MENU SALIR

UNIVERSIDAD DE MÁLAGA
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

TESIS DOCTORAL

REDES NEURONALES ARTIFICIALES PARA LA GESTIÓN DE SISTEMAS DE ENERGÍA ELÉCTRICA

AUTOR: Francisco García Lagos
Licenciado en Informática
2003

MENU SALIR

D. GONZALO JOYA CAPARRÓS, TITULAR DE UNIVERSIDAD DEL DEPARTAMENTO DE TECNOLOGÍA ELECTRÓNICA DE LA UNIVERSIDAD DE MÁLAGA

CERTIFICO:

Que D. Francisco García Lagos, Licenciado en Informática, ha realizado en el Departamento de Tecnología Electrónica de una Universidad de Málaga, bajo mi dirección el trabajo de investigación correspondiente a su Tesis Doctoral titulada:

"REDES NEURONALES ARTIFICIALES PARA LA GESTIÓN DE SISTEMAS DE ENERGÍA ELÉCTRICA".

Revisado el presente trabajo, estimo que puede ser presentado al Tribunal que ha de juzgarlo.

Y para que conste a efectos de lo establecido en el Real Decreto 778/98 regulador de los estudios de Tercer Ciclo-Doctorado, AUTORIZO la presentación de esta Tesis en la Universidad de Málaga

Málaga, a 29 de mayo de 2003

Fdo. Gonzalo Joya Caparrós
Titular de Universidad

MENU SALIR

Departamento de Tecnología Electrónica
Universidad de Málaga

TESIS DOCTORAL

REDES NEURONALES ARTIFICIALES PARA LA GESTIÓN DE SISTEMAS DE ENERGÍA ELÉCTRICA

AUTOR: Francisco García Lagos
Licenciado en Informática
DIRECTOR: Gonzalo Joya Caparrós
Licenciado en Ciencias Físicas
Dr. en Informática

MENU SALIR

MENU SALIR

UNIVERSIDAD DE MÁLAGA
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

Reunido el tribunal compuesto por:

Presidente: Dr. D.
Secretario: Dr. D.
Vocales:
Dr. D.
Dr. D.
Dr. D.

para juzgar la tesis doctoral titulada "REDES NEURONALES ARTIFICIALES PARA LA GESTIÓN DE SISTEMAS DE ENERGÍA ELÉCTRICA", presentada por D. Francisco García Lagos y dirigida por Dr. D. Gonzalo Joya Caparrós,

acordó por _____ otorgar la calificación de

_____.

Málaga, _____ de _____ de 2003

El Presidente El Secretario

Fdo. Fdo.

Vocal 1 Vocal 2 Vocal 3

Fdo. Fdo. Fdo.

MENU SALIR

A mi madre.

MENU SALIR

Agradecimientos

Muchos han sido los que han colaborado en este trabajo. Gracias a todos ellos. En particular, quiero manifestar mi gratitud a:

Gonzalo Joya Caparrós, mi director de tesis, por su tiempo, dedicación y apoyo en la realización de este trabajo, pero sobre todo, por transmitirme su filosofía de trabajo.

A Javier Marín, con quién comencé mis trabajos de investigación cuando aún era estudiante, por su apoyo y ayuda a lo largo de todos estos años.

A Francisco Sandoval, por iniciar en el Departamento de Tecnología Electrónica la línea de investigación en la que se encuadra este trabajo, así como por la confianza depositada en mí para formar parte del grupo de investigación.

A la empresa ELIOP S.A., por la financiación de proyectos de investigación y desarrollo que han contribuido a este trabajo, por su colaboración y confianza, así como por su ayuda en su ejecución.

A Red Eléctrica de España y, en especial a José Luis Mata y Guillermo Juberías, por su colaboración técnica imprescindible a lo largo del desarrollo de varios proyectos de investigación, así como por sus comentarios e ideas en la prueba de varias de las aplicaciones que han sido desarrolladas.

Y en último lugar, reconocer a todos aquellos compañeros que me han prestado ayuda cuando la he necesitado.

A todos y cada uno, mi agradecimiento.

MENU SALIR

Resumen

Esta tesis tiene dos objetivos generales principales: por un lado, estudiar las Redes Neuronales Artificiales (RNAs), tanto en sus aspectos teóricos como de aplicación a problemas reales. Por otro, la propuesta de soluciones realistas, basadas en las RNAs, a diversos problemas relacionados con la gestión de un sistema de energía eléctrica.

Ante un problema particular, el primer objetivo general se ha concretado en los siguientes objetivos específicos: a) establecer las características que ese problema debe presentar para que sea recomendable su resolución mediante RNAs; b) establecer el paradigma neuronal más adecuado para su resolución; c) una vez elegido un paradigma, determinar su estructura topológica adecuada; d) conocer las peculiaridades de cada una de las funciones de activación que pueden ser incorporadas a una neurona, así como su efecto en el funcionamiento general del sistema, y e) conocer los distintos algoritmos de entrenamiento o de evolución dinámica de los diferentes paradigmas y su incidencia en el tiempo de convergencia y en la precisión de la solución alcanzada.

En cuanto al segundo objetivo general, se han analizado y propuesto diversas soluciones neuronales a las siguientes operaciones implicadas en un sistema de gestión de energía eléctrica: predicción de la demanda eléctrica, análisis de contingencias y estimación de estado, la cual incluye las subtareas de estimación topológica y observabilidad. Todos estos problemas reales presentan características que hacen recomendable la utilización de técnicas neuronales para su solución. Para cada uno de ellos se realiza un estudio exhaustivo de sus condiciones, se proponen y justifican diversos paradigmas neuronales o estructuras mixtas y se obtienen y analizan los resultados. En este último aspecto, las aportaciones principales pueden resumirse como sigue.

Predicción de Demanda de Energía Eléctrica

Se propone una arquitectura neuronal compleja constituida por la secuenciación de dos paradigmas diferentes. En una primera etapa, se utiliza un paradigma de aprendizaje no supervisado (Mapas Auto-Organizativos de Kohonen) para la clasificación de patrones. En una segunda etapa, un paradigma de topología realimentada y aprendizaje supervisado (Red de Elman) realiza la predicción de la demanda de manera independiente para los días de cada una de las clases obtenidas. Las ventajas de este modelo de predicción frente a otros existentes son las siguientes: 1) el usuario dispone de un perfil de 24 horas en cada momento que se actualiza hora a hora. Esto le permite llevar un control exhaustivo para participar en las ofertas de generación en cada uno de los mercados diarios e intradiarios del sistema actual; 2) el módulo implementado de entrenamiento periódico (on-line) activado a petición del usuario, hace que la influencia de la no-estacionariedad de la curva de la demanda sea menor que en el caso de realizar la predicción con paquetes estadísticos, menos flexibles y con un mayor tiempo de actualización en la búsqueda de la estimación de los parámetros de predicción; 3) la adaptabilidad a otras zonas de regulación es altamente facilitada por el módulo de clasificación no supervisada, lo que reduce la intervención de un experto respecto a los paquetes estadísticos existentes.

Estimación de Estado.

El problema de la estimación de estado ha sido planteado como un proceso de

optimización utilizando redes recurrentes de Hopfield, convirtiendo la expresión de la suma ponderada de residuos al cuadrado utilizada en los algoritmos clásicos, en una función de energía de Hopfield. Las aportaciones principales de este trabajo son las siguientes: a) se han simulado diferentes métodos de activación de neuronas y su influencia en la solución final obtenida; b) se propone e implementa un método de adaptación automática del parámetro de simulación de la ecuación diferencial del modelo independiente del problema considerado y c) se realiza una implementación de la red utilizando técnicas de almacenamiento y manipulación de matrices dispersas que optimiza los recursos necesarios para la simulación de la red.

Estimación de Topología

Para esta tarea se propone un sistema neuronal compuesto por dos fases: fase de preprocesamiento y fase de clasificación. La fase de preprocesamiento convierte las medidas disponibles en vectores que serán fácilmente clasificables por la segunda de potencial gaussiana basada en la distancia de Mahalonobis. En esta capa hay una unidad por cada posible topología del bus, de manera que para cada patrón producido por la etapa de preprocesado, la unidad con valor de activación alto indicará la topología asociada con el vector de medidas original. Las ventajas de este nuevo módulo de estimación topológica son las siguientes: los parámetros implicados en el modelo son obtenidos de manera simple y directa a partir de un conjunto histórico de medidas, siendo innecesario un proceso de aprendizaje; el proceso de estimación es local y la topología de cada bus se obtiene de manera independiente y paralela; para buses con todas sus medidas conocidas u obtenidas mediante pseudomedidas, la topología correcta es obtenida en todos los casos, incluso cuando se cuenta con medidas altamente erróneas; la estimación puede realizarse incluso cuando no se dispone de una medida de flujo o de inyección; el sistema es altamente inmune a grandes cambio en la curva de carga de potencia; la complejidad del sistema crece de manera lineal con el tamaño de la red eléctrica, ya que cada nuevo bus sólo necesita un nuevo módulo de preprocesado y clasificación.

Análisis de seguridad

Los Mapas Autoorganizativos de Kohonen han sido aplicados a la monitorización visual de la evolución del sistema con respecto al nivel de peligrosidad de cada contingencia particular, y las redes alimentadas hacia delante como el Perceptrn Multicapa o las redes de funciones de base radial han sido aplicadas a la ordenación de contingencias por medio de la evaluación numérica de algún índice de prestación. En este trabajo se muestra la alta aplicabilidad de las redes de Kohonen a este problema, debido a que facilita una rápida visualización de la severidad de una contingencia y a que posibilita la predicción de la tendencia del sistema hacia un punto de operación posiblemente peligroso respecto a una contingencia particular. Respecto a la ordenación de contingencias por su nivel de peligrosidad, se han estudiado y comparado las soluciones que aportan dos paradigmas supervisados clásicos como son el perceptron multicapa y las redes de función de base radial.

Abstract

This work has two main general objectives: in one hand, the study of Artificial Neural Networks (ANNs) in its theoretical aspects, and its application to the resolution of real problems. In the other hand, the proposal of neural network based realistic solutions to problems related with energy power management systems.

For each particular problem, the first general objective has been split in other specific ones: a) to establish the characteristics that the problem should present to be solved by means of RNAs; b) to establish the more appropriate neuronal paradigm for its resolution; c) once a paradigm has been selected, to determine their structure and appropriate topology; d) to know the peculiarities of each activation and their effects in the general operation of the system; and e) to know the different algorithms and dynamics for different paradigms, and their effect in the solution obtained.

For the second general objective, we analyse and propose several complex neural network solutions for the following energy management system tasks: power demand forecasting, contingency analysis, and state estimation, including topology estimation. All these real problems present characteristic that make advisable the application of neural network techniques for their solution. For each one of these problems, it is carried out an exhaustive study of their conditions, and the neural paradigm applied and the results obtained are justified. In this sense, the main contributions of this work can be summarized as follow:

Power Demand Forecasting

The proposed solution for this task is composed by a complex neural network architecture constituted by two different paradigms. In a first stage, a Self-Organizing Map of Kohonen is used for patterns classification. In a second stage, a supervised feedback paradigm (Elman neural network) carries out the forecasting in an independent way for each cluster obtained in the first stage. The advantages of this model are the following ones: 1) the user has a profile of 24 hours in each moment that hour is upgraded. This allows to take exhaustive control to participate in the generation offers in each one of the daily markets of the current system; 2) the implemented periodic retraining process speed up the adaptation of the model to new consumption trend; 3) robustness and adaptability of our system to other regulation areas is based on the capability of Kohonen maps to extract non evident environmental, cultural and economic factors, and the special ability of recurrent ANNs to model time series.

State Estimation

The problem of the state estimate has been outlined as a optimization process using Hopfield's recurrent network, converting the expression from the weighted squared sum of residuals used in the classic algorithms, in an energy function of the Hopfield network. The main contributions of this work are the following ones: to) different methods of neuron activation functions and their influence have been simulated in the obtained final solution; b) it implements a method for automatic the parameter for equation discretization, and c) the simulation of the neural network is implemented by mean of sparse techniques, that reduces significantly the amount of required memory and convergence time to the solution.

Topology Estimation

For this task we propose a system architecture comprises two stages: the first one, the Preprocessing stage, transforms each current measurement set available from SCADA to produce a vector in the $[0,1]^n$ space. This stage produces clusters of very similar preprocessing output vectors for each grid topology. The second stage, the Classification stage, consists in a layer of Gaussian potential units with Mahalanobis distance, and classifies the preprocessing output vectors to identify the actual topology. This model presents the following advantages: a) the estimation process is local, and the topology of each bus is obtained by an independent and parallel module; b) the system complexity grows linearly with the size of the power network; c) the parameters involved in the model are obtained through a straightforward method from a previously generated measurement database and training process is not needed; d) for buses with all their power flow measurements known or obtained as pseudomeasurements, the correct topology is obtained in all cases, even for very bad measurements; e) the topology assessment can be used when a flow or injection measurement is not available; in this situation the correct topology is obtained even with very bad measurements; f) the system performance is highly immune to large changes in the power load curve.

Contingency Analysis

The Contingency Analysis is a classification problem, oriented to provide an estimation of the security level of the current system state regarding the occurrence of a possible contingency. This task may be approached by both functional methods, characterizing the severity of each contingency by a numerical value, and graphical methods, permitting the contingency evaluation in a visual way. In this work, both approaches have been studied. Supervised feed-forward neural networks have been applied to the functional contingency analysis. These paradigms generate the numerical function that evaluates the security level of each contingency for each current state, permitting the construction of a contingency ranking. In particular, we analyse and compare the performance of Multilayer Perceptron (MLP) with Backpropagation learning algorithm and Radial Basis Function (RBF) Networks. For the second task, bi-dimensional and linear Seft-Organaizing Maps (SOM) of Kohonen have been applied to the visual classification and monitoring of the evolution of the system concerning of the security level of the current system state regarding the occurrence of a possible contingency. The main advantages of SOM are the following ones: a) for the linear SOM, each contingency is represented by a segment in the SOM, and this is divided in successive subsegments representing an increasing severity scale. Thus, a very high number of contingencies may be presented simultaneously in only one screen, and the analysis of the complete contingency set may be done in a very reduced time by an human operator; b) the representation by a SOM of the successive power system operation points permits to follow the system tendency with respect to each particular contingency, and to predict its approaching to a dangerous zone with enough advance; c) each contingency produces a different distribution of the severity intervals in the original SOM, but the training process is unique for all the contingencies. Thus, the complexity of the off-line training process does not significantly increase with the power system size. Of course, the process in the operation model is carried out in real time.

Acrónimos

AR	Modelo autoregresivo
ARX	Modelo autorregresivo con variables de entrada exógenas
ARMA	Modelo autorregresivo de media móvil
ARMAX	Modelo autorregresivo de media móvil de entradas exógenas
ARIMA	Modelo autorregresivo integrado de media móvil
ARIMAX	Modelo autorregresivo integrado de media móvil de entradas exógenas
$C_i(t)$	Función que determina la clase a la que pertenece el patrón t para la predicción del consumo de energía eléctrica.
EMS	Sistema Gestión de Energía Eléctrica (Energy Management System)
h()	Función no lineal que relación el vector de estado con las medidas disponibles en el proceso de estimación de estado de un sistema de potencia.
H()	Matriz jacobiano.
J()	Vector de residuos al cuadrado ponderados. Función a minimizar en la estimación de estado.
LVQ	Learning Vector Quantization
MA	Modelo de media móvil
MAPE	Error relativo medio (Mean Absolute Percentage Error)
MLP	Perceptrón Multicapa (del inglés Multilayer Perception)
PI	Ïndice de prestación (performance index).
R	Matriz de covarianzas de errores en las medidas en la estimación de estado.
RBF	Red de funciones base radiales (Radial Basic Function network)
RNA	Red Neuronal Artificial
SCADA	Sistema de supervisión, control y adquisición de datos (Supervisory Control And Data Acquisition)
SOM	Mapas auto-organizativos (Seft-Organaizing Map)
v	Vector de errores de las medidas en la estimación de estado.
W	Matriz inversa de la matriz de covarianzas de errores en las medidas R.
WLS	(Estimador WLS) Estimador de mínimos cuadrados ponderados (Weighted Least Squares).
x	Vector de estado para la estimación de estado.
Y_H	Función que determina el consumo eléctrico de la hora H.
z	Vector de medidas disponibles para la estimación de estado.

MENU SALIR

MENU SALIR

Índice general

1. Introducción **1**

1.1. Introducción 1

1.2. Estructura de la tesis 5

2. Predicción de la demanda de energía eléctrica **9**

2.1. Introducción 9

2.2. Predicción de series temporales 10

2.2.1. Predicción de la demanda de energía eléctrica 12

2.3. Redes neuronales para procesamiento de series temporales 16

2.3.1. Redes neuronales alimentadas hacia delante 17

2.3.2. Aplicación de las redes alimentadas hacia delante a la predicción de la demanda de energía eléctrica 20

2.3.3. Redes neuronales recurrentes 22

2.3.4. Aplicación de las redes recurrentes a la predicción de la demanda . . . 26

2.4. Problemas en la predicción de la demanda eléctrica con RNAs 28

2.5. Un nuevo modelo de predicción de carga eléctrica 29

2.5.1. Preprocesamiento de los datos 30

2.5.2. Clasificación mediante mapas auto-organizados de Kohonen 31

2.5.3. Aproximación de la función de consumo mediante redes de Elman . . . 35

2.5.4. Resultados de funcionamiento y discusión 41

2.6. Conclusiones 45

3. Estimación de estado **49**

3.1. Introducción 49

3.2. La estimación de estado como problema matemático 50

3.3. Estimación de estado con redes de Hopfield 52

3.3.1. Adaptación de los pesos de la red de Hopfield para la estimación de estado 55

3.3.2. Implementación del modelo neuronal de Hopfield 58

3.3.3. Utilización de matrices dispersas con redes de Hopfield 62

3.3.4. Resultados de la estimación de estado utilizando el modelo neuronal de Hopfield 65

3.4. Conclusiones 68

4. Estimación topológica **71**

4.1. Introducción 71

4.2. Métodos de estimación topológica 73

4.3. Estimación topológica con Perceptrones multicapa 80

4.3.1. Estimación de topología local 81

4.3.2. Estimación de topología global 89

4.3.3. Proceso de Estimación Topológica 90

4.3.4. Resultados de la estimación de topología 91

4.3.5. Conclusiones sobre los resultados del estimador topológico basado en MLP 100

4.4. Asistente topológico basado en funciones base radiales con distancia de Mahalanobis 101

4.4.1. Etapa de preproceso 101

4.4.2. Etapa de clasificación 102

4.4.3. Resultados experimentales 105

4.4.4. Conclusiones sobre los resultados del asistente topológico basado en unidades de funciones base radiales con distancia de Mahalanobis . . . 109

5. Análisis de seguridad 113

5.1. Introducción . . . 113

5.2. Principales métodos de análisis de contingencias . . . 116

5.3. Generación de patrones e índices de prestación . . . 125

5.3.1. Selección de características . . . 126

5.4. Evaluación numérica y ordenación de contingencias con redes neuronales alimentadas hacia delante y aprendizaje supervisado . . . 129

5.4.1. Evaluación numérica de contingencias con Perceptrón Multicapa y *Backpropagation* . . . 130

5.4.2. Evaluación numérica de contingencias con redes de funciones base radiales . . . 132

5.5. Visualización de contingencias con mapas Auto-Organizativos de Kohonen . . 136

5.5.1. SOM bidimensional para la visualización de contingencias . . . 138

5.5.2. SOM lineal para la visualización de contingencias . . . 142

5.6. Conclusiones . . . 147

6. Conclusiones y líneas futuras 151

6.1. Conclusiones . . . 151

6.2. Líneas futuras . . . 155

A. Descripción de los paradigmas neuronales empleados 157

A.1. Perceptrón Multicapa (MLP) . . . 157

A.1.1. Simulación o ejecución de la red neuronal . . . 158

A.1.2. Entrenamiento de un MLP . . . 159

A.2. Redes de funciones base radiales (RBF) . . . 164

A.2.1. Entrenamiento de RBF . . . 165

A.3. Red recurrente de Elman 166
A.4. Redes de Kohonen 168
A.4.1. Entrenamiento de un SOM 170
A.5. Red de Hopfield 171
A.5.1. Modelo discreto 171
A.5.2. Modelo continuo 174

B. Descripción básica de un sistema de gestión de energía eléctrica 177

B.1. Introducción 177
B.2. Predicción de carga eléctrica 180
B.3. Estimación de estado 181
B.3.1. Análisis de observabilidad 182
B.3.2. Estimación de topología 182
B.3.3. Detección de medidas erróneas 182
B.4. Análisis de seguridad 182
B.5. Planificación operacional del sistema de energía eléctrica 183

C. Diagramas de las redes IEEE estándares utilizadas en esta tesis 185

Índice de tablas

2.1. Detalle de la clasificación de los datos de consumo eléctrico obtenidos por la red de Kohonen 35

2.2. Número de épocas de entrenamiento para cada red de Elman. 40

2.3. Errores medios de predicción para cada clase para el módulo de predicción diaria. 43

3.1. Ejemplos de tiempos de cómputo (en segundos) para la estimación de estado utilizando una implementación no dispersa y utilizando la versión propuesta en este trabajo. 65

3.2. Residuos finales y tiempo requerido por los estimadores de estado neuronal y clásico. 67

4.1. Resultados de la estimación topológica local para la red neuronal del bus 2. . 95

4.2. Resultados representativos de la interacción del estimador de estado y del estimador de topología. *E.* significa Erróneo y *C.* significa Correcto. 97

5.1. Porcentaje total de explicación de la variación del conjunto de patrones originales de la red IEEE-14, tomando diferentes componentes principales 129

5.2. Errores relativos medios en tanto por ciento (MAPE) para los índices de prestación de las contingencias problemáticas de la red estándar IEEE-14. . . 132

5.3. Ordenación de las contingencias según el valor de su índice de prestación. Las columnas con título OR representan la ordenación real de la contingencia, OO la ordenación obtenida por las redes MLP, OR-OO la diferencia (en número de posiciones) entre la ordenación real y obtenida, PIR representa el índice de prestación real, PIO el índice de prestación obtenido. La última columna (PIR - PIO) muestra la diferencia entre los índices de prestación real y obtenido para cada contingencia. Las contingencias mostradas corresponden a diferentes estados de la red eléctrica. 133

5.4. Errores relativos medios en tanto por ciento (MAPE) para los índices de prestación de las contingencias problemáticas de la red estándar IEEE-14. . . 136

5.5. Ordenación de las contingencias según el valor de su índice de prestación. Las columnas con título OR representan la ordenación real de la contingencia, OO la ordenación obtenida por las redes RBF, OR-OO la diferencia (en número de posiciones) entre la ordenación real y obtenida, PIR representa el índice de prestación real, PIO el índice de prestación obtenido y la última columna la diferencia entre estos dos últimos índices. 137

5.6. Identificación de las neuronas activadas y el valor correspondiente PI correspondientes a la figura 5.8. 143

Índice de figuras

1.1. Esquema de las operaciones implicadas en la gestión de un sistema eléctrico estudiadas en esta tesis y la solución neuronal correspondiente. 7

2.1. Diagrama de un modelo de predicción general. 14

2.2. Diagrama de las redes realimentadas tipo Jordan (a), Elman (b) y Residual (c). 23

2.3. Red neuronal con la salida realimentada para implementar un modelo ARMA no lineal. 24

2.4. Arquitectura del sistema de predicción de carga basado en RNAs 30

2.5. Valor del radio de vecindad durante el entrenamiento de la red de Kohonen. El eje x representa el porcentaje de iteraciones. 33

2.6. Distribución final de los perfiles de consumo diario usados en el entrenamiento de la red de Kohonen para la Zona Centro de España. S.C. indica superclase. 34

2.7. Esquema del funcionamiento on-line del predictor de carga. 41

2.8. MAPE para todos los patrones de la clase 14. 44

2.9. Resultados de predicción y curva real para los días laborables de la segunda semana de julio de 1997 (clase 14). 44

2.10. Resultados de predicción y curva real para el periodo de Semana Santa de 1998. 45

3.1. Representación de una red de Hopfield. 54

3.2. Diagrama de flujo con las tareas principales implicadas en la estimación de estado con una red de Hopfield. 57

3.3. Evolución de la energía de una red de Hopfield aplicada a problemas de estimación de estado para diferentes métodos de activación de sus neuronas: activación secuencial (S), activación aleatoria restringida (RR) y activación aleatoria (R). 59

3.4. Valores óptimos obtenidos para el parámetro α en la estimación de estado. . . 61

3.5. Representación de la dispersidad de la matriz de pesos para la red de Hopfield aplicada al problema de la estimación de estado. Los elementos diferentes de cero se representan con un rectángulo no blanco. 63

3.6. Estructuras de datos utilizadas en la implementación de la red de Hopfield aplicada a problemas de estimación de estado. 64

3.7. Evolución del residuo en cada iteración de los algoritmos de estimación neuronal y clásico. (a) Red IEEE-14. (b) Red IEEE-30. (c) Red IEEE-57. (d) Red IEEE-118. 66

4.1. Estructura general del estimador topológico. 80

4.2. Representación de un bus de una red, con un generador, una carga y dos líneas. 82

4.3. Función de transferencia de base radial para diferentes valores de σ. 84

4.4. Representación esquemática de la etapa de preprocesamiento de base radial para un patrón con 3 medidas de potencia activa y 3 de reactiva. 84

4.5. Representación de las salidas de la etapa de preprocesamiento para patrones típicos de dos topologías diferentes. 86

4.6. Arquitectura del estimador topológico neuronal local. 87

4.7. Márgenes asignados a las salidas de la red neuronal para determinar el estado abierto, cerrado o de indecisión. 87

4.8. Evolución de la suma del error cuadrático y de la tasa de aprendizaje durante el entrenamiento de la red neuronal. 89

4.9. Representación esquemática del bus 2 y sus vecinos en la red IEEE-14. 96

ÍNDICE DE FIGURAS

4.10. Representación de la etapa de preproceso y la salida del asistente topológico basado en funciones potenciales de Gauss. (a) Etapa de preproceso. (b) salida de la etapa de preproceso. 103

4.11. Esquema de la etapa de clasificación del asistente topológico basado en funciones potenciales de Gauss. (a) Representación de las neuronas de base radial con distancia de Mahalanobis, que constituyen la etapa de clasificación del asistente topológico. (b) Salida. 104

4.12. Patrón para la variación de la carga diaria de los buses. 106

4.13. Salidas de la etapa de clasificación para patrones pertenecientes a diferentes topologías, cuando todas las medidas del bus están disponibles. 108

4.14. Salidas de la etapa de clasificación para patrones pertenecientes a diferentes topologías, cuando una de las medidas del bus no está disponible. 110

5.1. Coeficientes aplicados al patrón base de cada red para obtener patrones con diferentes cargas del sistema 126

5.2. Representación esquemática de la selección de características mediante Análisis por Componentes Principales para RNAs.. 127

5.3. Representación gráfica de los valores real y calculado del índice de prestación $PIp, 1$ para el conjunto de los patrones de prueba. 131

5.4. Representación gráfica de los valores real y calculado del índice de prestación $PIp, 1$ para el conjunto de los patrones de prueba. 135

5.5. Representación esquemática de la distribución de un espacio n-dimensional de entrada en un espacio bidimensional mediante un Mapa Auto-Organizativo de Kohonen. 138

5.6. Representación de la clasificación de los patrones obtenida por la red de Kohonen para la contingencia 1. Los símbolos de la figura se han seleccionado en función del índice de prestación de cada patrón. Cada subgráfico o rectángulo representa una neurona del mapa. El rectángulo de la esquina superior izquierda es la neurona 1, el rectángulo de la esquina superior derecha es la neurona 10 y el rectángulo de la esquina inferior derecha corresponde a la neurona 100. Los huecos de la figura representan neuronas a la que no pertenece ningún patrón. El mapa de Kohonen representado es de dimensión 10x10. . . 140

5.7. Representación de la clasificación de los patrones obtenida por la red de Kohonen para la contingencia 13. 141

5.8. Mapa de Kohonen para la contingencia 10 y las neuronas activadas para patrones con diferentes PI no utilizados durante el entrenamiento. 143

5.9. Representación de los índices de prestación de la contingencia 10 para varios patrones. 144

5.10. Visualización de varios Mapas de Kohonen lineales en una misma figura. Se han representado 10 Mapas de Kohonen lineales de 1 x 30 neuronas cada uno. Cada fila de la figura representa el Mapa de Kohonen visualizado para cada contingencia particular. Las contingencias visualizadas son la 3, 4, 7, 8, 9, 11, 14, 17, 26 y 33 de la red IEEE-118. 145

5.11. Representación del nivel de peligrosidad que ha producido un determinado patrón presentado a la red de Kohonen Lineal. La neurona ganadora es la neurona 7, pero su nivel de peligrosidad es distinto para cada contingencia, como muestra el correspondiente atributo de la neurona 7 en cada fila de la figura. 145

5.12. PI, para la contingencia 3, de 26 patrones no usados durante el entrenamiento del SOM. 146

5.13. Respuesta del SOM lineal de la contingencia 3 a cada uno de los patrones de la figura 5.12. 146

ÍNDICE DE FIGURAS

A.1. Red neuronal alimentada hacia delante tipo MLP, con dos capas de neuronas ocultas. . . . 158

A.2. Dirección del flujo de información durante la ejecución (o *recall*) de un MLP. 159

A.3. Dirección del flujo de información durante la fase propagación hacia atrás del error. . . . 161

A.4. Estructura de una red de funciones base radiales. . . . 164

A.5. Representación de una red de Elman. . . . 167

A.6. Disposición bidimensional de las neuronas de una red de Kohonen. . . . 169

A.7. Representación esquemática de la distribución de un espacio n-dimensional de entrada en un espacio bidimensional mediante un Mapa Auto-Organizativo de Kohonen. . . . 169

A.8. Vecindad topológica cuadrada para una red de Kohonen bidimensional. Las neuronas vecinas de n para varios valores de la función A son todas las incluidas en el cuadrado correspondiente.. . . . 171

A.9. Arquitectura de una red de Hopfield. . . . 172

B.1. Componentes principales de un análisis de seguridad *on-line* ([Balu et al., 1992]). . . . 179

C.1. Diagrama de la red IEEE 14. . . . 186

C.2. Diagrama de la red IEEE 30. . . . 187

C.3. Diagrama de la red IEEE 57. . . . 188

C.4. Diagrama de la red IEEE 118. . . . 189

Capítulo 1

Introducción

1.1. Introducción

Las Redes Neuronales Artificiales (RNAs) constituyen una herramienta de computación fácilmente aplicable -incluso como caja negra mediante programas de ordenador específicos- a problemas generales como predicción de series temporales, clasificación, optimización y aproximación funcional. Esto permite su aplicación de manera unificada a una amplia gama de problemas reales en diversos campos de la ciencia y la ingeniería. Así, problemas como la predicción de la demanda de energía eléctrica, la predicción del consumo de una materia prima o la predicción del comportamiento de un determinado índice económico pueden ser modelados como un problema de predicción de series temporales y resueltos por un mismo paradigma neuronal de aprendizaje supervisado como el Perceptrón Multicapa o las Redes de Funciones Base Radiales; el análisis de contingencias o de fallos en un sistema o la división de una determinada población según determinados factores socioeconómicos puede ser entendido como un problema de clasificación y resuelto mediante paradigmas neuronales no supervisados como los mapas de Kohonen; la identificación de sistemas, esto es, la estimación de los parámetros que intervienen en un modelo basado en Ecuaciones Diferenciales Ordinarias, puede ser descrito como un problema de optimización y resuelto mediante redes neuronales de Hopfield, etc. [Joya et al., 2002b].

Por otro lado, estas técnicas están basadas en paradigmas que presentan un comportamiento interno altamente complejo y muy difícil de describir de manera sistemática.

Estas dos características, facilidad para ser implementadas y dificultad para comprender

en profundidad cómo trabajan, han favorecido, por una parte, su rápida extensión en prácticamente todos los campos de la ciencia y la técnica pero, por otra parte, un alto porcentaje de uso espurio y de resultados engañosos.

Este uso espurio que con frecuencia podemos encontrar publicado en congresos y revistas, incluso avalado por grupos de investigación con una larga historia en el campo, contribuye sin duda a minar la reputación de las Redes Neuronales Artificiales, de las que suele decirse de manera irónica que "son la segunda mejor herramienta para resolver cualquier problema". Consideramos, por tanto, que se hace necesario, por parte de quienes usan habitualmente estas herramientas, un esfuerzo por conocer con la mayor profundidad posible las características teóricas de cada paradigma neuronal, sus restricciones y su campo de aplicación.

En este contexto se enmarca el primer objetivo general de esta tesis: *el estudio de las Redes Neuronales Artificiales, tanto en sus aspectos teóricos como en sus aspectos de aplicación a la resolución de problemas reales*

Este objetivo general puede descomponerse en otros específicos entre los que destacaríamos los siguientes:

- Establecer las características que debe presentar un determinado problema para que sea recomendable su resolución mediante RNAs. En este sentido, podría proponerse como regla básica la siguiente: *Si un problema puede resolverse utilizando métodos analíticos, numéricos o al menos algorítmicos satisfaciendo las condiciones establecidas en cuanto a tiempo de cálculo y carga computacional, entonces, no intentar resolverlo con métodos neuronales.*

- Para un problema dado, establecer el paradigma neuronal más adecuado para su resolución. Cada paradigma neuronal tiene características que lo hacen adecuado para problemas de un tipo particular. Así, para problemas de clasificación, los paradigmas más utilizados son el Perceptrón Multicapa (MLP), los Mapas Auto-Organizativos (SOM) o el *Learning Vector Quantization* (LVQ). Por otro lado, para problemas de aproximación funcional, la red de funciones base radiales (RBF) es comúnmente elegido como mejor candidato, debido a que permite dividir el espacio de entrada en subregiones, cada una de las cuales es aproximada por una o más neuronas. En otros muchos

casos, el problema sólo podrá ser resuelto mediante la construcción de otras estructuras neuronales más complejas.

- Una vez elegido un paradigma, determinar su estructura topológica adecuada. Esto supone disponer de criterios para seleccionar las variables que deben intervenir como entradas a la red, su número de neuronas ocultas y la forma en que se producirá la información de salida.

- Conocer las peculiaridades de cada una de las funciones de activación que pueden ser incorporadas a una neurona, así como su efecto en el funcionamiento general del sistema.

- Conocer los distintos algoritmos de entrenamiento o de evolución dinámica de los diferentes paradigmas y su incidencia en el tiempo de convergencia y en la precisión de la solución alcanzada.

Pero, para avanzar en la consecución de estos objetivos, se hace necesario disponer de un conjunto de problemas que sirvan de entorno de utilización de los distintos paradigmas estudiados. Estos problemas deben presentar unas características que no sólo hagan posible sino justificable, recomendable y útil su ataque mediante Redes Neuronales Artificiales. En este trabajo, se han elegido como casos de estudio los problemas reales relacionados con la Gestión de un Sistema de Energía Eléctrica (EMS, *Energy Management System* en su versión inglesa), en particular, los relacionados con la predicción de la demanda eléctrica, el análisis de contingencias y la estimación de estado, la cual incluye las subtareas de estimación topológica y observabilidad. Entendemos que estos problemas se adaptan de manera óptima a los intereses de este trabajo por tres razones fundamentalmente:

1.- Cada uno de ellos puede ser descrito de manera genérica como alguna de las tareas recomendadas para el uso de redes neuronales. Así, la predicción de la demanda eléctrica puede definirse como un problema de predicción de series temporales y nos permitirá analizar la aplicabilidad y limitaciones de los paradigmas basados en el aprendizaje*Backpropagation* (Perceptrón Multicapa y Redes de Elman); el análisis de contingencias puede plantearse como un problema de aproximación funcional o de monitorización gráfica, lo que permitirá analizar el comportamiento de redes alimentadas hacia delante con aprendizaje supervisado

en el primer caso (Perceptrón Multicapa y Redes de Funciones Base Radiales) y el de redes de tipo Auto-Organizativo en el segundo (Mapas Auto-Organizativos de Kohonen); finalmente, la estimación de estado puede ser enfocada como un problema de optimización, por lo que será utilizada para estudiar el margen de aplicabilidad de redes cuya dinámica sigue la evolución de una función de energía (redes recurrentes de Hopfield).

2.- Estas tareas presentan muchas de las características que dificultan una solución basada en métodos numéricos o algorítmicos, y que por tanto, hacen más justificable el uso de redes neuronales. Estas características son: a) existencia de un gran número de datos tales como medidas de fasores de tensión y corriente, flujos de potencia activa y reactiva medidos en ambos extremos de la línea, estados de interruptores, posición de tomas de transformadores, etc.; b) las relaciones entre las distintas variables son excesivamente complejas y, en muchos casos, desconocidas; c) estos datos son difícilmente interpretables y manejables en tiempo real por un operador humano, y la adquisición de una experiencia completa por parte de éste requiere un tiempo muy largo; d) los métodos numéricos que pueden ser aplicados impiden en un gran número de casos encontrar soluciones en tiempo real; e) la solución no puede ser descrita por medio de un conjunto simple de reglas basado en el conocimiento experto.

3.- Los problemas elegidos, no son sólo ejercicios académicos, sino que siguen siendo problemas abiertos en el sentido de que las soluciones actualmente en funcionamiento son manifiestamente mejorables.

Como resultado de la elección de estos casos de estudio, podemos hablar de un segundo objetivo general para nuestra tesis, el cual contiene la parte práctica y de aplicación de la misma: *la resolución de diversos problemas relacionados con la gestión de un sistema de energía eléctrica mediante técnicas neuronales artificiales*

En definitiva, y como resumen, podemos decir que el objeto de esta tesis es el estudio de aplicabilidad y limitaciones de los paradigmas más representativos de las Redes Neuronales Artificiales y su aplicación a tareas relacionadas con la Gestión de un Sistema de Energía Eléctrica.

1.2. Estructura de la tesis

El resto de este trabajo se distribuye del siguiente modo. El capítulo 2 está dedicado a la predicción de la demanda de energía eléctrica mediante redes neuronales artificiales. Tras un repaso bibliográfico del estado del arte, se presenta un modelo neuronal compuesto de dos etapas. En la primera fase se realiza una clasificación no supervisada mediante redes Auto-Organizativas de tipo SOM, que permite dividir la serie temporal original en subseries atendiendo a criterios de estacionalidad y laboralidad. La segunda etapa consiste en la predicción de la función de consumo. Esta etapa se ha implementado mediante redes neuronales recurrentes de Elman. Este modelo neuronal contiene las ventajas de los modelos recurrentes frente a los alimentados hacia delante, mientras que permite el aprovechamiento de la gran variedad de algoritmos de entrenamiento desarrollados para el Perceptrón Multicapa.

El capítulo 3 describe la aplicación de las RNAs al problema de la estimación de estado. Se estudia el paradigma de Hopfield y se proponen modificaciones en la evolución temporal de la red y de otros parámetros para su aplicación a este problema particular.

El capítulo 4 se dedica al problema de la estimación de la topología de una red eléctrica a partir de medidas disponibles para la estimación de estado. Se proponen dos modelos neuronales complejos para abordar este problema, siendo el segundo una mejora sustancial del primero. En primer lugar, se propone un modelo neuronal compuesto por dos etapas. Una etapa de preproceso, cuya función es la de convertir las medidas asociadas a un bus en patrones característicos de la topología real del bus. En la segunda etapa, se realiza una clasificación de los patrones preprocesados mediante un MLP. En segundo lugar, se propone un asistente topológico que reemplaza al MLP por una red neuronal de funciones base radiales con distancia de Mahalanobis, más eficiente que el MLP y que no necesita de un entrenamiento supervisado previo.

El capítulo 5 se dedica a la aplicación de las RNAs a problemas de seguridad en un sistema de Energía Eléctrica, concretamente al análisis de contingencias. El problema se aborda tanto desde el punto de vista de la ordenación de contingencias según su gravedad, como desde su monitorización. En el primer caso, se utilizan tanto MLP como RBF para encontrar los índices de prestación asociados a las posibles contingencias, y se realiza un estudio de las soluciones aportadas por los dos modelos. En el segundo, se utilizan redes

tipo SOM para la visualización y monitorización del estado actual del sistema frente a contingencias. Se presentan dos modelos diferentes, el primero un SOM bidimensional y el segundo un SOM lineal que mejora en gran medida la aplicabilidad de este paradigma al problema de la visualización y monitorización de contingencias.

El capítulo 6 presenta las conclusiones y líneas futuras.

La figura 1.1 muestra, a modo de resumen, las diferentes operaciones implicadas en un EMS que han sido estudiadas en este trabajo, y la correspondiente solución neuronal que se ha aportado.

Puesto que esta tesis unifica dos campos de trabajo bastante distantes en principio: las redes neuronales artificiales y los sistemas de gestión de energía eléctrica, se han añadido dos apéndices (apéndices A y B) describiendo las características básicas de cada uno de ellos. El apéndice C presenta los diagramas de las redes estándares IEEE utilizadas como banco de pruebas a lo largo de este trabajo.

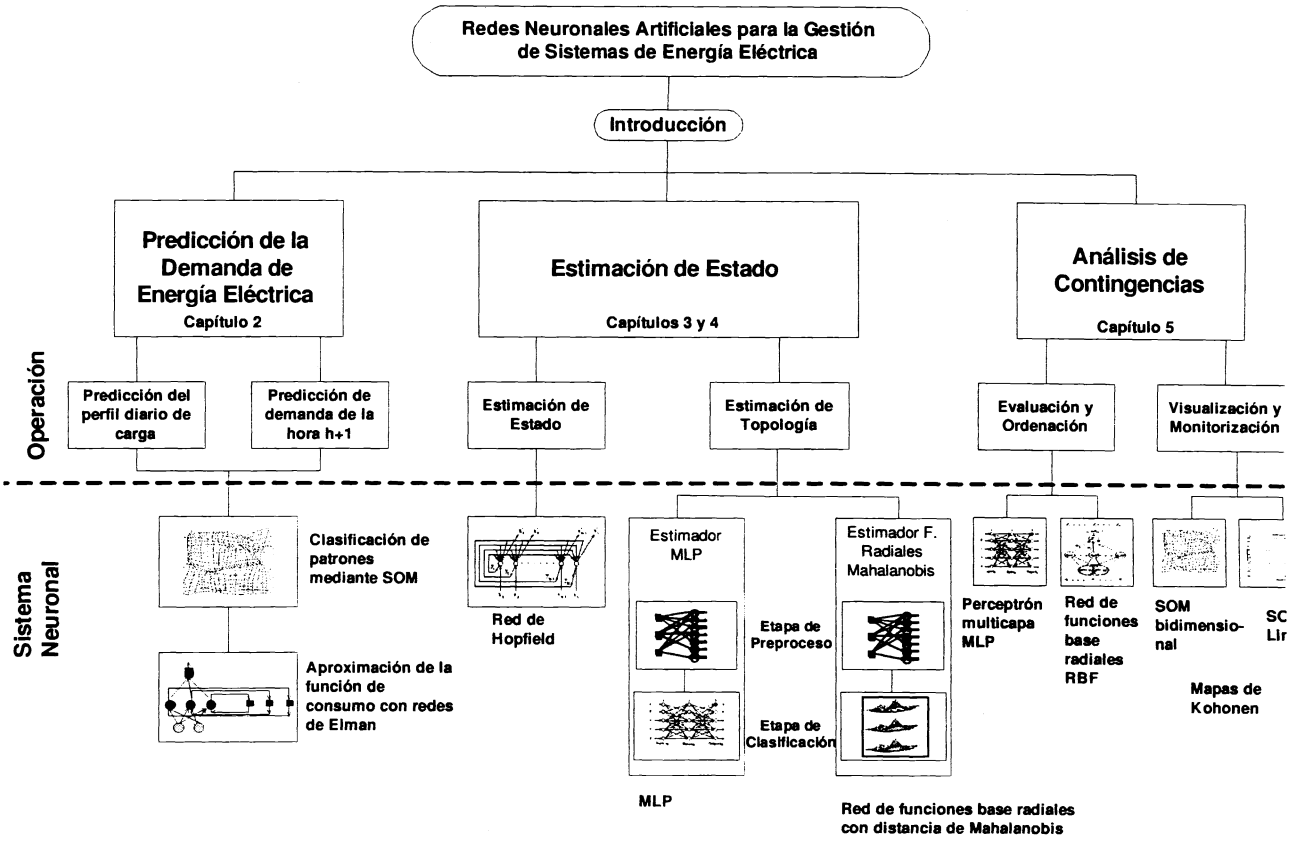


Figura 1.1: Esquema de las operaciones implicadas en la gestión de un sistema eléctrico estudiadas en esta tesis y la solución neuronal correspondiente.

Capítulo 2

Predicción de la demanda de energía eléctrica

2.1. Introducción

El objetivo de la predicción de la demanda de energía eléctrica es obtener los valores futuros del consumo eléctrico en función de los valores del consumo pasado. Los valores predichos de demanda se utilizan para la planificación del sistema eléctrico de potencia, principalmente para la elección de las plantas de generación que producirán la energía requerida. Debido a que existen diferentes tipos de plantas de generación, cada una con restricciones de funcionamiento y con costes de producción diferentes, es esencial conocer con suficiente antelación y precisión el consumo de energía futuro con diferentes horizontes. Por otro lado, se debe garantizar el suministro seguro y, dado que la energía eléctrica no se puede almacenar eficientemente, es necesario que en todo momento se genere exactamente la energía necesaria para mantener el sistema en un estado seguro. Sin embargo, la tarea se presenta como un problema complejo debido a que la demanda exhibe grandes variaciones diarias, semanales y estacionales, que dependen de muchas variables y que son difíciles o imposibles de modelar con exactitud. Por estas razones, la investigación en campos relacionados con la predicción de la demanda de energía eléctrica ha tenido en los últimos años un auge muy importante, sobre todo debido a las condiciones de deregularización a la que se están sometiendo muchos países, entre ellos el nuestro. Esta deregularización del mercado ha condicionado el periodo de predicción para ajustarlo a los periodos de operación del mercado y exige disponer de predicciones con una exactitud cada vez mayor. Esto está impulsando la utilización de

nuevas técnicas de computación inteligente como son las redes neuronales artificiales.

En este capítulo se presenta una revisión de las propuestas más significativas que abordan el problema de la predicción de carga utilizando redes neuronales artificiales, y se propone un modelo neuronal de predicción del perfil diario de demanda eléctrica basado en la secuenciación de dos paradigmas: en un primer nivel se realiza una clasificación de la serie temporal mediante mapas Auto-Organizados de Kohonen [Kohonen, 1990], que ha permitido dividir la serie temporal en subseries más regulares, que son mejor procesadas por las redes neuronales. Además, esta clasificación nos ha permitido eliminar la necesidad de utilizar como entradas al modelo variables exógenas cuyo efecto en la serie no es fácilmente ponderable y cuya codificación adecuada para ser utilizada en un modelo neuronal se presenta difícil. En un segundo nivel, el modelo de predicción propuesto utiliza redes recurrentes de Elman [Elman, 1990] para encontrar los valores futuros de cada subserie disponible. Como complemento al predictor del perfil diario de carga se propone un predictor de demanda con un horizonte de una hora (predictor horario), que se adapta mejor a cambios en la curva de consumo dentro de las 24 horas, debidos a condiciones imprevistas como factores meteorológicos.

El capítulo se estructura de la siguiente forma. En primer lugar, se formaliza el problema de la predicción de series temporales en general y el de la demanda de energía eléctrica en particular (sección 2.2). En la sección 2.3 se presentan los modelos neuronales más utilizados para abordar este problema y se realiza un repaso bibliográfico de los modelos más significativos. En la sección 2.4 se analizan los problemas que se encuentran en la aplicación de RNAs a los problemas de predicción, y que intentamos resolver con nuestro modelo. En la sección 2.5 se presenta en detalle el modelo de predicción propuesto en este trabajo y se analizan los resultados de funcionamiento. Por último, la sección 2.6 presenta las conclusiones del capítulo.

2.2. Predicción de series temporales

Una serie temporal $T = \{\vec{x}_1, \vec{x}_2, ..., \vec{x}_{n-1}, \vec{x}_n\}$ es un conjunto de datos discretos capturados o medidos desde algún proceso o fenómeno. Cada variable $\vec{x}$ representa un vector, que si tiene un único componente se denomina serie univariante, mientras que si tiene más

de un componente se denomina multivariante. Se dice que la serie temporal es de orden$p$ si es posible encontrar un valor futuro de la misma utilizando únicamente p valores pasados.

Formalmente, el problema de la predicción de series temporales se puede formular como el proceso de encontrar una función $F : \Re^{kn+l} \rightarrow \Re^k$ que permita obtener una estimación $\hat{\vec{x}}(t+d)$ del vector $\vec{x}$ en el instante $t+d$, a partir de valores pasados de $\vec{x}$ y de variables independientes, denominadas variables exógenas, u_i:

$$\hat{\vec{x}}(t+d) = F(\vec{x}(t), \vec{x}(t-1), ...\vec{x}(t-n-1), u_1, ..., u_l) \tag{2.2.1}$$

donde l es el número de variables exógenas, d se denomina horizonte de predicción y n es el número de valores pasados utilizados en la predicción. El vector $\vec{x}$ tiene dimensión k.

Con el planteamiento anterior, el problema de predicción de series temporales se convierte en un problema de **aproximación funcional**, que puede ser resuelto mediante una amplia gama de procedimientos como regresión lineal o no lineal, redes neuronales, etc. La evaluación de la calidad de la predicción se mide en términos del valor de una función de error sobre cada uno de los patrones de la serie temporal. Esta función de error tiene la forma general de la ecuación (2.2.2):

$$E = \sum_{i=0}^{N} e(\hat{\vec{x}}(t-i), \vec{x}(t-i)) \tag{2.2.2}$$

donde e es una función que evalúa el error entre la estimación del vector $\vec{x}$ ($\hat{\vec{x}}$) y el valor real de dicho vector.

En algunas aplicaciones no es necesario obtener el valor exacto de $\vec{x}(t+d)$, sino una indicación de si $\vec{x}(t+d)$ es mayor, menor o igual que $\vec{x}(t)$. En este caso, el problema se convierte en uno de **clasificación de series temporales** que puede definirse formalmente como encontrar una función $F_c : \Re^{kn+l} \rightarrow \beta^k$ que asigna una de varias clases a un patrón particular de la serie, donde β es un vector de componentes binarios, como expresa la ecuación (2.2.3):

$$F_c : (\vec{x}(t), \vec{x}(t-1), ..., u_1, ..., u_l) \rightarrow \hat{c}_i \in \zeta \tag{2.2.3}$$

donde ζ es el conjunto de clases. Para este tipo de aplicaciones se suele utilizar como medida de error la ecuación (2.2.4):

$$E = (1 - \frac{1}{N} \sum_{i=1}^{N} \delta_{\hat{c}_i, c_i}) * 100 \tag{2.2.4}$$

donde δ_{ij} es la función de Kronecker y c_i es la clase del patrón de entrada i. Esta expresión representa el porcentaje de entradas que el sistema ha clasificado correctamente.

Existe otro tipo de aplicaciones que plantean el problema del **modelado de la serie temporal**. En estos casos, la función F de la ecuación (2.2.1) se considera un modelo de otra u otras series temporales. A partir del modelado de la serie es posible explicar o generar los valores de otra serie temporal (p.e. encontrar los valores de la serie 'precio del petróleo' conociendo los valores de la serie 'tipos de interés').

En la ecuación (2.2.1) se ha supuesto que puede obtenerse un modelo exacto de la serie temporal. Sin embargo, en aplicaciones reales como la predicción de la demanda de energía eléctrica, esta suposición no es realista. Existen factores tales como errores en las mediciones, factores externos no considerados o no controlables como los hábitos de los usuarios, etc, que nos obligan a asumir que incluso el modelo más exacto va a generar un error residualϵ que no puede ser filtrado. Por tanto, es necesario reformular la ecuación (2.2.1) de la siguiente forma:

$$\hat{\vec{x}}(t+d) = F(\vec{x}(t), \vec{x}(t-1), ..., u_1, ..., u_l) + \hat{\epsilon}(t) \tag{2.2.5}$$

El error $\hat{\epsilon}(t)$ no puede incluirse explícitamente en el modelo, pero sí puede modelarse estadísticamente, de forma que no sólo es posible dar una predicción de la serie temporal, sino también dar una medida del error de dicha estimación [Dorffner, 1996].

2.2.1. Predicción de la demanda de energía eléctrica

La predicción de la demanda de energía eléctrica es un tipo particular de predicción de series temporales que se define formalmente de la siguiente forma: dados los vectores de entrada u^t y de salida y^t observados desde un sistema dinámico discreto, tal que $u^t = [u(1), u(2), ..., u(t)]$ e $y^t = [y(1), y(2), ..., y(t)]$, encontrar los valores $y(t)$ futuros a partir de los pares de observaciones de entrada/salida pasados $[u^{t-1}, y^{t-1}]$ (ver figura 2.1) mediante la siguiente ecuación recurrente (2.2.6):

$$y(t) = g(u^{t-1}, y^{t-1}) + e(t) \tag{2.2.6}$$

donde $e(t)$ es un ruido que expresa que la próxima salida no es una función exacta de los datos anteriores. El objetivo final de la predicción es encontrar la función $g()$ que haga mínimo

el error $e(t)$. Si en la ecuación (2.2.6) se ignoran las variables u, el modelo se denomina modelo de serie temporal de una variable. En éste, la carga se modela como una función de los valores pasados de consumo. Si por el contrario se utilizan factores exógenos, el modelo se denomina causal.

La predicción puede ser clasificada en función de varios criterios. Así, si tenemos en cuenta la magnitud que va a ser predicha, podemos clasificarla en predicción de pico de carga (valor máximo de consumo para un día), predicción de la carga integral (consumo total de un día) o carga horaria (consumo hora a hora). Por otro lado, si atendemos a criterios de intervalo de predicción, se puede distinguir entre predicción a muy corto plazo (predicción de la demanda desde varios minutos a varias horas), predicción a corto plazo (desde unas horas a varias semanas) y predicción a medio y largo plazo (desde meses a varios años). Los horizontes de predicción más importantes son el semanal, el diario y el horario. Especialmente, el horizonte de 24 horas juega un papel vital debido a que puede condicionar el funcionamiento óptimo de la planificación horaria de unidades térmicas, la entrada o salida de servicio de las plantas de generación, la compra/venta en sistemas interconectados, etc.

Las técnicas de regresión permiten hacer predicciones de los valores de demanda a partir de los valores de otras variables independientes. Estos métodos buscan la función$g()$ de la ecuación (2.2.6) como una parametrización con un vector de dimensión finitaΘ, convirtiendo el problema en uno de minimización de la diferencia cuadrática entre el modelo y los datos reales de todos los patrones según la ecuación (2.2.7):

$$min \sum_{i}^{P} \|y(t) - g(u^{t-1}, y^{t-1}, \Theta)\|^2 \tag{2.2.7}$$

En estos modelos se pueden considerar factores exógenos como meteorológicos (temperatura, grado de nubosidad, velocidad del viento, grado de humedad, niveles pluviométricos), estacionales, condiciones socioeconómicas (días laborales o festivos) y condiciones culturales (periodos especiales como Navidad, Semana Santa, Ramadán, etc.). La función$g()$ se suele descomponer en dos partes: por un lado, el vectorξ de observaciones pasadas; por otro, en una aplicación de dicho vector en el espacio de salidas a predecir:

$$g(u^{t-1}, y^{t-1}, \Theta) = g(\xi(t), \Theta) \tag{2.2.8}$$

$$\xi(t) = \xi(u^{t-1}, y^{t-1}) \tag{2.2.9}$$

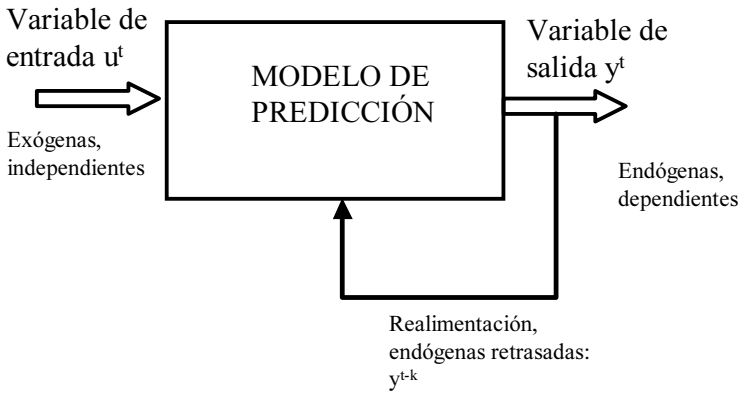


Figura 2.1: Diagrama de un modelo de predicción general.

donde:

- $\xi(t)$ es el **vector de regresión**.
- Las componentes u^t son los **regresores**, variables de entrada, explicativas, exógenas o independientes.
- Las componentes y^t son las variables de salida, endógenas o dependientes.

Entre los modelos que han sido aplicados con éxito en la tarea de predicción de demanda eléctrica cabe destacar los modelos autorregresivos multiplicativos (AR) ([Mbamalu and El-Hawary, 1993]), modelos lineales y no lineales dinámicos ([Sadownik and Barbosa, 1999]), modelos autorregresivos *threshold* ([Huang, 1997]) y los modelos basados en filtros de Kalman ([Infield and Hill, 1998; Park et al., 1991b; Sargunaraj et al., 1997]), los basados en funciones de transferencia de Box Jenkins ([Hagan and Behr, 1987], [Juberias et al., 1999]), los modelos ARMAX ([Yang and Huang, 1998], [Yang et al., 1996]), técnicas de optimización ([Yu, 1996]), regresión no paramétrica ([Charytoniuk et al., 1998]), modelos estructurales ([Harvey and Koopman, 1993]) y los modelos basados en ajuste de curvas ([Taylor and Majithia, 2000]). Sin embargo, los métodos más empleados son los basados en regresión lineal ([Engle et al., 1992], [Haida and Muto, 1994], [Papalexopoulos and Hesterberg, 1990], [Ramanathan et al., 1997], [Soliman et al., 1997]) y los que descomponen la carga en *carga base* y *carga dependiente de las condiciones meteorológicas* ([Bunn and Farmer, 1985], [Fan and McDonald, 1994], [Hyde and Hodnett, 1997], [Park et al., 1991b]). En [Amjady, 2001] se utiliza un modelado de series

temporales similar al modelo ARIMA, pero que incorpora explícitamente el conocimiento de operadores experimentados.

Dependiendo de los regresores utilizados para modelar el sistema, se suele hablar de los siguientes modelos más significativos:

- Modelo de media móvil (MA, del inglés *Moving Average*), también llamado método de respuesta de impulsos finitos, que es el modelo más simple debido a que el vector de regresión sólo depende de las salidas anteriores, $\xi(t) = [y(t-1), y(t-2), ..., y(t-k)]$.
- Modelo autorregresivo con variables de entrada exógenas (ARX, del inglés *AutoRegressive with eXogenous inputs*), el cual utiliza las entradas y salidas anteriores como regresores, permitiendo variables exógenas. También se conoce como modelo serie-paralelo.
- Modelo autorregresivo de media móvil con variables de entrada exógenas (ARMAX, del inglés *AutoRegressive Moving Average with eXogenous inputs*), que, además de los regresores ARX, permite introducir también como regresores los errores de predicción, es decir, $\xi(t-k) = y(t-k) - \hat{y}(t-k|\Theta)$.
- Modelo autorregresivo de media móvil integrado con variables de entrada exógenas (ARIMAX, del inglés *AutoRegressive Integrated Moving Average with eXogenous inputs*). Este modelo permite introducir, además de los regresores ARMAX, las sumas de los valores actuales y pasados.

Entre los inconvenientes de los métodos clásicos cabe destacar los siguientes:

1. La mayoría de estos sistemas son básicamente lineales, mientras que la serie temporal que intentan modelar es altamente no lineal.
2. Para el uso efectivo de estos métodos es necesario realizar un análisis de intervención posterior. Esto conlleva graves limitaciones para las características de flexibilidad y exportabilidad.
3. Dificultad para adaptarse a variaciones meteorológicas y socioeconómicas imprevistas.
4. Necesidad de conocer con precisión las series exógenas que intervienen en su modelo.

Por otro lado, la ventaja principal de estos métodos es que, en general, son fácilmente interpretables, permitiendo a los usuarios del sistema comprender en gran medida su comportamiento.

Si bien en la práctica es relativamente sencillo diseñar modelos estadísticos que obtengan predicciones de demanda con un error relativo medio de un 10 % [Hippert et al., 2001], es necesaria la reducción de este error por el ahorro económico que implica, incluso aunque la reducción del error relativo sea mínima ([Bunn and Farmer, 1985; Bunn, 2000]). Sin embargo, dicha reducción es una tarea muy compleja debido principalmente a las siguientes dificultades:

a) La demanda de energía eléctrica no sólo varía con la hora del día, sino que se ve influida por la temperatura, la nubosidad, el día de la semana, la estación del año, además de otros factores, con lo que es necesario considerar un gran número de variables muy diferentes entre sí en naturaleza (binarias, analógicas) y magnitud.

b) Las funciones que relacionan entradas y salidas son extremadamente complejas y desconocidas.

c) Las reglas de predicción usadas en un entorno climático, social, cultural y económico particular no pueden ser exportadas a otros entornos diferentes.

d) El cambio en los hábitos de consumo y su crecimiento a lo largo del tiempo, obliga a la revisión periódica del modelo matemático desarrollado.

Estos inconvenientes han motivado el uso de RNAs para la predicción de series temporales en general y para la predicción de la demanda eléctrica en particular.

2.3. Redes neuronales para procesamiento de series temporales

Una de las principales ventajas de las RNAs es su capacidad de simular sistemas complejos cuyo comportamiento no puede ser predicho en simulaciones por ordenador siguiendo algoritmos tradicionales. La predicción de la demanda de energía eléctrica es uno de estos sistemas, ya que contiene un gran número de variables con relaciones muy complejas para

que sean puestas de manifiesto de forma explícita. Gracias a su capacidad de aprendizaje a partir de ejemplos, corto tiempo de respuesta en la fase de operación, y flexibilidad, las RNAs han encontrado en el problema de la predicción de carga una amplia aplicabilidad, convirtiéndose en la tarea más frecuentemente realizada mediante este tipo de técnicas entre todas las implicadas en un sistema de gestión de energía.

A continuación, presentamos los modelos neuronales más utilizados para la predicción de series temporales, diferenciando entre los modelos que utilizan redes alimentadas hacia delante (*feed-forward* en inglés) y redes realimentadas.

2.3.1. Redes neuronales alimentadas hacia delante

Entre los paradigmas más utilizados para problemas asociados a la aproximación de funciones se encuentran el Perceptrón multicapa (MLP, *MultiLayer Perceptron*) [Rumelhart et al., 1986] y las redes de funciones base radiales (RBF, *Radial Basis Functions networks*) [Broomhead and Lowe, 1988; Moody and Darken, 1989] . Ambos paradigmas pueden ser utilizados como aproximadores funcionales universales [Cybenko, 1989], es decir, que pueden aproximar cualquier función $F(\vec{p}) : \Re^n \rightarrow \Re^m$ con un error arbitrariamente bajo. En el caso de un MLP con capa de salida lineal, la función F es aproximada mediante la ecuación (2.3.1):

$$F_l^{MLP} = (\sum_{j=1}^{k} v_{jl}\sigma(\sum_{i=1}^{n} w_{ij}p_i - \theta_j) - \theta_l), l = 1..m \tag{2.3.1}$$

donde σ es la función de activación sigmoide[1], n y k son el número de unidades de la capa de entrada y de la capa oculta respectivamente, w_{ij} y v_{il} son pesos, θ_l son polarizaciones o *bías*, l es el número de unidades de salida y p_i son las entradas. En el caso de la RBF, F es aproximada por la ecuación (2.3.2):

$$F_l^{RBF} = (\sum_{j=1}^{k} v_{jl}\Gamma(\sum_{i=1}^{n}(w_{ij} - p_i)^2) - \theta_l), l = 1..m \tag{2.3.2}$$

donde Γ es la función dada por la ecuación (2.3.3).

$$\Gamma(x) = e^{\frac{-x^2}{\sigma^2}} \tag{2.3.3}$$

[1] O cualquier otra función que cumpla las siguientes condiciones: i) debe ser no lineal, ii) no puede ser una función polinomial y iii) debe ser derivable

En ambos casos, para conseguir la aproximación requerida es necesario que el número de neuronas ocultas sea suficiente.

Tanto el Perceptrón multicapa como la red de funciones base radiales pueden utilizarse para producir un modelo equivalente al utilizado en los modelos autorregresivos lineales. Estos modelos suponen que la ecuación (2.2.5) es una combinación lineal de un número fijo de vectores anteriores de la serie temporal[2]:

$$x(t) = \sum_{i=1}^{p} \alpha_i x(t-i) + \epsilon(t) \tag{2.3.4}$$

Esta ecuación utiliza p secuencias previas de la serie temporal, por lo que define un modelo autorregresivo de orden p (AR[p]), y requiere aproximar la función lineal $F^L(x(t-1), ..., x(t-p)) + \epsilon(t)$, que puede ser implementada mediante la ecuación (2.3.1) o la ecuación (2.3.2), que pueden aproximar una función lineal de la forma $F^{NN}(x(t-1), ..., x(t-p)) + \epsilon(t)$. Las principales limitaciones de este método son la suposición de que existe una dependencia lineal entre los elementos de la serie y la suposición de que la serie temporal es estacionaria. La aproximación lineal por los modelos MLP y RBF se realiza utilizando como entradas las p secuencias anteriores (una ventana de p unidades), transformando la dimensión temporal del problema en una dimensión espacial. La aproximación no lineal de los paradigmas MLP y RBF presenta la siguientes características:

- Teóricamente no requieren que la serie temporal sea estacionaria.
- Puesto que representan modelos matemáticos no lineales, pueden modelar características más complejas de la serie temporal que los métodos clásicos.
- Para obtener un buen modelo se requiere una gran cantidad de datos.
- Presenta los problemas característicos de las redes MLP y RBF en cuanto al proceso de aprendizaje, es decir, posibilidad de sobreentrenamiento, posible caída en mínimos locales, etc.

En aplicaciones prácticas donde la cantidad de datos disponibles no es muy grande puede ser preferible utilizar un modelo lineal, incluso aunque el sistema presente características no lineales [Dorffner, 1996].

[2]Por simplicidad en la notación se asume que la serie temporal es univariante

Con los modelos neuronales descritos anteriormente la codificación temporal de los patrones se hace de forma implícita en forma de vectores de entrada o de ventanas, modificando los patrones de entrada para que cada vector contenga los patrones temporales ordenados con respecto al tiempo.

La mayoría de los trabajos basados en redes alimentadas hacia delante pueden clasificarse en dos grandes grupos en función del número de neuronas de salida que utilizan. En primer lugar están los modelos que implementan una sola neurona de salida. Esta neurona se utiliza para predecir el consumo de la siguiente hora, el pico de consumo para el día siguiente o el consumo total (carga integrada) para el día siguiente. En un segundo grupo tenemos redes neuronales con varios nodos de salida, típicamente 24, para predecir el consumo hora a hora del día siguiente (normalmente denominado perfil diario de carga). Entre los trabajos que pertenecen al primer grupo cabe destacar [Park et al., 1991a], que utiliza tres redes neuronales para predecir la carga de la hora siguiente, el pico de carga para el día siguiente y la carga integrada para el día siguiente. En [Ho et al., 1992] se utiliza un MLP para predecir el pico de carga del día siguiente, que posteriormente es utilizado por un sistema experto para predecir el perfil del día siguiente. En [Senjyu et al., 2002] se clasifican los días según la similitud de los perfiles de carga, y se utiliza una red neuronal que es entrenada para generar una corrección a dicho perfil. Eligiendo un perfil similar, junto con la corrección predicha por la red, se obtiene el perfil de carga para el nuevo día. La principal ventaja de este modelo está en que para el entrenamiento de la red neuronal se utiliza un número pequeño de datos históricos, lo que hace el entrenamiento mucho más rápido que en los demás casos.

Los modelos que utilizan redes neuronales con una única salida también han sido empleados para predecir perfiles de carga, generalmente utilizando uno de los siguientes métodos:

a) Realizar predicciones de carga repetidamente y hora a hora, utilizando como datos de entrada los datos predichos anteriormente cuando éstos no están disponibles ([Drezga and Rahman, 1998]).

b) Utilizando en paralelo 24 redes neuronales, cada una de las cuales realiza la predicción de una hora del día siguiente ([Connor, 1996], [McMenamin and Monforte, 1998], [Vermaak and Botha, 1998])

2.3.2. Aplicación de las redes alimentadas hacia delante a la predicción de la demanda de energía eléctrica

La mayoría de los trabajos que predicen el perfil de consumo para el día siguiente utilizan redes neuronales de 24 salidas, una para cada hora a predecir. Los sistemas propuestos para abordar este problema varían en el tipo de clasificación previa de datos que realizan, en el número de redes neuronales que utilizan y en cómo se preprocesan los datos antes de ser utilizados por la red. En [Lee et al., 1992] se divide el perfil diario en 3 periodos, que son predichos independientemente con 3 redes neuronales diferentes. En [Chow and Leung, 1996] se utilizan las redes neuronales como extensión del modelo ARIMA, utilizando como datos de entrada a la red las diferencias de primer orden de la serie. La principal ventaja de este método es que convierte la serie temporal en una más estacionaria. Otros trabajos ([Mohammed et al., 1995], [Srinivasan et al., 1994], [Srinivasan et al., 1999]) abordan la no estacionalidad de la serie mediante la modificación de la serie temporal original en otra cuyo crecimiento medio es cero. En [Connor, 1996] se utilizan filtros de Kalman para esta misma tarea.

En otro grupo de trabajos se proponen sistemas con más de una red neuronal para realizar la predicción. Así, en [AlFuhaid et al., 1998] se utiliza una RNA para preprocesar algunos de los datos de la serie temporal y producir una estimación de la carga de pico, de la carga valle y de la carga integrada para el día siguiente. Éstas estimaciones, junto con otros datos históricos y predicciones para datos exógenos, son utilizadas como entradas a otra RNA que predice el perfil de carga para el día siguiente. En [Swarup and Satish, 2002] se utilizan tres redes neuronales cuyos resultados promediados generan el valor del perfil de consumo para el día siguiente. La primera red se entrena para predecir la componente base del consumo (también denominado consumo en estado estable), así como los componentes aleatorios de este consumo; la segunda red predice los picos y valles de consumo; finalmente, una tercera red neuronal predice el consumo medio. Los resultados de los tres módulos son promediados para generar la predicción final. En [Lamedica et al., 1996] se proponen un total de 12 RNAs, para predecir el perfil de cada uno de los meses del año. Adicionalmente, para mejorar los resultados para días festivos, se utiliza un clasificador de Kohonen para obtener días festivos de perfil similar.

En [Khotanzad et al., 1995] se propone un sistema compuesto por 38 redes. Éstas

producen resultados de predicción con horizontes horario, diario y semanal. Los resultados se combinan linealmente para obtener la predicción final. Este sistema es mejorado en [Khotanzad et al., 1997], donde el número de redes se reduce a 24, una para cada hora del día, obteniendo errores de predicción mejores. Posteriormente, en [Khotanzad et al., 1998] se reduce el número de redes neuronales a dos. La primera red se entrena para predecir un primer perfil para el día siguiente. La segunda red se utiliza para predecir el cambio en el incremento o decremento del consumo del día siguiente con respecto al día actual. La predicción final es una combinación lineal de las predicciones de ambas redes. Según los autores, la ventaja de este último sistema es que la segunda red neuronal se adapta mejor a cambios bruscos en consumo debidos a cambios en las condiciones meteorológicas.

En [Mohammed et al., 1995] se realiza una clasificación previa de los datos en tres niveles diferentes: una primera clasificación crea 7 clases atendiendo a estaciones del año; una segunda clasificación crea 3 clases para días de la semana y, finalmente, un día se divide en 5 periodos. Para cada clase existe una red neuronal especializada que realiza la predicción del consumo.

Si bien las redes neuronales alimentadas hacia delante son las que más se han utilizado en la bibliografía para realizar tareas de predicción de demanda, éstas presentan una serie de problemas importantes frente a los modelos recurrentes, entre los que cabe destacar los siguientes:

1) Hay que implementar un sistema que realice la colocación de datos en el buffer de forma ordenada.

2) Es necesario desarrollar un mecanismo de sincronización para que la red examine el buffer de entrada únicamente cuando éste esté lleno

3) El buffer de entrada debe ser lo suficientemente grande para soportar la entrada de vectores con una dimensión suficientemente grande.

4) Todos los patrones de entrada deben tener el mismo tamaño.

5) Esta codificación no distingue bien entre posiciones temporales relativas y absolutas.

6) Empleo conjunto de variables analógicas y digitales. [Khotanzad et al., 1998]

Estos problemas justifican, desde el punto de vista teórico, la utilización de paradigmas recurrentes para el análisis de series temporales.

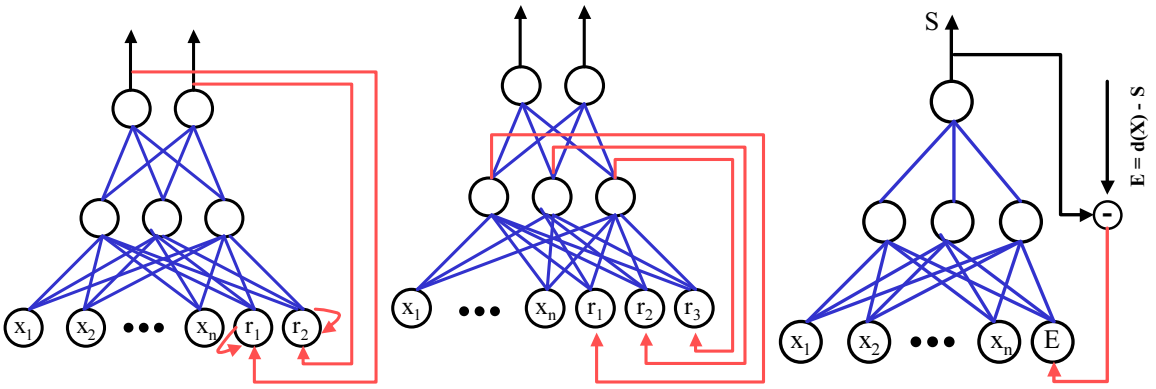
2.3.3. Redes neuronales recurrentes

Una alternativa a la utilización de buffers o ventanas para la inclusión del tiempo en el modelo neuronal se basa en la utilización de realimentación. Esta metodología permite representar el tiempo por el efecto que éste tiene en el procesamiento [Elman, 1990] y aporta propiedades dinámicas que son sensibles a secuencias temporales, es decir, aporta elementos de memoria adicional en el modelo neuronal. Las conexiones recurrentes permiten a las neuronas ocultas conocer sus salidas anteriores, y así el comportamiento futuro puede ser modelado por respuestas previas a la actual.

Existen diferentes modelos de RNA recurrentes, en función de la configuración de las realimentaciones. El modelo más directo consiste en la utilización de un perceptrón donde se añaden conexiones entre cada par de neuronas. Esta estructura tiene desventajas importantes con respecto a las redes alimentadas hacia delante, fundamentalmente debido a que los algoritmos de entrenamiento son mucho más complejos de implementar con esta estructura.

Una alternativa es la utilización de las denominadas redes parcialmente recurrentes. La figura 2.2 representa la estructura de tres redes de este tipo. En la figura 2.2(a) se muestra una red de Jordan [Jordan, 1989]. En esta red, las salidas son realimentadas como nuevas entradas. La figura 2.2(b) muestra una red de Elman [Elman, 1990], donde el valor de cada neurona de la capa oculta es realimentada como una nueva entrada para la siguiente iteración. Estos dos modelos neuronales tienen la ventaja de que, con mínimas adaptaciones, pueden entrenarse utilizando los mismos algoritmos que han sido diseñados para el Perceptrón Multicapa, lo que implica disponer de muchos algoritmos altamente optimizados. Finalmente, la figura 2.2(c) muestra una red del tipo recurrente residual [Connor et al., 1994]. Esta arquitectura realimenta el error de predicción actual como una entrada para la próxima iteración. En comparación con los modelos de Jordan y Elman, este modelo tiene el inconveniente de que sólo puede utilizarse para producir una salida.

A continuación, presentamos brevemente la aplicabilidad de cada uno de estos modelos al problema particular de la predicción de series temporales.



(a) (b) (c)

Figura 2.2: Diagrama de las redes realimentadas tipo Jordan (a), Elman (b) y Residual (c).

2.3.3.1. Redes recurrentes de Jordan

El modelo de media móvil de orden q MA[q] [Box et al., 1994] supone que una serie temporal puede modelarse como una combinación lineal de$q$ señales que contienen ruido ϵ, según la ecuación (2.3.5):

$$x(t) = -\sum_{i=1}^{q} \beta_i \epsilon(t-i) + \epsilon(t) \tag{2.3.5}$$

El modelo MA[q] implementa un filtro lineal que puede eliminar cualquier rango de frecuencias dentro de los límites de linealidad del modelo, conduciendo a una serie temporal no aleatoria específica.

Es posible utilizar una combinación de los modelos AR[p] y MA[q], que genera el denominado modelo ARMA[p,q] dado por la ecuación (2.3.6):

$$\begin{aligned} x(t) &= \sum_{i=1}^{p} \alpha_i x(t-i) - \sum_{i=1}^{q} \beta_i \epsilon(t-i) + \epsilon(t) \\ &= F^L(x(t-1), .., x(t-p), \epsilon(t-1), ..., \epsilon(t-q)) + \epsilon(t) \end{aligned} \tag{2.3.6}$$

Al igual que los modelos AR[p] y MA[q], el modelo ARMA[p,q] está limitado por la linealidad del modelo matemático y la necesidad de que la serie temporal sea estacionaria. En [Connor et al., 1992] se presenta una extensión de este modelo al caso no lineal mediante RNAs. Para implementar este modelo es necesario conocer los valores deϵ, que realmente no se conocen.

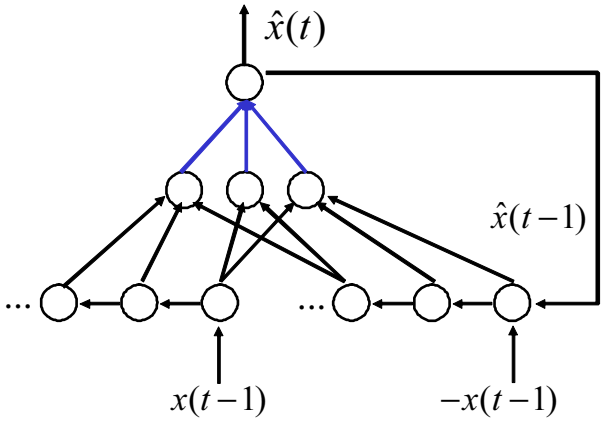


Figura 2.3: Red neuronal con la salida realimentada para implementar un modelo ARMA no lineal.

Una alternativa es utilizar una estimación $\hat{\epsilon}$ para el termino ϵ dada por la ecuación (2.3.7):

$$\hat{\epsilon}(t) = \hat{x}(t) - x(t) \tag{2.3.7}$$

Es decir, utilizamos como estimación del error la diferencia entre el valor estimado (predicho) y el valor real en el instante t. Para el caso univariante, este sistema se implementa mediante la red de la figura 2.3, cuya salida es $\hat{x}(t)$, que es realimentada a una neurona de entrada especial. Esta neurona también recibe el valor $-x(t)$ en la siguiente iteración (es decir, cuando en la salida de la red se genera el valor $\hat{x}(t)$, la salida de la neurona especial es $\hat{x}(t-1) - x(t-1)$). Empleando una capa de entrada con retrasos o mediante una ventana adecuada, esta red puede implementar un modelo ARMA[p,q] no lineal arbitrario para el modelado de la serie temporal dada por la ecuación (2.3.8):

$$x(t) = F^{NN}(x(t-1), ..., x(t-p), \hat{\epsilon}(t-1), ..., \hat{\epsilon}(t-q)) + \epsilon(t) \tag{2.3.8}$$

La red neuronal de la figura 2.3 es un caso particular[3] de la red neuronal de Jordan representada en la figura 2.2(a). Ignorando las autoconexiones, esta red crea un modelo no lineal que es una función de las p secuencias anteriores y de las q estimaciones pasadas, dado por la ecuación (2.3.9).

$$x(t) = F^{NN}(x(t-1), ..., x(t-p), \hat{x}(t-1), ..., \hat{x}(t-q)) \tag{2.3.9}$$

[3]Red con una única salida y con los pesos de autoconexiones con un valor cero.

Por otro lado, las recurrencias de la red permiten hacer dependiente la salida actual de todas las estimaciones anteriores [Dorffner, 1996], es decir, el valor de las unidades de contexto se calculan recursivamente como una combinación no lineal de todas las estimaciones $\hat{x}$ anteriores. Por tanto, pueden contener información temporal más allá de la ventana de tiempo utilizada (p o q). En un modelo más general con autorecurrencias, las salidas de las neuronas de contexto pueden llegar a saturarse (pesos mayores que 1) o tender a cero (pesos menores que 1), lo que hace que esta información pasada se pierda. De esta forma, las redes recurrentes con auto-conexiones en las unidades de contexto están limitadas en la representación de información pasada [Dorffner, 1996].

2.3.3.2. Redes recurrentes de Elman

Otro método clásico para procesamiento de series temporales es el denominado *Modelo de Espacio de Estado*, que se fundamenta en la suposición de que una serie temporal puede ser descrita como una transformación lineal de un vector de estado $\vec{s}(t)$ según la ecuación (2.3.10):

$$\vec{x}(t) = C\vec{s}(t) + \vec{\epsilon}(t) \tag{2.3.10}$$

siendo C una matriz (matriz de transformación del sistema). Por su parte, el vector de estado se puede describir como un modelo lineal mediante la ecuación (2.3.11):

$$\vec{s}(t) = A\vec{s}(t-1) + B\vec{\eta}(t) \tag{2.3.11}$$

donde A y B son matrices y $\vec{\eta}(t)$ es un vector de errores. Este modelo es básicamente un modelo ARIMA[1,1] [Dorffner, 1996], que está basado en la denominada hipótesis de Markov, que establece que toda la historia pasada de una serie necesaria para producir un elemento de la secuencia puede ser expresada mediante un vector de estado. Si, además, se asume que el vector de estado depende de la secuencia anterior, el vector de estado se puede expresar según la ecuación (2.3.12):

$$\vec{s}(t) = A\vec{s}(t-1) + D\vec{x}(t-1) + B\vec{\eta}(t) \tag{2.3.12}$$

donde D es una matriz. Si se elimina el término de media móvil $B\vec{\eta}(t)$, la ecuación (2.3.12) describe el modelo matemático de una red recurrente de Elman: si el vector de activación

de la capa oculta de esta red se iguala al vector de estado $\vec{s}$, la red de Elman puede verse como un sistema cuyo vector de estado viene dado por la ecuación (2.3.13):

$$\vec{s}(t) = F(A\vec{s}(t-1) + D\vec{x}(t-1)) \tag{2.3.13}$$

donde F es la función de activación de las neuronas ocultas de la red.

Los modelos neuronales recurrentes para la predicción de carga han sido utilizados en menor medida que los alimentadas hacia delante. En [Darbellay and Slama, 2000] se utilizan tanto redes alimentadas hacia delante como recurrentes. En este trabajo los autores llegan a la conclusión de que los modelos no lineales tanto de la redes alimentadas hacia delante como recurrentes no aportan ventajas a los sistemas lineales en cuanto a la calidad de los resultados de la predicción, conclusión que está en clara contradicción con las de otros muchos trabajos publicados y con los que se presentarán en este trabajo. En [Choueiki et al., 1997a] se implementa un procedimiento para encontrar un configuración neuronal *'casi óptima'* para el problema de predicción, considerando como variables el número de neuronas, capas, funciones de activación, criterio de parada durante el entrenamiento, etc. La arquitectura que mejores resultados obtiene resulta ser una red recurrente con funciones de transferencia senoidales. El algoritmo de entrenamiento de este modelo se mejora en [Choueiki et al., 1997b] mediante un proceso iterativo de mínimos cuadrados.

2.3.4. Aplicación de las redes recurrentes a la predicción de la demanda

En [Connor et al., 1991] se compara la capacidad de las redes recurrentes y las alimentadas hacia delante. Los autores concluyen que, en general, las redes recurrentes pueden aportar resultados mejores en la tarea de predicción de carga, debido a su capacidad de modelar adecuadamente y de forma natural la serie temporal y, sobre todo, a su mejor capacidad de adaptación a la tendencia de los nuevos datos no utilizados en la fase de entrenamiento. En [Connor et al., 1994] se estudia la equivalencia del modelo de redes alimentadas hacia delante con un modelo autorregresivo no lineal (NAR) y la del modelo de redes recurrentes con un modelo autorregresivo no lineal de media móvil, concluyendo que ésta es la razón por la que las redes recurrentes son mejores para atacar problemas de predicción de series temporales con esta característica, lo que en la práctica ocurre en la mayoría de los sistemas, debido al incremento continuado del consumo eléctrico.

En [Srivastava and Veankataraman, 1997] se propone una red recurrente derivada de una red de base radial. La red se utiliza para predecir el consumo de pico y el consumo valle para un tipo de día particular, utilizando estos resultados para la predicción de un perfil horario a partir de curvas normalizadas. Los autores señalan que los resultados obtenidos por este modelo son superiores a los obtenidos por ellos mismos utilizando redes MLP y RBF, en el sentido de reducción considerable de los errores de predicción.

En [Vermaak and Botha, 1998] se presenta un modelo recurrente que captura de forma efectiva el comportamiento dinámico de una serie temporal de demanda eléctrica. Los resultados de este sistema son comparados con una red tipo MLP, mostrando que la red recurrente ha sido capaz de capturar de forma más compacta (reduciendo el número de neuronas y pesos de la red) y más natural (en el sentido de una representación interna más similar) que el MLP.

En [Costa et al., 1999] se implementa una red recurrente que es modificada para convertirla en una equivalente alimentada hacia delante con el objetivo de mejorar los algoritmos de entrenamiento. Durante el proceso de entrenamiento, el gradiente de los pesos de la red es aproximado usando una red alimentada hacia delante con un número fijo de capas y con el algoritmo de entrenamiento *Resilient Propagation* ([Riedmiller and Braun, 1993]), que es un tipo de aprendizaje con ganancia adaptativa más eficiente que el clásico *Backpropagation*.

En [Teixeira et al., 1999] se propone una predictor neuronal que combina aprendizaje no supervisado y supervisado para la predicción del perfil diario de carga, mediante una red tipo *Recurrent Neural Gas* (RNG) y varias redes tipo Elman. Este trabajo muestra que este nuevo paradigma es muy útil para la clasificación de series temporales, consiguiendo mejores resultados que con redes alimentadas hacia delante y recurrentes a través del tiempo.

En la sección siguiente presentamos los inconvenientes que presentan las soluciones neuronales disponibles, y que nos han llevado a proponer un nuevo modelo neuronal para esta tarea.

2.4. Problemas en la predicción de la demanda eléctrica con RNAs

Como se ha mostrado en la sección 2.3, existe una amplia gama de modelos neuronales, tanto alimentados hacia delante como recurrentes, que pueden utilizarse para la predicción de series temporales de demanda de energía eléctrica. Como se ha comentado, cualquier modelo lineal o no lineal puede ser, en principio, modelado por una RNA, presentando teóricamente ventajas sobre los modelos clásicos. Sin embargo, los modelos particulares que se han utilizado en la bibliografía no han obtenido unos resultados aceptables en muchos casos debido, entre otros, a los siguientes factores:

a) Las variables de entrada son muy diferentes en naturaleza, rango de valores y codificación. Así, un predictor de demanda puede utilizar como entradas los valores de demanda hora a hora, demanda integrada para un día, temperaturas máxima y mínima, día de la semana, estación del año, etc. Si cada una de estas variables se codifica de forma independiente, los distintos rangos y naturaleza de cada una enmascararán su papel y hará que el proceso de aprendizaje de la red sea más difícil. Esta variabilidad enmascara el papel jugado por cada variable y dificulta el aprendizaje de la ANN.

b) Si la predicción de la demanda se realiza sobre una amplia región (p.e. un pais), las variables exógenas como temperaturas, grado de nubosidad, etc., pueden presentar valores muy diferentes en distintas zonas para una misma hora. Esto conduce a la necesidad de encontrar un valor de la variable de entrada que represente lo más exactamente posible el dato para la zona en conjunto (p.e. realizando la media ponderada).

c) Algunas variables exógenas pueden tener efectos contradictorios dentro de la misma zona de estudio.

d) La selección de los datos de la serie histórica presenta graves dificultades, entre las que cabe destacar que un rango de valores anteriores óptimo para un periodo de predicción, puede no ser adecuado para otro periodo.

e) En general, los días anómalos tales como días festivos, festivos consecutivos, día anterior y posterior a día festivo, presentan dificultades extraordinarias a la hora de realizar

una predicción de demanda eléctrica, ya que sus efectos en el consumo son muy diferentes según en qué día de la semana caigan, según el mes, el año, etc. A esto se une el hecho de que hay una cantidad pequeña de datos históricos para realizar un análisis.

f) La demanda de energía crece y los hábitos de consumo cambian con el tiempo, lo que reduce el tamaño del conjunto de entrenamiento utilizable e impone un reentrenamiento periódico del sistema.

g) La mayoría de los modelos propuestos en la bibliografía pueden padecer de sobreparametrización, esto es, el número de parámetros a estimar es mayor que el número de datos de entrenamiento. La causa fundamental de este comportamiento está en una excesiva división de la serie temporal en muchas clases.

2.5. Un nuevo modelo de predicción de carga eléctrica

En esta sección presentamos la aplicación de técnicas neuronales a la predicción de la curva horaria de consumo del próximo día, y a la predicción del consumo en la próxima hora. Para soslayar en lo posible los problemas indicamos en la sección 2.4 se propone una arquitectura neuronal compleja constituida por la secuenciación de dos paradigmas diferentes (ver figura 2.4) ([Marín et al., 2000], [Marín et al., 2002]). En una primera etapa, se utiliza un paradigma de aprendizaje no supervisado (*Mapas Auto-Organizativos de Kohonen*) [Kohonen, 1990] para la clasificación de patrones. En una segunda etapa, un paradigma de topología realimentada y aprendizaje supervisado (*red de Elman*) [Elman, 1991] realiza la predicción de la demanda de manera independiente para cada una de las clases obtenidas en la primera etapa. La predicción de demanda se realiza con dos horizontes diferentes: consumo del día siguiente (perfil de 24 horas) y consumo de la hora siguiente.

La sección se ha organizado como sigue: en primer lugar se analiza el preproceso que se ha aplicado a la serie histórica de datos para poder aplicar de forma efectiva los paradigmas neuronales mencionados en el párrafo anterior. En segundo lugar, se analiza en detalle el modelo utilizado para obtener la clasificación de datos y la clasificación obtenida. En la sección 2.5.3 se analiza la aproximación de la función de consumo, junto con los resultados obtenidos. Finalmente, en la sección 2.6 se presentan las conclusiones.

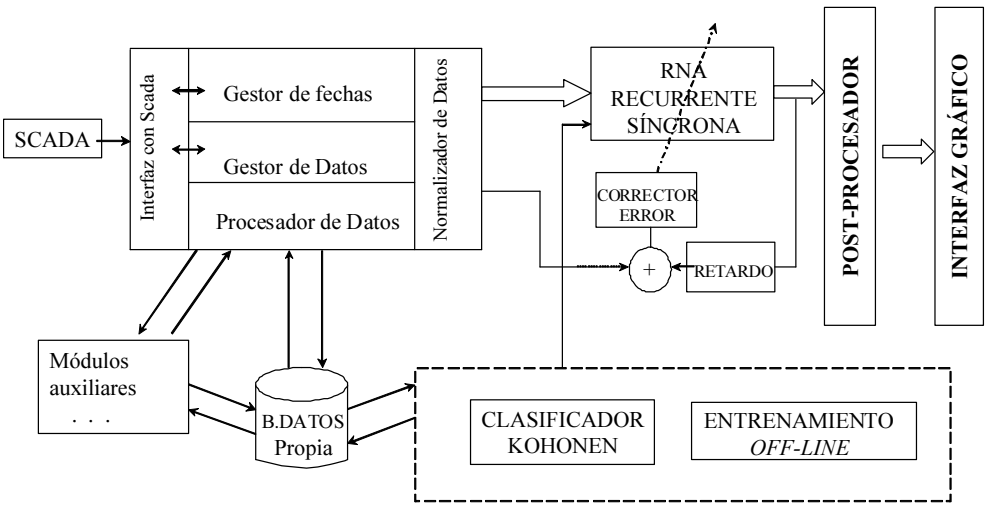


Figura 2.4: Arquitectura del sistema de predicción de carga basado en RNAs

2.5.1. Preprocesamiento de los datos

La utilización efectiva de las redes neuronales requiere la preparación o adaptación de los datos del problema a los requisitos inherentes al modelo computacional elegido. Para el caso de la predicción de la demanda de energía eléctrica, en este trabajo se han utilizado datos históricos del consumo de energía eléctrica en la zona centro de España desde el 1 de enero de 1989 hasta el 17 de febrero de 1999. Estos datos han sido preprocesados con las siguientes operaciones:

1. Se han eliminado los datos erróneos. Aunque cabe esperar que las redes neuronales sean capaces de filtrar este tipo de datos, el entrenamiento puede degradarse debido a los mismos, por lo que se han eliminado aquellos datos que resulta evidente que son erróneos.

2. Se han completado datos ausentes. Los datos horarios de consumo no disponibles han sido reemplazados por la media aritmética del consumo de la hora anterior y posterior.

3. Todos los datos son escalados mediante la transformación lineal de la ecuación (2.5.1), lo que garantiza la distribución de todas las variables en el intervalo [-1,1].

4. Se ha realizado un estudio de las correlaciones entre las variables endógenas y exógenas para conocer las variables candidatas a ser entradas a las redes neuronales.

$$Y_S(t) = I_{min}\frac{Y_U(t) - Y_{max}}{Y_{min} - Y_{max}} + I_{max}\frac{Y_U(t) - Y_{min}}{Y_{max} - Y_{min}} \tag{2.5.1}$$

En la ecuación (2.5.1), $Y_U(h)$ e $Y_S(h)$ son los valores no escalado y escalado respectivamente de la variable o variables de entrada; Y_{max} e Y_{min} son los valores máximo y mínimo, respectivamente, alcanzados por la variable a normalizar durante el periodo de datos históricos utilizado para el entrenamiento, e I_{min} e I_{max} son los límites del intervalo de escalado, [-1,1] en nuestro caso. Las salidas de la red son desnormalizadas para obtener los valores finales predichos por el sistema.

2.5.2. Clasificación mediante mapas auto-organizados de Kohonen

La clasificación de los datos se ha realizado bajo la hipótesis de que la serie temporal es más predecible cuanto más regular sea. En el caso particular del consumo de energía eléctrica, la serie está influenciada especialmente por las condiciones sociales (laboralidad, vacaciones) y las condiciones climáticas. El objetivo principal es encontrar una clasificación de forma no supervisada, que atienda tanto a criterios conocidos a priori como a similitudes en los patrones que no es posible obtener mediante un análisis manual de los datos. En este sentido, los mapas auto-organizativos de Kohonen [Kohonen, 1990] son un instrumento adecuado para la clasificación de datos ya que permiten la proyección de vectores de dimensión n en un espacio de dimensión 2, fácil de analizar de forma visual. La clasificación final se obtiene mediante una auto-organización de forma que neuronas espacialmente próximas (vecinas) responderán de una forma más fuerte a patrones de entrada similares. En nuestro caso, como medida de similitud se ha elegido la distancia euclídea.

El algoritmo que se ha utilizado para el entrenamiento de la red se esboza a continuación ([García-Lagos et al., 1999]):

1. Inicialización aleatoria de los pesos de la red.

2. Inicialización de los parámetros de entrenamiento: radio de vecindad (r_v), tasa de aprendizaje (l_r) y número máximo de iteraciones (MI).

3. Desde t = 1 hasta MI hacer

   a) Presentar un nuevo patrón a la red

   b) Seleccionar la neurona ganadora.

   c) Encontrar las neuronas vecinas de la ganadora, según el valor actual del parámetro r_v.

   d) Actualizar los pesos de la neurona ganadora y sus vecinas en el sentido de acercase al vector de entrada (ver ecuación (2.5.2)).

   e) Actualizar los parámetros r_v y l_r.

4. Terminar.

En este trabajo, se utiliza una red de Kohonen de 10x10 neuronas, formando, por tanto, un mapa de dimensión 2. Los vectores de entrada son las curvas horarias de consumo de cada día obtenidas a partir de un fichero histórico correspondiente a los años 1989 a 1997. El algoritmo de adaptación del vector de pesos viene dado por la ecuación (2.5.2),

$$\vec{w}_i(t+1) = \vec{w}_i(t) + l_r h_{iv}(\vec{x} - \vec{w}_i(t)) \tag{2.5.2}$$

donde $w_i(t)$ es el vector de pesos de la neurona i en la iteración t. Por construcción de la red, este vector tendrá la misma dimensión que el vector de entrada$x$. $h_{iv}()$, la función de vecindad, determina el incremento del peso de cada neurona como función de su proximidad a la neurona ganadora en cada momento. En nuestro caso, el área de vecindad viene determinado por un cuadrado centrado en la neurona ganadora, cuyo lado disminuye de manera discreta hasta cero a lo largo del aprendizaje, tal y como muestra la figura 2.5.l_r es la razón de aprendizaje dinámica, la cual evoluciona a lo largo de la simulación de acuerdo con la ecuación (2.5.3) [Cottrell and Letremey, 1995]:

$$l_r(t) = \frac{l_{r0}}{(1 + \frac{c \cdot t}{nn})} \tag{2.5.3}$$

siendo l_{r0} la razón de aprendizaje inicial (0,3), c una constante (0,2), t la iteración actual y nn el número de neuronas en la red.

La clasificación de los patrones que la red ha obtenido se muestra en la figura 2.6. En esta figura puede observarse cómo la red ha realizado una clasificación atendiendo principalmente

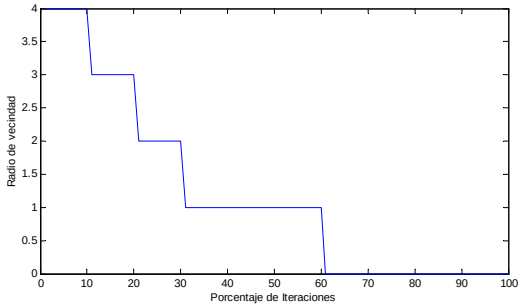


Figura 2.5: Valor del radio de vecindad durante el entrenamiento de la red de Kohonen. El eje x representa el porcentaje de iteraciones.

a dos criterios. En un primer nivel ha producido una clasificación atendiendo a criterios de estacionalidad, distinguiendo entre las estaciones del año. En un segundo nivel la clasificación atiende a criterios de laboralidad, distinguiendo entre días laborables, sábados y domingos. En cuanto a los días laborables, los lunes son separados del resto, lo que se corresponde con los hábitos de consumo de la zona analizada. Los detalles de la clasificación obtenida se muestran en la tabla 2.1.

En resumen, y de manera esquemática, el análisis de esta clasificación nos lleva a las siguientes conclusiones:

- La clasificación incorpora en un primer nivel las condiciones meteorológicas de la zona estudiada.
- En un segundo nivel, las clases se corresponden con características de laboralidad.
- Dentro de los días laborables, los lunes son clasificados separadamente, lo que corresponde con los hábitos a la zona analizada. Esta separación no ocurre en el mes de agosto, mes en el que menos afecta el carácter de laboralidad debido a que es el mes elegido por la mayoría como mes de vacaciones.
- Los días festivos son clasificados en las clases correspondientes a los domingos del mismo periodo.
- Existe cierta componente cultural como se refleja de la clasificación de los meses de julio y agosto como meses de vacaciones.

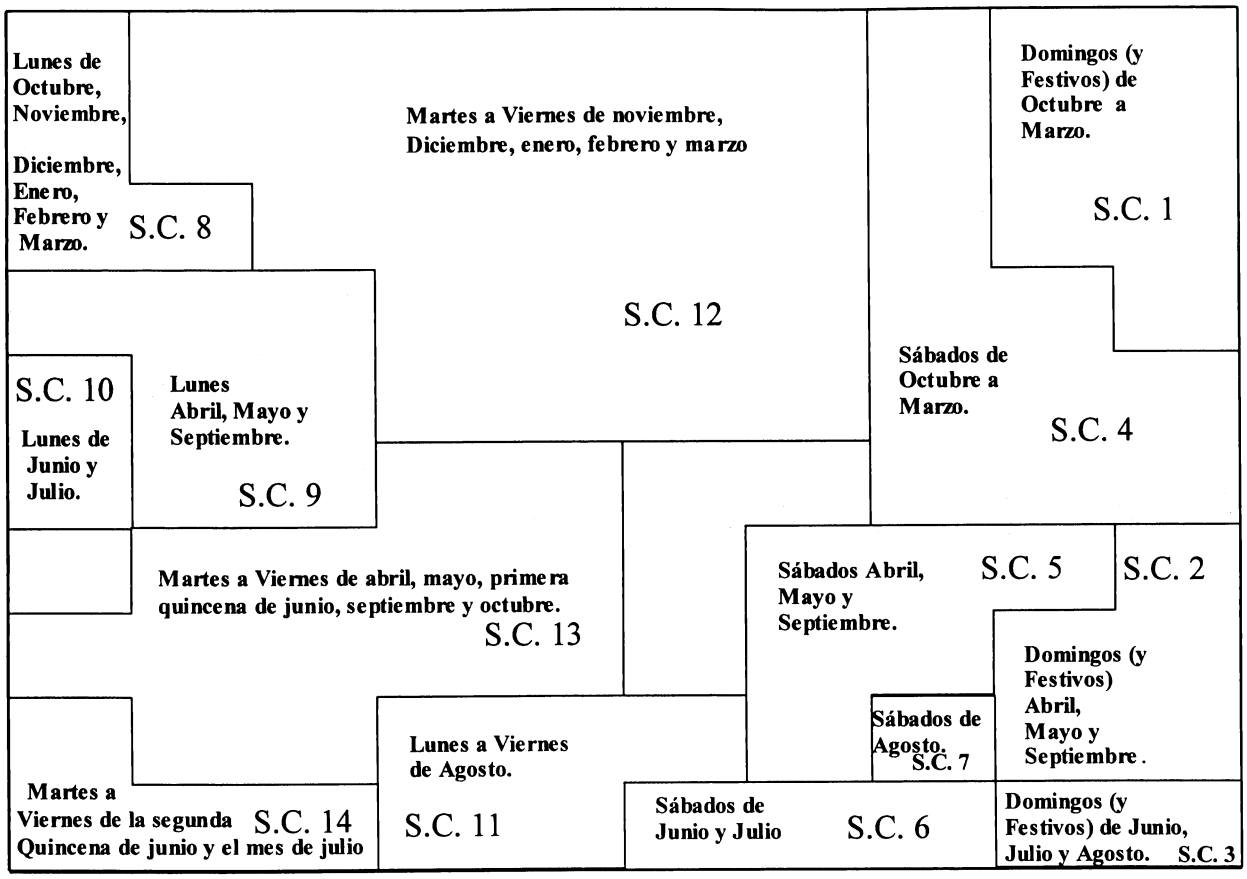


Figura 2.6: Distribución final de los perfiles de consumo diario usados en el entrenamiento de la red de Kohonen para la Zona Centro de España. S.C. indica superclase.

La aplicación de este proceso de clasificación debe hacerse una sola vez previamente a la puesta en marcha del paquete de predicción. Su importancia estriba en las siguientes aportaciones:

a) En cada ámbito de aplicación, la red obtendrá de manera no supervisada la clasificación de patrones de acuerdo con los factores socioeconómicos y culturales propios de ese ámbito. Por ejemplo, es altamente probable que en un ámbito musulmán la clase de días festivos no correspondiese a los domingos sino a los viernes, mientras que en Israel esta clase estaría compuesta por los sábados. Esto reduce al mínimo la necesidad de un análisis de intervención previo realizado por un experto, proporcionando al sistema una mayor flexibilidad y exportabilidad.

Tabla 2.1: Detalle de la clasificación de los datos de consumo eléctrico obtenidos por la red de Kohonen

Clase	Descripción
1	Domingos y festivos de Octubre a Marzo
2	Domingos y festivos de abril, mayo y septiembre
3	Domingos y festivos de junio, julio y agosto
4	Sábados de octubre a marzo
5	Sábados de abril, mayo y septiembre
6	Sábados de junio y julio
7	Sábados de agosto
8	Lunes de octubre, noviembre, diciembre, enero, febrero y marzo
9	Lunes de abril, mayo y septiembre
10	Lunes de junio y julio
11	Lunes a viernes de agosto
12	Martes a viernes de noviembre, diciembre, enero, febrero y marzo
13	Martes a viernes de abril, mayo, primera quincena de junio, septiembre y octubre
14	Martes a viernes de la segunda quincena de junio y el mes de julio

b) La separación en clases ha tenido en cuenta de manera implícita la influencia de las principales variables exógenas que intervienen en el consumo. Esto supone que el estudio posterior para cada clase por separado puede hacerse prescindiendo de estos factores y teniendo en cuenta únicamente variables de consumo (consumo horario y consumo integral de cada día en nuestro caso).

2.5.3. Aproximación de la función de consumo mediante redes de Elman

Concluida la fase de clasificación de los datos, debemos aproximar la función de consumo futuro. Para ello, el predictor propuesto utiliza una red tipo Elman para cada clase obtenida previamente. La elección de estas redes frente a otras analizadas puede justificarse escuétamente con las siguientes razones: a) Su estructura realimentada proporciona una cierta memoria a la red, que favorece su capacidad de adaptación y de predicción del valor siguiente de consumo a partir de una serie de valores sucesivos previos; b) su algoritmo de aprendizaje supervisado es una adaptación del *Backpropagation*, por lo que su aplicación es sencilla, eficiente y robusta. c) Su estructura realimentada le confiere la capacidad de adaptarse automáticamente (sin entrenamiento) a series de datos con media móvil, característica que presenta la serie temporal utilizada en este trabajo, como se mostró en la sección 2.3.3.2.

A continuación, se detalla el proceso de construcción de cada red neuronal, comenzando en la sección 2.5.3.1 por la descripción de las entradas de cada red, continuando en la sección 2.5.3.2 con el análisis del proceso de entrenamiento, para finalizar con los resultados obtenidos en la fase de operación en sección 2.5.3.3.

2.5.3.1. Selección de las entradas a las redes de Elman

La selección de las entradas a la red neuronal se ha realizado mediante un proceso de análisis de las correlaciones entre los siguientes datos históricos:

- Valor de la demanda integral del día a predecir. Demanda integrada del consumo total de cada día.
- Temperaturas del día a predecir. Estas temperaturas se corresponden a valores medios ponderados de mediciones tomadas a las 6:00 y 18:00 horas, respectivamente, en diversos observatorios meteorológicos de la zona de estudio. La ponderación tiene en cuenta la participación de cada área en la demanda total.
- Nubosidad del día a predecir. Al igual que las temperaturas mínima y máxima, estos datos corresponden a valores promediados de las diferentes áreas que componen la zona centro de España.
- Valores pasados de la serie temporal. Estos datos corresponden a la carga horaria, hora a hora.

En todos los casos se trata de valores consolidados, es decir, valores reales.

Las conclusiones principales del análisis de correlaciones se pueden resumir en los siguientes puntos:

- Existe una alta correlación lineal entre los dos datos de temperatura disponibles para cada día (6:00 y 18:00 horas).
- Correlación débil entre las temperaturas y el grado de nubosidad.

Durante el proceso de construcción de los modelos neuronales se ha comprobado que la inclusión de los datos meteorológicos no mejoran las prestaciones del sistema. Aunque

es conocido que las temperaturas mínima y máxima son datos esenciales en la predicción del pico de carga y de la carga integrada [Marín and Sandoval, 1997], en el modelo que se propone, estos datos no aportan mejoras en la función de consumo debido a los siguientes hechos:

a) Las temperaturas disponibles en la serie histórica no se corresponde con las temperaturas mínima y máxima de cada día, sino que son temperaturas medidas en dos momentos determinados y fijos (6:00 y 18:00 horas).

b) Para realizar una predicción de carga horaria, los datos de temperatura requeridos deben estar disponibles para cada hora, de forma que pueda discernirse su efecto a lo largo del día.

c) Se ha comprobado que la carencia anterior puede mitigarse, en gran parte, utilizando como entrada al modelo la demanda integrada para el día a predecir. Este dato incluye los efectos de las variables meteorológicas en el consumo, y está disponible en la serie histórica bajo estudio.

Debido a los puntos anteriores, se han descartado como entradas a las redes las temperaturas disponibles, así como el grado de nubosidad. A cambio, se utilizará la demanda integrada como entrada al modelo. Sin embargo, debido a que en nuestro país existe una alta correlación entre la demanda de energía eléctrica y la temperatura, cabe esperar que la inclusión de las temperaturas hora a hora (si estuviesen disponibles) mejore la calidad de la predicción de demanda.

Debido al proceso de clasificación, la serie de datos histórica ha quedado dividida en 14 subseries que serán predichas de forma independiente por diferentes redes neuronales. Para presentar de forma simple las entradas y salidas de la red en la siguiente sección, se introduce aquí la siguiente nomenclatura:

- $C_i(t)$ es una función que determina la clase a la que pertenece el día t, a partir de la red de Kohonen previamente entrenada.

- $Y_H(C_i(t), t-i, h-j)$ representa el consumo de carga horaria de la hora $h-j$ del día i-ésimo anterior de la subserie de la clase $C_i(t)$. Si $h-j$ es menor que 1, la función es

igual a $Y_H(C_i(t), t-i-1, h-j+24)$, es decir, los valores anteriores de demanda que esta función devuelve son siempre datos pertenecientes a la misma clase.

- $Y_I(t)$ es la demanda integrada predicha para el día t.

Como conclusión final del análisis de los datos disponibles, las entradas a las redes neuronales se muestran a continuación:

Entradas al módulo de predicción del perfil diario para predicción del perfil del día t

Consumo horario del día anterior de la misma clase, es decir, $Y_H(C_i(t), t-1, h)$ para $h = 1, ..., 24$, junto con la demanda integrada predicha para el día t:$Y_I(t)$. Para este módulo, todas las redes neuronales tienen las mismas entradas.

Entradas al módulo de predicción de la demanda horaria de la hora $h+1$ **del día** t

- Para clases de periodicidad semanal (clases 1 a 10):$Y_H(C_i(t), t, h-1)$, $Y_H(C_i(t), t, h-2)$, $Y_H(C_i(t), t, h-3)$, $Y_H(C_i(t), t, h-166)$, $Y_H(C_i(t), t, h-167)$, $Y_H(C_i(t), t, h-168)$, $Y_H(C_i(t), t, h-169)$, $Y_H(C_i(t), t, h-170)$, $Y_H(C_i(t), t, h-171)$, $Y_H(C_i(t), t, h-172)$, $Y_H(C_i(t), t, h-173)$, $Y_H(C_i(t), t, h-174)$, $Y_H(C_i(t), t, h-175)$, $Y_H(C_i(t), t, h-176)$, $Y_H(C_i(t), t, h-177)$, y la demanda integrada predicha $Y_I(t)$.

- Para las clases 11 a 14: $Y_H(C_i(t), t, h-1)$, $Y_H(C_i(t), t, h-2)$, $Y_H(C_i(t), t, h-3)$, $Y_H(C_i(t), t, h-22)$, $Y_H(C_i(t), t, h-23)$, $Y_H(C_i(t), t, h-24)$, $Y_H(C_i(t), t, h-25)$, $Y_H(C_i(t), t, h-26)$, $Y_H(C_i(t), t, h-27)$, $Y_H(C_i(t), t, h-166)$, $Y_H(C_i(t), t, h-167)$, $Y_H(C_i(t), t, h-168)$, $Y_H(C_i(t), t, h-169)$, $Y_H(C_i(t), t, h-170)$, $Y_H(C_i(t), t, h-171)$, $Y_H(C_i(t), t, h-172)$, $Y_H(C_i(t), t, h-173)$, $Y_H(C_i(t), t, h-174)$, $Y_H(C_i(t), t, h-175)$, $Y_H(C_i(t), t, h-176)$, $Y_H(C_i(t), t, h-177)$, y la demanda integrada predicha $Y_I(t)$.

Como entrada a la red del predictor de demanda horaria no se ha utilizado el dato de consumo de la hora h, momento en el cual se procede a la predicción de la hora $h+1$, ya que este dato no está consolidado en la hora h, disponiéndose sólo de un valor predicho. En las distintas pruebas del modelo durante la fase de construcción, hemos podido constatar que la inclusión de un dato predicho (y, por tanto, con un error) en nuestro modelo produce peores resultados.

2.5.3.2. Estructura de las redes neuronales de Elman

Una vez fijados los parámetros neuronales del número de neuronas de entrada y de salida, es necesario fijar el número de neuronas ocultas. Esta selección condicionará la calidad de la predicción de la demanda, ya que es conocido su influencia tanto en la fase de entrenamiento como en la de operación. Por un lado, si el número de neuronas ocultas es muy pequeño, la red será incapaz de modelar las relaciones entre las entradas y salidas. Por otro lado, si el número de neuronas ocultas es excesivamente alto, la red neuronal estará sobreparametrizada, lo que producirá un deterioro importante en los resultados en la fase de operación, debido a que la red tendrá una menor capacidad de extrapolación o generalización.

La elección de la estructura de la red neuronal puede abordarse mediante técnicas de diseño automático basadas en algoritmos evolutivos ([Marín, 1997], [Sandoval et al., 2002]), sin embargo, en este trabajo este problema ha sido abordado mediante técnicas de prueba y error. Así, independientemente para cada clase, se han probado diversas configuraciones hasta encontrar la que produce mejores resultados para cada caso. El número de neuronas ocultas varía desde 30 para la clase 7, hasta 100 para la clase 12, en el caso de la predicción del perfil de 24 horas. Para el predictor de la demanda de la hora $h+1$ se ha encontrado que una red con 100 neuronas ocultas es apropiada para todas las clases, por lo que se ha elegido este número para hacer el sistema más homogéneo. Esta asimetría en el número de neuronas ocultas se debe a la disparidad en el número de patrones de entrenamiento disponibles para los módulos horario y diario. En el caso del módulo diario, cada día es un patrón de entrenamiento, mientras que para el caso horario cada día permite la construcción de 24 patrones de entrenamiento.

Finalmente, el conjunto de datos de entrenamiento ha estado constituido por los perfiles de consumo diario de ocho años consecutivos (1 de enero de 1989 a 31 de diciembre de 1996). Los datos de este periodo se han dividido a su vez en dos conjuntos, uno para entrenamiento y otro para validación. El número de épocas de entrenamiento para cada red de Elman se muestra en la tabla 2.2, utilizándose el método de validación cruzada (*cross-validation*) como criterio de parada para el entrenamiento. Para el predictor horario el número de iteraciones para cada clase ha sido de 100.000 épocas.

Tabla 2.2: Número de épocas de entrenamiento para cada red de Elman.

Clase	Épocas (x 10^3)
1	65
2	12
3	15
4	35
5	18
6	14
7	6
8	25
9	25
10	15
11	20
12	25
13	65
14	35

2.5.3.3. Fase de operación

El esquema global del predictor de demanda en su fase de operación se muestra en la figura 2.7. Para realizar la predicción del perfil diario de carga para el día d, la primera operación que se realiza es determinar la clase a la que pertenece este día. En segundo lugar, se buscan en la base de datos los datos de entrada para alimentar a la red de Elman correspondiente a esa clase, así como el estado interno de las neuronas que implementan las realimentaciones de la red. Tras desnormalizar los datos, la red de Elman proporciona el consumo previsto para las 24 horas del día d. Finalmente, para su posterior uso, se almacena en la base de datos el estado interno de la red de Elman.

Para la evaluación del método propuesto durante la fase de prueba, se han utilizado como vectores de prueba los perfiles de consumo diario desde el 1 de enero de 1997 hasta el 17 de febrero de 1999. Este conjunto de datos es completamente independiente del conjunto de entrenamiento.

Como medidas de bondad se han utilizado el error relativo medio MAPE (del inglés *Mean Absolute Percentage Error*), es decir, la cantidad media en la que la magnitud predicha se desvía de la real, y su desviación típica. La expresión para el MAPE viene dada por la

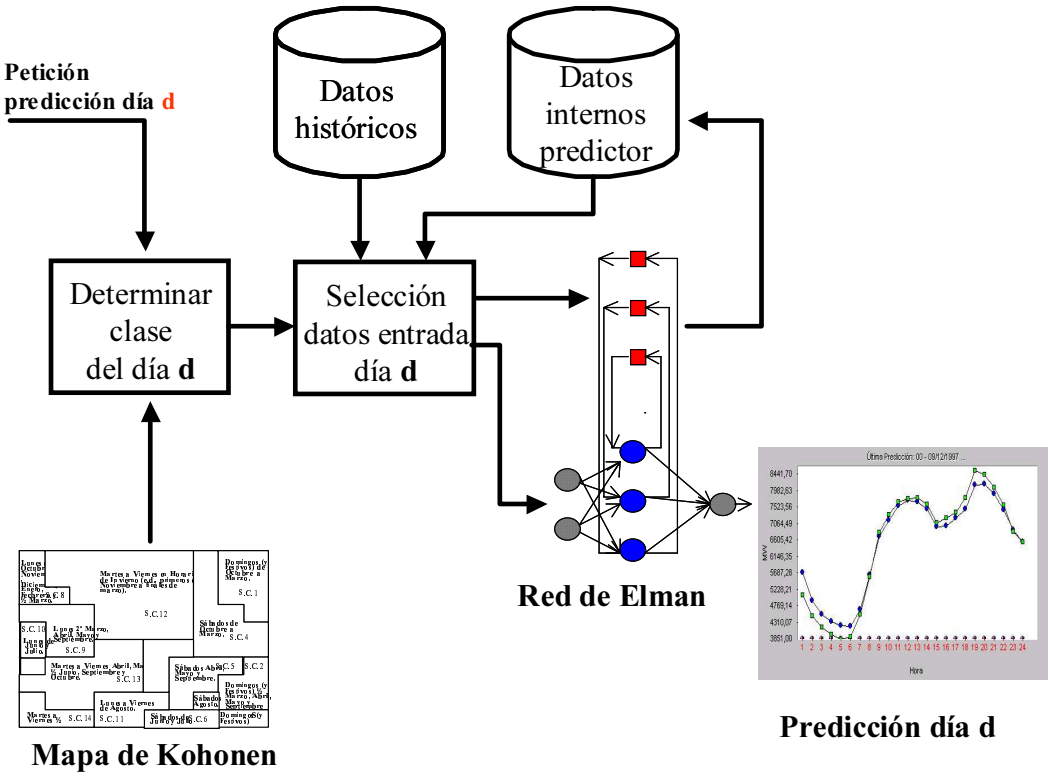


Figura 2.7: Esquema del funcionamiento on-line del predictor de carga.

ecuación (2.5.4):

$$MAPE = \frac{1}{N} \sum_{h=1}^{N} \frac{|Y_F(h) - Y_A(h)|}{Y_A(h)} \; x \; 100 \qquad (2.5.4)$$

donde $Y_F(h)$ es la carga predicha para la hora h, $Y_A(h)$ es la carga real de la hora h, y N es el número total de horas consideradas.

2.5.4. Resultados de funcionamiento y discusión

La tabla 2.3 muestra los resultados obtenidos para el valor MAPE para todas las clases en los periodos de entrenamiento y de predicción *on-line* diaria. Los resultados para el módulo horario son similares. En esta tabla se ha incluido una nueva fila, que representa una clase especial dedicada a los días de Semana Santa, como se comenta más adelante. Los errores de predicción medios de cada tipo de día han resultado ser de 1.65 % para los lunes (clases 8, 9 y 19); 1.45 % para el resto de días laborables (clases 11, 12, 13 y 14); 1.46 %

para sábados (clases 5, 6 y 7); y 1.62 % para domingos; y 1.87 % para el periodo de Semana Santa. Podemos resaltar los siguientes aspectos:

i) Las clases de días laborales (8 a 14), las cuales contienen un gran número de patrones, presentan muy buenos resultados. Es necesario remarcar que estos errores están afectados por la presencia de días especiales, cuyo efecto no ha sido tenido en cuenta en modo alguno durante el entrenamiento. Como sabemos, la predicción de carga en el sistema español es una tarea especialmente ardua debido a la distribución cambiante de estos días especiales en cuanto a día de la semana y del año. En una región en la que un alto porcentaje de días especiales fuesen fijos, el error de estas clases se reduciría apreciablemente. La figura 2.8 muestra los errores medios de entrenamiento y prueba para cada hora, para todos los patrones de la clase 14 (martes a viernes de la segunda y tercera semana de junio y todo julio).

ii) El periodo de verano (clases 3, 6, 7, 10 y 14) presenta también muy buenos resultados, tanto para días laborales como de fin de semana y festivos. La figura 2.9 muestra las curvas de predicción y consumo real para los días laborales de la segunda semana de julio (de martes a viernes). El error de predicción en este caso es de 0.61 %. Puesto que el verano constituye un periodo especial, tanto desde el punto de vista meteorológico como vacacional, estos resultados ilustran una de las principales ventajas de nuestro predictor: su capacidad de adaptación a diferentes entornos sociales, culturales y meteorológicos.

iii) Las clases con los mayores errores (clases 1, 2, 4, 5, 8 y 9) corresponden a periodos con un gran número de días especiales, tales como Navidad. La clase 1 presenta grandes errores tanto para diciembre como para los meses de transición entre estaciones (octubre y marzo). Para las clases 2, 5 y 9 el error medio para los meses de mayo y septiembre de 1997 es de un 1.42 % y 1.58 %, respectivamente, pero abril tiene un error medio de un 2.19 %, debido a que generalmente contiene el periodo festivo de Semana Santa. Este periodo es de especial interés debido a que una parte importante de los habitantes de la zona de estudio se toma días de vacaciones durante esta semana. Esto repercute en el consumo eléctrico no sólo en esta semana, sino en los días previos y siguientes a

Tabla 2.3: Errores medios de predicción para cada clase para el módulo de predicción diaria.

Clase	Entrenamiento		Prueba					
	1989-1996		1997		1998		1/1/1999 - 17/2/1999	
	MAPE(%)	STD	MAPE(%)	STD	MAPE(%)	STD	MAPE(%)	STD
1	1.39	0.76	1.55	1.33	1.74	1.42	1.42	0.91
2	1.65	1.03	1.74	1.10	1.87	1.22		
3	1.20	0.69	1.26	0.87	1.54	1.02		
4	1.37	1.06	1.52	1.19	1.66	1.28	1.61	1.23
5	1.32	0.94	1.43	1.12	1.39	0.97		
6	1.24	0.78	1.36	0.84	1.40	0.89		
7	1.09	1.02	1.32	0.68	0.64	0.42		
8	1.71	1.31	1.70	1.38	1.90	1.53	1.17	0.93
9	1.50	1.11	1.62	1.23	1.63	1.11		
10	1.30	0.90	1.38	1.26	1.34	0.99		
11	1.49	1.07	1.52	1.19	1.55	1.15		
12	1.29	0.96	1.34	0.97	1.30	0.98	1.15	0.84
13	1.40	1.05	1.47	1.14	1.67	1.20		
14	1.03	0.75	1.30	0.94	1.48	1.09		
15	1.70	1.14	1.82	1.27	1.91	1.31		

la misma. Por este motivo, se ha incluido una nueva clase[4] (clase 15) que incluye 12 días. La figura 2.10 muestra los resultados de esta clase para el año 1998. Finalmente, señalamos que no se ha incluido una clase especial para el periodo de Navidad debido a que éste produce diferentes comportamientos sociales en función del día de la semana en el que caigan los días festivos.

Como muestra la tabla 2.3, para algunos periodos de las clases 7, 8 y 12 se han obtenido errores de predicción menores que los correspondientes errores de entrenamiento. Esto se debe a que en estos periodos la serie temporal de consumo de energía eléctrica es más regular que en el correspondiente periodo de entrenamiento. Esta mayor regularidad puede deberse a muchos factores, como las condiciones climáticas, ausencia de días especiales, etc.

Como comentario final sobre el funcionamiento del sistema, señalar que para cada clase se ha implementado un algoritmo de reentrenamiento periódico. Este reentrenamiento permite a las redes neuronales adaptarse más rápidamente a los cambios en las tendencias de

[4]La inclusión de una nueva clase en el modelo neuronal de predicción propuesto supone únicamente la modificación de la red de Kohonen y la creación y entrenamiento de dos redes neuronales de Elman, una para la predicción del consumo diario y otra para la predicción del consumo de la hora $h+1$.

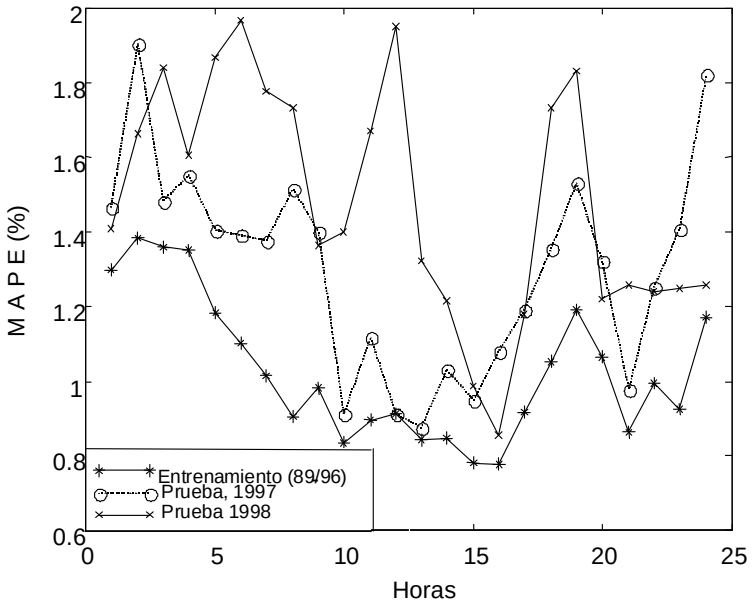


Figura 2.8: MAPE para todos los patrones de la clase 14.

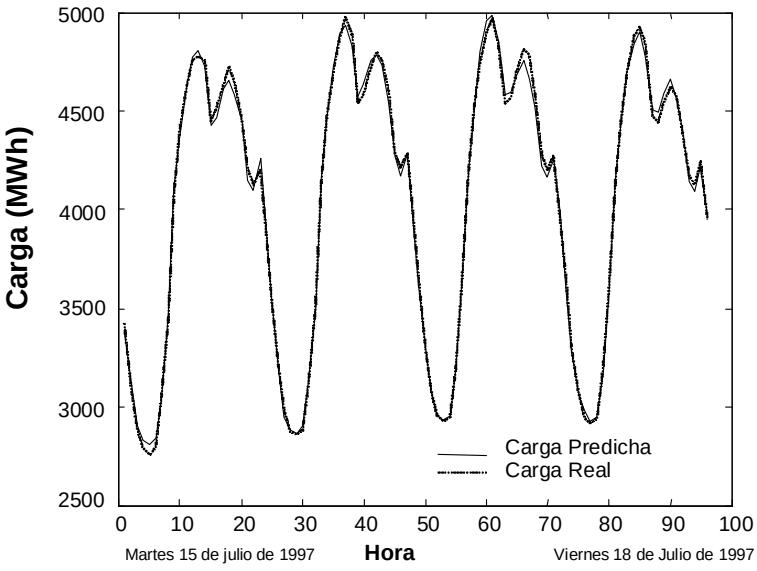


Figura 2.9: Resultados de predicción y curva real para los días laborables de la segunda semana de julio de 1997 (clase 14).

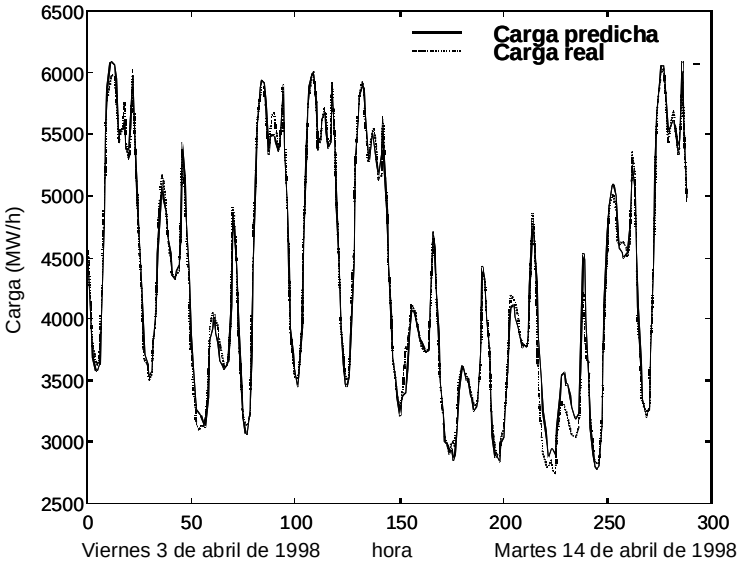


Figura 2.10: Resultados de predicción y curva real para el periodo de Semana Santa de 1998.

consumo a medio y largo plazo. El reentrenamiento periódico se realiza automáticamente a comienzo de año, y su coste computacional es pequeño en comparación con el coste del entrenamiento principal de las redes.

2.6. Conclusiones

En este capítulo, se ha mostrado el proceso de construcción de un modelo neuronal complejo dedicado a la predicción de la demanda de energía eléctrica. La serie temporal utilizada corresponde a valores aportados por Red Eléctrica de España S.A., correspondientes a la zona centro de España. El modelo utiliza dos paradigmas neuronales diferentes para la implementación del modelo: un paradigma auto-organizativo de Kohonen realiza, de forma no supervisada y sólo una vez en la vida del sistema, una clasificación de los perfiles de demanda que produce varias series temporales más fácilmente tratables que la original. En segundo lugar se utilizan redes recurrentes tipo Elman para la predicción de valores futuros de cada subserie temporal.

Entre las ventajas de este modelo de predicción se destacan las siguientes:

1. El usuario dispone de un perfil de 24 horas que se actualiza hora a hora. Esto le

permite llevar un control más exhaustivo para participar en las ofertas de generación del mercado diario y en cada uno de los 6 mercados intradiarios actuales. Puesto que ya se ha realizado la etapa de clasificación definitiva para esa zona de regulación y se tienen las arquitecturas de las redes neuronales utilizadas para la predicción, se puede ampliar el horizonte de predicción a las siguientes 24 horas sin demasiado esfuerzo en la fase de experimentación. Los resultados serían sensiblemente peores por el error que tendría la variable de entrada demanda integral.

2. El módulo implementado de entrenamiento periódico (*on-line*) activado a petición del usuario, hace que la influencia de la no-estacionariedad de la curva de la demanda sea menor que en el caso de realizar la predicción con paquetes estadísticos, menos flexibles y con un mayor tiempo de actualización en la búsqueda de la estimación de los parámetros de predicción.

3. La adaptabilidad a otras zonas de regulación se logra fácilmente debido al módulo de clasificación no supervisada, lo que reduce la intervención de un experto respecto a los paquetes estadísticos existentes.

4. Los resultados de predicción obtenidos hacen al módulo predictor competitivo frente a otros basados en métodos estadísticos y análisis de intervención previa actualmente en funcionamiento.

5. Ninguno de los dos predictores implementados depende de datos de entrada de consumo horario no consolidados, eliminándose así la influencia de errores acumulados.

6. Cuando existan condiciones meteorológicas, económicas o sociales bruscas que hagan que el perfil de consumo para un día determinado varíe en demasía del perfil normal de la clase, el módulo de predicción horario tendrá en cuenta estos cambios a través del consumo de las horas previas (datos de entrada), rectificando una predicción anormalmente errónea del predictor de perfil diario.

7. El módulo de predicción horaria proporciona la predicción de todas las horas del día y de todos los días, incluyendo días festivos entre semana y fechas atípicas. Esto permite utilizar la predicción de la hora $h+1$ como primera entrada del predictor de perfiles para las clases de periodicidad diaria (clases 12, 13 y 14).

2.6 Conclusiones

Entre las limitaciones del paquete de predicción neuronal, se destacan las siguientes:

1. La discontinuidad de las series obtenidas al realizar la clasificación de los patrones, hace que para predecir un día determinado se utilice un día que no esté próximo en el tiempo, con el consiguiente aumento del error de predicción. Esta discontinuidad es especialmente importante en clases que tienen un número pequeño de días.

2. En las clases de periodicidad diaria (clases de días laborales) es necesario esperar a la primera hora del día para realizar la predicción de 24 horas. De lo contrario, se obtendrían errores del 2 % aproximadamente frente al 1.73 % que se obtiene de estar las últimas horas del día ya consolidadas. En cualquier caso, el módulo de predicción horaria no tiene esta dependencia y el usuario está informado de la predicción de todas las horas.

3. El paquete de predicción neuronal toma como entrada la demanda integral del día cuyo perfil se va a predecir. Este dato no está consolidado y su predicción requiere el conocimiento de datos adicionales como temperaturas horarias en curso.

4. El predictor, en su versión actual, no resuelve de manera satisfactoria el efecto de la variabilidad de los días festivos sobre la demanda de energía de los días anterior y posterior.

La disponibilidad de un mayor número de datos históricos, tanto de consumo eléctrico como de variables exógenas para cada hora, puede soslayar en parte estas limitaciones. En particular, mejoraría significativamente el tratamiento de los días festivos. Además, la incorporación al modelo del conocimiento de los operadores, por ejemplo mediante técnicas de lógica borrosa, permitiría realizar una etapa correctora para estos días especiales, que mejoraría significativamente los resultados.

Capítulo 3

Estimación de estado

3.1. Introducción

La estimación de estado de un sistema eléctrico de potencia es el procedimiento por el cual se obtiene el valor más probable de sus variables de estado a partir de un conjunto redundante de medidas como flujos de potencia activa y reactiva, módulos de tensión, etc. [Wood and Wollenberg, 1994; Zarco Periñán and Gómez Espósito, 1999; Monticelli, 2000]. A partir de estas variables y de un modelo matemático del sistema, podrán ser calculadas las restantes magnitudes que lo definen, permitiendo el conocimiento de las condiciones de operación del mismo. Este proceso queda en principio justificado por su aportación a los objetivos siguientes:

- Mejorar el sistema de control, facilitando al operador una base de datos en tiempo real para la detección y localización de anomalías, diagnóstico de fallos, control de la carga y la frecuencia, etc.
- Facilitar la implementación de ciertas funciones de control como el despacho óptimo de las potencias activas y reactivas, minimizando los costes del sistema, etc.

Para conocer el estado del sistema se requiere disponer de información sobre el estado de interruptores y el valor de medidas de potencia y tensión de la red bajo supervisión. Si esta información fuese completa y exacta, el problema de la estimación de estado quedaría resuelto matemáticamente con toda exactitud. Desgraciadamente, mantener un sistema perfecto de adquisición de medidas resulta técnica y económicamente inviable, lo que obliga al

centro de control a trabajar con un conjunto de medidas aquejadas, principalmente, por los siguientes defectos:

- Ruido introducido por los instrumentos y canales de telemedida.
- Ausencia de instrumentos de medida en muchas partes del sistema.
- Asincronismo en la producción y recepción de medidas.
- Utilización de pseudomedidas erróneas.
- Errores en la información sobre el estado de interruptores y demás elementos topológicos de la red.

En estas circunstancias, dado que resulta imposible el cálculo exacto de las variables de estado, se hace necesario encontrar un método de estimación que las proporcione con el menor error posible. La estimación de estado obtendrá el estado de la red estadísticamente óptimo [Zarco Periñán and Gómez Espósito, 1999] a partir del conjunto de medidas disponibles. El resultado de la estimación de estado será el punto de partida de otras aplicaciones del centro de control.

Este capítulo aborda el problema de la aplicabilidad de las redes neuronales artificiales a la estimación de estado, y queda organizado de la siguiente forma: en la sección 3.2 se plantea el problema desde el punto de vista matemático y se enumeran algunos de los algoritmos disponibles para su realización. En la sección 3.3 se estudia la aplicabilidad de las redes de Hopfield, y se proponen los parámetros óptimos para este problema particular. Finalmente, en la sección 3.4, se presentan las conclusiones del capítulo.

3.2. La estimación de estado como problema matemático

Dado un sistema eléctrico de potencia con topología conocida de n nodos para el que se dispone de m medidas de flujos, inyecciones y módulos de tensión ($m > 2n-1$), la estimación de estado consiste en la resolución del sistema redundante de ecuaciones (3.2.1):

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + v \tag{3.2.1}$$

donde $\mathbf{z}$ representa el vector de medidas disponibles, $\mathbf{x}$ es el vector de estado verdadero desconocido (módulo y fase de las tensiones de los nodos de la red),$\mathbf{h}$ representa el vector de ecuaciones no lineales que relacionan las medidas con las variables de estado yυ es el vector de errores de las medidas. En una estimación clásica, las medidas del vector$\mathbf{z}$ están constituidas por flujos de potencia activa y reactiva (P_{ij}, Q_{ij}), inyecciones de potencia activa y reactiva (P_i, Q_i), y módulos de tensión de nudo (V_i); y el vector $\mathbf{x}$ corresponde a n módulos de tensión de nodo y $n-1$ fases $(\mathbf{V_i}, \theta_\mathbf{i})$, utilizándose una de las fases como fase de referencia.

Se supone que los errores $\upsilon_\mathbf{i}$ son variables aleatorias independientes que siguen una distribución gaussiana de media cero y varianza σ_i^2 conocida.

El método de estimación de estado más extendido es el denominado estimador de mínimos cuadrados ponderados o WLS (*Weighted Least Squares*). Éste tiene como objetivo minimizar la ecuación (3.2.2):

$$J(\mathbf{x}) = \sum_i \frac{(z_i^m - h_i(\mathbf{x}))^2}{\sigma_i^2} = [\mathbf{z} - \mathbf{h}(\mathbf{x})]^t W [\mathbf{z} - \mathbf{h}(\mathbf{x})] \tag{3.2.2}$$

donde W es la inversa de la matriz de covarianzas de errores de las medidas$R = E[\upsilon\upsilon^t] = diag(\sigma_1^2\sigma_2^2, ..., \sigma_m^2)$.

Para encontrar el mínimo de la función (3.2.2), debe satisfacerse que$\frac{\partial J(\mathbf{x})}{\partial \mathbf{x}} = 0$, o

$$G(\hat{\mathbf{x}}) = H(\hat{\mathbf{x}})^T W \left[\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}})\right] = 0 \tag{3.2.3}$$

donde $H(\mathbf{x}) = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}$ es la matriz jacobiano y $\hat{x}$ es el vector de estado estimado. La solución de esta ecuación se puede obtener por el método iterativo que resuelve en cada iteración la ecuación (3.2.4) ([Zarco Periñán and Gómez Espósito, 1999]):

$$A(\mathbf{x}^k)\Delta\mathbf{x}^k = -H^T(\mathbf{x}^k) W \left[\mathbf{z} - \mathbf{h}(\mathbf{x}^k)\right] \tag{3.2.4}$$

obteniéndose la corrección del vector de estado$\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}^k$, donde $A(\mathbf{x}^k)$ es una matriz no singular que depende del método utilizado.

El método de Newton utiliza $A(\mathbf{x}) = H^t(\mathbf{x})W[\mathbf{z} - \mathbf{h}(\mathbf{x})]$, con lo que el sistema a resolver viene dado por la ecuación (3.2.5):

$$G(\mathbf{x})\Delta\mathbf{x} = H^T(\mathbf{x}) W \left[\mathbf{z} - \mathbf{h}(\mathbf{x})\right] \tag{3.2.5}$$

donde $G(\mathbf{x})$ se denomina matriz de ganancia. Esta matriz es dispersa, definida positiva y simétrica. La ecuación (3.2.5) se resuelve generalmente mediante la factorización triangular de la matriz de ganancia $G = U^t U$, donde U es una matriz triangular superior.

Aunque éste método de estimación de estado es el más extendido, se han formulado diversas alternativas que buscan optimizar la estabilidad numérica y la eficiencia computacional del método WLS. Entre las alternativas más notables cabe destacar las que utilizan transformaciones ortogonales, el método de *Hachtel* [Gjelsvik et al., 1985] o el método de mínimos valores absolutos.

Estos estimadores parten de una información topológica que es considerada como verdadera en todo momento, produciendo una estimación que debe ser coherente con dicha topología. Si el estado de alguno de los elementos que definen la topología es erróneo, esto es, se considera el estado de un interruptor abierto cuando realmente esta cerrado o viceversa, el estimador puede dar una respuesta errónea, sin que sea detectable de forma automática la causa del error. El modelado explícito del estado de interruptores o seccionadores como variables de estado se realiza en los denominados estimadores de estado generalizados ([Monticelli, 2000], [de la Villa Jaén and Expósito, 2001]), donde una parte de la red se modela en el nivel de buses y ramas (estimación de estado normal) y otra parte, más reducida, se modela en el nivel físico o de nodos para realizar la estimación del estado de los elementos topológicos a partir del uso de medidas tomadas en ramas o dispositivos de conmutación con impedancia nula. La principal ventaja de este tipo de estimación de estado es que facilita el análisis de las medidas erróneas producidas por errores en la topología de la red, aunque su aplicación queda limitada por el aumento computacional que requiere la modelización de toda la red en el nivel de nodo, siendo sólo aplicable a un reducido número de buses en cada estimación de estado.

3.3. Estimación de estado con redes de Hopfield

Varios son los autores que han aplicado el paradigma de Hopfield [Hopfield, 1982; Hopfield and Tank, 1985] a problemas relacionados con la gestión de sistemas eléctricos de potencia. Así, en [Mori, 1991] se utiliza esta red para determinar la observabilidad topológica de un sistema de potencia. En [King et al., 1995] se aborda el problema del despacho

económico y medioambiental óptimo con redes de Hopfield. En [Zivanovic and Petroianu, 1994] se aplica el principio de desacoplamiento y se obtiene un modelo neuronal para la estimación de estado con dos redes de Hopfield funcionando en paralelo. En [Vinod Kumar et al., 1996] se aplica a la estimación de estado y a la determinación de la topología. En [Nakawaga et al., 1991] utiliza este paradigma para la estimación de estado de una red de seis nodos. Nuestro estudio toma como punto de partida este último trabajo, aplicando el paradigma propuesto a la estimación de las redes estándares de 14, 30, 57, 118 y 300 buses.

La red de Hopfield se caracteriza por tener una sola capa, ser realimentada y operar en tiempo continuo [Sympson, 1990]. El modelo continuo es una modificación de la red original que opera en tiempo discreto y sólo puede manejar datos binarios/bipolares. Esta transición abrió la aplicabilidad de la red de Hopfield a muchos problemas de optimización en el que las variables son continuas.

La figura 3.1 muestra el esquema de una red de Hopfield. Analíticamente, la evolución de la red, en el modelo de Abe [Abe, 1989], viene dada por las ecuaciones (3.3.1), (3.3.2) y (3.3.3):

$$\frac{du_i}{dt} = \sum_i T_{ij} V_i + I_i \tag{3.3.1}$$

$$V_i = f_i(U_i) \tag{3.3.2}$$

$$f_i(U_i) = \frac{1}{1 + e^{-\alpha U_i}} \tag{3.3.3}$$

donde

- T_{ij}: Peso de la conexión que va desde la neurona j a la i.
- I_i: Entrada (también denominada polarización o *bías*) de la neurona i.
- U_i: Potencial de la neurona i.
- f_i: Función de transferencia de la neurona i. Existen diversas posibilidades para la elección de esta función, pero la más común es la función sigmoide.
- α: Constante que define la pendiente de la función de transferencia f_i.
- V_i: Salida de la neurona i.

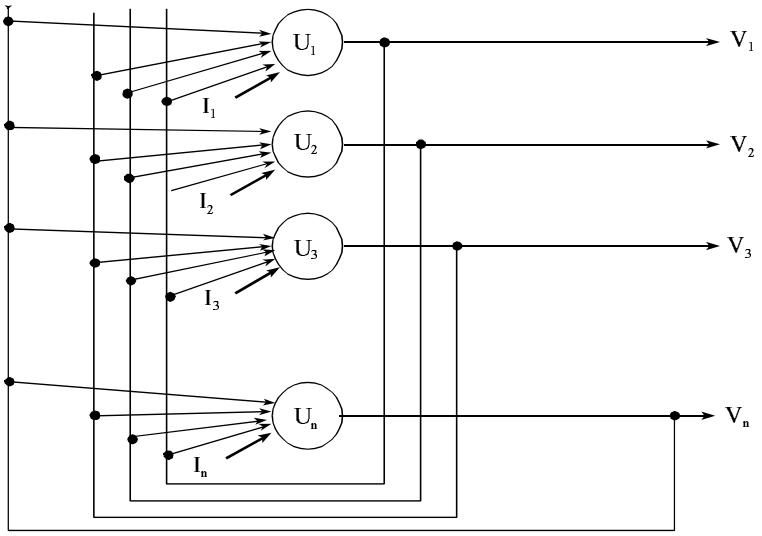


Figura 3.1: Representación de una red de Hopfield.

La función de transferencia dada por la ecuación (3.3.3) produce salidas en el intervalo $[0, 1]$. Cuando se requiere que las salidas de las neuronas estén en el intervalo $[l_i, h_i]$ puede utilizarse la expresión dada por la ecuación (3.3.4):

$$V_i = f_i(U_i) = \frac{h_i - l_i}{1 + e^{-\alpha U_i}} + I_i \tag{3.3.4}$$

Una de las características más importantes de este tipo de redes es la simetría de la matriz de pesos, es decir, $T_{ij} = T_{ji}$, y que los autopesos T_{ii} son siempre positivos. La red tiene una función de energía que viene dada por la ecuación (3.3.5) [Abe, 1989]:

$$E = -\frac{1}{2}\sum_i \sum_j T_{ij} V_i V_j - \sum_i I_i V_j \tag{3.3.5}$$

Esta función de energía decrece monótonamente cuando la red evoluciona a lo largo del tiempo. Esta característica es la clave en la aplicación de este paradigma a la resolución de problemas de optimización [Joya et al., 2002a]. Por tanto, para la aplicación de este modelo neuronal es necesario convertir la función que se desea optimizar en una función de energía de Hopfield, a partir de la cual se derivan los parámetros (pesos y entradas) de la red. Este es el objetivo de la siguiente sección para el problema particular de la estimación de estado.

3.3.1. Adaptación de los pesos de la red de Hopfield para la estimación de estado

La red de Hopfield evoluciona de tal manera que la función de energía que la define decrece en cada iteración hasta encontrar un mínimo local o global. En el problema de estimación de estado estamos interesados en minimizar el error residual de las mediciones. El objetivo es calcular los parámetros de una red de Hopfield para convertir el error residual de la estimación en una función de energía para dicha red.

Sea $h(x)$ el vector de ecuaciones no lineales que relaciona el vector de estado x, en coordenadas rectangulares, con las medidas z. La expansión de $h(x)$ en serie de Taylor alrededor del punto de operación x_e tiene la expresión exacta dada por la ecuación (3.3.6) ([Iwamoto and Tamura, 1978; Habiballah and Quintana, 1991]):

$$h(x) = h(x_e) + H(x_e)\Delta x + h(\Delta x) \tag{3.3.6}$$

donde $H(x_e) = \frac{\partial h(x)}{\partial x} \mid_{x=x_e}$ es la matriz jacobiano, $h(\Delta x) = h(x) \mid_{x=\Delta x}$ y $\Delta x = x - x_e$.

Reemplazando el valor de $h(x)$ dado por la ecuación (3.3.6) en la ecuación (3.2.2), se obtiene la ecuación (3.3.7):

$$J(x) = \frac{1}{2}\left[z - h(x_e) - H\Delta x - h(\Delta x)\right]^T R^{-1}\left[z - h(x_e) - H\Delta x - h(\Delta x)\right] \tag{3.3.7}$$

Definiendo ahora Δz según la ecuación (3.3.8):

$$\Delta z = z - h(x_e) - h(\Delta x) \tag{3.3.8}$$

podemos expresar (3.3.7) según la ecuación (3.3.9):

$$J(x) = \frac{1}{2}\left[\Delta z - H\Delta x\right]^T R^{-1}\left[\Delta z - H\Delta x\right] \tag{3.3.9}$$

Utilizando las igualdades $(A-B)^T = A^T - B^T$ y $(A*B)^T = B^T * A^T$, y resolviendo los productos, la ecuación (3.3.9) queda como la ecuación (3.3.10):

$$J(x) = \frac{1}{2}\left[\Delta z^T R^{-1}\Delta z - \Delta x^T H^T R^{-1}\Delta z - \Delta z^T R^{-1} H\Delta x + \Delta x^T H^T R^{-1} H\Delta x\right] \tag{3.3.10}$$

Definiendo ahora T, I y K según las ecuaciónes (3.3.11), (3.3.12) y (3.3.13) :

$$T = -H^T R^{-1} H \tag{3.3.11}$$

$$I = H^T R^{-1} \Delta z \tag{3.3.12}$$

$$K = \frac{1}{2} \Delta z^T R^{-1} \Delta z \tag{3.3.13}$$

la ecuación (3.3.10) se transforma en

$$J(x) = -\frac{1}{2} \sum_i \sum_j T_{ij} \Delta x_i \Delta x_j - \sum_i I_i \Delta x_i + K \tag{3.3.14}$$

Dado que añadir una constante a una función no modifica el proceso de minimización para dicha función, la ecuación anterior es una función de energía para una red de Hopfield si tomamos como pesos de la red la matriz T, como entradas (o bias) el vector I y como salidas de las neuronas el vector Δx:

$$E = -\frac{1}{2} \sum_i \sum_j T_{ij} V_i V_j - \sum_i I_i V_j + K \tag{3.3.15}$$

El proceso completo de estimación de estado utilizando la red de Hopfield se representa en el diagrama de flujo mostrado en la figura 3.2. El primer bloque, denominado Inicialización, consiste en determinar las islas observables de la red, el conjunto final de medidas disponibles y determinar un vector de estado inicial (solución inicial). En el segundo bloque, se calcula el jacobiano del sistema para la solución actual, así como la diferencia entre el vector de medidas y el vector de medidas estimadas. La matriz jacobiano y el vector diferencia se utilizan para determinar los pesos T y las entradas I de la red de Hopfield en el siguiente bloque. El siguiente paso consiste en ejecutar la red de Hopfield hasta llegar a un mínimo de su función de energía. Una vez terminada la ejecuación de la red, las salidas de la misma se toman como los incrementos de las variables de estado, que permite actualizar el vector de variables de estado estimadas x_e. El proceso se repite, para el nuevo vector de estado calculado, hasta que se alcance el objetivo de convergencia (p.e. que el incremento último de las variables de estado esté por debajo de un determinado umbral).

En las siguientes secciones se aborta el estudio del bloque principal del estimador de estado basado en redes de Hopfield, constituido por el bloque que realiza la ejecución de la red. El funcionamiento de este bloque determinará en última instancia la aplicabilidad y eficiencia del estimador completo.

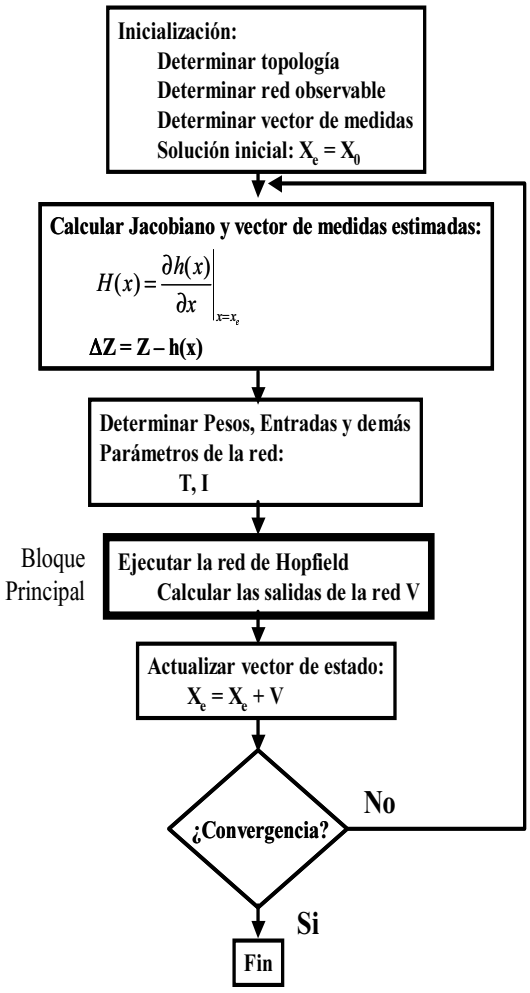


Figura 3.2: Diagrama de flujo con las tareas principales implicadas en la estimación de estado con una red de Hopfield.

3.3.2. Implementación del modelo neuronal de Hopfield

Debido a su naturaleza dinámica, la simulación por computador de una red de Hopfield requiere considerar varios factores que determinan la efectividad del modelo para abordar muestro problema. Concretamente, los factores a considerar son los siguientes: a) elección del método de activación de las neuronas; b) determinación de la pendiente de la función de transferencia; y c) resolución numérica de la ecuación (3.3.1). Estos factores suelen tener una alta dependencia con el problema particular que se trata de resolver, siendo generalemente imposible fijar unos valores genéricos adecuados para cualquier problema.

A continuación, se realiza el estudio de estos factores para el problema particular de la estimación de estado, y se propone un método de activación óptimo de neuronas, la elección de la pendiente de transferencia para las neuronas de la red y, finalmente, una ecuación para calcular automáticamente el paso de integración para aproximar numéricamente la ecuación (3.3.1) de forma automática y eficiente ([García-Lagos et al., 1998a], [García-Lagos et al., 2000a]).

3.3.2.1. Selección del método de activación de neuronas en la red de Hopfield

Para la activación de las neuronas se han considerado los siguientes métodos:

- Activación aleatoria. En cada iteración se elige una neurona al azar y se activa.
- Activación aleatoria modificada. En cada iteración se elige una neurona al azar y se activa, pero esta misma neurona no podrá volver a activarse hasta que las demás lo hayan sido por lo menos una vez. A este método de activación lo denominaremos activación aleatoria restringida.
- Activación por orden, o secuencial, en el que cada neurona se activa después de la que la precede y antes de la que la sucede, según una disposición inicial cualquiera.

Utilizando los tres métodos de activación anteriores, la red de Hopfield ha sido utilizada para realizar la estimación de estado de varias redes eléctricas estándar. La figura 3.3 muestra la evolución de la función de energía (ecuación (3.3.15)) para cada método para varias simulaciones con configuración de medidas típicas. Los mejores resultados se obtienen con el

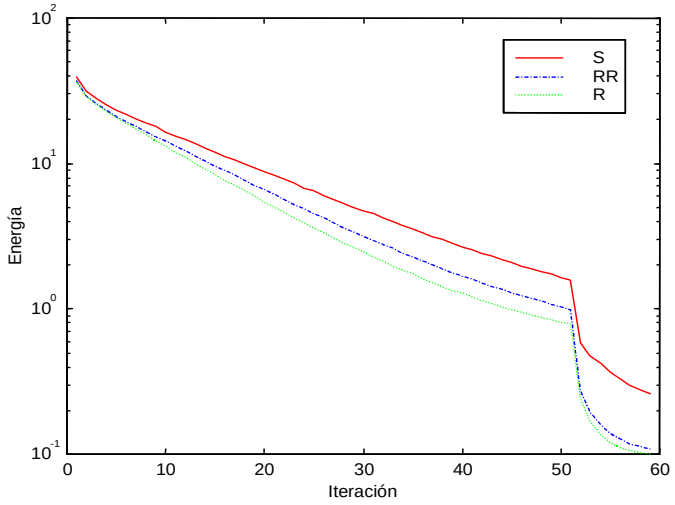


Figura 3.3: Evolución de la energía de una red de Hopfield aplicada a problemas de estimación de estado para diferentes métodos de activación de sus neuronas: activación secuencial (S), activación aleatoria restringida (RR) y activación aleatoria (R).

método de activación aleatorio, mientras que los peores resultados se producen con el método de activación secuencial. La elección final del método de activación de las neuronas no debe considerar únicamente los resultados obtenidos en la aplicación concreta, sino también la eficiencia computacional de cada caso. En la implementación que se realiza aquí, el método de activación aleatorio restringido necesita tiempo de proceso adicional para memorizar las neuronas que han sido seleccionadas y que no deben serlo de nuevo hasta que todas las neuronas de la red sean seleccionadas una vez. Los resultados de simulación muestran que los métodos de activación aleatorio y aleatorio restringido necesitan aproximadamente entre un 70 % y un 80 % del tiempo requerido por el método de activación secuencial para obtener una misma solución, respectivamente.

3.3.2.2. Selección de la pendiente de las funciones de transferencia

Normalmente, la función sigmoide (ecuación (3.3.4)) debe adaptarse para que las salidas de las neuronas tengan un rango apropiado al problema particular. En esta ecuación, h_i determina el valor máximo de salida de la neurona i, mientras que l_i determina el valor mínimo. Por otro lado, la constante α determina la pendiente de la función de transferencia,

lo que a su vez determina las características de convergencia de la red. Un valor de α muy grande hace que la función de transferencia sea muy abrupta, lo que puede provocar problemas de convergencia en la red. Por otro lado, un valor muy pequeño produce una convergencia muy lenta en la evolución de la red hacia el mínimo de su función de energía. El valor óptimo de este parámetro depende en gran medida de cada problema particular y, en general, no es posible utilizar un valor predeterminado. Por tanto, es necesario realizar una búsqueda, para cada red particular, de los valores óptimos de α, en el sentido de que su valor minimice la función de energía de la red, objetivo último de la dinámica de la red de Hopfield, empleando el menor tiempo posible. Así, dada una matriz de pesos T y unas entradas I, es posible obtener un valor óptimo para el parámetro α en el sentido de minimizar la función de energía, según la ecuación (3.3.16) ([Nakawaga et al., 1991]):

$$\frac{\partial E}{\partial \alpha} = -\frac{1}{2}\sum_i \sum_j T_{ij} m_{ij} - \sum_i I_i n_i = 0 \tag{3.3.16}$$

donde, para $l_i = -0{,}5$ y $h_i = 0{,}5$,

$$m_{ij} = (\frac{U_i a_i}{b_j^2})(\frac{1}{b_j - 0{,}5}) \tag{3.3.17}$$

$$n_i = \frac{U_i a_i}{b_i^2} \tag{3.3.18}$$

$$a_i = e^{-\alpha U_i} \tag{3.3.19}$$

$$b_i = 1 + e^{-\alpha U_i} \tag{3.3.20}$$

Aplicando el método de Newton a la minimización de la ecuación (3.3.16) para cada iteración de la red de Hopfield se obtiene el valor óptimo de α que debemos utilizar en la ejecución de la red neuronal. La figura 3.4 presenta valores típicos obtenidos por este procedimiento en diversas ejecuciones. Esta gráfica muestra que en las iteraciones iniciales el valor óptimo es pequeño (0,1 aproximadamente), valor que va aumentando hasta estabilizarse en 0,11 después de las 5 primeras iteraciones.

A partir de estos resultados hemos podido verificar heurísticamente que un valor de 0,11 es válido para garantizar, incluso en las primeras iteraciones, la convergencia de la red. Este estudio nos ha permitido prescindir del proceso de búsqueda de una pendiente óptima en cada iteración del algoritmo. Así, utilizando un valor $\alpha = 0{,}11$ queda garantizada la

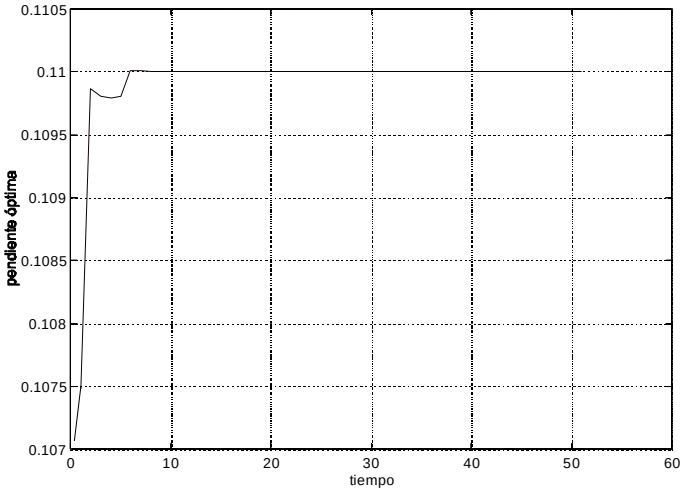


Figura 3.4: Valores óptimos obtenidos para el parámetro α en la estimación de estado.

convergencia en la ejecución de la red, y se elimina el computacionalmente costoso proceso de encontrar el valor óptimo.

3.3.2.3. Determinación del paso de integración

La simulación mediante ordenador de la dinámica de una red de Hopfield obliga a realizar una aproximación numérica de la ecuación (3.3.1). Un método eficiente y simple basado en la fórmula de integración de Euler viene dado por la ecuación (3.3.21):

$$\Delta U_i = (T * V + I^T)\Delta t \tag{3.3.21}$$

donde Δt es el parámetro que denominamos *paso de integración*. La elección del parámetro Δt condiciona extraordinariamente la precisión de la solución final obtenida por la red [Atencia et al., 2001], ya que la asociación de una función de energía al sistema es estrictamente cierta únicamente para una dinámica continua en el tiempo. De hecho, puede comprobarse experimentalmente que una red de Hopfield con un $\Delta t = 1$ puede perder su estabilidad, de manera que no alcance un estado estable, sino que describe un ciclo en el espacio de estado. Se comprueba que un Δt suficientemente pequeño garantiza la estabilidad del sistema, aunque cuanto más pequeño sea Δt más lenta será la dinámica y, por tanto, la convergencia al mínimo de la función de energía. Además, el valor óptimo de este parámetro puede ser diferente en cada iteración de la simulación de la red.

Una solución trivial para asignar valores a este parámetro consiste en comenzar con un valor grande e ir reduciéndolo hasta comprobar que la red converge, para lo cual hay que comprobar que la energía de la red decrece. Sin embargo, este método de prueba y error presenta el inconveniente de su alto coste computacional. Para evitar este problema, en este trabajo se desarrolla un método que permite el cálculo del valor correcto paraΔt de forma directa. El método se basa en la normalización de la matriz de covarianzas de los errores asociados a las medidas en un problema de estimación de estado. Usando las medidas y variables de estado en valores por unidad, y normalizando la matriz de covarianzas de los errores R, se ha podido comprobar que los valores de Δt que verifican la ecuación (3.3.22) son válidos para asegurar la convergencia de la ecuación (3.3.1).

$$\Delta t \leq \frac{C}{max\, T * V + I^T} \tag{3.3.22}$$

donde C es un valor constante que depende del intervalo en el que se normalize la matriz R. El parámetro C únicamente debe ser calculado en la primera iteración de la simulación de la red para un problema particular.

3.3.3. Utilización de matrices dispersas con redes de Hopfield

En las secciones anteriores se han derivado los parámetros que permiten utilizar una red de Hopfield para abordar el problema de la estimación de estado. En esta configuración, la red de Hopfield resuelve un sistema de ecuaciones en cada iteración para encontrar los incrementos de las variables de estado que minimizan los residuos. Debido a que los pesos de la red de Hopfield se obtienen a partir de la matriz Jacobiano del sistema (ecuación (3.3.11)), el número y estructura de los pesos de la red será similar a la matriz de ganancia de la estimación de estado clásica. Es bien conocido que esta matriz es dispersa (sólo un pequeño porcentaje de sus elementos tiene un valor no nulo), simétrica, y tiene una estructura que depende de la topología de la red eléctrica y de las medidas disponibles. Por tanto, la matriz de pesos de la red tendrá también una estructura similar. Los métodos de estimación clásica aprovechan la dispersidad de la matriz de ganancia para reducir los requisitos de memoria y de tiempo de cómputo necesarios para la resolución del sistema de ecuaciones utilizando técnicas de almacenamiento y aritmética para matrices dispersas. La figura 3.5 muestra la estructura de pesos para una estimación de estado particular de la red IEEE-14, donde puede apreciarse la estructura dispersa y simétrica de esta matriz.

3.3 Estimación de estado con redes de Hopfield

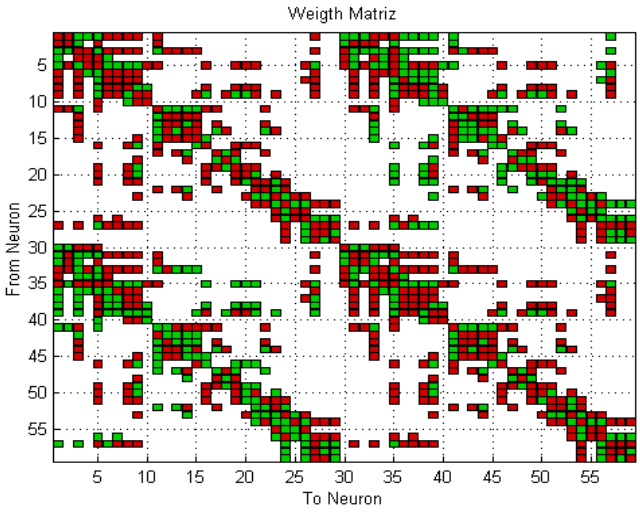


Figura 3.5: Representación de la dispersidad de la matriz de pesos para la red de Hopfield aplicada al problema de la estimación de estado. Los elementos diferentes de cero se representan con un rectángulo no blanco.

En el caso de la red de Hopfield, la dispersidad de la matriz de pesos implica que la mayoría de las conexiones tienen un peso de valor cero, es decir, en su conceptualización, sus conexiones asociadas no existen. Además, puesto que la matriz de pesos es simétrica, sólo es necesario almacenar la parte inferior o superior de la matriz, además de la diagonal. Con estas consideraciones, la implementación de la red que de Hopfield para estimación de estado que se propone en este trabajo se muestra en la figura 3.6.

Esta implementación requiere dos estructuras de datos. En la primera, se almacenan las entradas I_i (array I)), los valores internos U_i (array U) y las salidas V_i (array V) de cada neurona. Por otro lado, la segunda estructura se utiliza para almacenar los pesos no nulos de la red. Así, si el peso T_{ij} es no nulo, se crea una entrada donde se almacenan tres datos: el valor de T_{ij} en el array T, el índice de la neurona i en el array F, y el índice de la neurona j en el array C. Con esta implementación, el cálculo de las salidas de la red neuronal se puede realizar eficientemente según el pseudocódigo que se muestra a continuación:

```
Para i desde 1 hasta n
   U(i) = I(i);
Para i desde 1 hasta el número de pesos no nulos
   U( F(i) ) = U( F(i) ) + T( F(i) , C(i) ) * V( C(i) );
   Si i /= j
```

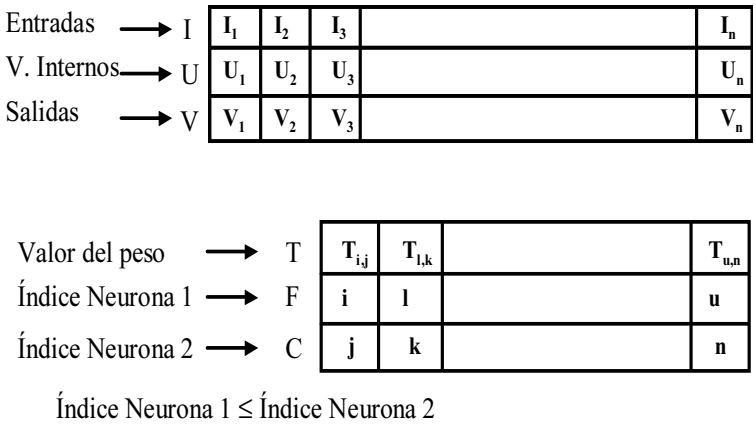


Figura 3.6: Estructuras de datos utilizadas en la implementación de la red de Hopfield aplicada a problemas de estimación de estado.

```
        U( C(i) ) = U( C(i) ) + T( F(i) , C(i) ) * V( F(i) );
Para i desde 1 hasta n
    V(i) = f( U(i) )
```

Con esta implementación se consiguen dos objetivos: ahorro considerable de memoria, al no almacenarse los pesos cuyo valor es cero, y reducción del tiempo de cómputo para el cálculo de la salida de la red, ya que no se consideran los pesos nulos (multiplicación por cero). Además, los pesos que se almacenan en las listas T, F y C (ver figura 3.6) no tienen por qué seguir un orden particular, por lo que la inserción de nuevos elementos es una operación computacionamente muy simple.

Los resultados que se obtienen utilizando la implementación dispersa, comparados con los resultados normales, muestran que se consigue una reducción significativa del tiempo de computo necesario para alcanzar un mínimo en la función de energía de la red, lo que significa a su vez que el tiempo requerido para obtener el vector de estado en una estimación se reduce considerablemente, como muestra la tabla 3.1. Para la red IEEE de 300 buses el tiempo de cómputo se reduce al 2 % utilizando la implementación propuesta.

Tabla 3.1: Ejemplos de tiempos de cómputo (en segundos) para la estimación de estado utilizando una implementación no dispersa y utilizando la versión propuesta en este trabajo.

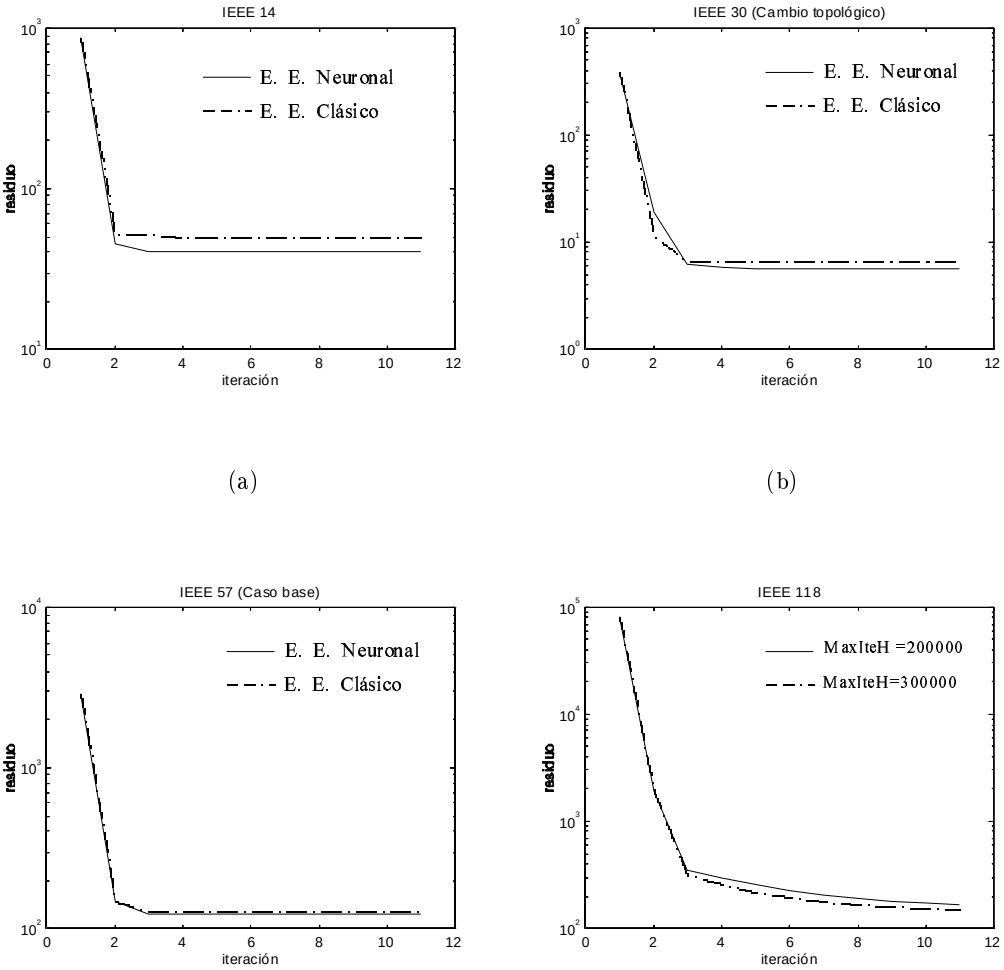
Red	Tiempo Implementación Normal	Tiempo Implementación Propuesta
IEEE 14	17	6
IEEE 30	137	22
IEEE 57	3602	64
IEEE 118	30780	324
IEEE 300	100800	2350

3.3.4. Resultados de la estimación de estado utilizando el modelo neuronal de Hopfield

En esta sección se presenta un resumen de los resultados de simulación obtenidos utilizando el modelo neuronal continuo de Hopfield para el problema de estimación de estado. El objetivo de la red será la obtención de un vector de estado para cada patrón de entrada disponible. Para la validación del modelo se han utilizado diversas redes eléctricas de potencia estándares, concretamente, las redes IEEE de 14, 30, 57, 118 y 300 buses. A continuación, se comparan los resultados obtenidos por la red de Hopfield con los obtenidos por un estimador clásico basado en mínimos cuadrados ponderados (ver sección 3.2), tanto desde el punto de vista de la calidad de la solución obtenida como del tiempo de cómputo requerido.

La tabla 3.2 resume los resultados obtenidos usando el estimador neuronal y clásico para diferentes redes y patrones de simulación. En lo que se refiere al error en la estimación (residuos finales), los valores finales obtenidos por el estimador neuronal son ligeramente menores y, por tanto, mejores, que los del estimador clásico. La figura 3.7 muestra la evolución de los residuos para ambos estimadores de estado para las redes IEEE de 14, 30, 57 y 118 buses. Por otra parte, en cuanto a los tiempos de convergencia, los resultados que se presentan en la tabla 3.2 muestran claramente que para la red IEEE-14 el estimador neuronal es1,5 veces más lento que el estimador clásico; para las redes IEEE-30, 57 y 118 el estimador neuronal es 4 veces más lento que el clásico y, finalmente, para la red IEEE-300 el estimador clásico es 2,8 veces más rápido que el neuronal.

Estos resultados muestran que la red de Hopfield no es una buena solución para tareas de estimación de estado en línea, ya que, aunque produce buenos resultados, sus prohibitivos



(a) (b)

(c) (d)

Figura 3.7: Evolución del residuo en cada iteración de los algoritmos de estimación neuronal y clásico. (a) Red IEEE-14. (b) Red IEEE-30. (c) Red IEEE-57. (d) Red IEEE-118.

Tabla 3.2: Residuos finales y tiempo requerido por los estimadores de estado neuronal y clásico.

Estimador	Neuronal		Clásico	
Red IEEE	Residuo J	Tiempo (seg)	Residuo J	Tiempo
14	40.03	6	49.83	4
30	5.72	22	6.44	5
57	123.7	64	127.22	16
118	163.3	324	204.1	82
300	1287.8	2350	1727	835

tiempos de cómputo lo hacen inaceptable. Por contra, en los últimos años se han mejorado notablemente las técnicas numéricas para la resolución del problema de estimación de estado, tanto desde el punto de vista de los tiempos de respuesta como de la estabilidad numérica. A partir de estos resultados podemos concluir que la utilización de redes neuronales (y en particular de la red de Hopfield) para un problema que está perfectamente resuelto con técnicas numéricas bien establecidas, robustas y eficientes, no aporta ninguna ventaja y debería, por tanto, ser descartada.

Con el objetivo de reducir los tiempos de cómputo requeridos por la red de Hopfield, se ha estudiado la posibilidad de la implementación hardware de este modelo neuronal, concretamente la implementación mediante FPGAs. En esta implementación, se diseña una red de Hopfield para resolver un problema de estimación de estado particular para una red y topología concreta. Los pesos de la red sólo cambian cuando los valores de las medidas disponibles cambian, mientras que las conexiones entre neuronas dependen del número de medidas disponibles, su disposición a lo largo de la red y la topología de la misma. El hecho de que el conjunto de medidas y su disposición cambie con frecuencia, así como el menos frecuente, pero posible cambio en la topología de la red, hacen que la implementación hardware de este modelo neuronal para resolver problemas de estimación de estado no aporte ventajas significativas con respecto a la solución software del modelo estudiado en este trabajo.

3.4. Conclusiones

En este capítulo se ha estudiado la aplicabilidad de la red de Hopfield al problema de estimación de estado de redes eléctricas de potencia. La red de Hopfield se utiliza para calcular, en cada iteración del estimador de estado, los incrementos de las variables de estado a partir de la matriz jacobiano, el vector de medidas disponibles y el vector de varianzas asociadas a cada medida. Estos incrementos se obtienen de las salidas de la red una vez que ésta ha alcanzado un punto estable, que por construcción es un mínimo de su función de energía.

Se han estudiado los parámetros básicos necesarios para simular la red de Hopfield. Así, se han estudiado los métodos de activación aleatoria, aleatoria restringida y secuencial. Los mejores resultados en tiempo de cómputo se han obtenido con el método de activación aleatoria, que consigue en promedio reducir el tiempo en un 30 % aproximadamente con respecto al método de activación secuencial. En lo que se refiere a la solución final obtenida, se ha mostrado que los tres métodos de activación conducen, finalmente, a la misma solución.

Por otro lado, se ha propuesto una expresión para encontrar directamente el valor del parámetro *paso de integración* que permite la simulación por computador del modelo continuo de Hopfield. La expresión es válida para cualquier red y configuración de medidas una vez que se ha normalizado la matriz de pesos de covarianzas de los errores de las medidas. Esto permite utilizar un valor de Δt lo suficientemente pequeño para asegurar la convergencia de la red a una solución y lo suficientemente grande para evitar una evolución lenta.

Para verificar el modelo se han utilizado cinco redes eléctricas estándares con diferentes configuraciones de medidas. Los resultados obtenidos por el estimador propuesto en este trabajo se han comparado con los obtenidos con un estimador de estado WLS. Los resultados muestran, con respecto a la calidad de la solución final obtenida, que el estimador de estado que se propone en este trabajo obtiene resultados finales ligeramente mejores que los obtenidos por el estimador clásico.

Debido a que las matrices a partir de las que se determinan los pesos de la red son altamente dispersas, la matriz de pesos de la red también lo es. Con el objetivo de reducir significativamente los requerimientos de memoria y los tiempos de cómputo implicados, se

ha diseñado una estructura de datos especial para la simulación de la red neuronal que sólo almacena los pesos no nulos de la red así como su utilización en la ejecución del modelo, lo que ha reducido considerablemente los tiempos de cómputo requeridos para que la red se estabilice en el mínimo de su función de energía. Sin embargo, aún con estas optimizaciones, los tiempos de cómputo requeridos para obtener estos resultados son mucho mayores que los requeridos por un estimador clásico.

Con estas características de la solución del problema de estimación de estado con redes de Hopfield, es necesario concluir que el paradigma de Hopfield no aporta ninguna ventaja frente a los métodos clásicos, bien establecidos, robustos y eficientemente implementados.

Capítulo 4

Estimación topológica

4.1. Introducción

El estimador de estado opera con medidas en tiempo real de dos tipos: medidas analógicas y digitales. El estado activo/no activo o cerrado/abierto de los mecanismos de interrupción de corriente tales como seccionadores, interruptores, derivaciones de transformadores, etc., determina la configuración de la red, que cambia cada vez que estos dispositivos actúan. Las medidas de tiempo real están disponibles cada pocos segundos, y proceden de las unidades remotas localizadas a lo largo de toda la red eléctrica. Periódicamente se comprueban las estaciones remotas para adquirir el conjunto completo de medidas disponibles que permitirán encontrar el estado de la red. Éste queda determinado si conocemos su *vector de estado*, que generalmente está compuesto por el módulo de tensión y la fase relativa de cada uno de los buses que componen la red. La función del estimador de estado es encontrar el vector de estado a partir de las medidas disponibles, partiendo de la configuración actual de la red (generalmente un modelo de red orientado a buses y líneas). Encontrar la configuración actual de la red es una función del denominado *procesador topológico*. Es, por tanto, una operación, previa a la estimación de estado en sí, que debe procesar los datos digitales para determinar la conectividad eléctrica de cada elemento de la red. Esta tarea presenta dificultades importantes debido a los siguientes hechos:

- Los estados de los elementos topológicos recibidos desde las remotas pueden ser erróneos.

- Muchos dispositivos de interrupción no están monitorizados (medidos), o bien su estado no es transmitido al sistema de adquisición.

- Debido a problemas en la comunicación de los datos, no todos los estados de los elementos de interrupción monitorizados están disponibles en un momento determinado.

- Para una red real de tamaño medio o grande existe una gran cantidad de medidas digitales que deben ser procesadas para determinar qué elementos están eléctricamente conectados, eliminando la posibilidad de que los estados erróneos o no conocidos sean corregidos por un operador.

Debido a estos problemas la configuración de la red no siempre puede determinarse con la información disponible sobre el estado de los elementos de la red, y se hace necesario realizar una *estimación topológica* que, haciendo uso de información adicional a la de los estados de los elementos de interrupción, proponga una configuración de la red como base para la realización de la estimación de estado. Si la configuración de la red, en adelante topología de la red, que utiliza el estimador de estado no es correcta, tampoco serán correctos los valores estimados de las variables de estado, invalidando el proceso de estimación de estado en su conjunto en el peor de los casos.

La detección de la causa de los errores se presenta difícil ya que el error en el estado de un interruptor puede producir una medida errónea en una variable físicamente alejada, de manera que la conexión entre ambos errores sólo podrá ser establecida por un operario con un profundo conocimiento del sistema. Los dos estimadores topológicos que se proponen en este trabajo producen una propuesta topológica a partir de los valores de las medidas de flujo e inyecciones, las cuales serán obtenidas, bien directamente de los aparatos de medida, bien indirectamente a través del proceso de estimación de estado. Finalmente, la topología propuesta será enfrentada a la suministrada por el sistema de adquisición y control (SCADA), resolviéndose las contradicciones entre ambas de acuerdo con un sistema de reglas preestablecido.

Este capítulo se estructura de la siguiente forma: en la sección 4.2 se resumen los métodos de estimación topológica más relevantes presentes en la bibliografía especialilzada; en la sección 4.3 se analiza el estimador topológico basado en perceptrones multicapa. En la

sección 4.4 se presenta un segundo estimador topológico que supone una mejora significativa al basado en perceptrones multicapa, basado en funciones radiales con distancia de Mahalanobis. Las conclusiones del capítulo se han dividido en dos secciones, una para el módulo de estimación topológica basado en MLP (sección 4.3.5) y otra el el módulo basado en funciones base radiales (sección 4.4.4).

4.2. Métodos de estimación topológica

En esta sección, se hace un breve estudio de los métodos analizados en la bibliografía en la tarea de estimación topológica. En primer lugar, se comentan los métodos clásicos de estimación, distinguiendo entre estimadores topológicos independientes del estimador de estado y los estimadores topológicos implementados como parte del estimador de estado. Por último, se analizan los métodos basados en redes neuronales.

Dentro del grupo de procesadores topológicos independientes del estimador de estado, el primer método utilizado explícitamente para identificar errores topológicos se describe en [Harris et al., 1976]. Está basado en la utilización de las relaciones lógicas entre el estado de los elementos interruptores y las medidas analógicas existentes, de forma que cuando en una línea hay una medida de flujo de potencia mayor que cero, la línea debe estar activa (en servicio). Para cada una de las líneas del sistema se aplican 3 reglas simples para validar el estado de la misma o proponer un nuevo estado (activo o no activo) compatible con el conjunto de medidas para esa línea. Aunque es un método muy simple y rápido, tiene una aplicabilidad limitada debido principalmente a que no considera el ruido inherente a las medidas de flujo disponibles. Además, no utiliza la información de las líneas del mismo bus para comprobar si la medida de flujo es errónea.

Otro método basado en el análisis de las relaciones entre las posiciones de los elementos topológicos y las medidas analógicas se fundamenta en la creación de ecuaciones del tipo $P_{ij} + b_{ji} * P_{ji} = 0$, donde P_{ij} es el flujo de potencia de la línea i-j, medido en el extremo i, P_{ji} es el flujo de potencia medido en el extremo j y b_{ji} es la posición del interruptor a la entrada del nodo j (0 cuando el interruptor está fuera de servicio y 1 cuando está en servicio). El método valida todas las ecuaciones posibles en el sistema para una configuración de medidas determinada. Cuando una de las ecuaciones no se cumple, se puede inferir que existe un error

topológico o un error en una medida de flujo. Los inconvenientes principales de este método son dos: no permite la detección de errores múltiples y es dependiente de las características y la carga actual del sistema. El método se discute en [Wilkosz, 1995].

En [Borkowska, 1981] se presenta un método que intenta imitar a un experto, cuyo conocimiento se utiliza para definir ciertas reglas o funciones denominadas*estrategias*, para cada elemento del sistema. Sus ventajas principales son que es un método simple y computacionalmente eficiente, y permite la detección de la mayoría de los errores más notables. Presenta dos grandes inconvenientes: se requiere definir, para cada elemento susceptible de ser estimado, una entrada en una tabla que defina la estrategia a seguir, en base al conocimiento de un experto, y, no todos los errores topológicos pueden ser detectados.

En [Bonanomi and Gramberg, 1983] se presenta un método de estimación basado en procedimientos de búsqueda, comprobando durante el proceso que se verifican las leyes de tensión y corriente de Kirchhoff. Aquellos elementos encontrados durante la búsqueda que no verifiquen las reglas anteriores son marcados como sospechosos. En particular, cuando se encuentra que la carga de un bus no está equilibrada, se asume que hay un error topológico. Aunque simple, este método presenta la limitación de que depende de la carga y características de cada bus de la red, y, por lo tanto, es poco portable.

Otra alternativa utilizada es la del empleo de índices*apropiadamente definidos* [Mansour and Eldin, 1987]. La idea principal es emplear índices que tengan una gran sensibilidad a errores que influyen negativamente en otras medidas. El procesamiento topológico consiste en la comprobación de los índices para descartar errores en cada una de las medidas disponibles, ya que éstos son indicativos de la coherencia entre las medidas y la estructura de la red. El método sólo permite detectar si una línea no ha sido incluida en el modelo de la red cuando realmente está en servicio.

El método presentado en [Singh and Glavitsch, 1991] utiliza un sistema basado en reglas. La reglas se definen en base a la información disponible en el sistema e intentan comprobar la coherencia entre las medidas analógicas y las digitales. El sistema define unas 130 reglas en total. A cada elemento topológico se le puede aplicar un subconjunto de estas reglas, dependiendo tanto del tipo de elemento en sí como de los elementos a los que está conectado. Las ventajas de este método son: no depende de la carga actual del sistema ni de su estructura; utiliza información local a cada elemento y no tiene problemas de

convergencia. Sin embargo, no es posible detectar errores topológicos en inyecciones con una medida de valor cero, y la localización de desacoplo de barras sólo puede hacerse si las medidas disponibles en el bus están bien balanceadas [Wilkosz, 1995].

Los métodos de estimación topológica dependientes del estimador de estado utilizan el vector residual de las medidas definido por la ecuación (4.2.1):

$$\mathbf{r} = \mathbf{z} - h(\hat{\mathbf{x}}) \tag{4.2.1}$$

donde $\mathbf{z}$ es el vector de medidas analógicas disponibles y $\hat{\mathbf{x}}$ es el vector de estado estimado. El vector de medidas verifica la ecuación (4.2.2):

$$\mathbf{z} = h(\mathbf{x}) + \mathbf{e} \tag{4.2.2}$$

donde $\mathbf{e}$ es un vector de errores, verificando $E(\mathbf{e}) = 0$ y $E(\mathbf{e}\mathbf{e}^T) = R$, siendo R la matriz diagonal de las varianzas de los errores de las medidas en $\mathbf{z}$. Estos métodos se basan en la relación entre los vectores $\mathbf{r}$ y $\mathbf{e}$ dada por por la ecuación (4.2.3):

$$\mathbf{r} = W\mathbf{e} \tag{4.2.3}$$

donde W es la matriz de sensibilidad residual. La ecuación (4.2.3) se obtiene a partir del estimador de estado WLS.

El método propuesto en [Le and Outhred, 1977] comienza analizando el residuo final $J(\hat{\mathbf{x}})$ obtenido por el estimador de estado. Cuando el valor de este índice sugiere la presencia de errores, comienza un proceso en el que se utilizarán los criterios definidos por las ecuaciones (4.2.4), (4.2.5) y (4.2.6) para poner las medidas sospechosas en un conjuntoθ.

Test r_N:

$$r_i/r_{N,i} > \gamma_{N,i} \tag{4.2.4}$$

Test r_W:

$$r_i/r_{W,i} > \gamma_i \tag{4.2.5}$$

Test $W - m$:

$$r_i/(\frac{r_i}{\sigma_i})^2 > J(\mathbf{x})/m \tag{4.2.6}$$

donde $r_{N,i}$ es el vector de residuos normalizados dado por la ecuación (4.2.7),$\gamma_{N,i}$ y γ_i son umbrales configurables, σ_i^2 es la varianza de la medida i-ésima del vector $\mathbf{z}$, y $r_{W,i}$ viene dado

por la ecuación (4.2.8).

$$r_{N,i} = \frac{\mid r_i \mid}{\sigma_i w_{ii}} \tag{4.2.7}$$

$$r_{W,i} = \frac{\mid r_i \mid}{\sigma_i} \tag{4.2.8}$$

Una vez que todas las medidas han pasado el test anterior, el objetivo es reducir el conjunto θ para detectar las medidas erróneas o los errores topológicos. La reducción comienza seleccionando aquellas medidas con mayor residuo para cada tipo particular de medida (p.e. flujo de potencia activa, módulo de tensión, etc) y pasando cada una de ellas una serie de test de selección de hasta tres niveles. Una vez que se han seleccionado las medidas erróneas se realiza una nueva estimación de estado ignorándolas. La principal desventaja de este método es que requiere del cálculo de las matrices W y los residuos correspondientes, lo que es computacionalmente intensivo.

En [Wu and Wen-Hsiung, 1989] también se utiliza el test r_N para la detección de errores topológicos. El fundamento de este método es que sin errores topológicos se verifica que $E(\mathbf{e}) = 0$, mientras que con errores topológicos no se verifica. Una vez que el error ha sido detectado mediante el test, la localización del mismo se basa en la utilización de la matriz de sensibilidad S, cuyo elemento (i,j) es la sensibilidad del residuo $r_{N,i}$ con respecto al cambio en el flujo de la rama f_j. En este trabajo, se verifica que el vector $E(r_N)$ es una combinación lineal de las columnas de la matriz S que corresponden a la línea que tiene un error topológico. Sin embargo, es necesario hacer notar que el conjunto de columnas de S necesarias para hacer la combinación lineal no es único, y que encontrar la combinación lineal en función de las columnas es una tarea computacionalmente intensiva y difícil. La principal ventaja del método es que permite la identificación de múltiples errores.

El método propuesto en [Lugtu et al., 1980] comienza con una lista de las medidas eliminadas de la estimación de estado actual por ser sospechosas de ser erróneas. Teniendo en cuenta esta lista y examinando el resultado de una estimación de estado nueva, junto con la posibilidad de incluir o excluir alguna línea sospechosa del modelo en una nueva estimación de estado, se puede concluir cuál es la línea que está siendo mal modelada. Así, en varias estimaciones de estado consecutivas en las que se van eliminado medidas y se van utilizando diferentes combinaciones de elementos sospechosos, se obtiene una estimación de estado sin medidas sospechosas. El tiempo requerido para realizar el proceso completo es

uno de sus principales inconvenientes, junto con el hecho de que la reducción del conjunto de medidas sólo se puede realizar mientras exista suficiente redundancia local.

En otros trabajos se utiliza la interpretación geométrica de los residuos causados por errores topológicos. Así, en [Clements and Davis, 1988], se proponen las condiciones suficientes para la detección, identificación y ausencia de errores topológicos. El proceso de detección e identificación de errores se realiza mediante el denominado test coseno. El inconveniente principal de este método está en que si están presentes tanto errores en las medidas como errores topológicos, el método no encontrará una solución válida. Por otro lado, la realización del test es computacionalmente costosa para redes con un gran número de buses y se requiere detectar múltiples errores.

El método propuesto en [Koglin and Neisius, 1990] se basa en el hecho de que cuando existe un error topológico, las inconsistencias en las medidas estimadas que encuentra el estimador de estado están localizadas cerca del error topológico. El método consiste en ir eliminando las medidas redundantes en la proximidad de un bus que se sospecha contiene un error topológico, sin eliminar otras medidas sospechosas que no estén próximas. Utilizando la lista de medidas eliminadas, se crea una tabla con todas las subestaciones y ramas sospechosas de contener un error topológico. Una vez construida la tabla, se seleccionan aquellas subestaciones que tienen inyecciones en el propio bus o en sus vecinos sospechosas de contener error o que contienen flujos saliendo de la subestación sospechosos. Sólo las subestaciones seleccionadas de esta forma se tienen en cuenta en la búsqueda de errores topológicos. La búsqueda consiste en la resolución de un problema de mínimos cuadrados para una red pequeña consistente únicamente en las subestaciones sospechosas y sus vecinos directos, tratando de realizar una estimación de estado cambiando el estado de un interruptor sospechoso de contener un error topológico. Si la función objetivo que se obtiene por este método es mejor que la calculada sin considerar el cambio topológico, se acepta el cambio y se da por localizado el error. La principal desventaja de éste método es que el número de variantes posibles crece exponencialmente con el número de líneas, y debe resolverse una estimación de estado para cada una de ellas. El método puede presentar problemas de convergencia.

En [Simoes Costa and Leao, 1993] se utilizan índices que permiten medir el grado de correlación entre medidas estimadas con síntomas de ser erróneas y la sensibilidad a los

errores que con mayor probabilidad pueden ocurrir bajo las condiciones de funcionamiento actual del sistema. El método utiliza la lista de medidas sospechosas de ser erróneas para seleccionar un conjunto de componentes de la red relacionados con esas medidas. Este conjunto se va reduciendo mediante el análisis de la matriz de sensibilidad, tomando aquellos elementos relacionados con magnitudes mayores en las columnas de esta matriz. Partiendo del conjunto reducido de elementos sospechosos, se repite el proceso (estimación de estado, detección de medidas y creación del conjunto de elementos sospechosos) tantas veces como sea necesario, habiendo cambiado el estado del elemento que se supone contiene el error topológico. Este algoritmo permite distinguir entre errores topológicos y errores en las medidas, aunque a cambio de un coste computacional alto, lo que constituye su desventaja principal.

Los métodos más recientes tratan de modelar el estado de cada interruptor explícitamente en el proceso de estimación de estado ([Monticelli, 1993], [Slutsker and Mokhtari, 1995], [Monticelli, 2000] y [de la Villa Jaén and Expósito, 2001]). Partiendo de la base de que la impedancia de un interruptor tiene dos valores posibles: cero cuando está cerrado e infinito cuando está abierto, estos métodos modelan las relaciones entre los flujos de potencia que circulan por él y la tensión en sus terminales, permitiendo utilizar los flujos por estos elementos como variables de estado. Se basan, por tanto, en incluir tanto los flujos de potencia activa como reactiva como nuevas variables de estado a estimar. Los valores estimados para estos flujos permiten determinar el estado abierto o cerrado de los correspondientes elementos topológicos. Cada elemento topológico se modela de forma diferente, según su estado inicial se suponga en servicio, fuera de servicio o desconocido ([Slutsker and Mokhtari, 1995]). Estos métodos presentan dos características principales: el modelado explícito de los elementos topológicos requiere modelar una parte de la red a nivel de nodos y equipos físicos, en vez de utilizar el modelo más simple de buses y ramas; la inclusión de un elemento topológico en el proceso de estimación de estado supone la inclusión de cuatro nuevas variables de estado en el sistema (dos variables para la tensión en un nuevo bus y dos variables de flujo) y cuatro pseudomedidas. La principal desventaja del método radica en que para mantener el tamaño del problema manejable sólo algunos elementos topológicos deberían ser incluidos en el proceso de estimación de estado. En [Singh and Alvarado, 1995] se utiliza esta misma idea, pero particularizada para la estimación de estado por el método

LAV (mínimos valores absolutos, del inglés *Least Absolute Value*).

El modelado explícito de los elementos topológicos también se utiliza en [Mili et al., 1999]. En este trabajo se desarrolla un método de preprocesamiento topológico basado en el análisis de todos los flujos de potencia activa y reactiva obtenidos después de una estimación de estado basada en el método de Huber ([Huber, 1981]). La ventaja principal de este método frente a los discutidos en el párrafo anterior es que no sufre de posibles problemas de convergencia. Por otro lado, su principal inconveniente es el tiempo de cómputo adicional requerido para su implementación.

Un tercer tipo de estimadores topológicos se basan en la utilización de Redes Neuronales Artificiales. El primer intento se describe en [Vinod Kumar et al., 1996], donde se utilizan dos redes neuronales para realizar de forma independiente una estimación de estado y una estimación topológica. Para el primer problema se utiliza una red del tipo *Functional Link Network* (FLN) ([Yoh-Han, 1989]), mientras que para el procesamiento topológico los mejores resultados se obtienen utilizando una red del tipo *Counterpropagation* ([Hecht-Nielsen, 1987]). Las entradas a la red son los flujos de potencia activa de cada línea así como el estado recibido de cada elemento topológico de la red. La salida de la red es una combinación de -1 y 1 que representa la configuración topológica estimada para toda la red. La principal limitación de este trabajo es que sólo permite discernir entre 16 topologías posibles, donde de una a otra sólo cambia el estado de una única línea, lo que limita su aplicabilidad a sistemas reales.

Otro método basado en redes neuronales se presenta en [Souza et al., 1996], que posteriormente es mejorado en [Souza et al., 1998]. El método se basa en la utilización de las innovaciones normalizadas como entradas a una red neuronal del tipo GMDH (*Group Method of Data Handling*) ([Farlow, 1984]). Las innovaciones normalizadas son índices que se calculan en función de las medidas recibidas y las medidas estimadas mediante un proceso de predicción de series temporales, que puede perder fiabilidad en situaciones donde se produzcan cambios bruscos en la carga del sistema. Por otro lado, la capacidad de este método para distinguir entre dos errores topológicos se muestra únicamente para medidas no erróneas y en situaciones donde una rama de la red está realmente abierta, pero es considerada por el estimador cerrada. Estos efectos negativos son pequeños comparados con otros tipos de errores. Por otra parte, este método puede producir resultados erróneos cuando

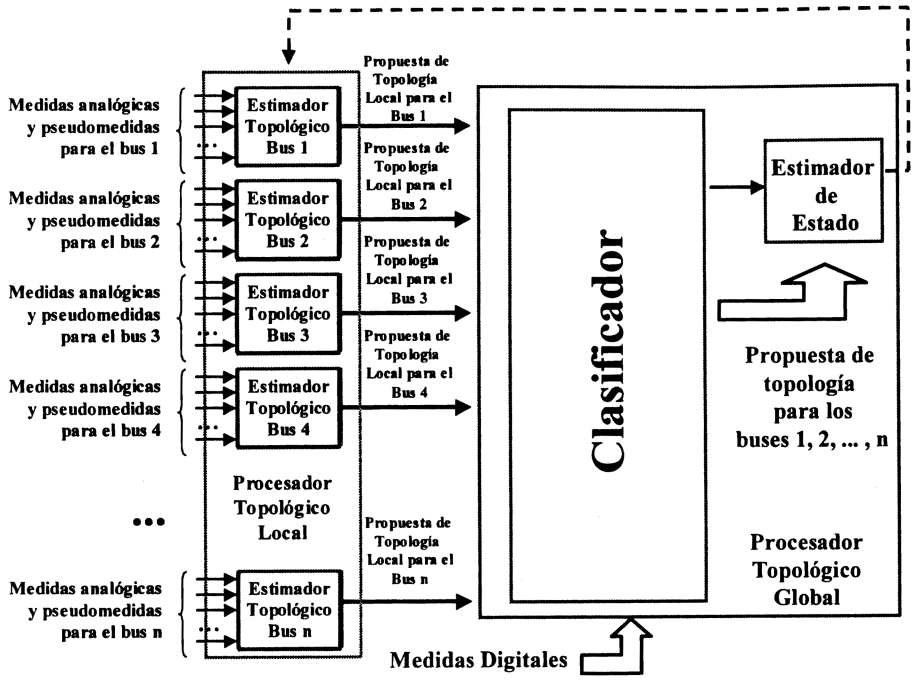


Figura 4.1: Estructura general del estimador topológico.

ocurren varios cambios topológicos simultáneamente en la red, especialmente cuando sólo uno de ellos es comunicado a SCADA.

4.3. Estimación topológica con Perceptrones multicapa

En esta sección describimos el primero de los dos estimadores topológicos desarrollados, basado en RNAs tipo Perceptrón multicapa [García-Lagos et al., 1998b]. Este estimador se basa en la utilización de un modelo del sistema orientado a buses y ramas, y presenta dos procesos claramente diferenciados. En una primera etapa, se estima de manera independiente y paralela la topología de cada bus del sistema, esto es, el estado de los interruptores asociados a las líneas que confluyen en cada bus, en las inmediaciones de ese bus (Estimación de Topología Local). En una segunda etapa, las topologías estimadas para cada bus son confrontadas entre sí y con la aportada por el módulo SCADA, realizándose la propuesta de topología definitiva de acuerdo con un conjunto de reglas (Estimación de Topología Global). El esquema general del sistema propuesto se ha representado en la figura 4.1.

4.3.1. Estimación de topología local

En esta sección discutimos los fundamentos del módulo dedicado a la estimación topológica local de cada bus, que será denominado estimador topológico local. Esta fase está constituida por dos etapas secuenciadas: preprocesamiento y clasificación, y en ella se realiza la estimación topológica de cada bus independientemente y de forma paralela.

El objetivo de este módulo es establecer una asociación entre el conjunto de medidas que pueden asociarse a un bus y la topología del mismo. Esta asociación es difícil de establecer debido a que para una misma topología pueden existir un número infinito de combinaciones de valores de las medidas. Para conseguir esta asociación nos basamos en el principio de conservación de la energía que establece que *"... en una red eléctrica alimentada por fuentes independientes, todas de la misma frecuencia, la suma de la potencia compleja suministrada por las fuentes independientes es igual a la suma de la potencia compleja recibida por todas las demás líneas de transmisión de la red eléctrica"*.[Bergen, 1986].

Utilizaremos el bus de la figura 4.2 como modelo en el análisis del estimador. Este bus tiene dos líneas (línea 1 y línea 2), un generador y una carga. P_g representa la medida de potencia activa que el generador inyecta en la red a través del bus. P_l es la medida de potencia activa asociada a la carga (inyección de potencia que sale de la red a través del bus). P_1 y P_2 son los flujos de potencia activa de las líneas 1 y 2. I_g, I_l, I_1 e I_2 representan interruptores que pueden estar en servicio o cerrados (conectan eléctricamente el elemento al bus) o fuera de servicio o abiertos (desconectan eléctricamente el elemento del bus). Suponiendo que todos los interruptores de la figura 4.2 están cerrados y que las medidas no presentan errores, si sumamos todas las posibles combinaciones de flujos de potencia activa del bus, en condiciones normales de funcionamiento, sólo una de estas sumas sería exactamente cero, debido al principio de conservación de la energía. En este caso sólo la primera combinación de sumas es cero, es decir, $P_g + P_l + P_1 + P_2 = 0$. En general, si aplicamos la función δ a cada combinación, se obtiene un vector de 14 componentes de ceros y unos que, de forma única, se puede asociar a la topología actual de la red. El vector$v$ de

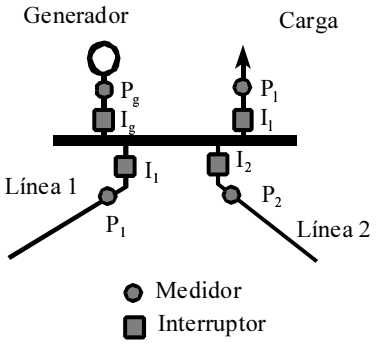


Figura 4.2: Representación de un bus de una red, con un generador, una carga y dos líneas.

unos y ceros para el bus bajo análisis viene dado por la ecuación (4.3.1).

$$v = \begin{pmatrix} \delta(P_g + P_l + P_1 + P_2) \\ \delta(P_g + P_l + P_1) \\ \delta(P_g + P_l + P_2) \\ \delta(P_g + P_1 + P_2) \\ \delta(P_l + P_1 + P_2) \\ \delta(P_g + P_l) \\ \delta(P_g + P_1) \\ \delta(P_g + P_2) \\ \delta(P_l + P_1) \\ \delta(P_1 + P_2) \\ \delta(P_g) \\ \delta(P_l) \\ \delta(P_1) \\ \delta(P_2) \end{pmatrix} \tag{4.3.1}$$

Si las medidas no presentan errores, sería directo implementar un sistema que asigne a cada vector obtenido de esta forma una topología compatible para el bus. El problema es que en un sistema real las medidas contienen errores, que harían que ninguna de las sumas fueran cero, es decir, la función δ no es utilizable en la práctica. Para soslayar este hándicap, teniendo en cuenta que cada medida tienen un error asociado, la funciónδ será reemplazada por una función de transferencia radial como se describe a continuación. La construcción del

vector utilizando funciones radiales constituye la etapa de preprocesamiento del estimador topológico. Esta etapa produce agrupaciones (*cluster*) de vectores, en el sentido de que todas las entradas asociadas a un misma topología producirán vectores en el espacio $[0,1]^n$, no exactamente iguales, pero muy próximos entre sí. La salida de esta etapa será utilizada por la etapa de clasificación, constituida por una red neuronal, que identificará cada *cluster*, es decir, asociará cada vector con la topología actual.

4.3.1.1. Etapa de preprocesamiento

Como se ha señalado arriba, la aplicación de la función impulso no es efectiva en la construcción de un vector que pueda ser utilizado para determinar de forma directa la topología de un bus debido a los errores presentes en las medidas. Esto ha sido resuelto utilizando funciones radiales. La función de transferencia radial viene dada por la ecuación (4.3.2):

$$y(x) = e^{\frac{-x^2}{\sigma^2}} \tag{4.3.2}$$

Esta función se caracteriza por tener una salida y máxima ($y = 1$) cuando x es cero y tiende a cero cuando x se aleja de este punto. Es decir, la función tiene forma de campana de Gauss. El parámetro σ determina el ancho de la campana y lo denominaremos en adelante parámetro de normalización. En la figura 4.3 se representa la función de la ecuación (4.3.2) para diferentes valores del parámetro de normalización. La etapa de preproceso estará constituida por tantas neuronas radiales como combinaciones de flujos de potencia activa y reactiva existan para un bus (figura 4.4). Cada neurona recibe la suma de una posible combinación de flujos y produce un valor en el intervalo $[0,1]$. La salida de la etapa de preproceso será, por tanto, un vector cuyos componentes pertenecen al intervalo $[0,1]$ y constituyen *una respuesta localizada ante los estímulos de entrada*, es decir, la salida de las neuronas es significativamente mayor que cero sólo cuando la entrada esté situada en una porción cercana a cero. El efecto de los errores en las mediciones se atenúa gracias al ajuste del ancho de la campana (parámetro de normalización). Cuanto mayores son los errores, el saldo de sumas se aleja de cero, pero si hacemos mayor el ancho de campana, su salida puede seguir estando próxima a uno. Por tanto, el ancho de la campana es proporcional a la estimación del error que tienen asociadas las medidas que entran en la neurona. Cuando la neurona tiene una única entrada, hemos encontrado que el valor del parámetro de normalización óptimo es dos veces la desviación

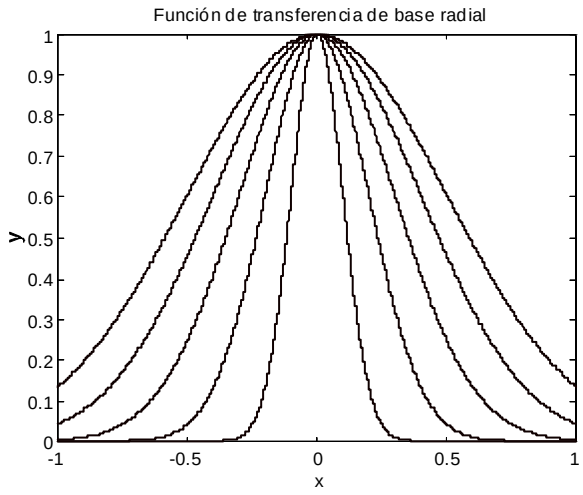


Figura 4.3: Función de transferencia de base radial para diferentes valores deσ.

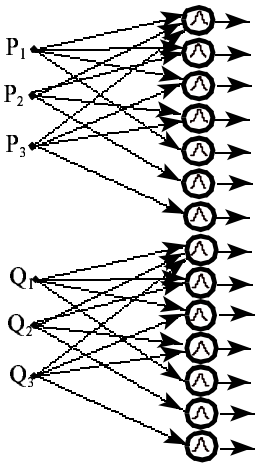


Figura 4.4: Representación esquemática de la etapa de preprocesamiento de base radial para un patrón con 3 medidas de potencia activa y 3 de reactiva.

típica del error asociada a la medida. Cuando la neurona recibe una combinación de n entradas, el parámetro de normalización se calcula de acuerdo a la ecuación (4.3.3).

$$p_i = 2\sqrt{\sum_j \sigma_j^2} + 1 \tag{4.3.3}$$

La salida de cada neurona vendrá dada por la ecuación (4.3.4):

$$y_i = e^{-(\sum_{j \in C_i} p_j)^2 / p_i^2} \tag{4.3.4}$$

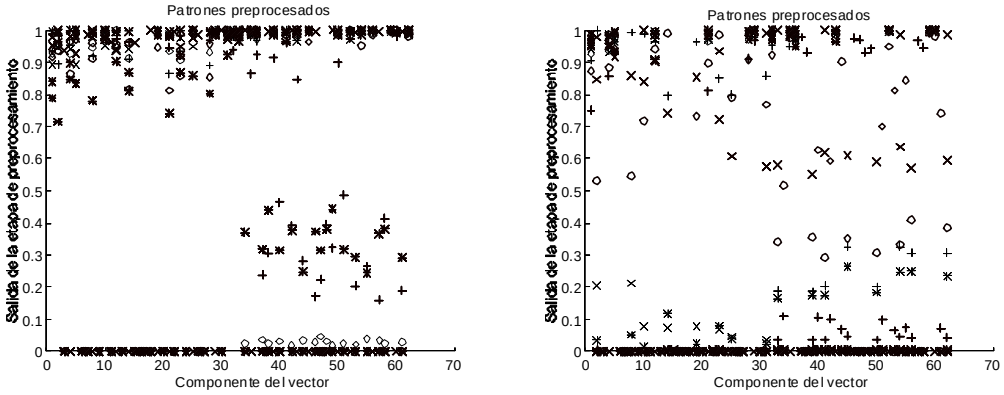
donde y_i es la salida de la neurona i, asociada con el subconjunto i-ésimo de flujos, C_i representa el conjunto de índices de ramas incluidas en el subconjunto$i$, y p_i es el parámetro de normalización.

Como puede comprobarse experimentalmente, patrones pertenecientes a topologías diferentes producen patrones muy diferenciados en algunas componentes del vector preprocesado (salida de la etapa de preproceso), y patrones muy diferentes para una misma topología producen vectores preprocesados cuyos componentes son similares. Así, las figuras 4.5(a) y 4.5(b) representan las salidas preprocesadas para 100 patrones de flujos de dos topologías diferentes de la red estándar IEEE-14. Comparando ambas, puede observarse que la salida de la etapa de preprocesamiento produce patrones característicos diferentes para cada topología.

Una vez disponemos de los vectores preprocesados y teniendo en cuenta que sus componentes tienen características que permiten identificar la topología a la que pertenecen, en una segunda etapa se utilizará un clasificador neuronal para asociar los vectores preprocesados con la topología del bus. Este es el objetivo de la etapa de clasificación-codificación que se analiza a continuación.

4.3.1.2. Etapa de clasificación

Las redes neuronales del tipo Perceptrón Multicapa con algoritmo de aprendizaje*Back-propagation* [Rumelhart et al., 1986] resultan especialmente aptas para la agrupación en clases de un conjunto de n-uplas basándose en la distancia entre los puntos representados por éstas en un espacio n-dimensional. Consecuentemente, será una red de este tipo la que constituya el módulo de clasificación de topologías locales. Estas redes recibirán como entrada la salida de la etapa de preprocesamiento, y deberán producir a su salida la codificación



(a) (b)

Figura 4.5: Representación de las salidas de la etapa de preprocesamiento para patrones típicos de dos topologías diferentes.

del estado de cada una de las líneas que une el bus en cuestión con el resto de la red. La figura 4.6 muestra el esquema de la etapa de clasificación. Puesto que las neuronas de salida tienen una función de activación tipo tangente hiperbólica, valores próximos a +1 serán interpretados como línea cerrada, valores próximos a -1 como línea abierta y valores intermedios como indecisión (figura 4.7). Los umbrales que determinan la pertenencia de un valor de salida a alguno de los tres tramos anteriores son obtenidos de manera heurística como un compromiso entre el índice de fiabilidad de la red al proponer un estado de línea y la probabilidad de que este valor sea una indecisión.

Entrenamiento de la red neuronal

Cada bus de la red eléctrica dispone de una red neuronal local, que de forma independiente estima la topología para las líneas pertenecientes al bus. La red neuronal empleada es un Perceptron Multicapa con una capa de entrada, una oculta y una de salida. La estructura del estimador topológico neuronal para cada bus se muestra en la figura 4.6. Para el proceso de entrenamiento supervisado, la salida de la red neuronal se entrenará para que produzca una codificación única para el estado de cada línea del bus. Una salida +1 indicará que la línea está en servicio y una salida de -1 indicará que la línea está fuera de servicio.

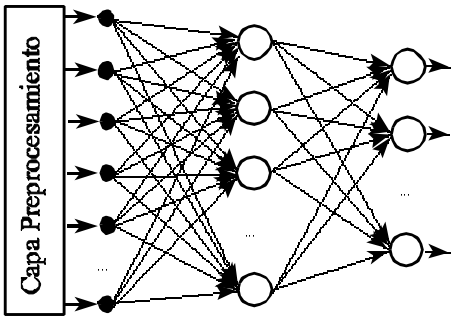


Figura 4.6: Arquitectura del estimador topológico neuronal local.

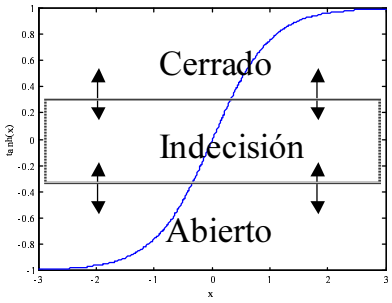


Figura 4.7: Márgenes asignados a las salidas de la red neuronal para determinar el estado abierto, cerrado o de indecisión.

La generación de los patrones de entrenamiento de la red neuronal sigue el siguiente proceso:

1. Se fija una de las topologías posibles de la red, así como una configuración aleatoria de cargas y generaciones para cada bus a partir de un caso base.

2. Mediante la solución del problema de flujo de carga asociado, se obtiene el valor verdadero de cada medida.

3. Se añade a la medida verdadera un error de media cero y varianza prefijada para cada medida.

4. Se selecciona la información de flujos de potencia activa y reactiva por cada línea que parte del bus considerado.

En la generación de patrones, de todas las topologías posibles del bus, no se considera aquella en la que todas las líneas están abiertas, es decir, la topología que aísla el bus del resto de la red. Para cada topología considerada, se han generado 100 patrones diferentes. Así, si tenemos un bus con 4 líneas, se han considerado 15 topologías y, por tanto, se han generado un total de 1500 patrones. Para realizar el entrenamiento supervisado de la red neuronal se han construido pares de patrones entrada - salida deseada (*target*), siendo la entrada el vector de patrones de flujo de potencia y la salida deseada la topología real del bus.

Las pruebas se han realizado para la red IEEE-14. Como algoritmo de entrenamiento se ha elegido el denominado *Backpropagation* con ganancia adaptativa.

Cada bus de la red tiene su propia red neuronal, que es entrenada de manera independiente. La convergencia del algoritmo de aprendizaje hasta un error global $SSE = 0{,}01$ se produce, en todos los casos, en menos de 10000 épocas. El número de unidades ocultas para cada red varía desde 20 hasta 60 según el número de entradas (que a su vez depende del número de líneas que tenga el bus). La evolución del error cuadrático y el valor de la tasa de aprendizaje durante el entrenamiento de una red neuronal típica se ha representado en la figura 4.8. En ella se puede observar cómo, para la última parte del entrenamiento, ha sido necesario utilizar una tasa de aprendizaje variable en cada iteración.

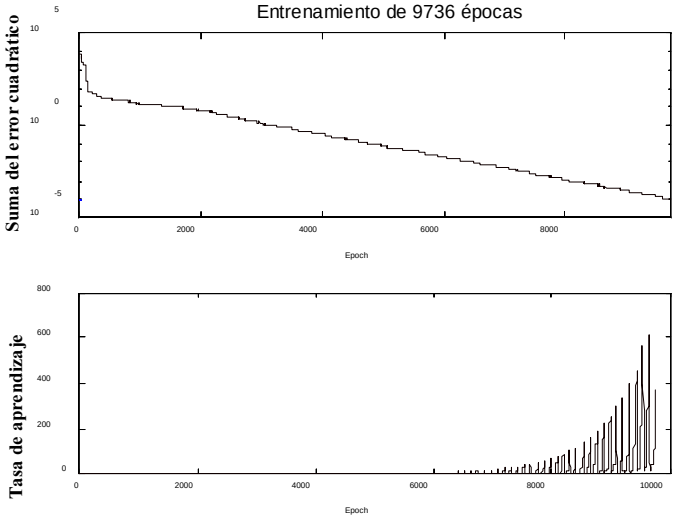


Figura 4.8: Evolución de la suma del error cuadrático y de la tasa de aprendizaje durante el entrenamiento de la red neuronal.

En resumen, la salida de la etapa de estimación topológica local está constituida por las salidas en el rango $[-1, 1]$ de sus neuronas. Si todas las salidas tomaran exactamente los valores -1 (abierto) ó 1 (cerrado), el vector de salida representaría un código que indica la topología a la que pertenece el vector de medidas de entrada, puesto que el proceso de aprendizaje de la red se ha realizado con este objetivo. Sin embargo, en la mayoría de los casos las salidas no serán -1 ó 1, sino valores próximos a éstos. En estos casos, cuando la salida está próxima a -1 ó 1, ésta se toma respectivamente como abierto o cerrado. Sin embargo, es posible que algunas salidas estén más próximas a 0 que de -1 ó 1, en cuyo caso decidir si representa a un -1 o un 1 no es posible. Para solventar esta limitación se ha implementado una segunda fase del estimador topológico, el módulo de estimación topológica global.

4.3.2. Estimación de topología global

La función principal de la etapa de estimación topológica global es confrontar las estimaciones obtenidas para cada uno de los buses de la red por el estimador topológico local. Así, este módulo es el encargado de comparar, para cada línea de la red, la coherencia de la estimación local en cada bus de sus extremos, procediendo de la siguiente forma cuando las estimaciones obtenidas en cada bus es contradictoria:

- Si una estimación es indecisión y la otra no, ignorar la indecisión.
- Si ambas estimaciones son indecisiones o ninguna lo es pero son estimaciones contradictorias, utilizar el estado proporcionado originalmente por SCADA.

4.3.3. Proceso de Estimación Topológica

Como se ha mostrado en las secciones anteriores, el proceso de estimación de estado está compuesto por dos fases: la fase de estimación local (compuesta a su vez por dos etapas, la etapa de preproceso y la etapa de clasificación) y la fase de estimación global. La primera fase recibe un conjunto de combinaciones de flujos de potencia y produce una propuesta topológica para cada línea de la red (línea en servicio o fuera de servicio) o una indeterminación. La segunda fase permite unificar estimaciones locales, así como utilizar la información de estado de interruptores y seccionadores procedentes de SCADA. Sin embargo, la aplicación del estimador de topología local correspondiente a un determinado bus requiere conocer las medidas de flujo de todas las líneas que confluyen en dicho bus. Puesto que esta exigencia está muy lejos de poder cumplirse en todos los buses de una red eléctrica real, debe establecerse un proceso que proporcione un conjunto de medidas apropiadas que sustituyan a aquellas no suministradas por el sistema de adquisición de datos siempre que sea posible. En este sentido, es posible reemplazar medidas de flujo ausentes por pseudomedidas del otro extremo de la línea, ponderadas de forma adecuada. Además, el proceso de estimación topológica puede repetirse después de realizar una estimación de estado en aquellos buses que no disponían de todas las medidas, utilizando las medidas estimadas. El proceso completo de estimación topológica se resume a continuación:

```
Estimador Topológico basado en MLP:

1.  Para cada bus de la red eléctrica, recopilar las medidas
    de inyección y flujo disponibles desde SCADA.

2.  Para cada bus que no disponga del conjunto de medidas
    completo, generar todas las posibles pseudomedidas.

3.  Estimador Topológico Local.
    Para cada bus con el conjunto de medidas (o pseudomedidas )
    completo, aplicar el Estimador Topológico Local
```

```
    3.1.  Presentar el vector de medidas a la etapa de preproceso,
          para obtener el vector preprocesado.
    3.2.  Presentar el vector preprocesado a la etapa de
          clasificación.
    3.3.  Las salidas no indeterminadas de la etapa de clasificación
          constituyen una propuesta de topología para el bus.
          Algunas salidas de esta etapa pueden ser
          indeterminaciones, en cuyo caso la topología del bus
          aún no puede ser estimada.

4.  Estimador Topológico Global.
    Para cada bus que ha generado una salida indeterminada o
    contradictoria en el paso 3.3, aplicar las reglas del estimador
    topológico global para estimar la topología del bus.

5.  Para las topologías de los buses que no han sido estimadas,
    utilizar la información topológica recibida desde SCADA.

6.  Realizar una estimación de estado.

7.  Para los buses que en el paso 2 no se pudo completar el
    conjunto de medidas, completarlo utilizando las medidas
    estimadas en el paso 6.

8.  Repetir el proceso con la nueva configuración de medidas.
```

4.3.4. Resultados de la estimación de topología

En esta sección, nos centramos en el bus 2 de la red IEEE-14. Este bus tiene 4 líneas, un generador y una carga, lo que lo hace suficientemente genérico (las dos líneas que unen los buses 1 y 2 se han modelado como una línea equivalente). Para poner a prueba el funcionamiento de las redes neuronales, se ha generado un nuevo conjunto de patrones correspondientes a todas las topologías posibles para este bus. Estos patrones son diferentes a los utilizados en la fase de entrenamiento de la red neuronal.

4.3.4.1. Prueba del modelo de estimación topológica local

El objetivo es obtener la topología del bus 2 cuando se presentan a la red las entradas de los patrones generados. La red neuronal, idealmente, responde poniendo cada una de

sus salidas a -1 o 1 según estime que la línea está abierta o cerrada, respectivamente. Sin embargo, no todas las salidas serán valores bipolares -1 o 1, sino que es posible cualquier valor en el intervalo [-1,1]. Esto nos lleva a la siguiente definición:

El ***intervalo de incertidumbre*** representa los valores de salida de las neuronas que no serán interpretados como estado de línea abierta o cerrada, sino que indican que la red neuronal simplemente no es capaz de estimar el estado de dicha línea.

Cuando la salida de la red neuronal, para una línea, está dentro del intervalo de incertidumbre, asumimos que la red neuronal no es capaz de decidir sobre el estado de la línea, es decir, ha producido una indecisión. Estas indecisiones serán utilizadas en una segunda etapa del proceso que, con información adicional sobre buses vecinos e información digital sobre el estado de los elementos topológicos, podrá en muchos casos resolver la indecisión.

Para la validación del modelo se han realizado dos bancos de pruebas claramente diferenciados. En una primera batería de pruebas se realizan estimaciones topológicas suponiendo que todas las medidas de un bus están disponibles. En el segundo caso, se supondrá que una medida de inyección o flujo para un bus no está disponible. Esta medida ausente será sustituida por una pseudomedida.

Para la primera prueba, donde todos los flujos de potencia e inyecciones para el bus están disponibles, el estimador topológico ha sido capaz de clasificar correctamente todos los patrones excepto uno, es decir, ha encontrado la topología correcta que los generó. El intervalo de incertidumbre se ha fijado en $[-0{,}4, 0{,}4]$. El patrón no clasificado ha producido que las salidas de la red neuronal estén en el intervalo de incertidumbre. Las salidas de la red neuronal fueron, $y = [-1{,}0000, 0{,}9298, -0{,}9343, -0{,}5589]^T$. El último componente del vector y es el que está dentro del intervalo de incertidumbre.

La causa por la que este patrón no es correctamente clasificado por la red neuronal está en lo atípico del mismo. Concretamente, representa una topología en la que hay una línea cerrada por la que fluye una cantidad muy pequeña de potencia. Esto hace que la red neuronal no pueda decidir si la línea está abierta o cerrada. En general, todos los patrones pertenecientes a esta topología presentan unas salidas de la etapa de preprocesamiento atípicas. Esto se puede comprobar comparando la figura 4.5(a), correspondiente a una clase topológica típica, con la figura 4.5(b), que representa a los patrones de entrenamiento

pertenecientes a la misma clase que el patrón mal clasificado.

Como segundo banco de pruebas se generan patrones y se simula la ausencia de una medida de flujo o de inyección. Esta situación es más realista que la correspondiente a la primera prueba, ya que en un sistema real es poco probable que todas las medidas estén disponibles para un bus. El flujo o inyección no disponible será sustituido por una pseudomedida que reemplazará al dato ausente. Concretamente, para cada patrón p del conjunto de patrones de la prueba 1, se generan 5 nuevos patrones de la siguiente forma:

1. El propio vector p, que denominaremos p_1.

2. Se supone que el vector p tiene una de sus medidas muy errónea. Así, el nuevo vector p_2 tiene todas sus componentes idénticas a p excepto el dato correspondiente a un flujo, que se reemplaza por su valor más un ruido proporcional a su desviación típica (esto es, se vuelve a añadir un error, puesto que p ya es un vector medido). Con esto simulamos la elección de una pseudomedida buena, esto es, una pseudomedida próxima a la medida real.

3. Estamos en el mismo caso que el anterior, pero el flujo que se reemplaza será mucho más ruidoso que el anterior, concretamente un ruido de distribución normal con desviación típica 10σ, donde σ es la desviación típica del error original del componente a reemplazar. Con esto, generamos un patrónp_3 que simula a p con una pseudomedida no próxima a la medida real.

4. En este caso, seleccionamos un flujo y suponemos que éste no se conoce (no tenemos un aparato de medida para la línea en el extremo del bus 2 o bien éste está fuera de servicio). Este dato se sustituirá por una pseudomedida, concretamente por el flujo del medidor del otro extremo de la línea, con signo cambiado. Así,p_4 tiene, en uno de sus componentes, el flujo medido en el extremo opuesto de la línea.

5. Para este caso tenemos la misma situación que en 4, pero supondremos que el flujo que utilizamos como sustituto es muy ruidoso, es decir,p_5 utiliza como pseudomedida el flujo medido en el otro extremo, pero, añadiéndole un ruido adicional de valor10σ, donde σ es la desviación típica del error del medidor.

Con estos nuevos patrones podremos estudiar las soluciones que aporta el estimador topológico ante la situación que cada uno simula. En la tabla 4.1 se resumen los resultados de esta prueba. Cada fila corresponde a un patrón del conjunto de prueba, que llamaremos patrón base, y cada columna p_1 a p_5 indica el vector generado a partir del patrón base. La columna fallos indica el número total de vectores que no fueron correctamente clasificados. En cada casilla aparece un '0' cuando la estimación topológica es correcta para el patrón considerado, un '1' cuando es incorrecta y una 'x' cuando la red neuronal ha generado una indecisión. Si exceptuamos el caso del patrón base 19, en todos los demás el error se produjo por indecisión, es decir, no se obtuvo una topología correcta ni incorrecta. El caso del patrón base 19 presenta los mismos problemas que se comentaron anteriormente. El número de patrones no clasificados es de 9 frente a un total de 150, lo que supone un 6 % de patrones no clasificados correctamente o que produjeron una indecisión. Sin embargo, el porcentaje de patrones mal clasificados se reduce a la mitad (3 %) si no tenemos en cuenta las indecisiones.

En las pruebas anteriores no se ha considerado el caso más desfavorable, que se presenta cuando no se dispone de medida de flujo de potencia en ningún extremo de una línea que está en servicio, y los flujos se reemplazan por pseudomedidas de valor0. Esto lleva al estimador topológico a considerar que la línea está abierta cuando en realidad está cerrada.

4.3.4.2. Prueba del estimador topológico completo

Para verificar el funcionamiento del el estimador topológico se han generado dos pruebas diferentes. En la primera, se supone que todos los flujos de potencia activa y reactiva están disponibles (estas medidas pueden proceder de SCADA o ser pseudomedidas). Esta prueba nos permitirá evaluar el sistema sin la necesidad de utilizar los resultados de un estimador de estado. En la segunda prueba se asume que no todos los flujos están disponibles, y que hay presentes varios errores topológicos en el modelo de red obtenido desde SCADA. En este caso, es necesario realizar una estimación de estado para reemplazar los flujos no conocidos. De nuevo y sin pérdida de generalidad, utilizaremos como caso de estudio el bus número 2 de la red IEEE-14. La figura 4.9 muestra este bus y sus vecinos. La notación que empleamos es la siguiente: $L_{i,j}$ expresa el estado del interruptor de la línea i-j en el extremo i; $Z_{i,j}$ es el par de medidas de flujo de activa y reactiva de la línea i-j, medidas en el extremo i. Finalmente,

Tabla 4.1: Resultados de la estimación topológica local para la red neuronal del bus 2.

P. Base	fallos	p_1	p_2	p_3	p_4	p_5
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	1	0	0	0	0	x
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	5	1	1	1	1	x
20	0	0	0	0	0	0
21	1	0	0	0	0	x
22	2	0	0	0	x	x
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
30	0	0	0	0	0	0

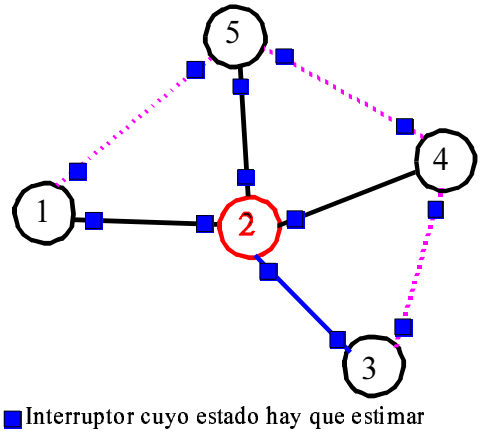


Figura 4.9: Representación esquemática del bus 2 y sus vecinos en la red IEEE-14..

si a un interruptor se le asigna el valor 0 se indica que está cerrado y si se le asigna el valor 1 se indica que está abierto.

En la tabla 4.2 se muestra un resumen de los resultados de estas dos pruebas. En la mayoría de los casos la topología encontrada por el sistema es correcta, de forma que la tabla sólo muestra aquellos casos en los que el resultado ha sido erróneo o bien casos típicos del bus bajo estudio. En cada caso se realizan las mismas pruebas para diferentes patrones de carga de forma que los resultados son generales para todos ellos. En la primera columna se muestra un número que referencia a cada caso; en la segunda columna se muestra la topología verdadera de la red en la simulación (el vector mostrado corresponde a la configuración de línea $[L_{1,2}, L_{2,3}, L_{2,4}, L_{2,5}]$); en la tercera se muestra la topología que utiliza el estimador de estado para encontrar el conjunto de medidas que faltan; la columna cuatro muestra el conjunto de medidas que no están disponibles ni para el estimador topológico ni para el estimador de estado. Finalmente, en la columna de resultados se indica, para cada caso, el resultado de la estimación topológica de todas las líneas del bus 2. La indicación `correcta` (C) significa que todas las líneas han sido estimadas correctamente; la indicación `indecisión 2-x` o `error 2-x` (E) significa que la estimación topológica realizada ha fallado únicamente en el estado de la línea 2-x. Para estos dos últimos resultados se indica la estimación realizada para la misma línea en el bus x. A continuación, se comentan aquellas simulaciones que producen un resultado problemático en la estimación topológica.

Tabla 4.2: Resultados representativos de la interacción del estimador de estado y del estimador de topología. *E.* significa Erróneo y *C.* significa Correcto.

N	T. Real	T. Entrada E.E	Medidas que faltan	Resultado
1	0000	0000	Z23	Indecisión 2-3 - C. 3-2
2	0000	0000	Z21	C.
3	0000	0000	Z21, Z23 y Z24	Indecisión 2-3 - C. 3-2
4	1000	1000	Z23	E. 2-4- C. 4-2
5	1000	0000	Z23, Z24	E. 2-4- C. 4-2
6	1000	0101	Z23, Z21	E. 2-4 - C. 4-2
7	1000	1010	Z24, Z23	E. 2-4 - C. 4-2
8	0010	0010	Z23	C.
9	0010	0010	Z21, Z23, Z24	C.
10	0010	0000	Z21, Z23	C.
11	0010	0000	Z23, Z24	C.
12	0010	1000	Z24	C.
13	0010	1000	Z21	E. 2-1 - C. 1-2
14	0010	1000	Z24	C.
15	0010	1001	Z23, Z24	C.
16	0100	0100	Z23, Z24	C.
17	0100	0100	Z21, Z23, Z24, Z25	C.
18	0100	0000	Z23	C.
19	0001	0000	Z23, Z25	C.
20	1010	1010	Z21, Z23, Z24, Z25	C.
21	1010	0010	Z21, Z24	C.
22	1010	0010	Z21, Z23, Z25	C.
23	1010	0000	Z23, Z25	C.
24	1010	0000	Z21	E. 2-1 - C. 2-1
25	1010	0000	Z24	E. 2-4 - C. 4-2
26	1010	0000	Z24, Z42	E. 2-4 - E. 4-2
27	0110	0000	Ninguna	C.
28	0110	0000	Z23	E. 2-3 - C. 3-2
29	0110	0000	Z24	E. 2-4 - C. 4-2
30	0110	0000	Z21, Z25	C.
31	0110	0000	Z21, Z25 + (*)	C.
32	1110	1110	Z21, Z23, Z24, Z25	C.
33	1110	0110	Z23, Z24	C.
34	1110	0110	Z21	E. 2-1 - C. 1- 2
35	1110	0110	Z24	E. 2-4 - C. 4-2
36	1110	0100	Z23	C.
37	1110	0100	Z21	E. 2-1 - C. 1-2
38	1110	0100	Z23, Error topológico en 3-4	C.
39	1110	0100	Z23, Error topológico en 3-4, 5-6 y 13-14	C. .
40	0111	0111	Z21, Z23, Z24, Z25	C.
41	0111	0110	Z21	C.
42	0111	0110	Z25	E. 2-5 - C. 5-2
43	0111	0110	Z23, Z25	E. 2-5 - C. 5-2
44	0111	0100	Ninguna	E. 2-4 - C. 4-2
45	0111	0100	Z25	E. 2-4 - C. 4-2E. 2-5 - C. 5-2

La indecisión del caso 1 se debe a que el flujo que realmente circula por la línea 2-3 en la configuración de la red utilizada es muy bajo, de menos de 1 MW de activa, lo que no permite al estimador topológico decidir sobre el estado del interruptor. En el caso número 3 se tiene el mismo problema.

En los casos del 4 al 7 se tiene que el flujo que se mide en la línea 2-4 es muy pequeño (-0.8 MW y -0.34 MVAr), lo que determina que en la etapa de preprocesamiento el vector generado sea uno típico de topologías con la línea 2-4 abierta.

En el caso 13 el estimador de estado parte de una topología con un error en la línea 2-1, para la cual tampoco hay medida disponible (Z21), con lo que el estimador de estado parte de que la línea está erróneamente abierta. Consecuentemente, estima un flujo de activa y reactiva de valor 0, lo que fuerza al estimador topológico a cometer un error en la estimación de esa línea.

En los casos 24 y 25 se presenta el problema de que el estimador de estado parte de dos errores topológicos para el bus 2. Como una de las dos líneas con error tiene una medición disponible con, posiblemente, un valor pequeño o cero, el estimador de estado asigna una gran parte del saldo de flujo del bus hacia la línea que no dispone de medida. Este tipo de error no puede ser corregido debido a que el vector de estado que se encuentra se adapta para ser coherente con los errores topológicos. En el caso 26 este efecto se acentúa al encontrar el estimador local del bus 4 la línea como cerrada debido, de nuevo, a un flujo muy grande asignado por el estimador de estado. Los casos 28-29 presentan los mismos problemas, pero para una topología diferente.

Con el caso 31 se intenta simular la situación en la que el estimador de estado parte de una topología con dos errores en el bus 2 y, además, se añade otro error en una línea de un vecino (1, 3, 4 o 5) pero que no pertenece al bus 2. En este caso todos los errores son corregidos por el estimador topológico. En el caso 32 se añaden otros dos errores en buses de líneas no próximas al bus 2, concretamente en las líneas 5-6 y 12-13 de la red. Los resultados muestran que los errores topológicos son corregidos en todos los casos (con la configuración de medidas señalada).

Para el caso 34 se tiene el mismo problema discutido anteriormente, esto es, el error topologíco no puede ser corregido si el estimador no dispone de la medida de una línea que

está abierta y el estimador de estado la considera como cerrada y, además, hay otro error topológico en otra línea del mismo bus.

En los casos 35 a 45 se presentan situaciones similares en las que también aparecen errores topológicos por las mismas causas discutidas anteriormente y otros en los que se han añadido errores topológicos a líneas no adyacentes al bus 2. En esta última situación, se comprueba que el estimador local del bus 2 no se ve afectado por tales errores.

En resumen, podemos afirmar que el estimador topológico de un bus falla únicamente en las tres situaciones siguientes:

1. Cuando una línea que está cerrada tiene una medida de flujo muy próxima a cero y desviación típica grande; en este caso el estimador de topología clasifica la línea como abierta.

2. Cuando se parte de un estado de línea erróneo (se considera abierta cuando realmente está cerrada) y no se dispone de medida o pseudomedida para la línea, el estimador de estado genera una medida de flujo cero que es clasificada por el estimador topológico como característica de línea abierta.

3. Cuando se parte de un estado de línea erróneo (se considera cerrada cuando realmente está abierta) y no se tiene medida para ésta misma línea y hay otros errores topológicos en líneas que pertenecen al mismo bus. En este caso, el estimador de estado asigna un flujo generalmente de valor muy alto que es interpretado como característico de un estado de línea cerrada por el estimador topológico.

Por lo tanto, la aplicabilidad de esta herramienta para una red eléctrica dada pasará por un estudio detallado de las características de cada bus con respecto a la probabilidad de que se presenten para éste las causas de fallos 1, 2 y 3. Estos fallos podrían evitarse configurando el sistema de medidas de forma que se reduzca al máximo el número de buses en los que se pueden dar las condiciones anteriores; o asignando un índice de fiabilidad a cada estimador local en función de la probabilidad que tenga el bus asociado de presentar las circunstancias problemáticas 1, 2 y 3. Cuando se produzcan contradicciones en la estimación del estado de una línea, se dará mayor importancia a la estimación realizada por el bus con una mayor fiabilidad.

4.3.5. Conclusiones sobre los resultados del estimador topológico basado en MLP

.

En esta sección se ha presentado un estimador de topología neuronal basado en un Perceptrón Multicapa, con una etapa de preprocesamiento de neuronas de base radial. Este estimador está formado por un conjunto de redes neuronales, una para cada bus del sistema, que actúan paralelamente en una primera fase. Una vez realizada la etapa de estimación local, el estimador global, basado en reglas muy simples, se encarga de compatibilizar las estimaciones locales para construir la topología de una red eléctrica orientada a buses y ramas.

En las pruebas realizadas con los estimadores topológicos locales para cada bus, los resultados señalan que, salvo para patrones muy atípicos, el porcentaje de errores cuando se dispone de todo el conjunto de medidas es 0. Los patrones atípicos son aquellos en los que líneas cerradas presentan medidas de flujo muy próximas a cero y que tienen una estimación de su error muy alta, esto es, medidas con una desviación típica grande. Este tipo de vectores no puede ser clasificado por la red neuronal ya que presentan características similares a patrones con una topología diferente. Por otro lado, el residuo que genera el estimador de estado cuando parte de un error topológico en estas líneas es pequeño, es decir, desde el punto de vista de la estimación de estado este tipo de errores no invalida la estimación realizada.

Cuando el conjunto de mediciones no es completo pero se dispone de pseudomedidas, incluso muy ruidosas, el porcentaje de aciertos es muy alto, produciendo sólo errores en el caso de utilizar pseudomedidas de flujo 0 en líneas que realmente están cerradas.

Cuando el conjunto de medidas y pseudomedidas para un bus no es completo, es necesario realizar una estimación de estado previa para encontrarlas. En estos casos, cuando el estimador de estado parte de una topología correcta, los resultados muestran un 100 % de éxitos en la estimación topológica final cuando las indecisiones se corrigen con la estimación realizada en el otro extremo de la línea, lo que significa que el estimador topológico diseñado no rectifica topologías que realmente son correctas.

También, se han visto situaciones en las que el estimador topológico no es capaz de

corregir una topología errónea que ha sido usada para generar el conjunto de mediciones incompleto para un bus. En estos casos, el estimador de estado genera un conjunto de estimaciones que se adaptan a la topología errónea que el estimador topológico no es capaz de filtrar.

Se ha mostrado que los estimadores topológicos locales no se ven afectados por errores topológicos en líneas no adyacentes al propio bus, lo que permite realmente la utilización paralela de cada estimador.

Finalmente, señalamos que el estimador topológico diseñado puede completarse con un análisis de residuos para detectar medidas muy erróneas producidas por algún error topológico no detectado. En definitiva, se trata de aprovechar el hecho de que los errores topológicos no detectados pueden generar residuos muy grandes para mediciones y estimaciones de líneas adyacentes.

4.4. Asistente topológico basado en funciones base radiales con distancia de Mahalanobis

En esta sección proponemos otro método de estimación topológica que mejora significativamente los resultados aportados por el estimador basado en Perceptrones Multicapa. El asistente topológico, tal y como se denominará a partir de ahora, está basado en neuronas de funciones base radiales con distancia de Mahalanobis ([García-Lagos et al., 2000b], [García-Lagos et al., 2003]).

4.4.1. Etapa de preproceso

La etapa de preproceso para este asistente topológico es idéntica a la etapa de preproceso del estimador topológico basado en Perceptrones Multicapa (ver sección 4.3.1.1). Por tanto, el análisis y resultados de esta etapa son los mismos que en la sección señalada anteriormente. Pasamos sin más a la descripción de la etapa de clasificación, basada en un modelo neuronal completamente diferente. La figura 4.10 presenta un esquema de esta etapa.

4.4.2. Etapa de clasificación

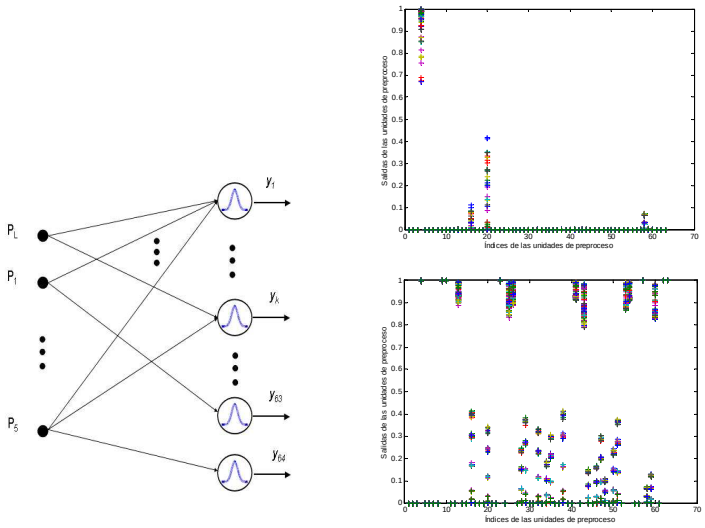
La etapa de clasificación (ver figura 4.11.a)) se compone de una capa de neuronas de potencial Gaussiano con distancia de Mahalanobis [Lee and Kil, 1991]. La función de activación para estas neuronas viene definida por la ecuación (4.4.1):

$$y(\mathbf{x}) = e^{-(\mathbf{x}-\mathbf{C_i})\mathbf{H}(\mathbf{x}-\mathbf{C_i})^T} = e^{-\sum_j \frac{(x_{ij}-Cij)^2}{\sigma_{ij}^2}} \tag{4.4.1}$$

donde $\mathbf{x}$ es el vector de entrada (salida de la etapa de preproceso),C_i es el centro del grupo o *cluster* i, y $\mathbf{H}$ es la matriz de distancias. Para implementar el sistema,$\mathbf{C_i}$ será el vector teórico asociado a la topología i, esto es, la salida preprocesada para medidas exactas (sin error) de una topología particular, y $\mathbf{H_i}$ será una matriz diagonal de covarianzas inversas asignada a cada componente de los vectores de la topologíai. Esta matriz se obtiene desde un conjunto de patrones preprocesados correspondientes a una base de datos obtenida a partir de soluciones de un programa de flujo de carga para cada topología particular. Como puede observarse de la definición de la función de transferencia, la forma de estas funciones no es una clásica Gaussiana simétrica, sino una Gaussiana asimétrica donde la influencia de cada componente de los vectores de entrada está ponderada por su correspondiente covarianza.

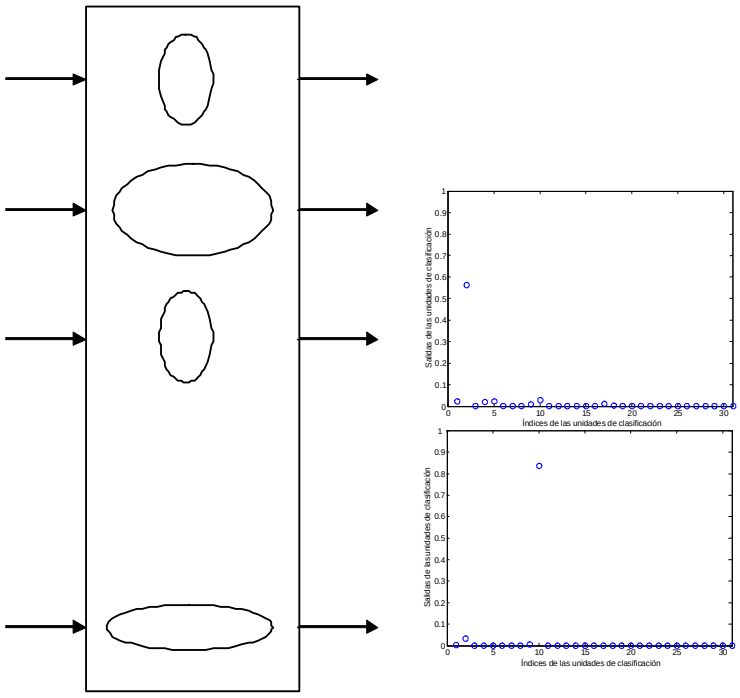
Cada una de estas unidades indicará el grado de pertenencia de un vector de entrada a una topología particular. De esta forma, un vector de entrada correspondiente a la topología i producirá una salida próxima a 1 en la unidad$i$, y una salida próxima a 0 en el resto de las unidades. La figura 4.11.b) representa, como ejemplo, las salidas de la etapa de clasificación para dos patrones de dos topologías diferentes. La topología propuesta por el estimador será aquella correspondiente al punto con un valor mayor en el eje y. En en primer caso de la figura 4.11.b) el patrón se asocia a la topología 2, y en el segundo caso a la topología 10.

Resumiendo, el asistente topológico, para cada bus, consiste en una etapa de preproceso a la que se presenta un conjunto de flujos de potencia. El vector de salida de la etapa de preproceso se presenta a la capa de neuronas de la etapa de clasificación. La topología propuesta para el patrón de flujos será aquella asociada a la neurona cuya salida sea mayor.



(a) (b)

Figura 4.10: Representación de la etapa de preproceso y la salida del asistente topológico basado en funciones potenciales de Gauss. (a) Etapa de preproceso. (b) salida de la etapa de preproceso.



(a) (b)

Figura 4.11: Esquema de la etapa de clasificación del asistente topológico basado en funciones potenciales de Gauss. (a) Representación de las neuronas de base radial con distancia de Mahalanobis, que constituyen la etapa de clasificación del asistente topológico. (b) Salida.

4.4.3. Resultados experimentales

En primer lugar señalamos que los resultados de aplicación de este método no están afectados por el tamaño de la red, debido a que se aplica independientemente a cada uno de los buses de la misma. No obstante, con el objetivo de comprobar la efectividad de método, se han utilizado las redes estándares IEEE de 14, 30, 57 y 118 buses, donde se han comprobado los resultados para todas las posibles topologías de varios buses. Por razones de brevedad sólo se muestran los resultados del bus número 15 de la red IEEE-118, pero los resultados son extensibles a los demás buses de la red. El bus número 15 tiene una inyección y líneas de conexión a los buses 13 (línea 15-13), 14 (línea 15-14), 17 (línea 15-17), 19 (línea 15-19) y 33 (línea 15-33).

Las simulaciones se han realizado para un periodo completo de 63 días. Cada patrón diario se ha obtenido a partir de la curva de la Figura 4.12 variando la carga base de cada bus de la red desde un 50 % a un 140 %. Adicionalmente, las cargas de cada bus se cambian añadiendo un ruido gaussiano de media cero y desviación estándar correspondiente a un 2 % de la carga base. Una vez definido el patrón de carga y generación de cada bus de la red, se calcula el vector de estado a partir de la solución de un problema de flujo de carga. Con este vector de estado se obtiene el conjunto de patrones de flujo e inyecciones*reales* del sistema. Para simular el conjunto de medidas que se utilizarían en un sistema real, a cada patrón de flujos e inyecciones obtenidos en el paso anterior se añade un ruido gaussiano con media 0 y desviación estándar dada por la ecuación (4.4.2) [Souza et al., 1997].

$$3\,\sigma_Z = ACC \times \mid Z \mid + FS \tag{4.4.2}$$

En esta ecuación, $|Z|$ es el valor absoluto de la medida, ACC=0.02 es un parámetro denominado exactitud y FS=0.035 es el parámetro *full-scale*. Además, se ha simulado la presencia de medidas anormalmente erróneas mediante la inclusión de un error normal con desviación estándar de 3σ a 15σ. Siguiendo este procedimiento, disponemos de un conjunto de 1500 medidas de flujo e inyecciones, algunas de ellas anormalmente erróneas, para las diferentes topologías de cada bus de las redes analizadas.

Para comprobar la funcionalidad del asistente topológico se han realizado diferentes experimentos atendiendo principalmente a dos criterios:

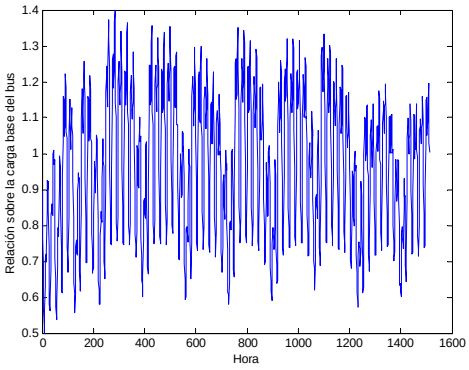


Figura 4.12: Patrón para la variación de la carga diaria de los buses.

a) Disponibilidad del conjunto de medidas asociadas a un bus o la ausencia de ellas.

b) Existencia o no de medidas anormalmente erróneas.

En las siguientes secciones se comentan los resultados obtenidos.

4.4.3.1. Todas las medidas disponibles.

Cuando todas las medidas están disponibles[1], se han realizado las siguientes pruebas: i) cada medida se ha obtenido utilizando la desviación estándarσ_i dada por la ecuación (4.4.2) ii) Se han producido algunas medidas muy erróneas, calculadas a partir de sus valores reales utilizando desviaciones estándares de $5\sigma_i$, $10\sigma_i$, y $15\sigma_i$.

Se han analizado todas las topologías posibles de cada bus. El asistente topológico propone, en todos los casos, la topología correcta. Como ejemplo mostramos los resultados paras las siguientes situaciones: la figura 4.13(a) muestra las salidas del sistema para un conjunto de medidas normales cuando la línea 15-33 está abierta (fuera de servicio). Esta configuración está asociada a la unidad 2 de la etapa de clasificación. La figura 4.13(b) muestra la salida del sistema cuando la línea 15-17 esta abierta y la inyección ha sido obtenida usando una desviación estándar de $5\sigma_i$. Esta topología está asociada a la unidad 5. La figura 4.13(c) muestra la salida del sistema cuando las líneas que están abiertas son la 15-13 y la 15-33, y el flujo de potencia de la línea 15-17 se obtiene utilizando una desviación

[1] Al igual que para el estimador de topología basado en perceptrones multicapa, para obtener el conjunto completo de medidas para un bus podemos utilizar pseudomedidas.

estándar de $10\sigma_i$. Esta topología está asociada con la unidad 18. Por último, la figura 4.13(d) muestra la salida del sistema cuando las líneas que están abiertas son la línea 15-13 y la línea 15-17, y el flujo de potencia de la línea 15-14 se obtiene usando una desviación estándar de $15\sigma_i$. Esta topología está asociada a la unidad 21. En estas figuras, el eje X representa las 31 unidades de la etapa de clasificación, donde la neurona i se asocia con la topología número i (se ha obviado la configuración en la que todas las líneas del bus están fuera de servicio). El eje Y representa las salidas de cada unidad para un patrón de entrada particular.

4.4.3.2. No todas las medidas disponibles.

En este experimento repetimos las simulaciones anteriores asumiendo que la inyección del bus o un flujo no está disponible para el bus, y esta medida no puede ser reemplazada por una pseudomedida. En esta situación, anulamos todas las componentes del vector preprocesado en las cuales interviene la medida ausente, asociando a estas componentes un valor de σ muy alto cuando se calcula la distancia de Mahalanobis. Esto significa que todos los componentes del vector preprocesado relacionados con la medida no disponible son ignorados. Sin embargo, la discriminación entre diferentes topologías es aún posible debido a que el resto de componentes es aún significativo. Esto se debe a la ley de Kirchhoff, según la cual, cuando sólo falta una de las medidas de bus, ésta puede ser calculada como la suma algebraica del resto de medidas.

En este caso, el asistente topológico produce un 100 % de situaciones correctas usando medidas erróneas con una desviación estándar de hasta $6\sigma_i$. Los resultados de estos experimentos se muestran en la figura 4.14. Así, la figura 4.14(a) muestra la salida del sistema para un conjunto de medidas normales cuando las líneas 15-14 y 15-33 están abiertas, y la inyección de potencia no está disponible. Esta topología está asociada con la unidad 10. La figura 4.14(b) muestra la salida del sistema cuando está abierta la línea 15-17, el flujo de potencia de la línea 15-13 no está disponible y la inyección se obtiene utilizando una desviación de $3\sigma_i$. Esta topología está asociada con la unidad 5. La figura 4.14(c) muestra la salida del sistema cuando las líneas abiertas son la 15-17 y la 15-19, el flujo de potencia de la línea 15-17 no está disponible y el flujo de la línea 15-33 se obtiene mediante una desviación de $5\sigma_i$. Esta topología está asociada con la unidad 7. Por último, la figura 4.14(d) muestra la salida cuando las líneas abiertas son las 15-13, 15-17, 15-19 y 15-33, el flujo de potencia de

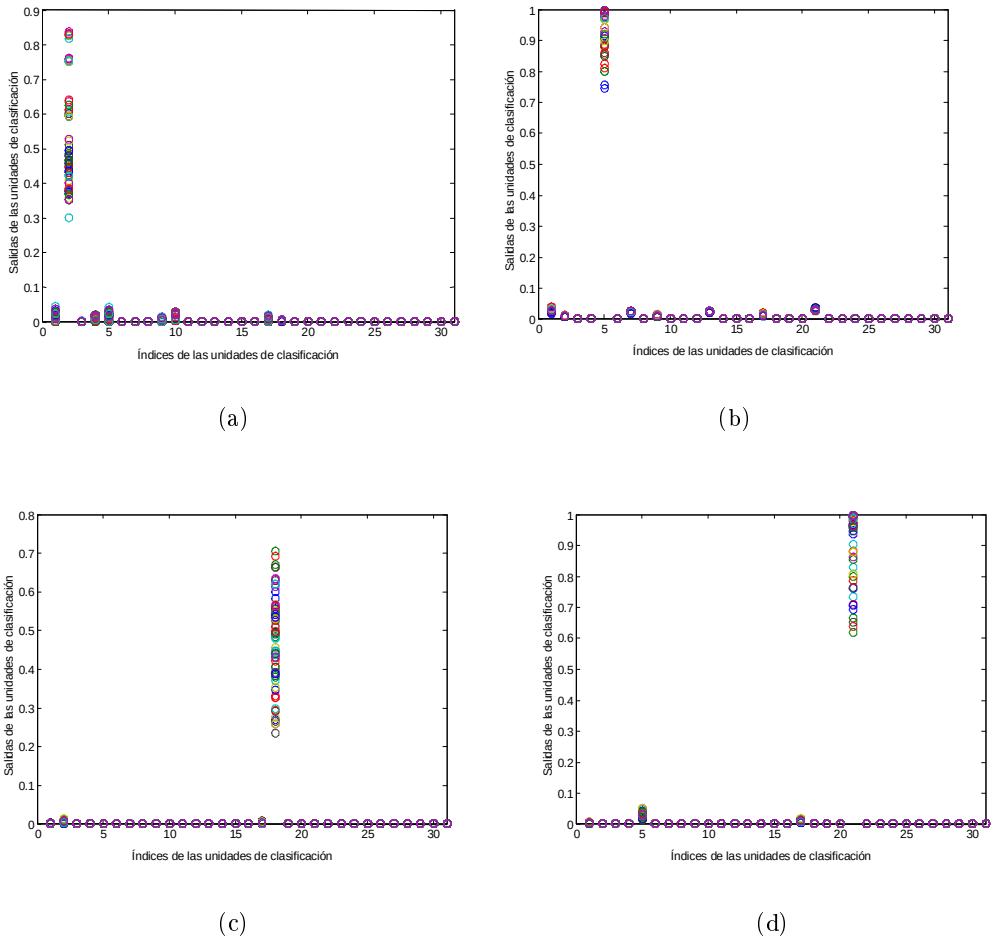


Figura 4.13: Salidas de la etapa de clasificación para patrones pertenecientes a diferentes topologías, cuando todas las medidas del bus están disponibles.

la línea 15-17 no está disponible y la inyección de potencia se obtiene usando una desviación estándar de $6\sigma_i$. Esta configuración de las líneas está asociada a la unidad 24. Cuando los errores en las medidas erróneas son mayores de $6\sigma_i$ los resultados de la clasificación no son satisfactorios en muchos casos, por lo que se ha fijado este umbral como valor máximo para el funcionamiento correcto del asistente topológico en el caso de ausencia de una medida.

4.4.4. Conclusiones sobre los resultados del asistente topológico basado en unidades de funciones base radiales con distancia de Mahalanobis

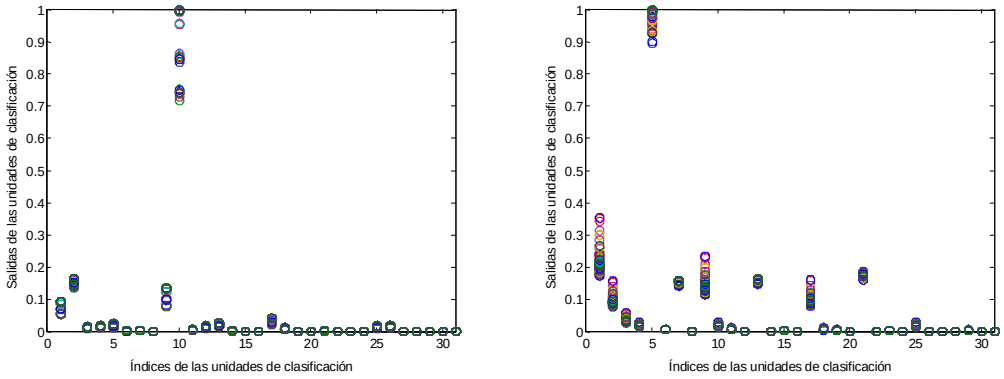
En esta sección, se ha presentado un nuevo método para la determinación de la topología de una red compatible con el conjunto de medidas disponibles en el sistema. El asistente topológico está formado por dos etapas. La primera es una etapa de preprocesamiento, que está compuesta por una capa de elementos de proceso con función radial, que transforman las inyecciones y flujos de potencia activa de cada bus en un nuevo vector cuyas componentes pertenecen al intervalo [0,1]. Esta transformación produce agrupaciones de patrones similares, de tal forma que dos patrones de entrada correspondientes a la misma topología producirán patrones preprocesados muy similares. La segunda etapa, de clasificación, se compone de una capa de unidades con función de activación Gaussiana con distancia de Mahalanobis. En esta capa hay una unidad por cada posible configuración topológica de un bus, de forma que cuando se presenta a la misma un vector preprocesado, la unidad cuya salida sea mayor indica la topología que debe asociarse al vector de entradas de inyección y flujos original. En comparación con otras técnicas numéricas y neuronales, este modelo presenta las siguientes ventajas:

a) El proceso de estimación es local, y la topología de cada bus se obtiene de una forma independiente y en paralelo.

b) Como consecuencia, la complejidad del sistema crece linealmente con el tamaño del sistema de potencia.

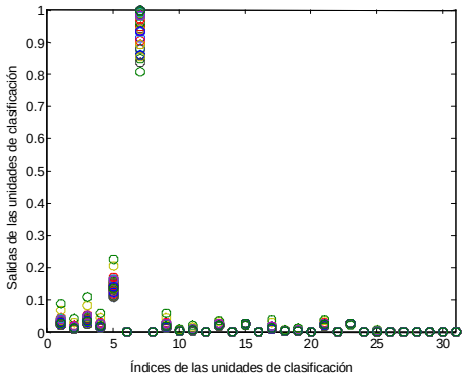
c) Los parámetros involucrados en el modelo se obtienen de forma fácil y directa desde una base de datos histórica, no requiriendo de ningún proceso de entrenamiento.

d) Para los buses en los que se dispone de todas las medidas asociadas (medidas de campo

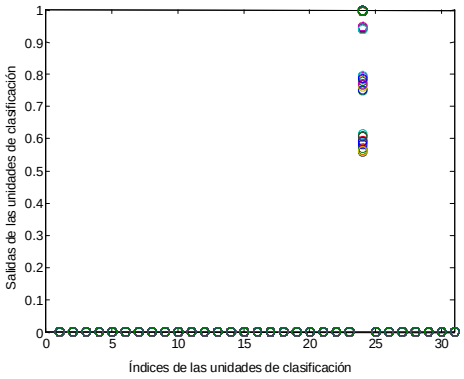


(a)

(b)



(c)



(d)

Figura 4.14: Salidas de la etapa de clasificación para patrones pertenecientes a diferentes topologías, cuando una de las medidas del bus no está disponible.

o pseudomedidas), el sistema obtiene la topología correcta del bus incluso para medidas generadas con un error muy alto, de hasta 6σ.

e) El asistente topológico puede usarse cuando un flujo de medida o inyección no está disponible; en esta situación, se obtiene la topología correcta incluso con medidas anormalmente erróneas.

f) El rendimiento del sistema es altamente inmune a grandes cambios en la curva de carga del sistema.

Capítulo 5

Análisis de seguridad

5.1. Introducción

El objetivo principal de un sistema de seguridad para un sistema de energía eléctrica es mantener el punto de operación de éste dentro de una zona de seguridad, garantizando así un suministro de energía continuo y aceptable. Las nuevas condiciones del mercado de la energía, originadas por nuevas circunstancias como la deregulación, el incremento del consumo, y las restricciones económicas, sociales y medioambientales en la construcción de nuevas redes, fuerzan al sistema a trabajar cerca de sus límites de seguridad, produciendo puntos de operación cada vez menos conservativos. En consecuencia, se hace necesaria la monitorización continua del sistema para detectar situaciones de peligro lo antes posible. En particular, la tarea de Análisis de Contingencias debe informar de si el estado actual del sistema es seguro, crítico o inseguro con respecto a cada posible contingencia, esto es, a un posible fallo en un componente particular del sistema. Esta tarea puede ser abordada bien por métodos funcionales, caracterizando la gravedad de cada contingencia mediante un valor numérico, o bien por métodos gráficos, permitiendo la evaluación de cada contingencia de una manera visual. En ambos casos, el análisis de contingencias está condicionado por dos factores: por una parte, el modelado del sistema requiere del uso de miles de variables con valores ruidosos o parcialmente conocidos; por otra parte, el comportamiento del sistema es altamente no lineal con respecto a estas variables.

Como se acaba de señalar, la evaluación de la seguridad del sistema incluye la clasificación del estado actual para decidir si opera en un estado seguro, crítico o inseguro. La

seguridad del sistema se evalúa tanto para el estado actual como para un número de estados simulados derivados del estado actual asumiendo que una o varias líneas, transformadores o generadores queden fuera de servicio (*outage*).

Las técnicas más empleadas para realizar esta tarea se basan, en mayor o en menor medida, en la utilización de soluciones de flujo de carga para evaluar el estado del sistema [Warwick et al., 1997]. El modelo de flujo de carga de un sistema de potencia consiste en una serie de ecuaciones algebraicas no lineales cuyas variables dependientes son las inyecciones netas de potencia de los buses (la generación de potencia en el bus menos la carga); los parámetros son las impedancias; y las variables independientes son las tensiones complejas de cada bus. Para el cálculo de las tensiones complejas de cada bus y la clasificación del estado actual del sistema es necesario resolver numéricamente este sistema de ecuaciones, generalmente utilizando un modelo linealizado tal como el método de*Newton*. Para decidir sobre la seguridad del sistema, la tensión de cada bus y el flujo de potencia por cada línea deben compararse con los valores máximos y mínimos soportados por los diferentes componentes de la red, es decir, para que el sistema sea clasificado como seguro, la solución del problema de flujo de carga debe satisfacer los límites de corriente máxima que puede circular por una línea, los límites de potencia reactiva, los límites de tensión y los límites de caída de tensión. Para realizar un análisis de seguridad completo es necesario considerar todas las posibles combinaciones de fallos en los elementos de la red, lo que convierte al problema en intratable debido al crecimiento exponencial del número de casos posibles a resolver con respecto al número de contingencias posibles.

La estimación de la seguridad que se acaba de describir se denomina análisis de seguridad estática. Cuando se tiene en cuenta el comportamiento temporal del sistema, el sistema de potencia viene descrito por un conjunto de ecuaciones diferenciales no lineales cuyas condiciones límites (también llamadas condiciones frontera) vienen dadas por las ecuaciones no lineales del problema de flujo de carga descritas anteriormente. Por tanto, la seguridad estática es una aproximación de primer orden de la seguridad dinámica [Warwick et al., 1997]. En este trabajo, sólo se considera la seguridad estática.

Aunque los avances en telecomunicación y computación permiten un conocimiento cada vez más detallado de los sistemas de energía, las técnicas de análisis clásicas, basadas en la solución de problemas de flujo de carga siguen siendo insatisfactorias ya que no pueden

garantizar soluciones en tiempo real para todas las posibles contingencias. Sin embargo, son estos métodos los más extendidos en al actualidad, principalmente, los basados en el cálculo desacoplado rápido de los flujos de carga (*Fast Decoupled Load Flow* en la nomenclatura inglesa) [Monticelli et al., 1990]. Aunque con este método se consigue reducir significativamente el número de cálculos implicados, su principal inconveniente sigue siendo la cantidad de cálculos de flujos de carga necesarios para la determinación de los flujos de línea y tensiones de buses para cada contingencia, lo que convierte a estos métodos en inaplicables para propósitos de tiempo real. Además, puesto que están basados en algoritmos de convergencia rápida para el cálculo de flujos de carga, la convergencia no está garantizada en casos de sobrecarga del sistema.

En este capítulo, se profundiza en el análisis de la aplicabilidad de las RNAs al problema del análisis de contingencias. En primer lugar, se analiza el enfoque funcional del análisis de contingencias describiendo la aplicación de los paradigmas Perceptrón Multicapa (MLP) con aprendizaje *Backpropagation* y Redes de Funciones Base Radiales (RBF). En segundo lugar, se analiza el enfoque visual describiendo la aplicación de los Mapas *Auto-Organizativos* de Kohonen tanto bidimensionales como lineales. Esta última aplicación nos parece especialmente interesante ya que proporciona un sistema de monitorización de contingencias extremadamente rápido y amigable desde el punto de vista de un operador del sistema. Como casos de estudio se han utilizado las redes estándar IEEE-14 e IEEE-118 y los patrones de entrada han sido obtenidos a partir de un caso base siguiendo la metodología explicada más adelante. En consecuencia, la estructura del capítulo queda como sigue. En la sección 5.2 se presenta un breve repaso bibliográfico sobre las técnicas más extendidas para análisis de contingencias. La sección 5.3, *Generación de patrones e Índices de Prestación* (*Performance Index* en inglés) explica el proceso de generación de los patrones usados como entradas a los distintos sistemas neuronales, así como la expresión matemática de los índices de prestación usados para la evaluación numérica de las contingencias. La sección 5.4, *Evaluación numérica y ordenación de contingencias con redes neuronales alimentadas hacia delante y aprendizaje supervisado*, está dedicada al estudio de este tipo de redes, concretamente el MLP y RBF, aplicadas a la evaluación numérica de contingencias. La sección 5.5, *Visualización de contingencias con Mapas Auto-Organizativos de Kohonen*, desarrolla el análisis de este tipo de redes como herramientas para la visualización rápida del nivel

de gravedad de contingencias. Finalmente, la sección 5.6, *Conclusiones*, realiza un breve resumen de las principales aportaciones del capítulo.

5.2. Principales métodos de análisis de contingencias

El análisis de contingencias debe considerar la posibilidad de fallo de diferentes componentes del sistema para evaluar la seguridad del mismo, estimando la probabilidad y gravedad de una situación insegura. Debido a la naturaleza no ideal del funcionamiento de los diferentes elementos físicos del sistema, al comportamiento del consumo eléctrico o a condiciones externas climáticas que pueden influir en el funcionamiento del sistema de potencia, el análisis de seguridad presenta características no deterministas.

Los métodos Monte Carlo se han utilizado ampliamente para el análisis probabilístico de los sistemas de potencia, tanto para tareas de planificación ([Bertoldi et al., 1988]) como para determinar su fiabilidad ante fallos en determinados componentes. Así, en [Lerma, 1990] se presenta la aplicación de este método para el análisis de seguridad estático, donde para varios elementos de la red (generadores, cargas y líneas) se estima su probabilidad de estar fuera de servicio, y se supone que el consumo de energía en cada nodo de la red alcanza su valor máximo. El análisis completo del sistema permite obtener la probabilidad de que el consumo de energía no sea satisfecho, así como obtener los puntos de la red más vulnerables, lo que puede ser utilizado para tareas futuras de planificación. En [Pereira et al., 1992] se presenta una estructura general que permite combinar los métodos analíticos y las simulaciones Monte Carlo para el análisis de sistemas de potencia. La idea principal de este trabajo es la utilización de un método analítico aproximado simple como aproximación inicial para un modelo más detallado tipo Monte Carlo. Analizando las diferencias encontradas por el método analítico aproximado y el obtenido por simulaciones Monte Carlo, se generan unos índices probabilísticos que permiten identificar las situaciones en las que el método analítico, computacionalmente menos costoso, puede ser aplicado con garantías, utilizando las simulaciones Monte Carlo en los casos donde los resultados analíticos no son aceptables. Esta combinación hace compatibles los métodos analíticos con los probabilísticos, permitiendo en muchos casos obtener lo mejor de cada uno de ellos. En [Billinton and Sankarakrishnan, 1995] se analizan varios métodos basados en simulaciones Monte Carlo,

comparando especialmente los métodos secuenciales y no secuenciales. El trabajo justifica la validez del método secuencial para la obtención de índices que permiten cuantificar la frecuencia y duración de interrupciones del suministro eléctrico. El modelado de la incertidumbre del sistema puede ser mejorado considerando que la demanda del sistema también tiene comportamiento probabilístico. Así, en [Saraiva et al., 1996] se utiliza la simulación Monte Carlo junto con lógica difusa. El consumo de potencia de cada bus se modela mediante reglas difusas, obteniendo de esta forma una *función de pertenencia* para la potencia no suministrada por el sistema, mediante la resolución de un problema de carga óptimo difuso (*fuzzy optimal power flow*) ([Miranda and Saraiva, 1992]), lo que permite obtener un estado más exacto del sistema, a costa de un tiempo de cómputo mayor. Otros trabajos [Billinton and Jonnavithula, 1996] intentan reducir la complejidad computacional mediante técnicas de reducción de varianza del método Monte Carlo. En [Mori and Yuihara, 1998] se utiliza un método basado en análisis de intervalos que permite reemplazar a los métodos Monte Carlo, métodos de flujo de carga estocásticos y a los métodos de flujo de carga difusos en el sentido de que simplifica el manejo de variables estocásticas, mejorando la eficiencia de los métodos probabilísticos.

Los métodos Monte Carlo presentan las ventajas de que son flexibles y permiten un alto detalle en la simulación de sistemas complejos, tanto en su modo de operación como en su configuración. Sin embargo, en general, los métodos basados en técnicas Monte Carlo presentan los siguientes problemas: requieren de un alto tiempo de cómputo para obtener una estimación fiable del estado de la red, incluso utilizando nuevas técnicas de reducción de complejidad; debido a su base probabilística, es difícil decidir cuándo terminar la simulación; la obtención de las funciones de distribución de probabilidad es a veces una tarea muy compleja; pueden presentar problemas de convergencia; el modelado más preciso de variables con componentes aleatorias como la demanda de energía, aunque hace más fiable la evaluación del estado actual del sistema, incrementan los recursos necesarios para realizar el análisis. Debido a estos inconvenientes, estos métodos se han utilizado principalmente para tareas de planificación, y no para el análisis de contingencias en tiempo real.

Otros métodos de análisis de contingencias no probabilísticos se basan en la obtención de índices, generalmente mediante la solución de problemas de flujo de carga, que permiten evaluar la gravedad de las contingencias consideradas en la simulación. Estos índices suelen

considerar como problemas independientes el análisis de seguridad de tensiones (mantenimiento de los niveles de tensión en cada bus dentro de sus márgenes de seguridad), y el análisis de la capacidad de suministro de la potencia demandada en cada bus del sistema (máximo consumo permitido en un bus para mantener la estabilidad de la red), ya que las contingencias que causan sobrecargas en las líneas no necesariamente causan violaciones en los márgenes de tensión y viceversa ([Albuyeh et al., 1982]). Otra alternativa, dentro de los métodos clásicos, aparece en [Vaahedi et al., 1999], donde se describe un módulo de monitorización (*screening*) y ordenación (*ranking*) de contingencias basado en Índices de Soporte Reactivo RSI (en inglés, *Reactive Support Index*) y un proceso de filtrado iterativo.

Con el objetivo de reducir los tiempos de cómputo requeridos para realizar la simulación del gran número de posibles contingencias en un sistema grande, se han propuesto diferentes técnicas como la utilización de algoritmos de flujos de carga más eficientes, la reducción (mediante selección) del número de contingencias consideradas o la distribución del trabajo. Así, en [Ejebe et al., 1988] se propone un algoritmo que identifica la localización de los buses que potencialmente pueden presentar problemas de márgenes de tensión, definiendo una subred sobre la que realizar el análisis de contingencias, que permite el análisis de un número mayor de ellas. Sin embargo, la definición de la subred debe realizarse para cada posible contingencia. En [Ajjarapu and Christy, 1992] se presenta un método para encontrar de forma continua soluciones de problemas de flujo de carga comenzando con un caso base y continuando hasta encontrar los límites de estabilidad de tensión del sistema (punto crítico). La idea general de este método es emplear un esquema de predicción-corrección para encontrar un camino hacia la solución de un conjunto de ecuaciones que se han modificado para considerar cambios en la carga del sistema. La principal ventaja de este método es que no encuentra problemas de condicionamiento numérico cerca del punto crítico. En [Ejebe et al., 1996] se comparan cuatro métodos diferentes para evaluar la distancia hasta el punto de colapso de tensión para un nivel de carga particular y se proponen mejoras tanto en eficiencia como en exactitud de resultados. En [Lemaître and Thomas, 1996] se describe el proceso de paralelización de un sistema de análisis de seguridad y de coste de operación basado en simulaciones Monte Carlo que permite reducir los tiempos de cómputo. En [Santos et al., 1999] se propone el análisis de contingencias mediante la resolución de problemas de flujo

modificados en dos aspectos importantes. En primer lugar, se proponen cambios en el proceso de solución para ahorrar operaciones aritméticas en relación al método clásico. Por otro lado, se propone una estructura cliente/servidor heterogénea (los clientes no tienen por qué tener las mismas características) para la distribución de la carga computacional a través de computadores conectados en red mediante *sockets*. La estructura del análisis es la siguiente: el servidor realiza la resolución del caso base y selecciona aquellas contingencias que deben ser analizadas. Cada vez que un cliente queda libre, el servidor le envía datos (vectores y matrices) del caso base y de la contingencia que debe analizar. Cuando el cliente termina de analizar la contingencia, envía los resultados al servido y queda libre para procesar otra contingencia. Las ventajas de este sistema distribuido son: la flexibilidad, ya que el número de clientes puede variar dinámicamente y éstos pueden ser diferentes tanto en configuración hardware como software; la eficiencia, ya que mediante una política de distribución de tareas dinámica permite asignar contingencias a clientes según su carga computacional actual (y no sólo potencial); y por último, fiabilidad, ya que si un cliente queda fuera de servicio, la contingencia que está analizando será asignada a otro cliente.

Aunque se han propuesto numerosas mejoras a los métodos clásicos de análisis de contingencias, todos ellos sufren en mayor o mejor medida de limitaciones temporales debido a que la evaluación de un gran número de contingencias no es posible de realizar en tiempo real. Otros métodos, como el presentado en [Vaahedi et al., 1999] permiten mejorar los resultados, pero a costa de añadir etapas de filtrado que sobrecargan el sistema. Por otro lado, aunque las técnicas de paralelización y distribución de la carga computacional presentan ventajas potenciales frente a los sistemas centralizados, la disponibilidad de hardware y software específico supone una desventaja importante en muchos casos, además de que suponen un aumento de la complejidad en la gestión de los sistemas.

Otros métodos para abordar el problema de la seguridad están basados en sistemas expertos [Fujiwara et al., 1986], que intentan explotar el conocimiento humano formulando explícitamente reglas que permitan predecir los efectos de diferentes contingencias y asistir al operador en la toma de decisiones. La lógica difusa es otro método basado en reglas que intenta cuantificar el conocimiento cualitativo del sistema por parte del operador, y que ha sido aplicado a seguridad exitosamente. Así, en [Hsu and Kuo, 1992] se presenta un sistema basado en lógica borrosa para la ordenación (*ranking*) de contingencias. El proceso

de ordenación comprende dos etapas. En primer lugar, se resuelve un problema de flujo de carga simulando una contingencia. Los valores obtenidos, codificados convenientemente, se emplean en una segunda etapa como entrada al controlador difuso, cuyas reglas han sido generadas a partir de la experiencia de los operadores del sistema. La salida del controlador difuso es un índice de peligrosidad de la contingencia que se utiliza para clasificarla dentro de la lista de contingencias analizadas anteriormente. La ventaja principal de este método es que permite incorporar la experiencia de los operadores en el proceso de ordenación. Además, las reglas difusas pueden ser calibradas durante el funcionamiento del sistema, lo que permite la mejora del producto a lo largo del tiempo de funcionamiento del mismo. Otro método basado en lógica borrosa se presenta en [Lo and Abdelaal, 2000], cuyos resultados tienen una calidad comparable a los obtenidos con métodos clásicos computacionalmente mucho más costosos. En [Matos et al., 2000] se presenta una solución basada también en lógica borrosa. El método consta de varias etapas. En la primera, la etapa de generación de patrones para entrenamiento, se deben generar numerosos patrones correspondientes a distintos puntos de operación, que deben contener gran diversidad en niveles de carga. Para cada patrón generado se determina si el sistema es seguro o inseguro. La segunda etapa se denomina de selección de características, en la que los patrones generados en la etapa anterior se simplifican mediante técnicas estadísticas. La tercera etapa consiste en el diseño de un clasificador difuso para cada una de las contingencias disponibles. La última etapa es la de clasificación *on-line*, en la que cada nuevo patrón se clasifica en un estado seguro o inseguro, con un coste computacional mínimo. La ventaja de este método radica en que la carga computacional es alta únicamente en las tres primeras fases del método, que se realizan *off-line* y que sólo hay que realizar una vez. Además, a diferencia de la mayoría de los trabajos relacionados, permite analizar contingencias dobles para una topología particular.

Los métodos que emplean lógica difusa presentan el inconveniente de que es necesario *fuzzilizar* las variables del sistema, para que el método pueda ser aplicado satisfactoriamente. Para realizar esta tarea es necesario definir diferentes umbrales para las variables que intervienen en el sistema, realizar una cuantificación difusa (*fuzzy quantization*) para construir la funciones de pertenencia y seleccionar el conjunto de operadores difusos a utilizar para el problema particular. La precisión con la que realice esta operación determinará la eficiencia del sistema. En todo caso, este tipo de solución suele ser poco portable, requiriendo la

reconstrucción del sistema cuando se quiere llevar a otro entorno diferente.

Las técnicas basadas en Redes Neuronales Artificiales constituyen una muy interesante alternativa a los métodos clásicos de evaluación de seguridad. Estas técnicas disponen de diferentes paradigmas especialmente aptos para llevar a cabo el análisis de contingencias, tanto desde el enfoque funcional como visual. Así, los paradigmas neuronales alimentados hacia delante y aprendizaje supervisado como el Perceptrón Multicapa o las redes de Funciones Base Radiales, pueden ser utilizados como aproximadores de las funciones numéricas que evalúan el nivel de seguridad de cada contingencia, permitiendo así la construcción de un ranking de contingencias. Por otra parte, los paradigmas no supervisados como los Mapas Auto-Organizativos de Kohonen, caracterizados por su capacidad para extraer criterios de clasificación no evidentes a partir del procesamiento no supervisado de un conjunto de patrones y por su capacidad para mostrar posibles relaciones entre patrones en un espacio dimensionalmente reducido, pueden ser usados en un sistema de clasificación visual de contingencias de acuerdo con su nivel de seguridad.

En [Yan et al., 1991] se presenta un modelo basado tanto en sistemas expertos como en RNAs para la monitorización de la seguridad del sistema. La primera etapa consiste en un sistema experto que determina si una contingencia es segura o no, y se clasifica según el tipo de problemas que puede generar. Las contingencias que son potencialmente peligrosas se pasan a un segunda etapa, compuesta por tres RNAs, que determina su nivel de gravedad. Cada una de las tres redes neuronales de la segunda etapa está entrenada para especializarse en un tipo particular de contingencia.

En [Chung and Ying, 1998] se utiliza un MLP como equivalente al modelo externo de una red eléctrica, de forma que se consigue una reducción en el modelo equivalente del sistema que se resolverá mediante flujos de carga, cuyos resultados son utilizados para el análisis de contingencias. El objetivo final es la reducción de los tiempos de cómputo en el análisis de contingencias.

Un sistema basado en redes neuronales tipo *Counterpropagation* ([Hecht-Nielsen, 1987]) se presenta en [Lo et al., 1998]. La salida de la red neuronal es el valor de un índice que permite determinar de forma aproximada la gravedad de la contingencia. Los patrones de entrenamiento se generan mediante la solución de problemas de flujo de carga, y se simulan un número de contingencias predefinidas por el usuario. Para cada solución se calcula el

correspondiente índice de severidad. Los resultados son prometedores en el sentido de que la red clasifica la gravedad de las contingencias de la misma forma que lo hace la ordenación generada por un problema de flujo de carga.

La capacidad de las redes neuronales del tipo Redes de Funciones Base Radiales [Moody and Darken, 1989] para el análisis de contingencias se analiza en [Maghrabi et al., 1998] y [Refaee et al., 1999]. En [Maghrabi et al., 1998] se utilizan para tareas de planificación, no requiriendo un funcionamiento en tiempo real. Este trabajo se mejora y amplía en [Refaee et al., 1999], donde se explota la capacidad de este tipo de redes para estimar los flujos de potencia de las líneas y la tensión de los buses en el caso de la ocurrencia de una contingencia. Se utilizan dos RBF: la primera, se entrena para hacer corresponder la matriz de admitancias de la red, la carga y generación de cada bus con los flujos por las distintas líneas de la red; la segunda red, utiliza las mismas entradas, pero obtiene las tensiones de cada bus. El sistema, al igual que su predecesor, sólo se utiliza para tareas de planificación de la expansión del sistema de potencia.

En [Luan et al., 2000] se utiliza el método de regresión R-ReliefF ([Robnik-Sikonja and Kononenko, 1997]) para reducir la dimensión de los patrones de entrada a un modelo neuronal. Los patrones de entrada están compuestos por las inyecciones de potencia en cada bus de la red en una situación de pre-contingencia, y la salida es un índice de prestación correspondiente a la situación post-contingencia. El método se basa en considerar como un problema de regresión no lineal las relaciones entre las entradas a la red y el índice de prestación (PI) correspondiente a varias contingencias.

Una aplicación del MLP para monitorización y ordenación de contingencias se describe en [Sidhu and Cui, 2000]. A partir de la solución de problemas de flujo de carga realizados *off-line*, se generan patrones para entrenar un MLP. Las entradas son los 27 armónicos principales obtenidos a partir de la aplicación de una transformada rápida de Fourier al vector cuyas componentes son la generación y la carga de cada uno de los buses de la red IEEE-14, además de un número entero que indica la contingencia que se está analizando. Esta transformación permite disminuir el número de iteraciones necesarias para entrenar a la red neuronal. Las salidas de la red son dos índices que representan el grado de peligrosidad del vector de entrada original. Los resultados demuestran que el MLP es capaz de generalizar bien, requiriendo en la fase de ejecución un tiempo considerablemente menor que el requerido

para realizar el mismo análisis mediante la solución de flujos de carga utilizando el método rápido desacoplado. El principal inconveniente de este método está en la existencia de una única red neuronal que debe evaluar todas las contingencias, diferenciando contingencias únicamente mediante una entrada adicional. Este tipo de modelos hace que el proceso de entrenamiento de la red sea muy costoso y difícil y, en la mayoría de los casos, no produce buenos resultados de generalización durante la fase de operación.

En [Chicco et al., 2001] se analiza la capacidad de generalización de tres paradigmas neuronales diferentes (RBFs, PLN (*Progressive Learning Network*) y SOM) para la predicción *on-line* de tensiones en los buses de un sistema, con el objetivo de obtener de forma rápida las tensiones del sistema ante contingencias. Los mejores resultados son obtenidos por redes tipo RBF.

En [Kim and Singh, 2002] se aborda el problema del análisis de seguridad probabilístico, utilizando tanto SOMs como simulaciones Monte Carlo. El análisis se lleva a cabo tanto para estabilidad transitoria como estabilidad en estado estable, mediante la definición de tres índices probabilísticos para cada contingencia. A partir del análisis de estos tres índices para cada posible contingencia, se decide si el sistema es seguro o inseguro. Para reducir la cantidad de cálculos que se necesitarían para evaluar, para cada contingencia, estos índices por simulaciones Monte Carlo, se utiliza una SOM que realiza una clasificación de los patrones para varios años. Para cada clase del mapa, se evalúan estos índices, decidiendo si el sistema es o no seguro. Cada neurona se marca como segura o insegura en función de los índices. Esta operación se realiza únicamente en la fase de entrenamiento del sistema. En la fase de ejecución sólo se utiliza el SOM, presentando un patrón a su entrada y declarándolo seguro o inseguro según la neurona que ha ganado.

A partir del estudio de las técnicas de análisis de contingencias que utilizan modelos neuronales, se pueden obtener las siguientes conclusiones:

- En la mayoría de los trabajos en los que se utilizan RNAs para el análisis de contingencias el objetivo o salida de la red es la clasificación de un patrón de entrada en uno de dos posibles estados: seguro o inseguro. En general esta información no es suficiente, ya que es conveniente indicar además el grado de seguridad o inseguridad del patrón, así como una evolución del mismo. Esto permite tener una perspectiva general del estado

actual y una predicción a corto plazo del estado del sistema.

- Los modelos neuronales que emplean métodos de regresión (p.e. [Luan et al., 2000]) tienen el inconveniente de que para realizar el modelo de regresión, deben considerar solamente un número reducido de contingencias, ya que de otra forma el modelo neuronal resulta muy complejo.

- La inclusión de la topología de la red en el modelo es una idea interesante porque permite analizar múltiples contingencias indirectamente. Esto se ha hecho por ejemplo en [Refaee et al., 1999], utilizando la matriz de admitancias. Sin embargo, la capacidad de generalización de la red dependerá en gran medida de los patrones utilizados en la fase de entrenamiento (es imposible generar todas las Y-matriz combinaciones para todas las topologías diferentes.

- La inclusión de una entrada particular que indica la topología que se está estudiando, exige el uso de patrones de entrada heterogéneos, en los que aparecen mezcladas variables cualitativas (variables discretas, generalmente) y magnitudes medidas (variables continuas, generalmente) que convierten el proceso de aprendizaje de la red en una tarea muy difícil. Por otro lado, la consideración de contingencias distintas a la n-1 requeriría la creación de un modelo neuronal nuevo.

- La aplicación de los modelos analizados a redes de gran tamaño requiere la reducción de la dimensionalidad de los patrones de entrada (*feature selection*), pero en general no se proponen métodos efectivos para esta tarea, o los que se proponen son computacionalmente muy complejos.

En este trabajo proponemos el uso de técnicas neuronales que evitan en gran medida estos inconvenientes. Así, describimos sistemas neuronales apropiados tanto para la selección de variables, la monitorización visual o la evaluación numérica y ordenación de contingencias.

5.3. Generación de patrones e índices de prestación

Como casos de estudio se han utilizado las redes estándares IEEE-14 e IEEE-118. Para ambas, se han definido dos índices diferentes a partir del principio de desacoplo entre las potencias activa y reactiva: Índice de Prestación de Potencia Activa (PIp) e Índice de Prestación de Potencia Reactiva (PIv). Las expresiones que describen ambos índices aparecen en las ecuaciones (5.3.1) y (5.3.2) ([Sidhu and Cui, 2000]),

$$PIp = \sum_{\alpha} w_{pl}(\frac{P_l}{P_{l,lim}})^2 \tag{5.3.1}$$

$$PIv = \sum_{\beta} w_{vi}\frac{\mid V_i - V_{i,lim} \mid}{V_{i,lim}} \tag{5.3.2}$$

donde P_l es el flujo de potencia activa en la línea l, $P_{l,lim}$ es el límite permitido para el flujo de la potencia activa en la línea l, w_{pl} es el factor de ponderación de gravedad de una sobrecarga en la línea l, V_i es la magnitud de tensión en el bus i, $V_{i,lim}$ es el límite permitido para la tensión en el bus i, w_{vi} es el factor de ponderación de la gravedad de una sobretensión en el bus i, y α y β son los conjuntos conteniendo únicamente las líneas cuyo flujo de potencia activa es superior a su potencia límite y los buses cuya tensión es superior a su tensión límite, respectivamente. En adelante, los índices de prestación de una contingencia particular i se indicarán como PIp, i y PIv, i.

Los patrones de entrada a las redes neuronales implementadas se obtienen a partir de un análisis en componentes principales (ACP) del conjunto de los flujos de potencia activa y reactiva de todas las líneas del sistema. El patrón de flujos de potencia utilizado consiste en los valores de potencia horarios de 41 días obtenidos a partir del caso base de cada una de las redes IEEE mediante el siguiente procedimiento:

- Cada carga horaria del caso base se multiplica por un coeficiente variable entre0,5 y 1,8, lo que simula un patrón de carga próximo a la realidad (ver figura 5.1).

- Esta carga global se distribuye entre todos los buses de acuerdo con su capacidad de carga predefinida.

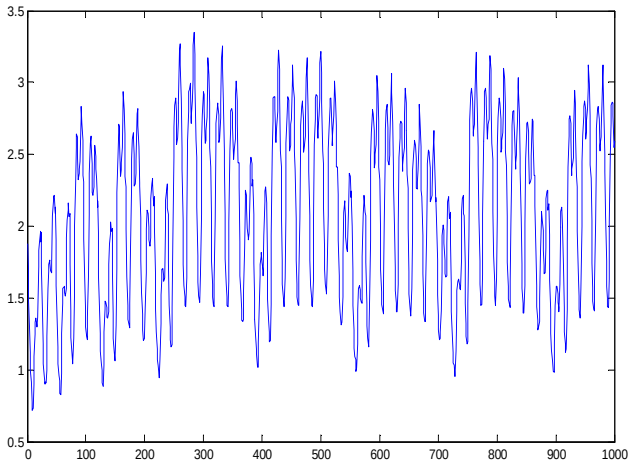


Figura 5.1: Coeficientes aplicados al patrón base de cada red para obtener patrones con diferentes cargas del sistema

- Cada carga de bus se somete a un ruido normalmente distribuido de media 0 y desviación típica 2 %.

- Finalmente, a partir del patrón de cargas de buses, se lleva a cabo la solución del problema de flujo carga obteniendo el patrón definitivo de flujos de potencia activa y reactiva.

5.3.1. Selección de características

Uno de los inconvenientes del análisis de contingencias para redes eléctricas medianas o grandes es la gran cantidad de variables que deben ser analizadas para decidir sobre el estado seguro o inseguro de una contingencia particular. Para resolver este problema se utilizan técnicas conocidas como de selección o extracción de características (*feature selection* en inglés), cuyo objetivo es encontrar un nuevo conjunto de variables de entrada menor que el original (figura 5.2). Existen varias técnicas de extracción de características aplicadas especialmente al análisis de contingencias, entre las que cabe destacar las descritas en [Luan et al., 2000], [Mazon et al., 2001], [Fidalgo and Lopes, 2001] y [Jensen et al., 2001].

5.3 Generación de patrones e índices de prestación

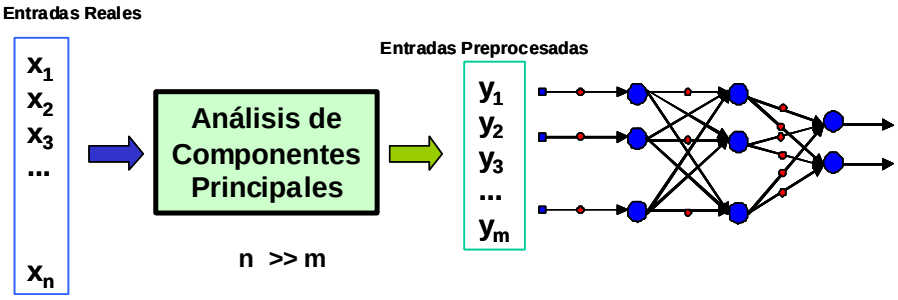


Figura 5.2: Representación esquemática de la selección de características mediante Análisis por Componentes Principales para RNAs..

En este trabajo se ha empleado la técnica denominada Análisis por Componentes Principales ([Jollife, 1986], [Jackson, 1991]), que se describe brevemente en esta sección.

El coeficiente de correlación lineal, o la medida asociada denominada covarianza, constituye la base del análisis por componentes principales. Considerando todas las covarianzas de todas las parejas de variables de entrada y sus valores, es posible definir un nuevo conjunto de variables de entrada denominadas componentes principales. El objetivo final del análisis de componentes principales es transformar un espacio de representación P en un nuevo espacio P' en el que los datos estén incorrelados, es decir, busca encontrar un nuevo conjunto de ejes ortogonales en el que la varianza de los datos sea máxima, de forma que la dimensionalidad del problema se reduzca. Esta transformación se puede expresar mediante una función g tal que verifique la ecuación (5.3.3):

$$Y = g(X) \tag{5.3.3}$$

donde X representa el espacio de patrones original, e Y el conjunto de patrones derivados o *características*. En la práctica, la transformación que se realiza es lineal, por lo que la transformación dada por la ecuación (5.3.3) puede expresarse como en la ecuación (5.3.4):

$$Y = GX \tag{5.3.4}$$

donde G es una matriz de coeficientes de transformación. El objetivo de esta transformación es la obtención de nuevos atributos con algún significado, siendo éste el que determina el método para la obtención de la matriz G. Adicionalmente, algunas de las nuevas variables podrán eliminarse de forma directa. El primer componente principal es la variable que explica la mayor heterogeneidad entre los patrones (es decir, la que tiene mayor varianza).

El segundo componente principal es la variable que es más heterogénea después del primer componente y que no está correlacionada con él. El tercer componente principal no está correlacionado ni con el primero ni con el segundo, y tiene la tercera varianza mayor, y así sucesivamente. Existen dos casos límites poco probables en situaciones reales. El primero es que los datos originales no estén correlacionados, en cuyo caso habrá tantas componentes principales como componentes originales. La segunda situación se tiene cuando la redundancia entre las variables de entrada es total, existiendo en este caso un único componente principal.

La ordenación de los componentes principales en función de la varianza tiene ventajas evidentes a la hora de seleccionar una dimensionalidad particular para los patrones transformados.

La obtención de la matriz de transformación puede formularse como un problema de maximización de la varianza en el espacio P' con la restricción de mantener la ortogonalidad de G. Una técnica adecuada es la utilización de los multiplicadores de Lagrange.

En resumen, el uso de componentes principales para transformar el conjunto original de patrones se justifica por dos razones. En primer lugar, el análisis puede resaltar ciertos aspectos o propiedades sobre los datos que no era posible distinguir en los datos originales. En segundo lugar, puede reducir la información original en un conjunto más reducido de variables. Para el caso particular del análisis de contingencias, la reducción de la dimensión del espacio de entrada es evidente a partir de los datos mostrados en la tabla 5.1. Según esta tabla, tomando una transformación de vectores de 6 componentes obtenidos mediante un análisis por componentes principales de un conjunto original de 20 componentes, es posible explicar el 99,9 % de la variabilidad de los mismos. En el caso de la red IEEE-118, el mismo porcentaje se consigue tomando las 18 componentes principales.

Tabla 5.1: Porcentaje total de explicación de la variación del conjunto de patrones originales de la red IEEE-14, tomando diferentes componentes principales

Número de Componentes	Porcentaje de Variación explicado
3	90
4	92
5	93
6	99,9
8	99,95
9	99,98
12	99,99

5.4. Evaluación numérica y ordenación de contingencias con redes neuronales alimentadas hacia delante y aprendizaje supervisado

El enfoque funcional del análisis de contingencias consiste en obtener una evaluación numérica de la peligrosidad de cada contingencia particular. Esto permite la posterior ordenación de contingencias en función de dicho valor de peligrosidad. Esta ordenación o*ranking* permite al sistema estar preparado para hacer frente a las averías más peligrosas. Normalmente, esta evaluación numérica se obtiene a partir del cálculo de un índice de prestación predefinido.

Las redes neuronales artificiales de topología alimentada hacia delante y aprendizaje supervisado, tales como el Perceptrón Multicapa con *Backpropagation* o las redes de Funciones Base Radiales se caracterizan por su capacidad para aproximar funciones no lineales, a partir de un proceso de aprendizaje que puede ser relativamente largo, pero con respuesta en tiempo real en su modo de operación. En este trabajo, se aplican los dos paradigmas mencionados al cálculo de los índices PIp, i y PIv, i para cada una de las contingencias simples de la red IEEE-14.

Por otra parte, el uso directo de patrones de entrada formados por todos los flujos de potencia del sistema produce una alta dimensionalidad del espacio de entrada, lo que dificulta la convergencia del proceso de aprendizaje. Esta limitación se resuelve en nuestro caso, mediante la formación de un reducido número de variables de entrada a partir del análisis de

componentes principales descrito anteriormente. En definitiva, para el análisis de contingencias de la red estándar IEEE-14, descrito a continuación, han sido implementadas 20 redes neuronales de cada paradigma, habiéndose reducido el patrón original de 40 componentes de flujo a uno de seis componentes. En las siguientes subsecciones se muestran los resultados de simulación obtenidos para el cálculo de cada $PI_{p,i}$ mediante redes Perceptrón Multicapa y Funciones Base Radiales (RBF).

5.4.1. Evaluación numérica de contingencias con Perceptrón Multicapa y *Backpropagation*

Se ha implementado una red Perceptrón Multicapa para el cálculo de cada uno de los índices de prestación $PI_{p,i}$ de la red IEEE-14. La estructura de las redes es la siguiente: capa de entrada de seis neuronas, cada una de ellas correspondiendo a una de las componentes principales obtenidas; capa intermedia de veinte neuronas con función de activación tangente hiperbólica; capa de salida con una neurona con función de activación lineal, correspondiente a un PIp, i particular. El conjunto de patrones generado ha sido dividido en un conjunto de entrenamiento (900 patrones) y un conjunto de prueba (100). El algoritmo de aprendizaje ha sido el de Levenberg-Marquardt [Hagan and Menhaj, 1994] y la estrategia de aprendizaje usada ha sido la conocida como validación cruzada ([Tibshirani, 1996]) (del inglés *cross-validation*). Los mejores resultados han sido obtenidos para un proceso de treinta épocas de entrenamiento. La tabla 5.2 muestra los errores relativos medios (en tanto por ciento) para las fases de entrenamiento y prueba para las contingencias con PIp, i mayor que cero para algún patrón de entrada. Aunque los errores de prueba para las contingencias 2 y 8 son mayores en media al 5 %, hay que hacer notar que esto se debe a la pequeña cantidad de patrones que producen, para estas contingencias, un valor del PI mayor que cero. Para otras contingencias con un número mayor de patrones con valores altos de su PI los resultados presentan valores mucho más bajos del error. La figura 5.3 muestra de manera gráfica los valores real y calculado del $PI_{p,i}$ para el conjunto de patrones de prueba (la contingencia 1 corresponde a un fallo en la línea que une los buses 1 y 2 de la red).

La tabla 5.3 muestra la ordenación de contingencias realizada por el sistema neuronal junto con la ordenación real para los patrones de prueba. Es necesario hacer notar que las 39 contingencias que se muestran en esta tabla corresponden a diferentes contingencias

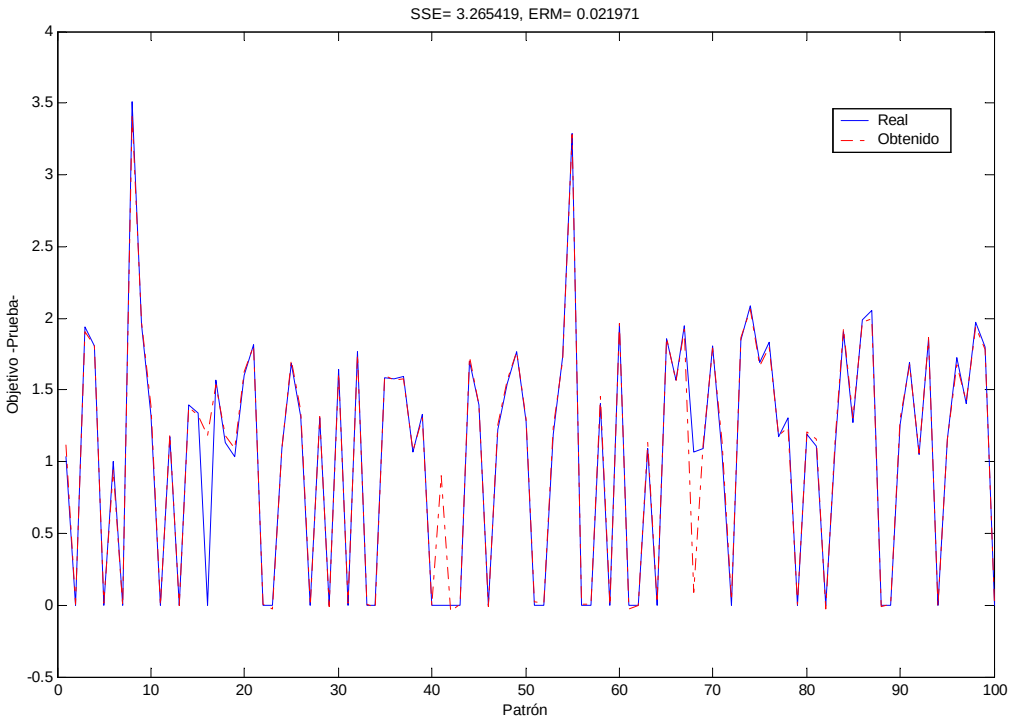


Figura 5.3: Representación gráfica de los valores real y calculado del índice de prestación $PIp, 1$ para el conjunto de los patrones de prueba.

Tabla 5.2: Errores relativos medios en tanto por ciento (MAPE) para los índices de prestación de las contingencias problemáticas de la red estándar IEEE-14.

Contingencia	MAPE(%) - Entrenamiento	MAPE(%) - Prueba
1	0.6226	1.21
2	0.1519	5.041
8	0.03	6.18
10	2.53	2.05
13	0.18	0.28
16	0.56	0.17

para varios estados diferentes de la red eléctrica. Con este experimento queremos probar la habilidad del sistema neuronal para mantener el orden de cualquier contingencia para cualquier rango de valores de los índices de prestación. Obviamente, en una situación real donde sólo es necesario ordenar las contingencias del estado actual de la red, el porcentaje de contingencias correctamente ordenadas es mucho mayor.

Del análisis de la tabla 5.3 se puede concluir que las contingencias más peligrosas han sido clasificadas correctamente: el orden de las 3 primeras se mantiene y las 7 primeras, con algún pequeño error, también. Los resultados peores se presentan en la clasificación de los patrones con órdenes 20, 29, 37 y 38, que producen unos resultados totalmente erróneos.

5.4.2. Evaluación numérica de contingencias con redes de funciones base radiales

Las redes de Funciones Base Radiales son ampliamente referenciadas en aplicaciones de aproximación de funciones, especialmente por su capacidad de aprendizaje local. En estas redes, cada intervalo de la función se aproxima por un conjunto limitado de neuronas especializadas, lo que, al menos teóricamente, permite un mejor ajuste de la función. Por otra parte, una vez resuelto el problema de la distribución de neuronas en el espacio de definición de la función, el cálculo del conjunto óptimo de pesos de la red se realiza de forma rápida mediante la resolución de un sistema de ecuaciones lineal. Esta última característica evita la existencia de un proceso de aprendizaje probabilístico que, además de ser largo en el tiempo, puede no llegar a la solución óptima. En este apartado, se reproduce la aplicación de este tipo de redes para la aproximación de los Índices de Prestación. Como caso de estudio, se calcula el Índice de Prestación PIp, i de cada posible contingencia de la red IEEE-14.

5.4 Evaluación numérica y ordenación de contingencias con redes neuronales alimentadas hacia delante y aprendizaje supervisado

Tabla 5.3: Ordenación de las contingencias según el valor de su índice de prestación. Las columnas con título OR representan la ordenación real de la contingencia, OO la ordenación obtenida por las redes MLP, OR-OO la diferencia (en número de posiciones) entre la ordenación real y obtenida, PIR representa el índice de prestación real, PIO el índice de prestación obtenido. La última columna (PIR - PIO) muestra la diferencia entre los índices de prestación real y obtenido para cada contingencia. Las contingencias mostradas corresponden a diferentes estados de la red eléctrica.

OR	OO	OR - OO	PIR	PIO	PIR - PIO
1	1	0	11.3518	11.2070	0.1448
2	2	0	10.9391	10.9489	-0.0098
3	3	0	10.6498	10.7936	-0.1438
4	6	-2	10.6238	10.5003	0.1236
5	4	1	10.5042	10.5862	-0.0820
6	7	-1	10.4816	10.4827	-0.0011
7	5	2	10.4726	10.5386	-0.0660
8	9	-1	5.4933	5.4461	0.0472
9	10	-1	5.3665	5.3994	-0.0329
10	11	-1	4.9147	4.9165	-0.0018
11	12	-1	4.7929	4.7777	0.0151
12	13	-1	4.7443	4.7585	-0.0141
13	15	-2	4.7190	4.7091	0.0099
14	14	0	4.7046	4.7392	-0.0346
15	16	-1	3.1873	3.1977	-0.0103
16	20	-4	2.3398	2.3133	0.0265
17	19	-2	2.3306	2.3214	0.0091
18	39	-21	2.2186	-1.7914	4.0100
19	21	-2	2.1799	2.1966	-0.0167
20	38	-18	2.1018	0.0745	2.0274
21	22	-1	2.0735	2.0655	0.0080
22	23	-1	2.0438	2.0509	-0.0071
23	24	-1	1.9861	2.0080	-0.0219
24	18	6	1.9734	2.7148	-0.7415
25	26	-1	1.9297	1.8588	0.0709
26	25	1	1.9244	1.9113	0.0131
27	27	0	1.2476	1.2499	-0.0023
28	31	-3	1.1315	1.1014	0.0301
29	37	-8	1.1227	0.3031	0.8197
30	28	2	1.1157	1.1151	0.0006
31	29	2	1.1134	1.1138	-0.0004
32	30	2	1.1024	1.1030	-0.0006
33	32	1	1.1003	1.1006	-0.0002
34	33	1	1.0844	1.0899	-0.0055
35	34	1	1.0683	1.0671	0.0012
36	35	1	1.0485	1.0484	0.0001
37	17	20	1.0318	3.0806	-2.0488
38	8	30	1.0171	6.8917	-5.8747
39	36	3	1.0147	0.5912	0.4235

Del mismo modo que el caso de la aplicación del MLP, se ha utilizado una red neuronal para cada contingencia. Cada una de estas redes cuenta con seis entradas (correspondientes a las seis componentes principales) y una salida (el valor del PIp, i correspondiente a la contingencia en cuestión). El número de neuronas intermedias, cuyas funciones de activación son las funciones radiales, varía entre 68 y 215 dependiendo de la contingencia tratada. Este número de neuronas intermedias y su distribución en el espacio de entrada han sido calculados a partir del algoritmo de entrenamiento de Mínimos Cuadrados Ortogonales para Redes de Funciones Base Radiales ([Chen et al., 1991]). En este algoritmo, los centros donde se ubican las neuronas se estiman mediante un problema de regresión utilizando mínimos

cuadrados ortogonales. En cada paso del algoritmo, se selecciona un subconjunto de centros para las neuronas de entre todos los posibles, de forma que se maximice la varianza de la salida deseada. Este algoritmo presenta dos grandes ventajas frente a otros disponibles para las redes de funciones base radiales: aporta un método sistemático y eficiente para determinar los centros de las neuronas de la red; y elimina los problemas relacionados con la creación de muchas neuronas innecesarias y de condicionamiento numérico deficiente de los métodos basados en la selección aleatoria de centros.

Otro parámetro importante en el diseño de la red es el denominado *spread*, que determina, para cada neurona de base radial, qué distancia debe tener un vector de entrada a su centro para que ésta produzca una salida de 0,5. Un valor alto de este parámetro hace que exista un área relativamente grande alrededor del vector de entrada donde la neurona responde con valores cercanos a uno. Esto permite que varias neuronas puedan responder con salidas mayores que cero a un mismo vector de entrada. Por el contrario, un valor muy pequeño del parámetro *spread* hace que ante un vector de entrada particular sólo una neurona responda con una salida mayor que cero. Aunque existen varios métodos heurísticos que tratan de buscar un valor adecuado para este parámetro, son en general muy dependientes de la función que se está aproximando. En este trabajo, se ha buscado un valor óptimo realizando un estudio estadístico de los resultados para diversos valores del parámetro. El valor más pequeño considerado viene dado por la expresión $|mean(d) - 2std(d)|$ y el mayor valor es $|mean(d) + 2std(d)|$, donde $mean(d)$ es el valor medio de las distancias entre cada dos vectores de entrada y $std(d)$ la desviación típica. Este proceso lleva a un valor óptimo del parámetro para cada contingencia analizada. Sin embargo, se ha podido comprobar que estos valores son muy similares para las diversas contingencias analizadas, por lo que en el diseño del sistema no es necesario realizar el estudio por cada contingencia.

La tabla 5.4 muestra los errores relativos medios (en tanto por ciento) para las fases de entrenamiento y prueba para las contingencias con PIp, i mayor que cero para algún patrón de entrada. La figura 5.4 muestra de manera gráfica los valores real y calculado para el $PIp, 1$ (contingencia 13 correspondiente a un fallo en la línea que une los buses 6 y 13).

La tabla 5.5 muestra la ordenación de contingencias realizada por el sistema neuronal junto con la ordenación real para los patrones de prueba. Al igual que en la tabla 5.3, las contingencias que se muestran corresponden a diferentes contingencias para varios estados

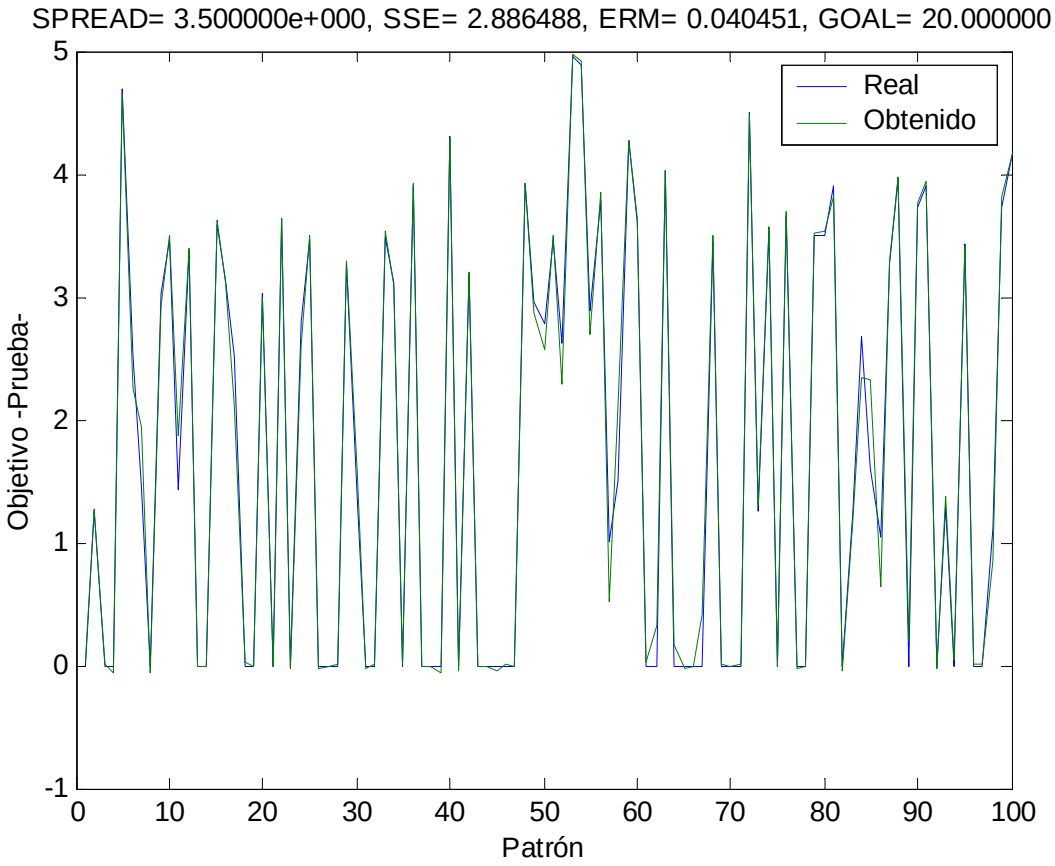


Figura 5.4: Representación gráfica de los valores real y calculado del índice de prestación $PIp, 1$ para el conjunto de los patrones de prueba.

Tabla 5.4: Errores relativos medios en tanto por ciento (MAPE) para los índices de prestación de las contingencias problemáticas de la red estándar IEEE-14.

Contingencia	MAPE(%) - Entrenamiento	MAPE(%) - Prueba
1	3.6	4.2
2	3.4	5.7
8	1.2	8
10	1.8	10
13	3.9	4.0
16	2.6	3.2

diferentes de la red eléctrica

Como muestra la tabla 5.5, las primeras 9 contingencias han sido ordenadas perfectamente. Excepto para las contingencias con posiciones reales $27, 30, 33, 34$ y 37, la clasificación es casi perfecta, obteniendo una clasificación que sólo desplaza el orden de la contingencia en una o dos posiciones.

El resultado de la ordenación de la tabla 5.3 es peor que la ordenación de la tabla 5.5. Aunque los resultados en cuanto a errores relativos del MLP son mejores que el caso de las redes RBF, la ordenación es más fiable en el caso de las redes de funciones base radiales. La justificación de este resultado aparentemente contradictorio se debe a que las redes de funciones base radiales implementadas tienen mayor dificultad en aproximar la función correspondiente a los P.I. cuando éstos tienen un valor cero o próximo a cero (contingencia nada o poco peligrosa), lo que empeora los resultados globales sobre el conjunto de patrones de prueba. Sin embargo, la aproximación funcional de las RBF es mejor cuando los valores a predecir son mucho mayores que cero.

5.5. Visualización de contingencias con mapas Auto-Organizativos de Kohonen

Los Mapas Auto-Organizativos de Kohonen se caracterizan por su capacidad para clasificar un conjunto complejo de patrones de manera no supervisada, extrayendo criterios de clasificación no obvios ni expresados de manera explícita. Esta clasificación se lleva a cabo mediante la distribución de un espacio de entrada V_I de alta dimensión en un espacio de salida V_O de menor dimensión, preservándose las relaciones topológicas existentes entre los

Tabla 5.5: Ordenación de las contingencias según el valor de su índice de prestación. Las columnas con título OR representan la ordenación real de la contingencia, OO la ordenación obtenida por las redes RBF, OR-OO la diferencia (en número de posiciones) entre la ordenación real y obtenida, PIR representa el índice de prestación real, PIO el índice de prestación obtenido y la última columna la diferencia entre estos dos últimos índices.

OR	OO	OR - OO	PIR	PIO	PIR - PI
1	1	0	11.9711	11.6672	0,3039
2	2	0	11.5060	11.5566	-0,0505
3	3	0	11.3253	11.4034	-0,0782
4	4	0	11.0233	11.1157	-0,0924
5	5	0	10.9274	11.0231	-0,0957
6	6	0	10.8771	10.9453	-0,0682
7	7	0	10.7677	10.8477	-0,0799
8	8	0	5.6128	5.3694	0.2433
9	9	0	5.2736	5.2871	-0,0135
10	11	-1	5.2355	5.2264	-0.0011
11	10	1	5.2292	5.2366	0.0028
12	13	-1	5.0543	5.0638	-0,0676
13	12	1	5.0513	5.1219	-0.0125
14	14	0	4.9673	4.9780	-0,0107
15	15	0	3.3668	3.6850	-0.3182
16	16	0	3.3159	2.9007	0.4152
17	18	-1	3.2500	2.7756	0.5163
18	19	-1	3.1887	2.7337	0.3684
19	17	2	3.1613	2.8203	0.3856
20	20	0	3.1045	2.4106	0.6939
21	21	0	2.2791	2.2456	0.0334
22	24	-2	2.2318	1.8301	0.5396
23	23	0	2.1897	1.8385	0.3512
24	25	-1	2.1553	1.6922	0.4681
25	26	-1	2.1539	1.6872	0.4818
26	27	-1	2.1279	1.6721	-0.0846
27	22	5	2.1165	2.2125	0.2864
28	29	-1	1.2476	1.5355	-0.0203
29	31	-2	1.2183	1.2679	0.2545
30	37	-7	1.1420	0.6053	0.1275
31	34	-3	1.1368	0.9638	-0.4446
32	33	-1	1.1352	1.0109	-0.1646
33	30	3	1.1244	1.2998	0.5191
34	28	6	1.1237	1.5814	-0.4118
35	35	0	1.0864	0.8415	0.2449
36	38	-2	1.0553	0.0446	0.2186
37	32	5	1.0394	1.0145	0.0285
38	36	2	1.0318	0.8367	0.9872
39	39	0	1.0043	0.0318	0.9725

patrones de entrada. El espacio de salida viene constituido por un conjunto de neuronas ordenadas en un plano o una línea, en el cual se define una función de vecindad. Tras el proceso de aprendizaje, un vector de entrada $x = (x_1, ..., x_n)$ activará la neurona i del espacio de salida cuyo vector de pesos $w_i = (w_{i1}, ..., w_{in})$ tenga menor distancia al vector x. De esta manera, el vector w_i podrá ser considerado como el prototipo de la región del espacio de entrada cuyos vectores activan a la neurona i. Finalmente, dos vectores de entrada similares según la relación definida en V_I, activarán la misma neurona o dos neuronas cercanas en el espacio de salida (figura 5.5).

Las características brevemente descritas de los Mapas Auto-Organizativos resultan en principio prometedoras para la implementación de un sistema de monitorización visual de

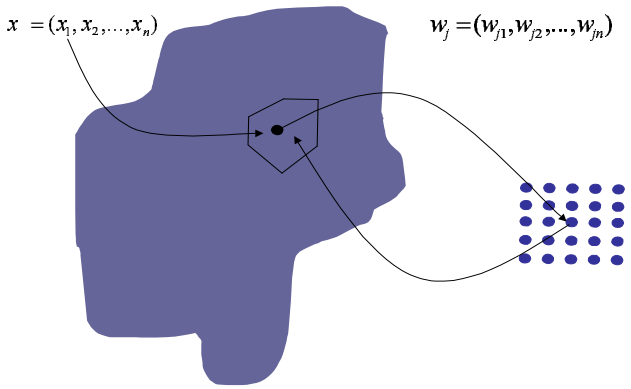


Figura 5.5: Representación esquemática de la distribución de un espacio n-dimensional de entrada en un espacio bidimensional mediante un Mapa Auto-Organizativo de Kohonen.

contingencias, ya que en este entorno, es a veces más interesante y práctico poder visualizar de manera directa la posición relativa en cuanto a peligrosidad del estado actual en un ranking de contingencias que conocer el valor exacto del Índice de Prestación que define esa peligrosidad. Para analizar de manera experimental esta posibilidad, en primer lugar se ha desarrollado una SOM bidimensional destinada a la visualización de contingencias, y se ha probado con la red IEEE-14 [García-Lagos et al., 2001]. A partir del análisis de esta aplicación dedujimos que el uso de redes SOM lineales parecía especialmente indicado en este contexto, por lo que, en segundo lugar, se ha desarrollado una SOM lineal que ha sido probada con la red IEEE-118. A continuación se describen de manera detallada ambas implementaciones, discutiendo sus prestaciones y ventajas.

5.5.1. SOM bidimensional para la visualización de contingencias

En este apartado se analiza la aplicación de una SOM bidimensional de 10x10 neuronas a la clasificación de los patrones generados previamente para la IEEE-14. Estos patrones están compuestos por los Flujos de Potencia Activa de cada línea de la red. De manera similar, se ha repetido el estudio usando como patrones los Flujos de Potencia Reactiva, aunque no se reproducen aquí por presentar análogos resultados. El algoritmo de adaptación de pesos

para la SOM ha sido el indicado por la ecuación (5.5.1):

$$\vec{w}_i(t+1) = \vec{w}_i(t) + l_r h_{iv}(\vec{x} - \vec{w}_i(t)) \tag{5.5.1}$$

donde $\vec{x}$ es un vector de entrada, $\vec{w}_i$ es el vector de pesos de la neurona i, y $h_{iv}()$, la función de vecindad, determina el incremento del peso de cada neurona como una función de su proximidad a la neurona ganadora para el patrón $\vec{x}$. En nuestro caso, el área de vecindad viene dada por un cuadrado centrado en la neurona ganadora, cuyo lado decrece hasta cero a lo largo del entrenamiento. lr es la razón dinámica de aprendizaje, la cual evoluciona a lo largo del proceso de entrenamiento según la ecuación (5.5.2):

$$l_r(t) = \frac{l_{r0}}{(1 + \frac{c \cdot t}{nn})} \tag{5.5.2}$$

siendo $lr0$ la razón de aprendizaje inicial (0.3, en nuestro caso), c una constante (0.2), t la iteración actual y nn el número de neuronas de la red [Cottrell and Letremey, 1995] [Haykin, 1994].

Para cada contingencia, el conjunto de patrones de flujos de potencia activa disponible se ha dividido previamente según el valor real de su $PI_{p,i}$. De esta manera, se han establecido siete intervalos de peligrosidad, asociándose cada uno de ellos a un símbolo para su visualización en el mapa: punto, que representa un $PI_{p,i}$ de valor cero, círculo, signo más, cuadrado, triángulo invertido, rombo, pentágono y hexágono, respectivamente, estados con un $PI_{p,i}$ en (0,1.5], (1.5, 3], (3, 4.6], (4.6, 6.1], (6.1, 7.6], (7.6, 9], y (9,inf). El proceso de entrenamiento se ha realizado para una única SOM a partir de los patrones de entrenamiento de flujos de potencia activa disponibles. La obtención de cada figura se realiza mediante el procedimiento siguiente:

1. Se entrena una única red neuronal. Es necesario advertir de que al tratarse de un entrenamiento no supervisado, este proceso realiza exclusivamente una distribución de los patrones de entrada, sin relación en principio con ninguna contingencia particular.

2. Para cada contingencia considerada en el análisis se crea una copia idéntica de la red entrenada, que será utilizada de forma exclusiva para esa contingencia. Se dispone, por tanto, de N redes idénticas, donde N es el número de contingencias que se desean considerar.

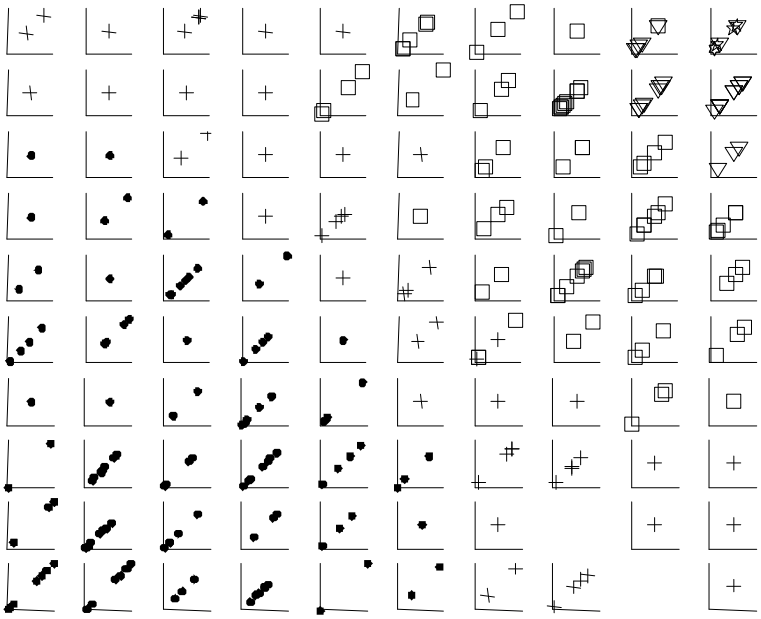


Figura 5.6: Representación de la clasificación de los patrones obtenida por la red de Kohonen para la contingencia 1. Los símbolos de la figura se han seleccionado en función del índice de prestación de cada patrón. Cada subgráfico o rectángulo representa una neurona del mapa. El rectángulo de la esquina superior izquierda es la neurona 1, el rectángulo de la esquina superior derecha es la neurona 10 y el rectángulo de la esquina inferior derecha corresponde a la neurona 100. Los huecos de la figura representan neuronas a la que no pertenece ningún patrón. El mapa de Kohonen representado es de dimensión 10x10.

3. Los patrones de entrada a todas las redes son los mismos. Cada patrón (flujos de potencia activa) representa el estado de la red para la topología base y un nivel de carga determinado antes de la ocurrencia de cualquier contingencia.

4. Para cada patrón se dispone del valor $PI_{p,i}$ para cada contingencia.

5. Para el SOM de una contingencia particular i:

   a) Se presenta el patrón j-ésimo a la red y se obtiene la neurona ganadora g_j.

   b) En el rectángulo g_j, correspondiente a la neurona g_j ganadora, se representa un punto con un atributo obtenido en función del valor de $PI_{p,i}$.

La respuesta de la SOM es distinta para cada contingencia analizada, produciendo para cada una de ellas una distribución particular de las zonas asociadas a cada nivel de

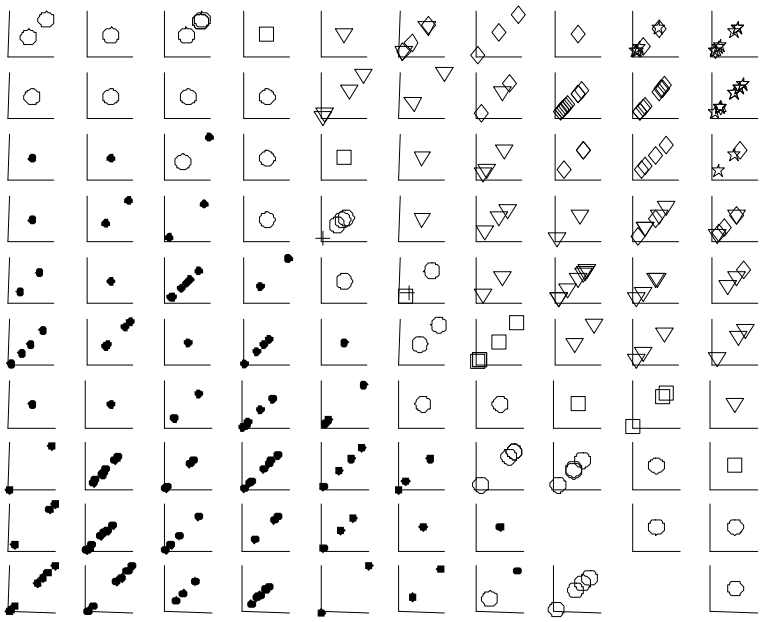


Figura 5.7: Representación de la clasificación de los patrones obtenida por la red de Kohonen para la contingencia 13.

peligrosidad. Los resultados de simulación muestran claramente que la organización final de la red incorpora el criterio del nivel de peligrosidad de cada contingencia. Como puede apreciarse en las figuras 5.6 y 5.7, que representan la distribución de patrones para las contingencias 1 y 13, respectivamente, las clases se ordenan de manera continua en orden creciente de su $PI_{p,i}$.

Los resultados obtenidos muestran la alta aplicabilidad del paradigma en dos importantes operaciones relacionadas con el análisis de contingencias:

En primer lugar, la presentación del estado actual del sistema al mapa asociado a cada contingencia permitirá la rápida visualización del nivel de seguridad de ese estado actual. En este sentido, la figura 5.8, muestra la distribución en el mapa elaborado para la contingencia 10. Los patrones señalados corresponden a estados no utilizados en el entrenamiento y su PI puede verse en la tabla 5.6. En la figura 5.8 se ha representado con un círculo negro la neurona que cada patrón ha activado. Junto al círculo aparece el número del patrón que ha activado esa neurona (ver tabla 5.6). Los patrones 1, 7, 8, 9 y 10, que tienen todos un $PI_{p,10}$ cero, activan neuronas cuyo atributo es un punto, indicando que presentan estados

que ante la ocurrencia de la contingencia 10 no suponen peligro. Los patrones 2, 3 y 4 han activado las neuronas 3 y 14, correspondiente a una zona marcada para patrones que tienen un PI en el intervalo (0, 1.5], etc. Como puede comprobarse, todos los patrones se han clasificado correctamente (es decir, han activado una neurona marcada con el símbolo correspondiente al intervalo al que pertenece su PI). En algunos casos, las neuronas activadas tienen atributos pertenecientes a dos intervalos diferentes (patrones 12 y 14), pero estas neuronas se encuentran siempre en la frontera entre zonas del mapa correspondiente a los dos intervalos.

En segundo lugar, los mapas obtenidos pueden ser utilizados como herramientas para el seguimiento de la evolución del sistema hacia estados de mayor o menor peligrosidad y a la predicción de esta evolución respecto a cada contingencia. En este sentido, la figura 5.8 también muestra la evolución del estado del sistema para un segmento particular del conjunto de patrones de prueba utilizados. La figura 5.9 muestra los valores PI para la contingencia 10, y la secuencia de 16 puntos correspondientes a los estados que han activado las neuronas de la figura 5.8. Se aprecia claramente que la evolución del sistema hacia estados de mayor peligrosidad va aparejada a la evolución del mapa hacia regiones ascendentes a lo largo de la diagonal secundaria, es decir hacia regiones asociadas a una mayor peligrosidad de la contingencia analizada.

Justamente esta ordenación de las diferentes clases o regiones a lo largo de una diagonal de la SOM nos lleva a plantear la hipótesis de que la relación entre patrones puede ser proyectada sobre un mapa lineal de Kohonen. Esta posibilidad proporcionaría una mayor claridad a la monitorización de contingencias descrita anteriormente, ya que a un operador le resultará más fácil comparar posiciones activadas en líneas que en planos, y podrán ser monitorizadas a la vez muchas más contingencias, ya que obviamente, en una pantalla caben más líneas que planos. La comprobación de esta hipótesis se describe en la siguiente sección [García-Lagos et al., 2002].

5.5.2. SOM lineal para la visualización de contingencias

Una SOM lineal corresponde a un única fila de neuronas. Cada neurona tiene únicamente dos vecinas directas. La función de vecindad viene dada, por un segmento centrado en la neurona, en vez de por un cuadrado. Los algoritmos de entrenamiento han sido adaptados,

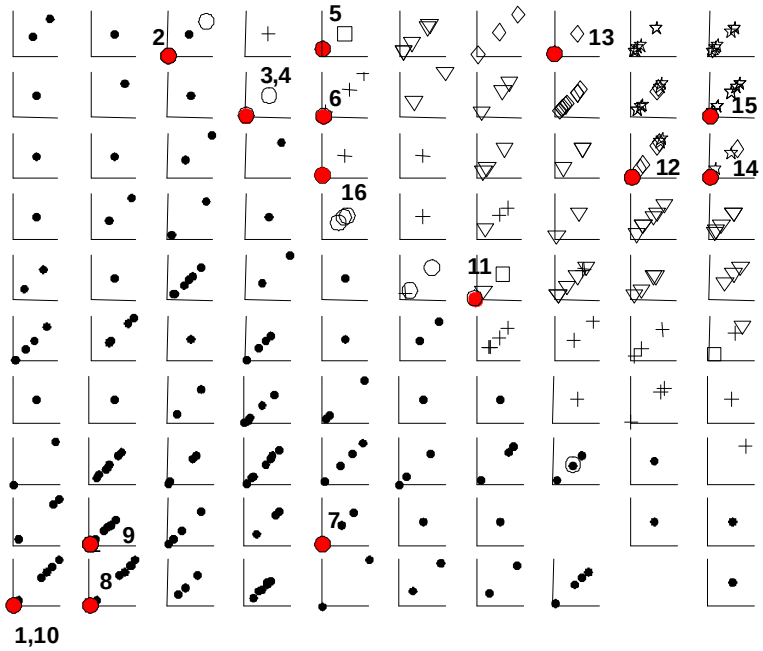


Figura 5.8: Mapa de Kohonen para la contingencia 10 y las neuronas activadas para patrones con diferentes PI no utilizados durante el entrenamiento.

Tabla 5.6: Identificación de las neuronas activadas y el valor correspondiente PI correspondientes a la figura 5.8.

Número	Neurona Activada	PI
1	91	0
2	3	1.2253
3	14	1.4415
4	14	1.3438
5	5	3.6199
6	15	2.4723
7	85	0
8	92	0
9	82	0
10	91	0
11	47	3.9644
12	29	8.9394
13	8	7.0063
14	30	9.7227
15	20	10.4199
16	35	2.6166

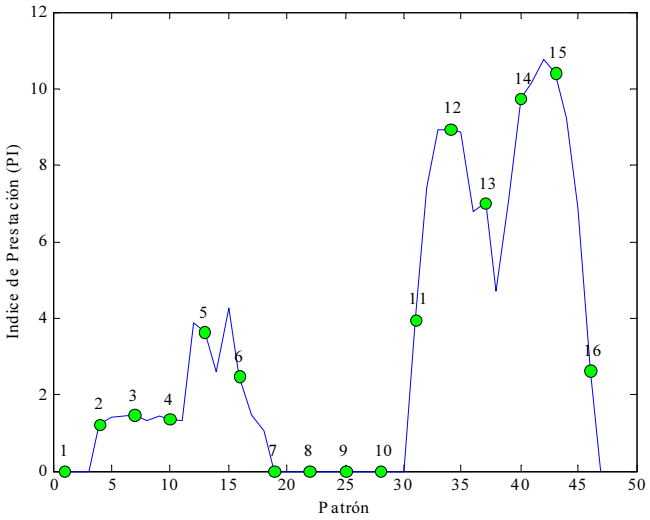


Figura 5.9: Representación de los índices de prestación de la contingencia 10 para varios patrones.

y se han utilizado SOM de 10 neuronas.

Como caso de estudio para la visualización de contingencias mediante SOMs lineales hemos utilizado la red estándar IEEE-118. Los patrones, generados como se describe en la sección 5.3, han sido nuevamente sometidos a un análisis en componentes principales obteniendo en este caso 18 componentes con las que es posible explicar el 99.99 % del la funcionalidad del sistema. Al igual que en los casos anteriores, el conjunto completo de patrones ha sido dividido en un conjunto de entrenamiento y otro de prueba. Las simulaciones muestran que en efecto, la ordenación de los patrones se realiza de manera ordenada a lo largo de cada línea variando desde estados con bajo PI a estados con muy alto PI. La figura 5.10 muestra la distribución obtenida para las contingencias 3, 4, 7, 8, 9, 11, 14, 17, 26 y 33.

La figura 5.11 muestra la situación de peligrosidad de un determinado patrón de entrada con respecto a las contingencias 3, 4, 7, 8, 9, 11, 14, 17, 26 y 33. De esta figura puede deducirse que el patrón presentado representa un estado de peligrosidad alto para las contingencias 3 y 4, un nivel medio de peligrosidad para las contingencias 9, 14 y 33, un nivel bajo de peligrosidad para las contingencias 7, 11, 17 y 26, y no es peligroso para las contingencias 8 y 26.

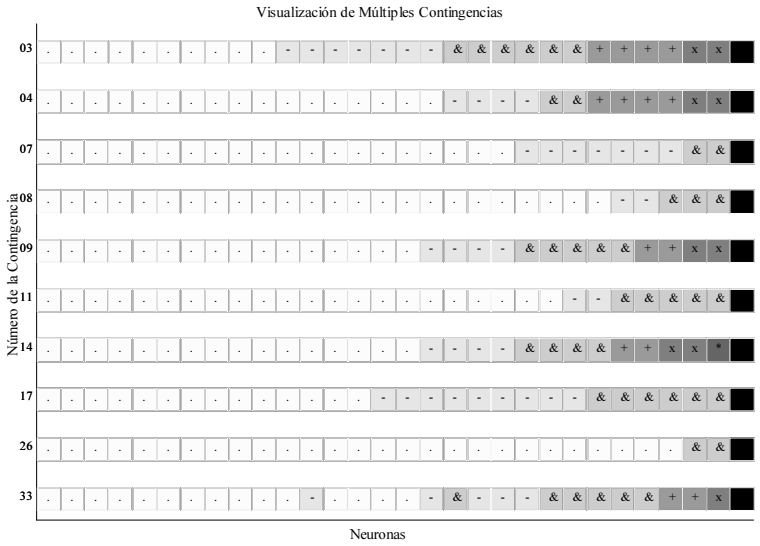


Figura 5.10: Visualización de varios Mapas de Kohonen lineales en una misma figura. Se han representado 10 Mapas de Kohonen lineales de 1 x 30 neuronas cada uno. Cada fila de la figura representa el Mapa de Kohonen visualizado para cada contingencia particular. Las contingencias visualizadas son la 3, 4, 7, 8, 9, 11, 14, 17, 26 y 33 de la red IEEE-118.

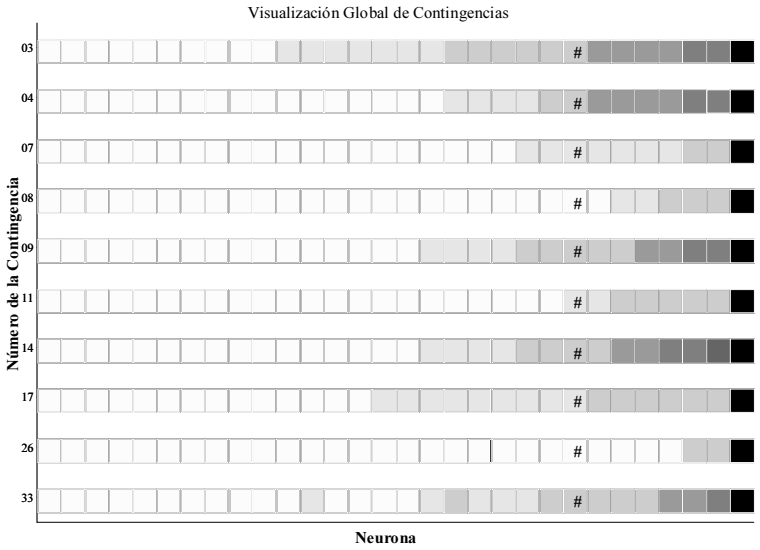


Figura 5.11: Representación del nivel de peligrosidad que ha producido un determinado patrón presentado a la red de Kohonen Lineal. La neurona ganadora es la neurona 7, pero su nivel de peligrosidad es distinto para cada contingencia, como muestra el correspondiente atributo de la neurona 7 en cada fila de la figura.

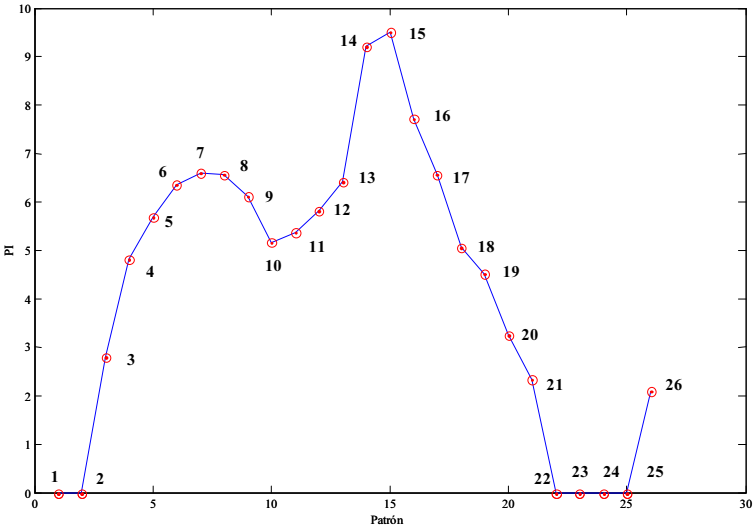


Figura 5.12: PI, para la contingencia 3, de 26 patrones no usados durante el entrenamiento del SOM.

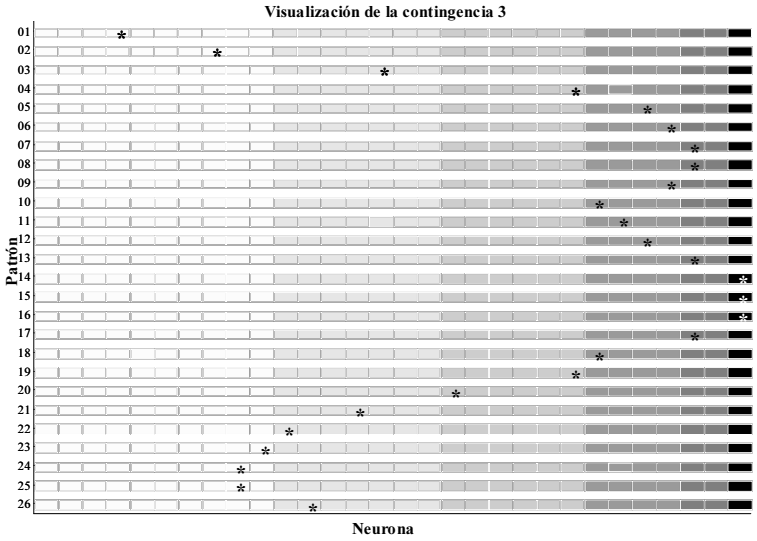


Figura 5.13: Respuesta del SOM lineal de la contingencia 3 a cada uno de los patrones de la figura 5.12.

La aplicación del SOM a la monitorización *on-line* de la evolución del estado de la red se muestra en las figuras 5.12 y 5.13. En la primera de ellas, se representan los PI de 26 patrones de entrada no utilizados en la fase de entrenamiento de la red. En la figura 5.13 se muestra la respuesta del SOM para la contingencia 3 a cada uno de estos patrones. La fila i-ésima corresponde a la respuesta del SOM ante el patrón i. Como puede observarse, la red ha respondido ante cada patrón activando la neurona correcta según el valor del PI correspondiente, lo que prueba su utilidad como herramienta para la monitorización de la evolución del sistema ante contingencias.

5.6. Conclusiones

Este capítulo ha estado dedicado al estudio de la aplicabilidad de distintos paradigmas neuronales a la tarea de Análisis de Contingencias en un sistema de energía eléctrica. En este sentido, hemos estudiado dos aspectos del Análisis de Contingencias. Por un lado, aparece la tarea de evaluar numéricamente la gravedad de cada una de las contingencias que pueden generarse a partir del estado actual del sistema, lo que nos permitirá establecer una ordenación de contingencias en orden creciente de gravedad (*Contingency Ranking*). Por otro lado, aparece la tarea de monitorizar de forma visual y gráfica la gravedad de dichas contingencias. Ambos aspectos presentan características que hacen en principio aconsejable el uso de determinados paradigmas neuronales en su solución. Así, en cuanto a la evaluación numérica, se trata de aproximar una función, el correspondiente Índice de Prestación, cuyo cálculo para cada estado de la red exige la utilización de métodos numéricos computacionalmente costosos y prohibitivos para operaciones en tiempo real. Sin embargo, para esta función, podemos obtener en modo simulación un amplio conjunto de patrones entrada-salida. En consecuencia, las redes neuronales alimentadas hacia adelante con aprendizaje supervisado, cuyo aprendizaje a partir de ese conjunto de patrones puede ser lento, pero cuyo funcionamiento en modo de operación es en tiempo real, parecen perfectamente recomendables. En cuanto a la monitorización visual, paradigmas como los Mapas Auto-Organizativos de Kohonen, por su capacidad para clasificar un conjunto de patrones complejos de acuerdo con criterios difícilmente explicitables por un operador, y de mostrar esa clasificación de manera gráfica y visualmente interpretable parecen igualmente recomendables. Nosotros hemos puesto a prueba las hipótesis anteriores para los paradigmas Perceptrón Multicapa y

Redes de Funciones Base Radiales en el caso de la evaluación numérica y para el paradigma SOM de Kohonen en el caso de la monitorización visual. En el primer caso, se han elegido como funciones para ser aproximadas los Índices de Prestación PIv y PIp de cada una de las posibles contingencias de la red estándar IEEE-14, generándose una red (MLP o RBF) para la evaluación de cada Índice de Prestación y cada contingencia. Los resultados experimentales muestran que las redes MLP constituyen una herramienta muy recomendable para la evaluación numérica de dichos Índices y para la obtención de la ordenación (*ranking*) de contingencias en tiempo real. Así, los resultados de los errores relativos para los patrones de prueba han sido muy aceptables, quedando comprendidos entre un mínimo de 0.17 % para la contingencia 16 y un máximo de 6.18 % para la contingencia 8. Esto se refleja en un*ranking* en el que la mayoría de los patrones han sido ordenados correctamente. En el caso de las redes RBF se ha obtenido una menor capacidad de generalización. Así, los errores para los patrones de prueba presentan un mínimo de 3.2 % para la contingencia 16 y un máximo de un 10 % para la contingencia 8. Sin embargo, hay que hacer notar que el empeoramiento de los resultados con respecto a los producidos por el MLP se debe a errores en la aproximación de patrones que tienen un PI cero. Esto se traduce en un índice de error para el *ranking* menor que en el caso del MLP. Por otra parte, los resultados muestran una mayor dependencia respecto a la variabilidad en el número de neuronas ocultas de la red, lo que merma en gran parte su utilidad ya que no se conoce un método eficiente que determine este número de forma exacta.

En el segundo caso, en primer lugar, se ha estudiado la capacidad de una SOM bidimensional para la clasificación de los estados del sistema en función de su peligrosidad. Una única red SOM de 10x10 ha sido entrenada con los patrones de aprendizaje, proporcionando una distribución particular de los patrones en clases de distinto nivel de peligrosidad para cada contingencia. La característica principal de estas distribuciones es que las distintas clases aparecen siempre ordenadas de menor a mayor peligrosidad a lo largo de una diagonal del mapa. Desde un punto de vista práctico, este hecho no sólo facilita enormemente la visualización de la peligrosidad de una contingencia particular, sino que permite el seguimiento de la evolución del sistema y la predicción de su acercamiento o no a un estado de peligro. Desde un punto de vista organizativo, permite plantear la hipótesis de que la estructura ideal para la SOM en este caso es la lineal. La hipótesis ha sido confirmada utilizando como

caso de estudio la red estándar IEEE-118. Al igual que en el caso bidimensional, a partir del entrenamiento de una única SOM lineal se obtiene una distribución distinta de los niveles de peligrosidad a lo largo de la línea para cada contingencia, pero en todos los casos estos niveles mantienen el orden de peligrosidad creciente. El uso de SOM lineales añade, a las ventajas ya señaladas para las bidimensionales, la facilidad de presentar en una única pantalla un gran número de contingencias (una por línea) y la facilidad de comparar la peligrosidad de distintas contingencias cuando éstas se distribuyen en una línea y no en un plano.

Capítulo 6

Conclusiones y líneas futuras

6.1. Conclusiones

En esta tesis doctoral se han estudiado varios paradigmas neuronales, tanto desde el punto de vista teórico como de su aplicación a la resolución de problemas reales complejos. Así, se han estudiado el Perceptrón Multicapa, las Redes de Funciones Base Radiales, los Mapas Auto-Organizativos de Kohonen, las redes recurrentes de Elman y el modelo neuronal de Hopfield. Cada uno de ellos ha sido aplicado a un problema complejo relacionado con tareas importantes en la Gestión de Sistemas de Energía Eléctrica (EMS).

Cuando la solución obtenida con los paradigmas simples no ha aportado una solución aceptable, se han propuesto paradigmas complejos que sí aportan soluciones adecuadas.

En todos los casos se ha hecho una revisión bibliográfica para mostrar el estado del arte.

A continuación enumeramos, agrupadas según la tarea particular dentro de la gestión de un EMS, la aportaciones principales de este trabajo.

Predicción de la demanda de energía eléctrica

Se ha propuesto un sistema neuronal complejo compuesto de dos etapas. En la primera etapa, mediante un mapa Auto-Organizativo de Kohonen, se realiza una clasificación de los perfiles de demanda diaria. Esta clasificación permite la agrupación de los mismos atendiendo a criterios de laboralidad, climáticos y sociales. El resultado de esta primera fase, que sólo debe realizarse una vez, es la división de la serie temporal en otras más regulares y, por tanto, más fácilmente predecibles.

La segunda etapa consiste en la aproximación de la función de consumo futuro mediante redes neuronales recurrentes de Elman. Aunque se han realizado numerosas pruebas con otros modelos, los mejores resultados se han obtenido con este paradigma neuronal. Para cada clase obtenida en la primera etapa, se han construido dos redes recurrentes de Elman, una para la predicción del perfil horario de carga para 24 horas, y otra para la predicción de la demanda de la hora $h+1$. El perfil de 24 horas permite conocer la demanda de energía eléctrica con un horizonte de 24 horas. La aportación del módulo de predicción de la hora $h+1$ es la incorporación al modelo de las condiciones meteorológicas, económicas o sociales que tienen lugar dentro de las 24 horas del día, que pasarían desapercibidas en el predictor del perfil, que utiliza como entradas los consumos del día anterior de la misma clase del que se quiere predecir. En ambos casos, las entradas de las redes neuronales son los consumos horarios consolidados y una predicción de la demanda integrada, eliminándose la influencia de errores acumulados en los datos de carga horaria.

En todos los casos se ha propuesto un proceso de reentrenamiento periódico (*on-line*) que hace que la influencia de la no-estacionaridad de la curva de demanda sea menor que en el caso de realizar la predicción con paquetes estadísticos, menos flexibles y con un tiempo mucho mayor de adaptación al entorno particular.

La elección del paradigma de Elman para la aproximación de la función de consumo ha simplificado el proceso de construcción de las redes neuronales, ya que el número de neuronas ocultas y el número de pesos tienen menor influencia en el proceso de entrenamiento que el caso de las redes MLP o RBF, con la ventaja de poder utilizar los algoritmos desarrollado para el MLP.

Estimación de estado

El problema de la estimación de estado clásica de un sistema de energía eléctrica ha sido analizado desde el punto de vista de la solución que aporta una red neuronal de Hopfield. Se han presentado los parámetros de la red para que ésta resuelva un problema particular. Se han estudiado las dinámicas de activación de neuronas que aportan una solución más eficiente, y se ha propuesto un valor óptimo para la función de transferencia de cada neurona.

Los resultados obtenidos muestran que la precisión que aporta frente a un estimador clásico WLS es competitiva, encontrando en muchos casos un valor del residuo final (función

a minimizar) más pequeño que el obtenido por el estimador WLS.

Debido a que las matrices a partir de las que se determinan los pesos de la red son altamente dispersas, la matriz de pesos de la red también lo es. Para aprovechar esta situación se ha diseñado una estructura de datos especial para la simulación de la red neuronal que optimiza el proceso de evolución natural de la red, lo que ha reducido considerablemente los tiempos de cómputo requeridos para que la red se estabilice en el mínimo de su función de energía. Sin embargo, aun con estas optimizaciones, los tiempos de cómputo requeridos para obtener estos resultados son mucho mayores que los requeridos por un estimador clásico.

Con estas características de la solución del problema de estimación de estado con redes de Hopfield, es necesario concluir que el paradigma de Hopfield no aporta ninguna ventaja frente a los métodos clásicos, bien establecidos, robustos y eficientemente implementados.

Estimación de topología

Para la tarea de estimación de topología se han propuesto dos soluciones diferentes, ambas basadas en la utilización directa de las medidas disponibles desde SCADA y, cuando es necesario, en la creación de pseudomedidas o en la utilización de estimaciones producidas por un estimador de estado. Las dos soluciones han supuesto la utilización de dos estructuras neuronales complejas.

Las dos soluciones tienen en común que las medidas o pseudomedidas disponibles para realizar la estimación de topología son preprocesadas por una capa de neuronas base radiales. Esta etapa de preproceso lleva a cabo un filtrado que produce, para cada conjunto de medidas reales, un vector que forma agrupaciones, en el sentido de que todas las medidas de una topología dada producen patrones no exactamente iguales pero sí muy similares.

La primera solución se completa con un MLP por cada bus de la red, que generará una propuesta topológica local. Cada MLP es entrenado de forma supervisada para generar las topologías que han producido los correspondientes patrones preprocesados.

La segunda solución reemplaza el MLP por una red de una sola capa de neuronas con función de transferencia Gaussiana con distancia de Mahalanobis, una por cada topología posible de un bus de la red. Cada una de estas neuronas recibe como entradas el vector preprocesado, y genera una salida que indica el grado de pertenencia del vector preprocesado a una topología particular. La topología correcta es tomada como la correspondiente a la

neurona cuyo valor de salida es mayor.

La segunda solución aporta varias ventajas sobre la primera. En primer lugar, elimina la necesidad de realizar un proceso de entrenamiento supervisado, que en el caso del MLP requiere de un gran número de patrones por cada bus de la red y que, por tanto, es lento. En segundo lugar, la utilización de la distancia de Mahalanobis permite utilizar el estimador topológico aún en el caso de que una medida del bus no esté disponible.

En ambos casos, el proceso de estimación topológica se realiza de forma local para cada bus, lo cual implica que la complejidad computacional del método crece sólo de forma lineal con el número de buses de la red.

Análisis de Seguridad

Se ha estudiado la aplicabilidad de varios paradigmas neuronales a la tarea del análisis de contingencias, abordando el problema tanto desde el punto de vista de la ordenación de las mismas en función de su gravedad, como desde el punto de vista de su monitorización de forma visual y gráfica.

Para el caso de la ordenación de contingencias, se han utilizado los paradigmas neuronales con entrenamiento supervisado MLP y RBF. Ambos paradigmas, en su fase de operación, funcionan en tiempo real, lo que los hace válidos para ser utilizados como alternativa a otros métodos clásicos con costes computacionales mucho mayores. Los resultados comparativos de estas dos redes muestran un porcentaje mayor en los errores relativos medios y un porcentaje mayor en la correcta ordenación final de contingencias para el caso de la red RBF. Esta aparente contradicción puede explicarse debido a que la red RBF produce errores mayores que la red MLP para patrones con un índice de prestación igual a cero. Sin embargo, los errores en estos patrones, que representan estados no peligrosos, no afectan a la ordenación final.

En la monitorización visual de contingencias se ha utilizado el paradigma Auto-Organizativo de Kohonen (SOM). En primer lugar, se ha propuesto la utilización de una única SOM bidimensional para la clasificación de los estados del sistema. Una única red permite, para los patrones de una contingencia dada, su ordenación en función del grado de peligrosidad de la contingencia. Además, la ordenación de los mismos sigue una distribución lineal a lo largo

de la diagonal del mapa. Esta distribución facilita enormemente la visualización de la peligrosidad de una contingencia particular y, sobre todo, permite el seguimiento de la evolución del sistema y la predicción de su acercamiento o no a un estado de peligro.

Por otro lado, la distribución lineal sobre la diagonal del mapa ha permitido formular y verificar la hipótesis de que la arquitectura ideal de la SOM para este problema es lineal. Así, el entrenamiento de una única SOM lineal ha permitido una distribución de los niveles de peligrosidad que, aunque distinta para cada contingencia, mantiene el orden de peligrosidad creciente en todos los casos. Esta solución aporta la facilidad de permitir la presentación en una única pantalla de un gran número de contingencias.

6.2. Líneas futuras

Como líneas futuras de investigación en los temas abordados en este trabajo, destacamos los siguientes:

- Los paradigmas propuestos para tareas relacionadas con EMS pueden ser aplicados a otros muchos campos, tales como economía, ingeniería y otras áreas sociales.

- Utilización de modelos híbridos como RNAs y lógica borrosa para el tratamiento específico de la predicción de la demanda de energía eléctrica en los periodos especiales, de forma que podrían constituir una etapa correctora que permita incorporar, mediante reglas, el conocimiento experto de la influencia de factores como la laboralidad, variación meteorológica, etc.

- En entornos donde se disponga de gran cantidad de datos históricos para periodos especiales como Navidad o Semana Santa, sería factible la creación de clases propias para cada uno de estos periodos, produciendo series temporales mucho más regulares.

- La aplicación del modelo neuronal propuesto para la predicción de la demanda por separado de cada nodo de la red (predicción de carga por nodos) podría aportar información útil para tareas relacionadas con el análisis de seguridad. Esta aplicabilidad requiere de la disponibilidad de gran cantidad de datos históricos separados para cada nodo de la red.

- Estudio de la aplicabilidad del paradigmas de Hopfield a problemas de optimización dentro de las tareas de un EMS.

- Estudio de nuevos paradigmas para tareas de seguridad dinámica, estimación de estado dinámica, etc.

- Inclusión de las medidas de fasores en el estimador topológico basado en funciones Gaussianas con distancia de Mahalanobis.

- Ampliación del número de contingencias simultáneas consideradas en el análisis de contingencias.

Apéndice A

Descripción de los paradigmas neuronales empleados

En este apéndice describimos las características principales de los paradigmas neuronales utilizados a lo largo de esta tesis doctoral.

A.1. Perceptrón Multicapa (MLP)

La figura A.1 muestra la estructura básica de un Perceptrón Multicapa (del inglés *Multilayer Perceptron* o MLP) con una capa de entrada, dos capas ocultas y una de salida. En este tipo de arquitectura neuronal, todas las neuronas de una capa están conectadas a las neuronas de la capa precedente. Todas las señales fluyen en la misma dirección: hacia adelante. La capa de entrada generalmente no está constituida por neuronas (representadas en la figura como pequeños cuadros), ya que no realizan procesamiento alguno, siendo su única función la de alimentar el vector de entrada a la primera capa oculta.

La salida de una capa de neuronas queda según la ecuación (A.1.1):

$$\bar{a} = f\left(\bar{W}\bar{p} + \bar{b}\right) \tag{A.1.1}$$

siendo $\bar{a}$ el vector de salidas de la capa, $\bar{W}$ el vector de pesos, b el vector *bías*, $\bar{p}$ el vector de entradas a la capa y f es la función de transferencia de la neurona.

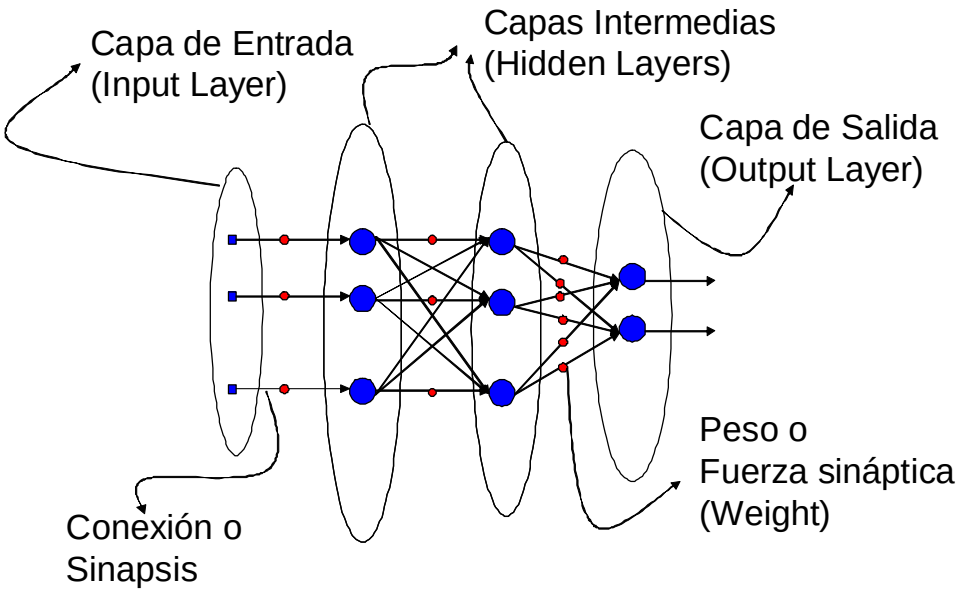


Figura A.1: Red neuronal alimentada hacia delante tipo MLP, con dos capas de neuronas ocultas.

A.1.1. Simulación o ejecución de la red neuronal

El comportamiento de las redes neuronales artificiales viene dado por su función de transferencia. Una red de N capas, con unos determinados pesos y*bías* en las conexiones entre cada una de las capas que conforman la red neuronal, posee un comportamiento definido por la ecuación (A.1.2).

$$\bar{a} = f_N\left(\bar{b}^N + \bar{W}^N \cdot \left[f_{N-1}\left(\bar{b}^{N-1} + \bar{W}^{N-1} \cdot \left[\ \ldots\ f_1\left(\bar{b}^1 + \bar{W}^1 \cdot \bar{p}\right)\ \ldots\ \right]\ \right)\ \right]\ \right) \qquad \text{(A.1.2)}$$

donde N es el número de capas de la red y $\bar{p}$ es un vector de entrada. Esta función indica que para cada vector de entrada $\bar{p}$ se obtiene un vector de salida $\bar{a}$, mediante la propagación de señales desde la capa de entrada hacia la capa de salida, como muestra la figura A.2. Cada uno de los vectores de entrada se denomina patrón de entrada a la red neuronal.

La fase de ejecución de un MLP es una operación computacional muy rápida consistente en la realización de sumas de productos, que puede realizarse en tiempo real.

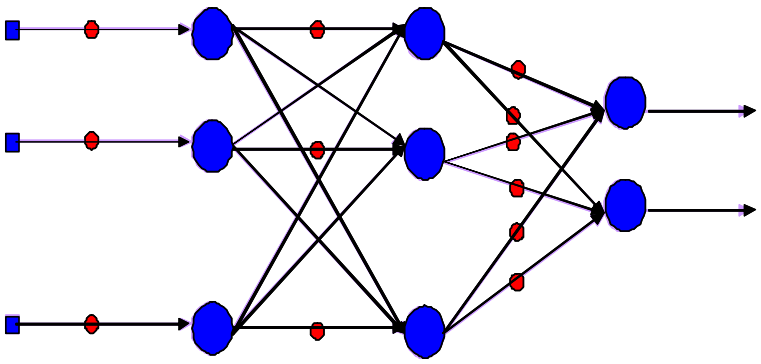


Figura A.2: Dirección del flujo de información durante la ejecución (o*recall*) de un MLP.

A.1.2. Entrenamiento de un MLP

El *conocimiento* de la red neuronal se encuentra en los pesos de las conexiones entre las distintas capas que forman la red neuronal. Durante el proceso de entrenamiento, la red va cambiando los pesos de sus conexiones de forma que la función de transferencia de la red se ajuste al objetivo. El algoritmo más utilizado para realizar este ajuste es el denominado *Backpropagation* ([Rumelhart et al., 1986]) (propagación hacia atrás del error).

Este algoritmo de aprendizaje se basa en propagar el error de salida de cada capa hacia la capa anterior, de forma que se ajusten los pesos de cada capa de manera independiente. El error en la capa de salida $e_j(k)$ se define como la diferencia entre los objetivos (salidas deseadas) $t_j(k)$ y las salidas reales obtenidas de la red $a_j(k)$ para los diversos patrones de entrada. El error en cada nodo de salida j de la red neuronal viene dado por la ecuación (A.1.3):

$$e_j(k) = t_j(k) - a_j(k) \tag{A.1.3}$$

donde k indica el número de la iteración.

El objetivo del algoritmo *Backpropagation* es minimizar la suma de los errores cuadráticos medios de la salida de la red neuronal, dada por la ecuación (A.1.4):

$$\varepsilon\,(k) = \frac{1}{2}\sum_{j=1}^{S} e_j^2(k) \tag{A.1.4}$$

El error medio de todos los patrones de entrada del sumatorio del error cuadrático viene

dado por la ecuación (A.1.5):

$$\varepsilon = \frac{1}{Q}\sum_{k=1}^{Q}\varepsilon\,(k) \tag{A.1.5}$$

siendo Q el número de patrones de entrada disponibles.

El objetivo del entrenamiento es minimizar este error ε a la salida. Para ello se aplica una corrección de los pesos y *bías* de la red neuronal proporcional al gradiente del error instantáneo, que utilizando la regla de la cadena queda como muestra la ecuación (A.1.6):

$$\frac{\partial\varepsilon\,(k)}{\partial w_{i,j}\,(k)} = \frac{\partial\varepsilon\,(k)}{\partial\, e_j\,(k)}\frac{\partial e_j\,(k)}{\partial a_j\,(k)}\frac{\partial a_j\,(k)}{\partial v_j\,(k)}\frac{\partial v_j\,(k)}{\partial w_{i,j}\,(k)} \tag{A.1.6}$$

donde v_j es el estado interno de la neurona j. Evaluando por partes esta ecuación, obtenemos:

$$\varepsilon\,(k) = \frac{1}{2}\sum_{j=1}^{S} e_j^2(k) \quad \Rightarrow \quad \frac{\partial\varepsilon\,(k)}{\partial\, e_j\,(k)} = e_j\,(k) \tag{A.1.7}$$

$$e_j(k) = t_j(k) - a_j(k) \quad \Rightarrow \quad \frac{\partial e_j\,(k)}{\partial a_j\,(k)} = -1 \tag{A.1.8}$$

Si llamamos φ_j a la función de activación de la neurona

$$a_i\,(k) = \varphi_j\,(v_j\,(k)) \quad \Rightarrow \quad \frac{\partial a_j\,(k)}{\partial v_j\,(k)} = \varphi_j'\,(v_j\,(k)) \tag{A.1.9}$$

$$v_j = \sum_{i=0}^{P} w_{ij}\,(k)\;\, a_i\,(k) \quad \Rightarrow \frac{\partial v_j\,(k)}{\partial w_{i,j}\,(k)} = a_i\,(k) \tag{A.1.10}$$

La variación de los pesos de cada capa viene dado por la ecuación (A.1.11):

$$\frac{\partial\varepsilon\,(k)}{\partial w_{i,j}\,(k)} = -e_j\,(k)\cdot\varphi_j'\,(v_j\,(k))\cdot a_i\,(k) \tag{A.1.11}$$

Por tanto, el incremento viene dado por la ecuación (A.1.12):

$$\Delta w_{i,j}\,(k) = -\alpha\,\frac{\partial\varepsilon\,(k)}{\partial w_{i,j}\,(k)} = \alpha\, e_j\,(k)\cdot\varphi_j'\,(v_j\,(k))\cdot a_i\,(k) = \alpha\,\delta_j\,(k)\cdot a_i\,(k) \tag{A.1.12}$$

siendo δ_j el gradiente local.

$$\delta_j\,(k) = e_j\,(k)\cdot\varphi_j'\,(v_j\,(k)) \tag{A.1.13}$$

Para calcular el gradiente del error de cada neurona de cada capa necesitamos conocer el estado interno de la neurona, al que se le aplica la derivada de la función de activación, y el error de salida de la neurona. Este error de salida de cada neurona se conoce en las capas de salida. Para conocer el error de salida en las capas ocultas, el error se propaga hacia atrás desde la capa de salida.

Si el nodo j pertenece a la capa de salida, el error en esa capa es conocido, tal y como se mostró en la ecuación (A.1.3). Si el nodo j pertenece a una capa oculta n, y el nodo h a la capa $n+1$ el error en la capa n queda según la ecuación (A.1.14):

$$e_j(k) = -\frac{\partial \varepsilon(k)}{\partial a_j(k)} = \sum_{h \subset n+1} \frac{\partial \varepsilon(k)}{\partial v_h(k)} \frac{\partial v_h(k)}{\partial a_j(k)} = \sum_{h \subset n+1} e_h(k)\, \varphi_h'(v_h(k)) \qquad \text{(A.1.14)}$$

Para calcular el error por retropropagación utilizamos el método de dos pasadas.

1.- En la pasada hacia delante se simula la red, almacenando el estado interno de las neuronas de cada capa (figura A.2).

2- En la pasada hacia atrás, se retropropaga el error (figura A.3).

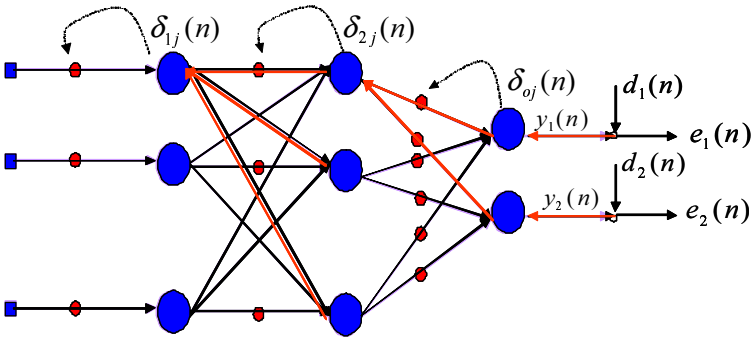


Figura A.3: Dirección del flujo de información durante la fase propagación hacia atrás del error.

Se ajustan los pesos y *bías* de cada capa en función del gradiente local del error:

$$\Delta w_{i,j}(k) = \alpha\, \delta_j(k) \cdot a_i(k) \qquad \text{(A.1.15)}$$

Este gradiente se calcula de forma recursiva:

$$\delta_j(k) = \varphi_j'(v_j(k)) \cdot e_j(k) = \varphi_j'(v_j(k)) \cdot \sum_h \delta_h(k)\, w_{hj}(k) \qquad \text{(A.1.16)}$$

A.1.2.1. Mejoras al algoritmo *Backpropagation*. Métodos alternativos

Para aumentar la velocidad de convergencia del algoritmo *Backpropagation* se han propuesto numerosos métodos que hacen más eficiente el proceso de aprendizaje de la red. La primera mejora se consiguió añadiendo a la regla de modificación de los pesos un término denominado momento (*momentum*). Este término añade un efecto de inercia que permite acelerar la velocidad de convergencia cuando el incremento del los pesos tiene siempre el mismo signo, mientras que su aportación queda anulada si el signo del incremento oscila.

Otras propuestas han consistido en la utilización de funciones de transferencia alternativas a la *sigmoide*, como por ejemplo la función tangente hiperbólica ($f(x) = \alpha tanh(bx)$).

Otras propuestas han utilizado implícitamente o explícitamente las derivadas segundas. Así, del denominado algoritmo *Quickprop* utiliza heurísticas para encontrar información de cómo son las derivadas segundas, pero sin tener que calcularlas.

Finalmente, los métodos de segundo orden se basan en realizar el descenso por el gradiente de pesos utilizando información proporcionada por el ritmo de cambio de la pendiente. Ejemplos de estos métodos son los denominados Gradientes Conjugados, Gradientes Conjugados Escalados y *Levenberg-Marquardt*.

A continuación esbozamos el algoritmo de entrenamiento con ganancia adaptativa, utilizado en varios capítulos de esta tesis doctoral.

A.1.2.2. *Backpropagation* con ganancia adaptativa

El entrenamiento *Backpropagation* adaptativo sigue los principios de retropropagación del error que se han visto anteriormente, pero utiliza una tasa de aprendizaje adaptativa. Una tasa de aprendizaje pequeña hará el entrenamiento más largo y estable, mientras que un valor alto incrementará considerablemente el tiempo de convergencia. Sin embargo, si el valor de la tasa de aprendizaje es muy alta, puede hacer el proceso de entrenamiento inestable. Con el ajuste del paso del entrenamiento *Backpropagation* adaptativo se consigue, por tanto, un entrenamiento más eficiente, intentando ajustarlo para conseguir la velocidad de entrenamiento máxima posible, asegurando en todo momento que el entrenamiento no llegue a la inestabilidad.

Para ello, se requieren tres parámetros adicionales en el entrenamiento de la red neuronal:

- Tasa de incremento. Indica cuánto se debe aumentar el paso si el entrenamiento así lo cree oportuno.
- Tasa de decremento. Indica cuánto se debe decrementar el paso si el entrenamiento debe reducir el tamaño del paso porque se ha vuelto inestable.
- Tasa de error. Rango donde el error puede aumentar sin que se considere que el sistema puede estar oscilando.

El valor de estos parámetros para realizar un correcto aprendizaje de la red neuronal se deben establecer de forma heurística, mediante el método de prueba y error, la mayoría de las veces basados en la propia experiencia del diseñador de la red neuronal.

En cada iteración del entrenamiento de la red neuronal se evalúa la variación del error cometido respecto a la iteración anterior. En este análisis del error se pueden llegar a tres estados deferentes:

1. Si el error ha disminuido, el entrenamiento de la red es correcto y se puede aumentar la tasa de aprendizaje para acelerar el entrenamiento.
2. Si el error ha aumentado por encima del la tasa de error, la red tiende hacia un estado inestable. No se modifican los pesos y *bías* en esa iteración y se reduce la tasa de aprendizaje para evitar la inestabilidad.
3. Si el error ha aumentado por debajo de la tasa de error, es posible que la red deba superar un mínimo local en su función de aprendizaje. Por ello, se modifican los pesos y *bías* de la red de acuerdo a los valores obtenidos en esa iteración, pero no se altera el valor del paso del entrenamiento.

Con estas modificaciones, a costa de un incremento computacional pequeño, la convergencia del MLP suele ser mucho más rápida que con el método *Backpropagation* básico y, en la mayoría de los casos, suele evitar muchos mínimos locales.

A.2. Redes de funciones base radiales (RBF)

Las redes neuronales de Funciones Base Radiales (del inglés, *Radial Basis Function Networks*) son del tipo alimentadas hacia delante, con una arquitectura de tres capas, como muestra la figura A.4. La capa de entrada se utiliza para presentar a la red los patrones de entrada. La segunda capa o capa oculta está constituida por neuronas no lineales, y la capa de salida por neuronales con funciones de transferencia lineales.

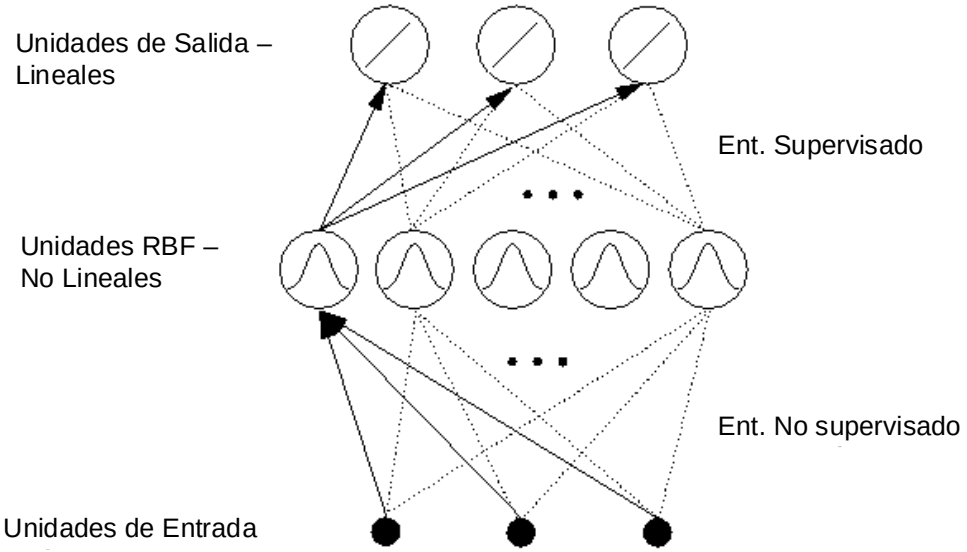


Figura A.4: Estructura de una red de funciones base radiales.

La capa oculta proporciona un conjunto de funciones que constituyen una base arbitraria para los patrones de entrada (vectores de entrada) cuando éstos son alimentados al espacio de las unidades ocultas ([Haykin, 1994]). De esta forma, las neuronas ocultas se denominan funciones base radiales (el término inglés correspondiente es *radial-basis functions*). Estas funciones de transferencia tienen dos parámetros, centro y ancho. La ecuación (A.2.1) muestra una posible elección:

$$f_i(x) = e^{\frac{-\|c_i - x\|}{2\sigma_i^2}} \tag{A.2.1}$$

donde $\mathbf{x}$ representa el vector de entrada, c_i es el centro de la neurona i y σ_i determina el ancho de la neurona i. Los parámetros c_i y σ_i de cada neurona se fijan durante el proceso

de entrenamiento, así como los pesos de la capa de salida. El término radial se debe a la simetría radial de estas funciones con respecto al centro c_i. La función de transferencia de una red RBF viene dada por la ecuación (A.2.2):

$$y_k(x) = \sum_{i=1}^{M} w_{k,i} f_i(\|c_i - x\|, \sigma_i) \tag{A.2.2}$$

donde y_k es la salida de la k-ésima neurona de salida de la red, $w_{k,i}$ es el peso de la conexión entre la neurona de salida k y la neurona oculta i, M es el número de neuronas ocultas, y f_i viene dada por la ecuación (A.2.1).

A.2.1. Entrenamiento de RBF

El primer paso en el entrenamiento de una red RBF (al igual que con otros paradigmas) es encontrar el número de neuronas óptimo para resolver un problema particular. En el caso de las neuronas RBF, cada una de ellas cubre una parte del espacio de entrada, por lo que habrá que elegir un número suficiente que cubra todo el espacio de entrada. Sin embargo, debido a que el número de neuronas necesarias para cubrir este espacio crece exponencialmente con la dimensión de los vectores de entrada ([del Brío and Molina, 2001], [Haykin, 1994]), generalmente, se hace necesario llegar a un compromiso entre la precisión de la solución y el número de neuronas. Además, como con la red MLP, un excesivo número de neuronas ocultas puede producir una sobreparametrización de la red, lo que repercutirá negativamente en su capacidad de generalización. En este sentido, las técnicas que se emplean comúnmente para la determinación del número de neuronas ocultas en un MLP pueden también ser aplicadas al caso de las redes RBF. Entre ellas, el método de prueba y error suele emplearse frecuentemente, aunque existen otras técnicas más eficientes como el algoritmo Auto-Organizado jerárquico ([Haykin, 1994]).

Una vez determinado el número de neuronas base radiales, debemos encontrar los parámetros de la arquitectura. En este sentido, a diferencia del entrenamiento del la red MLP, suele utilizarse un método por etapas, donde en primer lugar se entrenan las neuronas ocultas y, en una segunda etapa, las neuronas de salida. En la primera etapa, el proceso de entrenamiento de la red consiste en encontrar los parámetros c_i y σ_i, de forma que minimicen alguna medida de error. Para los parámetros c_i, entre los algoritmos más utilizados podemos destacar los siguientes:

- Selección aleatoria.
- Selección de centros mediante el algoritmo k-medias.

Una vez fijados los parámetros c_i, debemos encontrar los parámetros de escala σ_i, que suele hacerse mediante técnicas heurísticas. Las ecuaciones (A.2.3) y (A.2.4) muestran dos procedimientos ampliamente utilizados. La primera ecuación calcula de forma aproximada la influencia en el espacio de entrada de cada neurona en relación a las demás. La segunda se basa en el cálculo del promedio de la distancia de diversos patrones representativos al centro c_j.

$$\sigma_i^2 = \frac{1}{N}\sum_{l=1}^{N}\|c_l - c_j\|^2 \tag{A.2.3}$$

$$\sigma_i^2 = \frac{1}{N_j}\sum_{x\in nodoj}^{p}\|x - c_j\|^2 \tag{A.2.4}$$

La última etapa del entrenamiento consiste en el aprendizaje de la capa de salida, esto es, determinar el valor de los pesos $w_{i,j}$. En este caso, una vez fijados los parámetros de las neuronas ocultas c_i y σ_i, los pesos $w_{i,j}$ se calculan aplicando el algoritmo LMS (Least Mean Squares) a la expresión de la salida de la capa de salida según la ecuación (A.2.5):

$$z_k = \sum_j w_{k,j} f_r(c_i, \sigma_i) + \theta_k \tag{A.2.5}$$

donde f_r representa la salida de una neurona de la capa oculta y θ_k es el *bías* de la neurona. Los pesos se obtienen mediante la ecuación (A.2.6):

$$w_{k,j}(t+1) = w_{k,j}(t) + \varepsilon(t_k - z_k) f_r(c_i, \sigma_i) \tag{A.2.6}$$

donde t_k son los valores objetivos o salidas deseadas.

Finalmente, señalar que con respecto al entrenamiento de una red MLP, el entrenamiento de una red RBF es más eficiente debido a que éste se realiza en dos etapas.

A.3. Red recurrente de Elman

La red de Elman forma parte de las denominadas redes parcialmente recurrentes. Tienen la forma general de un MLP, pero se añaden conexiones recurrentes. Está constituida por una

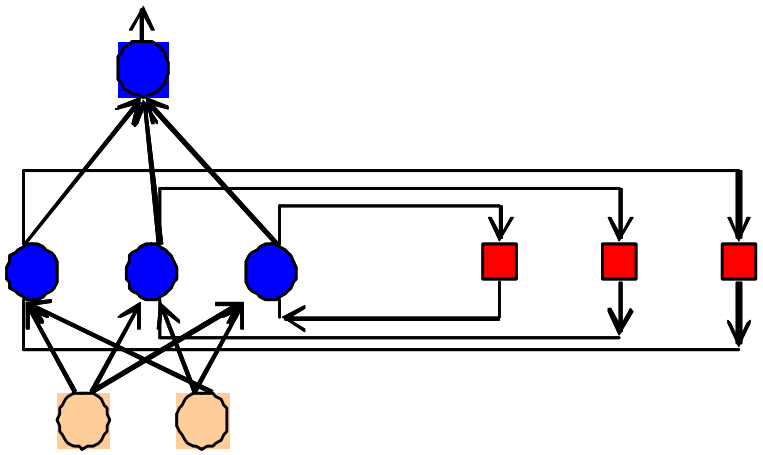


Figura A.5: Representación de una red de Elman.

capa oculta con función de activación tangente hiperbólica y una capa de salida con función de activación lineal. Las conexiones recurrentes parten de la salida de la capa oculta y se realimentan como entradas a esta misma capa, como muestra la figura A.5. Las recurrencias de este modelo neuronal le permite tanto detectar como generar patrones que van cambiando a lo largo del tiempo ([Elman, 1990], [Elman, 1991]).

Al igual que el MLP, su estructura de dos capas le permite aproximar cualquier función que tenga un número finito de discontinuidades con una precisión especificada, con la única condición de que el número de neuronas ocultas sea suficiente [Haykin, 1994]. Las realimentaciones aportan a la red, en la iteración t, información sobre el estado de la misma en la iteración t-1.

Al igual que el modelo de Jordan, esta red puede ser entrenada con los algoritmos diseñados para el MLP con un mínimo de cambios. A continuación, esbozamos el algoritmo básico de entrenamiento para la red de Elman:

Para cada época:

1. Presentar los patrones a la red, y calcular las salidas.

2. Calcular el error de las salidas, comparándolas con las salidas deseadas.

3. Para cada patrón presentado, calcular el incremento de los pesos y *bías* utilizando el algoritmo *Backpropagation*. En esta etapa el gradiente de los pesos y *bías* de las

conexiones recurrentes se ignoran.

4. Actualizar el vector de pesos de cada neurona.

El inconveniente de esta red frente al MLP es que la adaptación de los pesos utiliza una aproximación del gradiente del error. Por esta razón, para que una red de Elman pueda resolver el mismo problema que un MLP, generalmente requerirá de un número mayor de neuronas ocultas ([Demuth and Beale, 2000]). Por otro lado, aporta varias ventajas importantes. En primer lugar, la realimentación de señales proporciona a la red estados internos y memoria, haciendo que las salidas no dependan sólo de las entradas actuales, sino de las entradas previas. En segundo lugar, tiene la ventaja de que la solución a un problema es menos sensible a los parámetros de la misma (número de neuronas y número de pesos). En tercer lugar, frente a otras redes realimentadas, el modelo de Elman aporta la gran ventaja de poder utilizar todos los algoritmos de entrenamiento diseñados para el MLP, más eficientes, estables y robustos que los diseñados para redes recurrentes genéricas.

A.4. Redes de Kohonen

Las redes de Kohonen, también denominadas Mapas Auto-Organizativos (SOM , de*Self -Organizing Maps* en inglés) se caracterizan por su capacidad para clasificar un conjunto complejo de patrones de manera no supervisada, extrayendo criterios de clasificación no obvios ni expresados de manera explícita. Esta clasificación es llevada a cabo mediante la distribución de un espacio de entrada V_I de alta dimensión en un espacio de salida V_O de menor dimensión, preservándose las relaciones topológicas existentes entre los patrones de entrada. El espacio de salida viene constituido por un conjunto de neuronas ordenadas generalmente en un plano o una línea, en el cual se define una función de vecindad (figura A.6).

Tras el proceso de aprendizaje, un vector de entrada$\vec{x} = (x_1, ..., x_n)$ activará la neurona i del espacio de salida cuyo vector de pesos $\vec{w}_i = (w_{i1}, ..., w_{in})$ tenga menor distancia al vector x. De esta manera, el vector $\vec{w}_i$ podrá ser considerado como el prototipo de la región del espacio de entrada cuyos vectores activan a la neurona i. Finalmente, dos vectores de entrada similares según la relación definida enV_I, activarán la misma neurona o dos neuronas cercanas en el espacio de salida (figura A.7).

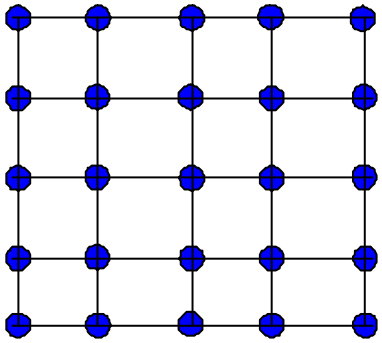


Figura A.6: Disposición bidimensional de las neuronas de una red de Kohonen.

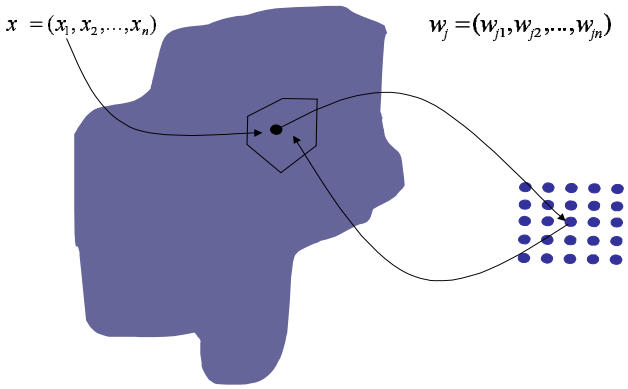


Figura A.7: Representación esquemática de la distribución de un espacio n-dimensional de entrada en un espacio bidimensional mediante un Mapa Auto-Organizativo de Kohonen.

A.4.1. Entrenamiento de un SOM

En el proceso de entrenamiento, cada vez que se presenta a la red un nuevo patrón de entrada $\vec{x}$, la red busca la neurona que menor distancia tiene a este vector. Esta neurona se denomina neurona ganadora, y cambia su vector de pesos en el sentido de aproximarse a$\vec{x}$, como refleja la ecuación (A.4.1):

$$\vec{w}_i(t+1) = \vec{w}_i(t) + l_r h_{iv}(\vec{x} - \vec{w}_i(t)) \tag{A.4.1}$$

donde $\vec{x}$ es un vector de entrada, $\vec{w}_i$ es el vector de pesos de la neurona i, y $h_{iv}()$, la función de vecindad, determina el incremento del peso de cada neurona como una función de su proximidad a la neurona ganadora para el patrón $\vec{x}$. Generalmente, el área de vecindad viene dada por un cuadrado centrado en la neurona ganadora, cuyo lado decrece hasta cero a lo largo del entrenamiento, como muestra la figura A.8. lr es la razón dinámica de aprendizaje, la cual evoluciona a lo largo del proceso de entrenamiento según la ecuación (A.4.2):

$$l_r(t) = \frac{l_{r0}}{(1 + \frac{c \cdot t}{nn})} \tag{A.4.2}$$

siendo l_{r0} la razón de aprendizaje inicial (0.3 en nuestro caso), c una constante (0.2), t la iteración actual y nn el número de neuronas de la red [Cottrell and Letremey, 1995], [Haykin, 1994].

El algoritmo de entrenamiento básico se esboza a continuación:

1. Inicializar los pesos de la red asignando a cada componente de los vectores de pesos números aleatorios en el intervalo [-1,1].
2. Presentar un nuevo vector de entrada $\vec{x}$.
3. Encontrar la neurona g cuyos pesos están más próximos (en el sentido de la distancia Euclidea) al vector $\vec{x}$.
4. Modificar el vector de pesos de la neurona g y de sus neuronas vecinas utilizando la ecuación (A.4.1).
5. Modificar el tamaño de vecindad y la tasa de aprendizaje.
6. Ir al paso 2 hasta mientras no concluya el proceso de entrenamiento.

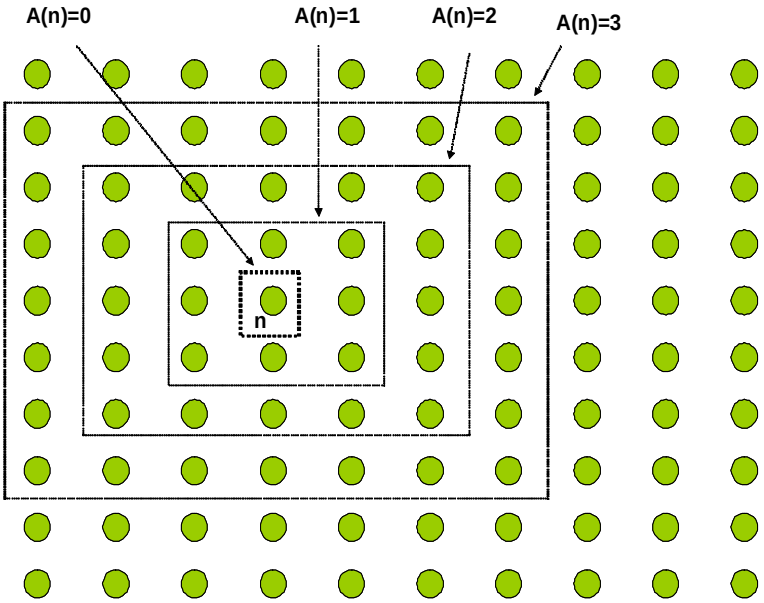


Figura A.8: Vecindad topológica cuadrada para una red de Kohonen bidimensional. Las neuronas vecinas de n para varios valores de la función A son todas las incluidas en el cuadrado correspondiente..

A.5. Red de Hopfield

Una red de Hopfield ([Hopfield, 1982], [Hopfield, 1984], [Hopfield and Tank, 1985]) tiene una única capa de neuronas, donde cada neurona se conecta con todas las demás, configurando una red recurrente completamente conectada, como muestra la figura A.9.

Este modelo neuronal presenta dos variantes, el modelo discreto y el modelo analógico o continuo, que se describen en las secciones siguientes.

A.5.1. Modelo discreto

En el modelo discreto tanto las salidas como las entradas son valores binarios ({0,1} o {-1,1}), mientras que los pesos $w_{i,j}$ son valores reales (continuos). El potencial de la neurona se calcula según la ecuación (A.5.1):

$$y_i(t) = f(\sum_j w_{i,j} x_j(t) - \theta_i) \qquad \text{(A.5.1)}$$

donde f es la función de transferencia de las neuronas.

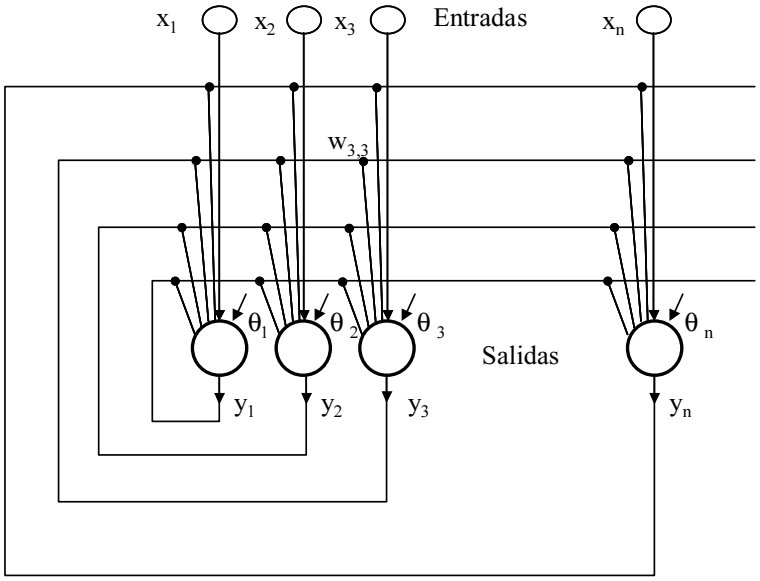


Figura A.9: Arquitectura de una red de Hopfield.

En el proceso de ejecución de la red, en t=0, las neuronas reciben el valor de las entradas x_j, calculando las salidas asociadas. Debido a las realimentaciones, las salidas se convierten en nuevas entradas en t=1, y así sucesivamente. En general, la dinámica de una neurona de una red de Hopfield viene dada por la ecuación (A.5.2):

$$x_i(t+1) = f(\sum_j w_{i,j}x_j(t) - \theta_i) \qquad \text{(A.5.2)}$$

Esta ecuación proporciona la dinámica de una neurona de la red. Para obtener la dinámica de la red debe especificarse el orden en el que se evalúa la ecuación (A.5.2) para cada neurona, es decir, debemos decidir si todas las neuronas calculan su salida en paralelo (simultáneamente), secuencialmente (una tras otra), o de forma aleatoria. Esta ordenación se denomina dinámica de la red, y se suelen distinguir dos tipos genéricos ([del Brío and Molina, 2001]):

a) Dinámica asíncrona o modo serie de operación. En el instante t sólo una neurona de la red actualiza su estado. La elección de la neurona se puede hacer aleatoriamente o siguiendo un orden preestablecido.

b) Dinámica síncrona o modo paralelo, en el que varias neuronas actualizan simultáneamente su estado en el instante t. Si todas las neuronas actualizan simultáneamente su estado estamos ante el denominado modo de operación completamente paralelo.

La aplicación de una u otra dinámica aplicada sobre la misma red hace que opere de forma diferente y que el estado final de las neuronas sea diferente.

Por ser una red realimentada, la respuesta de la red a lo largo del tiempo podrá estabilizarse, convergiendo a un estado estable, punto fijo o atractor. En otros casos, la respuesta puede no converger, describiendo las salidas una trayectoria aleatoria. Existen teoremas generales que establecen las condiciones que debe cumplir una red realimentada para que su respuesta sea estable. Una forma de hacerlo es mediante el método de*Lyapunov* ([Sympson, 1990]). La función de *Lyapunov* se denomina también función de energía de *Lyapunov* debido a que constituye una generalización del concepto físico de energía. Esta función permite garantizar que si se puede encontrar una función de energía para una red que disminuya con la evolución de la misma, entonces su dinámica será estable. Hopfield demostró ([Hopfield, 1982]) que su modelo era estable en el caso de que la matriz de pesos sea simétrica y de diagonal nula. Esta función de energía tiene la forma de la ecuación (A.5.3):

$$E = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{i,j}x_i x_j + \sum_{i=1}^{n}\theta_i x_i \qquad \text{(A.5.3)}$$

donde debe verificarse que $w_{i,i} = 0$ y $w_{i,j} = w_{j,i}$. Por construcción, $\Delta E \leq 0$, por lo que la energía de la red nunca crece. Como consecuencia, la red siempre llegará a un estado de mínima energía que se corresponderá con un estado estable o punto fijo.

A.5.1.1. Determinación de los pesos

La aplicación más importante de este modelo es su utilización como memoria asociativa. En este sentido, para almacenar un conjunto de patrones se suele emplear la regla de Hebb, dada por la ecuación (A.5.4):

$$w_{i,j} = \frac{1}{n}\sum_{\mu=1}^{p} x_i^{\mu} x_j^{\mu} \qquad \text{(A.5.4)}$$

para el caso de neuronas binarias de valores {-1,1}. En esta ecuación, p representa el conjunto de patrones. En este caso, no existe un proceso de entrenamiento como en el caso de los

otros paradigmas descritos en este apéndice, sino que directamente se obtienen los valores de los pesos aplicando la ecuación (A.5.4) para un conjunto dado de patrones.

A.5.2. Modelo continuo

El modelo discreto de la red de Hopfield tiene el inconveniente de que las salidas únicamente pueden estar en dos estados, 0 (o -1) para neuronas inactivas y 1 para neuronas activas (o +1). Esto no se corresponde con los sistemas biológicos que inspiraron los modelos neuronales, donde las salidas de las neuronas pueden tomar cualquier valor en el intervalo [0,1]. Para solventar este problema, en [Hopfield, 1984] Hopfield modificó su modelo permitiendo que las neuronas presentasen una respuesta analógica mediante el uso de la función de transferencia *sigmoide* dada por la ecuación (A.5.5):

$$y_i(t) = f(x) = \frac{1}{1 + e^{-\beta x}} \tag{A.5.5}$$

Al igual que para el caso discreto, para el modelo continuo puede definirse una función de energía que garantiza su estabilidad. Otra característica importante es que puede implementarse fácilmente mediante electrónica analógica, utilizando amplificadores. En este caso, la función de energía viene dada por la ecuación (A.5.6):

$$E = -\frac{1}{2}\sum_{i,j} w_{i,j} y_i y_j + \sum_i C_i \int_0^{y_i} g^{-1}(y)dy - \sum_i x_i y_i \tag{A.5.6}$$

donde C_i es una constante. Hopfield demostró que esta función no es positiva si la función g^{-1} es monótona creciente y la matriz de pesos simétrica.

Cuando la función g es abrupta (ganancia elevada), los estados estables del sistema continuo y discreto coinciden ([del Brío and Molina, 2001]) si se cumple que$w_{i,j} = w_{j,i}$ y $w_{i,i} = 0$.

A.5.2.1. Determinación de los pesos

El modelo continuo de Hopfield se suele utilizar en problemas de optimización. El modelo debe basarse en la suposición de que las salidas de la red se corresponden con las variables del problema a resolver. Las condiciones o restricciones del problema deben incorporarse explícitamente como términos de la función de energía de la red. En este caso, cada valor de

la función de energía se corresponde con una solución del problema. Si es posible encontrar un conjunto de pesos y entradas que formen una función de energía, dejando evolucionar a la red, ésta alcanzará un mínimo local, que representará una solución localmente óptima del problema.

Apéndice B

Descripción básica de un sistema de gestión de energía eléctrica

En este apéndice se realiza una revisión esquemática de las principales tareas implicadas en la Gestión de un Sistema de Energía Eléctrica.

B.1. Introducción

Debido a que la generación y el consumo de energía eléctrica generalmente tienen lugar en lugares geográficos muy distantes, es necesario *transportar* grandes cantidades de energía eléctrica a lo largo de grandes distancias, y debe realizarse de la forma más eficiente posible. La función de un sistema de transporte de energía eléctrica es llevar la energía desde donde se produce hasta los puntos en los que se distribuye. En este sentido, se distingue entre redes de transporte y redes de distribución. Estas últimas realizan la distribución de la energía hasta el usuario final. Entre las diferencias más importantes entre ambos sistemas están los niveles de tensión a los que se trabaja. Para la red de transporte, los niveles típicos son 400kV, 220kV y 132kV. Para las redes de distribución, los niveles de tensión suelen estar por debajo de los 110kV. Las tareas que se describen en este apéndice son las típicas de la red de transporte.

Durante la operación de la red de transporte pueden presentarse condiciones de funcionamiento anormales que deben ser resueltas en un corto espacio de tiempo. La tarea de los operadores del sistema es realizar la operaciones de control necesarias para mantener la

seguridad de la red ante estas eventualidades. Sin embargo, sin la ayuda de aplicaciones para la monitorización y control, preservar la seguridad puede ser una tarea muy difícil o, bajo circunstancias especiales, imposible. Por tanto, para asistir a los operadores se han diseñado una serie de aplicaciones o funciones que permiten monitorizar el sistema de forma que pueda mantenerse lejos de situaciones en las que puedan producirse sobrecarga de equipos, niveles de tensión anormales, inestabilidad del sistema, pérdida del suministro de energía a alguna carga e incluso la posibilidad de suspender el suministro en todo el sistema. Estas funciones utilizan periódicamente la información disponible en tiempo real de diversas partes de la red, para determinar si el sistema se encuentra cerca de, o en un estado de emergencia. Estas funciones de seguridad, junto con otras funciones de control y restablecimiento de la seguridad, constituyen lo que se denomina Sistema de Gestión de Energía Eléctrica (del inglés *Energy Management System* o EMS).

La figura B.1 muestra las tareas principales que componen un análisis de seguridad *on-line*. Este análisis de seguridad constituye la tarea principal en la gestión del sistema. El análisis de seguridad comienza con la recepción en SCADA (sistema de supervisión, control y adquisición de datos, o, en inglés, *Supervisory Control and Data Acquisition system*) de las medidas analógicas (inyecciones y flujos de potencia, corrientes, módulos de tensión, fasores de tensión y corriente) y digitales (estado de interruptores y seccionadores) a través de las unidades remotas. Las medidas recibidas se someten a un primer filtrado que permite descartar aquellas que tienen valores no coherentes con las cantidades físicas que representan. Los datos digitales son procesados de forma que se construye un modelo topológico de la red que determina el conexionado de sus elementos (conecciones de generadores, cargas, líneas de transmisión, transformadores, etc.). A partir del modelo topológico y de las medidas analógicas disponibles se realiza una *estimación de estado* que permite obtener el vector de estado de la red, que puede utilizarse para determinar otras variables de interés en el sistema. Como se describe más adelante, las tareas de análisis de observabilidad y de detección de medidas erróneas son parte de la estimación de estado. La estimación de estado permite conocer si la red se encuentra en un estado de emergencia mediante la verificación de ciertos límites de los valores de las variables más importantes.

Si el estado actual del sistema es normal, el siguiente paso es determinar la seguridad del mismo ante la posibilidad de que se produzcan determinados fallos en uno o varios

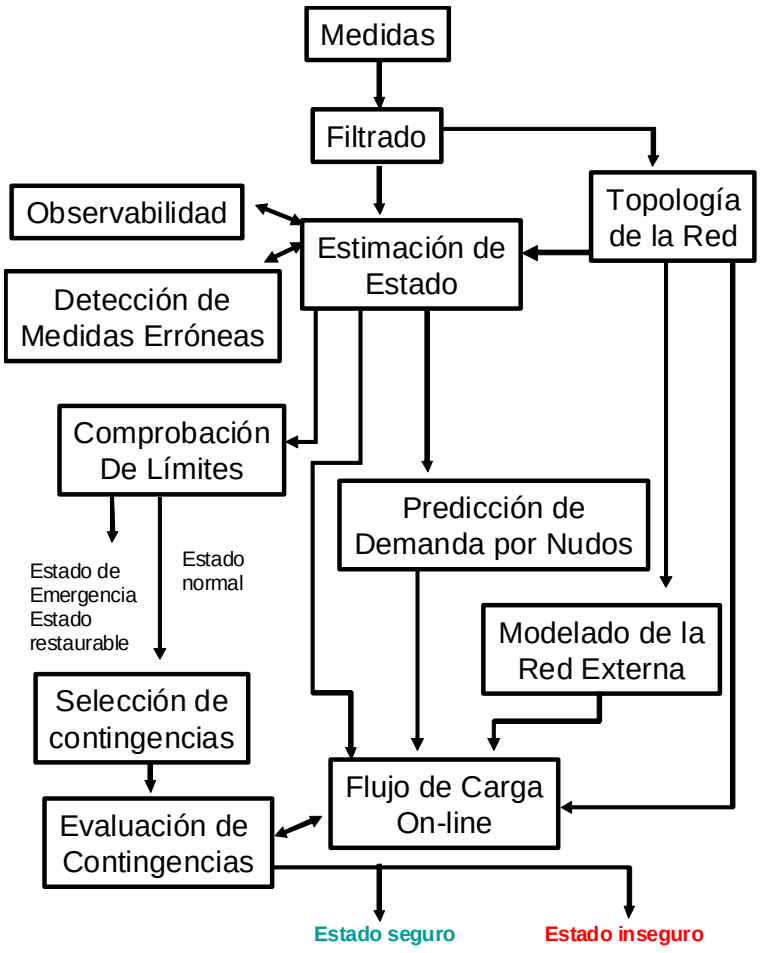


Figura B.1: Componentes principales de un análisis de seguridad*on-line* ([Balu et al., 1992]).

equipos de la red o ante el incremento de la demanda de energía. Esta tarea se denomina *análisis de contingencias*. Debido a que la cantidad de posibles contingencias es muy grande, normalmente se necesita un procedimiento que seleccione un reducido número de ellas entre las posibles o más probables. Esta tarea requiere generalmente de la resolución de problemas de flujo de carga y, en muchos casos, debido a que una contingencia es un evento futuro, de la predicción del consumo de cada nodo de la red (predicción de demanda por nudos). Para cada contingencia seleccionada se evaluará su gravedad, de forma que se puedan tomar las acciones de control anticipadas para asegurar estabilidad del sistema ante la eventualidad de que esa contingencia se produzca. Estas y otras tareas relacionadas se describen brevemente en las siguientes secciones.

Por otro lado, la energía no puede ser almacenada de forma eficiente. Debido a esto, la generación y el consumo de energía deben igualarse en cada momento. Para garantizar el cumplimiento de esta restricción se requiere conocer con suficiente antelación el consumo futuro de energía. Esta es la función de la tarea de *predicción de carga*.

B.2. Predicción de carga eléctrica

El objetivo de la predicción de carga eléctrica es obtener los valores futuros de demanda eléctrica mediante la extrapolación de los datos de consumo pasado. El proceso puede considerar factores exógenos como meteorológicos (temperatura, grado nubosidad, velocidad del viento, grado humedad, nivel de lluvia), estacionales, condiciones socioeconómicas (días laborales o festivos) y condiciones culturales (periodos especiales como Navidad, Semana Santa, Ramadán, etc.).

La predicción puede ser clasificada en función de la magnitud que va a ser predicha (pico de carga, carga integral, carga horaria) o en función del intervalo de predicción (muy corto plazo, que considera la predicción de la demanda desde los próximos minutos hasta la próxima hora, corto plazo, que va desde unas horas a varias semanas, largo plazo, implicando periodos de años). Puesto que la energía eléctrica no puede ser eficientemente almacenada, ajustar las curvas de demanda y producción es una tarea vital para un sistema de gestión de energía. Así, en tareas como Programación Horaria de Unidades Térmicas, Despacho Económico, Mantenimiento, Planificación de Expansión, Seguridad en Tiempo

Real y Control, es necesaria una buena predicción. En los últimos años, las nuevas condiciones de deregularización y competición en el mercado de la energía eléctrica han dado un nuevo impulso a esta tarea, y han condicionado el periodo de predicción para ajustarlo a los periodos de operación del mercado.

B.3. Estimación de estado

La Estimación de Estado clásica calcula los valores más probables de las variables de estado de un sistema de energía (magnitud y fase de tensión en cada bus) a partir de un conjunto redundante de medidas, entre las que se encuentran las de magnitud de tensión, inyecciones de potencia activa y reactiva en los buses, y flujos de potencia activa y reactiva por las líneas. A partir de las variables de estado estimadas podrán ser obtenidas el resto de variables de interés del sistema.

Puesto que es imposible obtener un conjunto de medidas exactas, el módulo de estimación de estado se enfrenta a los siguientes problemas:

1. Sólo una parte de las medidas que pueden ser usadas como variables de entrada están disponibles en cada momento.
2. Todas las medidas analógicas son erróneas debido, entre otros factores, al ruido de los instrumentos de medida y del canal de transmisión de la información.
3. La recepción de las medidas no siempre ocurre en el mismo orden en que fueron generadas, ni todas las medidas de determinado conjunto de entrada tienen por qué corresponder al mismo instante de tiempo.
4. Es frecuente el uso de pseudomedidas que pueden ser excesivamente erróneas.
5. La información topológica puede ser errónea o incompleta.

La estimación de estado necesita de un mínimo número de medidas que garanticen la observabilidad del sistema, y una información topológica correcta para desarrollar el modelo matemático del mismo. Por tanto, un módulo de estimación de estado debe contener, adicionalmente al estimador propiamente dicho, un submódulo de observabilidad y otro de estimación de topología.

B.3.1. Análisis de observabilidad

El Análisis de Observabilidad consiste en determinar si el conjunto de medidas disponibles es suficiente (tanto en número como en ubicación) para resolver el problema de estimación de estado, y en caso negativo, determinar las regiones observables más extensas. Básicamente, un sistema es observable si se puede asociar una medida de flujo o inyección a cada bus del sistema.

B.3.2. Estimación de topología

Los actuales sistemas llevan a cabo tanto la estimación de estado como el análisis de datos erróneos presuponiendo que la información topológica es correcta, de manera que un error en ésta producirá una configuración de estado incorrecta y una errónea detección de malas medidas. En consecuencia, la identificación de errores topológicos, o estimación de topología, constituye una autentica necesidad en un Sistema de Gestión de Energía. Esta tarea es esencial para que los operadores obtengan el conjunto de datos usados para el análisis de seguridad, en tiempo real.

B.3.3. Detección de medidas erróneas

Durante el proceso de estimación de estado clásica se parte de la hipótesis de que los errores en las medidas analógicas son pequeños y aleatorios con una distribución normal de parámetros conocidos. Sin embargo, en algunas ocasiones puede ocurrir que una o varias medidas presenten errores grandes debido a fallos en la comunicación u otros motivos. Si estas medidas erróneas no se eliminan como entradas al algoritmo de estimación, los resultados de éste pueden no se suficientemente precisos. Por tanto, se hace necesario detectar, identificar y eliminar la medida o medidas erróneas.

B.4. Análisis de seguridad

La carga de un sistema de energía eléctrica está continuamente cambiando como consecuencia de las conexiones y desconexiones de las unidades de consumo. Aunque cambios individuales son en su mayoría pequeños comparados con el tamaño del sistema, pueden aparecer situaciones de gravedad cuando ocurren sucesos tales como el fallo de una línea

o la caída de un generador. En todo momento se requiere un ajuste continuo entre la energía suministrada por los generadores y la carga eléctrica del sistema, o de lo contrario tanto la frecuencia del sistema como los valores de tensión podrán desviarse de sus niveles aceptables provocando una situación de colapso. El objetivo del análisis de seguridad es detectar y corregir estas desviaciones y actuar de forma que se restaure tan pronto como sea posible los valores normales de funcionamiento. Las nuevas condiciones de financiación, deregularización o ambientales que aparecen en el nuevo mercado de la energía, están forzando grandes cambios en el funcionamiento de los sistemas de energía, poniendo el énfasis en la maximización del uso de los recursos de generación y transporte ya existentes. En consecuencia, los sistemas de energía trabajan muy cerca de sus límites de seguridad, y cualquier brecha en ésta puede tener un impacto no previsto.

El análisis de seguridad puede dividirse según las condiciones de estabilidad que deben ser respetadas:

- Estabilidad en estado estable (*Steady State Stability*). Este análisis de seguridad consiste en asegurar la capacidad del sistema de mantenerse estable y controlable cuando se producen pequeñas variaciones de carga.
- Estabilidad en estado transitorio (*Transient stability*). En este caso se estudia la capacidad del sistema de permanecer estable desde que se produce una perturbación hasta que tienen lugar las actuaciones de control correspondientes.
- Seguridad dinámica. Se estudia la capacidad del sistema de retornar a un estado estacionario aceptable después de que las acciones de corrección hayan sido tomadas.

En el análisis de seguridad estático, una vez garantizado éste, continua la estabilidad del sistema en caso de que ocurra un evento crítico. Esta última tarea se suele denominar análisis de contingencias.

B.5. Planificación operacional del sistema de energía eléctrica

Un sistema de energía debe operar de tal manera que los costes de producción sean mínimos mientras se satisfagan determinadas restricciones temporales y de seguridad. Las tareas implicadas en este objetivo son globalmente denominadas planificación operacional.

Estas tareas pueden ser clasificadas en planificación a largo plazo (Planificación de Expansión) y planificación a corto y medio plazo (Programación Horaria de Unidades Térmicas y Despacho Económico de Carga).

Apéndice C

Diagramas de las redes IEEE estándares utilizadas en esta tesis

En este apéndice se presentan los diagramas unifilares de las redes estándares IEEE utilizadas como banco de pruebas en las diferentes tareas de la gestión de un sistema de energía tratadas en esta tesis. Los datos descriptivos de las mismas, en formato*IEEE Common Data Format* ([on a Common Format for the Exchange of Solved Load Flow Data, 1973]), pueden encontrarse en la dirección http://www.ee.washington.edu/research/pstca/.

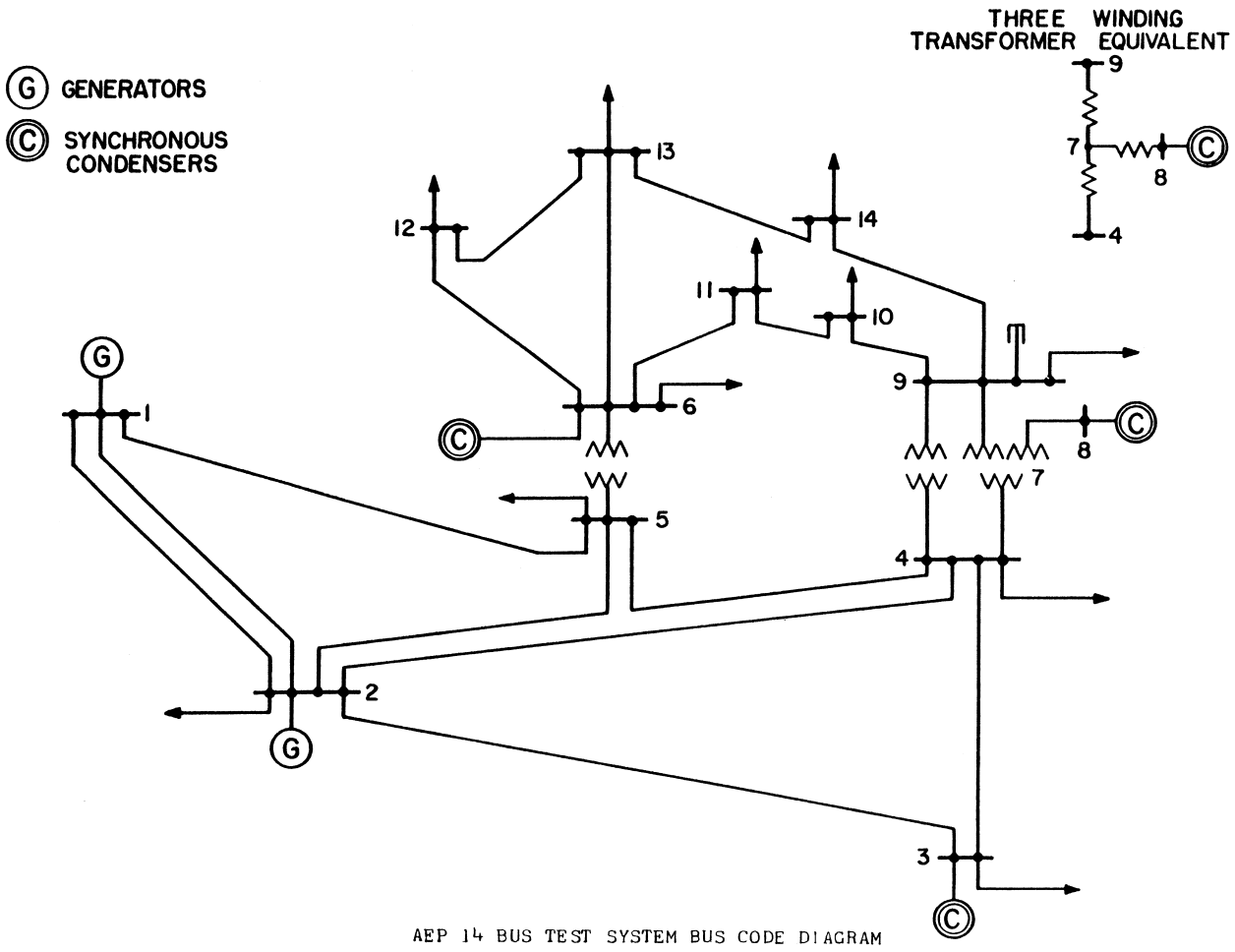


Figura C.1: Diagrama de la red IEEE 14.

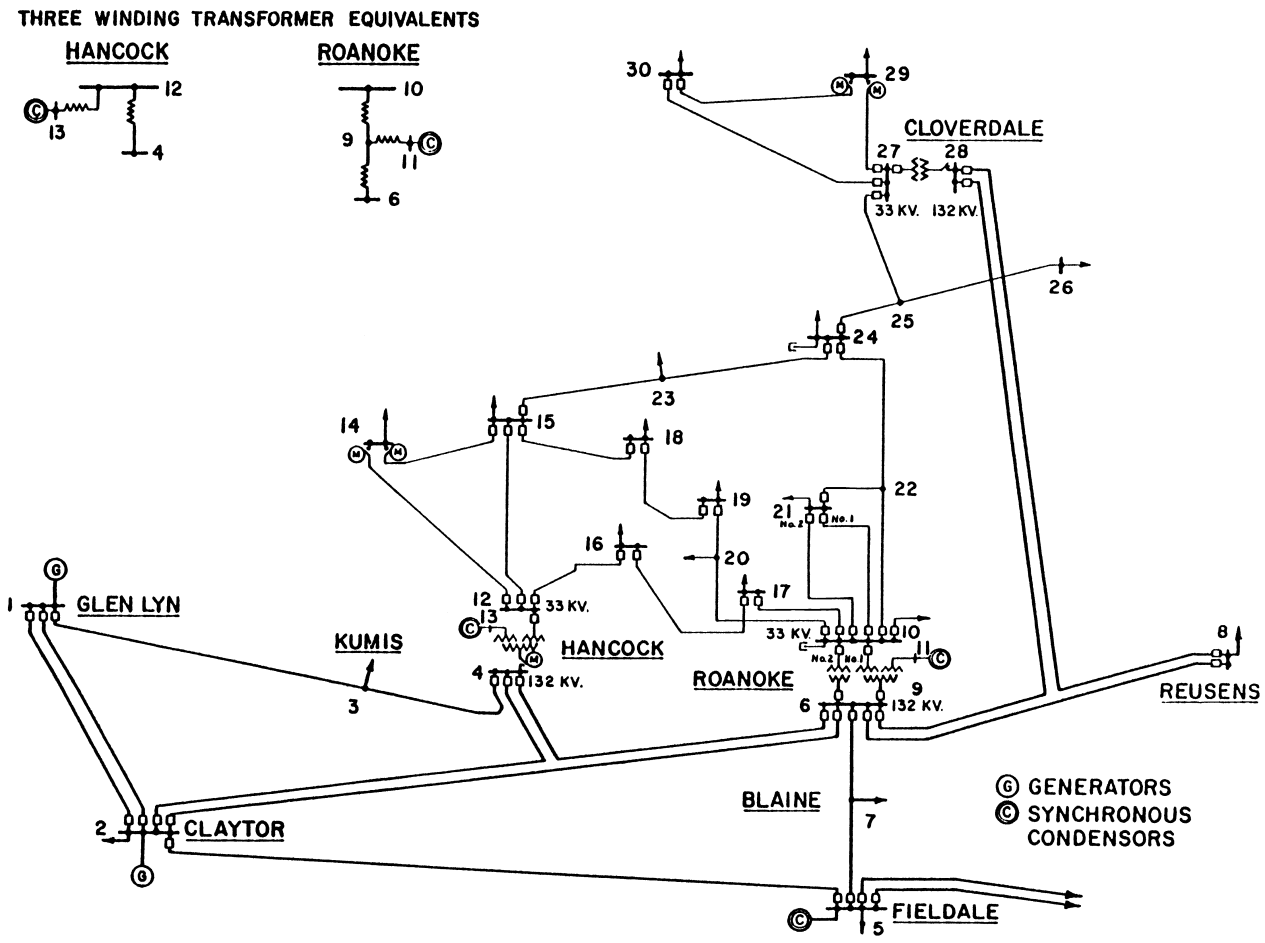


Figura C.2: Diagrama de la red IEEE 30.

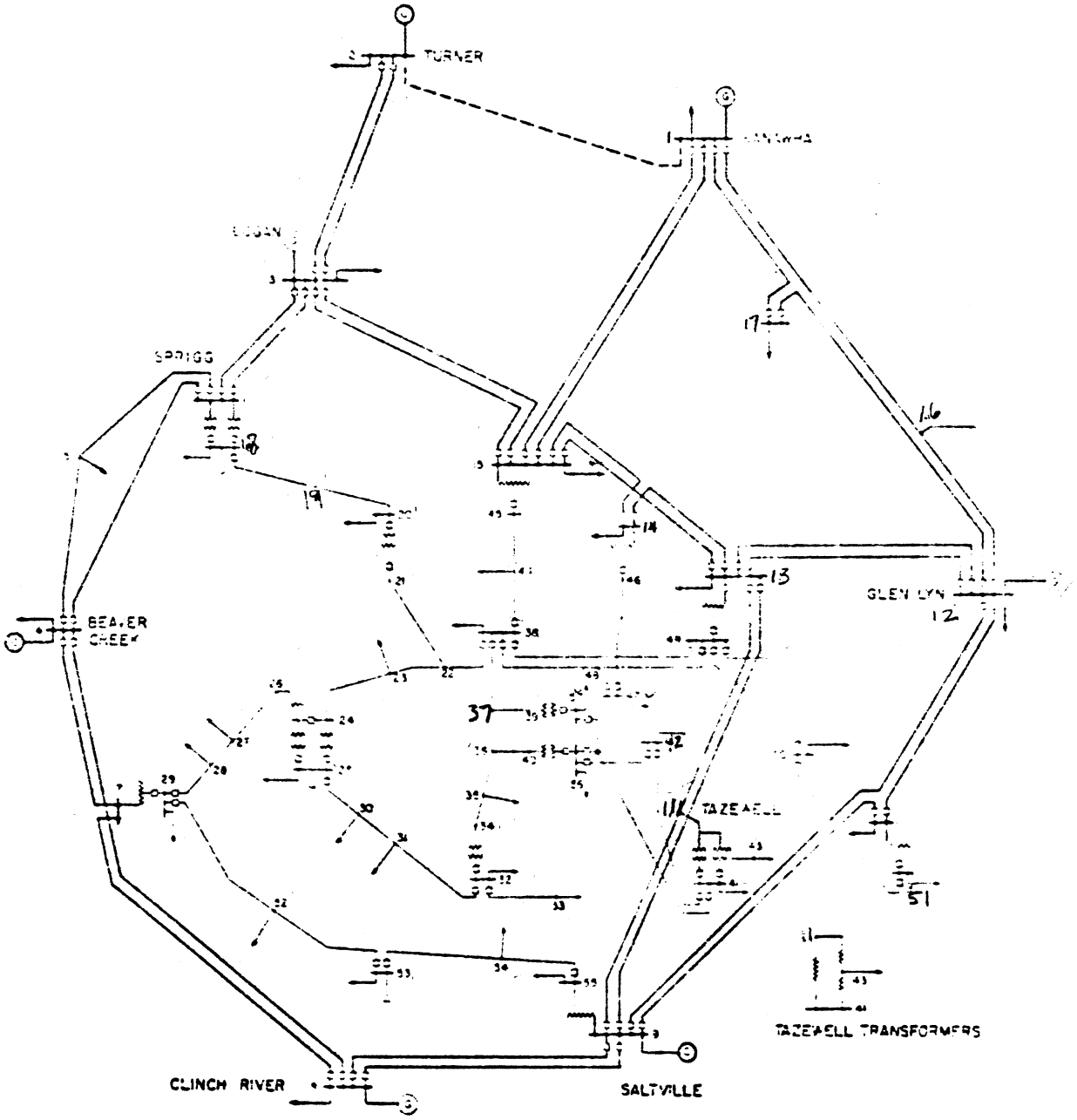


Figura C.3: Diagrama de la red IEEE 57.

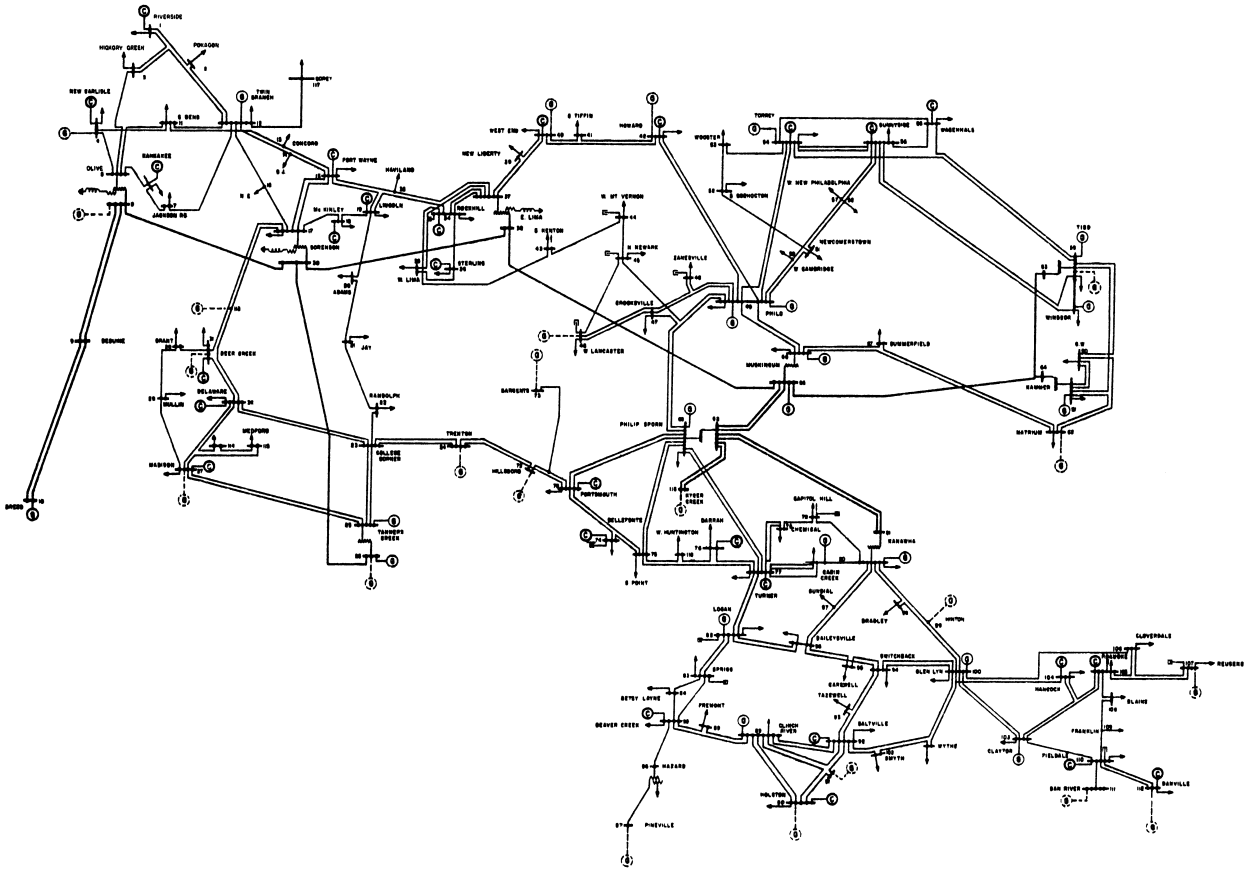


Figura C.4: Diagrama de la red IEEE 118.

Referencias

Abe, S. (1989). Theories on the Hopfield neural networks. *IEEE International Joint Conference on Neural Networks I*, pages 557–564.

Ajjarapu, V. and Christy, C. (1992). The continuation power flow: A tool for steady state voltage stability analysis. *IEEE Transaction on Power Systems*, 7(1):416–423.

Albuyeh, F., Bose, A., and Heath, B. (1982). Reactive power considerations in automatic contingency selection. *IEEE Transactions on Power Apparatus and Systems*, 101:107–112.

AlFuhaid, A. S., El-Sayed, M. A., and Mahmoud, M. S. (1998). Cascaded artificial neural networks for short-term load forecasting. *IEEE Transactions on Power Systems*, 12(4):1524–1529.

Amjady, N. (2001). Short-term hourly load forecasting using time-series modeling with peak load estimation capability. *IEEE Transactions on Power Systems*, 16(3):498–505.

Atencia, M., Joya, G., and Sandoval, F. (2001). Numerical implementation of continuous Hopfield networks for optimization. *Proceeding of the European Symposium on Artificial Neural Networks (ESANN'01).*, pages 359–364.

Balu, N., Bertram, T., Bose, A., Brandwajn, V., Cauley, G., Curtice, D., Fouad, A., Fink, L., Lauby, M. G., Wollenberg, B., and Wrubel, J. (1992). On-line power system security analysis. *Proceeding of the IEEE*, 80(2):262–280.

Bergen, A. R. (1986). *Power Systems Analysis*. Prentice-Hall Inc., Englewood Cliffs, New Jersey 07632.

Bertoldi, O., Salvaderi, L., and Scalcino, S. (1988). Monte Carlo approach in planning studies: An application to IEEE RTS. *IEEE Transactions on Power Systems*, 3(3):1146–1154.

Billinton, R. and Jonnavithula, A. (1996). Variance reduction techniques for use with sequential Monte Carlo simulation in bulk power system reliability evaluation. *Canadian Conference on Electrical and Computer Engineering, 1996*, 1:416–419.

Billinton, R. and Sankarakrishnan, A. (1995). A comparition of Monte Carlo simulation techniques for composite power system reliability assessment. *Proceedings of the IEEE WESCANEX'95*, pages 145–150.

Bonanomi, P. and Gramberg, G. (1983). Power system data validation and state calculation by network search techniques. *IEEE Transaction on Power Apparatus and Systems*, 102(1):238–249.

Borkowska, B. (1981). How to deal with errors in network data. *Proceeding of the 1981 PSCC, Lausanne*, pages 513–519.

Box, G., Jenkins, G., and Reinsel, G. (1994). Time series analysis: Forecasting and control.

Broomhead, D. S. and Lowe, D. (1988). Multivariable functional interpolation and adaptive netowks. *Complex Systems*, 2:321–355.

Bunn, D. W. (2000). Forecasting loads and prices in competitive power markets. *Proceeding of the IEEE*, 88(2):163–169.

Bunn, D. W. and Farmer, E. D., editors (1985). *Comparative Models for Electrical Load Forecasting*. John Wiley and Sons.

Charytoniuk, W., Chen, M. S., and Olinda, P. V. (1998). Nonparametric regresion based short-term load forecasting. *IEEE Transactions on Power Systems*, 13(3):725–730.

Chen, S., Cowan, C. F. N., and Grant, P. M. (1991). Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks*, 2(2):302–309.

REFERENCIAS

Chicco, G., Napoli, R., and Piglione, F. (2001). Neural networks for fast voltage prediction in power systems. *Power Tech. Proceedings, 2001 IEEE Porto*, 2:5.

Choueiki, M. H., Mount-Campbell, C. A., and Ahalt, S. C. (1997a). Building a 'quasi optimal' neural network to solve the short-term load forecasting problem. *IEEE Transactions on Power Systems*, 12(4):1432–1439.

Choueiki, M. H., Mount-Campbell, C. A., and Ahalt, S. C. (1997b). Implementing a weighted least squares procedure in training a neural network to solve the short-term load forecasting problem. *IEEE Transactions on Power Systems*, 12(4):1689–1694.

Chow, T. W. S. and Leung, C. T. (1996). Neural network based short-term load forecasting using weather compensation. *IEEE Transactions on Power Systems*, 11(4):1736–1742.

Chung, T. S. and Ying, F. (1998). An ANN-based vetwork equivalent approach for power system on-line voltage security assessment. *International Conference on Power System Technology POWERCON '98*, 2:1504 –1507.

Clements, K. A. and Davis, P. W. (1988). Detection and indentification of topology errors in electric power system. *IEEE Transactions on Power Systems*, 3(4):1748–1753.

Connor, J., Atlas, L., and Martin, D. (1992). Recurrent networks and NARMA modeling.

Connor, J., Martin, R., and Atlas, L. (1994). Recurrent neural networks and robust time-series prediction. *Neural Networks*, 5:240–254.

Connor, J. T. (1996). A robust neural network filter for electricity demand prediction. *Int. J. Forecast*, 15(6):437–458.

Connor, J. T., Atlas, L. E., and Martin, D. (1991). Recurrent neural networks and load forecasting. *Proceedings of the First International Forum on Applications of Neural Networks to Power Systems*, pages 22–25.

Costa, M., Pasero, E., Piglione, F., and Radasanu, D. (1999). Short term load forecasting using a synchronously operated recurrent neural network. *International Joint Conference on Neural Networks IJCNN '99*, 5(4):3478–3482.

Cottrell, M. and Letremey, P. (1995). Classification et analyse des correspondances au moyen de l'algoritme de Kohonen: Application à l'Ètude de donnèes Socio-Économiques. *Prépublication du SAMOS*, 42.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function.*Math Control Signal Syst.*, 2:303–314.

Darbellay, G. A. and Slama, M. (2000). Forecasting the short-term demand for electricity –do neural network stand a better chance? *Int. J. Forecast*, 16:71–83.

de la Villa Jaén, A. and Expósito, A. G. (2001). Modeling unknown circuit breakers in generalized state estimators. *Power Tech Proceedings, 2001 IEEE Porto*, 3:4.

del Brío, B. M. and Molina, A. S. (2001). *Redes Neuronales y Sistemas Borrosos*. Ra-Ma, segunda edition.

Demuth, H. and Beale, M. (2000). *Neural Network Toolbox for Use with Matlab. User's Guide*. The MathWorks, Inc., http://www.mathworks.com, 4 edition.

Dorffner, G. (1996). Neural networks for time series processing. *Neural Network World*, 4:447–468.

Drezga, I. and Rahman, S. (1998). Input variable selection for ANN-based short-term load forecasting. *IEEE Transactions on Power Systems*, 13(4):1238–1244.

Ejebe, G. C., Irisarri, G. D., Mokhtari, S., Obadina, O., Ristanovic, P., and Tong, J. (1996). Methods for contingency screening and ranking for voltage stability analysis of power system. *IEEE Transactions on Power Systems*, 11(1):350–356.

Ejebe, G. C., Meeteren, H. P. V., and Wollenberg, B. F. (1988). Fast contingency screening and evaluation for voltage security analysis. *IEEE Transactions on Power Systems*, 3(4):302–307.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14:179–211.

Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7(2-3):95–126.

REFERENCIAS

Engle, R. F., Mustafa, C., and Rice, J. (1992). Modeling peak electricity demand. *Int. J. Forecast*, 11:241–151.

Fan, J. Y. and McDonald, J. D. (1994). A real-time implementation of short-term load forecasting for distribution power systems. *IEEE Transactions on Power Systems*, 9(2):988–994.

Farlow, S. J., editor (1984). *Self-Organizing Methods in Modeling*. Marcel Dekker.

Fidalgo, J. N. and Lopes, J. A. P. (2001). ANN sensitivity analysis for identification of relevant features in security assessment. *Power Tech Conferente, PPT'2001*.

Fujiwara, R., Sakaguchi, T., Kohno, Y., and Suzuki, H. (1986). An intelligent load flow engine for power system planning. *IEEE Transactions on Power Systems*, PAS-3:302–307.

García-Lagos, F., Joya, G., Marín, F. J., and Sandoval, F. (1998a). Application of Hopfield neural networks to state estimation. *Proceeding of the International ICSC/IFAC Symposium on Neuronal Computation*, 1:405–411.

García-Lagos, F., Joya, G., Marín, F. J., and Sandoval, F. (1998b). A neural topology estimator for power systems. *Proceeding of the Fourth International Conference on Engineering Applications of Neural Networks June 10-12, 1998*, 4:245–252.

García-Lagos, F., Joya, G., Marín, F. J., and Sandoval, F. (1999). Study of the evolution of a stock market by means of Self-organizing maps of Kohonen. *Proceeding of the Sixth International Meeting on Advances in Computational Management*, pages 1–10.

García-Lagos, F., Joya, G., Marín, F. J., and Sandoval, F. (2000a). Hopfield neural networks for state estimation: Parameters, efficient implementation and results. *Elektrotechnik und Informationstechnik*, 1:4–7.

García-Lagos, F., Joya, G., Marín, F. J., and Sandoval, F. (2000b). A modular power system topology assessment based on Gaussian potential functions. *Proceedings of the IASTED International Conference. Power and Energy Systems*, Sep:217–223.

García-Lagos, F., Joya, G., Marín, F. J., and Sandoval, F. (2001). Neural networks for contingency evaluation and monitoring in power systems. *From Natural to Aritificial Neural Computation*, June:505–512.

García-Lagos, F., Joya, G., Marín, F. J., and Sandoval, F. (2002). Self-organizing maps for contingency analysis: Visual classification and temporal evolution. *Conference of the IEEE Industrial Electronics Society - Sevilla 2002*

García-Lagos, F., Joya, G., Marín, F. J., and Sandoval, F. (2003). Modular power system topology assessment using gaussian potential functions. *Aceptado para su publicación en IEE Proceedings on Generation, Transmission and Distribution*

Gjelsvik, A., Aam, S., and Holten, L. (1985). Hachtel's augmented matrix method – a rapid method improving numerical stability in power system static state estimation. *IEEE Trans. Power Apparatus and Systems*, 104(11):2987–2993.

Habiballah, I. O. and Quintana, V. H. (1991). Fast-decoupled rectangular co-ordinate state estimation with efficient data structure management. *IEE Procceedings-C*, 138(5):462–468.

Hagan, M. T. and Behr, S. M. (1987). The time series approach to short term load forecasting. *IEEE Transactions on Power Systems*, PWRS-2(3):785–791.

Hagan, M. T. and Menhaj, M. B. (1994). Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, 4(6):989–993.

Haida, T. and Muto, S. (1994). Regression based peak load forecasting using a transformation tecnique. *IEEE Transactions on Power Systems*, 9(4):1788–1794.

Harris, J. H., Kellie, G. H., Prewett, J. N., and Jerwis, P. (1976). Two implementations of state estimators for power systems. *Proceeding of the 1976 CIGRE, 32-06*, pages 1–11.

Harvey, A. and Koopman, S. J. (1993). Forecasting hourly electricity demand using time-varying splines. *J. American Stat. Assoc*, 88(424):1228–1236.

Haykin, S. (1994). *Neural Networks. A Comprehensive Foundation* Macmillan College Publishing Company Inc., New York.

Hecht-Nielsen, R. (1987). Counterpropagation networks. *Applied Optics*, 26(3):4979–4984.

Hippert, H. S., Pedreira, C. E., and Souza, R. C. (2001). Neural networks for short-term load forecasting: a review and evaluation. *IEEE Transactions on Power Systems*, 16(4):44–55.

REFERENCIAS

Ho, K. L., Hsu, Y. Y., and Yang, C. C. (1992). Short term load forecasting using a multilayer network with an adaptive learing algorithm. *IEEE Transactions on Power Systems*, 7(1):141–149.

Hopfield, J. J. (1982). Neural network and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Scie. USA*, 79:2554–2558.

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Procceeding of the National Academy of Sciences*, 81:3088–3092.

Hopfield, J. J. and Tank, D. W. (1985). Neural computation decisions in optimization problems. *Biol. Cybern.*, 52.

Hsu, Y.-Y. and Kuo, H.-C. (1992). Fuzzy-set based contingency ranking. *IEEE Transactions on Power System*, 3(3):1189–1194.

Huang, S. R. (1997). Short-term load forecasting using threshold autoregresive models. *IEE Proceeding on Generation, Transmission and Distribution*, 144(5):477–481.

Huber, P. J. (1981). *Robust Statistics*. John Wiley.

Hyde, O. and Hodnett, P. F. (1997). An adaptable automated procedure for short-term electricity load forecasting. *IEEE Transactions on Power Systems*, 12(1):84–93.

Infield, D. G. and Hill, D. C. (1998). Optimal smooting for trend removal in short term electricity demand forecasting. *IEEE Transactions on Power Systems*, 13(3):1115–1120.

Iwamoto, S. and Tamura, Y. (1978). A fast load flow method retaining nonlinearity. *IEEE Transactions on Power Apparatus and Systems*, PAS-97(5):1586–1599.

Jackson, J. E. (1991). *A Userś Guide To Principal Components*. John Wiley and Sons.

Jensen, C. A., El-Sharkawi, M. A., and II, R. J. M. (2001). Power system security assessment using neural networks: Feature selection using Fisher discrimination. *IEEE Transaction on Power Systems*, 16(4):757–763.

Jollife, I. T. (1986). *Principal Component Analysis*. Springer-Verlag.

Jordan, M. I. (1989). Serial order: A parallel, distributed processing approach. In Elman, J. L. and Rumelhart, D. E., editors, *Advances in Connectionist Theory: Speech*. Lawrence Erlbaum, Hillsdale.

Joya, G., Atencia, M. A., and Sandoval, F. (2002a). Hopfield neuronal networks for optimization: Study of the different dynamics. *Neurocomputing*, 43:219–237.

Joya, G., García-Lagos, F., Atencia, M. A., and Sandoval, F. (2002b). Artificial neural networks for energy management system. applicability and limitations of the main paradigms. *Approaches Connexionnistes en Sciences Economiques et de Gestion (ACSEG 2002)*, pages 129–143.

Juberias, G., Yunta, R., Garcia-Moreno, J., and Mendivil, C. (1999). A new ARIMA model for hourly load forecasting. *IEEE Transmission and Distribution Conference, New Orleans*, 1:314–319.

Khotanzad, A., Afkhami-Rohani, R., Lu, T. L., Abaye, A., Davis, M., and Maratukulam, D. J. (1997). ANNSTLF - a neural-network-based electric load forecasting system. *IEEE Transactions on Neural Nets*, 8(4):835–846.

Khotanzad, A., Afkhami-Rohani, R., and Maratukulam, D. (1998). ANNSTLF - artificial neural network short-term load forecaster - generation three. *IEEE Transactions on Power Systems*, 13(4):1413–1422.

Khotanzad, A., Hwang, R. C., Abaye, A., and Maratukulam, D. (1995). An adaptive modular artificial neural network hourly load forecaster and its implementation at electric utilities. *IEEE Transactions on Power Systems*, 10(3):1716–1722.

Kim, H. and Singh, C. (2002). Probabilistic security analysis using SOM and Monte Carlo simulation. *Power Engineering Society Winter Meeting*, 2:755–760.

King, D. D., El-Hawary, M. E., and El-Hawary, F. (1995). Optimal environmental dispatching of electric power sytems via an improved Hopfield neural network model. *IEEE Transactions on Power Systems*, 3.

Koglin, J. J. and Neisius, H. T. (1990). Treatment of topological erros in substations. *Proceeding of the 1990 Power Systems Computation Conference*, pages 1045–1053.

REFERENCIAS

Kohonen, T. (1990). The Self-organizing Map. *Proc. IEEE*, 78(9):1464–1480.

Lamedica, R., Prudenzi, A., Sforma, M., Caciotta, M., and Cencelli, V. O. (1996). A neural network based technique for short-term forecasting of anomalous load periods. *IEEE Transactions on Power Systems*, 11(4):1749–1756.

Le, N. Q. and Outhred, H. R. (1977). Identification and elimination of bad data and line errors from power system state estimators. *Proceeding of the 1977 IFAC Symposium (Melbourne)*, pages 459–463.

Lee, K. Y., Cha, Y. T., and Park, J. H. (1992). Short-term load forecasting using an artificial neural network. *IEEE Transactions on Power Systems*, 7(1):124–132.

Lee, S. and Kil, R. M. (1991). A Gaussian potential function network with hierarchically self-organizing learning. *Neural Networks*, 4:207–224.

Lemaître, C. and Thomas, B. (1996). Two applications of parallel processing in power system computation. *IEEE Transactions on Power Systems*, 11(1):246–253.

Lerma, I. J. (1990). Application of Monte Carlo method to power system reliability calculations. *Proceedings of the 25th Intersociety Energy Conversion Engineering Conference. IECEC-90*, 5:502–507.

Lo, K. L. and Abdelaal, A. K. I. (2000). Fuzzy logic based contingency analysis.*International Conference on Electric Utility Deregulation and Restructuring and Power Technologies 2000,City University, London*, pages 499–504.

Lo, K. L., Peng, L. J., Macqueen, J. F., Ekwue, A. O., and Cheng, D. T. Y. (1998). Fast real power contingency ranking using a Counterpropagation network. *IEEE Transactions on Power Systems*, 13(4):1259–1264.

Luan, W. P., Lo, K. L., and Yu, Y. X. (2000). ANN-based pattern recognition technique for power system security assessment. *International Conference on Electric Utility Deregulation and Restructuring and Power Technologies 2000, City University, London*, pages 197–202.

Lugtu, R. L., Hackett, D. F., Liu, K. C., and Might, D. D. (1980). Power system state estimation: Detection of topological errors. *IEEE PAS*, PAS-99(6):2406–2412.

Maghrabi, H., Raaee, J. A., and Mohandes, M. (1998). Contingency analysis of bulk power system using neural networks. *Proceeding of the International Conference on Power System Technology. POWERCON'98*, 2:1251–1254.

Mansour, A. E. and Eldin, H. A. N. (1987). Fast and robust bad data detection, identification and correction with subsequent direct recursive state estimation. *Proceeding of the 1987 PSCC (Cascais)*, pages 516–522.

Marín, F. J., García-Lagos, F., Joya, G., and Sandoval, F. (2000). A new short-term load forecaster with artificial neural network. *Proceeding of the IASTED International Conference. Power and Energy Systems.*, Sep:177–182.

Marín, F. J., García-Lagos, F., Joya, G., and Sandoval, F. (2002). Global model for short-term load forecasting using artificial neural networks. *IEE Proceedings on Generation, Transmission and Distribution*, 149(2):121–125.

Marín, F. J. and Sandoval, F. (1997). Short-term peak load forecasting: Statistical methods versus artificial neural networks. *in J. Mira, R. Moreno-Díaz and J. Cabestany (Eds.), Biological and Artificial Computation: From Neuroscience to Technology, Spriger-Verlag, LNCS*, 1240(2):1334–1343.

Marín, J. (1997). *Optimización de Redes Neuronales Artificiles Mediante Algoritmos Genéticos. Aplicación a la Predicción de Carga.* Tesis doctoral, Universidad de Málaga.

Matos, M. A., Hatziargyriou, N. D., and Lopes, J. A. P. (2000). Multicontingency steady state security evaluation using fuzzy clustering techniques. *IEEE Transactions on Power System*, 15(1):177–183.

Mazon, A. J., Zamora, I., Gracia, J., Sagastabeitia, K. J., and Saenz, J. R. (2001). Selecting ANN structures to find transmission faults. *IEEE Computer Applications in Power*, 14:44–48.

REFERENCIAS

Mbamalu, G. A. N. and El-Hawary, M. E. (1993). Load forecasting via suboptimal seasonal autoregressive models and iteratively reweighted least squares estimation. *IEEE Transactions on Power Systems*, 8(1):343–348.

McMenamin, J. S. and Monforte, F. A. (1998). Short-term energy forecasting with neural networks. *Energy J.*, 19(4):43–61.

Mili, L., Steeno, G., Dobraca, F., and French, D. (1999). A robust estimation method for topology error identification. *IEEE Transactions on Power Systems*, 14(4):1469–1476.

Miranda, V. and Saraiva, J. T. (1992). Fuzzy modelling of power system optimal load flow. *IEEE Transaction on Power Systems*, 7(2):843 –849.

Mohammed, O., Park, D., Merchant, R., Dinh, T., Tong, C., Azeem, A., Farah, J., and Drake, C. (1995). Practical experiences with an adaptive neural network short-term load forecasting. *IEEE Transactions on Neural Nets*, 10(1):254–265.

Monticelli, A. (1993). Modeling circuit breakers in weighted least squares estimation.*IEEE Transactions on Power Systems*, 8(3):0–0.

Monticelli, A. (2000). Electric power system state estimation. *Proceeding of the IEEE*, 88(2):262–282.

Monticelli, A., Garcia, A., and Saavedra, O. R. (1990). Fast decoupled load flow: Hypothesis, derivations, and testing. *IEEE Transactions on Power Systems*, 5(4):1425–1431.

Moody, J. E. and Darken, C. J. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1:281–294.

Mori, H. (1991). Application of a revised Boltzmann machive to topological observability analysis. *First Int. Forum on Application of Neural Network to Power Systems*, pages 281–287.

Mori, H. and Yuihara, A. (1998). Contingency screening using interval analysis in power systems. *Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium on*, 3:444 –447.

Nakawaga, H., Yayashi, Y., and Iwamoto, S. (1991). Neural network application to state estimation. *Compuation. First Intenational Forum on Application of N.N. to Power Systems*, pages 188–194.

on a Common Format for the Exchange of Solved Load Flow Data, W. G. (1973). Common data format for the exchange of solved load flow data. *IEEE Transactions on Power Apparatus and Systems*, PAS-92(6):1916–1925.

Papalexopoulos, A. D. and Hesterberg, T. C. (1990). A regression-based approach to short-term system load forecasting. *IEEE Transactions on Power Systems*, 5(4):1535–1547.

Park, D. C., El-Sharkawi, M. A., II, R. J. M., Atlas, L. E., and Damborg, M. J. (1991a). Electric load forecasing using an artificial neural network. *IEEE Transactions on Power Systems*, 6(2):442–449.

Park, J. H., Park, Y. M., and Lee, K. Y. (1991b). Composite modeling for adaptive short-term load forecasting. *IEEE Transactions on Power Systems*, 6(2):450–457.

Pereira, M. V. F., Maceira, M. E. P., Oliveira, G. C., and Pinto, L. M. V. G. (1992). Combining analytical models and Monte-Carlo techniques in probabilistic power system analysis. *IEEE Transactions on Power Systems*, 7(1):265–272.

Ramanathan, R., Engle, R., Granger, C. W. J., Vahid-Araghi, F., and Brace, C. (1997). Short-run forecasts of electricity loads and peaks. *Int. J. Forecast*, 13:161–174.

Refaee, J. A., Mohandes, M., and Maghrabi, H. (1999). Radial basis function networks for contingency analysis of bulk power systems. *IEEE Transactions on Power Systems*, 14(2):772–778.

Riedmiller, M. and Braun, H. (1993). A direct adaptive method for faster Backpropagation learning: the RPROP algorithm. *IEEE International Conference on Neural Networks*, 1:586 –591.

Robnik-Sikonja, M. and Kononenko, I. (1997). An adaptation of Relief for attibute estimation in regression. *Proceeding of the 14th International Conference on Machine Learning ICML'97*.

REFERENCIAS

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. *Parallel Distributed Proccessing*, 1:310–362.

Sadownik, R. and Barbosa, E. P. (1999). Short-term forecasting of industrial electricity consumption in Brazil. *Int. J. Forecast*, 18:215–224.

Sandoval, F., García-Lagos, F., and Joya, G. (2002). Design of artificial neural networks using evolutionary computation. *Approaches Connexionnistes en Sciences Economiques et de Gestion (ACSEG 2002)*, pages 181–200.

Santos, J. R., Expósito, A. G., and Ramos, J. L. M. (1999). Distributed contingency analysis: Practical issues. *IEEE Transactions on Power Systems*, 14(4):1349–1354.

Saraiva, J. T., Miranda, V., and Pinto, L. M. V. G. (1996). Generation / transmission power system reliability evaluation by Monte-Carlo simulation assuming a fuzzy load description. *IEEE Transaction on Power Systems*, 11(2):690–695.

Sargunaraj, S., Gupta, D. P. S., and Devi, S. (1997). Short-term load forecasting for demand side management. *IEE Proceeding on Generation, Transmission and Distribution*, 144(1):68–74.

Senjyu, T., Takara, H., Uezato, K., and Funabashi, T. (2002). One-hour-ahead load forecasting using neural network. *IEEE Transactions on Power Systems*, 17(1):113–118.

Sidhu, T. S. and Cui, L. (2000). Contingency screening for steady-state security analysis by using FFT and artificial neural networks. *IEEE Transactions on Power Systems*, 15(1):421–426.

Simoes Costa, A. and Leao, J. A. (1993). Identification of topology errors in power systems state estimation. *IEEE Transactions on Power Systems*, 8(4):1531–1538.

Singh, H. and Alvarado, F. (1995). Network topology determination using least absolute value state estimation. *IEEE Transactions on Power Systems*, 10(3):1159–1165.

Singh, N. and Glavitsch, H. (1991). Detection and identification of topological errors in online power system analysis. *IEEE Transactions on Power Systems*, 6(1):324–331.

Slutsker, I. W. and Mokhtari, S. (1995). Comprehensive estimation in power systems: State, topology and parameter estimation. *Proceeding of the 1995 American Power Conference*, pages 1–7.

Soliman, S. A., Persaud, S., El-Nagar, K., and El-Hawary, M. (1997). Application of least absolute value parameter estimation based on linear programming to short-term load forecasting. *Elect. Power and Energy Systems*, 19(3):209–216.

Souza, J. C. S., Leite da Silva, A. M., and Alves da Silva, A. P. (1996). Data debugging for real-time power system monitoring based on pattern analysis. *IEEE Transactions on Power Systems*, 11(3):796–803.

Souza, J. C. S., Leite Da Silva, A. M., and Alves Da Silva, A. P. (1997). Data visualization and identification of anomalies in power systems state estimation using artificial neural network. *IEE Proceeding on Generation, Transmission and Distribution*, 144(5):445–455.

Souza, J. C. S., Leite da Silva, A. M., and Alves da Silva, A. P. (1998). Online topology determination and bad data suppression in power system operation using artificial neural networks. *IEEE Transactions on Power Systems*, 13(3):796–803.

Srinivasan, D., Liew, A. C., and Chang, C. (1994). Forecasting daily load curves using a hybrid fuzzy-neural approach. *IEE Proceeding on Generation, Transmission and Distribution*, 141(6):561–567.

Srinivasan, D., Tan, S., Chang, C., and Chan, E. K. (1999). Parallel neural network-fuzzy expert system strategy for short-term load forecasting: System implementation and performance evaluation. *IEEE Transactions on Power Systems*, 14(3):1100–1106.

Srivastava, S. C. and Veankataraman, D. (1997). Short-term load forecasting using recurrent neural networks. *Proceeding of the 4th Conference on Advances in Power System Control, Opearation and Management, APSCOM-97, Hong Kong*, pages 145–150.

Swarup, D. S. and Satish, B. (2002). Integrated ANN. approach to forecast load. *IEEE Computer Application in Power*, pages 46–51.

Sympson, P. K. (1990). *Artificial Neural Systems. Fundations, Paradigms, Application, and Implementation*. Pergamon Press.

Taylor, J. W. and Majithia, S. (2000). Using combined forecasts with changing wights for electricity demand profiling. *J. Oper. Res. Soc.*, 51(1):72–82.

Teixeira, M. A., Zaverucha, G., da Silva, V. N. A. L., and Ribeiro, G. F. (1999). Recurrent neural gas in electric load forecasting. *International Joint Conference on Neural Networks*, 5:3468–3473.

Tibshirani, R. (1996). A comparison of some error estimates for neural network models. *Neural Computation*, 8:152–163.

Vaahedi, E., Fuchs, C., Xu, W., Mansour, Y., Hamadanizadeh, H., and Morison, G. (1999). Voltage stability contingency screening and ranking. *IEEE Transaction on Power Systems*, 14(1):256–265.

Vermaak, J. and Botha, E. (1998). Recurrent neural networks for short-term load forecasting. *IEEE Transactions on Power Systems*, 13(1):126–132.

Vinod Kumar, D. M., Srivastava, S. C., Shah, S., and Mathur, S. (1996). Topology processing and static state estimation using artificial neural networks. *IEE Proceeding in generation, transmission and distribution*, 143(1):99–105.

Warwick, K., Ekwue, A., and Aggarwal, R., editors (1997). *Artificial Intelligence Techniques in Power Systems*. IEE Power Engineering Series 22. The Institution of Electrical Engineers.

Wilkosz, K. (1995). Critical evaluation of methods for power system topology verification. *Procceeding of the IEEE/KTH Stockholm power Tech. Conference. Stockholm, Sweden*, pages 152–157.

Wood and Wollenberg (1994). *An Introduction to State Estimation in Power System*, chapter 12, pages 391–439. John Whiley.

Wu, F. F. and Wen-Hsiung, E. L. (1989). Detection of topology errors by state estimation. *IEEE Transaction on Power Systems*, 4(1):176–183.

Yan, H. H., Chow, J. C., Kam, M., Fischl, R., and Sepich, C. R. (1991). Hybrid expert system neural network hierarchical architecture for classifying power system contingencies.

Proceedings of the First International Forum on Applications of Neural Networks to Power Systems, pages 76–82.

Yang, H. T. and Huang, C. M. (1998). A new short-term load forecasting approach using Self-organizing fuzzy ARMAX models. *IEEE Transactions on Power Systems*, 13(1):217–225.

Yang, H. T., Huang, C. M., and Huang, C. L. (1996). Identification of ARMAX model for short term load forecasting: An evolutionary programming approach. *IEEE Transactions on Power Systems*, 11(1):403–408.

Yoh-Han, P. (1989). *Adaptive pattern Recognition and Neural Networks* Addison-Wesley.

Yu, Z. (1996). A temperature match based optimization method for daily load prediction considering DLC effect. *IEEE Transactions on Power Systems*, 11(2):728–733.

Zarco Periñán, P. J. and Gómez Espósito, A. (1999). *Estimación de Estado y de Parámetros en Redes Eléctricas*. Universidad de Sevilla. Secretariado de Publicaciones.

Zivanovic, R. and Petroianu, A. (1994). Application of neural computing to the state estimation problem. *ICPST'94. Bijing, China*, pages 87–92.