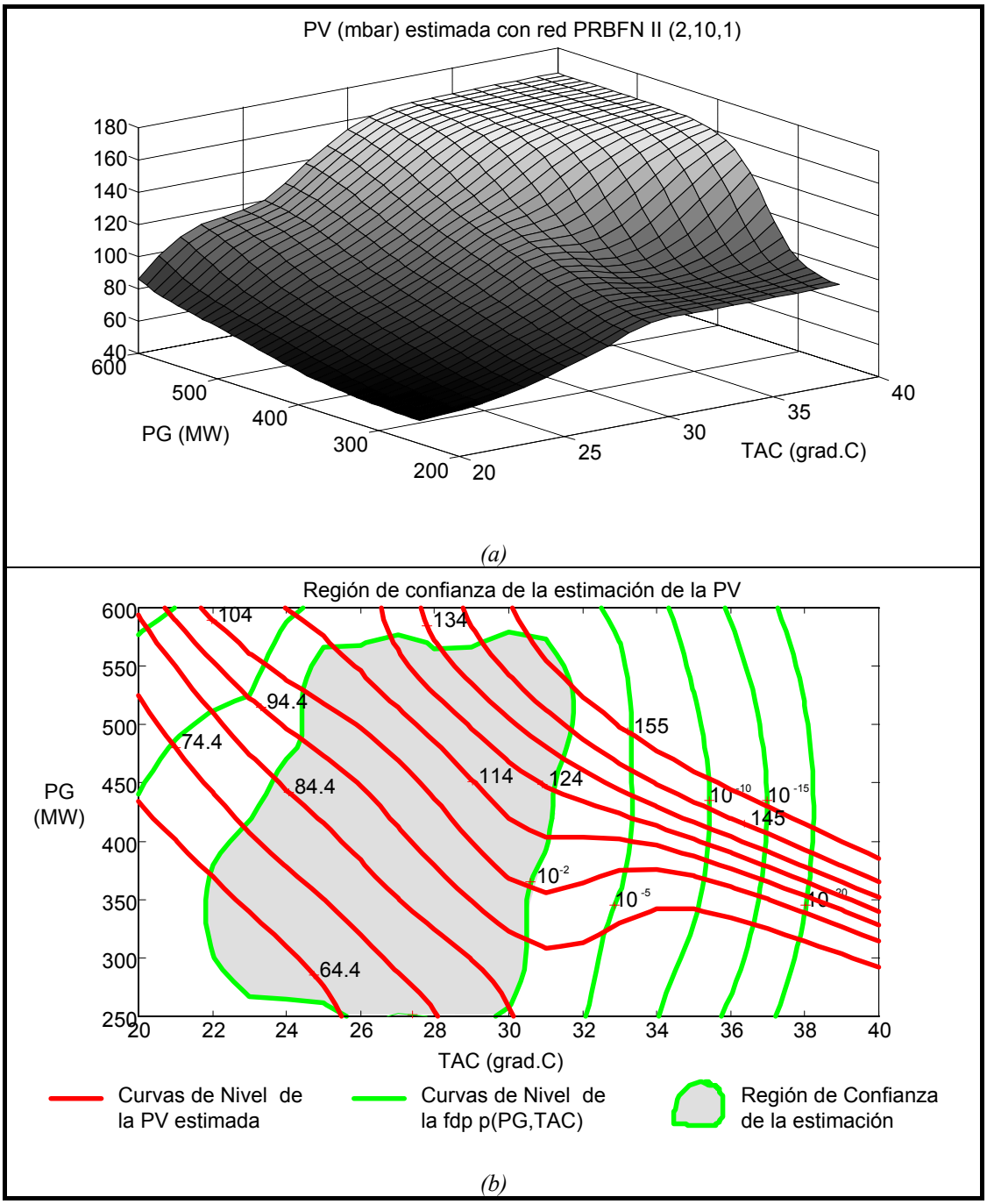

Figura 6.11: Estimación de la Presión de Vacío del condensador con red PRBFN II (2,10,1)
(a) Superficie de Presión de Vacío
(b) Curvas de nivel y región de confianza de la estimación

b) Modelo de la fdp del vector de entradas: *p(PG,TAC)*

Es importante señalar que al tratarse de un modelo de caja negra, la estimación de la presión de vacío del condensador es únicamente fiable en la región del espacio de entrada que ha quedado representada por los ejemplos de entrenamiento: la región de confianza de la estimación.

Tomaremos como región de confianza aquella región del espacio de entrada del modelo de funcionamiento normal (*PG*×*TAC*) a la que corresponden valores de *p(PG,TAC)* por encima de la cota de extrapolación. De esta forma, toda estimación de la Presión de Vacío irá acompañada de una medida de extrapolación que indicará si nos hallamos en el interior o en el exterior de la región de confianza de la estimación.

En el interior de la región de confianza, el modelo ajustado será un patrón de funcionamiento normal de tal forma que el error de estimación en condiciones de funcionamiento normal deberá mantenerse dentro de unos límites preestablecidos. Fuera de esta región, el modelo ajustado pierde toda su validez como patrón de funcionamiento normal, por lo que no podrá realizarse la labor de detección de anomalías en el componente modelado. Sin embargo, el bajo valor de la fdp del vector de entradas es indicativo de un valor anómalo en al menos una de las variables de entrada, pudiendo ser el resultado de un fallo en otro componente anterior del mismo proceso productivo.

Como estimador de la fdp *p(PG,TAC)* se ha utilizado una red PRBFN tipo I, con 10 unidades radiales. El ajuste de sus centros y factores de escala (únicos parámetros necesarios para esta estimación) se ha realizado maximizando la función de verosimilitud logarítmica de los vectores de entrada del conjunto de entrenamiento. Las curvas de nivel de la superficie *p(PG,TAC)* estimada se muestran en la Figura 6.12, junto con los datos de entrenamiento.

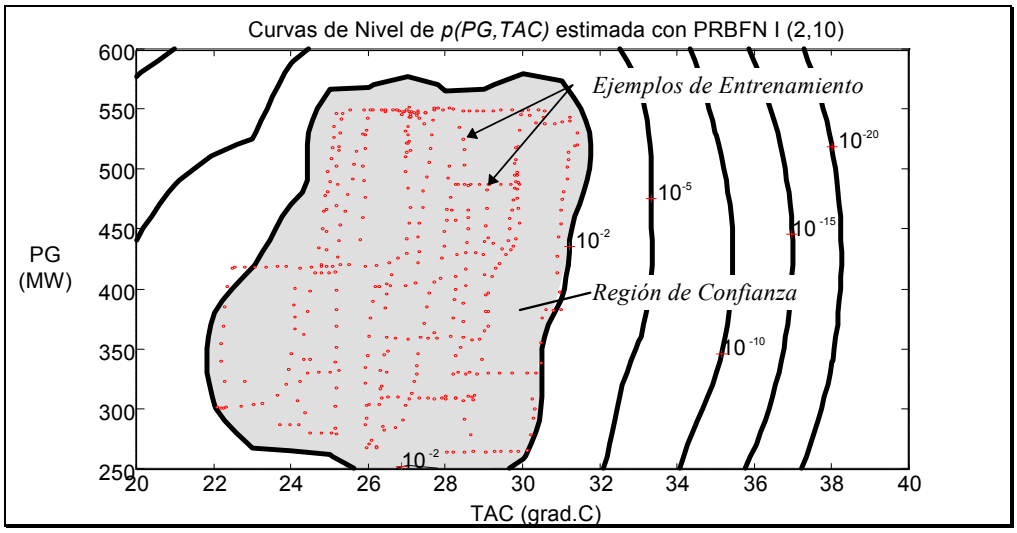

Figura 6.12: Curvas de nivel de la fpd p(PG,TAC) estimada con PRBFNI

Para establecer la cota de extrapolación ($p_{mín}$), basta con calcular la función de distribución de la fdp *p(PG,TAC)*, estimada con el conjunto de entrenamiento:

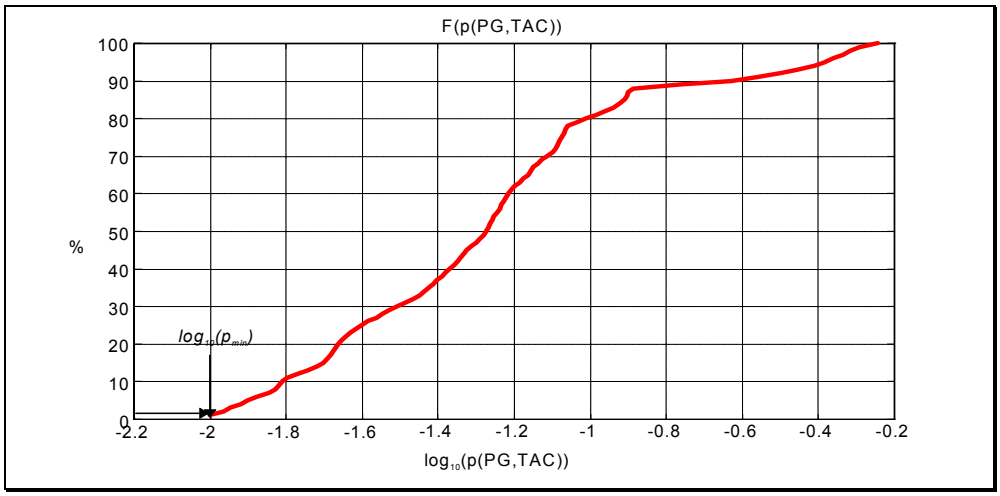

Figura 6.13: Función de distribución de p(PG,TAC) estimada sobre el conjunto de entrenamiento. Determinación de la cota de extrapolación.

y tomar como cota aquél valor de *p(PG,TAC)* correspondiente a un valor de función de distribución del 2%. En este caso, el 98% de los datos de entrenamiento producen una *p(PG,TAC)* superior a $p_{mín}=0.01$.

c) Modelo de la cota máxima de los residuos

Una vez ajustado el modelo de la presión de vacío del condensador y el modelo de la fdp del vector de entradas, podemos ajustar el estimador de cota máxima de los residuos realizando la estimación local de la varianza del error de entrenamiento. Para ello utilizaremos la red de estimación de la fdp *p(PG,TAC)* estimando la varianza local del error de estimación de la presión de vacío en cada una de sus unidades radiales, como se vió en el apartado 6.2.4. Una vez calculadas estas varianzas locales, la varianza del error de estimación (*e*) en un punto $\boldsymbol{x}$ del espacio de entrada se obtiene mediante regresión generalizada como la esperanza matemática $E(e^2/\boldsymbol{x})$.

La desviación típica estimada del error de estimación de la presión de vacío se muestra en la Figura 6.14. Tomaremos como cota máxima admisible de error de estimación, con una confianza del 95%, el doble de la desviación típica estimada, tomando como hipótesis una distribución local normal de los errores de estimación.

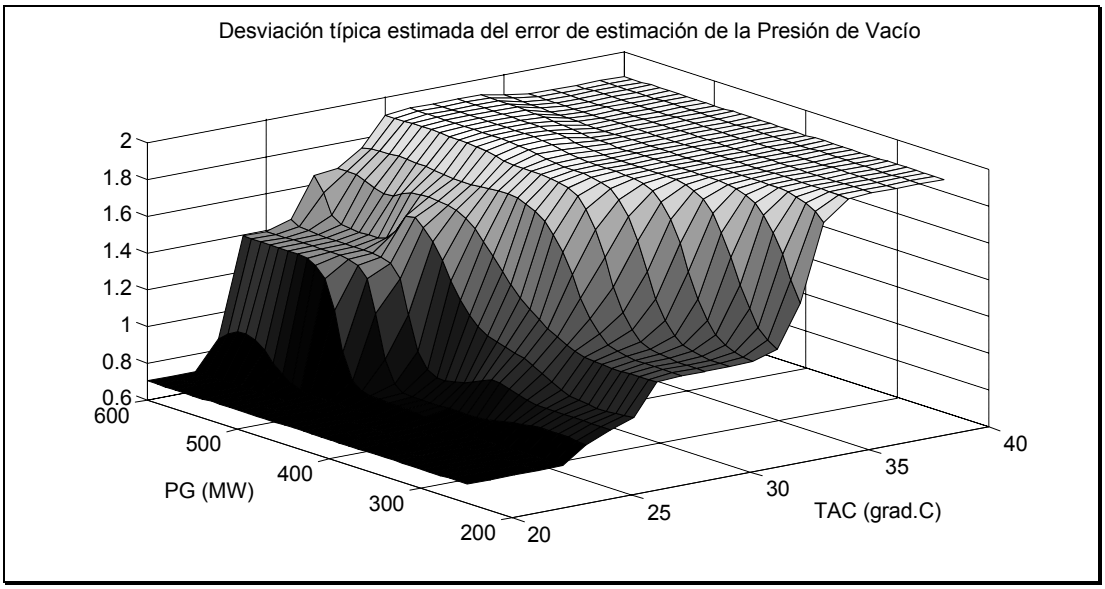

Figura 6.14: Desviación típica estimada del error de estimación de la presión de vacío

En la superficie anterior puede comprobarse cómo la desviación típica estimada del error de estimación de la Presión de Vacío es una función creciente con la Potencia Generada y de la Temperatura de entrada del Agua de Circulación. Este tipo de relación funcional refleja el incremento de inestabilidad de la Presión de Vacío que se produce tanto al aumentar la Potencia Generada en la central (debido al incremento correspondiente en el flujo de vapor a condensar), como al aumentar la

Temperatura de entrada del Agua de Circulación (debido a un menor rendimiento en el intercambio de calor entre el vapor y el Agua de Circulación).

Por último señalar nuevamente que esta estimación de la desviación típica del error de estimación sólo tiene validez en la región de confianza. Fuera de ella no se dispone de datos suficientes como para cuantificar el error de estimación de la Presión de Vacío.

6.5.4 Resultados

A continuación se presentan las bandas de funcionamiento normal de la evolución de la Presión de Vacío, así como la fdp *p(PG,TAC)*, estimadas para distintos conjuntos de datos.

En la Figura 6.15 aparecen las estimaciones correspondientes al conjunto de entrenamiento. Tal como ha sido establecida la cota de extrapolación, el 98% de los datos de entrenamiento quedan dentro de la región de confianza del estimador, resultando para estos datos una fdp *p(PG,TAC)* superior a la cota de extrapolación.

La Presión de Vacío medida entra dentro de las bandas de funcionamiento normal, correspondientes a errores de estimación inferiores a dos veces la desviación típica estimada del error, indicando la condición de funcionamiento normal. En la Figura 6.16 se ha ampliado parte de la Figura 6.15, presentando las estimaciones correspondientes al primer día del conjunto de entrenamiento.

La Figura 6.17 presenta los resultados obtenidos con los conjuntos de test y de validación. Con el primero de ellos los resultados son muy similares a los obtenidos con el conjunto de entrenamiento. Con el de validación sin embargo se han detectado situaciones en las que el vector de entradas se salía fuera de la región de confianza. Analizando los datos correspondientes a estas situaciones, se comprueba efectivamente que las condiciones de operación son distintas de las representadas en el conjunto de entrenamiento, al estar funcionando la central a plena carga con una temperatura del agua de circulación por debajo de los 25ºC. En ningún caso sin embargo han sido detectados valores anómalos de la presión de vacío. Estos datos deberían ser introducidos en la base de datos de funcionamiento normal para que el sistema de detección de anomalías se aprenda estas nuevas condiciones de operación.

La Figura 6.18 recoge los resultados obtenidos con datos correspondientes a un fallo en el sistema de vacío del condensador. Es importante resaltar que durante la presencia de la anomalía las variables de entrada del modelo siguen manteniéndose en la región de confianza, lo que asegura la validez del modelo de funcionamiento

normal en estas condiciones y un elevado grado de certeza en la detección de una anomalía interna al componente en cuestión.

El caso representado en la Figura 6.19 corresponde también a una situación anómala, pero externa al componente supervisado. En este caso es el modelo de estimación de la fdp p(PG,TAC) el que detecta la anomalía externa e invalida el modelo de funcionamiento normal. Los datos corresponden a un disparo de la central por fallo de turbina y puede apreciarse perfectamente en esta gráfica el proceso de recuperación de la presión de vacío.

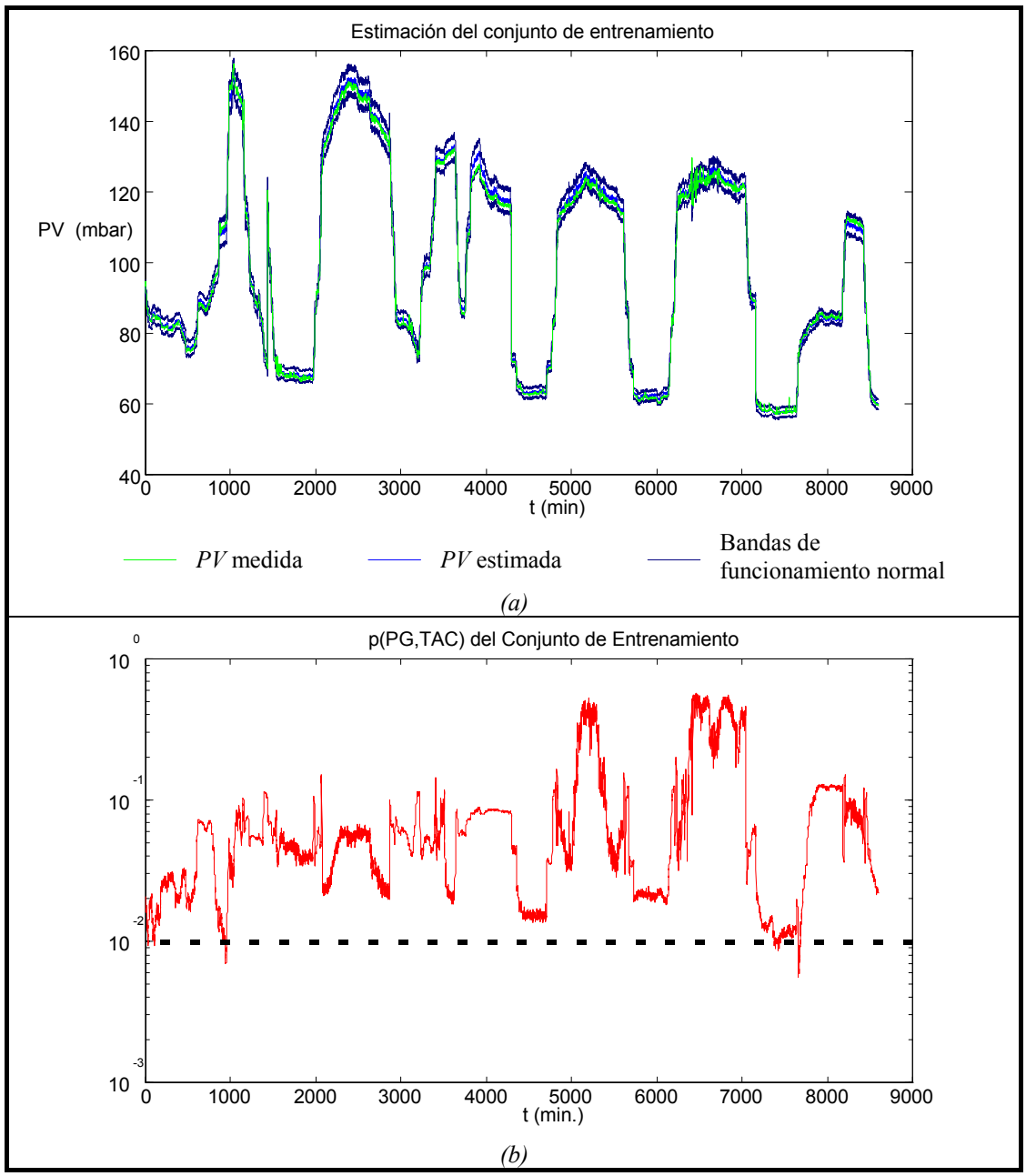

Figura 6.15: (a) Bandas de funcionamiento normal y (b) fdp p(PG,TAC) estimadas para el conjunto de entrenamiento

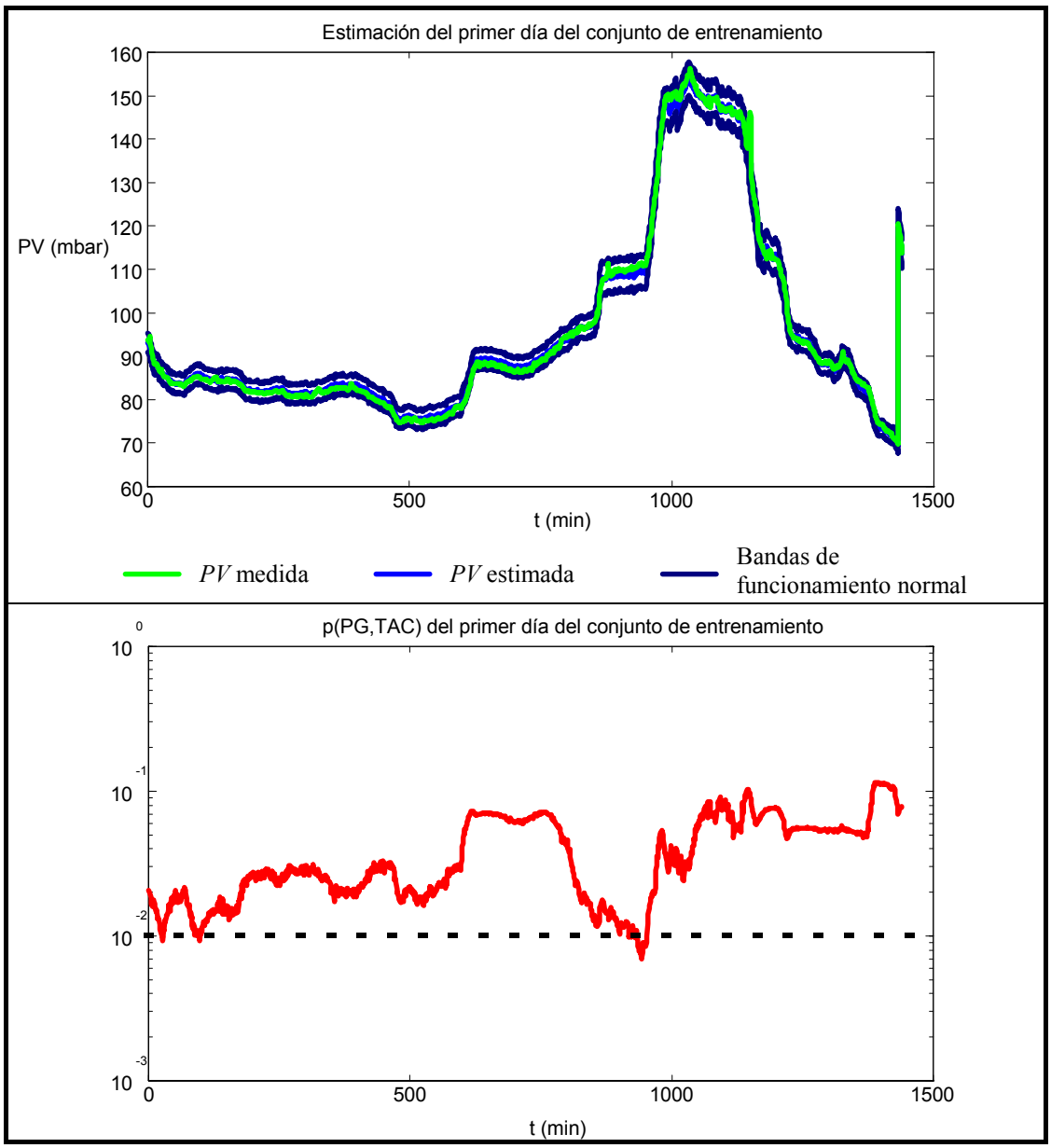

Figura 6.16: (a) Bandas de funcionamiento normal y (b) fdp p(PG,TAC) estimadas para el primer día del conjunto de entrenamiento

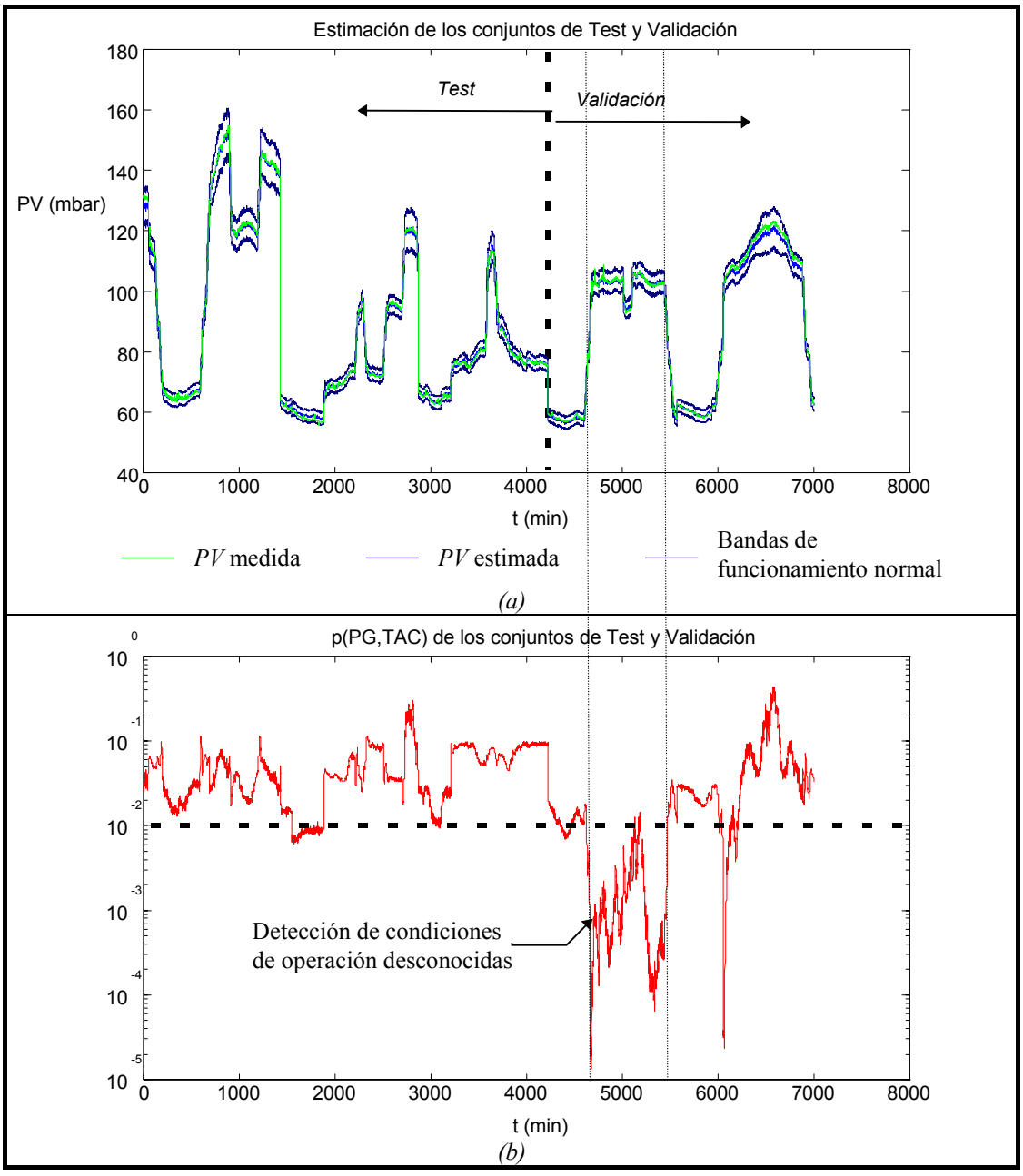

Figura 6.17: (a) Bandas de funcionamiento normal y (b) fdp p(PG,TAC) estimadas para los conjuntos de test y validación

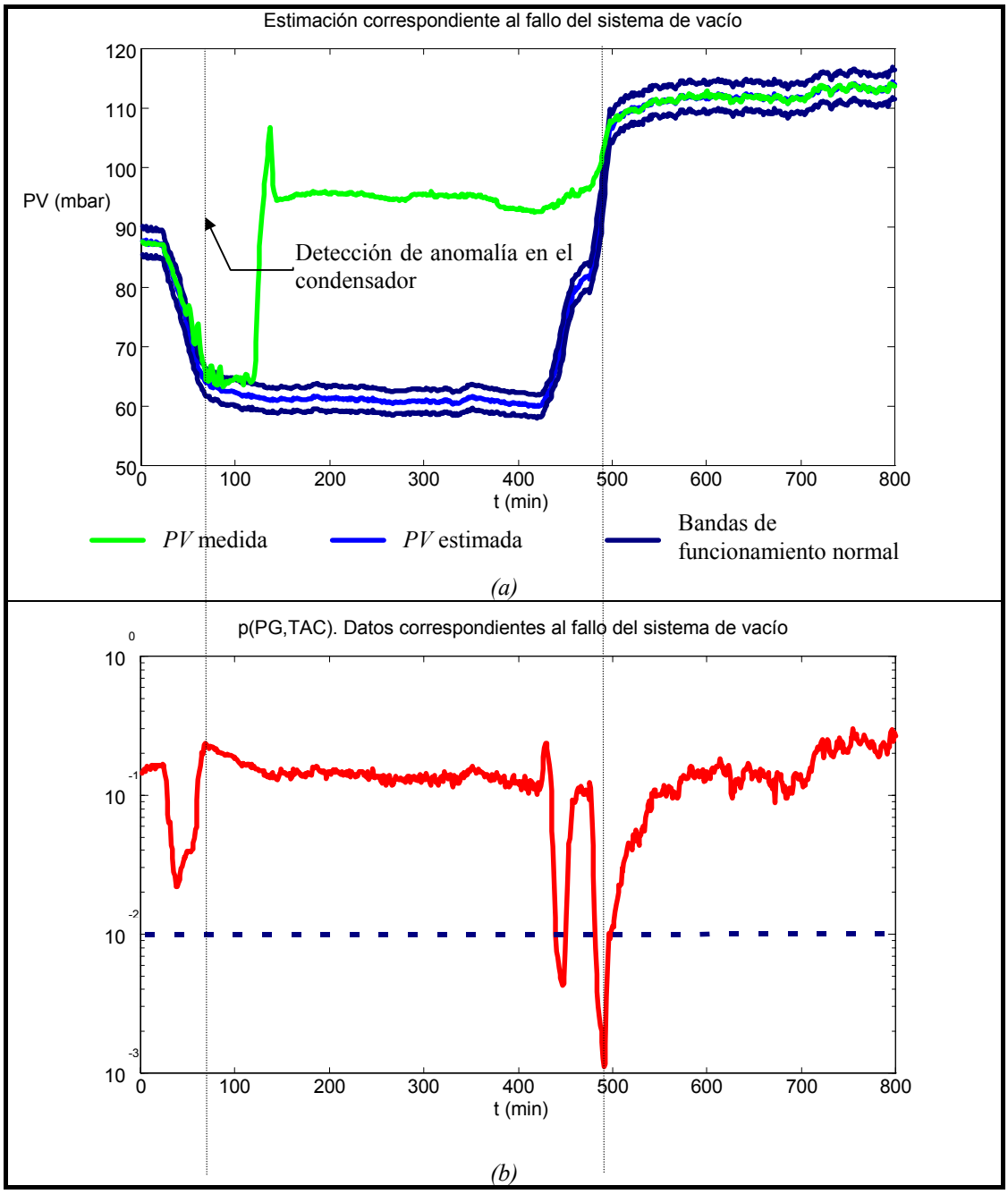

Figura 6.18: (a) Bandas de funcionamiento normal y (b) fdp p(PG,TAC) estimadas para el día de fallo en el sistema de vacío

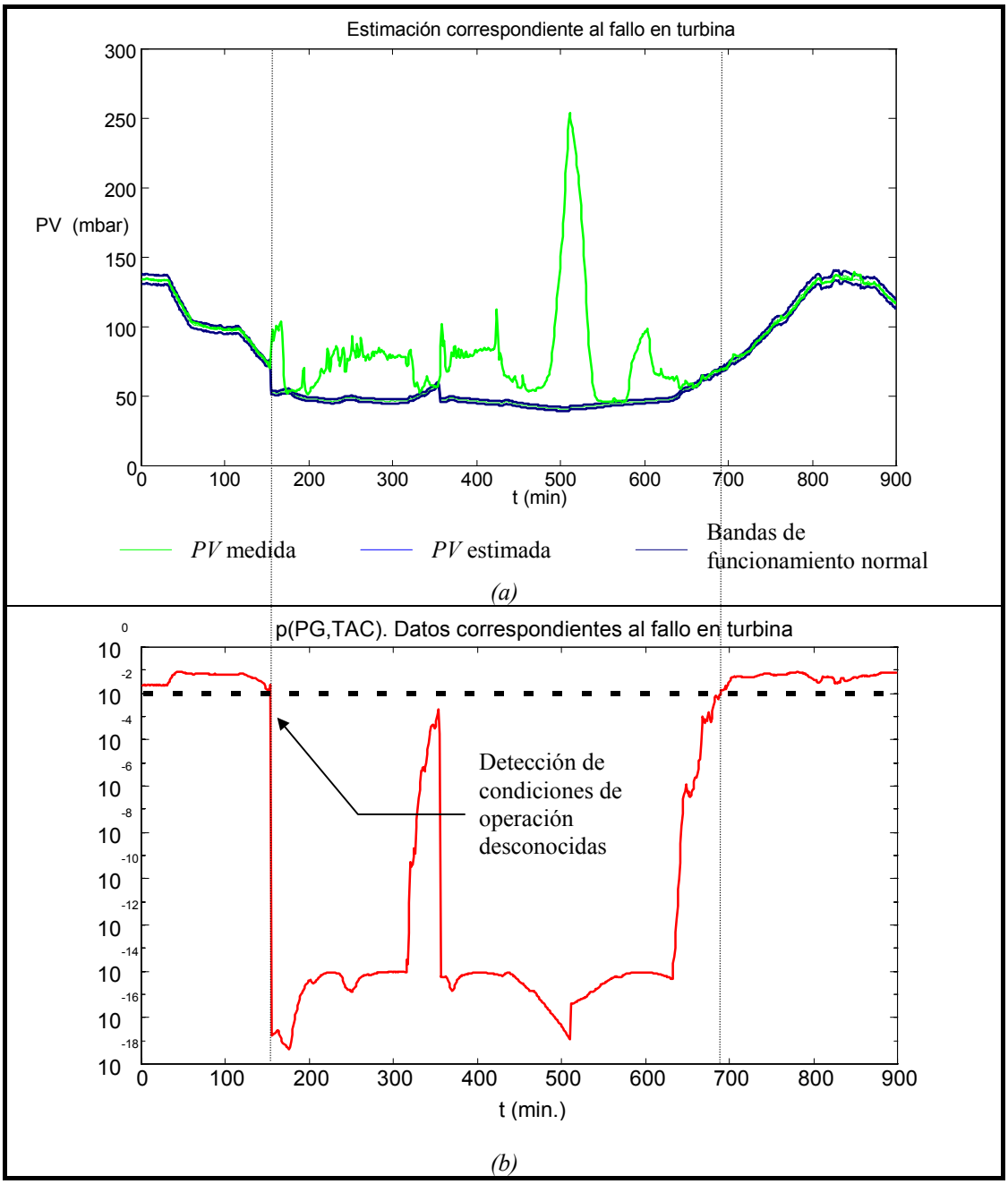

Figura 6.19: (a) Bandas de funcionamiento normal y (b) fdp p(PG,TAC) estimadas para el día de fallo en turbina

6.6 Aplicación a la detección de anomalías en la química del agua de una Central Térmica

6.6.1 Introducción

En este segundo ejemplo vamos a tratar el diagnóstico de la química del agua de una Central Térmica (C.T.). Los fenómenos de corrosión en componentes tales como la caldera, la turbina, o el condensador de una C.T. pueden provocar degradaciones de gravísimas consecuencias para el proceso de producción de energía eléctrica. Resulta por lo tanto de vital importancia el controlar la pureza del agua que interviene en el ciclo agua-vapor de la central, con el fin de evitar los mencionados fenómenos de corrosión. Esta labor queda encomendada al laboratorio químico de la C.T., que controla de forma permanente las propiedades químicas del agua del ciclo, a partir de ensayos periódicos realizados en el laboratorio, y de medidas tomadas directamente de los sensores instalados en los distintos componentes de la C.T.

Para apoyar la labor realizada por el personal químico de las C.T., se llevó a cabo un proyecto de colaboración entre Unión Eléctrica Fenosa S.A. y el Instituto de Investigación Tecnológica de la Universidad Pontificia Comillas, para el desarrollo de un sistema experto de control de la química del agua del ciclo agua-vapor. Este sistema está encargado de procesar toda la información que le llega de forma automática de los sensores, y los resultados de los ensayos de laboratorio, con el fin de detectar situaciones anómalas, localizar el origen de las mismas, y proponer posibles acciones correctoras. El resultado de esta colaboración fue el sistema experto SEQA (*[Sanz Bobi et al., 1994-2]*), que está en operación desde 1989.

6.6.2 Selección de las variables del modelo de funcionamiento normal y recogida de datos

El sistema experto SEQA fue instalado en la Central Térmica de Anllares, propiedad de Unión Eléctrica Fenosa. En esta central, el sistema experto recoge de forma automática, entre otras variables, las propiedades químicas en los siguientes puntos del ciclo (ver Figura 6.20):

- Vapor condensado o simplemente "condensado"
- Agua de alimentación
- Vapor saturado
- Vapor sobrecalentado
- Vapor recalentado

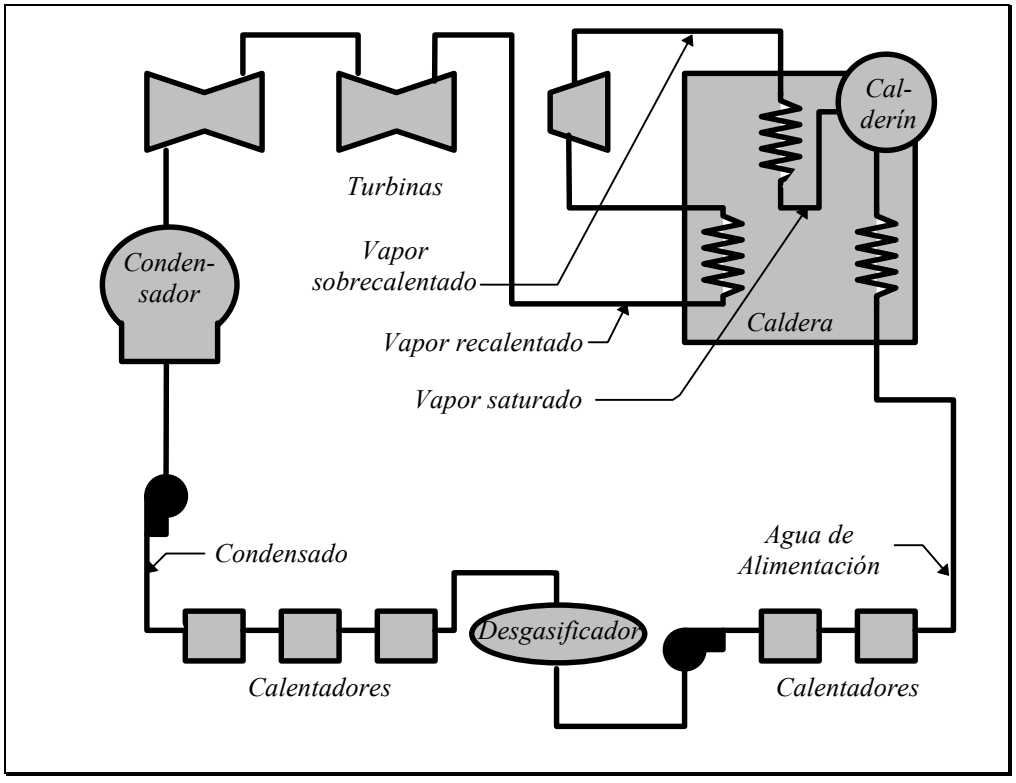

Figura 6.20: Esquema simplificado del circuito de Agua-Vapor de una Central Térmica

El peligro de corrosión aparece fundamentalmente bajo la presencia de O_2. Esta situación es susceptible de producirse en el condensado por dos razones fundamentales. La primera tiene su origen en las posibles infiltraciones de aire en el cuerpo del condensador, debidas a la baja presión reinante. La segunda radica en la presencia de O_2 en las aportaciones de agua desmineralizada que se realizan al ciclo en este punto.

Estas razones justifican la selección del condensado como uno de los puntos más importantes a supervisar dentro del ciclo. Las propiedades químicas del condensado que SEQA recoge de forma automática y almacena en forma de promedio cada 15 minutos en la base de datos histórica son:

- Conductividad catiónica
- Conductividad específica
- pH
- Contenido de O_2

Tomaremos la conductividad catiónica del condensado (CC_{cond}) como variable de salida del modelo de funcionamiento normal. Es importante señalar que la conductividad catiónica del agua alcanza su valor máximo precisamente en el condensado, reflejando la especial vulnerabilidad de este punto del ciclo.

Como posibles variables explicativas exógenas tomaremos la potencia generada en la central (PG) y la conductividad catiónica del vapor recalentado (CC_{recal}). Al igual que en el ejemplo anterior, la potencia generada es la variable de control de todo el proceso productivo. Por otro lado, el vapor recalentado es el punto de medida que precede al condensado según se cierra el ciclo.

Para formar los conjuntos de entrenamiento y test el laboratorio químico de la central snos facilitó un conjunto de 2000 muestras (con un periodo de muestreo de 15 minutos) correspondientes a un periodo de evolución normal de las variables químicas del ciclo, y se dedicaron 1500 para entrenamiento y 500 para test.

La Figura 6.21 muestra las 500 primeras muestras del conjunto de entrenamiento, mientras que la Figura 6.22 corresponde al conjunto de test.

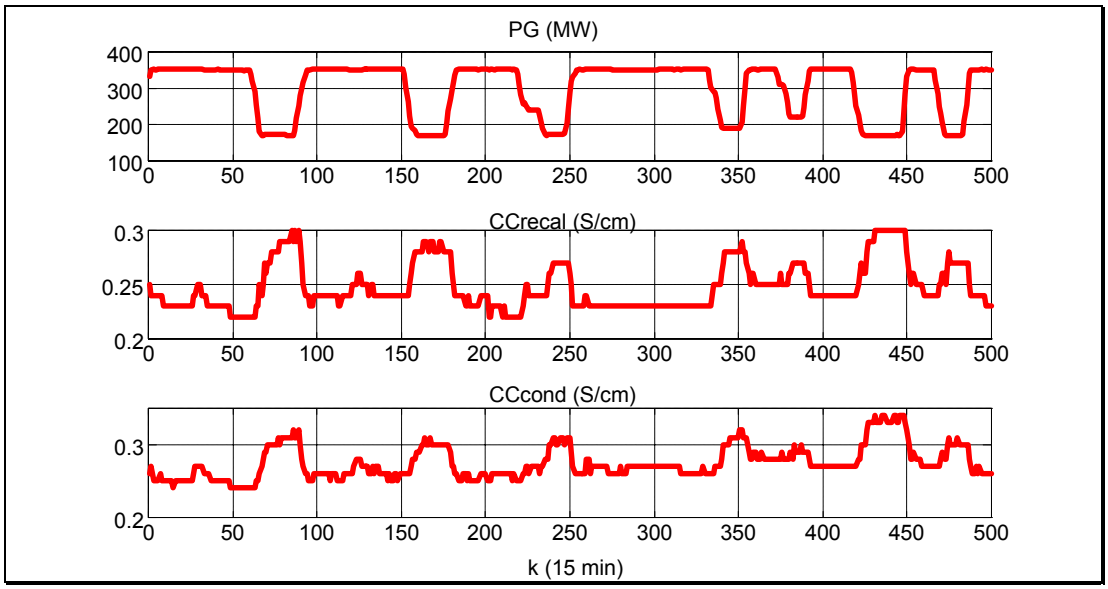

Figura 6.21: Modelo de la química del agua: Datos correspondientes a las 500 primeras muestras del conjunto de Entrenamiento (PG: Potencia Generada, CCrecal: Conductividad Catiónica de Recalentado, CCcond: Conductividad Catiónica de Condensado)

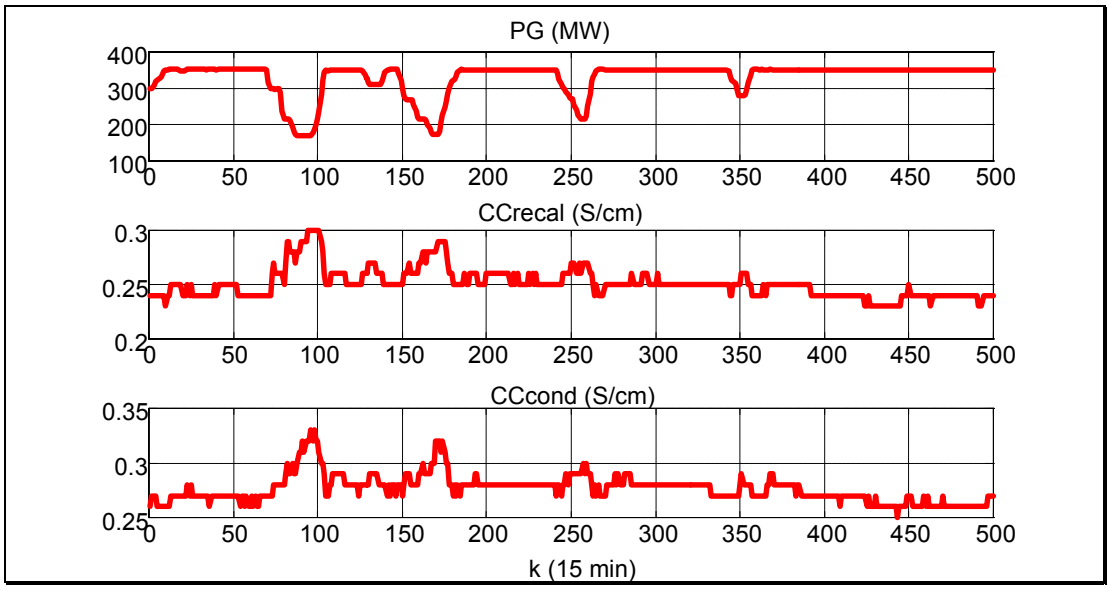

Figura 6.22: Modelo de la química del agua: Datos correspondientes al conjunto de Test (PG:Potencia Generada, CCrecal: Conductividad Catiónica de Recalentado, CCcond: Conductividad Catiónica de Condensado)

Para analizar la influencia de estas variables sobre la conductividad catiónica del condensado podemos representar la evolución conjunta de las tres señales y estudiar la sucesión de acontecimientos que tiene lugar. En la Figura 6.23 aparecen la respuestas de estas tres señales frente a dos subidas y bajadas de carga distintas.

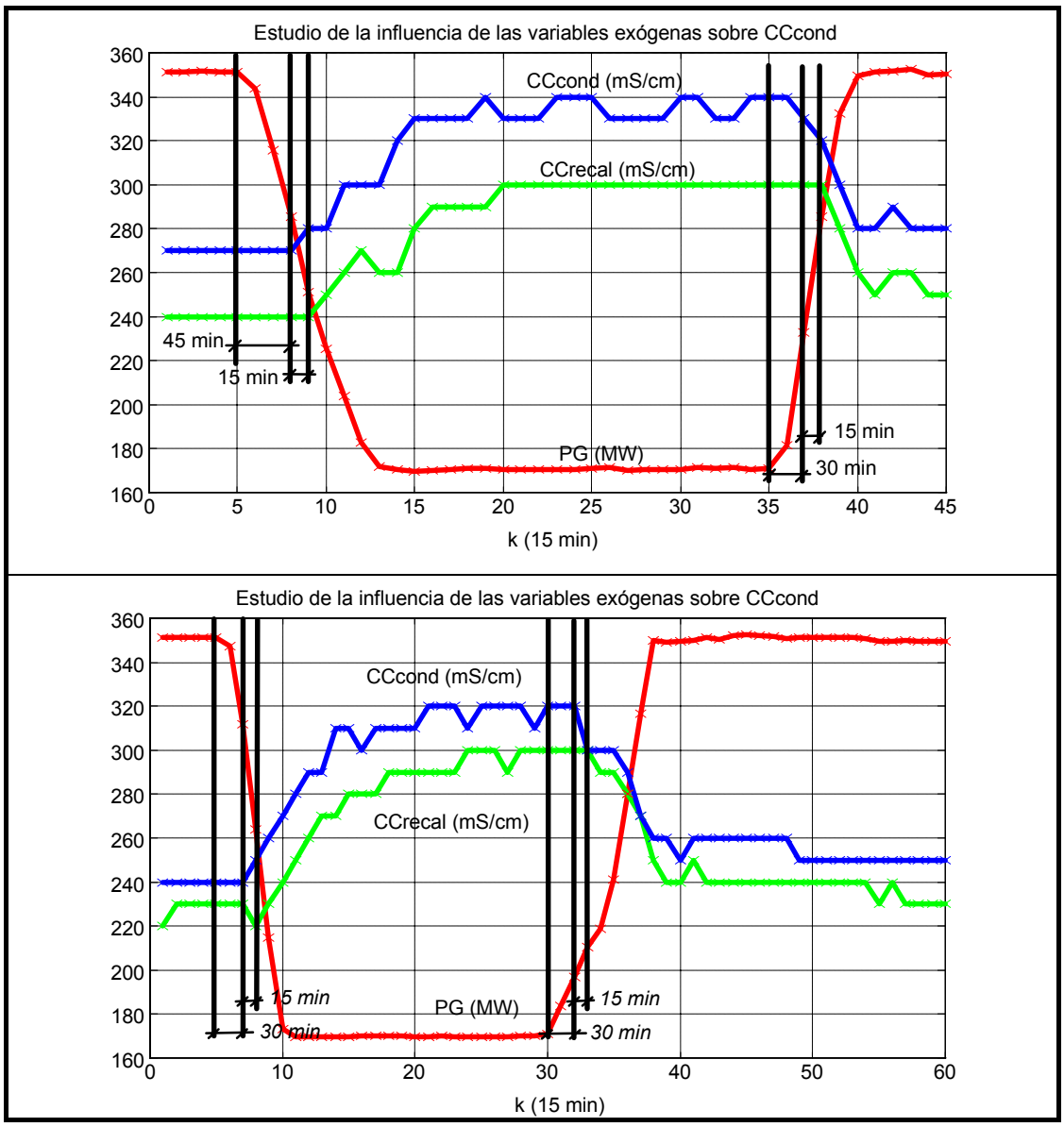

Figura 6.23: Ejemplos de evolución conjunta de la Potencia Generada, de la Conductividad Catiónica del Vapor Recalentado y de la Conductividad Catiónica de Condensado

En esta misma figura se aprecia claramente cómo la potencia generada es la primera en reflejar la dinámica del proceso. Tras ella se sitúa la conductividad catiónica de condensado, que tras un largo retardo de entre 30 y 45 minutos, evoluciona de forma

opuesta a *PG*. La última en responder es la conductividad catiónica del vapor recalentado, que lo hace unos 15 minutos más tarde que CC_{cond} y de forma semejante a ella. Esta secuencia de respuestas confirma la hipotesis de que el origen de las impurezas que hacen crecer la conductividad catiónica en el ciclo queda normalmente localizado a nivel del condensado.

La conclusión de este análisis es que para predecir el valor de CC_{cond} podemos utilizar como variables explicativas valores pasados de *PG* y de la propia CC_{cond}, no aportando ninguna información en condiciones de funcionamiento normal la CC_{recal}, ya que es un reflejo de las otras dos.

Para ilustrar la selección de variables de entrada mediante el Análisis Estadístico de Sensibilidades (*[Muñoz & Czernichow, 1995]*), ensayaremos sin embargo en primer lugar un modelo del tipo NARX que considere las dos variables exógenas *PG* y CC_{recal}. El modelo ensayado toma la forma:

$$CC_{cond}[k] = f\left(PG[k-4],\ldots,PG[k], CC_{recal}[k-1], CC_{recal}[k], CC_{cond}[k-2], CC_{cond}[k-1]\right)$$

Ecuación 6.19

donde sólo han sido considerados dos retardos autorregresivos dada la forma de la función de autocorrelación parcial de la serie de conductividad catiónica de condensado (ver Figura 6.24).

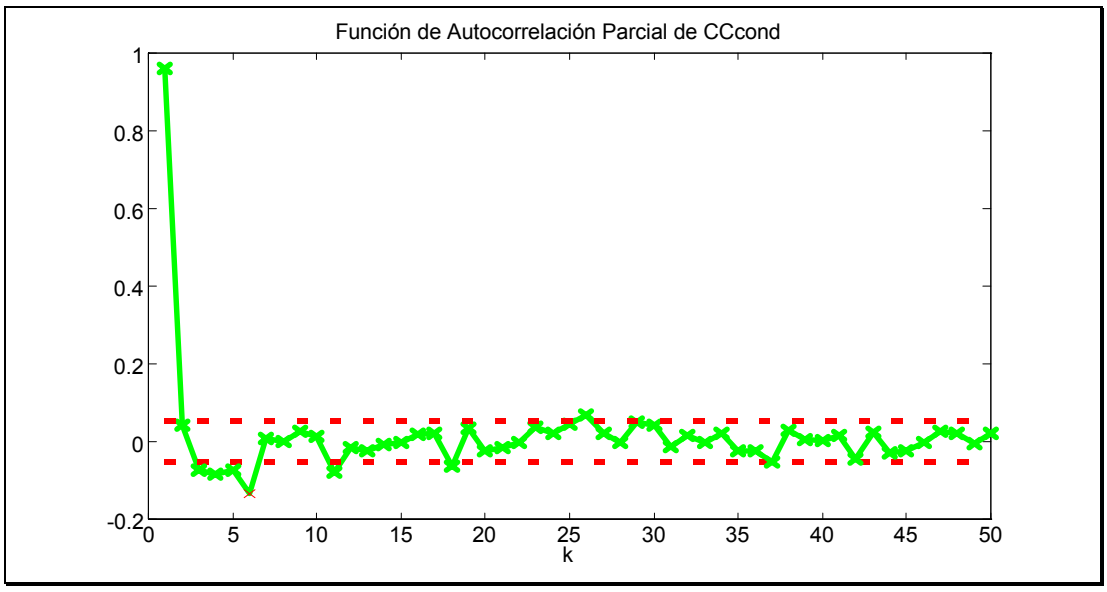

Figura 6.24: Función de Autocorrelación Parcial de la conductividad catiónica de condensado

El aproximador funcional ensayado en este caso fue un Perceptrón Multicapa (PM) con las 9 entradas que aparecen como argumento de la función *f* en la Ecuación 6.19. La optimización estructural determinó como estructura óptima un PM con tan sólo 5 unidades ocultas. Tras el ajuste, se calcularon las sensibilidades de la salida estimada frente a variaciones en cada una de las entradas, utilizando el conjunto de entrenamiento. La Figura 6.25 muestra los 95-centiles normalizados del valor absoluto de estas derivadas:

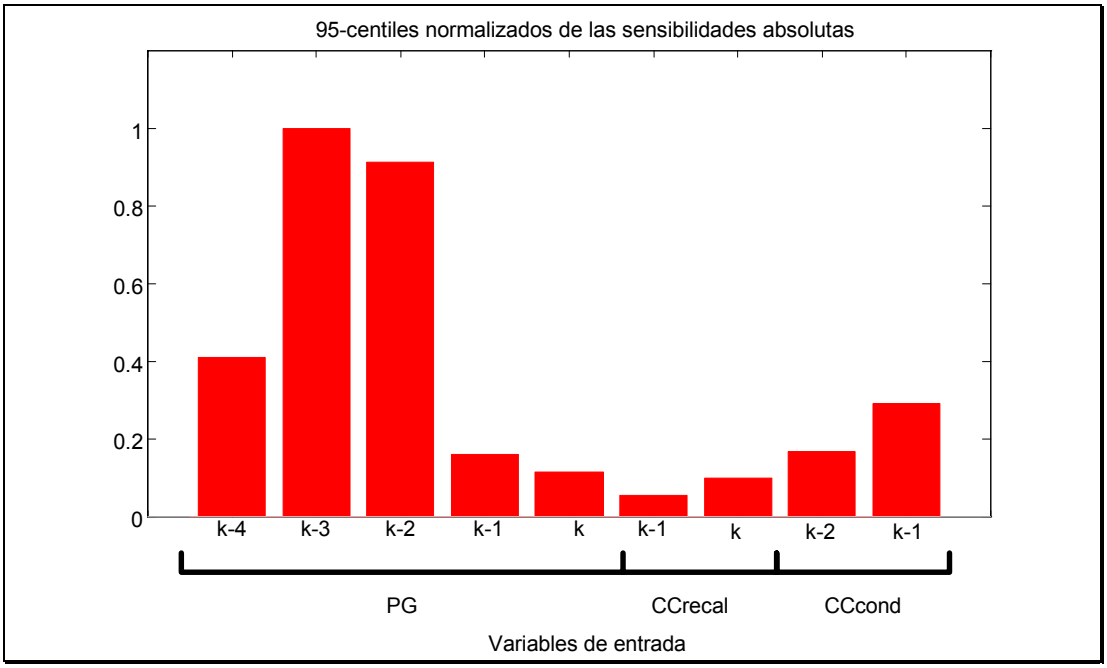

Figura 6.25: Modelo de la química del agua: 95-centiles normalizados de las sensibilidades absolutas

El análisis de este gráfico confirma las conclusiones extraídas de la inspección de la evolución conjunta de las tres señales involucradas. Centrándonos en primer lugar en la variable *PG*, que aparece como la más influyente, podemos rechazar las variables de entrada *PG[k-1]* y *PG[k]*, manteniendo como era de esperar los retardos correspondientes a 30, 45 y 60 minutos. Por otro lado, la variable CC_{recal} se muestra nuevamente como irrelevante a la hora de predecir la conductividad catiónica de condensado en condiciones normales de operación. Las sensibilidades asociadas a los distintos retardos de CC_{cond} ponen de manifiesto un cierto caracter autorregresivo de la serie. Eliminando por tanto las variables de entrada *PG[k-1]*, *PG[k]*, CC_{recal}*[k-1]* y CC_{recal}*[k]*, queda finalmente el modelo:

$$CC_{cond}[k] = f\left(PG[k-4], PG[k-3], PG[k-2], CC_{cond}[k-2], CC_{cond}[k-1]\right)$$

Ecuación 6.20

6.6.3 Ajuste del sistema de detección de anomalías

Una vez escogidas las variables de entrada y de salida del modelo de funcionamiento normal (Ecuación 6.20), y los conjuntos de entrenamiento, test y validación (este último no ha sido representado), resta tan sólo ajustar los modelos que componen el sistema de detección de anomalías, que en este caso toma la forma representada en la Figura 6.26:

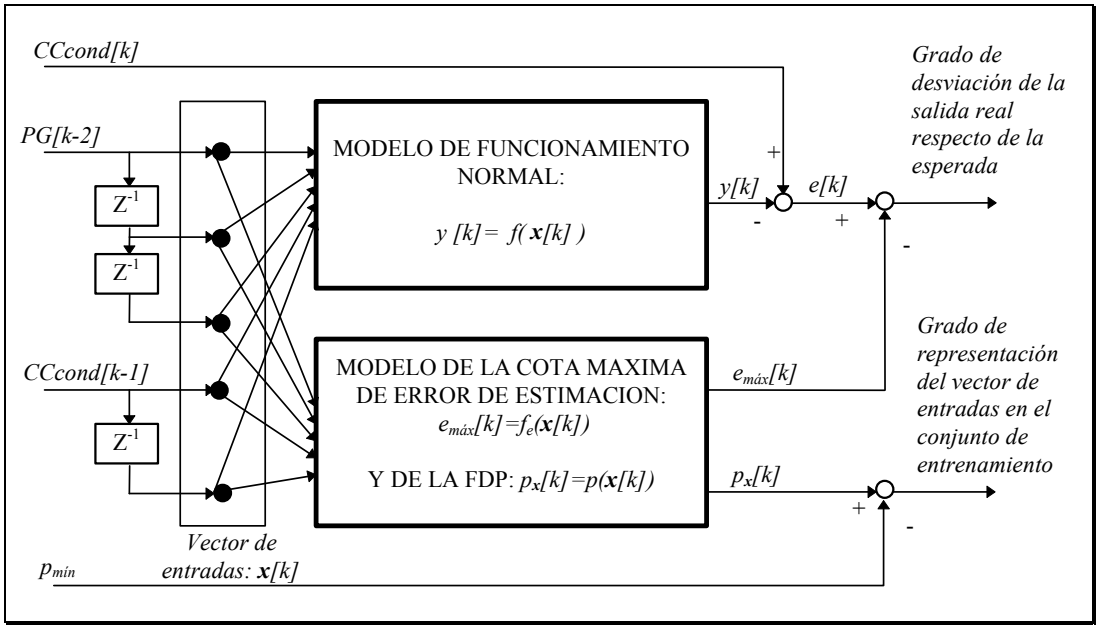

Figura 6.26: Sistema de detección de anomalías de la química del agua

El sistema de detección de anomalías vuelve a estar formado por dos modelos conexionistas: (1) el modelo de funcionamiento normal, encargado de estimar la conductividad catiónica de condensado, y (2) el modelo de la cota máxima de error de estimación y de la función de densidad probabilista del vector de entradas.

a) Modelo de funcionamiento normal: estimación de la conductividad catiónica de condensado

El modelo de funcionamiento normal queda en este caso definido por la Ecuación 6.20. El aproximador funcional utilizado en este caso fue un Perceptrón Multicapa, con funciones de activación del tipo tangentes hiperbólicas en su única capa oculta y función lineal en la de salida.

El ajuste del aproximador fue realizado según lo descrito en el Capítulo 2, es decir, aplicando dos optimizaciones parciales: la optimización estructural, encargada de determinar el número óptimo de unidades ocultas (que resultó ser 10), y la optimización paramétrica, encargada de ajustar el valor de sus parámetros. En esta última optimización fue nuevamente empleado un método quasi-Newton de "memoria reducida".

Sus capacidades de aproximación serán ilustradas en el apartado de resultados.

b) Modelo de la fdp del vector de entradas: *p(x)*

En este caso, el vector de entradas $\boldsymbol{x}$ utilizado para predecir el valor de *CCcond[k]* consta de 5 componentes:

$$\boldsymbol{x}=[PG[k-4],PG[k-3],PG[k-2],CCcond[k-2],CCcond[k-1]]^T$$

Ecuación 6.21

Para estimar la función de densidad probabilista de este vector ($p(\boldsymbol{x})$) se entrenó una red PRBFN tipo I con 10 unidades radiales, utilizando los datos de entrada del conjunto de entrenamiento del modelo de funcionamiento normal.

Para establecer la cota de extrapolación ($p_{mín}$),se evaluó el modelo resultante sobre el conjunto de entrenamiento y se construyó la función de distribución $F(p(\boldsymbol{x}))$:

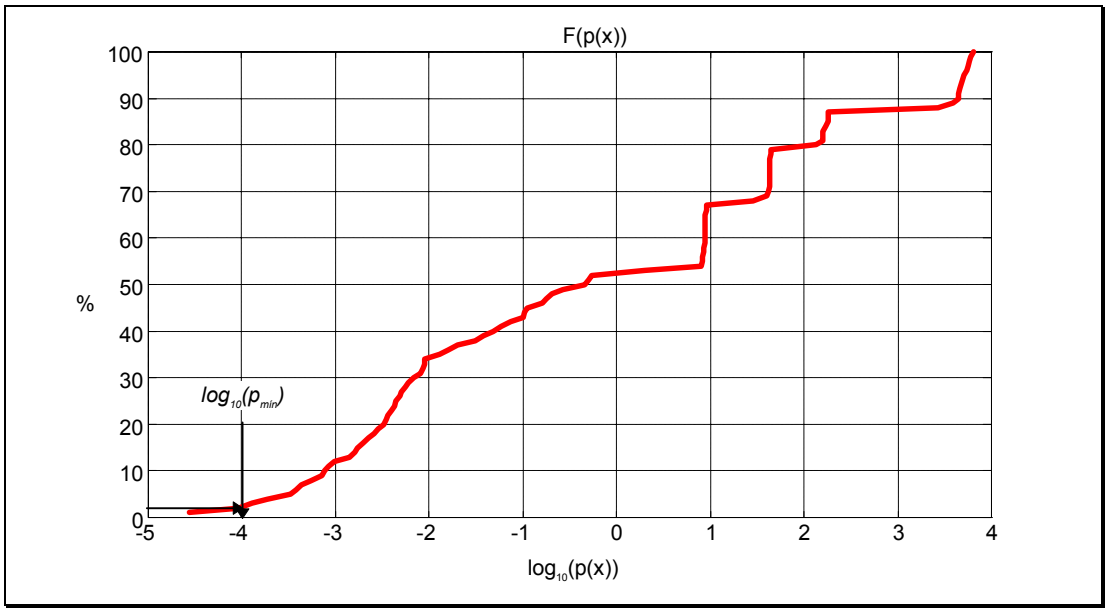

*Figura 6.27: Función de distribución de p(**x**) estimada sobre el conjunto de entrenamiento. Determinación de la cota de extrapolación.*

Como cota de extrapolación se tomó aquél valor de $p(\mathbf{x})$ tal que el 98% de los ejemplos de entrenamiento produjesen un valor de $p(\mathbf{x})$ superior a esta cota. El valor así obtenido fue $p_{mín}=10^{-4}$.

Una vez establecida la cota de extrapolación, podemos representar la región de confianza resultante restringiéndonos a situaciones de régimen permanente en las que se cumple:

$$PG[k-4]=PG[k-3]=PG[k-2]=PG \quad y \quad CCcond[k-2]=CCcond[k-1]=CCcond$$

Ecuación 6.22

De esta forma reducimos las cinco dimensiones del espacio de entrada a dos, permitiendo su representación gráfica, como muestra la Figura 6.28:

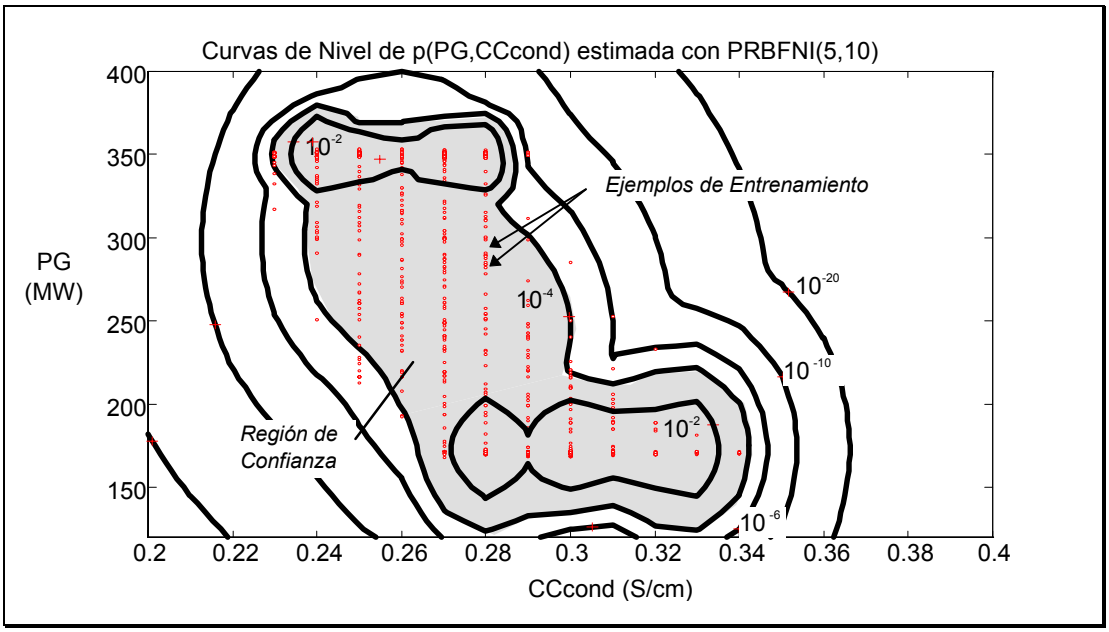

*Figura 6.28: Curvas de nivel de la fdp p(**x**) estimada con PRBFNI. Datos correspondientes a régimen permanente*

Esta figura toma en este caso una especial relevancia, al ser el modelo de funcionamiento normal un modelo de tipo autorregresivo, en el que algunas de las entradas son valores retrasados de las salidas medidas. Esta circunstancia implica que la región de confianza así obtenida coincida con el área de funcionamiento normal en régimen permanente, y que la fdp del vector de entradas ***x*** sea en sí misma una medida del grado de normalidad de la situación descrita por el vector ***x***.

La Figura 6.28 refleja claramente cómo las dos regiones de funcionamiento más habituales ($p(\mathbf{x})>0.01$) corresponden a las situaciones de plena y mínima carga, y pone de manifiesto la especial vulnerabilidad a la corrosión del régimen de mínima carga en el que son de esperar valores más elevados de conductividad catiónica (entre 0.28 S/cm y 0.23 S/cm frente a los de plena carga que varían entre 0.24 S/cm y 0.28 S/cm).

La condición: $p(\mathbf{x})<p_{mín}$ puede entonces ser interpretada como una evolución anómala de la conductividad catiónica de condensado en los instantes (*k-2,k-1*), dada la evolución de la potencia generada en los instantes (*k-4,k-3,k-2*).

c) Modelo de la cota máxima de los residuos

Una vez ajustado el modelo de funcionamiento normal y el modelo de la fdp $p(\boldsymbol{x})$, sólo queda por poner a punto el estimador de la cota máxima de los residuos. Para ello haremos uso de la red PRBFN tipo I utilizada para estimar $p(\boldsymbol{x})$, estimando de forma local la varianza del error de estimación $e(\boldsymbol{x})$ como se vió en el apartado 6.2.4

Tomaremos nuevamente como cota máxima del error de estimación el doble de la desviación típica estimada. La Figura 6.29 muestra los resultados obtenidos en condiciones de régimen permanente.

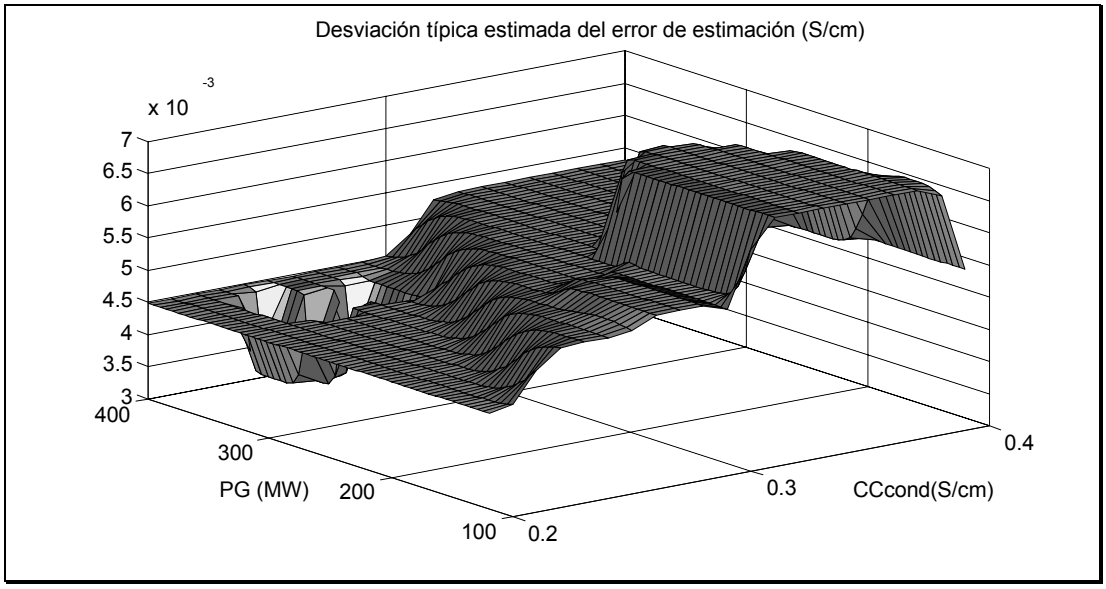

Figura 6.29: Desviación típica estimada del error de estimación de la conductividad catiónica de condensado. Datos correspondientes a condiciones de régimen permanente

Es importante recordar que esta estimación sólo tiene validez en la región de confianza mostrada en la Figura 6.28 (en la que $p(\boldsymbol{x})>p_{mín}$). En esta región puede comprobarse cómo la desviación típica del error decrece con la potencia generada y crece con la conductividad catiónica de condensado. Su máximo se alcanza en la zona más peligrosa, es decir, en la región de mínima carga donde se dan los valores más elevados de CC_{cond}.

6.6.4 Resultados

A continuación se presentan las bandas de funcionamiento normal de la conductividad catiónica de condensado y los valores estimados de $p(\mathbf{x})$ para distintos conjuntos de datos. Al tratarse en este caso de un modelo de funcionamiento normal con componentes autorregresivas, y suponiendo que todas las condiciones de funcionamiento normal han quedado representadas en el conjunto de entrenamiento, podremos interpretar como anómalas todas aquellas situaciones que produzcan un valor de la fdp $p(\mathbf{x})$ por debajo de la cota de extrapolación $p_{mín}$.

En la Figura 6.30 aparecen las estimaciones correspondientes a datos contenidos en el conjunto de entrenamiento. Como era de esperar, la CC_{cond} medida entra dentro de las bandas de funcionamiento normal y la fdp $p(\mathbf{x})$ queda por encima de la cota de extrapolación salvo en ciertas ocasiones puntuales.

Lo mismo ocurre con el conjunto de test, representado en la Figura 6.31. Ambas situaciones son ejemplo del comportamiento del sistema de detección de anomalías en condiciones de funcionamiento normal.

Para ilustrar el comportamiento del sistema en condiciones de funcionamiento anómalo, se ha ensayado el sistema con dos conjuntos de datos que el personal químico de la central consideraban como característicos de dos situaciones de anomalía distintas.

El primero de ellos queda representado en la Figura 6.32. Como puede comprobarse, estos datos corresponden al restablecimiento de la normalidad tras un periodo de funcionamiento anómalo en el que no se llegan a alcanzar valores extremadamente elevados de conductividad catiónica de condensado. El modelo de funcionamiento normal determina la vuelta a la normalidad en torno a la muestra 80, pero este restablecimiento no puede ser dado por bueno hasta la muestra 95, momento en el cual la fdp $p(\mathbf{x})$ alcanza su cota mínima $p_{mín}$.

La Figura 6.33 muestra una situación mucho más crítica que la anterior, en la que se alcanzan valores muy elevados de conductividad catiónica de condensado. Esta situación podría corresponder muy probablemente a una entrada de aire en el condensador de la central. La condición de normalidad se restablece con una constante de tiempo muy lenta, no llegándose a la normalidad hasta el final de la ventana de tiempo mostrada.

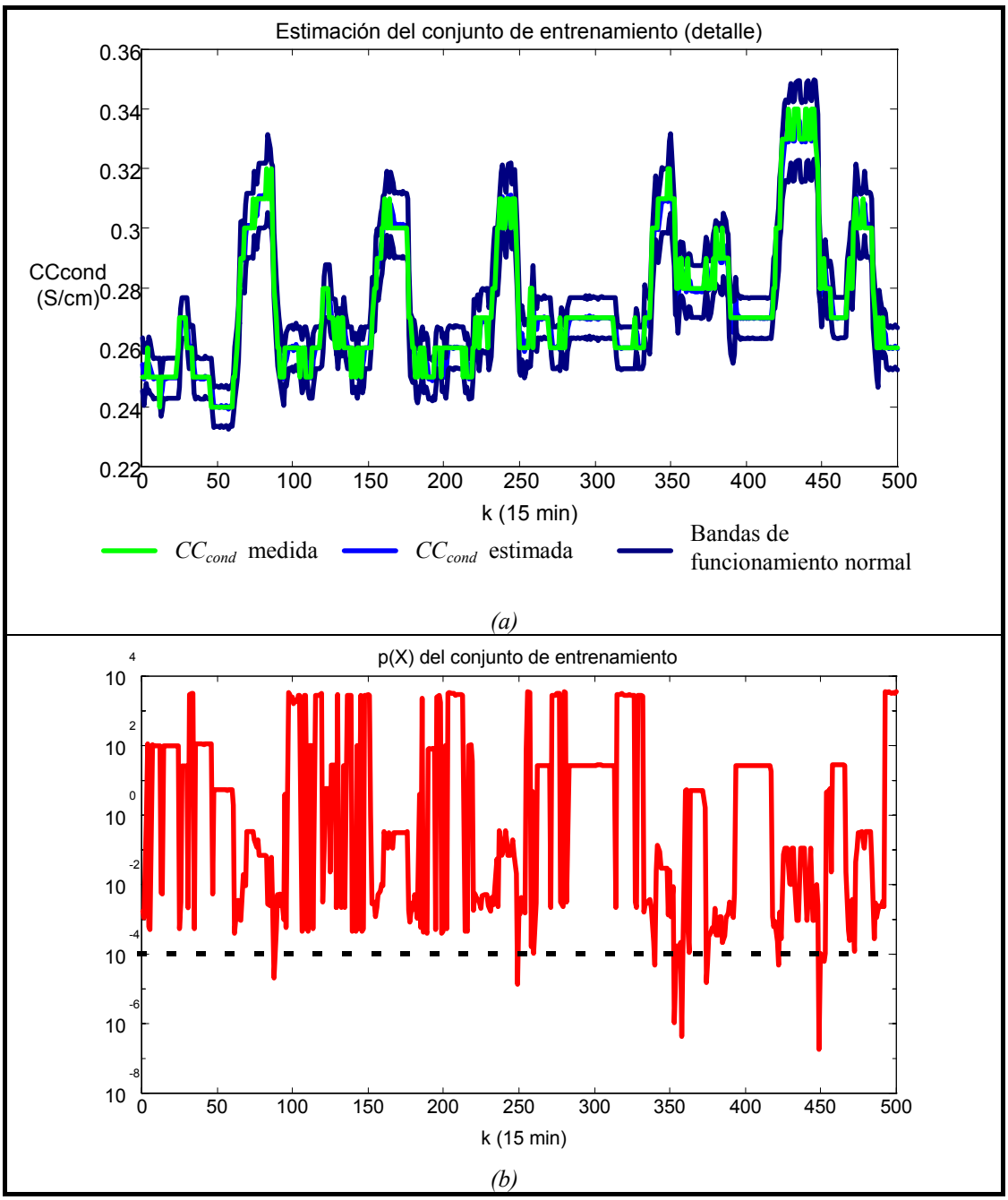

*Figura 6.30: (a) Bandas de funcionamiento normal y (b) fdp p(**x**) estimadas para el conjunto de entrenamiento.*

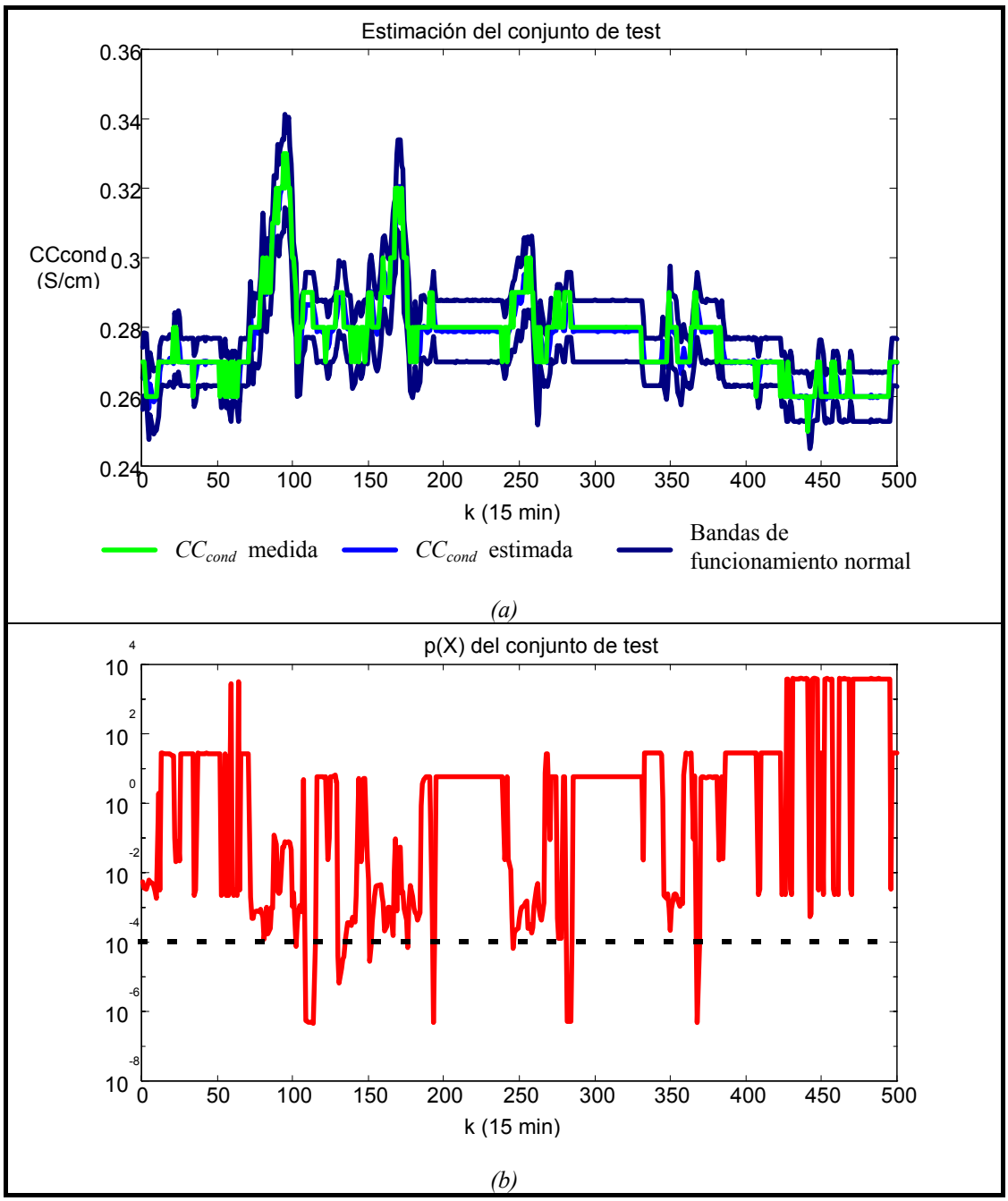

*Figura 6.31: (a) Bandas de funcionamiento normal y (b) fdp p(**x**) estimadas para el conjunto de test*

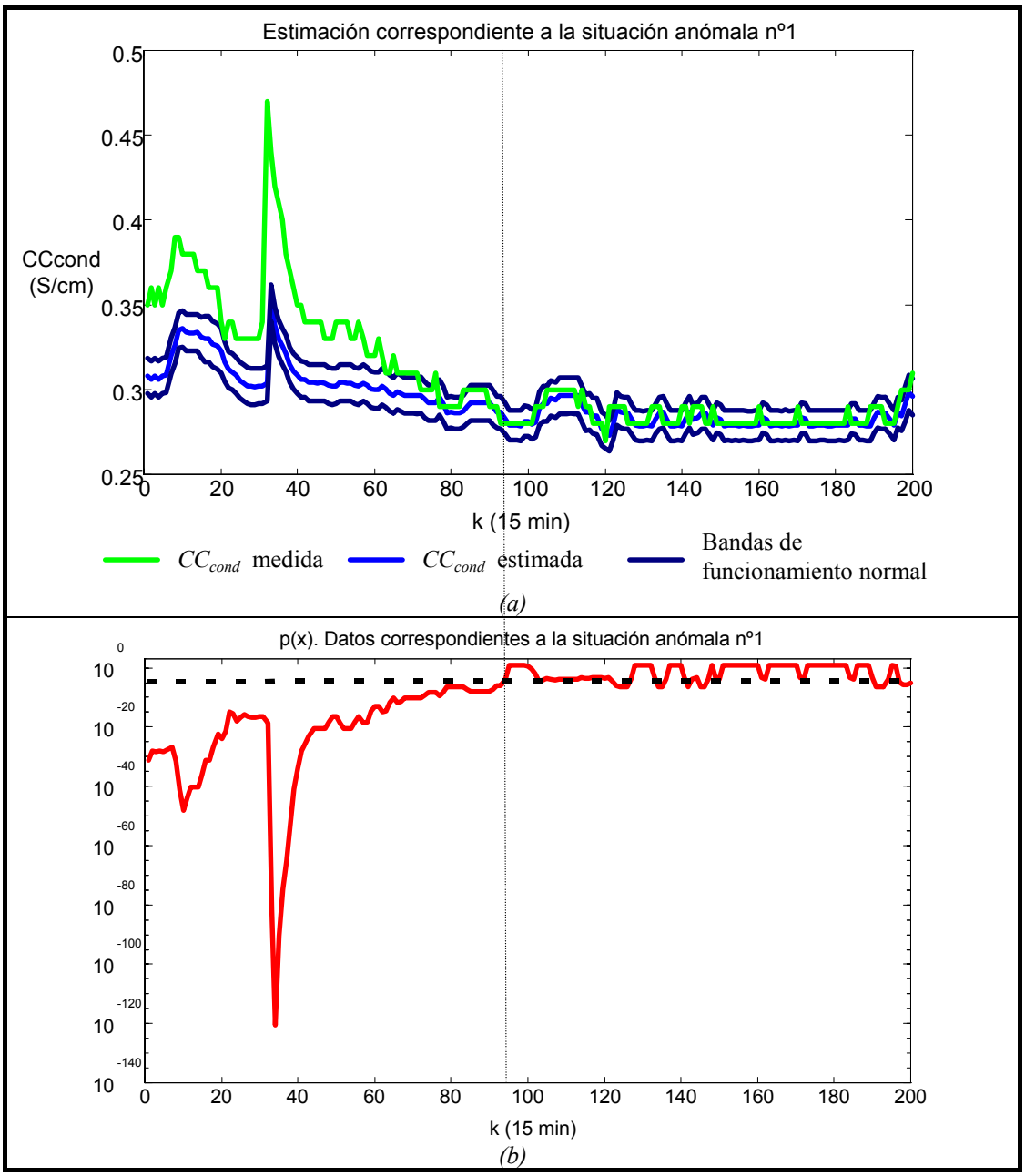

*Figura 6.32: (a) Bandas de funcionamiento normal y (b) fdp p(**x**) estimadas para la situación anómala nº1*

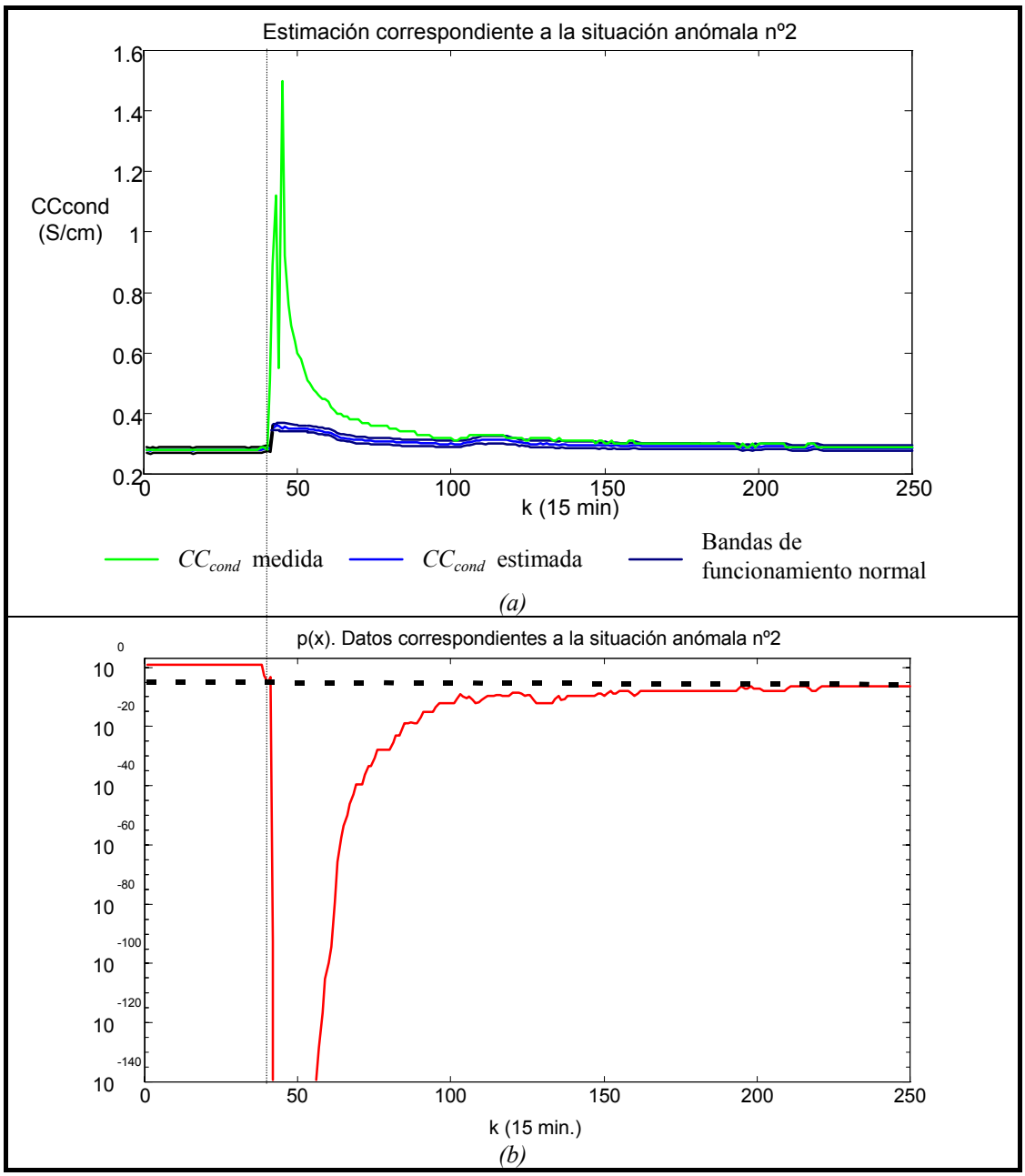

*Figura 6.33: (a) Bandas de funcionamiento normal y (b) fdp p(**x**) estimadas para la situación anómala nº2*

7. Conclusiones, aportaciones y líneas de futuros desarrollos

En este capítulo final de la tesis se presentarán en primer lugar las conclusiones metodológicas y específicas más significativas del trabajo realizado. Posteriormente se enunciarán las aportaciones más relevantes y finalmente se expondrán algunas líneas de futuros desarrollos que han quedado abiertas por el trabajo desarrollado.

7.1 Conclusiones

7.1.1 Metodológicas

En esta tesis se ha abordado la aplicación de técnicas de Redes Neuronales Artificiales al diagnóstico de procesos industriales, presentando un sistema de detección de anomalías incipientes basado en el modelado conexionista del comportamiento normal de los componentes.

El sistema presentado está especialmente dirigido a resolver el problema de la detección de anomalías en aquellos casos en los que no existe una completa base de datos de fallo, y en los que el modelado físico del comportamiento de los componentes resulta inviable. La experiencia demuestra que ésta suele ser la situación más común en los casos reales de diagnóstico de procesos industriales complejos.

Bajo estas circunstancias, la solución más adecuada consiste en caracterizar el comportamiento normal de los componentes a partir de una base de datos de funcionamiento normal que sea suficientemente representativa de la dinámica del proceso bajo las distintas condiciones de trabajo.

La solución propuesta para esta caracterización está basada en la obtención de modelos de funcionamiento normal de los componentes involucrados, mediante técnicas de modelado de procesos dinámicos no lineales con aproximadores funcionales.

Estas técnicas de modelado están inspiradas de la teoría clásica de identificación de sistemas, consiguiendo la característica no lineal mediante la sustitución de la transformación lineal del vector de regresores por un aproximador funcional no lineal.

Como aproximadores funcionales se propone utilizar Redes Neuronales Artificiales supervisadas, tales como el Perceptrón Multicapa y la red PRBFN. Estas herramientas, además de ofrecer una elevada capacidad de representación, poseen una estructura modular que las hacen altamente paralelizables y realizables en *"hardware"*.

Una vez que el modelo de funcionamiento normal de cada componente ha sido identificado y ajustado, la caracterización del comportamiento normal se consigue delimitando la región de confianza de cada modelo y estableciendo las cotas máximas admisibles de los residuos de la estimación.

Para delimitar la región de confianza de cada modelo, se propone utilizar una red PRBFN para estimar la función de densidad probabilista según la cual se distribuye el vector de regresores en el conjunto de entrenamiento utilizado para el ajuste del modelo. Esta misma red puede ser utilizada posteriormente para estimar de forma local la varianza de los residuos en condiciones de funcionamiento normal, obteniendo de esta forma las cotas máximas admisibles de los errores de estimación, y por consiguiente las bandas de funcionamiento normal.

El sistema de detección de anomalías resultante será por lo tanto capaz de identificar por sí mismo aquellas condiciones de operación, que por su novedad, no pueden ser tratadas con un grado de fiabilidad adecuado. La estimación de la cota máxima de los residuos en función del punto de operación permite además ajustar la sensibilidad del sistema de detección a las características propias de cada punto de operación. Estas dos propiedades, unidas a la intrínseca capacidad de adaptación del sistema y a la versatibilidad de los modelos de funcionamiento normal, constituyen las principales prestaciones del sistema de detección de anomalías presentado.

Las dos aplicaciones del sistema de detección de anomalías a casos reales de diagnóstico que fueron presentadas en el Capítulo 6 ilustran la flexibilidad, la eficacia y la viabilidad del método propuesto.

7.1.2 Específicas

a) Aproximación funcional

El esquema general de ajuste de aproximadores funcionales presentado en el Capítulo 2 puso de manifiesto la necesidad de realizar un seguimiento de la capacidad de generalización del aproximador durante su ajuste. Este seguimiento puede realizarse mediante la evaluación del aproximador sobre un conjunto de datos no utilizados directamente en la optimización paramétrica: el conjunto de test.

Esta práctica permite dividir el proceso de ajuste en dos optimizaciones parciales: la optimización estructural y la optimización paramétrica. La primera de ellas tiene como objetivo establecer la óptima estructura del aproximador, y se realiza incrementando de forma iterativa la capacidad de representación funcional del aproximador hasta hallar un máximo en su capacidad de generalización. Esta estrategia de búsqueda está basada en el siguiente principio: “A medida que la capacidad de representación funcional del aproximador funcional va creciendo, la capacidad de generalización del mencionado sistema, ajustado sobre el mismo conjunto de entrenamiento, se hace más incierta.”

Dada una estructura determinada, la optimización paramétrica tiene como objetivo hallar el óptimo vector de parámetros del aproximador. Este problema se ha planteado como un problema clásico de optimización no lineal sin restricciones, tomando como función de error a minimizar el error cuadrático medio de la estimación del conjunto de entrenamiento. Este criterio permite dirigir la búsqueda en el espacio de parámetros, pero será nuevamente la capacidad de generalización del aproximador estimada sobre el conjunto de test la que imponga los criterios de finalización de la misma. Como método de optimización no lineal sin restricciones para aquellos problemas de tamaño medio (del orden del centenar de variables a optimizar y del millar de ejemplos de entrenamiento) se propone utilizar un algoritmo quasi-Newton de "memoria reducida" dotado de una eficaz búsqueda unidimensional. Este método deja sin embargo de ser eficaz cuando el número de ejemplos es demasiado elevado, o cuando la evaluación de la función de error y de su derivada consume demasiados recursos. En estos casos resulta más conveniente utilizar métodos de revisión por caso, como la Regla Delta, en los que se actualiza el vector de variables a optimizar tras cada presentación de ejemplo. En cualquier caso es importante prestar una especial atención al escalado del problema, lo que puede conseguirse sin más que estandarizar las entradas del aproximador.

b) Aplicación de Redes Neuronales Artificiales a la aproximación funcional y la estimación de funciones de densidad

Como aproximadores funcionales se propone en esta tesis utilizar Redes Neuronales Artificiales del tipo supervisado, como el Perceptrón Multicapa y la red PRBFN. En ambos casos se han planteado estrategias de inicialización de los pesos que aceleran de forma significativa el proceso de aprendizaje. El primero de ellos realiza una aproximación de tipo global de la transformación entrada/salida, lo que favorece su capacidad de generalización en las regiones del espacio de entrada no representadas en el conjunto de entrenamiento. El segundo de ellos sin embargo realiza una aproximación de tipo local. Esta propiedad limita su capacidad de generalización a las regiones del espacio de entrada representadas en el conjunto de entrenamiento, por lo que resulta fundamental que la estimación ofrecida por estos estimadores se vea acompañada de una clara delimitación de su región de confianza.

La principal ventaja de la red PRBFN frente al oscuro Perceptrón Multicapa es que cada uno de sus parámetros tiene una interpretación clara en términos probabilistas. Esta red deja de ser una caja negra en la que resulta difícil interpretar cada una de sus señales internas, permitiendo de esta forma la extracción de conocimiento a partir del análisis de sus parámetros una vez que han sido ajustados. La característica de localidad permite además que la adición de nuevas unidades ocultas, que se realiza durante la optimización estructural, se haga allí donde sea necesario, es decir, en las regiones del espacio de entrada donde el error es mayor.

La red PRBFN permite además estimar la función de densidad según la cual se distribuye su vector de entradas en el conjunto de entrenamiento. Para ello es necesario que los pesos de su capa oculta hayan sido ajustados siguiendo un proceso de maximización de la verosimilitud logarítmica de los ejemplos de entremiento. Esta capacidad hace de la red PRBFN una pieza clave en el sistema de detección de anomalías propuesto ya que, además de poder ser utilizada como aproximador funcional en los modelos de funcionamiento normal, servirá para delimitar la región de confianza de estos modelos y las cotas máximas de sus residuos.

c) Modelado de procesos dinámicos no lineales con aproximadores funcionales

Como directa extensión de los modelos lineales al caso no lineal, fueron presentadas en el Capítulo 5 las arquitecturas de modelado de procesos dinámicos no lineales basadas en aproximadores funcionales. Las principales conclusiones de este capítulo se detallan a continuación.

En primer lugar es importante destacar que todo el conocimiento *a priori* que se tenga del proceso deberá ser incluido directamente en el modelo, especialmente en la etapa de selección de las variables de entrada.

En segundo lugar recordar la máxima de "probar primero lo más sencillo". Según este criterio se ensayarán primero los modelos lineales, haciendo uso de técnicas de identificación de sistemas bien establecidas. Si se detectasen características no lineales en el proceso, el siguiente modelo a ensayar sería el modelo NARX (modelo no lineal autoregresivo con entradas exógenas). Este modelo suele bastar en la mayoría de los casos y presenta la gran ventaja de no ser un modelo recurrente. pero si se detectase cierto comportamiento predecible en la serie de los residuos, será conveniente evaluar el comportamiento de los modelos NARMAX (modelo no lineal autoregresivo de media móvil con entradas exógenas).

Una herramienta fundamental para la selección de las variables de entrada de los modelos no lineales es el Análisis Estadístico de Sensibilidades. Este análisis permite identificar aquellas variables de entrada que no tienen influencia en las salidas de los modelos ajustados. La eliminación de estas variables reduce la complejidad artificial del modelo, aumentando generalmente su capacidad de generalización.

Otro punto importante es la validación de los modelos ajustados. A este efecto se recomienda el método de validación cruzada. Este método de validación tiene la gran ventaja de no haber tenido que hacer hipótesis sobre las distribuciones probabilísticas de las variables aleatorias involucradas. Su desventaja más palpable es tener que dejar aparte un conjunto de datos, y no utilizarlo para la estimación de los parámetros. Esta desventaja desaparece en el caso de disponer de un conjunto de

entrenamiento suficientemente rico, como suele ser el caso en la obtención de modelos de comportamiento normal en el ámbito del diagnóstico.

d) Sistema de detección de anomalías incipientes basado en el modelado conexionista del comportamiento normal de los componentes.

Además de las conclusiones metodológicas expuestas en el primer apartado de este capítulo, es importante señalar las conclusiones de tipo específico que se exponen a continuación.

La selección del conjunto de datos que será utilizado para el ajuste de los modelos de funcionamiento normal es un punto fundamental a la hora del desarrollo y puesta al día del sistema de detección de anomalías. Por un lado hay que prestar una especial atención para que este conjunto no contenga datos correspondientes a situaciones anómalas. Por otro lado resulta fundamental el que los datos queden repartidos de forma más o menos homogénea en el espacio de entrada, de tal modo que todos los puntos de operación del proceso y las transiciones de uno a otro queden representadas en el conjunto de entrenamiento. Los árboles de selección de datos permiten realizar esta tarea de una forma eficaz, facilitando además la continua puesta al día de estas base de datos para la revisión de los modelos.

Los modelos de funcionamiento normal pueden obtenerse aplicando las técnicas de modelado de procesos dinámicos no lineales presentadas en el Capítulo 5, utilizando como aproximadores funcionales Redes Neuronales Artificiales supervisadas como las presentadas en los Capítulos 3 y 4. El ajuste de estos aproximadores según el esquema propuesto en el Capítulo 2 permitirá alcanzar los requisitos de precisión en la estimación necesarios.

La delimitación de las regiones de confianza de los modelos de funcionamiento normal puede realizarse mediante la estimación de la función de densidad probabilista según la cual se distribuyen los vectores de entrada en sus respectivos conjuntos de entrenamiento. La región de confianza de un modelo de funcionamiento normal puede interpretarse como aquella región del espacio de entrada en la que el modelo es válido, entendiendo por modelo válido en un punto el que sus residuos en ese punto hayan quedado caracterizados en sentido estadístico (a partir de las muestras del conjunto de entrenamiento). La delimitación de estas regiones de confianza juega un papel fundamental en la tarea de detección, ya que permite descubrir nuevos puntos de operación y evita el dar falsas alarmas por invalidez de los modelos.

La estimación de la varianza residual de cada modelo de funcionamiento normal en función de sus vector de entradas, permite ajustar la sensibilidad del sistema de detección a las características propias de cada punto de operación. Esta estimación

puede ser realizada a partir de las varianzas residuales locales estimadas en cada unidad radial de la red PRBFN estimadora de la fdp del vector de entradas.

La conjunción de los tres estimadores (el modelo de funcionamiento normal, el estimador de la fdp del vector de entradas y el estimador de la cota máxima de los residuos), forma un eficaz y robusto sistema de detección de anomalías cuya aplicabilidad a casos reales de diagnóstico ha quedado ya plasmada en diversas aplicaciones.

7.2 Aportaciones

La principal aportación de esta tesis ha sido la de proponer una solución al problema de la detección de anomalías en procesos industriales complejos, cuando no existe una completa base de datos de fallo y el modelado físico de los componentes resulta inviable. En este sentido ha sido necesario relacionar distintos campos de la técnica y de la ciencia como son el diagnóstico, las redes neuronales artificiales, la identificación de sistemas, la teoría de series temporales, la teoría de aprendizaje, y la optimización. A continuación se detallan las aportaciones más significativas.

- Estructura del sistema de detección de anomalías incipientes

La aportación más importante de esta tesis es la propia estructura del sistema de detección de anomalías que combina la acción de tres estimadores conexionistas: el modelo de funcionamiento normal, el estimador de la fdp del vector de entradas y el estimador de la cota máxima de los residuos. Esta estructura, inspirada de los trabajos de J.A.Leonard y de M.A.Kramer en el área de diagnóstico de procesos químicos, presenta innovaciones importantes íntimamente ligadas a la incorporación de la red PRBFN. Entre ellas cabe destacar la utilización de un estimador de la función de densidad probabilista para la determinación de la región de confianza de los modelos de funcionamiento normal, y la utilización de esta misma estructura para la estimación de las cotas máximas de los residuos. Asimismo es de destacar la aplicación de técnicas de modelado de procesos dinámicos no lineales basadas en aproximadores funcionales para la obtención de los modelos de funcionamiento normal. Un aspecto importante en la aplicación de estos modelos es el esquema de ajuste de aproximadores funcionales presentado en el Capítulo 2.

- Arboles de selección de datos

Dada la importancia prestada a la selección de los conjuntos de entrenamiento y test de los modelos de funcionamiento normal, fue necesario desarrollar una nueva estructura de almacenamiento dinámico de información que cumpliese los requisitos previstos. En concreto se pretendía que la base de datos resultante fuese capaz de

actualizarse de forma secuencial con los datos que le irían llegando de forma continua. La actualización de la base de datos, de tamaño predefinido, debía realizarse atendiendo a dos criterios de optimalidad simultáneos: por una lado había que dar prioridad a los datos más recientes, y por otro había que conseguir un muestreo homogéneo del espacio muestral. La solución propuesta a este problema son los árboles de selección de datos presentados en el Apéndice A.1.

- La red PRBFN

La red PRBFN, que ha sido presentada como una extensión de la red GRNN de D.Specht, ha sido dotada de mayor flexibilidad y de mejores leyes de aprendizaje. El aumento de flexibilidad se ha conseguido mediante la inclusión de nuevos factores de escala en las unidades radiales. Esta medida aumenta el número de parámetros libres a optimizar durante el aprendizaje, pero disminuye significativamente el número de unidades radiales requeridas. La rígida ley de aprendizaje de la red GRNN ha sido sustituida en la red PRBFN por una efectiva inicialización de sus pesos y por su adaptación al esquema de ajuste de aproximadores funcionales presentado en el Capítulo 2. La red PRBFN, además de poder ser utilizada como aproximador funcioonal, puede también ser utilizada como estimadora de funciones de densidad probabilista, optimizando los pesos de su capa oculta mediante el criterio de máxima verosimilitud logarítmica.

- Algoritmo Mixto de Agrupamiento

Este algoritmo ha sido especialmente diseñado para inicializar los centros o representantes de las unidades radiales de la red PRBFN. El fin que se persigue con este algoritmo es que los representantes queden finalmente dispuestos en el subespacio de entrada representado por el conjunto de entrenamiento, de la forma más homogénea posible. Esto se consigue mediante la aplicación de dos algoritmos de agrupamiento basados en criterios distintos: el algoritmo *"leader"*, que selecciona las muestras más distantes entre sí, y el algoritmo *"k-means"*, que centra los representantes seleccionados en los centros geométricos de sus zonas de influencia.

- Análisis Estadístico de Sensibilidades

Una vez que un aproximador funcional ha sido ajustado a partir de un conjunto de muestras de la relación de entrada/salida que se pretende modelar, es posible evaluar el efecto que cada una de las entradas del aproximador tiene en su salida mediante el análisis estadístico de las sensibilidades correspondientes. Este análisis permite identificar las variables de entrada que no tienen influencia en la salida del aproximador, con el fin de que sean eliminadas para reducir la complejidad del modelo y aumentar su capacidad de generalización. Este análisis juega un papel fundamental en la selección de las variables de entrada de los modelos de funcionamiento normal, facilitando la tarea de identificación de sistemas no lineales.

El método propuesto bajo el nombre de Análisis Estadístico de Sensibilidades ha sido desarrollado en estrecha colaboración con mi colega T.Cernichow.

7.3 Líneas de futuros desarrollos

A continuación se presentan las principales líneas de desarrollo que han quedado abiertas en esta tesis:

- Integración del sistema de detección de anomalías en un sistema experto de diagnóstico

La línea de desarrollo más importante que ha quedado abierta es la integración del sistema de detección de anomalías presentado en un sistema experto de diagnóstico. Esta integración tendría como objetivo completar las tareas de diagnóstico que han quedado pendientes: identificar las causas que han producido las anomalías detectadas y proponer las acciones correctoras pertinentes. El sistema experto debería ser capaz de aprovechar la información suministrada por el sistema de detección de anomalías en términos de componentes afectados, niveles de sobrepaso de las bandas de funcionamiento normal y grados de pertenencia a las regiones de confianza de los modelos. La interpretación de esta información podría realizarse de forma más natural en términos de lógica borrosa, para poder interpretar de forma continua distintos grados de sobrepaso de los niveles de confianza y distintos grados de pertenencia a la región de confianza de los modelos. El sistema experto estaría también encargado de gestionar la continua adaptación de los modelos del sistema de detección, como se mencionó en el Capítulo 6. Una vez integrados ambos sistemas, podrían pasar a formar parte integral de la metodología de mantenimiento predictivo propuesta por el director de esta tesis en *[Sanz Bobi, 1992]*.

- Generación automática de patrones de fallo

Otra línea de desarrollo importante que ha quedado pendiente es la generación automática de patrones de fallo según se van detectando anomalías. A este efecto podrían aplicarse Redes Neuronales Artificiales no supervisadas, como los mapas autoorganizativos de Kohonen (*[Kohonen, 1990]*), para la caracterización de las evoluciones de las variables en condiciones anómalas. Una vez generados los patrones de fallo, éstos podrían asociarse a las causas que los producen, convirtiendo el problema de aislamiento de anomalías en un problema de reconocimiento de patrones.

- Aplicaciones médicas de la metodología de detección de anomalías desarrollada

Las técnicas desarrolladas en esta tesis para la caracterización de comportamientos pueden ser utilizadas como herramientas de diagnóstico médico. De hecho la medicina ha sido y sigue siendo uno de los grandes campos de aplicación de las RNA. Entre las numerosas aplicaciones de las RNA en el área de la bio-ingeniería podemos citar el análisis de electrocardiogramas, el análisis de electro-encefalogramas, la detección de cánceres, la detección de lesiones cerebrales, etc. La aplicación de la metodología de detección de anomalías propuesta en esta tesis permitiría desarrollar sistemas de diagnóstico médico encargados de comprobar el cumplimiento de las relaciones que ligan unas "variables vitales" con otras en ausencia de enfermedades.

- Aplicación de las técnicas de modelado de procesos dinámicos no lineales a la predicción de series temporales

Las mismas técnicas que se han aplicado para la obtención de los modelos de funcionamiento normal podrían aplicarse a la predicción de series temporales. El esquema de estimación planteado, además de la predicción, podría suministrar la región de confianza de la predicción y sus márgenes de error. Como posibles campos de aplicación dentro del área de energía eléctrica caben destacar la predicción de la demanda eléctrica y la predicción de series hidrológicas para la coordinación hidrotérmica. Otro área donde están despertando gran interés estas técnicas es el área de análisis financiero.

A. Árboles de selección de datos

Este apéndice describe una estructura de almacenamiento y de selección de datos que permitirá filtrar de forma secuencial las muestras disponibles para obtener un subconjunto de datos uniformemente repartidos en el espacio muestral, dando prioridad a las muestras más recientes.

Esta herramienta podrá ser utilizada para seleccionar los conjuntos de entrenamiento y de test de los modelos de funcionamiento normal, y permitirá actualizar periódicamente los modelos ajustados con los nuevos datos que han sido incorporados a la base.

A.1 Introducción

El objetivo de los árboles de selección de datos es proporcionar un método eficaz de mantenimiento dinámico de una base de datos de tamaño predefinido, formada por K datos representativos del espacio muestral. Es en definitiva una herramienta de muestreo selectivo que permitirá seleccionar de forma secuencial un conjunto de datos representativo de la población, entendiendo por representativo el que cubran de forma uniforme u homogénea el espacio muestral.

La eficacia de este método radica en su capacidad para revisar la base de datos sin necesidad de utilizar datos no contenidos en ella, y en la rapidez de posicionamiento de un nuevo dato en la base.

Partiremos de una secuencia fechada de datos enteros n-dimensionales o muestras:

$$S = \{(\mathbf{x}[i],t[i]),\ i=1,2,...\}$$

siendo:
- $\mathbf{x}[i] = [x_1[i],\ x_2[i],\ ...,x_n[i]]^T$ el dato entero n-dimensional nºi
- $t[i]$ la fecha del dato nºi codificada en forma de entero
- el superíndice T el operador transposición

de tal forma que el problema se reduce a actualizar con el par $(\mathbf{x}[i],t[i])$ la base de datos B existente, formada por los $k \leq K$ datos subjetivamente más representativos que han pasado por ella y que denominaremos *representantes muestrales*:

$$B = \{\ (\mathbf{r}(j),t(j)),\ j=1,2,...,k \leq K\}$$

siendo:
- $\mathbf{r}(j) = [r_1(j),\ r_2(j),\ ...,r_n(j)]^T$ el representante nºj
- $t(j)$ la fecha codificada en forma de entero del representante nºj

La codificación entera de los datos no supondrá en general ninguna limitación a la hora de su aplicación práctica, siendo además en muchos casos la codificación natural a la salida de los convertidores A/D. En el caso general bastará con asignar a cada una de las n variables originales (en general reales) x_i^r $(i=1,...,n)$ un rango de variación $[x_i^{min},x_i^{max}]$ y realizar la conversión:

$$x_i = E\left(a_i x_i^r + b_i\right) \qquad \text{con} \qquad \begin{cases} a_i = \dfrac{maxnum}{x_i^{max} - x_i^{min}} \\ b_i = a_i x_i^{min} \end{cases}$$

Ecuación A.1

y siendo $E(x)$ la parte entera de x y $maxnum$ el máximo entero codificable.

De esta forma todas las variables codificadas tendrán el mismo rango de variación (*[0,maxnum]*), habiendo quedado representadas como números enteros positivos.

La elección del rango de variación de las variables originales y de la precisión de la codificación entera (dada por *maxnum*) tendrá un efecto apreciable a la hora de la construcción del árbol, siendo en general perjudicial para nuestros objetivos el definir rangos de variación demasiado amplios que no se vean razonablemente cubiertos por las muestras. Más concretamente, si queremos dar el mismo peso a cada una de las variables, el ratio entre el rango muestral (definido por el máximo y mínimo muestral) y el rango definido debería ser similar para cada una de las variables.

A.2 Estructura

La estructura de los árboles de selección de datos es una estructura jerárquica, organizada en *niveles*, y cuyo elemento fundamental son los llamados *nodos* del árbol. Está inspirada en la codificación binaria entera positiva (cuyo operador será notado [[.]]), que en lógica de 3 bits resultaría:

con $[[x]]_i \in \{ 0 , 1 \}$ siendo i=1 el bit más significativo de la palabra [[x]] de tal forma que podemos :

$$x = [[x]]_1 \, 2^2 + [[x]]_2 \, 2^1 + [[x]]_3 \, 2^0$$

Tomando de una forma general palabras de L bits (de tal forma que $maxnum = 2^L-1$), el subespacio real n-dimensional $[0 , maxnum+1[^n$ quedará dividido por la codificación en $(2^L)^n$ *rejillas* o hipercubos n-dimensionales que contienen los datos que serán codificados de igual forma.

Sin embargo estas rejillas pueden interpretarse como el resultado final de una subdivisión progresiva del subespacio real n-dimensional $[0 , maxnum+1[^n$ en rejillas de tamaño decreciente, que quedan dispuestas jerárquicamente en $(L+1)$ niveles de tal forma que cada rejilla se subdivide en 2^n rejillas en su nivel inferior.

Así por ejemplo si fijamos n=2 y L=3, el espacio bidimensional muestral será $[0,8[^2$, y quedará subdividido en 4 niveles de la forma ilustrada en la Figura A.1:

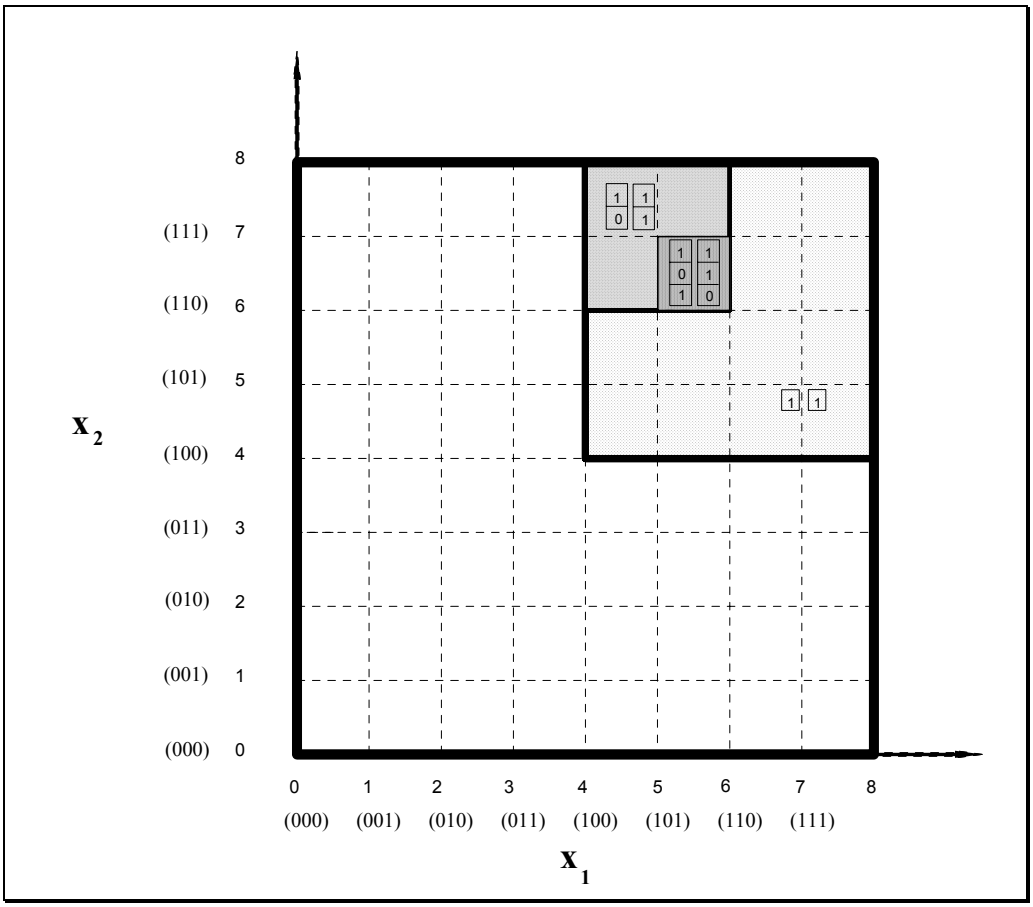

Figura A.1: Subdivisión jerárquica del espacio muestral en rejillas

donde aparece una rejilla de cada uno de los distintos niveles de codificación:

- nivel 0: 1 única rejilla de lado 8: $[0,8[^2$

- nivel 1: 4 rejillas de lado 4 como la $([4,8[\times[4,8[)$ que queda codificada por la tupla $((1),(1))$ equivalente al esquema $((1,x,x)^T,(1,x,x)^T)$

- nivel 2: 16 rejillas de lado 2 como la $([4,6[\times[[6,8[)$ que queda codificada por la tupla $((1,0)^T,(1,1)^T)$ equivalente al esquema $((1,0,x)^T,(1,1,x)^T)$

- nivel 3: 64 rejillas de lado 1 como la $([5,6[\times[[6,7[)$ que queda codificada por la tupla $((1,0,1)^T,(1,1,0)^T)$

Una posible representación en forma de árbol de esta discretización del espacio muestral es la representada en la Figura A.2:

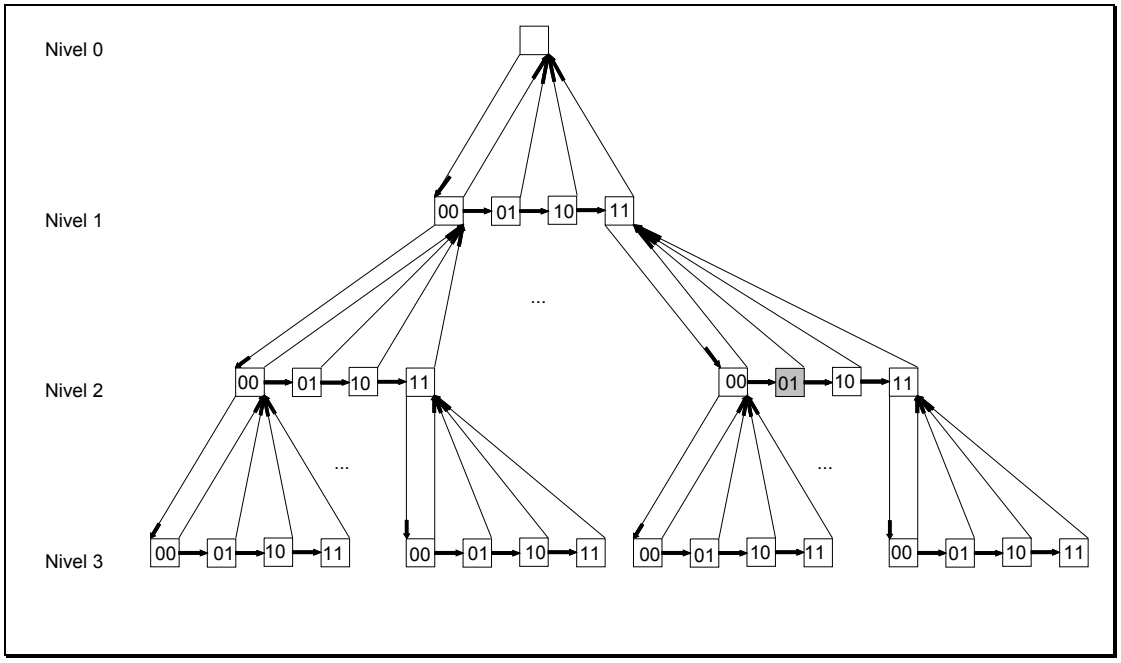

Figura A.2: Representación en forma de árbol de la subdivisión

Cada nodo del árbol de nivel l (l=0,...,L) representa una rejilla de ese nivel y almacena la siguiente información:

- Dirección de su nodo "padre"
- Dirección de su siguiente nodo "hermano"
- Dirección de su primer nodo "hijo"
- Número de muestras contenidas en su rejilla asociada
- l-ésimo nivel de la codificación de la rejilla que queda representa por el nodo (n bits): una rejilla de nivel l quedará identificada por una tupla compuesta por n palabras de l bits, quedando distribuida esta codificación en cada nivel del árbol, de tal forma que un nodo de nivel l hereda de sus (l-1) antecesores inmediatos los (l-1) primeros niveles de codificación de la rejilla que representa. Así por ejemplo la rejilla ([4,6[×[[6,8[) de codificación $((1,0)^T,(1,1)^T)$ estará representada en el árbol por el nodo (0-1) de nivel 2 que cuelga del nodo (1-1) de nivel 1 (correspondiente a la rejilla ([4,8[×[4,8[ ) de codificación ((1),(1))).

Una vez que ha quedado definida la discretización del espacio muestral y su representación en forma de árbol, veamos cómo puede ser utilizada para seleccionar *K* datos representativos de una secuencia de muestras *n*-dimensionales.

En primer lugar hemos de aclarar lo que entendemos por "*K* datos representativos de una secuencia de muestras". Nuestro objetivo no es obtener una sub-muestra aleatoria distribuida según la misma función de distribución que la original, sino obtener un conjunto de *K* datos repartidos lo más uniformemente posible en el espacio muestral. De esta forma los datos seleccionados cubrirán de forma homogénea la mayor extensión posible del espacio muestral.

Ahora bien, si asignamos a cada nodo del árbol un *representante n*-dimensional situado en el centro geométrico de la rejilla que representa, el conjunto formado por los $(2^l)^n$ representantes asociados a los nodos del nivel l será una buena representación de tamaño $(2^l)^n$ del espacio muestral, en el sentido expresado en el párrafo anterior:

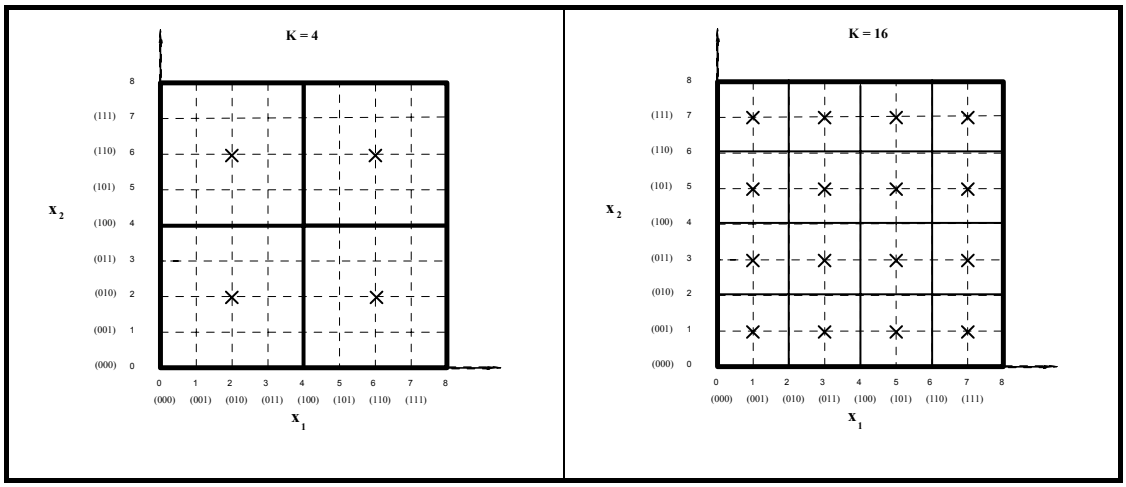

Figura A.3: Situación de los representantes ideales para distintos valores de K

Esta distribución de representantes está limitada a tamaños del tipo $(2^l)^n$ y pensada para cubrir todo el espacio muestral, sin tener en cuenta que sólo parte de espacio se verá cubierto por los datos contenidos en la secuencia de muestras. Por otro lado los representantes son en cierto modo artificiales, ya que no pertenecen a la secuencia de muestras original.

Estas limitaciones dan pie a la reestructuración del árbol de la forma siguiente:

- El árbol contendrá un máximo de K nodos terminales (u *hojas*), no necesariamente del mismo nivel, cuyos representantes serán tomados como representantes de la secuencia de datos.

- Los representantes de cada nodo (terminales y no terminales) serán datos de la secuencia cercanos (según la distancia euclídea) a su representante ideal, localizado en el centro geométrico de la rejilla por él representada.

- Los nodos terminales tendrán la menor profundidad posible en el árbol, de forma tal que los l primeros niveles de la codificación del representante de un nodo terminal del nivel l, sea suficiente para distinguirlo del resto de los representantes de otros nodos terminales.

De esta forma el proceso de selección de los representantes se convierte en un proceso de generación y revisión del árbol, compuesto por los siguientes pasos:

1. Lectura del siguiente dato fechado: $(\mathbf{x}[i],t[i])$

2. Búsqueda del nodo *padre* del nuevo dato, siendo éste el nodo asociado a la rejilla ya creada de menor tamaño que contiene al dato. Si el dato es igual al representante de su nodo padre, y su nodo padre es nodo terminal, ir a paso 4.

3.1 Si $k<K$, siendo k el número de representantes asociados a nodos terminales y K el número total de representantes a seleccionar, el nuevo dato se incorporará al árbol como representante de un nodo terminal. Actualizar $k=k+1$.

3.2. Si $k=K$ (el árbol ya está completo):

3.2.1. Si el nodo más profundo del árbol se encuentra en un nivel inferior al que ocuparía el nodo que admitiese como representante al nuevo dato, se elimina el nodo más profundo y se incorpora el nuevo dato como representante de un nuevo nodo terminal.

3.2.2. En otro caso:

3.2.2.1 Si el nodo padre del nuevo dato es nodo terminal y el nuevo dato es mejor representante de la rejilla asociada al nodo padre que su actual representante, se sustituye el representante.

3.2.2.2 Si el nodo padre del nuevo dato no es nodo terminal (tiene nodos hijos): se busca el nodo hijo que sea terminal y que tenga el representante de peor calidad. Si el nuevo dato diese lugar a una nueva rejilla en la que el dato tuviese una calidad de representanción superior a la que tiene el peor representante hallado, se elimina el nodo con el peor representante y se se incorpora el nuevo dato como representante de un nodo terminal.

4. El nuevo dato se copia como representante de todos aquellos nodos no terminales asociados a rejillas que lo contienen, si la calidad de su representante es inferior a la calidad del nuevo dato como representante de la rejilla en cuestión. Este proceso de búsqueda ascendente en el árbol se aprovecha para actualizar el número de muestras contenidas en la rejilla asociada a cada nodo, incrementándose en una unidad en aquellos nodos no terminales asociados a rejillas que contienen al dato.

   Ir a paso 1.

La calidad de un dato como representante de la rejilla asociada a un nodo deberá potenciar la selección de representantes cercanos al centro geométrico de la rejilla, pero otro tipo de consideraciones suplementarias, como la fecha del dato, permitirán dirigir la selección de los representantes hacia nuestros objetivos.

De esta forma, si utilizamos la siguiente medida de la calidad de un dato fechado *(**x**[i],t[i])* como representante de una rejilla de centro geométrico $\boldsymbol{x}^{opt}$:

$$C((\boldsymbol{x}(i), t(i)) , \boldsymbol{x}^{opt}, t) = - \left\| \boldsymbol{x}(i) - \boldsymbol{x}^{opt} \right\| - \gamma (t - t(i))$$

Ecuación A.2

siendo:

t: la fecha actual expresada en número de meses transcurridos desde cierto origen temporal predeterminado

t[i]: la fecha del dato expresada en número de meses transcurridos desde el origen temporal

γ: factor de ponderación

potenciaremos la selección de representantes del presente mes, repartidos lo más homogéneamente posible en el espacio muestral.

Por último quedan por especificar los procedimientos de incorporación y eliminación de datos como representantes de nodos terminales.

Una vez determinado el nodo padre o nodo asociado a la rejilla ya creada de menor tamaño que contiene al dato que se quiere incorporar como representante de un nodo terminal, pueden presentarse dos situaciones:

- Si el nodo padre es a su vez un nodo terminal (no tiene hijos), éste se expandirá de forma tal que su representante y el nuevo dato sean representantes de nodos hermanos, en el nivel del árbol que permita diferenciarlos. Así por ejemplo, si partimos del siguiente árbol (Figura A.4):

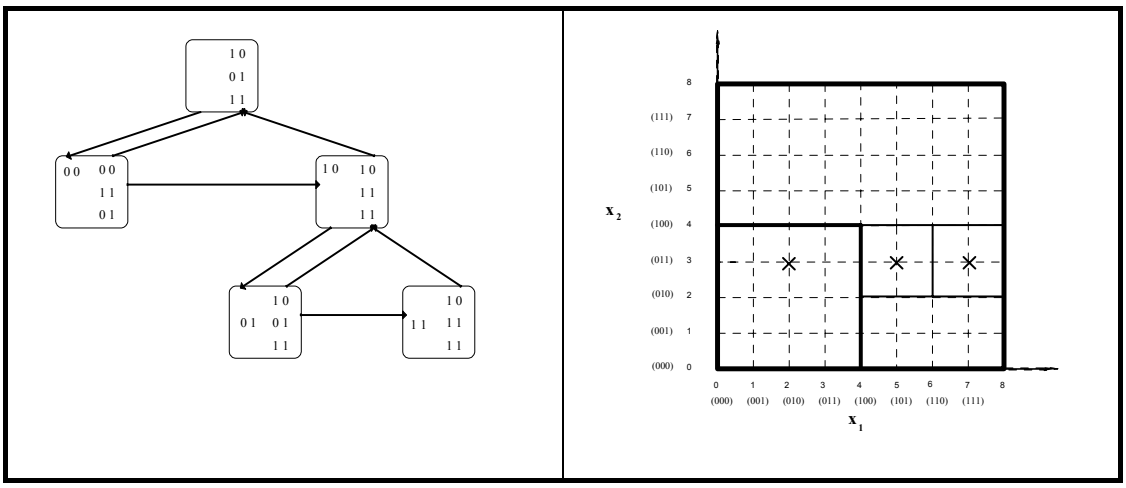

Figura A.4: Ejemplo de incorporación de un nuevo representante. Punto de partida

que contiene los datos representantes marcados con cruces en la gráfica de la derecha y donde cada nodo incluye su nivel de codificación (a la izquierda) y su representante (a la derecha) , al incorporar como representante el dato $((0,1,1)^T,(0,1,0)^T)$, el nodo sombreado de la figura habrá de expandirse dos niveles en profundidad, tal como muestra la Figura A.5:

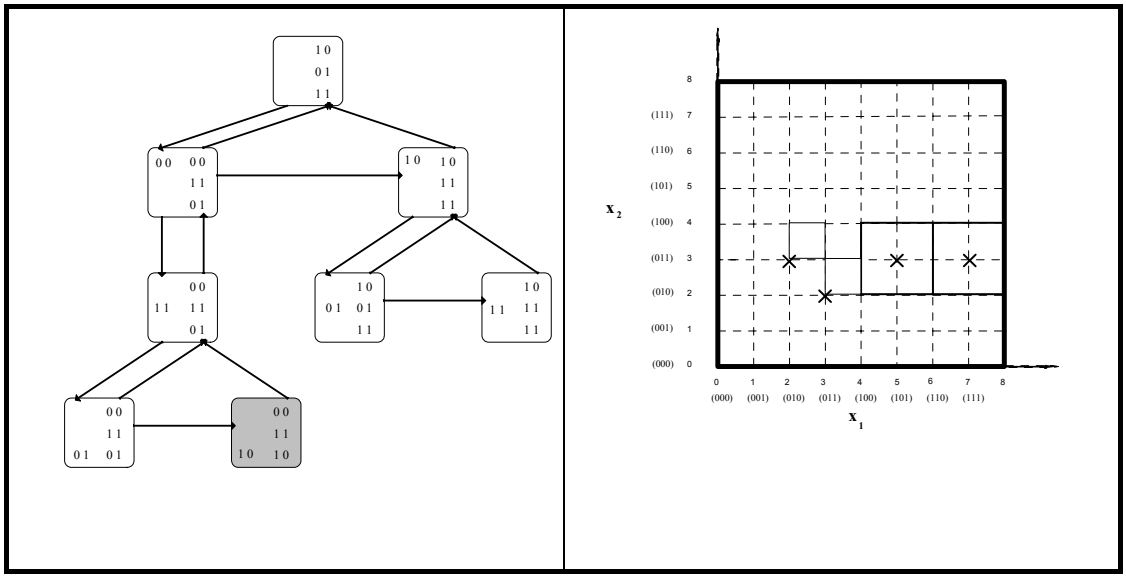

Figura A.5: Ejemplo de incorporación de un nuevo representante. Caso de nodo padre terminal.

- Si por el contrario el nodo padre no era un nodo terminal, el nuevo dato se convertirá en representante de un nuevo nodo hermano de los hijos ya existentes del nodo padre, como sucede en el caso de incorporar como representante el dato ($(1,0,1)^T$,$(0,0,1)^T$) al árbol de la Figura A.4, como muestra la Figura A.6:

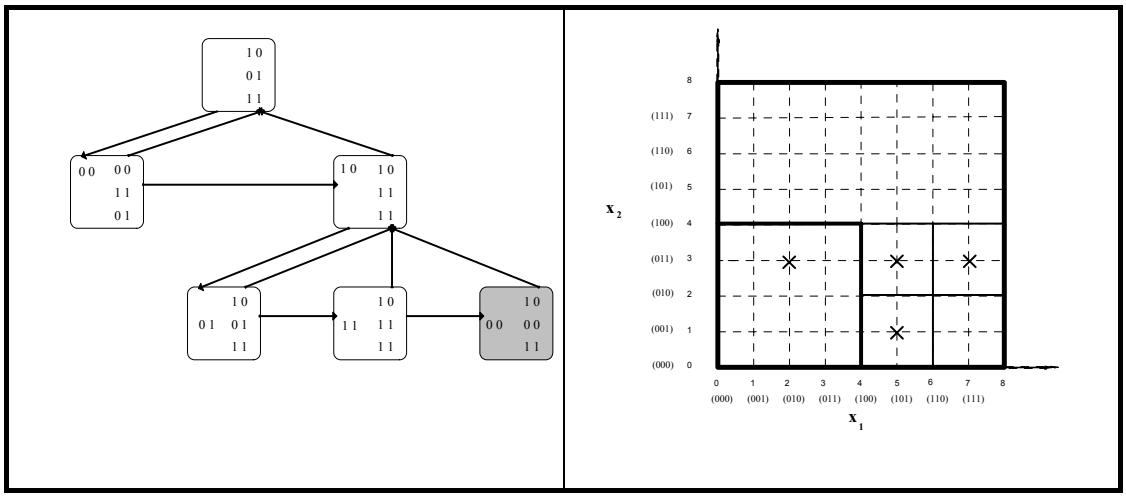

Figura A.6: Ejemplo de incorporación de un nuevo representante. Caso de nodo padre no terminal.

De forma análoga, el procedimiento para eliminar un nodo terminal del árbol contempla dos situaciones distintas:

- Si el nodo terminal a eliminar tiene un único nodo hermano, la eliminación de este nodo supondrá una búsqueda ascendente en el árbol hasta encontrar un antecesor del nodo que tenga al menos otro hermano, y que será transformado en nodo terminal al haber ido eliminando de forma sucesiva todos aquellos nodos antecesores que sólo tenían un hijo. Este caso ha quedado ya ilustrado de forma inversa: si partimos de la situación contemplada en la Figura A.5 y eliminamos el nodo terminal de representante $((0,1,1)^T,(0,1,0)^T)$, llegaremos a la situación mostrada en la Figura A.4, tras haber eliminado los nodos de representante $((0,1,0)^T,(0,1,1)^T)$ de los niveles 2 y 3, por haber quedado sin hermanos al eliminar el nodo de nivel 3 y representante ($(0,1,1)^T,(0,1,0)^T$).

- Si por el contrario el nodo terminal a eliminar tiene más de un nodo hermano, (caso de la Figura A.6), bastará con reajustar los enlaces del árbol para hacer desaparecer dicho nodo (llegando en nuestro caso a la situación de la Figura A.4).

A.3 Ejemplo

El siguiente ejemplo ilustra la selección de un conjunto de 8 representantes de una secuencia de datos donde las 500 primeros muestras están uniformemente repartidas en el subespacio ([0,1[x [0,1[) y las 500 últimas lo están en el ([1,2[x[1,2[)

La Figura A.7 muestra los 8 representantes seleccionados tras haber sido procesados las 500 primeras muestras. Los ocho representantes seleccionados en este momento son de nivel 3 y están situados en la cercandad del centro geométrrico de la rejilla que representan.

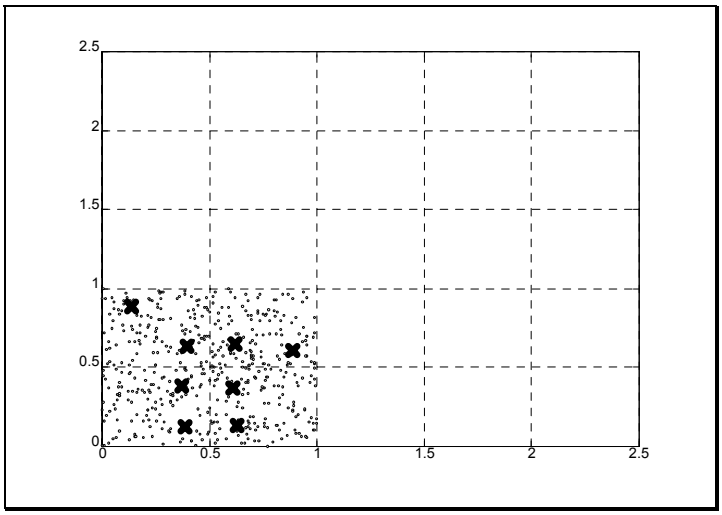

Figura A.7: Ejemplo de selección de 8 representantes tras haber sido procesadas 500 muestras

Al procesar las 500 restantes muestras, irán apareciendo representantes de niveles superiores que sustituirán nodos terminales ya creados de forma tal que se minimice la profundidad del árbol.

Al ir disminuyendo la profundidad del árbol, los representantes de nodos anteriormente no terminales pasarán a ser seleccionados de forma tal que los 8 representantes finalmente seleccionados cubran el espacio muestral de forma homogénea. Esta situación final es la representada en la Figura A.8:

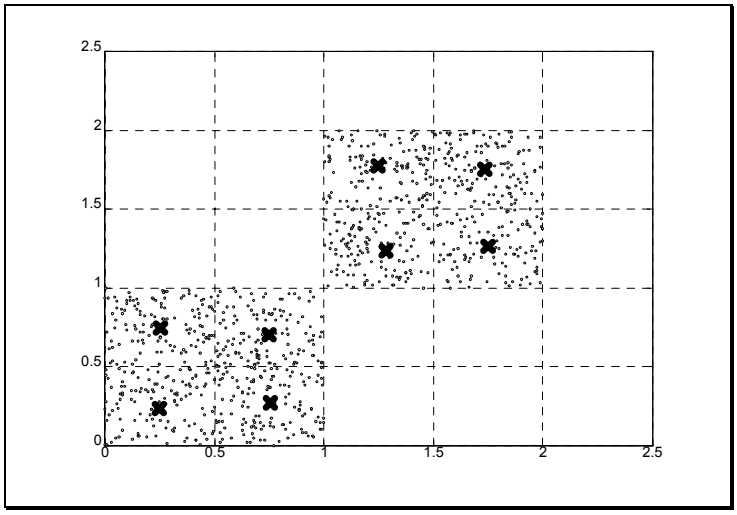

Figura A.8: Ejemplo de selección de 8 representantes tras haber sido procesadas 1000 muestras

Bibliografía

[Ackley et al., 1985] *"A learning algorithm for Boltzmann machine"*
D. Ackley, G. Hinton, T. Sejnowski
Cognitive Science, vol.9, pp.147-169, 1985

[Almeida, 1987] *"A learning rule for asynchronous perceptrons with feedback in a combinatorial environment"*
L.B. Almeida
IEEE 1st Int. Conf. on Neural Networks, vol.2, pp.609-618, San Diego, CA, 1987

[Almeida, 1988] *"Backpropagation in perceptrons with feedback"*
L.B. Almeida
Neural Computers (R. Eckmiller, C. von der Malsburg, eds) NATO ASI Ser., pp. 199-208, New York: Springer Verlag, 1988

[Ayoubi & Isermann, 94] *"Model-based fault detection and diagnosis with neural nets and application to a turbocharger"*
M. Ayoubi, R. Isermann
IFAC Artificial Intelligence in Real-Time Control, AIRTC'94, Valencia, Spain, 1994

[Barron, 1989] *"Statistical properties of artificial neural networks"*
A.R. Barron
Proc. of the 28th Conference on Decision and Control, pp.280-285, 1989

[Barron, 1991] *"Complexity regularization with application to artificial neural networks"*
A.R. Barron
Nonparametric Functional Estimation and Related Topics (G. Roussas, ed.), pp.561-576, 1991

[Barron, 1992] *"Neural net approximation"*
A.R. Barron
Proc. of the Seventh Yale Workshop on Adaptive and Learning Systems, pp.69-72. New Haven, CT: Yale University, 1992.

[Barschdorff, 1992] *"Comparison of Neural and Classical Decision Algorithms"*
D. Barschdorff
Fault Detection, Supervision and Safety for Technical Processes. IFAC Symposia Series, no.6, 1992

[Baum & Haussler, 1989] *"What size net gives valid generalization?"*
E.B. Baum, D. Haussler
Neural Computation 1, pp.151-160, 1989

[Bertsekas, 1979] *"Notes on Nonlinear Programming and Discrete-Time Optimal Control"*
Dimitri P. Bertsekas
MIT, CA, July 1979

[Bianchini & Frasconi, 1995] *"Learning without local minima in Radial Basis Function Networks"*
M.Bianchini, P.Frasconi
IEEE Trans. on Neural Networks, vol.6, n0.3, 1995

[Billings & Fung, 1995] *"Recurrent Radial Basis Function Networks for Adaptive Noise Cancellation"*
S.A.Billings, C.F.Fung
Neural Networks, vol.8, no.2, pp.273-290, 1995

[Blumer et al, 1989] *"Learnability and the Vapnik-Chervonenkis Dimension"*
A. Blumer, A. Ehrenfeucht, D. Haussler, M.K.Warmuth
Journal of the Association for Computing Machinery 36, pp.929-965, 1989

[Box & Jenkins, 1976] *"Time series analysis: forecasting and control"*
George E.P. Box, Gwilym M. Jenkins
Holden-Day Inc., CA, (Revised Edition) 1976

[Broomhead & Lowe, 1988] *"Multivariate functional interpolation and adaptive networks"*
D.S. Broomhead, D. Lowe
Complex Systems 2, pp.321-355, 1988

[Burattini & Tamburrini, 1991] *"A neural knowledge representation for diagnostic expert systems"*
E.Burattini, G.Tamburrini
Artificial Neural Networks,
T.Kohonen, K.Makisara, O.Simula, J.Kangas (Editors)
Elsevier Science Publisher B.V.(North Holland), 1991

[Burrascano, 1991] *"Learning vector quantization for the probabilistic neural network"*
P. Burrascano
IEEE Trans. on Neural Networks, vol.2, pp.458-461, July 1991

[Burrows & Niranjan, 1993] *"The use of feed-forward and recurrent neural networks for system identification"*
T.L.Burrows, M.Niranjan
CUED / F-INFENG / TR-158, December 1993

[Cacoullos, 66] *"Estimation of a multivariate density"*
T. Cacoullos
Ann. Inst. Statist. Math., vol.18, no.2, pp.179-189, 1966

[Carpenter & Grossberg, 1987] *"A massively parallel architecture for a self-organizing neural pattern recognition machine"*
G.A. Carpenter, S. Grossberg
Computer Vision, Graphics and image Processing 37, pp.54-115, 1987

[Chen & Billings, 1989] *"Recursive prediction error parameter estimator for non-linear models"*
S. Chen, S. Billings
Int. J. Control 49 (2), pp.569-594

[Chen et al., 1990] *"Nonlinear system identification using neural networks"*
S.Chen, S.Billings, P.Grant
Int. J. Control, vol.51, no.6, pp.1191-1214

[Chen et al., 1991] *"Orthogonal least squares learning algorithm for Radial Basis Function Networks"*
S.Chen, C.F.N.Cowan, P.M.Grant
IEEE Trans. on Neural Networks, vol.2, no.2, pp.302-309, 1991

[Chen et al., 1993] *"A clustering technique for digital comunications channel equalization using Radial Basis Function Networks"*
S.Chen, B.Mulgrew, P.M.Grant
IEEE Trans. on Neural Networks, vol.4, no.4, pp.570-579, 1993

[Chow & Willsky, 1984] *"Analytical redundancy and the design of robust failure detection systems"*
Edward Y.Chow, Alan S.Willsky
IEEE Trans. on Automatic Control, vol. AC-29, no.27, July 1984

[Chow et al., 1993] *"On the application and design of artificial neural networks for motor fault detection"*
M. Chow, R.N. Sharpe, J.C. Hung
IEEE Trans. on Industrial Electronics, vol.40, no.2, pp.181-196, April 1993

[Cuadra, 1990] *"El problema general de la optimización de diseño por ordenador: aplicación de técnicas de ingeniería de conocimiento"*
F. de Cuadra García
Tesis Doctoral. E.T.S. de Ingenieros Industriales. Universidad Pontificia Comillas. 1990.

[Cybenko, 1989] *"Approximation by Superpositions of a Sigmoidal Function"*
G. Cybenco
Mathematics of Control, Signals and Systems 2, pp.303-314
Springer Verlag, New York Inc., 1989

[Damitha et al., 1993] *"Power System Static Security Analysis using Radial Basis Function Neural Network"*
Damitha, K. Ranaweera, G.Karady
ESAP'93, pp.272-274

[Dasarathy, 1991] *"Nearest Neighbor (NN) norms: NN pattern classification techniques"*
B. V. Dasarathy
IEEE Computer Society Press, 1991

[Dekkers & Aarts, 1991] *"Global optimization and simulated annealing"*
Anton Dekkers, Emile Aarts
Mathematical Programming 50, pp.367-393, North Holland, 1991

[De Kleer & Brown, 1984] *"A qualitative physics based on confluences"*
J.De Kleer, J.S.Brown
Artificial Intellicence 24, 1984

[Demuth & Beale, 1992] *"Neural Network Toolbox User's Guide (For Use with MATLAB™)"*
H. Demuth, M. Beale
The Math Works, Inc. 1992

[Dietz, 1988] *"Pattern based fault diagnosis using neural networks"*
W.E.Dietz, E.L.Kiech, M. Al
Association for Computing Machinery 1988

[Duncan, 1974] *"Quality control and industrial statistics"*
A.J.Duncan
Richard D. Irwin, Inc., 4th edition, 1974

[Ebron et al., 1990] *"A neural network approach to the detection of incipient faults on power distribution feeders"*
S. Ebron, D.L. Lubkeman, M.White
IEEE Trans. on Power Delivery, vol.5, no.2, April 1990

[Faggin, 1991] *"VLSI implementation of neural networks"*
F. Faggin
International Joint Conference on Neural Networks. Seattle, WA, 1991

[Fahlman & Lebiere, 1990] *"The cascade correlation learning architecture"*
S.E. Fahlman, C. Lebiere
Advances in Neural Information Processing Systems 2 (D.S. Touretzky, ed.), pp.524-532. San Mateo, CA: Moran Kaufmann, 1990

[Forbus, 1984] *"Qualitative process theory"*
K.D.Forbus
Artificial Intelligence 24, 1984

[Frank, 1990] *"Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy - A survey and some new results"*
Paul M.Frank
Automatica, vol.26, no.3, pp.459-474, 1990

[Fukunaga, 1972] *"Introduction to the statistical pattern recognition"*
K. Fukunaga
Academic Press Inc. 1972

[Funahashi, 1989] *"On the approximate realization of continous mappings by neural networks"*
K. Funahashi
Neural Networks, vol. 2, pp.183-192, 1989

[Gertler, 1988] *"Survey of model-based failure detection and isolation in complex plants".*
J. Gertler
IEEE Control Systems Magazine, vol.8, no.6, 1988

[Gertler, 1991] *"Analytical redundancy methods in fault detection and isolation. Survey and Synthesis"*
J. Gertler
IFAC Fault Detection, Supervision and Safety for Technical Processes, Baden-Beden, Germany, 1991

[Gill et al., 1981] *"Practical Optimization"*
Philip E. Gill, Walter Murray, Margaret H. Wright
Academic Press Inc., 1981

[Girosi & Poggio, 1990] *"Networks and the best approximation theory"*
F.Girosi, T.Poggio
Biological Cybernetics 63, pp.169-176, 1990

[Girosi et al, 1995] *"Regularization Theory and Neural Networks Architectures"*
F. Girosi, M. Jones, T. Poggio
Neural Computation 7, pp.219-269, 1995

[Goldberg, 1989] *"Genetic Algorithms in Search, Optimization, and Machine Learning"*
D.E. Goldberg
Addison-Wesley, 1989

[Guo & Saul, 1991] *"Analysis of Gradient Descent Learning Algorithms for Multilayer Feedforward Neural Networks"*
H. Guo, S.B. Gelfand
IEEE Trans. on Circuits and Systems, vol.38, no.8, pp.883-894, August 1991

[Guyon et al., 1992] *"Estructural risk minimization for character recognition"*
I. Guyon, V. Vapnik, B. Boser, L. Bottou, S.A. Solla
Advances in Neural Information Processing Systems 4, pp.471-479. San Mateo, CA: Morgan Kaufmann, 1992.

[Hartigan, 1975] *"Clustering Algorithms"*
John A. Hartigan
John Wiley and Sons, 1975

[Hasselblad, 1966] *"Estimation of parameters for a mixture of Normal Distributions"*
V. Hasselblad
Technometrics, vol.8, no.3, pp.431-445, August 1966

[Haykin, 1994] *"Neural Networks, A Comprehensive Foundation"*
S. Haykin
Macmillan College Publishing Company, Inc.
IEEE Press, 1994

[Hecht-Nielsen, 1987] *"Kolomogorov's mapping neural network existence theorem"*
R. Hecht-Nielsen
1st IEEE Int. Conf. on Neural Networks, vol.3, pp.11-14, San Diego, CA, 1987.

[Hecht-Nielsen, 1989] *"Neurocomputing"*
R. Hecht-Nielsen
Addison-Wesley Publishing Company, 1989

[Hecht-Nielsen, 1990] *"On the algebraic structure of feedforward network spaces"*
R. Hecht-Nielsen
Advanced neural computers (R. Eckmiller, ed.), pp.129-135
North-Holland, Amsterdam, Netherlands, 1990

[Hernoth & Clark, 1995] *"Neural Networks that learn to predict probabilities: global models of nuclear stability and decay"*
K.A.Gernoth, J.W.Clark
Neural Networks, vol.8, no.2, pp.291-311, 1995

[Himmelblau et al., 1989] *"Incipient Fault Diagnosis of Chemical Processes via Artificial Neural Networks"*
D.M.Himmelblau, K.Watanabe, I.Matsuura, M. Abe, M.Kubota
AICh Journal, vol.35, no.11,Nov.1989

[Hinton et al., 1984] *"Boltzmann Machines: Constraint Satisfaction Networks That Learn"*
G. Hinton, T. Sejnowski, D. Ackley
Tech. Rep. CMU-CS-84-119. Carnegie Mellon University, 1984

[Hoff, 1962] *"Learning Phenomena in Networks of Adaptive Switching Circuits"*
M.E. Hoff Jr.
Ph.D. thesis, Tech. Report 1554-1, Stanford Electron. Labs
Stanford, CA, April 1962

[Hooke and Jeeves, 1961] *"Direct search: Solution for numerical and statistical problems"*
R. Hooke, T.A. Jeeves
Journal of the A.C.M, pp.212-229, 1961

[Hopfield, 1982] *"Neural Networks and phisical systems with emergent collective computational abilities"*
J.J. Hopfield
Proc. National Academy of Science, USA, 79, pp.2554-2558, 1982

[Hornik et al., 1989] *"Multilayer feedforward networks are universal approximators"*
K. Hornik, M. Stinchcombe, H. White
Neural Networks, vol.2, pp.359-366, 1989

[Hoskins & Himmelblau, 1988]"Artificial neural network models of knowledge representation in chemical engineering"
J.C. Hoskins, D.M. Himmelblau
Computer Chemical Engineering, vol.12, no.9/10, pp.881-890, 1988

[Huang, 1991] *"Bounds on the number of hidden neurons in Multilayer Perceptrons"*
Shih-Chi Huang, Yih-Fang Huang
IEEE Trans. on Neural Networks, vol.2, no.1, January 1991

[Hudson & Cohen, 1991] *"Combination of rule-based and connectionist expert systems"*
D.L. Hudson and M.E. Cohen
Microcomputer Applications, vol.10, no.2, 1991

[Hush & Horn, 1992] *"An overview of neural networks"*
R. Hush, B. Horne
Informática y Automática, vol.25 - 1 y 2/1992

[Isermann, 1984] *"Process fault detection based on modelling and estimation methods. A survey"*
R. Isermann
Automatica, vol.20,pp.387-404, 1984

[Jacobs, 1988] *"Increased rates of convergence through learning rate adaptation"*
R.A. Jacobs
Neural Networks, vol.1, pp.295-307, 1988

[Jang & Sun, 1993] *"Functional equivalence between Radial Basis Function Networks and fuzzy inference systems"*
J.S.R. Jang, C.T. Sun
IEEE Trans. on Neural Networks, vol.4, no.1, pp.156-158, 1993

[Janssen et al., 1988] *"Model structure selection for multivariate systems by cross-validation"*
P. Janssen, P. Stoica, T. Söderström, P. Eykhoff
International Journal of Control 47, pp.1737-1758, 1988

[Johnson & Wichern, 1982] *"Applied Multivariate Statistical Analysis"*
R.A. Johnson, D.W. Wichern
Prentice Hall, 1982

[Kardirkamanathan et al, 1991] *"Sequential adaptation of Radial Basis Function Neural Networks and its application to time series prediction"*
V. Kardirkamanathan, M.Niranjan, F.Fallside
In Advances in neural information processing systems 3 (Lippmann, Moody & Touretzky, eds.), pp.721-727, San Mateo: Morgan Kaufmann, 1991

[Kaufman & Rousseeuw, 1990] *"Finding groups in data: An introduction to cluster analysis"*
L.Kaufman, P.J.Rousseeuw
John Wiley & Sons, Inc. 1990

[Kohonen, 1988] *"Learning Vector Quantization"*
T. Kohonen
Abstracts of the 1st Annual INNS Meeting, Boston, MA, 303, 1988

[Kohonen, 1990] *"The self-organizing map"*
T. Kohonen
Proc. IEEE 78, pp.1464-1480, 1990

[Kosko, 1988] *"Bidirectional Associative Memories"*
B. Kosko
IEEE Trans. Systems, Man, Cybernetics, SMC-L8, pp.49-60, 1988

[Kuan et al, 1993] *"Recurrent back-propagation and Newton algotithms for training recurrent neural networks"*
Chung-Ming Kuan, Kurt Hornik, Tung Liu
SPIE'93.

[Kuipers, 1986] *"Qualitative simulation"*
B. Kuipers
Artificial Intelligence 29, 1986

[Kramer et al., 1989] *"Efficient parallel learning algorithms for neural networks"*
A.H. Kramer, A. Sangiovanni-Vincentelli
Advances in Neural Information Processing Systems 1 (D.S. Touretzky, ed.), pp.40-48, San Mateo, CA: Morgan Kaufmann, 1989

[Lang & Hinton, 1988] *"The development of the time-delay neural network architecture for speech recognition"*
K.J. Lang, G.E. Hinton
Tech. Rep. CMU-CS-88-152, Carnegie Mellon University, Pittsburgh, PA, 1988

[Lee et al. 1991] *"The effect of initial weights on premature saturation in backpropagation learning"*
Y. Lee, S. Oh, M. Kim
Int. Joint Conference on Neural Networks, vol.1, pp.765-770, Seattle, CA, 1991

[Leon-Garcia, 1994] *"Probability and Random Processes for Electrical Enginering"*
Alberto León-García
Addison Wesley. Second Edition. 1994

[Leonard et al., 1992] *"A neural network architecture that computes its own reliability"*
J.A.Leonard, M.A.Kramer, L.H.Ungar
Computers chem. Engng., vol.16, no.9, pp.819-835, 1992

[Leonard & Kramer, 1991] *"Radial Basis Function Networks for classifying process faults"*
J.A. Leonard, M.A. Kramer
IEEE Control Systems Mag. 4, pp.31-38, 1991

[Leonard & Kramer, 1993] *"Diagnosing dynamic faults using modular neural nets"*
J.A. Leonard, M.A. Kramer
IEEE Expert, pp.44-53, April 1993

[Lippmann, 1987] *"An Introduction to Computing with Neural Nets"*
R.P. Lippmann
IEEE ASSP Magazine, April 1987

[Liu & Nocedal, 1989] *"On the limited memory BFGS method for large scale optimization"*
Dong C. Liu, Jorge Nocedal
Mathematical Programming 45, pp.503-528, North Holland, 1989

[Ljung, 1987] *"System identification: Theory for the user"*
L. Ljung
Prentice Hall, Englewood Cliffs, NJ, 1987

[Ljung & Sjöberg, 1992] *"A system identification perspective on neural nets"*
L. Ljung, J. Sjöberg
1992 IEEE Workshop on Neural Networks for Signal Processing. IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854-4150. 1992

[Ljung & Wahlberg, 1992] *"Asymptotic properties of the least-squares method for estimating transfer functions and disturbances spectra"*
L. Ljung, B. Wahlberg
Adv. Appl. Prob. 24, pp.412-440, 1992

[Los Arcos et al., 1993] *"Diagnóstico basado en modelos de una planta de destilación de crudo"*
J.L. Los Arcos, L.M. Román, J.L. Malo, J. Pascual
Actas de la V Conferencia de la Asociación Española para la Inteligencia Artificial, CAEPIA'93, 1993

[Luenberger, 1984] *"Linear and Nonlinear Programming"*
David G. Luenberger
Addison-Wesley, 2ª Edición, 1984

[Luzzy & Dengel, 1993] *"A comparison of neural net simulators"*
O. Luzzy, A. Dengel
IEEE Expert, August 1993

[Madych & Nelson, 1990] *"Multivariate interpolation and conditionally positive definite functions"*
W.R.Madych, S.A.Nelson
Math.Comp.54 (189), pp.211-230, 1990

[Maren et al, 1990] *"Handbook of Neural Computing Applications"*
A. Maren, C. Harston, R. Pap
Academic Press Inc., 1990

[Maturana, 1988] *"Módulo de análisis estadístico de los datos de seguimiento de una máquina rectificadora"*
J.L. Maturana
Proyecto Fin de Carrera, E.T.S. de Ingenieros Industriales ICAI, Universidad Pontificia Comillas, 1988

[Milne, 1987] *"Strategies for diagnosis"*
IEEE Trans. on Systems, Man and Cybernetics, SMC-17,
pp.333-339, 1987

[Montgomery, 1985] *"Introduction to Quality Control"*
D.C. Montgomery
John Wiley and Sons, 1985

[Moody, 1992] *"The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems"*
J. Moody
Advances in Neural Information Processing Systems 4
Morgan Kaufmann Publishers, San Mateo, CA, 1992

[Moody & Darken, 1988] *"Learning with localized receptive fields"*
J. Moody, J.Darken
Proc. of the 1988 Connectionist Models Summer School, (G.Hinton, T. Sejnowski, and D.Touretzsky, eds.), pp.133-143, Palo Alto, CA, 1988

[Moody & Darken, 1989] *"Fast learning in networks of locally-tuned processing units"*
J. Moody, J.Darken
Neural Computation 1, pp.281-294, 1989

[Muñoz & Czernichow, 1995] *"Variable Selection through Statistical Sensibility Analysis: Application to Feedforward and Recurrent Neural Networks"*
Antonio Muñoz, Thomas Czernichow
Tech. Rep. 95-07-01 Institut National de Télécomunications
(INT-SIM). Paris 1995

[Muñoz et al., 1995-1] *"Neural network approach to the diagnosis of the boiler combustion in a coal power plant"*
Antonio Muñoz, José Villar, M.A Sanz-Bobi, Agustín Gimeno, Luis Zarauza
- Power-Gen Europe'95, Amsterdam 1995
- International Conference on Artificial Neural Networks
ICANN'95, Industrial Conference, Paris, 1995

[Musavi et al., 1992] *"On the training of Radial Basis Function Classifiers"*
M.T.Musavi, W.Ahmed, K.H.Chan, K.B.Faris, D.M.Hummels
Neural Networks, vol.5, pp.595-603, 1992

[Nguyen & Widrow, 1990] *"Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights"*
D. Nguyen, B. Widrow
International Joint Conference on Neural Networks
San Diego, CA, June 1990

[Nocedal, 1992] *"Theory of Algorithms for Unconstrained Optimization"*
Jorge Nocedal
Acta Numerica vol.1, pp.199-242, 1992

[Oppenheim & Willsky, 83] *"Signals and Systems"*
Alan V. Oppenheim, Alan S. Willsky and Ian T. Young
Prentice Hall, 1983

[Orr, 1995] *"Regularization in the selection of Radial Basis Function centers"*
M.J.L. Orr
Neural Computation 7, pp.606-623, 1995

[Pao, 1989] *"Adaptive Pattern Recognition and Neural Networks"*
Y.H. Pao
Addison Wesley, 1989

[Park & Sandberg, 1991] *"Universal approximation using Radial Basis Function Networks"*
J.Park, I.W.Sandberg
Neural Computation 3, pp.246-257, 1991

[Parker, 1985] *"Learning-logic: Casting the cortex of the human brain in silicon"*
D.B. Parker
Tech. Rep. TR-47, Center for Computational Research in Economincs and Management Science, MIT, Cambridge, MA, 1985

[Parker, 1987] *"Optimal algorithms for adaptive networks: Second order backpropagation, second order direct propagation and second order Hebbian learning"*
D.B. Parker
IEEE 1st Int. Conf. on Neural Networks, vol.2, pp.593-600, San Diegp, CA, 1987

[Parzen, 1962] *"On estimation of a probability density function and mode"*
E.Parzen
Ann. Math. Statist., vol.33, pp.1065-1076, 1962

[Patton & Chen, 1991] *"A review of parity space approaches to fault diagnosis"*
R.J. Patton, J.Chen
IFAC Fault Detection, Supervision and Safety for Technical Processes, Baden-Beden, Germany, 1991

[Pau, 1981] *"Failure diagnosis and performance monitoring"*
L.F. Pau
Marcel Dekker Inc., 1981

[Peña, 1986] *"Estadística. Modelos y métodos"*
Daniel Peña Sánchez de Rivera
Alianza Universidad Textos. 1986

[Pineda, 1987] *"Generalization of backpropagation to recurrent neural networks"*
F.J. Pineda
Physical Review Letters 59, pp.2229-2232, 1987

[Pineda, 1988] *"Generalization of backpropagation to recurrent and higher order neural networks"*
F.J. Pineda
Neural Information Processing Systems (D.Z. Anderson, ed.) pp. 602-611. New York: American Institute of Physics, 1988

[Poggio & Girosi, 1989] *"A theory of networks for approximation and learning"*
T. Poggio, F. Girosi
MIT AI Memo, no.1140, MIT, Cambridge, 1989

[Poggio & Girosi, 1990] *"Networks for Approximation and Learning"*
T. Poggio, F. Girosi
Proc. IEEE, vol.78, no.9, pp.1481-1497, Sept. 1990

[Polycarpou & Vemuri, 1995]"Learning methodology for failure detection and accomodation"
M.M. Polycarpou, A.T. Vemuri
IEEE Control Systems, June 1995

[Powell, 1985] *"Radial basis functions for multivariate interpolation: A review"*
M.J.D. Powell
IMA Conf. on Algorithms for the approximation of functions and Data, pp.143-167, RMCS, Shrivenham, UK, 1985

[Ramírez Vázquez, 1980] *"Centrales Eléctricas: Enciclopedia CEAC de electricidad"*
J. Ramírez Vázquez
Ediciones CEAC, S.A. 4ª Edición 1980

[Ramón y Cajal, 1911] *"Histología del sistema nervioso del hombre y de los vertebrados"*
Santiago Ramón y Cajal
Consejo Superior de Investigaciones Científicas Madrid, 1911

[Ranaweera, 1994] *"comparison of neural network models for fault diagnosis of power systems"*
D.K.Ranaweera
Electric Power Systems Research, 29, pp.99-104, 1994

[Renals & Rohwer, 1989] *"Phoneme classification experiments using Radial Basis Functions"*
S.Renals, R.Rohwer
Proc. on Int. Joint Conf. on Neural Networks, vol.I, pp.462-467, Washington DC, 1989

[Renders et al., 1995] *"A prototype neural network to perform early warning in nuclear power plant"*
J.M. Renders, A. Goosen, F. de Viron, M. De Vlaminck
Fuzzy Sets and Systems 74, pp.139-151, 1995

[Richardson, 1985] *"Artificial Intelligence in Maintenance"*
J.J. Richardson
Noyes Publications, 1985

[Rosenblatt, 1962] *"Principles of Neurodynamics"*
F. Rosenblatt
Washington, DC: Spartan Books. 1962

[Rumelhart et al., 1986a] *"Learning representations by back-propagating errors"*
D.E. Rumelhart, G.E. Hinton, R.J. Willliams
Nature (London), 323, pp.533-536, 1986

[Rumelhart et al., 1986b] *"Learning internal representations by error propagation"*
D.E. Rumelhart, G.E. Hinton, R.J. Willliams
Parallel Distributing Processing: Explorations in the Microstructure of Cognition (D.E. Rumelhart, J.L. McClelland, eds), vol.1, chapter 8, Cambridge, MA: MIT Press, 1986

[Rush et al., 1992] *"Error Surfaces for Multilayer Perceptrons"*
D.R. Hush, B. Horne, J.M. Salas
IEEE Trans. on Systems, Man ans Cybernetics, vol.22, no.5, pp. 1152-1161, September/October 1992

[Sanner & Slotine, 1992] *"Gaussian Networks for Direct Adaptive Control"*
R.M.Sanner, J.J.E.Slotine
IEEE Trans. on Neural Networks, vol.3, no.6, 1992

[Sanz Bobi, 1992] *"Metodología de mantenimiento predictivo basada en análisis espectral y temporal de la historia de los equipos industriales y enfoque de su aplicación a un sistema experto"*
M.A. Sanz Bobi
Tesis Doctoral, Universidad Politécnica de Madrid, E.T.S. de Ingenieros Industriales, 1992

[Sanz Bobi et al., 1993] *"TRAFES: Expert system for diagnosis of power transformers"*
M.A.Sanz Bobi, A.García Cerrada, R.Palacios, J.Villar, et al.
CIGRE, Berlin, April 1993

[Sanz Bobi et al., 1994-1] *"SEDFIPA: Diagnosis Expert System of the Preheating Circuit in a Coal Power Plant"*
M.A. Sanz Bobi, José Villar, Antonio Muñoz, Agustín Gimeno
Second Worls Congress on Expert Systems, Portugal, 1994

[Sanz Bobi et al., 1994-2] *"Control and Diagnosis of water chemistry in the water-steam cycle and water makeup in a fossil fueled power plant"*
M.A. Sanz Bobi, I.J. Pérez Arriaga, J.L. Serrano Carbayo, M.E. Ortiz Alfaro, J.J. Alba, A. Domenech, M.J. Villamediana
Electrical Power & Energy Systems, vol.16, no.4, 1994

[Shanno, 1990] *"Recent Advances in Numerical Techniques for Large-Scale Optimization"*
David F. Shanno
Neural Networks for Control (W. Thomas Muller III, R.S. Sutton and Paul Werbos, eds.), MIT Press, 1990

[Sima, 1995] *"Neural Expert Systems"*
Jiri Sima
Neural Networks, vol.8, no.2, pp.261-271, 1995

[Sjöberg, 1995] *"Non-Linear System Identification with Neural Networks"*
Jonas Sjöberg
Ph. D. Thesis, Linköping University, Sweden, 1995

[Sorsa et al., 1991] *"Neural Networks in Process Fault Diagnosis"*
T. Sorsa, H.N. Koivo, H. Koivisto
IEEE Trans. on Systems, Man. and Cybernetics, vol.21, no.4,
Jul/Aug 1991

[Specht, 1990] *"Probabilistic neural networks"*
D.F.Specht
Neural Networks, vol.3, pp.109-118, 1990

[Specht, 1991] *"A General Regression Neural Network"*
D.F. Specht
IEEE Trans. on Neural Networks, vol.2, no.6, Nov. 1991

[Stone, 1974] *"Cross-validatory choice and assesment of statistical predictions"*
M. Stone
Journal of the Royal Statistical Society, B36, pp.111-133, 1974

[Szu, 1987a] *"Fast Simulated Annealing"*
H. Szu
Neural Networks for Computing, AIP Conf., vol.15, pp.420-425, J.Denker (ed.), Snow Bird, UT, 1987

[Szu, 1987b] *"Nonconvex Optimization"*
H. Szu
SPIE, vol.968, pp.59-65, 1987

[Tank & Hopfield, 1987] *"Neural computation by concentrating information in time"*
D.W. Tank, J.J. Hopfield
Proc. of the National Academy of Sciences of the USA 84, pp.1896-1900, 1987

[Traven, 1991] *"A neural network approach to statistical pattern classification by semiparametric estimation of probability density functions"*
H.G.C. Traven
IEEE Trans. on Neural Networks, vol.2, no.3, pp.366-377, May 1991

[Tzafestas, 1989] *"System fault diagnosis using the knowledge-based methodology"*
S.G.Tzafestas
Fault Diagnosis in Dynamic Systems (Ed. R.Patton, P.Frank and R.Clark), Prentice Hall, 1989

[Vapnik, 1982] *"Estimation of Dependencies Based on Empirical Data"*
New York: Springer Verlag, 1982.

[Vapnik, 1992] *"Principles of risk minimization for learning theory"*
V.N. Vapnik
Advances in Neural Information Processing Systems 4 p.831-838. San Mateo, CA:Morgan Kaufmann, 1992

[Vapnik & Chervonenkis, 1971] *"On the uniform convergence of relative frequencies of events to their probabilities"*
V.N. Vapnik, A.Y. Chervonenskis
Theoretical Probability and Its Applications 17, pp264-280, 1971

[Velasco, 1991] *"Arquitectura para sistemas de control inteligentes"*
J.R.Velasco Pérez
Tesis Doctoral. 1991
Universidad Politécnica de Madrid
E.T.S. de Ingenieros de Telecomunicaciones

[Wahba, 1990] *"Splines models for observationnal data"*
G.Wahba
Series in Applied Mathematics, vol.59, SIAM, Philadelphia, 1990

[Wang, 1992] *"Generating Fuzzy Rules by Learning from Examples"*
L.X.Wang
IEEE Trans. on Systems, Man. and Cybernetics, vol.22, no.6, 1992

[Watanabe et al., 1989] *"Incipient fault diagnosis of chemical processes via artificial neural networks"*
K. Watanabe, I. Matsuura, M. Abe, M. Kubota, D.M.Himmelblau
AIChE Journal, vol.35, no.11, pp.1803-1812, 1989

[Webb, 1994] *"Functional approximation by feedforwaed networks: A least-squares approach to generalization"*
A.R.Webb
IEEE Trans. on Neural Networks, vol.5, no.3, 1994

[Wei, 1990] *"Time series analysis: univariate and multivariate methods"*
William W. S. Wei.
Addison-Wesley, 1990

[Werbos, 1974] *"Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Science"*
P. Werbos
Ph. D. dissertation, Harvard University, Cambridge, MA, 1974

[Werbos, 1988] *"Generalization of backpropagation with application to a recurrent gas model"*
P. Werbos
Neural Networks, vol.1, pp.339-356, 1988

[Werbos, 1989] *"Backpropagation and neurocontrol: A review and prospectus"*
P. Werbos
Proc. Int. Joint Conf. on Neural Networks, Washington DC, June 1989

[Widrow & Hoff, 1960] *"Adaptive switching circuits"*
B. Widrow, M.E. Hoff
Neurocomputing. pp.126-134.
J.Anderson and E. Rosenfield (Eds.)
MIT Press, Cambridge, MA.

[Widrow, 1962] *"Generalization and information storage in networks of adaline 'neurons'"*
B. Widrow
In Self-Organizing Systems (M.C. Yovitz, G.T. Jacobi and G.D. Goldstein, eds.) pp. 435-461. Washington, DC:Sparta.1962

[Widrow et al., 1987] *"Learning phenomena in layered neural networks"*
B. Widrow, R.G. Winter, R. Baxter
Proc. 1st IEEE Int. Conf. on Neural Networks, vol.2, pp411-429, San Diego, CA, June 1987.

[Widrow & Winter, 1988] *"Neural Nets for adaptive filtering and adaptive pattern recognition"*
B. Widrow, R.G.Winter
IEEE Computer, pp.25-39, Mar. 1988

[Widrow & Lehr, 1990] *"30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation"*
B. Widrow, M.A. Lehr
Proc. IEEE, vol.78, no.9, pp. 1415-1442, Sept. 1990

[Wienholt, 1993] *"Optimizing the structure of Radial Basis Function Networks by optimizing fuzzy inference systems with evolution strategy"*
W. Wienholt
Internal Report 93-07. Ruhr-Universität Bochum, Institut für Neuroinformatik, 1993

[Wilson et al., 1991] *"Power Plant Operation"*
B.J.Wilson, B.J.Vincent, G.R.L.Wright
Modern Power Station Practice, vol.G: Station Operation and Maintenance, Chapter 2, British Electricity International, London Pergamon Press, 1991

[Winter, 1989] *"MADALINE Rule II: A new method for training networks of Adalines"*
R.G. Winter
Ph.D. Thesis, Stanford University.Stanford, CA, Jan. 1989

[Xu et al., 1994] *"On Radial Basis Function Nets and kernel regression: Statistical consistency, convergence rates, and receptive field size"*
L. Xu, A. Krzyzak, A. Yuille
Neural Networks, vol.7, no.4, pp.609-629, 1994

[Yager & Zadeh, 1992] *"An introduction to fuzzy logic applications in intelligent systems"*
Editado por L.A.Zadeh y L.A.Zadeh
Kluwer Academic Publishers, 1992

[Yang et al., 1995] *"On-line fault diagnosis of power substation using connectionist expert system"*
H.T. Yang, W.Y.Chang, C.L.Huang
IEEE Trans. on Power Systems, vol.10, no.1, 1995

[Zadeh, 1965] *"Fuzzy sets"*
L.A.Zadeh
Information and Control 8, pp.338-353, 1965

[Zadeh, 1983] *"The role of fuzzy logic in the management of uncertainty in expert systems"*
L.A.Zadeh
Fuzzy Sets and Systems 11, pp.199-227, 1983