

UNIVERSIDAD AUTONOMA METROPOLITANA
DIVISION DE CIENCIAS BASICAS E INGENIERIA

CLASIFICACION DE POTENCIALES DE ACCION
CON UNA RED NEURONAL
QUE EMPLEA RETROPROPAGACION

TESIS
que para obtener el título de
MAESTRO EN INGENIERIA BIOMEDICA
PRESENTA

JORGE QUIZA TOMICH

DIRECTOR DE TESIS:



Casa abierta al tiempo

IZTAPALAPA
BIBLIOTECA

DR. ISMAEL ESPINOSA ESPINOSA

SINODALES:

M. en I. SONIA CHARLESTON VILLALOBOS

DR. ISMAEL ESPINOSA ESPINOSA

M. en C. RAFAEL GODINEZ FERNANDEZ.

MEXICO, D. F., JULIO DE 1991.

Jorge Lencina B

Este trabajo no hubiese sido posible sin el apoyo constante, desinteresado y generoso del Dr. Ismael Espinosa Espinosa. Por eso, a él está dedicado.

A mis amigos.

Este trabajo se realizó en el Laboratorio de Cibernética de la Facultad de Ciencias de la U.N.A.M., a cargo del Dr. Ismael Espinosa Espinosa, y, parcialmente, en el Centro de Cómputo de la UAM - Iztapalapa. Fue apoyado en parte por la DGAPA, proyecto IN-206189

INDICE

Introducción	1
1. Reconocimiento de patrones y redes neuronales.....	3
1.1 Inferencia y búsqueda asociativa en paralelo en bases de datos.....	6
1.2 Reconocimiento de patrones.....	8
1.2.1 Objetivos del reconocimiento de patrones.....	15
1.3 Redes neuronales artificiales.....	16
1.3.1 Características de las redes neuronales.....	17
1.3.2 Elementos procesadores.....	18
1.3.3 De la función de activación de los nodos.....	19
1.3.4 Arquitectura y dinámica.....	24
1.3.5 Aprendizaje.....	25
1.4 Redes de capas múltiples con propagación progresiva y retropropagación del error.....	26
1.4.1 Memoria distribuida.....	29
1.4.2 Aprendizaje supervisado en sistemas lineales.....	32
1.4.3 Perceptrones.....	35
1.4.4 Regla delta generalizada.....	36
2. Métodos	42
2.1 Métodos tradicionales de clasificación.....	43
2.2 Método empleado.....	45
2.2.1 De los datos de entrada.....	45
2.2.2 De las características de la red.....	46
2.3 Detección de impulsos en una traza nerviosa.....	52
3. Resultados	57
3.1 Red utilizada.....	57
3.2 Resultados del aprendizaje.....	61
3.3 Representación interna y generalización.....	73
3.4 Clasificación de impulsos en una traza nerviosa.....	80
4. Discusión y conclusiones.....	85
Referencias	92
Apéndices	99
Programas desarrollados.....	99
A. Programa para entrenar la red.....	100
B. Programa para detección y clasificación.....	103

INTRODUCCION

El análisis teórico y modelado de las funciones cerebrales de los organismos complejos ha dado origen a las neurociencias teóricas, cuya meta sería la de ampliar nuestros conocimientos acerca de las estrategias utilizadas por los cerebros biológicos para procesar información. Sin embargo, dichos propósitos distan mucho de ser alcanzados. Por una parte, la investigación teórica está basada ya bien en soluciones extremadamente abstractas para los problemas experimentales o bien en la descripción de sistemas limitados, con muy pocas neuronas, es decir, hace falta cerrar la brecha que lleva de la descripción determinista de arreglos neuronales relativamente simples a la descripción estadística de grupos grandes de neuronas. Por otro lado, se reconoce que el obstáculo para la adquisición de conocimientos nuevos es de carácter conceptual más bien que técnico; lo que hace falta no es tanto la tecnología sino la agudeza analítica que nos permita la descripción del funcionamiento de los cerebros biológicos con la exactitud suficiente.

Un ejemplo ilustrativo de tal carencia es evidente en los resultados conseguidos en los registros extracelulares con electrodos múltiples, donde la pericia tecnológica ha hecho posible la obtención de tal cantidad de datos, que aún no se sabe como utilizarlos plenamente. Si los seres vivos son procesadores distribuidos en paralelo, ¿cómo analizar e interpretar la compleja cantidad de datos logrados con estos experimentos?

El objetivo del presente trabajo se centra en el estudio de una técnica para ayudar a hacer más fácil el análisis automático de los datos obtenidos durante el registro de la actividad de un grupo de neuronas, ya sea conseguido con el empleo de microelectrodos múltiples que abarcan regiones más o menos distantes o con un solo electrodo de metal que registra una región pequeña. Dicha técnica debe permitir la separación de los diferentes potenciales de acción que aparecen en tales registros, con base en su morfología y se lleva a cabo empleando una de las

aplicaciones prácticas más inmediatas que han surgido de los resultados alcanzados por las neurociencias teóricas mismas : el uso de las redes conexionistas en el reconocimiento de patrones. Concretamente, se analiza la factibilidad de entrenar una red neuronal artificial de capas múltiples con el algoritmo de retropropagación, utilizando un conjunto de espigas nerviosas de morfologías típicas, de manera que al finalizar el aprendizaje se puedan clasificar los impulsos semejantes presentes en un tren de impulsos múltiple.

Cabe señalar que la (re)aparición de las redes neuronales artificiales puede interpretarse como un cambio en los enfoques de la ciencia contemporánea: si las neurociencias han dado una orientación multidisciplinaria a la interpretación de sus resultados experimentales, también otros campos han debido ajustar sus perspectivas de acuerdo a diferentes presiones interdisciplinarias. Así, los modelos de la psicología teórica han abandonado la idea que el cerebro es una especie de computador de von Neumann; el campo de la inteligencia artificial, luego de haber obtenido a los sistemas expertos como su producto final, ha reconocido la necesidad de nuevas arquitecturas computacionales; en física, el interés en los fenómenos colectivos no lineales, contemplando los comportamientos con equilibrio global, las transiciones de fase, oscilaciones no lineales y caos, ha llevado de forma natural a las redes neuronales; igualmente, el creciente interés de la comunidad matemática por el estudio de los sistemas con dinámicas no lineales ha derivado hacia las redes neuronales y, finalmente, la necesidad de resolver problemas no paramétricos en ambientes no estacionarios y ruidosos, ha hecho que la ingeniería adopte nuevos modelos.

En el primer capítulo se pasa revista a algunos conceptos allegados con el reconocimiento de patrones y se hace una introducción a las redes neuronales artificiales, incluyendo una deducción detallada del algoritmo de retropropagación. El método utilizado se describe en el segundo y los resultados conseguidos se presentan en el tercero. Finalmente, en el cuarto capítulo se discuten los resultados y se examinan sus implicaciones posibles.

CAPITULO 1

RECONOCIMIENTO DE PATRONES Y REDES NEURONALES

Cuando se habla del "computador del futuro", es casi un lugar común pensar que será paralelo y bastante tolerante a las fallas. Sin embargo, el diseño de tal máquina ha sido arduo y se hubiese creído imposible si no fuese porque el cerebro es una prueba viviente de que el procesamiento en paralelo y tolerante a las fallas es no sólo posible sino también sumamente poderoso.

Los sistemas biológicos han desarrollado habilidades impresionantes para analizar y reconocer patrones. Un ejemplo son los mecanismos de reconocimiento a nivel molecular que permiten que las células se identifiquen entre sí durante el desarrollo y den origen a órganos y organismos. El más alto grado de sofisticación de los procesos moleculares de reconocimiento ha sido alcanzado por el sistema inmunológico, con su sorprendente capacidad para analizar y reconocer estructuras moleculares. Por otro lado, la evolución del sistema nervioso ha proporcionado los cimientos para la instrumentación de estrategias alternativas para analizar y reconocer patrones. Es claro que la capacidad de las redes neuronales biológicas para procesar y almacenar información supera en mucho a la de otros sistemas biológicos o tecnológicos, y por lo tanto una de las principales tareas de la neurobiología ha sido la de investigar los algoritmos de los procesos de codificación, almacenamiento y recuperación de información en el sistema nervioso. A pesar del avance alcanzado en las últimas décadas en la comprensión de cómo codifica el cerebro los diferentes estímulos sensoriales y descompone patrones complejos en constituyentes más simples, se encuentran dificultades extraordinarias cuando se intenta comprender como las representaciones neuronales de ciertas características particulares se relacionan unas con otras hasta formar una codificación de patrones y cómo se comparan éstos con patrones

nuevos. Tales procesos tienen lugar a niveles donde el substrato anatómico alcanza grados de complejidad que están más allá de nuestras capacidades analíticas actuales. Más desconsolador que las mismas dificultades técnicas inherentes al gran número de elementos interactuantes en las redes neuronales es la noción que aún con una descripción detallada de la circuitería no podremos inferir los algoritmos del procesamiento.

Una posible solución a este problema analítico puede buscarse en el hecho de que los sistemas biológicos son el producto de desarrollos onto y filogenéticos. En ambos procesos de evolución, las estructuras más complejas han surgido de precursoras más simples, lo que nos da la posibilidad de estudiar los principios de organización de sistemas simples y derivar, de estos, hipótesis de trabajo para el análisis de los sistemas más complejos. El sistema nervioso usa su capacidad para analizar y reconocer patrones no solamente para adaptarse al medio ambiente externo sino también para ajustar unos con otros sus propios bloques constitutivos durante su desarrollo, es decir, parece hacer uso de sus capacidades de procesar información para promover su auto-organización en el transcurso de su evolución (Singer, 1988). Las capacidades de adaptarse a un medio ambiente cambiante y de aprender de las experiencias pasadas pueden considerarse como las características esenciales de un sistema cognoscitivo (Serra y Zanarini, 1990), entendiéndose como tal sistema todo aquél que sea capaz de efectuar tareas complejas que recuerden, en un sentido general, las características de la actividad mental humana, sin que esto implique que la reproduzcan con precisión.

Una de las líneas de investigación de la actividad cerebral revitalizada por los recientes avances en la neurobiología ha sido la de las redes neuronales. Según esta perspectiva, resulta útil construir y estudiar un modelo de red neuronal para entender el funcionamiento del cerebro, puesto que el comportamiento de un modelo puede ayudar a elucidar aspectos fundamentales en el procesamiento de información neuronal. Por supuesto, una red modelo será una aproximación muchísimo más simple que un cerebro

real, pero su análisis matemático puede ilustrar los posibles mecanismos de procesamiento de información neuronal y tales mecanismos, a su vez, pueden emplearse para construir un modelo más realista (Amari, 1988).

Al modelar estas redes neuronales, deben de tenerse en cuenta algunas de las características de las neuronas reales. Una descripción muy idealizada de una neurona la muestra compuesta de sitios receptores de información proveniente de otras células, las dendritas, así como de un axón encargado de transmitir información hacia otras células. Cada neurona tiene un nivel de actividad, que puede ser descrito cuantitativamente por un número, indicativo de la frecuencia a la que la neurona dispara impulsos a través de su axón. La conexión de un axón a las dendritas de otras neuronas se conoce como sinapsis y puede ser excitatoria o inhibitoria, dependiendo del influjo que ejerza sobre la actividad de la siguiente neurona.

En un cerebro humano típico hay de unas 10^{11} a 10^{12} neuronas. Algunas tienen unas cuantas sinapsis, mientras que otras tienen cientos de miles. Unos cuantos miles es lo típico en la corteza cerebral, por lo tanto, se puede pensar que existen unas 10^{15} sinapsis en un cerebro y, suponiendo que cada una de estas pueda tener dos estados (por ejemplo, excitatorio o inhibitorio), se tendría que puede existir una cantidad de 2 elevado a la 10^{15} configuraciones cerebrales diferentes.

Se cree que la memoria a largo plazo y el aprendizaje se producen por la formación de nuevas sinapsis, aunque los detalles no son conocidos exactamente. Según Denker (1986), algunos puntos dignos de tenerse en cuenta son los siguientes:

- ** Las neuronas son lentas (frecuencia de disparo de algunos cientos de Hertz).
- ** Cada neurona disipa muy poca potencia (alrededor de veinte vatios para todo el cerebro).
- ** El cerebro es extremadamente tolerante a las fallas: en el cerebro adulto probablemente mueren diariamente muchas neuronas sin ninguna consecuencia apreciable en su funcionamiento (Smith, 1986).

1.1. INFERENCIA Y BUSQUEDA ASOCIATIVA EN PARALELO EN BASES DE DATOS.

La noción que el cerebro opera de acuerdo con principios asociativos no es nueva: muchos avances culturales se basan en medios instrumentales tales como el lenguaje, la escritura, los sistemas y notaciones lógico-matemáticos, coreográficos y musicales, que se han desarrollado durante mucho tiempo y obedecen a sus propios mecanismos y la gente que los emplea debe aprenderlos primero como representaciones internas en su mente. Por otro lado, los procedimientos para resolver un problema pueden activarse sin tener que hacer referencia a procesos neuronales. Sin embargo, en tal caso el gran número de reglas lógicas involucrado puede hacer creer que el cerebro está lleno de circuitos que las ejecutan, aunque es más probable que sólo algunos procedimientos sean memorizados, mientras que la mayoría de las reglas es ejecutada con herramientas o ayudas artificiales (Rosenfield, 1988).

Al tratar de definir las características de una máquina "inteligente", una de las ideas más socorridas es aquella implicada en el concepto de los sistemas expertos, en los cuales se dan una serie de preguntas y recomendaciones para diferentes situaciones. La función que debe cumplir la máquina consiste en la búsqueda de cierto número de condiciones, para lo cual necesita una memoria de algún tipo. Si la información manejada tiene valores discretos, su almacenamiento es más efectivo en localidades diferentes de memoria. Para encontrar rápidamente un valor con base en su contenido existen soluciones programables (hash-coding) o circuitos especiales (memorias direccionables por contenido, CAM) (Kohonen, 1987). No obstante, lo más común es que la información consista de diferentes datos de alguna clase y de sus relaciones con otros subconjuntos de datos. El conocimiento adquirido en estas bases de datos se lleva a cabo y se maneja por medio de encadenamientos largos de tales relaciones, que ocurren cuando las diferentes asociaciones se traslapan. Este tipo de bases de datos sugiere las siguientes manipulaciones simples:

- ** Confrontar, ya sea en paralelo o de otra manera lo suficientemente rápida, un número de argumentos buscado con todos los valores almacenados en la memoria y marcarlos provisionalmente; y,
- ** Analizar la marcación efectuada y entregar los resultados que satisfagan todas las condiciones impuestas.

Cuando las representaciones son únicas, la identificación existe solamente cuando hay una similitud completa entre el espécimen almacenado y el argumento buscado. En estas condiciones, las computadoras convencionales pueden usarse con programas adecuados y es posible, incluso, hacer la identificación con partes de los diferentes objetos, con base en las relaciones de magnitudes numéricas. Sin embargo, las señales naturales tales como las imágenes visuales, la voz y las medidas físicas raras veces son únicas o tan siquiera numéricas y, generalmente, contienen una apreciable cantidad de ruido. Por supuesto, sus valores numéricos pueden almacenarse fácilmente en una memoria, pero una búsqueda en paralelo, basada en un reconocimiento aproximado, es una tarea que requiere circuitería especial, puesto que cada localidad de memoria debería estar provista con circuitos aritméticos capaces de analizar relaciones aproximadas de magnitud.

Es aquí donde una nueva generación de circuitos asociativos, conocidos como redes neuronales artificiales, se muestra bastante prometedora. Los argumentos buscados se dan como una condición inicial a la red y la solución para la "respuesta" ocurre cuando el circuito alcanza una especie de mínimo energético. Con el fin de entender las posibilidades y limitaciones de esta clase de circuitos se puede establecer un paralelo entre un circuito neuronal y uno digital. Siguiendo a Kohonen (1988, 1988a), algunos hechos que deben considerarse son:

- ** Los principios de los circuitos lógicos o digitales no se aplican a los sistemas neuronales biológicos. Por una parte, si se aplicara el principio computacional de asincronicidad, la duración de los impulsos nerviosos debería ser variable con el fin de mantener un valor binario durante un periodo indefinido. Si, por otra parte, se piensa en el principio de

sincronicidad, se requeriría de un reloj global para sincronizar los impulsos nerviosos.

- ** Ni las neuronas ni las sinapsis son elementos de memoria biestable. La evidencia fisiológica sugiere que las neuronas funcionan como integradores análogos no lineales y que la eficacia de las sinapsis cambia gradualmente.
- ** En la computación neuronal no se dan ni instrucciones de máquina ni códigos de control.
- ** Los circuitos cerebrales no implementan computación recursiva alguna y, por lo tanto, no son algorítmicos.

Estas consideraciones llevan a la conclusión que una red neuronal artificial debe tener las siguientes características:

- ** Sus elementos constitutivos son análogos y, debido a ello, la representación y comparación de valores numéricos sólo puede definirse con un grado de exactitud más bien bajo.
- ** Una gran cantidad de relaciones que se asemejan apenas aproximadamente con el argumento buscado pueden activarse en la memoria y, puesto que estos conflictos no pueden resolverse totalmente sino tan sólo minimizarse, el estado final hacia el cual converja el circuito representará una especie de respuesta óptima (generalmente en el sentido de la métrica euclidiana).
- ** El estado asintótico, o respuesta, que representa el resultado de la búsqueda, no tiene alternativas. Por esto, no es posible, en general, encontrar un conjunto completo de soluciones ni siquiera un número de los mejores candidatos a soluciones. Tampoco es seguro que el circuito converja al óptimo global, lo más usual es que alcance un mínimo local, lo que en la práctica suele ser aceptable.

1.2. RECONOCIMIENTO DE PATRONES

Las tareas agrupadas bajo la denominación de reconocimiento de patrones son lo suficientemente amplias como para abarcar tanto a actividades humanas tales como la sensación, la percepción y el juicio cognoscitivo como a su simulación mecánica por medio de computadoras. Al igual que lo ocurrido con la cibernética, las técnicas de mecanización se benefician por el estudio de los prototipos humanos, y, a su vez, los conocimientos biomédicos y

sicológicos lo hacen de los esfuerzos por desarrollar métodos mecánicos de simulación.

Antes de continuar, conviene preguntarse qué es un patrón. Cuando vemos un dibujo que representa a un río o a un perro, estamos de alguna forma asociando la imagen que presenciarnos con otros casos de ríos o perros, reconociendo una similitud entre ellas. Esta situación puede considerarse como un caso especial de lo que Wittgenstein decía acerca de "ver algo₁ como algo₂" (Wittgenstein, 1953). Según él, el verbo "ver" tiene dos acepciones, una cuando decimos, por ejemplo, "Veo este perro" y otra cuando decimos "Veo este dibujo como un perro". Lo que llamamos patrón corresponde a algo₂ en su enunciación previa. Algo₂ no existe en el mismo nivel de lenguaje que algo₁ en un dibujo, es una construcción ideal y es el resultado de la cooperación entre estímulos externos y cierta clase de actividad mental, que asocia un caso con otro similar o relacionado.

Si se sustituye "ver" por "reconocer" y "algo₂" por "patrón" en la frase citada arriba, se puede hablar más específicamente de reconocimiento de patrones. Si se toman ejemplos del habla común, se puede decir que reconocer es, generalmente, la identificación de un objeto como miembro de alguna familia o clase que ya se conoce, aunque no es absolutamente necesario que la clase sea conocida de antemano (Watanabe, 1984). De hecho, se puede asignar un objeto a una clase hasta ese momento desconocida. El término reconocimiento de patrones se refiere a ambas actividades, asignación a clases conocidas y formación de clases nuevas, aunque muchas veces se utiliza para referirse al caso más restringido que no incluye la formación de clases nuevas.

Podría caracterizarse el reconocimiento de patrones como el discernir una cosa o forma en un campo de percepciones compuesto por muchas partes o ejemplos. Así, podemos "ver uno en la multitud" al reconocer un objeto en un conjunto de partes, como al identificar una cara en un conjunto de puntos de diferentes texturas, o al llamar con un nombre único a objetos que pueden tener diferentes individualidades, por ejemplo al referirnos a un "libro".

El reconocimiento de patrones, en la acepción restringida mencionada arriba, consiste, básicamente, en la capacidad de identificar un objeto como perteneciente a una clase particular, una vez que se han conocido algunos paradigmas de dicha clase.¹ En este sentido, el reconocimiento de patrones se identifica con la inferencia inductiva en su función esencial, es decir, inferir una generalización a partir de algunos casos concretos (Watanabe, 1984).

Las primeras máquinas "asociativas" se diseñaron para la interpretación estadística de patrones, tales como las señales de voz, escritura y mapas meteorológicos. La tarea consistía en clasificar o categorizar los datos de entrada más bien que en buscar una información almacenada. Con el tiempo, sus métodos, basados en los computadores de propósito general, se hicieron cada vez más heurísticos. El reconocimiento de patrones tradicional hace énfasis en el tratamiento estadístico de los patrones y en el uso de la lingüística matemática para la clasificación de aquéllos con estructura sintáctica. Sin embargo, los objetivos siempre han sido los de instrumentar funciones sensoriales y cognoscitivas para interpretar las observaciones a diferentes niveles de abstracción, en otras palabras, hacer que nuestras máquinas tengan capacidades similares a las nuestras en cuanto al procesamiento de información y patrones se refiere.

Al analizar la manera como los seres vivos procesan información, surgen dos preguntas básicas:

- ** ¿Cómo está representado el conocimiento en el cerebro y cómo puede representarse en una máquina?
- ** ¿Cuál es la naturaleza del aprendizaje?

La primera da origen dos problemas colaterales, a saber:

¹. El término paradigma se emplea aquí en el sentido de un objeto individual que funge como prototipo de una clase, es decir, como un ejemplo típico y no en el del contenido ideológico de ciertas teorías, cuyo uso ha sido popularizado por las obras de Thomas Kuhn.

- ** ¿Cómo es convertida la información capturada por los sentidos en las formas abstractas que usa el cerebro?
- ** ¿Hasta qué punto las estructuras informativas construidas artificialmente, tales como el lenguaje y las matemáticas, reflejan la estructura y organización interna del conocimiento y de los medios, mentales o mecánicos, para su empleo?

La segunda de las preguntas iniciales conduce al siguiente interrogante:

- ** ¿Cuál es la relación entre las propiedades de la información por ser aprendida y el estado interno del aprendiz?

Resnikoff (1989) ha señalado que una respuesta satisfactoria a estas preguntas debe incluir una caracterización de los requerimientos de la base de conocimientos y de las reglas de procesamiento del sujeto que aprende como una función del conocimiento que está siendo adquirido.

El reciente auge de los modelos de procesamiento distribuido en paralelo ha conducido al estudio de modelos basados en el uso de una gran cantidad de procesadores simples, interconectados de manera que recuerdan a las redes nerviosas. Tales modelos exhiben capacidades poderosas de aprendizaje, memorización y asociación para información en forma de patrones. Este procesamiento a nivel subsimbólico parece ser adecuado para comprender algunas características de los fenómenos de la percepción e incluso de los del conocimiento. Como se indicaba antes, es posible utilizar estudios neurobiológicos o psicológicos como punto de partida para nuevos modelos teóricos y, de igual manera, cuando los resultados obtenidos con éstos modelos imitan a los del comportamiento humano, se puede pensar que la red biológica está compuesta por sistemas similares a los del modelo. Un aspecto importante de esta disciplina es su preocupación por el manejo de información en forma de patrones. La introducción del procesamiento distribuido paralelo permite un aprendizaje y clasificación adaptables de una manera que no era posible antes.

Se puede pensar en el reconocimiento de patrones restringido en términos de un mapeo correcto de un patrón desde un espacio de

patrones a uno de membresía de clases (Pao, 1989). El reconocimiento de patrones por los seres vivos puede verse como un mapeo opaco, es decir, un nuevo patrón puede ser clasificado correctamente por el observador sin que este sea consciente del proceso ejecutado. Por otra parte, el reconocimiento de patrones hecho por una máquina consiste en reemplazar este mapeo opaco por uno transparente, que pueda ser descrito precisamente a la máquina.

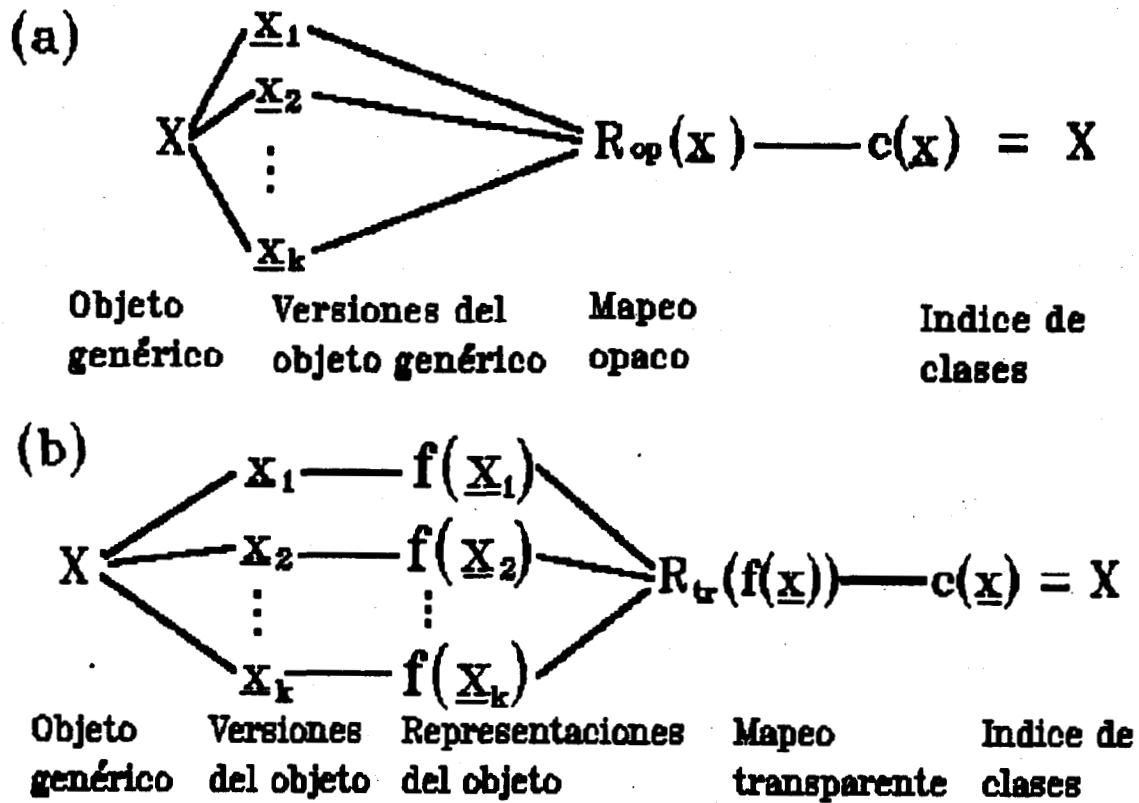


Fig. 1.1. Representación esquemática de los diferentes pasos en el reconocimiento de patrones tal como ocurre en: (a) los organismos vivos y (b) en las máquinas.

En la figura 1.1(a) se ve cómo un objeto X puede tener diferentes versiones, \underline{x} . En la naturaleza hay un mapeo opaco, $R_{op}(\underline{x})$, que proyecta a estos patrones en la clase correspondiente a X , $c(\underline{x})$, es decir, cualquier versión \underline{x} es reconocida y clasificada en la categoría del patrón X . Por otra parte, para efectuar el reconocimiento con una máquina, el mapeo opaco debe

ser substituido por un procedimiento explícito. Tal como aparece en la figura 1.1(b), este mapeo puede representarse como $R_{tr}(f(x)) \equiv c(x) = X$.

Este último procedimiento implica dos pasos. En el primero, se describen las manifestaciones específicas del objeto X en términos de características apropiadas, esto es, vamos de x a $f(x)$. Luego, en el segundo paso, la máquina ejecuta un proceso para conseguir un mapeo transparente, $R_{tr}(f(x))$, e ir de $f(x)$ a $c(x) = X$. De estas dos operaciones, la primera es la más difícil de planear. Una vez que $f(x)$ es conocida, se dispone de una guía teórica para sintetizar las transformaciones necesarias para obtener $c(x)$. Por el contrario, no existe ningún conocimiento a priori para obtener $f(x)$. Esto significa que, usualmente, no se sabe que aspectos de la representación son afines con el acto de reconocer patrones.

La elección de las características para describir los objetos -que pueden ser conceptos, rasgos físicos, situaciones o eventos- es una tarea crítica y difícil, pero es un preprocesamiento esencial la mayoría de las veces en el reconocimiento de patrones por máquinas. En cierta forma, no hay una elección correcta o equivocada mientras exista información suficiente en el conjunto de características elegido para describir los objetos. Sin embargo, una elección apropiada significa un mapeo $R_{tr}()$ más sencillo. En el reconocimiento de patrones adaptivo, la determinación de $f(x)$ sigue siendo una de las partes cruciales, y muchas veces es posible descubrir, adaptivamente, cuáles características son necesarias y cuáles no.

En lo que respecta a los sistemas biológicos procesadores de información, existe un principio general de omisión selectiva de información: los órganos de los sentidos organizan sus entradas de tal manera que simplifican la información que es entregada a los centros superiores de procesamiento; esta información, en gran medida, predetermina las formas de los patrones que pueden detectarse; a su vez, los centros superiores reducen la cantidad de información que será procesada en los siguientes estadios, al organizarla en formas más abstractas y universales, de tal forma

que los datos provenientes de los diferentes sentidos pueden mezclarse entre sí junto con la información generada internamente. Es importante resaltar el hecho de que esta omisión de información se dé no sólo al nivel de datos sino al de los procedimientos lógicos, los que aunque, aparentemente, deberían aplicarse a todos los datos no lo son, debido a un proceso sistemático de exclusión. Al parecer, existen al menos tres principios que juegan un papel decisivo en este proceso sucesivo de omisión e integración en formas cada vez más abstractas de codificación. Sin entrar en detalles que no vienen al caso, estos se pueden resumir brevemente como:

1. Ajuste de los sistemas procesadores de información para optimizar (no necesariamente maximizar) la cantidad de información con respecto a las restricciones de alguna función de costo del procesamiento.
2. Invariancia de las estructuras procesadoras bajo la acción de grupos apropiados. Las situaciones en las que se evalúa información están asociadas, a menudo, con una simetría natural, por ejemplo, una medida de la cantidad de información ganada al hacer una observación no debe depender del tipo de unidades de medida del instrumento de medición ni de la ubicación del cero de su escala. La elección arbitraria de unidades de medida y de la posición del cero de la escala pueden imaginarse como "simetrías" del instrumento de medida. Estas simetrías son una manifestación de las estructuras matemáticas conocidas como grupos. Una medida de la información que es independiente de estos cambios de "simetría" se conoce como invariante de grupo (Resnikoff, 1989). Su importancia fue claramente reconocida por Pitts y McCulloch (1947) en la percepción visual y auditiva.
3. Organización jerárquica de las estructuras procesadoras de información. Es ampliamente reconocido que la introducción de una estructura jerárquica aumenta la eficiencia, generalmente. Los ejemplos socioeconómicos son evidentes en la organización de gobiernos, fuerzas militares y grandes firmas comerciales. Un modelo biológico ocurre en los seres humanos, en quienes la relación de la visión periférica con la foveal es un ejemplo que muestra como se alcanzan objetivos comunes por medio de mecanismos jerárquicos similares en un nivel abstracto, aunque actúen en circunstancias físicas diferentes. Otro ejemplo es el de la relación entre la memoria a corto y largo plazo.

1.2.1. OBJETIVOS DEL RECONOCIMIENTO DE PATRONES

Tradicionalmente, el reconocimiento de patrones ha sido visto desde la angosta perspectiva de un modelo clasificacionista, más específicamente relacionado con:

1. La ejecución de extracción de características, es decir, decidir cómo las manifestaciones \underline{x} del objeto X pueden ser descritas simbólicamente en la forma de $f(\underline{x})$;
2. el aprendizaje del mapeo transparente $R_{tr}()$ o, lo que es equivalente, usar un conjunto de patrones de entrenamiento para inferir las reglas de decisión; y,
3. la instrumentación de $R_{tr}(f(\underline{x}))$ con el fin de hacer la clasificación.

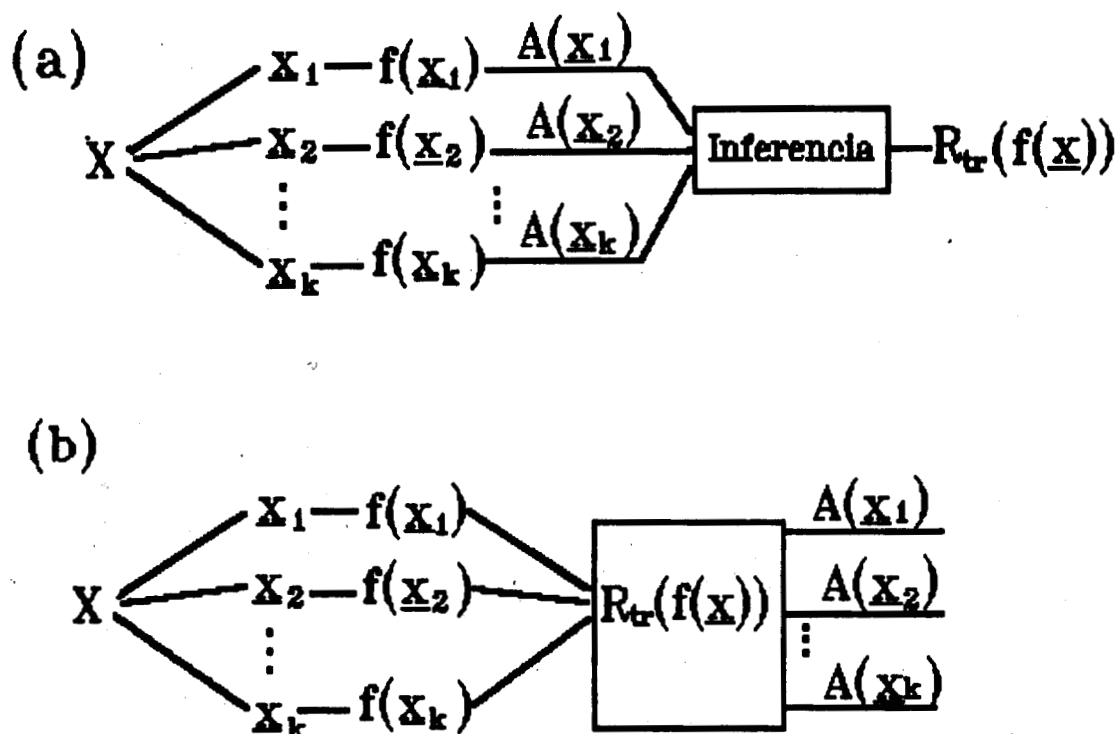


Fig. 1.2. Diagrama de un modelo de reconocimiento de patrones por estimación de atributos. (a) Empleo de un conjunto de entrenamiento con atributos asociados $A(\underline{x})$ y aprendizaje del mapeo transparente para estimar los valores de los atributos. (b) Uso del mapeo aprendido para estimar los valores de los atributos.

Sin embargo, esta perspectiva es muy limitada y restrictiva, pues no abarca los diversos usos de los métodos del reconocimiento de patrones en la resolución de problemas cotidianos ni aprovecha las recientes contribuciones de muchos campos relacionados con el reconocimiento tradicional de patrones.

Una posibilidad alterna se esboza en la figura 1.2, donde el énfasis se ha desplazado a los atributos de los patrones. Las versiones \underline{x} del objeto X siguen siendo $f(\underline{x})$, pero $R_{tr}()$ proyecta cada patrón $f(\underline{x})$ en un $A(\underline{x})$ apropiado. Este mapeo describe al modelo de estimación de reconocimiento de patrones más bien que al modelo tradicional de clasificación. El mapeo es aprendido, como es usual, con una muestra finita de paradigmas, cada uno etiquetado con un valor correspondiente de algún atributo o con varios valores si hay diferentes atributos. La relación así aprendida no sólo es capaz de reproducir todas las asociaciones de patrones-atributos del entrenamiento sino que también será capaz de acomodar un patrón nuevo de tal forma que los valores correctos de sus atributos son calculados por medio del mapeo $R_{tr}(f())$. Si se considera que el único atributo de interés es el etiquetamiento de clases, entonces el modelo de estimación se reduce al de clasificación tradicional.

1.3. REDES NEURONALES ARTIFICIALES

Sin pretender hacer una descripción detallada de las distintas redes neuronales artificiales existentes ni profundizar en sus principios básicos se hará una introducción a lo que éstas son y se describirá de una manera somera algunas de sus propiedades sobresalientes.

Ya arriba se había hecho referencia a varias características de las redes neuronales artificiales sin detallar exactamente lo

que éstas son.² Las redes neuronales están inspiradas por los sistemas biológicos en los que un gran número de células nerviosas, que individualmente pueden ejecutar funciones más bien de un alcance limitado, al actuar de manera colectiva llevan a cabo tareas sorprendentes, tanto que hasta los computadores más avanzados en la actualidad difícilmente las pueden emular, si es que lo logran de alguna manera. Así pues, están constituidas por muchos elementos relativamente sencillos conectados entre sí por elementos de memoria variables, cuyos "pesos" o intensidades se ajustan por experiencia.

Una red neuronal puede definirse como "un sistema computacional distribuido compuesto por un número de elementos procesadores individuales que operan, en un alto grado, en paralelo, interconectados de acuerdo a una topología específica (arquitectura) y con la capacidad de modificar la magnitud de la intensidad de las conexiones entre los diferentes elementos procesadores así como de variables asociadas con éstos (aprendizaje)" (Reilly y Cooper, 1990).

Sus aplicaciones prácticas incluyen áreas como el reconocimiento de patrones (espacial, temporal y espacio-temporal), procesamiento de imágenes, filtrado de señales, compresión de datos, búsqueda asociativa en bases de datos, control adaptivo, optimización, asignación de tareas, detección de fallas, diagnóstico médico, modelaje de fenómenos complejos e interfases adaptivas para sistemas hombre-máquina.

1.3.1. CARACTERISTICAS DE LAS REDES NEURONALES

Existe una gran variedad de redes neuronales que, de una manera general, pueden clasificarse de acuerdo a su arquitectura o conectividad, la dinámica de su funcionamiento y su método de

². De aquí en adelante, al hablar de redes neuronales se asumirá que son artificiales; en el caso de hacer referencia a una red neuronal biológica, se indicará explícitamente.

enseñanza. Aunque sería posible establecer otras subdivisiones y en el caso extremo considerar a cada red neuronal como una categoría única, las características mencionadas antes son útiles al menos para las redes existentes en la actualidad. La proliferación de nuevos modelos y de modificaciones a los ya existentes podría hacer reevaluar estas consideraciones y tener en cuenta otras características, tales como la velocidad de aprendizaje y ejecución, capacidad de memoria y generalización.

Se empezará con la descripción de los elementos básicos, extendiéndose un poco en sus características funcionales, para luego mencionar brevemente las peculiaridades relacionadas con la arquitectura y dinámica y finalizar con algunas anotaciones acerca del aprendizaje y sus algoritmos.

1.3.2. ELEMENTOS PROCESADORES

Los elementos procesadores de una red neuronal artificial se conocen como nodos, neuronas o neurodos. Sus interconexiones con los otros elementos sirven como canales de información. Cada elemento procesador está provisto de una memoria local y el procesamiento que tiene lugar en cada uno de ellos debe depender únicamente de los valores actuales de las señales de entrada y de los de la memoria local (Hecht-Nielsen, 1987). En la figura 1.3 aparece un elemento procesador típico.

La salida del elemento está caracterizada por su función de activación, cuyo argumento está determinado a su vez por el valor de las entradas y pesos que llegan al elemento; además, la salida es una de las variables de estado que describe a la red y puede considerarse ya sea como un potencial (potencial somático del axón) o ya bien como la frecuencia de disparo del nodo específico (Guez, Kam y Eilbert, 1989). Muchos de los primeros modelos de redes neuronales, tales como el perceptrón (Rosenblatt, 1962), empleaban la frecuencia de disparo como una de las variables de estado. Algunos autores, particularmente Grossberg, han abogado

por el uso del potencial como variable de estado, puesto que los potenciales pueden ser positivos o negativos y son capaces de determinar completamente la evolución dinámica del sistema a partir de un instante dado (Grossberg, 1980).

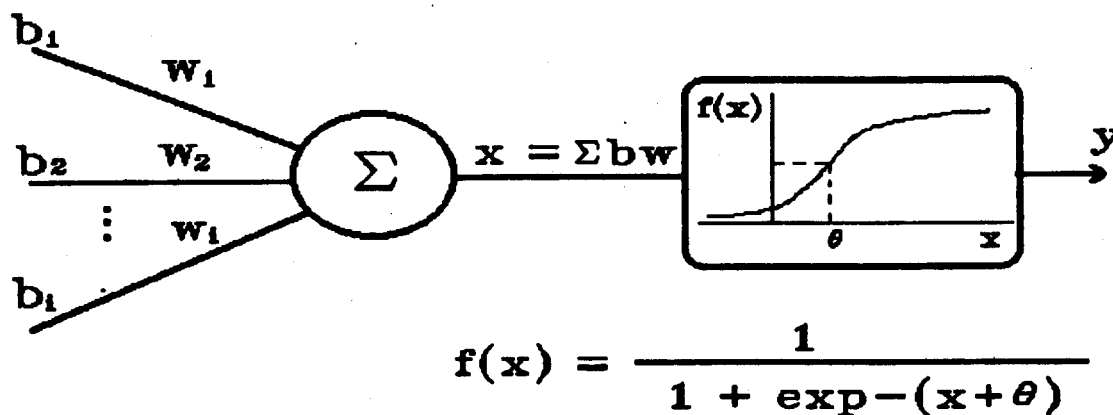


Fig. 1.3. Elemento procesador típico de una red neuronal. La señal de entrada es el vector b , w representa los pesos y $f()$ es la función de activación del elemento, sigmoideal en este caso.

1.3.3. DE LA FUNCION DE ACTIVACION DE LOS NODOS

Aunque el tratamiento matemático de las redes neuronales puede efectuarse sin tener que utilizar la analogía con los sistemas biológicos, resulta útil no olvidar su origen neurofisiológico y poder recurrir a él cuando los resultados no sean convincentes.

A pesar de la existencia de una gran variedad de células y redes neuronales biológicas, si se enfoca solamente la actividad de las células del sistema nervioso en los animales superiores, se puede considerar a una neurona como un generador de trenes de impulsos, cuya frecuencia promedio es una función de la excitación de entrada de la célula.

Las propiedades activas de la neurona tienen su origen en fenómenos biofísicos que ocurren en la membrana celular. El ciclo de disparo obedece, cuando menos en forma aproximada, las ecuaciones de Hodgkin-Huxley. *Grosso modo*, lo que ocurre es lo siguiente. En reposo existe una diferencia de potencial de unos 70 mV entre el interior y el exterior de la membrana celular; el interior de la membrana es negativo con respecto al exterior. Dicha diferencia es causada por la difusión selectiva de iones a través de la membrana, tanto por mecanismos pasivos debidos a la difusión (la concentración de K^+ intracelular es mayor que en el medio extracelular, mientras que ocurre lo contrario para los iones de Ca^{2+} y Na^+) como a mecanismos activos debidos a las bombas metabólicas. Cada terminal sináptico en una neurona, por medio de un complicado mecanismo bioquímico, es capaz de controlar la conductancia iónica de la membrana, de tal manera que las sinapsis excitatorias causan su despolarización y las inhibitorias su hiperpolarización, respectivamente. Cuando la suma de los potenciales locales despolarizadores alcanza cierto umbral, la permeabilidad de la membrana a ciertos iones cambia, se hace eléctricamente activa y se genera, en el segmento inicial del axón, un impulso eléctrico de unos 100 mV de amplitud y una duración de unos 0.5 a 2 ms. Hay un periodo refractorio, luego de cada impulso, de unos pocos milisegundos de duración, durante el cual la neurona no puede ser excitada. Si existe una señal excitatoria de entrada lo suficientemente fuerte, la neurona puede ser forzada a dispararse a una frecuencia de hasta unos cientos de Hz. No obstante, normalmente la actividad es de una velocidad más baja y ocasionalmente se pueden presentar ráfagas o salvas de impulsos.

Sin embargo, el comportamiento de la membrana excitable sólo puede describirse analíticamente con el empleo de muchas simplificaciones. Por otra parte, en el estudio del control sináptico de una neurona por otra existen muchos procesos que son aún oscuros. Parecería entonces que la teoría de sistemas de redes neuronales estuviese basada en componentes cuyas características son en gran parte desconocidas; pero si se

reconoce que el principal objetivo es el estudio del comportamiento global de los sistemas de redes neuronales, es aceptable simplificar las leyes funcionales elementales.

A partir de hechos experimentales conocidos y de algunas restricciones físicas, es factible establecer ciertas propiedades generales para dichas leyes. Por un lado, cada neurona puede dispararse a una frecuencia que está restringida, por diversas razones físico-químicas, entre dos valores, cero y unos cientos de hertz. Por otro lado, la frecuencia de aparición de los potenciales de acción es una función monótona de la entrada total a la neurona, es decir, del promedio de la despolarización de la membrana. No obstante, es preciso tener consideraciones especiales con el efecto combinado de las entradas.

Si los impulsos de entrada en cada sinapsis tienen un efecto sobre la membrana que es independiente de los impulsos en otras sinapsis y de la actividad de la propia neurona, es posible aplicar el principio de "suma espacial" a estos efectos. Si, además, se supone que la neurona actúa como un integrador con fugas de las señales presinápticas, es posible aplicar la siguiente ecuación diferencial para la frecuencia de disparo g_i de una neurona, como un promedio estadístico:

$$dg_i/dt = \sum_{j=1}^n \phi_{ij}(b_{ij}) + \mu_i(g_i), \quad (1.1)$$

donde b_{ij} es la frecuencia de impulsos producida por otra neurona en la entrada j de la neurona i , y ϕ y μ son funciones generales. Como la actividad debe ser una cantidad no negativa, se impone la condición adicional

$$g_i \geq 0. \quad (1.2)$$

La función $\phi_{ij}(b_{ij})$ describe, de manera general, la influencia de la señal b_{ij} sobre la activación de la membrana. La función de "pérdidas" $\mu_i(g_i)$ se considera no lineal con respecto a g_i , puesto que al hacerlo así se facilita la inclusión de efectos de saturación, tal como se mostrará más adelante.

En una descripción más detallada, ϕ_{ij} debe depender de la historia reciente de las señales de entrada, por ejemplo, la fatiga de las conexiones. También, los cambios a largo plazo en ϕ_{ij} pueden ser identificados como efectos de memoria.

Grossberg (1972) ha recalcado que cuando una neurona se dispara se deben de bloquear los efectos de control de las entradas y por lo tanto se debería usar una expresión de la forma $(1 - \tau g_i) b_{ij}$ para la señal de entrada efectiva, siendo τ una constante adecuada. Por otra parte, puesto que el número de impulsos de salida es una función complicada del potencial de membrana y de otros factores, tal vez sea más sencillo incluir todos los términos responsables de los efectos de saturación en un término separado, μ .

Aunque el estudio del comportamiento transitorio de g_i puede ser importante en aplicaciones de control, en este caso nos interesan más los fenómenos que ocurren cuando ha transcurrido un tiempo más o menos largo, de unos 40 a 50 ms, que es el tiempo típico que toma la ecuación (1.1) en estabilizarse. Para obtener la relación entrada-salida en el estado estacionario se hace $dg_i/dt = 0$. Si existe la función inversa de μ , se obtiene:

$$g_i = \sigma_i [\sum_{j=1}^n \phi_{ij}(b_{ij}) - \theta_i] = \sigma_i(I_i - \theta_i), \quad (1.3)$$

donde $\sigma_i()$ es una función sigmoideal. La constante θ_i es un "umbral" hipotético, aunque no es realmente el umbral de disparo de la neurona.

Como se sabe, hay dos tipos de sinapsis, excitatoria e inhibitoria. La eficacia de muchas sinapsis inhibitorias es más fuerte que la de las excitatorias, debido a que el cambio en la conductancia de los canales postsinápticos inducido por el potencial presináptico es de larga duración (Kandel y Schwartz, 1985). El efecto de tales entradas es bastante no lineal: algunas veces pueden bloquear completamente la actividad de la neurona o, en otras ocasiones, inactivar una rama de la neurona. No obstante, las inhibiciones débiles pueden considerarse dentro de sumatorias lineales (Kohonen, 1988).

Se considera, en términos generales, que la función de una neurona es la de dispararse cuando "reconoce" un valor particular de la combinación de las señales de entrada. Desde este punto de vista, puede decirse que la operación de la neurona es la de una función que al relacionar el conjunto de entradas con la frecuencia de disparo, define la "similitud" de la señal de entrada con un conjunto de parámetros internos de la neurona. Las ecuaciones (1.1) y (1.3) expresan el criterio de similitud mediante la sumatoria de las funciones $\phi_{ij}(b_{ij})$. Si se representan las "intensidades sinápticas" por un conjunto de pesos w_{ij} , se obtiene la siguiente aproximación:

$$\sum_{j=1}^n \phi_{ij}(b_{ij}) = \sum_{j=1}^n w_{ij} b_{ij}. \quad (1.4)$$

Por esta razón, una de las medidas de la similitud viene dada por el producto interno de los vectores w y b .

Una de las funciones más empleadas para representar la relación entre la salida y las señales de entrada es la función logística. Con ella la salida de la neurona se representa como

$$g_i = f(x_i) = 1/(1 + \exp(-(x_i + \theta_i)/\theta_0)), \quad (1.5)$$

donde x_i viene dada por una expresión como

$$x_i = \sum_{j=1}^n w_{ij} b_j, \quad (1.6)$$

con lo que se asume que en la ecuación (1.4) todas las neuronas tiene el mismo número de salidas y , por lo tanto, el otro subíndice puede desecharse. En la figura 1.4 se pueden apreciar las características de la función (1.5). La variable θ_i se conoce como umbral y su efecto es el de desplazar la función de activación a lo largo del eje horizontal. El efecto de θ_0 es el de variar la forma de la sigmoide; un valor pequeño hace que la función se comporte como una unidad lógica de umbral. El uso de una función sigmoide, en lugar de una de dos estados, reduce el número de estados espurios de sistema (Waugh et al., 1990) y

disminuye la cantidad de elementos procesadores necesarios para ejecutar una tarea determinada (Sontag, 1989).

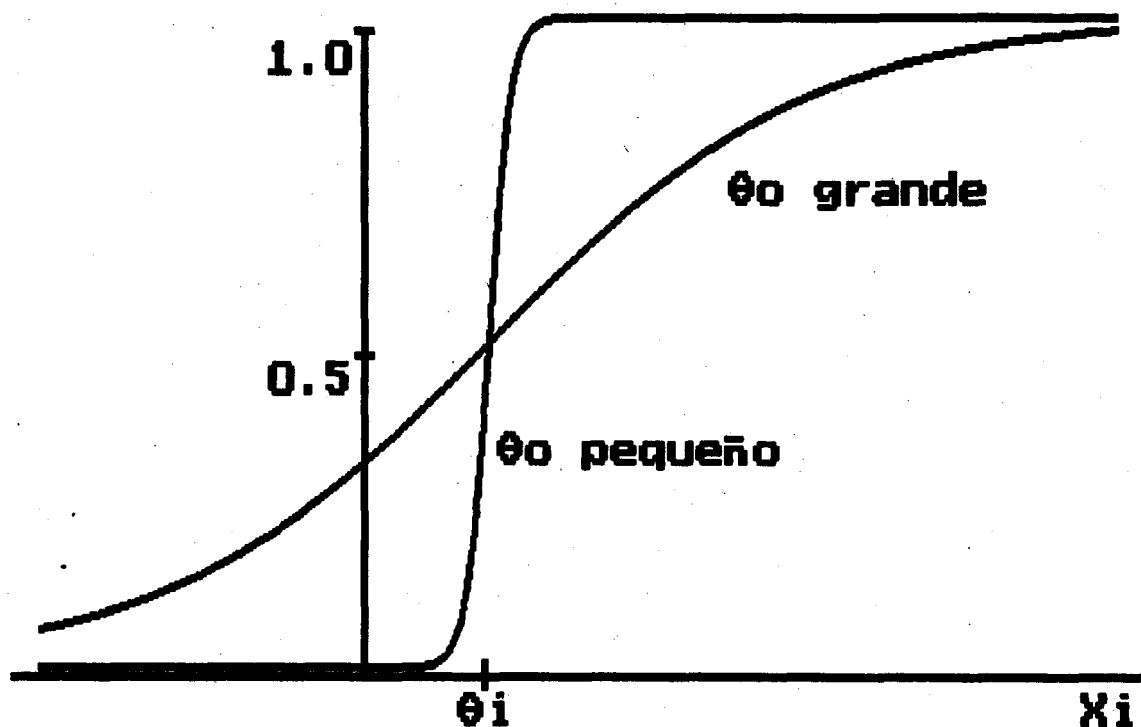


Fig. 1.4. Función de activación sigmoideal.

1.3.4. ARQUITECTURA Y DINAMICA

En lo que respecta a la arquitectura, las configuraciones van desde redes donde todos los elementos están interconectados entre sí hasta aquellas donde las conexiones se establecen aleatoriamente. Las conexiones entre los elementos pueden ser uni o bidireccionales y el flujo de información (de la región de entrada a la de salida) puede igualmente ser en un solo sentido y tener, o no, vías de retroalimentación.

Desde el punto de vista de la dinámica de la red, hay dos clases de comportamiento. En la primera, la evolución de la red está explícitamente determinada por la dinámica temporal de las ecuaciones que describen a la red; este es el comportamiento

propriadamente dinámico. Un ejemplo de este tipo son las redes de Hopfield (Hopfield, 1982; Geszti, 1990). En el segundo, o estático, el tiempo no juega un papel determinante en la evolución de la red, tal como sucede, por ejemplo, en el perceptrón.

1.3.5. APRENDIZAJE

Aunque no son los únicos sistemas que exhiben capacidad de aprendizaje, es esta capacidad, junto con su habilidad para el procesamiento en paralelo, lo que distingue a las redes neuronales.

Antes de seguir adelante, vale la pena aclarar que se entiende aquí por aprendizaje. Se puede definir el aprendizaje como un proceso mediante el cual la respuesta conductual converge a una alternativa de un conjunto de elecciones posibles, cuando un sujeto es expuesto repetidamente a una situación. Un procedimiento que permita tomar paradigmas de un concepto y encontrar una expresión que faculte describir si un objeto dado es o no un ejemplo del concepto subyacente, será por lo tanto un componente importante del aprendizaje. Tales procedimientos de inferencia pueden ser usados para clasificar ejemplos no vistos antes, predecir eventos futuros y almacenar datos en forma comprimida o en un formato determinado. Algunos modelos matemáticos del aprendizaje han sido propuestos recientemente, destacándose entre ellos el de Valiant (1984) que asocia al aprendizaje con un problema de generalización en el sentido de una tarea de predicción, pero su análisis está más allá de los alcances del presente trabajo.

La manera de instrumentar dichos procedimientos es una de las características esenciales de cualquier red neuronal y se conoce como método de enseñanza o regla de aprendizaje. Tal método permite que cada elemento procesador modifique algunos valores de su memoria interna como réplica a las señales de

entrada a la red y a otras de la propia red, con lo que la respuesta del elemento procesador cambiará con el tiempo.

Hay por lo menos seis clases de reglas diferentes de aprendizaje, que se mencionan a continuación: de *Hebb* (Hebb, 1949), que ha sido interpretada matemáticamente de diversas maneras (Hopfield, 1982; Grossberg, 1974), incluyendo el mecanismo complementario, conocido como anti-hebbiano (Lisman, 1989); del *perceptrón* (Rosenblatt, 1962); de *Widrow-Hoff* o mínimos cuadrados (Widrow y Hoff, 1960); de *Grossberg* (Grossberg, 1987); de *cuantización vectorial* (Kohonen, 1988) y de *Kosko-Klopf* (Kosko, 1988). Una red neuronal puede emplear más de una de estas reglas durante el aprendizaje.

La forma en que se ejecuta el entrenamiento de una red puede seguir alguno de los siguientes esquemas:

1. Entrenamiento supervisado, durante el cual se le proporciona a la red tanto la señal de entrada como la respuesta que se desea obtener, por parte de un instructor externo.
2. No supervisado o de auto-organización, donde la red responde espontáneamente a la señal de entrada y adquiere por sí misma la configuración más adecuada.
3. Entrenamiento calificado, en el que se le proporciona a la red la señal de entrada, pero no la respuesta deseada sino una señal de "castigo" o "recompensa" que le indica que tan "bien" lo está haciendo (Barto y Anandan, 1985).

1.4. REDES DE CAPAS MÚLTIPLES CON PROPAGACION PROGRESIVA Y RETROPROPAGACION DEL ERROR

El tipo de redes en que estamos interesados puede derivarse de una configuración bastante general de una red neuronal, compuesta por k clases de neuronas, cada clase con n_i elementos ($i=1, \dots, k$). Una red de este tipo puede describirse por medio de un sistema de ecuaciones diferenciales de la forma (Amari, 1977; Dimopoulos, 1989):

$$TX' + X = Wg(X) + b, \quad (1.7)$$

donde

$$\begin{aligned} X &= [X_1 \ X_2 \ X_3 \ \dots \ X_k]^T, \\ X &= [x_1 \ x_2 \ \dots \ x_{n_1}, \ x_{n_1+1} \ \dots \ x_{n_1+n_2}, \ \dots, \ x_{N-n(k-1)+1} \ \dots \ x_N]^T \end{aligned}$$

es el vector de estado del sistema, con x_i el potencial medio de soma de la i -ésima neurona, τ denota la traspuesta,

$$N = \sum_i^k n_i$$

el número total de neuronas en la red,

$$T = \text{diag}[t_i], \quad i = 1, 2, \dots, N; \quad t_i > 0$$

representa la constante de tiempo de la velocidad de cambio del potencial (de aquí en adelante se considerará que t_i es igual a uno),

$$W = \begin{bmatrix} W[11] & W[12] & \dots & W[1k] \\ W[k1] & W[k2] & \dots & W[kk] \end{bmatrix}$$

es la matriz de conectividad de la red, donde la submatriz $W[ij]$ representa la eficiencia sináptica de las neuronas j -ésimas a las i -ésimas.

$g_i(x_i)$ representa el promedio a corto plazo de la velocidad de disparo de la i -ésima neurona como función de su potencial de soma. g es una función continua, monotónicamente no decreciente, que satisface la condición de Lipschitz (con el fin de garantizar una solución única al sistema de ecuaciones diferenciales) y para la cual existe un valor real a tal que $g(a) = 0$. Finalmente, b es un vector de entrada externo a la red.

De particular importancia son la matriz de conectividad W y la función g . Recordando lo dicho antes, g describe el comportamiento de la frecuencia del tren de impulsos en el axón como una función del potencial de soma. Se considera que los potenciales de acción aparecen cuando el voltaje de soma sobrepasa determinado umbral, y que su frecuencia aumenta al

aumentar el voltaje. La forma preferida para g es la de una sigmoide. Por otra parte, la matriz W refleja la topología de la red. Cada uno de sus elementos puede ser positivo, negativo o cero, dependiendo de si excita, inhibe o no ejerce efecto directo sobre otras neuronas, respectivamente.

Las soluciones al sistema (1.7) pueden exhibir oscilaciones, convergencia a puntos fijos aislados o caos. En general, la convergencia a puntos aislados es el comportamiento deseado, ya que los valores de dichos puntos serán asociados con "memorias" del sistema.

Desde el punto de vista funcional, una de las configuraciones morfológicas más favorables para una red neuronal, en lo que se refiere a establecer una relación estructurada con el medio externo, es la de poseer tanto una región que reciba información de ese medio así como otra que le indique a éste las respuestas de la red a tal información. Esto es particularmente importante si se desea que la red aprenda a relacionar ciertos estímulos con determinadas respuestas.

Cualquier algoritmo que se proponga establecer relaciones entre estímulos y respuestas debe considerar que tanto el estímulo particular como su respuesta correspondiente han de ser "recordados" por el sistema cuando el instructor realimenta al sistema con la instrucción determinada. Esta observación sugiere la oportunidad de proporcionar un estímulo externo de tal manera que sea un vector de entrada independiente del tiempo en lugar de ser una condición inicial del sistema; de esta forma, se puede presentar la respuesta deseada simultáneamente con la señal de entrada. La entrada será una perturbación externa permanente y, desde el punto de vista dinámico, se tendrá un sistema forzado.

Si se considera que la red tiene dos regiones, una de entrada y otra de salida, y que la región de entrada tiene como única función la de recibir la señal de estímulo, cual si fuese una especie de retina, no es necesario que existan conexiones entre los elementos ésta sino sólo con los de la región de salida. Lo mismo puede pensarse para los elementos de la región de salida. Además, se pueden excluir las conexiones que vayan

hacia "atrás", de la salida a la entrada, pues su existencia involucra una realimentación sin supervisión, difícil de evaluar. En una red con esta configuración, la matriz W es triangular inferior y la estabilidad del sistema se encuentra garantizada (Pineda, 1987, 1989).

Mientras que la evolución dinámica es fundamental en el comportamiento de muchas redes, por ejemplo en las de Hopfield, es casi trivial para las redes con una capa de entrada y otra de salida, interconectadas de la manera descrita arriba. Para éstas, la dinámica se reduce al cómputo, a intervalos discretos de tiempo, de la respuesta de los elementos de la capa de salida. No obstante, puede decirse que existe una cierta especie de complementariedad entre los sistemas que se auto-organizan siguiendo su dinámica interna y aquéllos donde el aprendizaje está controlado, ya que en los últimos la dinámica del ajuste de pesos produce el efecto de una auto-organización.

Con el propósito de comprender más fácilmente el desarrollo del algoritmo de retropropagación, se empezará por estudiar el aprendizaje en una red con dos capas, una de entrada y otra de salida. El sistema recibe diversas señales de entrada y, junto con ellas, la respuesta correcta que se espera de él si todas las intensidades sinápticas (pesos) estuviesen ajustadas al valor adecuado. El objetivo del proceso de aprendizaje es el de modificar de tal manera los pesos que las respuestas correctas sean obtenidas para todas las señales de entrada. Para tal efecto, se comenzará por el estudio del concepto de memoria distribuida.

1.4.1. MEMORIA DISTRIBUIDA

Algunas de las primeras redes neuronales, de comienzos de los sesenta, empleaban modelos lineales para la función de activación de sus elementos. Así, para una red de arquitectura arbitraria, la salida de una neurona puede expresarse como

$$g_i^\alpha = \sum_{j=1}^N w_{ij} b_j^\alpha, \quad (1.8)$$

donde $b_1^\alpha \dots b_N^\alpha$ son los componentes de la α -ésima señal de entrada y W es la matriz $N \times N$ de intensidades sinápticas. Dicha matriz puede verse como un mapeo del espacio de actividades de la señal B , $b = [b_1, \dots, b_N]^T$, al espacio de actividades neuronales G , $g = [g_1, \dots, g_N]^T$. De manera más compacta, (1.8) puede expresarse como

$$g = Wb. \quad (1.9)$$

Se propuso que es en mapeos modificables del tipo de W donde se almacena la experiencia y memoria del sistema (Hebb, 1949). Como ya se había señalado antes, la memoria animal es, muy probablemente, distribuida y direccionable por contenido o asociación, a diferencia de la memoria de las máquinas, que está, por lo menos hasta hoy, almacenada en localidades específicas. Si se tienen en cuenta las señales que el sistema ha experimentado, la matriz W puede representarse por

$$W = \sum_{\mu\pi} c_{\mu\pi} g^\mu \times b^\pi, \quad (1.10)$$

donde g^μ y b^π son los patrones de actividad neuronal de salida y entrada, respectivamente, y los $c_{\mu\pi}$ son coeficientes que reflejan el grado de conectividad entre las entradas y salidas. El símbolo \times indica el producto externo, llamado algunas veces diádico, entre los vectores de salida y entrada (gb^T). El elemento ij de W representa la intensidad sináptica de la conexión que va de la neurona j , en el espacio B , a la neurona i en el espacio G (fig. 1.5). Si b_j es diferente de cero, la frecuencia de disparo a la salida de la i -ésima neurona será $g_i = \sum_j w_{ij} b_j$. De (1.10) se tiene que la intensidad sináptica de la conexión ij está formada por la suma de la experiencia completa del sistema, como un reflejo de las frecuencias de disparo de las neuronas conectadas con este par. Cada experiencia o asociación $\mu\pi$ está almacenada en el arreglo total de $N \times N$ conexiones y no en una neurona particular.

En otras palabras, en una memoria distribuida cada evento se encuentra almacenado en una porción extensa del sistema, mientras que en cada elemento local particular se pueden sobreponer muchos eventos (Reilly y Cooper, 1990).

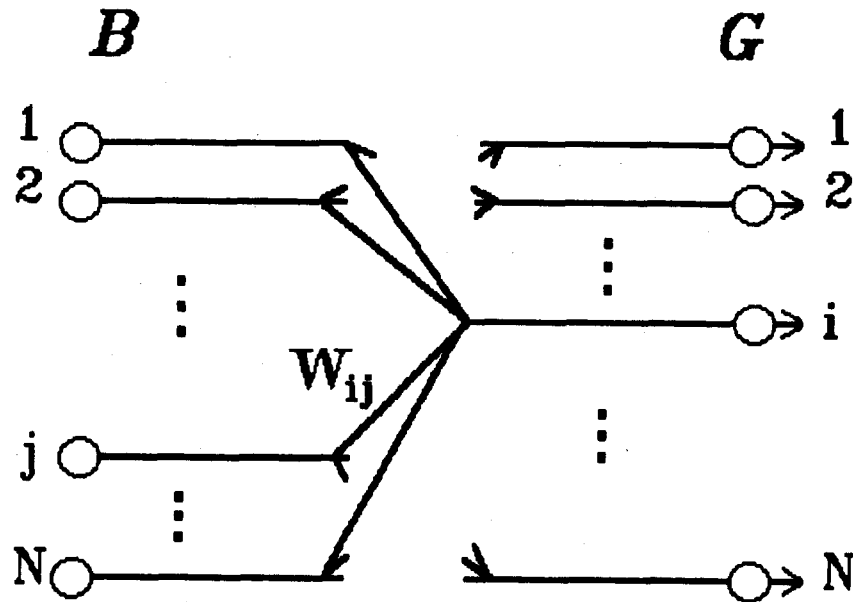


Fig. 1.5. Memoria distribuida en una red neuronal típica. Sólo se muestran las conexiones que llegan al nodo i de la capa G .

La matriz W puede recordar perfectamente si los b^π son ortogonales. Si los patrones de entrada no lo son, existen varias técnicas para ortogonalizarlos. Esto es lo que se conoce como mapeo óptimo (Kohonen, 1988), que se delinea a continuación.

Para todo el conjunto de entradas, b^π , y salidas deseadas, g^μ , se quiere que

$$g^\pi = W^* b^\pi, \quad \pi = 1, 2, \dots, K \quad (1.11)$$

En general, se desea

$$G = W^* B, \quad (1.12)$$

donde G y B son las matrices

$$G = \begin{bmatrix} g[11] & g[12] & \dots & g[1K] \\ g[N1] & g[N2] & \dots & g[NK] \end{bmatrix} \quad (1.13)$$

$$B = \begin{bmatrix} b[11] & b[12] & \dots & b[1K] \\ b[N1] & b[N2] & \dots & b[NK] \end{bmatrix}$$

Si $K = N$ y existen las inversas, se tiene que

$$W^* = GB^{-1}. \quad (1.14)$$

Si las matrices no son cuadradas y no tienen inversa, W^* puede obtenerse por el método de inversas generalizadas de Moore-Penrose o pseudoinversas.

1.4.2. APRENDIZAJE SUPERVISADO EN SISTEMAS LINEALES

El ejemplo clásico de aprendizaje supervisado es el del elemento lineal adaptivo (ADALINE), propuesto por Widrow y Hoff en 1960. El algoritmo de modificación de los pesos tiene la forma

$$W(t+1) = \tau W(t) + \delta W(t), \quad (1.15)$$

con

$$\delta W(t) = \beta (d^\alpha - g^\alpha) \times b^\alpha, \quad (1.16)$$

donde $g^\alpha(t) = W(t)b^\alpha(t)$ es la salida obtenida para la entrada $b^\alpha(t)$ y d^α es la respuesta deseada para el patrón α . De la expresión (1.16) viene el nombre de regla delta para este tipo de algoritmos. De aquí que

$$\delta W(t) = \beta (d^\alpha - W(t)b^\alpha) \times b^\alpha. \quad (1.17)$$

Si $W(t)b^\alpha = d^\alpha$, la ecuación (1.17) se hace cero, así que los valores correctos de W serán el punto fijo W^* (en $\mathbb{R}^{N \times N}$). Este valor de W corresponde a la pseudoinversa que se mencionaba arriba.

Si se define una función de error, costo o "energía",

$$E = \frac{1}{2} \sum_{\alpha} (d^\alpha - Wb^\alpha)^2, \quad (1.18)$$

la variación de E con respecto a W será

$$\delta E = -(\delta W) \sum_{\alpha} (d^\alpha - Wb^\alpha) \times b^\alpha. \quad (1.19)$$

Si

$$\delta W = \sum_{\alpha} (d^\alpha - g^\alpha) \times b^\alpha, \quad (1.20)$$

como en (1.16), δE será menor que cero y el método será obviamente uno de gradiente descendente.

Con este método de enseñanza es posible aprender hasta N vectores en un espacio N -dimensional, pero con restricciones importantes. La separación es posible sólo si el número de patrones de entrada, K , es menor o igual que la dimensión del espacio, N . Además, la separación conseguida es lineal, por lo que si una separación no lineal es buscada, deben introducirse no linealidades en la red.

El algoritmo empleado aquí es usado frecuentemente en los filtros adaptivos y, de hecho, su origen puede encontrarse en la teoría de tales filtros (Widrow et al., 1976). Si una señal $b(k)$, siendo k el tiempo, se aplica a la entrada de un filtro FIR, se obtendrá una señal de salida $g(k)$ de acuerdo a la convolución

$$g(k) = \sum_{j=0}^{N-1} b(k-j)w_j, \quad (1.21)$$

donde $\{w_j\}$ es la respuesta impulsiva del filtro. Si se supone que la respuesta es causal, es decir, de duración finita, w_j será cero para j menor que cero y j mayor que $N-1$. Si b y w se expresan como los vectores

$$B(k) = [b(k) \ b(k-1) \ \dots \ b(k-N+1)]^T, \quad (1.22)$$

$$W = [w_1 \ w_2 \ \dots \ w_{N-1}], \quad (1.23)$$

(1.21) puede escribirse en forma más compacta como

$$g(k) = W B(k), \quad (1.24)$$

que no es sino otra forma de la ecuación (1.9).

Si se desea encontrar el mejor vector W para la respuesta impulsiva, w_{SS} , de tal manera que una función de la diferencia entre la salida deseada, $d(k)$, y $g(k)$, la salida del filtro, sea lo más pequeña posible, se define la función

$$J_{SS} = \sum_k \|d(k) - g(k)\|^p, \quad (1.25)$$

conocida como función de rendimiento, y se minimiza. p es una constante no negativa en el intervalo $[1, \infty]$; usualmente se le asigna un valor de dos (Cadzow, 1990). J_{SS} es una función de $d(k)$ y de $B(k)$ y W , ya que estas dos últimas determinan a $g(k)$. Para una secuencia determinada de vectores $\{B(k)\}$ y escalares $\{d(k)\}$, J_{SS} es función de W únicamente, y por lo tanto será una medida de cuán bien actúa W como respuesta impulsiva para producir una salida $g(k)$ similar a la secuencia de referencia, $d(k)$.

Debe tenerse en cuenta, sin embargo, que en el caso de los filtros adaptivos el procesamiento está centralizado y las señales de entrada y salida se presentan, generalmente, en forma secuencial, mientras que en los sistemas con aprendizaje se busca que el procesamiento sea paralelo y distribuido entre los diferentes elementos y las señales de entrada se presentan en paralelo.

1.4.3. PERCEPTRONES

Uno de los primeros elementos procesadores reminisciente a las neuronas biológicas que introdujo no linealidades fue el perceptrón (Rosenblatt, 1962). Una de sus variantes más conocidas aparece en la figura 1.6.

La capa S está compuesta por sensores que reciben las señales externas. Las salidas de los sensores son combinadas en la segunda capa, A, por elementos asociativos. Si los patrones de entrada son binarios, los elementos A operan como simples compuertas lógicas. Los elementos de respuesta forman la tercera capa, R, y constituyen el núcleo del sistema. Cada elemento R es una unidad lógica de umbral y su salida tiene dos estados, activa o inactiva. Si se asume que la salida de los elementos A es b_i , que las intensidades de las conexiones de A a R son w_i y g la salida del elemento R, se tiene que

$$\begin{aligned} g = 0 & \quad \text{si } \sum_i w_i b_i < \theta, \\ g = 1 & \quad \text{si } \sum_i w_i b_i \geq \theta, \end{aligned} \tag{1.26}$$

donde θ es el umbral discriminador del elemento R considerado.

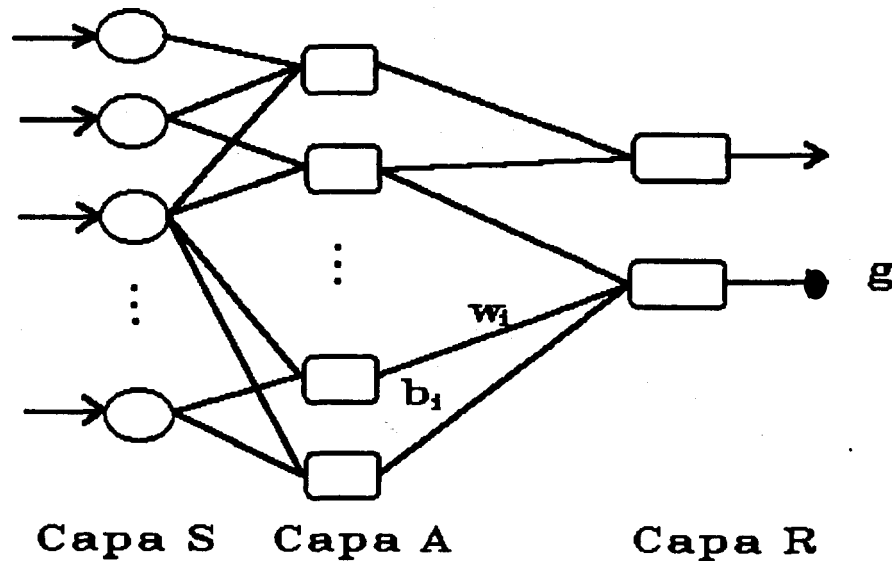


Fig. 1.6. Perceptrón.

El aprendizaje se basa en la diferencia entre la salida obtenida y la deseada. La regla de aprendizaje se deriva de la ecuación (1.15) con r igual a 1:

$$w_i(t+1) = w_i(t) + \delta w_i(t), \quad (1.27)$$

con $\delta w_i = \beta(d - g)b_i$, siendo d la respuesta esperada, g la respuesta actual del perceptrón y β una constante ($0 < \beta \leq 1$).

Se ha demostrado, por ejemplo en Nilsson (1965), que las relaciones de entrada-salida pueden encontrarse por este procedimiento, siempre y cuando sus valores existan. Tal resultado se conoce como el teorema de convergencia del perceptrón. Sin embargo, las regiones de decisión que puede encontrar un perceptrón son también lineales, aunque esta restricción ya no es válida si se emplean capas intermedias de perceptrones entre las de entrada y salida. No obstante que esto era conocido (Nilsson, 1965; Duda y Hart, 1973), las redes con varias capas de perceptrones enfrentaban el problema de asignación de créditos, es decir, la manera de ajustar los pesos para obtener los resultados deseados. La supuesta irresolubilidad de este problema fue uno de los principales argumentos esgrimidos contra la utilidad de los sistemas con aprendizaje por parte de Minsky y Papert, (1969).

1.4.4. REGLA DELTA GENERALIZADA

La incorporación de una estructura intermedia entre la entrada y la salida, a menudo conformada por una o más capas, permite que la red codifique internamente las características relevantes de las diferentes relaciones de entrada-salida. En la figura 1.7 se muestra una red típica con una capa intermedia o escondida, como se le llama algunas veces, puesto que no está comunicada directamente con el medio externo.

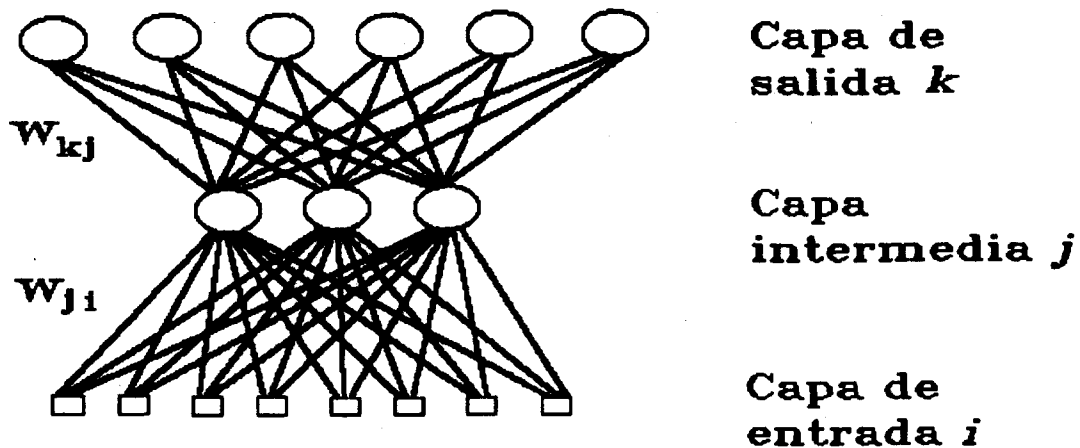


Fig. 1.7. Diagrama esquemático de una red de capas múltiples.

El problema de la asignación de créditos ha sido resuelto de varias maneras y uno de los métodos más usados es el conocido como de retropropagación, cuyas primeras formulaciones pueden remontarse a los finales de la década de los sesenta en el contexto de la teoría del control óptimo (Bryson y Ho, 1969). Fue redescubierto independientemente por Paul Werbos (1974), pero su rápida popularización surge a partir de la publicación del trabajo de Rumelhart, Hinton y Williams, (1986).

El método de entrenamiento usado es supervisado y básicamente consiste en la repetición de los siguientes pasos:

1. Se recibe una señal b_α del exterior, junto con la información de la respuesta esperada, $d_\alpha = [d_{1\alpha}, \dots, d_{k\alpha}]$.
2. Se propaga la señal de entrada hacia la capa de salida, comenzando por la capa intermedia más cercana.
3. Se compara de alguna forma la respuesta de la red con la respuesta deseada.
4. A partir del valor obtenido en 3. se calcula un cambio para todos los pesos, de tal manera que la respuesta de la red se aproxime a la esperada.
5. Se ajustan los pesos de acuerdo al paso 4 y se regresa al punto 1.

Al final, la red debe encontrar el conjunto de pesos que satisfaga todas las asociaciones (cfr. ecuación (1.12)).

En la figura 1.7 la señal de entrada llega a los nodos de la capa i y la salida de tales nodos se toma como igual a la señal de entrada ($g_i = x_i$). La señal de entrada a cada uno de los nodos de la capa intermedia j es

$$x_j = \sum_i w_{ji} g_i. \quad (1.28)$$

La salida de los nodos j será

$$g_j = f(x_j), \quad (1.29)$$

donde f es la función de activación de los nodos, como se había mencionado antes. En la capa k , de salida, similarmente se tiene que

$$x_k = \sum_j w_{kj} g_j, \quad (1.30)$$

$$g_k = f(x_k) = f(\sum_j w_{kj} g_j). \quad (1.31)$$

Una de las maneras de resolver el problema de la asignación de valores a los pesos es construir una función de error a partir de (1.18). Para cada patrón el error será

$$E_\alpha = \frac{1}{2} \sum_k (d_k^\alpha - g_k^\alpha)^2, \quad (1.32)$$

y el error promedio del sistema será

$$E = (2P)^{-1} \sum_\alpha \sum_k (d_k^\alpha - g_k^\alpha)^2, \quad (1.33)$$

con P el número total de patrones. La función de error puede ser diferente, pero la minimización del error cuadrático determina un clasificador bayesiano óptimo (Wan, 1990).

Se logra la convergencia hacia los valores deseados de los pesos si se hace que los cambios incrementales delta w_{kj} sean proporcionales a $\partial E / \partial w_{kj}$, donde E corresponde a la ecuación (1.32), con el sub-índice α omitido. Esto significa que

$$\delta w_{kj} = -\beta \partial E / \partial w_{kj}. \quad (1.34)$$

Sin embargo, el error E está expresado en términos de g_k . Usando la regla de derivación de la cadena, E puede evaluarse como

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial x_k} \frac{\partial x_k}{\partial w_{kj}}. \quad (1.35)$$

Usando (1.30), se concluye que

$$\frac{\partial x_k}{\partial w_{kj}} = \frac{\partial \sum w_{kj} g_j}{\partial w_{kj}} = g_j. \quad (1.36)$$

Si se define $\delta_k = -\partial E / \partial x_k$, se llega a la expresión

$$\delta w_{kj} = \beta \delta_k g_j, \quad (1.37)$$

que es formalmente equivalente a la ecuación (1.16).

Para calcular δ_k se puede usar de nuevo la regla de la cadena y escribir la derivada parcial en función de dos factores, uno que expresa la variación del error con respecto a la salida g_k y el otro que indica la variación de la salida referida con respecto a la entrada al nodo. Así, se tiene que

$$\delta_k = -(\partial E / \partial x_k) = - \frac{\partial E}{\partial g_k} \frac{\partial g_k}{\partial x_k}. \quad (1.38)$$

Ahora, ambos factores pueden ser evaluados como

$$\partial E / \partial g_k = -(d_k - g_k) \quad (1.39)$$

y

$$\partial g_k / \partial x_k = f'(x_k). \quad (1.40)$$

De aquí que

$$\delta_k = (d_k - g_k) f'(x_k) \quad (1.41)$$

para cualquier elemento procesador k de la capa de salida. El cambio en los pesos que conectan con la capa de salida será

$$\text{delta } w_{kj} = \beta (d_k - g_k) f'(x_k) g_j = \beta \delta_k g_j. \quad (1.42)$$

Para los pesos que no van directamente a la capa de salida la situación es diferente. No obstante, puede escribirse

$$\begin{aligned} \text{delta } w_{ji} &= -\beta \frac{\partial E}{\partial w_{ji}} \\ &= -\beta \left(\frac{\partial E}{\partial x_j} \right) \left(\frac{\partial x_j}{\partial w_{ji}} \right) \\ &= -\beta \left(\frac{\partial E}{\partial x_j} \right) g_i \\ &= \beta \left[-\left(\frac{\partial E}{\partial g_j} \right) \left(\frac{\partial g_j}{\partial x_j} \right) g_i \right] \\ &= \beta \left(-\frac{\partial E}{\partial g_j} \right) f'(x_j) g_i \\ &= \beta \delta_j g_i \end{aligned} \quad (1.43)$$

Sin embargo, el factor $(\partial E / \partial g_j)$ no puede evaluarse directamente. Por lo tanto, debe ser expresado en términos conocidos y que puedan calcularse. Específicamente,

$$\begin{aligned} \frac{-\partial E}{\partial g_j} &= \frac{-\partial E}{\partial x_k} \frac{\partial x_k}{\partial g_j} = \sum_k \frac{-\partial E}{\partial x_k} \frac{\partial (\sum_m w_{km} g_m)}{\partial g_j}, \quad (1.44) \\ &= \sum_k \left(\frac{\partial E}{\partial x_k} \right) w_{kj}, \\ &= \sum_k \delta_k w_{kj} \end{aligned}$$

Por lo que

$$\delta_j = f'(x_j) \sum_k \delta_k w_{kj}. \quad (1.45)$$

Esto no quiere decir otra cosa sino que las deltas en los nodos internos pueden calcularse en términos de aquellos de la capa de salida. Empezando por la capa de salida se evalúan los δ_k con la expresión (1.41) y luego los errores se retropropagan hacia las capas internas.

Resumiendo, las ecuaciones usadas durante la fase de entrenamiento del algoritmo de retropropagación son:

$$\text{delta}_\alpha w_{ji} = \beta \delta_j^\alpha g_i^\alpha, \quad (1.46)$$

donde α indica el número del patrón. Si los nodos j son de la capa de salida

$$\delta_j^\alpha = (d_j^\alpha - g_j^\alpha) f'(x_j^\alpha), \quad (1.47)$$

pero si los elementos j son de capas intermedias se tendrá que

$$\delta_j^\alpha = f'(x_j^\alpha) \sum_k \delta_k^\alpha w_{kj}. \quad (1.48)$$

Las expresiones obtenidas para actualizar el valor de los pesos muestran que el procedimiento no es sino un método básico de gradientes, más exactamente, el del gradiente descendente (ascendente) de Cauchy (Pierre, 1986) y, como todos estos métodos, presenta una serie de limitaciones, que se mencionarán posteriormente.

Es conveniente recalcar que las ecuaciones deducidas se utilizan solamente durante la fase de entrenamiento, cuando los pesos se modifican para reducir el error cuadrático de la capa de salida. Una vez que el aprendizaje ha finalizado, el valor de los pesos permanece constante y la única dinámica que se da es la de obtener la respuesta de la red ante un nuevo estímulo, propagando hacia la capa de salida las activaciones originadas por la presentación de un patrón en la capa de entrada. Aunque la actualización de las respuestas de los nodos de una misma capa se hace sincrónicamente, las diferentes capas lo hacen de manera progresiva, de la salida a la entrada, por lo que la señal presente en la capa de entrada debe mantenerse hasta que se obtenga la respuesta en la de salida, por lo menos.

→

CAPITULO 2

METODOS

El registro extracelular con un electrodo de baja impedancia hace posible la obtención de señales que se originan en varias de las neuronas que se encuentran alrededor de éste. La señal resultante está compuesta por varias familias de impulsos nerviosos, cada una de las cuales, se supone, es originada por una neurona particular. Si se asume que las neuronas forman un grupo interactivo, se puede tratar de descifrar su estructura funcional estudiando la correlación de sus activaciones.

Las señales de las neuronas individuales pueden caracterizarse por su morfología, que es una función del tipo de neurona, de las particularidades del electrodo usado y de la técnica de implantación, así como del tejido adyacente al sitio de registro. Las señales de neuronas diferentes pueden ser muy semejantes entre sí y la presencia de ruido hace la labor de clasificación aún más difícil. Sin embargo, si se desea un análisis correcto de los patrones de disparo de las neuronas, una tarea de clasificación confiable es indispensable.

El análisis de la señal registrada requiere dos etapas diferentes. En la primera, se deben detectar los impulsos presentes en la traza y luego, en la segunda, clasificarlos en la familia correspondiente. Puesto que es esta última tarea la que nos concierne más directamente, en las secciones siguientes se hará una revisión rápida de algunas de las técnicas existentes para la discriminación de las espigas nerviosas y una descripción del método usado en el presente trabajo. Al final se delinea el procedimiento empleado para detectar los impulsos nerviosos en una traza de registro.

2.1. METODOS TRADICIONALES DE CLASIFICACION

Se ha desarrollado una gran variedad de técnicas con el fin de clasificar impulsos nerviosos, con diversos grados de dificultad en su instrumentación. Algunas técnicas permiten un análisis en tiempo real, mientras que otras requieren ejecutarse fuera de línea. De igual forma, algunas están basadas en el análisis de la señal por medio de software mientras que otras requieren de circuitería especial para hacerlo.

Usualmente, los métodos consisten en analizar parte de la señal y extraer de ella ciertos parámetros que servirán para caracterizar a las diferentes clases de impulsos que aparecen en el registro; estos valores son empleados luego para clasificar los impulsos en el resto del registro. Los análisis más simples usan algunas variables tales como la amplitud de uno de los picos, o de dos si la onda es bifásica, el tiempo entre dichas amplitudes, el tiempo entre la máxima amplitud y el cruce por cero de la forma de onda, etc. Tales técnicas son eficientes cuando la relación señal a ruido es alta y cuando no hay traslapes temporales entre los diferentes impulsos. Los métodos más complejos involucran técnicas de filtros digitales y reconocimiento estadístico de patrones. Como es de esperarse, las técnicas más desarrolladas ofrecen mejores resultados, pero requieren más trabajo para su ejecución.

La amplitud de un impulso nervioso es la característica más extensamente usada para su clasificación. Otros parámetros, tales como la duración del impulso y el tiempo entre el pico positivo y negativo no parecen ser muy útiles en la mayoría de los casos (Camp y Pinsky, 1979).

En uno de los métodos, se construye una plantilla con la forma de onda correspondiente a cada neurona y la clasificación se efectúa al comparar la forma de onda estudiada con la de las diferentes plantillas, asignándole la clase de aquella con la que el error cuadrático sea mínimo. Con el propósito de hacer más rápida la computación, en lugar de la distancia euclidiana se puede calcular otro tipo de distancia, por ejemplo la de los

valores absolutos de las señales , aunque la confiabilidad de la clasificación disminuye a medida que aumenta la dimensión de las señales (D'Hollander y Orban, 1979).

Otra técnica para substituir a un conjunto de formas de onda mediante unas cuantas variables se basa en la construcción de una base de funciones ortogonales o de componentes principales, a partir de formas de onda ortonormales que proporcionan el error cuadrático mínimo para representar a las señales originales. Así, cada impulso es descrito por una suma ponderada de los dos o tres componentes principales de la forma de onda. Un tratamiento detallado de los fundamentos teóricos de esta metodología puede hallarse en Glaser y Ruchkin (1976).

Si se registra la actividad eléctrica en dos lugares diferentes a lo largo de un tronco nervioso, es posible usar la velocidad de conducción como otra de las variables para caracterizar a los impulsos nerviosos (Camp y Pinsker, 1979).

El uso de regresión lineal para calcular los coeficientes de alguna función a partir de un número limitado de muestras de cada impulso ha sido empleado por O'Connell et al. (1973), junto con otras características como el área total y el valor rms del estímulo estudiado.

Una descripción de los problemas encontrados en la obtención y el análisis de los datos de un registro multieléctrico puede encontrarse en Abeles y Goldstein (1977) y Gerstein et al. (1983). Schmidt (1984, 1984a) ha hecho una revisión bastante detallada de diferentes técnicas, tanto para la detección como para la clasificación. Diversas metodologías para abordar el problema de la clasificación pueden consultarse, entre otras, en Dining y Sanderson (1981), Schvertzman et al. (1982), McGillivray y Wald (1984), Vibert et al. (1987) y Yang y Shamma (1988). Una comparación de los resultados conseguidos con diferentes técnicas apareció en Wheeler y Heetderks (1981), quienes cotejan la eficacia de nueve métodos distintos.

2.2. METODO EMPLEADO

Para la tarea de clasificación que se deseaba ejecutar con una red neuronal, se contaba inicialmente con un conjunto de varias plantillas de impulsos nerviosos extracelulares típicos de la corteza auditiva del gato, uno para cada clase distinta. Cada plantilla está representada por 128 valores digitales, que se han obtenido al promediar la señal digitalizada de varios impulsos de la misma clase. Los registros fueron hechos con microelectrodos de metal y son característicos del campo eléctrico cercano al soma de las neuronas. Algunos de ellos aparecen en la figura 2.1.

De antemano se había elegido trabajar con una red de capas múltiples, entrenada con retropropagación del error, por ser uno de los diseños que parecen abrigar más perspectivas alentadoras en los sistemas con aprendizaje, como lo indica la creciente aparición de publicaciones acerca de sus aplicaciones. Además, el algoritmo de aprendizaje no está ceñido a una arquitectura particular y es relativamente fácil de instrumentar. Enseguida se detallan algunas consideraciones necesarias para su uso.

2.2.1. DE LOS DATOS DE ENTRADA

Las señales de entrada pueden tomar valores discretos, por ejemplo -1, 0 ó 1, o continuos, en cuyo caso conviene normalizar los valores dentro de algún intervalo determinado, por ejemplo, en el intervalo [-1, 1].

Tal como se había señalado, es posible caracterizar a los impulsos nerviosos mediante alguna variable relacionada con su morfología o aplicar algún tipo de análisis a las señales originales, por ejemplo, la transformada de Fourier, y presentar dichos valores como señal de entrada a la red. Sin embargo, teniendo en cuenta que una persona sin experiencia previa puede aprender rápidamente a clasificar correctamente los impulsos en una traza de registros con base en su morfología, se decidió por el empleo directo de los datos sin ningún procesamiento previo, a

excepción de una normalización. Esto no significa que se considere que la red sea capaz de ejecutar una labor de una complejidad similar a la llevada a cabo por el cerebro humano, sino, únicamente, que se espera que encuentre una representación interna de las características que describen a cada clase de impulsos a partir de la información más simple, sin que se le indique de antemano algún rasgo distintivo de cada impulso.

Como se contaba con un solo ejemplar de cada clase, con el fin de tener más ejemplos de cada clase se construyeron otros impulsos agregando ruido gaussiano de media cero y diferentes desviaciones estándar a las muestras originales (Chu, 1989). La amplitud de las señales resultantes se normalizó posteriormente en el intervalo $[-1, 1]$, de acuerdo a los valores mínimo y máximo de las amplitudes encontradas. Se hicieron algunas pruebas normalizando la señal con respecto a su valor medio y desviación estándar, lo cual se indicará cuando sea el caso.

Baum y Hausler (1989) han señalado la existencia de argumentos estadísticos que sugieren que el número de patrones de entrenamiento necesarios para determinar los pesos de la red es proporcional al número los últimos, limitación bastante seria cuando no se cuenta con datos suficientes para el entrenamiento, lo que es una situación frecuente en la práctica.

2.2.2. DE LAS CARACTERISTICAS DE LA RED

La fase de aprendizaje de una red neuronal con capas múltiples y retropropagación del error consiste en inicializar sus pesos con valores aleatorios pequeños y presentarle uno a uno los patrones de entrada que, junto con su clasificación correcta, conforman el conjunto de entrenamiento. Usando el patrón de entrada, se calcula el error en la capa de salida y con el algoritmo de retropropagación se computa el cambio necesario en los pesos para disminuir el error. Ahora bien, las ecuaciones deducidas en el capítulo anterior pueden tomar muchas iteraciones para lograr su objetivo, por lo que es útil introducir algunos cambios en ellas.

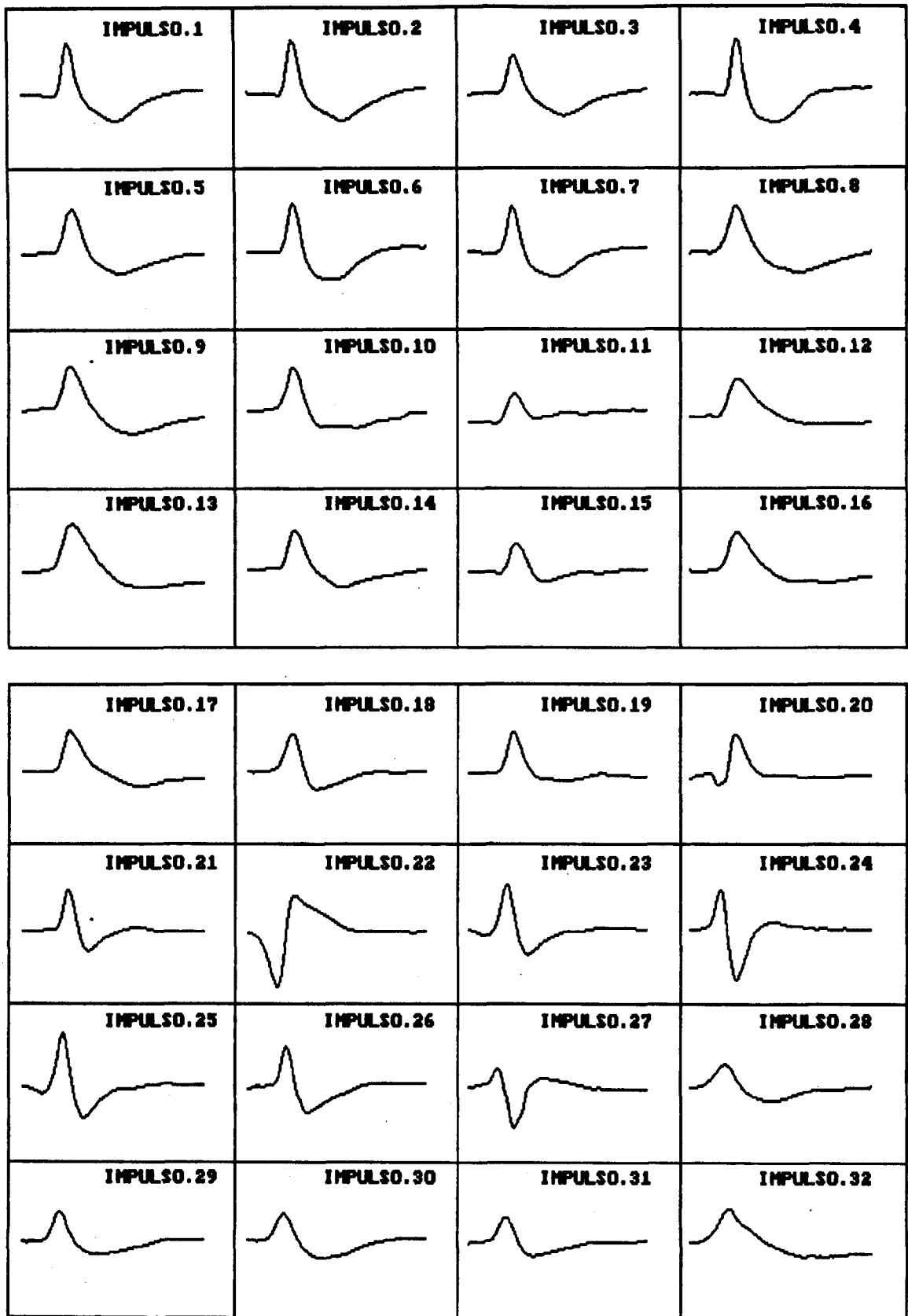


Fig. 2.1. Plantillas de diferentes impulsos nerviosos típicos.

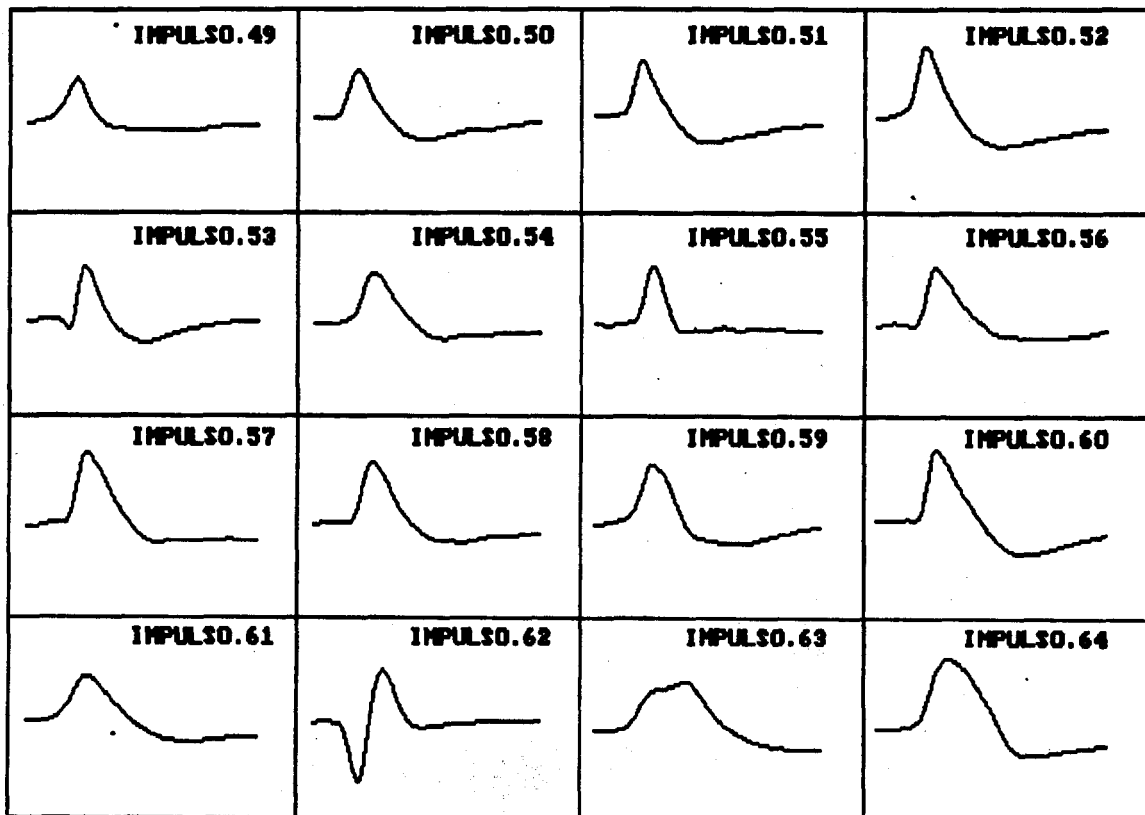
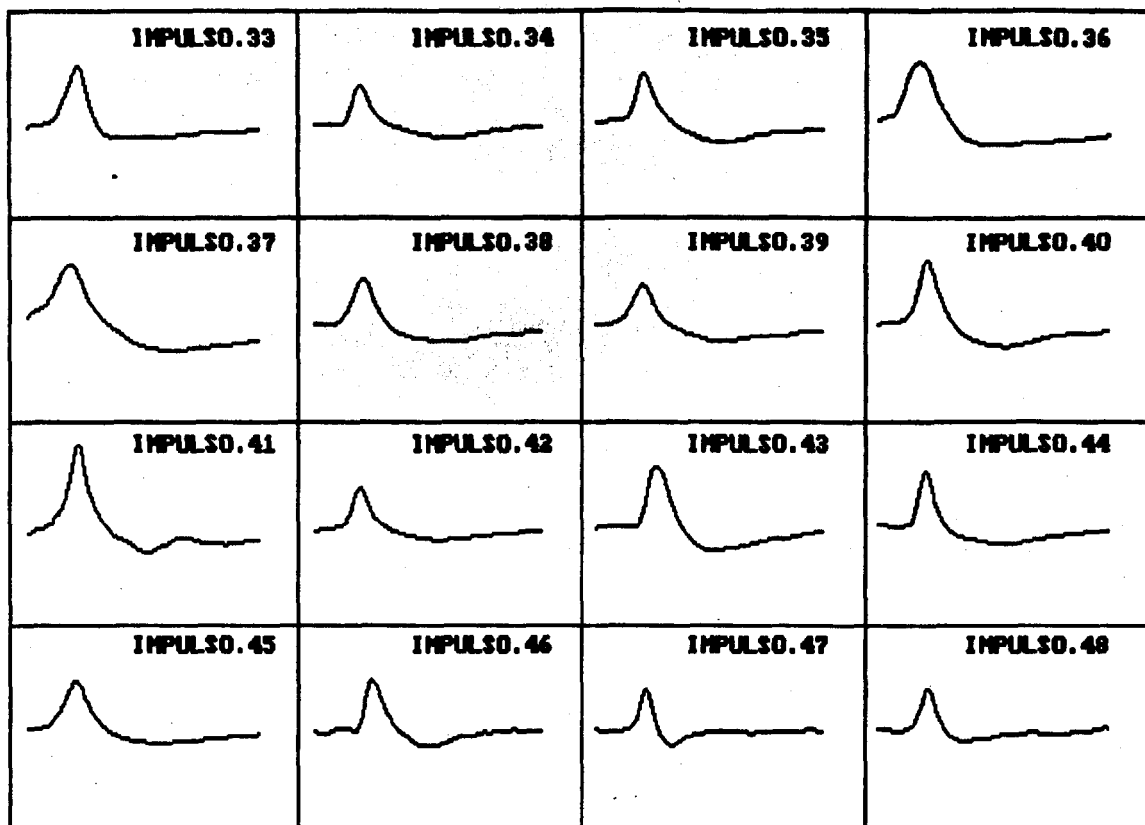


Fig. 2.1 (cont.). Plantillas de diferentes impulsos nerviosos.

Uno de los principales defectos de los métodos de gradiente descendente es la posibilidad de quedar atrapados en un mínimo local de la función que se quiere minimizar, sin que exista la certeza de si la respuesta encontrada es debida a uno de ellos, al mínimo global o a un punto de inflexión. Una opción para evitar que esto ocurra consiste en aumentar el valor de la constante de la velocidad de aprendizaje, β , pero esto puede hacer que el error oscile entre valores extremos sin llegar a encontrar un mínimo. De las diferentes modificaciones introducidas para tratar de disminuir las posibilidades de hallar una solución espuria, una de las más simples consiste en introducir un término adicional proporcional al cambio encontrado en la iteración anterior al calcular los nuevos cambios en los pesos (Rumelhart, Hinton y Williams, 1986). Este término puede considerarse como un momentum en el espacio de los pesos, que proporciona una especie de inercia en el sentido de la dirección del cambio anterior del peso. Más específicamente, la ecuación (1.46), se convierte en

$$\Delta w_{ji}(t) = \beta \delta_{\alpha j} g_{\alpha j} + \mu \Delta w_{ji}(t-1), \quad (2.1)$$

donde t indica el número de la presentación, β es la velocidad de aprendizaje y μ es una constante que determina el efecto de los cambios pasados y usualmente se toma en el intervalo $[0, 1]$. La inclusión del término adicional contribuye también a disminuir las oscilaciones en el valor del error total del sistema; en general, hace el aprendizaje más rápido, aunque en algunas aplicaciones específicas puede tener el efecto contrario, por lo que es necesario experimentar con diferentes valores para cada problema particular. La ecuación (2.1) fue la empleada para obtener los resultados reportados.

De las numerosas modificaciones propuestas para incrementar la velocidad de convergencia del algoritmo básico se mencionan algunas de las más relevantes: Jacobs (1988), Fahlman (1989) y Felten et al. (1990).

En la deducción de las ecuaciones para actualizar los pesos no se mencionó qué ocurría con los umbrales. Sin embargo, estos pueden tratarse como cualquier otro peso con el simple expediente de agregar un elemento procesador, cuya salida sea siempre uno, a cada una de las capas de entrada e intermedias. Los pesos de las conexiones de tales unidades a las de la capa siguiente serán los umbrales para las últimas.

En la práctica hay dos métodos para actualizar los pesos. De acuerdo al razonamiento seguido para obtener las expresiones (1.47) y (1.48), la forma correcta es calcular los cambios de los pesos para cada uno de los patrones de entrenamiento, sumar estos al finalizar y actualizar los pesos. Este modo se conoce como actualización por épocas o periódica, donde una época es la presentación a la red de cada uno de los patrones del conjunto de entrenamiento. Es improbable que este procedimiento sea biológicamente plausible (Pao, 1989). La otra manera, conocida como continua, consiste en actualizar los pesos después de la presentación de cada uno de los patrones. Usualmente, con este último método se obtienen mejores resultados y en menor cantidad de presentaciones del conjunto de entrenamiento.

Cuando se utiliza la actualización de pesos de manera continua, es posible experimentar con el orden en el que se muestran los patrones de entrenamiento. La forma más simple consiste en presentarlos en un orden fijo preestablecido. Otra, en una secuencia aleatoria dentro de cada época, denominada actualización estocástica en la literatura. Finalmente, es posible ordenarlos de tal modo que, para un conjunto determinado de pesos, se presenten primero aquellos patrones que producen un error mayor terminando con los de menor error, o viceversa, lo que está más acorde con las conclusiones obtenidas en algunos estudios biológicos del aprendizaje (Hampson y Volper, 1989).

En lo que concierne al diseño de la red misma, conviene tener en cuenta las siguientes consideraciones. El número de nodos de entrada dependerá de la cantidad de variables que se desee emplear para describir las características de los patrones por clasificar. El número de nodos en la capa de salida depende

de cómo se quiere codificar la señal de salida. Si se emplea una representación localizada, se utiliza un nodo para cada una de las clases, de tal manera que la clase elegida por la red se determina por el nodo de la capa de salida que se encuentre activo, es decir, con su señal de salida cercana a uno. Una representación distribuida es más conveniente cuando se quiere clasificar muchas categorías, con el propósito de no aumentar de manera excesiva el número de pesos de la red, por ejemplo, la indicación que se obtuvo la clase 25 puede conseguirse al codificar la activación de cinco elementos en lugar de los 25 que se necesitarían con una representación local; no obstante, esta codificación impone mayores exigencias a la red, por lo que puede requerirse el uso de más capas intermedias (Maren et al., 1990).

La decisión de elegir el número de capas intermedias necesarias y su cantidad de elementos es más difícil, pues no existe ninguna regla específica que permita su cálculo. Lippmann (1987) ha sugerido que en el peor de los casos se necesitarían dos capas intermedias, que el número de nodos en la capa intermedia más cercana a la de salida sería como máximo igual al número de clases por separar y que la relación óptima del número de elementos de la primera (la más cercana a la de entrada) a la segunda capa intermedia es de 3:1. Sin embargo, sus indicaciones son obtenidas con base en un análisis geométrico intuitivo de las regiones de separación creadas por las hipersuperficies que generan los elementos discriminadores de cada capa intermedia y no son una demostración rigurosa. Otros estudios indican que el número de nodos depende únicamente del número de patrones de entrada (Mirchandani y Cao, 1989; Mitchinson y Durbin, 1989). Al parecer, es posible usar sólo una capa intermedia para mapear un conjunto de entrada finito, de dimensión m , a uno de salida de dimensión d (Huang y Huang, 1991). Cuando la dimensión de la entrada es mucho mayor que la de salida, una buena aproximación práctica para el número óptimo de nodos en la capa intermedia es la media geométrica de ambas dimensiones (Maren, Jones y Franklin, 1990).

Conviene anotar que, puesto que el algoritmo de aprendizaje es determinista, la solución encontrada, es decir, el valor final de los pesos, dependerá de las condiciones iniciales. Por lo tanto, es aconsejable efectuar varias sesiones de entrenamiento, partiendo de diferentes valores iniciales de los pesos y comparar los resultados de cada una de ellas.

Por último, el criterio para decidir cuando terminar el entrenamiento de la red se puede escoger de diferentes maneras. Lo más sencillo consiste en alcanzar un valor mínimo, establecido previamente, del error total normalizado del sistema (ecuación 1.33). En este caso, es conveniente limitar el número máximo de épocas durante las cuales se aplicará el algoritmo de aprendizaje, con el fin de concluir el programa en el caso que el error normalizado de la red no alcance el valor deseado. Otra variante consiste en evaluar periódicamente la tarea de clasificación con un conjunto de prueba. Si el porcentaje de clasificaciones incorrectas con éste empieza a incrementarse, la etapa de aprendizaje se da por finalizada (Hampson y Volper, 1989).

2.3. DETECCION DE IMPULSOS EN UNA TRAZA NERVIOSA

Aunque el propósito principal del presente trabajo no está dirigido a la detección de los impulsos, para que el clasificador pueda ser empleado de una forma sencilla al estudiar una traza de un registro continuo, es preciso que primero se determine la ocurrencia de los impulsos.

Existen varias alternativas para seleccionar el esquema de detección, diferenciándose entre sí, fundamentalmente, de acuerdo al tipo de ruido esperado en la señal por analizar. Puesto que el ruido más frecuentemente encontrado en un registro de la actividad eléctrica neuronal es de características gaussianas (Abeles, 1982), un algoritmo basado en la amplitud de la señal será adecuado. La idea básica es que la amplitud de la señal

relacionada con la activación neuronal es lo suficientemente grande como para poder ser distinguida del ruido presente en el registro. Con el fin de hacer más notoria esta diferencia, se puede emplear la primera o segunda derivada de la señal. El algoritmo utilizado es una adaptación de un sistema análogo para la detección de complejos QRS desarrollado por Fraden y Neuman (1980). Debe mencionarse que en los potenciales de acción estudiados la pendiente no es muy pronunciada, puesto que la baja impedancia de los electrodos usados (0.3 - 1.0 MOhmios a 1 kHz.) hace que la respuesta obtenida sea un promedio del campo eléctrico a lo largo de su punta no aislada, de unos 10 a 20 micrómetros (Abeles, 1982).

Si se toma la señal de entrada como un arreglo unidimensional de puntos $\{X(n)\} = \{x(0), x(1), \dots, x(N-1)\}$ y una constante de umbral relacionada con la amplitud, por ejemplo, $A = 0.2 \max [X(n)]$, los datos originales son rectificadas y recortados de acuerdo con la relación

$$\begin{aligned} y_0(i) &= x(i) & \text{si } x(i) \geq A, & & 0 \leq i \leq N-1, \\ y_0(i) &= A & \text{si } x(i) < A. & & \end{aligned} \quad (2.2)$$

La primera derivada del arreglo Y_0 es calculada en cada punto como

$$y_1(i) = y_0(i+1) - y_0(i-1), \quad 1 \leq i \leq N-2. \quad (2.3)$$

Se considera que existe un impulso cuando determinado número de puntos consecutivos del arreglo Y_1 excede el valor de otro umbral constante, que se establece previamente de acuerdo con las características de la señal.

Con el fin de mejorar la relación señal a ruido del arreglo $\{X(n)\}$, se usó la transformada de Haar (Ahmed y Rao, 1975), pues las características impulsivas de sus funciones base la hacen muy adecuada para tratar señales semejantes a los potenciales de acción.

El conjunto de las funciones de Haar es periódico, ortonormal y completo para cualquier N positivo que sea una potencia entera de dos y está definido por las relaciones

$$\begin{aligned} \text{har}(0, 0, t) &= 1, & t \in [0, 1) \\ \text{har}(r, m, t) &= 2^{r/2}, & (m-1)/2^r \leq t < (m-1/2)/2^r \\ &= -2^{r/2}, & (m-1/2)/2^r \leq t < m/2^r \\ &= 0, & \text{para otros casos, } t \in [0, 1), \end{aligned} \quad (2.4)$$

con $0 \leq r < \log_2 N$ y $1 \leq m \leq 2^r$. La notación $\text{har}(r, m, t)$ representa un conjunto de funciones continuas de Haar de grado r , orden m y tiempo t (Ohkita et al., 1983).

Si $\{\text{har}(r, m, t)\}$ es muestreado a una frecuencia N , se obtiene una matriz ortogonal $N \times N$, la matriz de Haar $H^*(n)$ con $n = \log_2 N$, donde cada renglón corresponde a las muestras de cada una de las $\text{har}(r, m, t)$. Por ejemplo, la matriz de Haar (8x8) es

$$H^*(3) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix} \begin{array}{l} \} N/N \\ \} N/N \\ \} N/4 \\ \} N/4 \\ \} N/2 \\ \} N/2 \\ \} N/2 \\ \} N/2 \end{array}$$

Los coeficientes de la transformada de Haar $Z_X(k)$, con $k = 0, 1, \dots, N-1$, correspondientes a la secuencia de datos $\{X(m)\} = \{x(0) \dots x(N-1)\}$ se obtienen al calcular

$$Z_X(n) = H^*(n) X(m) / N. \quad (2.5)$$

donde $X(m) = [x(0), \dots, x(N-1)]^T$.

Al observar $H^*(3)$ se aprecia que $N/2$ coeficientes en el dominio de Haar miden la correlación de coordenadas adyacentes en el espacio de datos, tomadas de dos a la vez; $N/4$ miden coordenadas tomadas de cuatro a la vez, etc., hasta llegar a 2 coeficientes que tienen en cuenta a todos los datos de la señal

evaluada. Esto significa que la transformada de Haar proporciona un dominio que es sensitivo tanto local como globalmente, lo que no ocurre en el caso de las transformadas discretas de Fourier y de Walsh-Hadamard, donde cada coeficiente de la transformada es una función de todas las coordenadas del espacio original, es decir, es global. La transformada de Haar puede calcularse en $2(N-1)$ sumas y N multiplicaciones con el algoritmo desarrollado por Andrews y Capari (1970).

El ruido presente en la señal puede ser removido fácilmente en el dominio de la transformada por medio de un umbral, de tal forma que aquellos coeficientes que sean menores que el umbral se hagan cero y los demás permanezcan sin cambio. La secuencia filtrada es pasada de nuevo al dominio del tiempo y sobre ella se efectúa la detección de los impulsos.

La estimación de la variancia del ruido permite la elección del valor adecuado para el umbral del filtro del ruido. Si el ruido presente en la traza es una distribución independiente e idénticamente equivalente a un proceso gaussiano con media cero y variancia σ^2 en el dominio del tiempo, no es muy difícil encontrar que sus componentes en el dominio de la transformada (2.5) tendrán una media de cero y una variancia igual a σ^2/N . Por lo tanto, si se toma el umbral de ruido unas 3 ó 4 veces mayor que la desviación estándar, se tendrá una buena probabilidad de no obtener detecciones debidas al ruido ya que $P_{\text{Prob}}[\|Y_k\| \leq 3\sigma/\sqrt{N}] = 0.9973$ y $P_{\text{Prob}}[\|Y_k\| \leq 4\sigma/\sqrt{N}] = 0.99985$. En la práctica, el umbral de ruido se calculó encontrando la variancia de la señal en el dominio del tiempo en un intervalo que no tuviese impulsos, y multiplicando dicho valor por una función no lineal decreciente $U(\sigma)$ con rango $[3, 4]$, de manera que si $\sigma \geq \sqrt{N}$, $U(\sigma)$ tiende a 3 y. si $\sigma < \sqrt{N}$, $U(\sigma)$ tiende a 4.

En la figura 2.2 puede verse como al filtrar el ruido en el dominio de la transformada (fig. 2.2c), la señal reconstruida en el dominio del tiempo (fig. 2.2d) aparece con menos ruido ni componentes de alta frecuencia, lo cual hace más pronunciadas las pendientes de las espigas y facilita su identificación mediante los algoritmos que utilizan la pendiente de la señal.

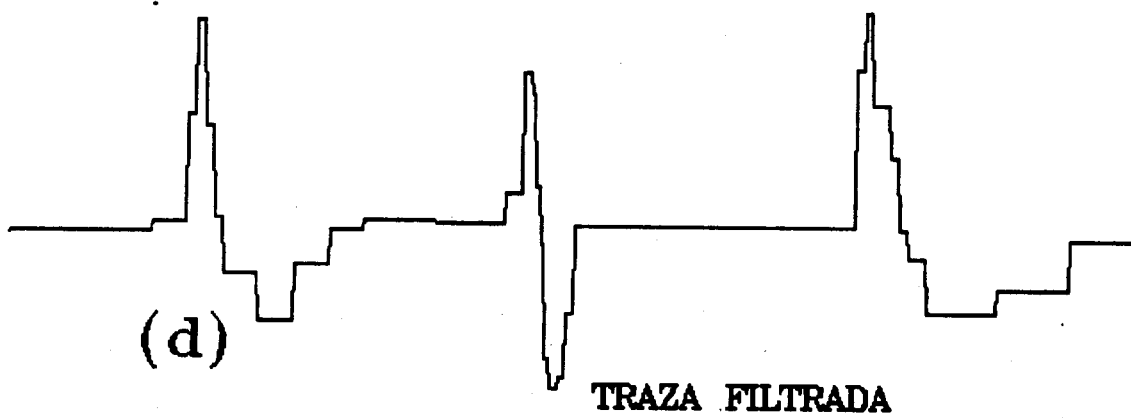
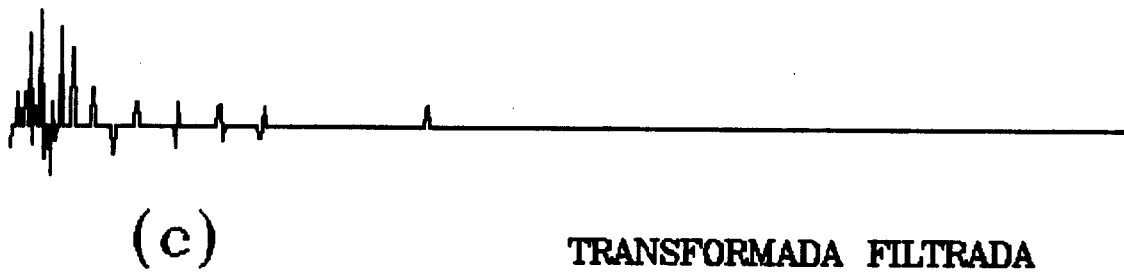
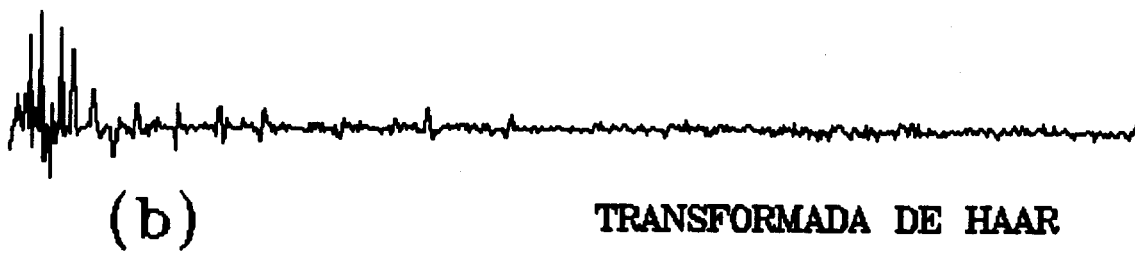


Fig. 2.2 Filtrado del ruido mediante el empleo de la transformada de Haar.

CAPITULO 3

RESULTADOS

3.1. RED UTILIZADA

La red se simuló con un programa escrito en C. Puesto que para el tipo de señales empleadas el tiempo de entrenamiento puede ser intolerablemente largo en una computadora personal, la mayoría de las sesiones de entrenamiento se realizaron en una computadora Stardent Titan con dos procesadores, cuya velocidad máxima de operación es de 16 MFLOPS por procesador. El programa ejecutable se optimizó con las opciones de vectorización y paralelización del compilador disponible.

En un comienzo se pretendió enseñar a una red con una arquitectura convencional, como la mostrada en la figura 1.7, a que aprendiese a identificar las 64 clases de impulsos nerviosos diferentes que aparecen en la figura 2.1. Se utilizaron redes con dos capas intermedias, 128 elementos en la capa de entrada y siete en la de salida, o sea una representación distribuida en la salida. Sin embargo, el tiempo de entrenamiento resultó extremadamente largo, tomando un promedio de una cinco a seis mil presentaciones del conjunto de entrenamiento, aunque en algunas ocasiones se requiriesen más de sesenta mil, como en el caso ilustrado en la figura 3.1, el cual tomo más de cinco días en ejecutarse.

Aunque el desempeño de la red al identificar señales contaminadas con ruido de hasta un 50%¹ es aceptable, si las

¹ El cálculo del porcentaje de ruido presente en un impulso se hizo mediante el empleo de la magnitud del valor rms de la señal de error, esto es, la diferencia entre la forma de onda de la señal estudiada y la de la misma sin ruido. Tal magnitud se divide por el valor rms de la señal sin ruido. El porcentaje de ruido obtenido es el inverso de la relación instantánea señal a ruido de la muestra estudiada (Wheeler y Heetderks, 1981).

señales se desplazan ligeramente en el tiempo, es decir, su ubicación espacial en la capa de entrada es cambiada, el porcentaje de clasificaciones correctas disminuye drásticamente, aún para las señales que se usaron durante el aprendizaje. En la tabla 3.1 se resumen los resultados obtenidos con tres configuraciones diferentes. Debido al largo tiempo de entrenamiento, no se hicieron varias corridas para obtener soluciones diferentes a partir de condiciones iniciales distintas.

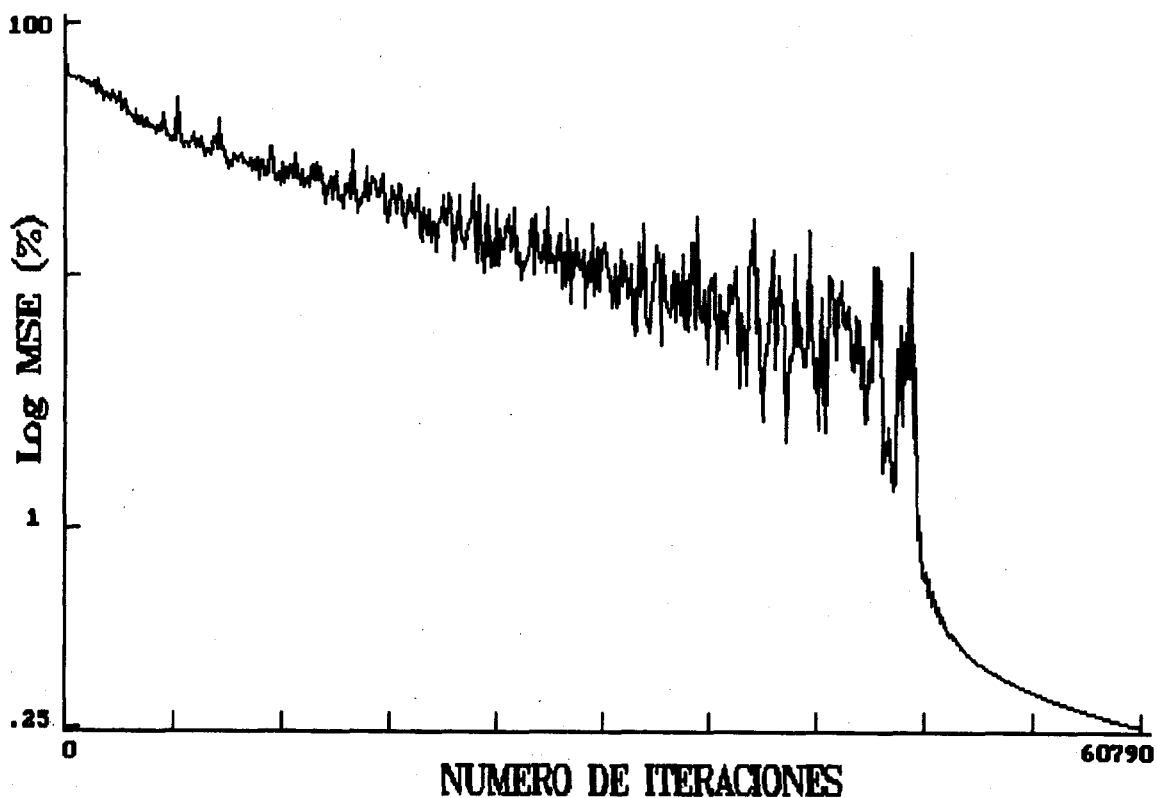


Fig. 3.1. Variación del error total medio en una red con 128 nodos en la capa de entrada, dos capas intermedias con 15 y 30 nodos y 7 en la de salida. Se usaron 64 impulsos diferentes, cada uno representativo de una clase. La actualización de pesos se hizo por épocas. Constantes de aprendizaje: $\beta = 0.07$ y $\mu = 0.08$.

Tabla 3.1

Desempeño de varias redes con arquitectura convencional para clasificar 64 clases diferentes de impulsos nerviosos. Para señales desplazadas se utilizaron como referencia las mismas del entrenamiento.

Nodos 1ra capa int.	Nodos 2da. capa int.	% error señales 50% ruido	% error señales desplazadas 1 punto
30	15	83	45
40	20	86	52
60	30	88	58

Los datos son una clara indicación de que la red es muy poco tolerante a traslaciones de la señal en la capa de entrada, lo que constituye una limitación muy seria en un sistema de clasificación de impulsos, pues exigiría que la detección se efectuase siempre en la misma posición del impulso, lo que es prácticamente imposible de lograr cuando existe ruido en la señal. Una solución a este problema sería la de entrenar a la red con ejemplos de las señales con diferentes desplazamientos, lo cual es obviamente impráctico, pues aumentaría considerablemente el tamaño del conjunto de entrenamiento y, consecuentemente, la duración de la fase de aprendizaje.

Otra solución posible a esta dificultad surge al considerar una red recurrente, es decir, con propagación de información de la capa de salida hacia la de entrada, puesto que es bien sabido que tal configuración permite manejar de manera mucho más eficiente señales que cambian con el tiempo (Almeida, 1989; Pearlmutter, 1989). Tal red debe permitir seguir usando el algoritmo de retropropagación del error. Puesto que el cómputo de n periodos hecho por cualquier red recurrente puede emularse con una red virtual de propagación progresiva (Minsky y Papert, 1969), mediante la duplicación de su arquitectura para cada uno de los pasos temporales considerado, se siguió la sugerencia formulada por Rumelhart, Hinton y Williams (1986, págs. 354-357) para convertir una red recurrente a una virtual equivalente con propagación en un solo sentido.

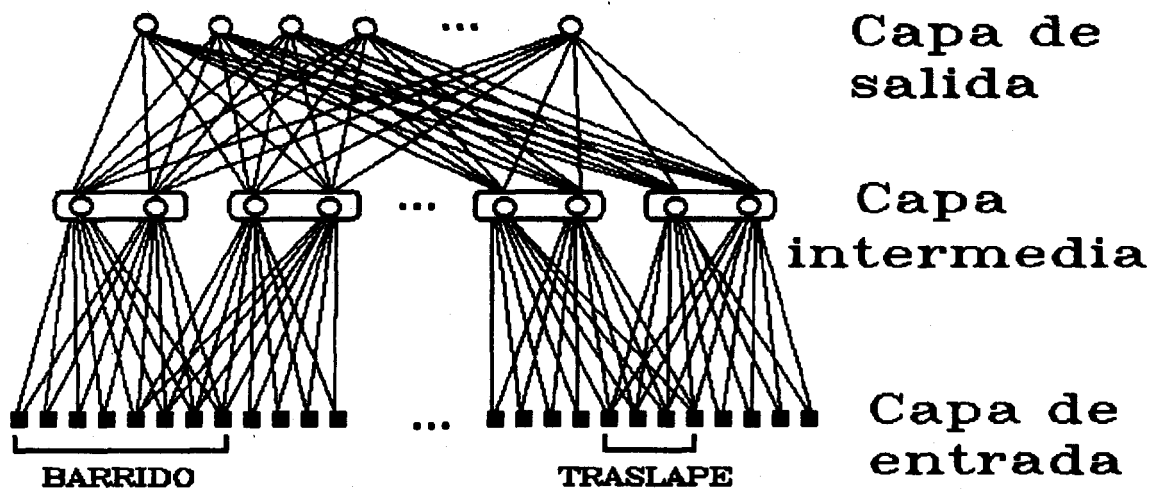


Fig. 3.2. Diagrama esquemático de la arquitectura de una red parcialmente invariante a los desplazamientos de la señal de entrada. El número de nodos dentro de cada grupo en la capa intermedia puede variarse.

La figura 3.2 muestra la arquitectura de la red empleada. Se puede apreciar que los elementos de la capa intermedia se han reunido y que cada grupo recibe ahora señales de cierta región de la capa de entrada y no de todos sus nodos, como era el caso anterior. También, hay un *traslape* entre las regiones *barridas* por cada grupo de nodos. Dichos valores, junto con el número de nodos de la capa precedente, serán los que determinen el número total de grupos, y de nodos, en la capa intermedia. Puesto que la cantidad de grupos debe ser un número entero, algunas combinaciones de los valores de *traslape*, *barrido* y número de nodos en la capa precedente no serán compatibles.

Cada nodo de la capa de salida está conectado con todos los nodos de la última capa intermedia, al igual que en el caso de la arquitectura convencional. Si se asume que la señal de entrada cambia con el tiempo, el intervalo *barrido* por cada uno de los grupos de nodos de la capa intermedia puede considerarse como el número de divisiones temporales abarcadas por cada uno de ellos, en otras palabras, cada grupo de nodos de la capa intermedia "examina" ventanas temporales diferentes de la señal de entrada. Ya que se quiere que las traslaciones de la señal de entrada no

cambien la respuesta de la red, es necesario que los conjuntos de pesos entre las capas de entrada e intermedia sean iguales, o sea, que cada grupo de nodos tenga la misma respuesta ante secuencias temporales iguales. Este enfoque de la red es empleado por Waibel et al. (1989), que las denomina redes neuronales con retardos temporales. Sin embargo, es posible considerarla como si cada grupo de nodos en la capa intermedia fuese un detector de características (Bottou et al., 1990); puesto que cada grupo tiene el mismo juego de pesos, responderá de igual manera a características morfológicas semejantes de la señal de entrada. Si se emplea más de una capa intermedia, las conexiones entre ellas tendrán la misma estructura que la que hay entre aquellas de las capas de entrada y primera intermedia.

La restricción que los pesos de cada grupo permanezcan iguales se consigue de la siguiente forma: se efectúa el cálculo de los cambios necesarios en los pesos de cada uno de los grupos, como si fuesen diferentes, se suman luego los cambios para los pesos equivalentes de cada grupo y éste resultado es el empleado para actualizarlos. Una justificación teórica del por qué es esta suma el factor de corrección y no su promedio, como podría pensarse a primera vista, se encuentra en LeCun (1988).

3.2. RESULTADOS DEL APRENDIZAJE

Ya que no es muy frecuente encontrar más de ocho clases diferentes de impulsos en un registro multicejular con electrodos de tungsteno convencionales y, además, con el fin disminuir el tiempo de entrenamiento, se redujo el número de categorías por clasificar a ocho. Los paradigmas de las clases utilizadas se muestran en la figura 3.3, y se eligieron al azar de los impulsos mostrados antes en la figura 2.1 (págs. 47-48). Se usó una representación localizada en la capa de salida, es decir, se emplearon ocho nodos, uno para cada categoría. En la de entrada se usaron 128 la mayoría de las veces, aunque también se experimentó con 72 y 64 nodos de entrada.

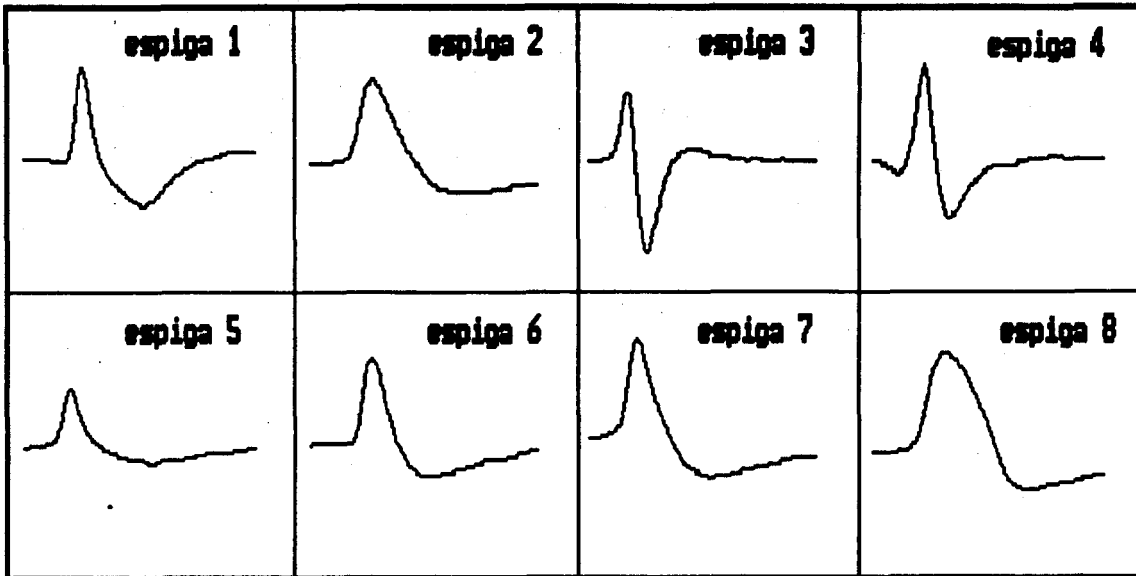


Fig. 3.3. Paradigmas de las ocho clases por ser aprendidas. Las nuevas clases corresponden a los impulsos 1, 13, 24, 25, 42, 43, 52 y 64, respectivamente, de la figura 2.1 (págs. 47-48).

Para valorar el desempeño del clasificador se empleó el método conocido como H (Duda y Hart, 1973), que consiste en usar un número de patrones para el entrenamiento y otro, con patrones diferentes, para las pruebas. Aunque no sea precisamente el ideal para evaluar el error de un clasificador, otras técnicas, como las de validación cruzada, "bootstrapping" y "leave-one-out", resultan costosas computacionalmente en este caso, pues requieren que el aprendizaje se efectúe con muchas configuraciones diferentes del conjunto de entrenamiento (Weiss y Kulikowski, 1991; Fukunaga y Hayes, 1989). Para el conjunto de entrenamiento se usaron 240 muestras, 30 para cada clase, con ruido de hasta un 30%. En el de prueba se utilizaron 60 señales para cada clase, divididas en tres grupos de 20 cada uno con ruido del 30%, 50% y 70%, respectivamente.

Se encontró que con el fin de mejorar la tolerancia de la red a los desplazamientos de la señal de entrada era conveniente presentar los patrones del conjunto de entrenamiento con variaciones aleatorias de estas traslaciones. En general, se utilizó un corrimiento entre 0 y 20 puntos con respecto a las señales mostradas en la figura 3.3.

Ya que existe un número considerable de variables (velocidad de aprendizaje, variación del momento, número de capas intermedias, número de entradas de barrido y traslape de cada grupo de nodos en las capas intermedias, cantidad de nodos dentro de cada grupo, actualización de pesos continua o periódica, presentación de patrones en un orden determinado o aleatorio, valores iniciales de los pesos), el criterio para comparar el desempeño de diferentes configuraciones no es tan sencillo. Sin embargo, si se tienen redes distintas con el mismo promedio efectivo de errores al clasificar los patrones de prueba, la más adecuada será aquella más simple, es decir, aquella que requiera la ejecución del menor número de operaciones para el cómputo de su respuesta.

En general, se halló que es suficiente el uso de una sola capa intermedia para llevar a buen término el aprendizaje. En la tabla 3.2 se resume el resultado obtenido con diferentes redes que tienen una capa intermedia. En todos los casos, los patrones del conjunto de entrenamiento se presentaron en el mismo orden, específicamente, un patrón de cada clase en forma alternada. Los datos mostrados para cada modelo corresponden al promedio de tres soluciones obtenidas a partir de condiciones iniciales diferentes. La columna OP indica el número de operaciones necesario para obtener la respuesta de la red, sin contar el cálculo de la función de activación en los elementos de las capas intermedia y salida, y el número de parámetros libres de la red, teniendo en cuenta que los umbrales de todos los elementos procesadores se calcularon por separado, incluso para los diferentes grupos de la capa intermedia. Cabe mencionar que la red 13 de la tabla 3.2 es equivalente a una red de arquitectura convencional con 15 nodos en la capa intermedia.

**Tabla 3.2.
Resultados de redes con una capa intermedia.**

RED	B	T	NNG	NTN	OP	Ep	%30	%50	%70
1	32	30	1	49	4034/481	371	96	98	86
2	32	29	1	33	2722/337	244	96	94	86
3	32	28	1	25	2066/265	502	94	98	88
4	32	26	1	17	1410/193	454	92	87	78
5	32	26	2	34	2804/378	80	98	99	94
6	32	24	1	13	1082/157	654	94	91	80
7	32	24	2	26	2148/306	104	98	98	87
8	32	20	1	9	754/121	1260	78	78	66
9	32	20	2	18	1492/234	142	96	98	90
10	28	24	1	26	1940/270	361	100	94	86
11	36	32	1	24	2176/260	526	100	96	86
12	64	60	1	33	4834/369	228	92	80	78
13	128	0	15	15	4126/2063	252	94	97	92

B: Número de entradas a cada grupo de la capa intermedia.
T: Número de puntos de traslape entre las entradas de los diferentes grupos de nodos de la capa intermedia.
NNG: Número de nodos en cada grupo de la capa intermedia.
NTN: Número total de nodos en la capa intermedia.
Ep: Epocas necesarias para disminuir el error total medio a 0.01.
OP: Número de operaciones que se efectúan para calcular la respuesta de la red (sumas y multiplicaciones) y número de pesos y umbrales de la red.
%30: Porcentaje correctamente clasificado cuando el conjunto de prueba tiene 30% de ruido (SNR = 10.5 dB).
%50: Porcentaje correctamente clasificado cuando el conjunto de prueba tiene 50% de ruido (SNR = 6 dB).
%70: Porcentaje correctamente clasificado cuando el conjunto de prueba tiene 70% de ruido (SNR = 3 dB).
 128 nodos de entrada y 8 de salida.
 Velocidad de aprendizaje β : 0.3.
 Variación del momento μ : 0.2.

Tabla 3.3
Resultados de redes con dos capas intermedias.

RED	BT1	NNG1	NTN1	BT2	NTN2	OP	%30	%50	%70
1	32-24	1	13	4-3	10	1134/115	88	80	72
2	32-24	2	26	4-3	10	2072/196	96	90	72
3	32-20	2	18	3-2	7	1386/124	94	88	72
4	16-12	1	29	9-7	11	1398/162	84	86	62
5	16/12	2	58	9-7	11	2582/199	95	97	83
6	26/29	2	68	7-4	10	4556/209	86	82	62
7	8/5	1	41	8-5	12	1162/173	81	90	76

BT1: Número de entradas a cada grupo de la capa intermedia 1 y número de puntos de traslape entre dichos grupos.

NNG1: Número de nodos en cada grupo de la capa intermedia 1.

NTN1: Número total de nodos en la capa intermedia 1.

BT2: Número de entradas a cada grupo de la capa intermedia 2 y número de puntos de traslape entre dichos grupos.

NTN2: Número total de nodos en la capa intermedia 2.

OP: Número de operaciones que se efectúan para calcular la respuesta de la red (sumas y multiplicaciones) y número de pesos y umbrales de la red.

%30: Porcentaje correctamente clasificado cuando el conjunto de prueba tiene 30% de ruido (SNR = 10.5 dB).

%50: Porcentaje correctamente clasificado cuando el conjunto de prueba tiene 50% de ruido (SNR = 6 db).

%70: Porcentaje correctamente clasificado cuando el conjunto de prueba tiene 70% de ruido (SNR = 3 dB).

128 nodos de entrada y 8 de salida.

Velocidad de aprendizaje β : 0.3.

Variación del momento μ : 0.2.

Promedio de tres corridas para cada red.

Los resultados conseguidos con redes con dos capas intermedias son menos convincentes. La tabla 3.3 muestra los resultados para algunas de las redes experimentadas.

Tabla 3.4
Resultados de redes con una capa intermedia y 64 nodos de entrada

RED	B	T	NNG	NTN	OP	Ep	%30	%50	%70
1	16	12	1	13	666/141	5340	86	80	76
2	16	12	2	26	1316/258	610	94	90	75
3	16	13	1	17	866/177	2890	86	86	66
4	32	28	2	9	754/121	545	90	78	52

B: Número de entradas a cada grupo de la capa intermedia.
T: Número de puntos de traslape entre las entradas de los diferentes grupos de nodos de la capa intermedia.
NNG: Número de nodos en cada grupo de la capa intermedia.
NTN: Número de nodos en la capa intermedia.
Ep: Epocas necesarias para disminuir el error total medio a 0.01.
OP: Número de operaciones que se efectúan para calcular la respuesta de la red (sumas y multiplicaciones) y número de pesos y umbrales de la red.
%30: Porcentaje correctamente clasificado cuando el conjunto de prueba tiene 30% de ruido.
%50: Porcentaje correctamente clasificado cuando el conjunto de prueba tiene 50% de ruido.
%70: Porcentaje correctamente clasificado cuando el conjunto de prueba tiene 70% de ruido.
64 nodos de entrada y 8 de salida.
Velocidad de aprendizaje β : 0.3.
Variación del momento μ : 0.2.
Promedio de tres corridas para cada red.

En la tabla 3.4 aparecen los resultados obtenidos al emplear 64 nodos de entrada, tomando uno de cada dos puntos de la señal original de 128. Cuando se usaron 72 nodos de entrada, correspondientes a los primeros 40 puntos de la señal original, luego a uno de cada dos, del 40 al 80, y a uno de cada cuatro, del 80 al 128, los resultados fueron verdaderamente pobres, alrededor del 40% de reconocimientos correctos, por lo que ni siquiera se tabulan.

Cuando cada valor de la señal de entrada se normaliza teniendo en cuenta su distancia al valor medio y su relación

respecto a la desviación estándar, el número de iteraciones requerido para obtener el mismo error total promedio se aumenta en un factor de tres, aproximadamente. Esto es comprensible si se piensa que la tarea de normalización disminuye la distancia, en el espacio de entrada, entre las regiones de cada clase, con lo cual la tarea de separación que debe efectuar el clasificador se hace más delicada, tomando más tiempo. Igualmente, la menor distancia entre clases hace que la red sea más propensa a clasificar equivocadamente a las señales ruidosas.

En la figura 3.4 se puede apreciar las variaciones que se obtienen en el número requerido de iteraciones del conjunto de entrenamiento para alcanzar un error medio total de 0.01, cuando se cambia la manera de presentar los patrones. Cada iteración equivale a enseñar todos y cada uno de los ejemplares del conjunto de entrenamiento, 240 en este caso. En (a) la actualización de los pesos es en forma continua, es decir, para cada patrón de entrada y el orden de presentación fijo; el procedimiento es similar en (b), pero con diferentes valores iniciales de los pesos; (c) y (d) tienen los mismos valores iniciales de los pesos que (a), pero mientras que en (c) el orden de presentación es aleatorio, en (d) la actualización se hizo por épocas, o sea, luego de haber presentado todos los patrones. La red del ejemplo es una del tipo 6 de la tabla 3.2, es decir 13 grupos de nodos en la capa intermedia, cada grupo con un elemento procesador, un barrido de 32 puntos del patrón de entrada y un traslape entre las medidas de los diferentes grupos de 24 puntos. En todos los casos la velocidad de aprendizaje fue de 0.3 y la variación del momento, μ , de 0.2.

La trayectoria seguida durante el entrenamiento por uno de los pesos de la red cuya variación del error se ilustra en la figura 3.4 (a) se puede apreciar en la figura 3.5. Aunque en ese caso el valor del peso se actualizó con la presentación de cada uno de los patrones, sólo se indica su valor al terminar cada presentación del conjunto de entrenamiento. El valor inicial del peso fue asignado al azar.

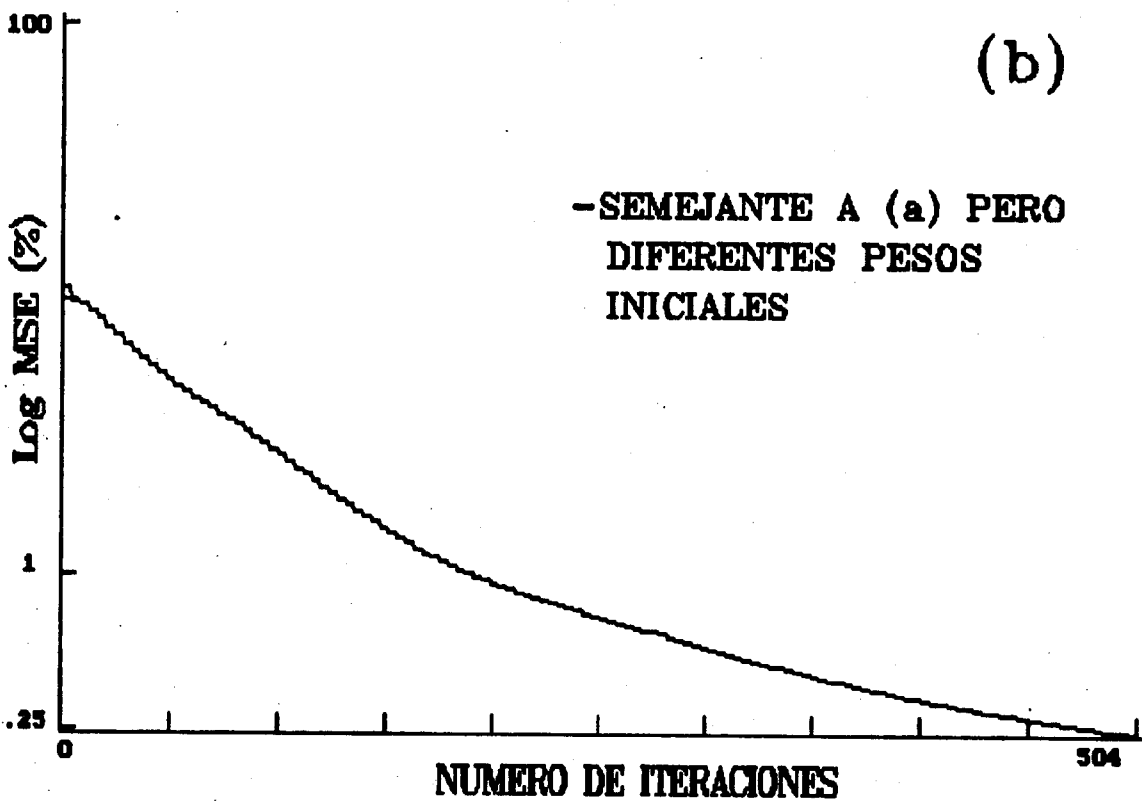
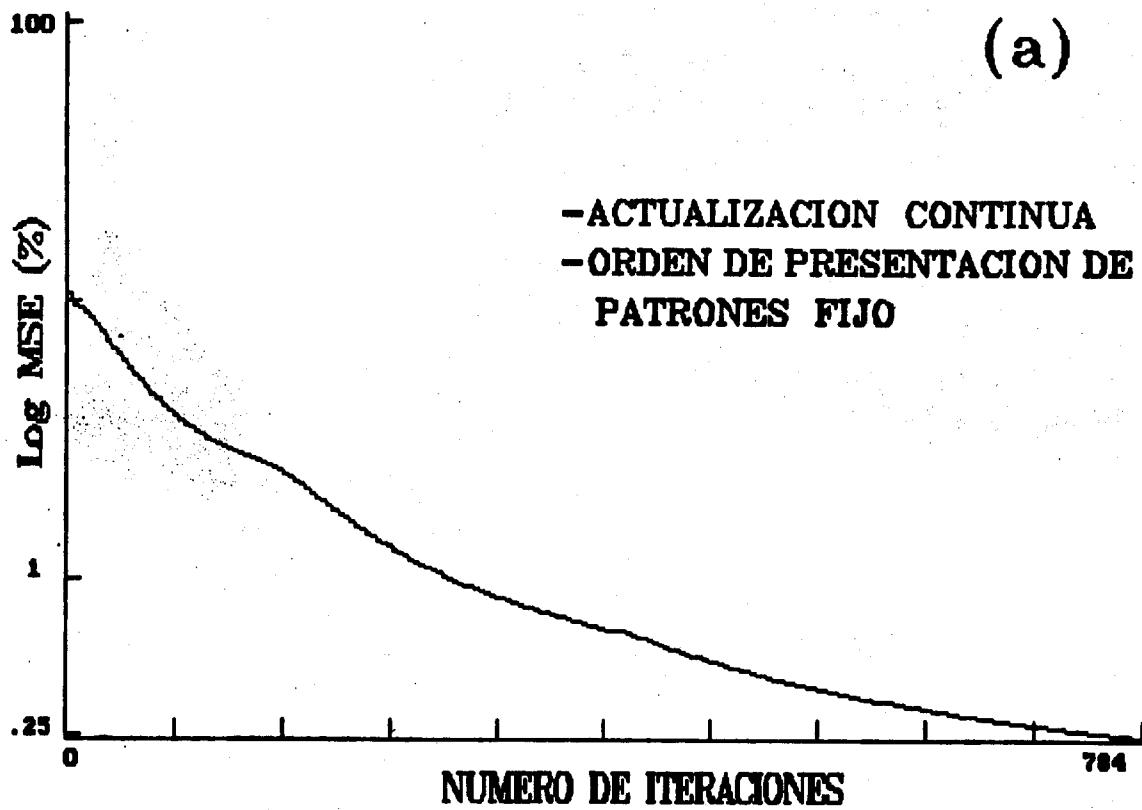


Fig. 3.4. Variaciones del error cuadrático medio para una misma red en diferentes condiciones.

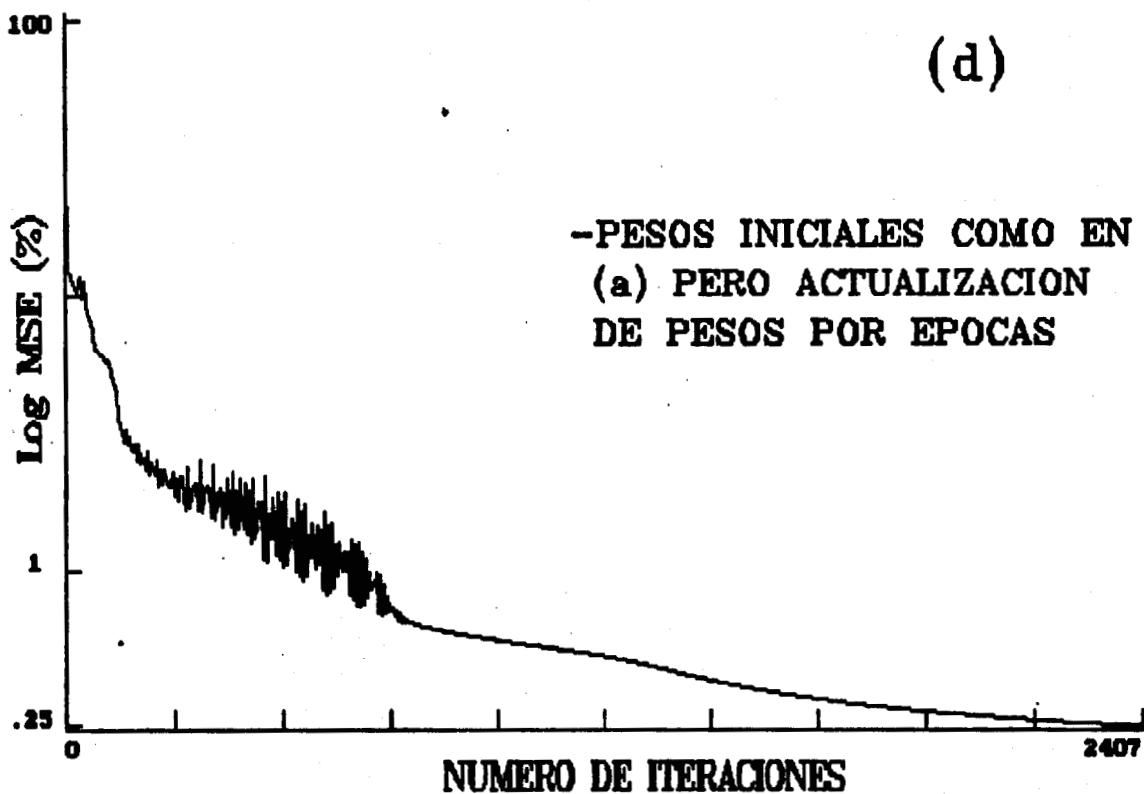
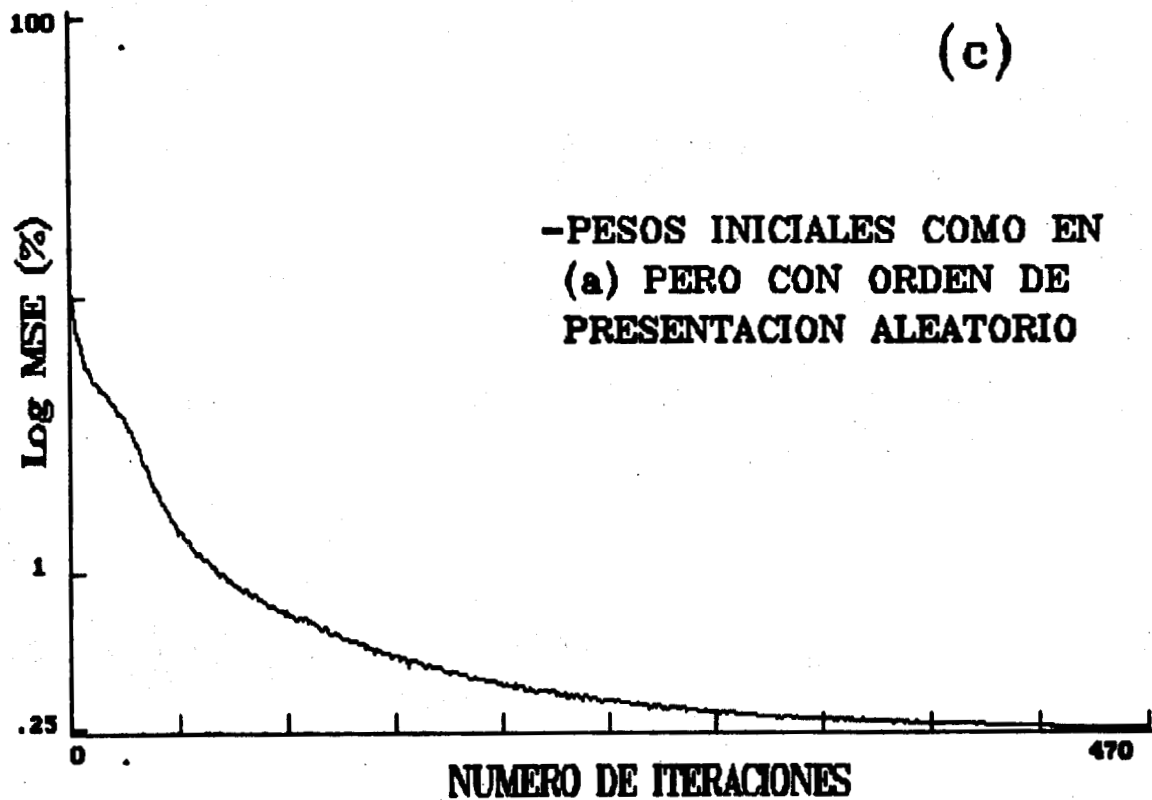


Fig. 3.4 (cont.). Variaciones del error cuadrático medio para una misma red en diferentes condiciones.

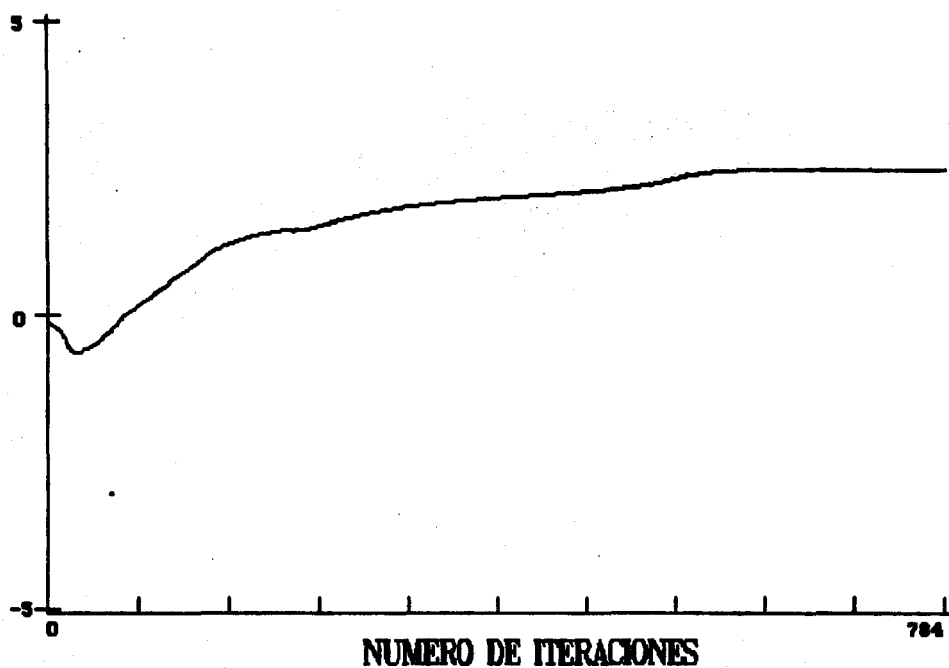


Fig. 3.5. Cambios del valor de uno de los pesos durante el entrenamiento de la red cuya variación del error aparece en la figura 3.4 (a).

En la tabla 3.5 se puede ver la matriz de confusión para una de las redes del tipo 5 de la tabla 3.2. Cada renglón indica cómo ha sido reconocida cada una de las ocho clases diferentes al evaluar 60 ejemplos de cada una; los elementos de la diagonal principal corresponden al número de muestras clasificadas correctamente. Por ejemplo, para los impulsos del tipo 2, 57 han sido clasificados correctamente y los tres restantes se asignaron uno a cada una de las clases 5, 6 y 7. Como ya se había dicho, los ejemplos usados en la evaluación son diferentes a los empleados durante el entrenamiento.

La figura 3.6 muestra los diagramas de estrella que representan los valores finales de los pesos encontrados en las redes de los casos (a) y (d) de la figura 3.4. Cada estrella tiene un número de "brazos" igual al número de pesos a la entrada del nodo correspondiente, la longitud de éstos es proporcional al valor del peso y los extremos de cada "brazo" se han conectado para formar el perímetro de la estrella.

Tabla 3.5
Matriz de confusión para una red con una capa intermedia
(red del tipo 5 de la tabla 3.2)

Clase entrada	Reconocidos como de la clase							
	1	2	3	4	5	6	7	8
1	58	0	0	1	0	1	0	0
2	0	57	0	0	1	1	1	0
3	0	0	60	0	0	0	0	0
4	2	0	0	56	2	0	0	0
5	0	0	0	0	60	0	0	0
6	1	0	0	0	1	55	3	0
7	1	0	0	0	0	1	58	0
8	0	0	0	0	0	0	0	60

Para la figura 3.6, teniendo como referencia la red de la figura 3.2 (pág. 60), el peso que conecta con el primer nodo (de izquierda a derecha) de la capa inferior está dibujado en la posición de las tres de las manecillas del reloj y los siguientes se ubican siguiendo el movimiento antihorario, a distancias angulares proporcionales. Puesto que el conjunto de pesos de los grupos de nodos de una misma capa intermedia es igual para todos, solamente es necesario dibujarlo una vez, lo que no ocurre con los pesos de las conexiones entre la última capa intermedia y la de salida, que deben representarse para cada uno de los nodos de salida. La circunferencia separa los valores negativos de los positivos; los positivos quedan en su interior. En la figura 3.6 no se ha graficado el valor de los umbrales de los diferentes nodos.

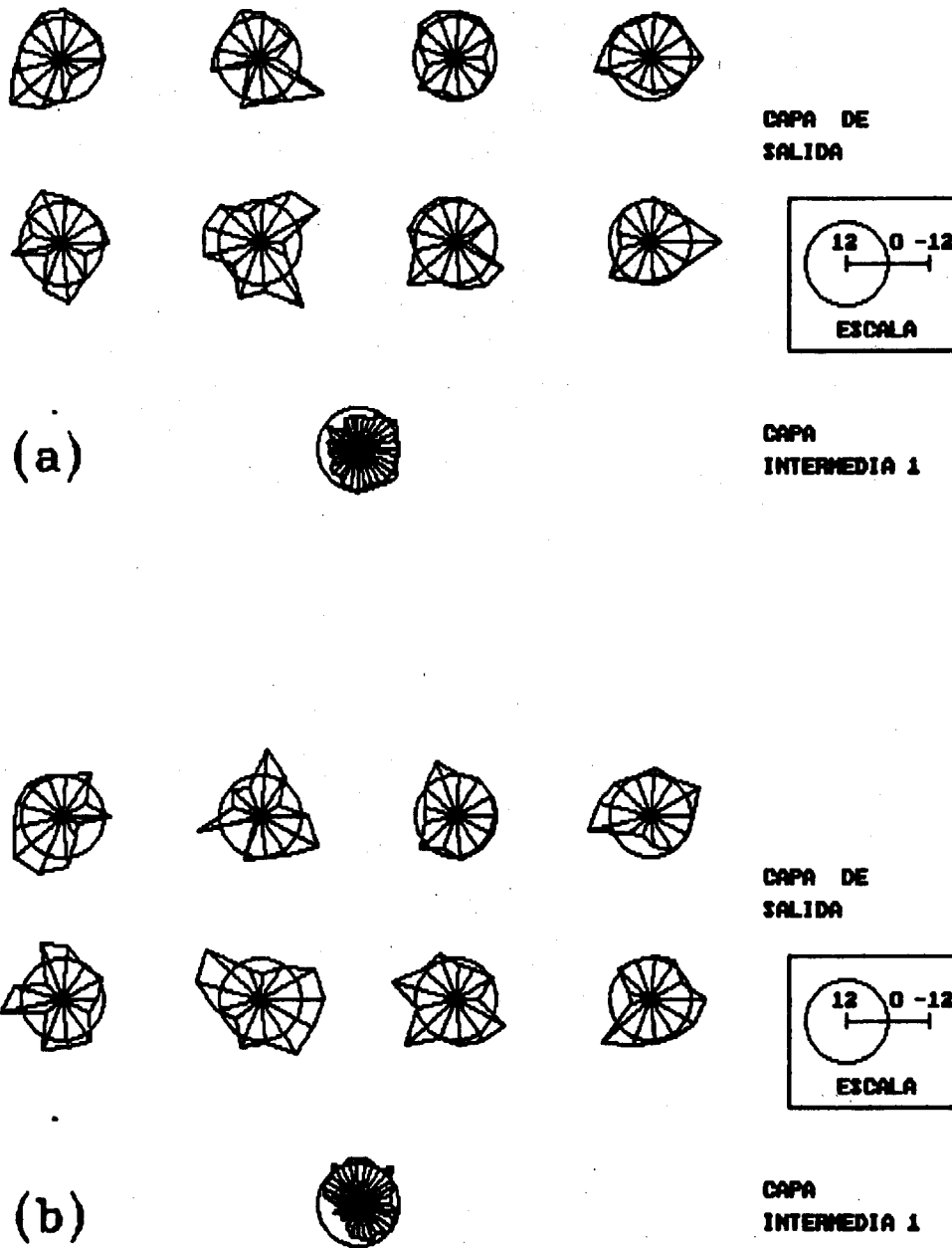


Fig. 3.6. Diagramas de estrellas con los valores finales de los pesos obtenidos por las redes cuyo error se graficó en: (a) la figura 3.4 (a) y (b) la figura 3.4 (d).

3.3. REPRESENTACION INTERNA Y GENERALIZACION

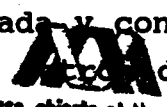
Uno de los atributos distintivos de todo sistema con capacidad de aprendizaje debe ser su habilidad para generalizar. Así, los patrones enseñados se consideran como ejemplos de una clase determinada y, a partir de su conocimiento, es posible identificar otros patrones que pertenezcan a la misma clase. En otras palabras, el sistema debe ser capaz de "encontrar" el prototipo de una clase con base en los ejemplos que le son enseñados.

Ahora bien, en su forma más simple este comportamiento requiere la comparación de las "similitudes" entre los diferentes patrones, pero en muchas tareas se requiere algo más que eso: es necesario ejecutar la clasificación de las señales de entrada como una función de algunas de sus características. La última clase de comportamiento está mucho más cerca de lo que sería una conceptualización genuina, que la de la simple comparación de patrones. Un sistema con esta característica debe ser capaz de desarrollar una especie de representación interna abstracta de la tarea que ha aprendido.

En una red neuronal, los pesos de las conexiones hallados durante el aprendizaje son los que codifican el "conocimiento" de la red. Sin embargo, al inspeccionar visualmente los valores de los pesos, a menudo no es fácil comprender la estrategia de clasificación encontrada por la red. Por ejemplo, al examinar los pesos graficados en la figura 3.6, aunque en ambos casos los valores finales sean vagamente semejantes entre sí, es casi imposible deducir qué peculiaridades de las señales de entrenamiento están siendo consideradas por la red. Esta situación muestra un agudo contraste con lo que ocurre en los sistemas con aprendizaje simbólico automático de la inteligencia artificial tradicional, como los sistemas expertos, donde es mucho más sencillo seguir la cadena de "razonamientos" empleada por el sistema para alcanzar una conclusión determinada.

Puesto que las diferentes señales de entrada interactúan de distinta manera con los pesos para determinar la actividad de los

elementos de la capa intermedia y, a su vez, ésta con los pesos que llegan a los nodos de la capa de salida, resulta conveniente estudiar la actividad en las capas intermedias para poder comprender de una manera más clara el comportamiento de la red clasificadora. De tal forma, la actividad de cada uno de los nodos de las capas intermedias puede ser usada para entender la estrategia global de la red (Gorman y Sejnowski, 1988).

En la figura 3.7 se puede apreciar el patrón de actividades en las capas de salida e intermedia de una red del tipo 6 de la tabla 3.2 (128 nodos de entrada, 13 nodos en la intermedia, cada uno barriendo 32 nodos de la de entrada y con un traslape de 24, y 8 en la capa de salida) cuando un  de la clase 1 le es presentado. Los pesos de la red son los que aparecen en la figura 3.6 (a). El producto de los pesos por la señal de entrada se ha ilustrado a la manera de "paisajes energéticos". Para los nodos de la capa intermedia, ya que únicamente reciben información de una ventana temporal de la señal de entrada, el producto de los pesos que llegan a un nodo por la señal de entrada es diferente de cero únicamente durante tal ventana. Hay una línea horizontal para cada nodo (13 en la intermedia) y su desplazamiento vertical indica el producto de los pesos que llegan al nodo por el valor de la señal de entrada en el intervalo barrido por el nodo. En el caso mostrado, el nodo de la capa intermedia que "recibe" los primeros 32 puntos del patrón de entrada se ha dibujado en la línea superior y en la inferior aquel que "recibe" los 32 puntos finales, y la "actividad" es diferente de cero sólo en la diagonal. Los pesos entre la capa intermedia y la de salida se han multiplicado por el valor de la activación de los nodos de la capa intermedia y todos los nodos de una capa están conectados con todos los de la otra. Los diagramas de Hinton representan el valor de la salida de los nodos en las capas intermedia y de salida. Si la salida es mayor que 0.9, el cuadrado es negro completamente; por el contrario, si el cuadrado es totalmente blanco indica que la salida del nodo correspondiente es menor que 0.1. El nodo activado a la salida indica la clase asignada al patrón de entrada.

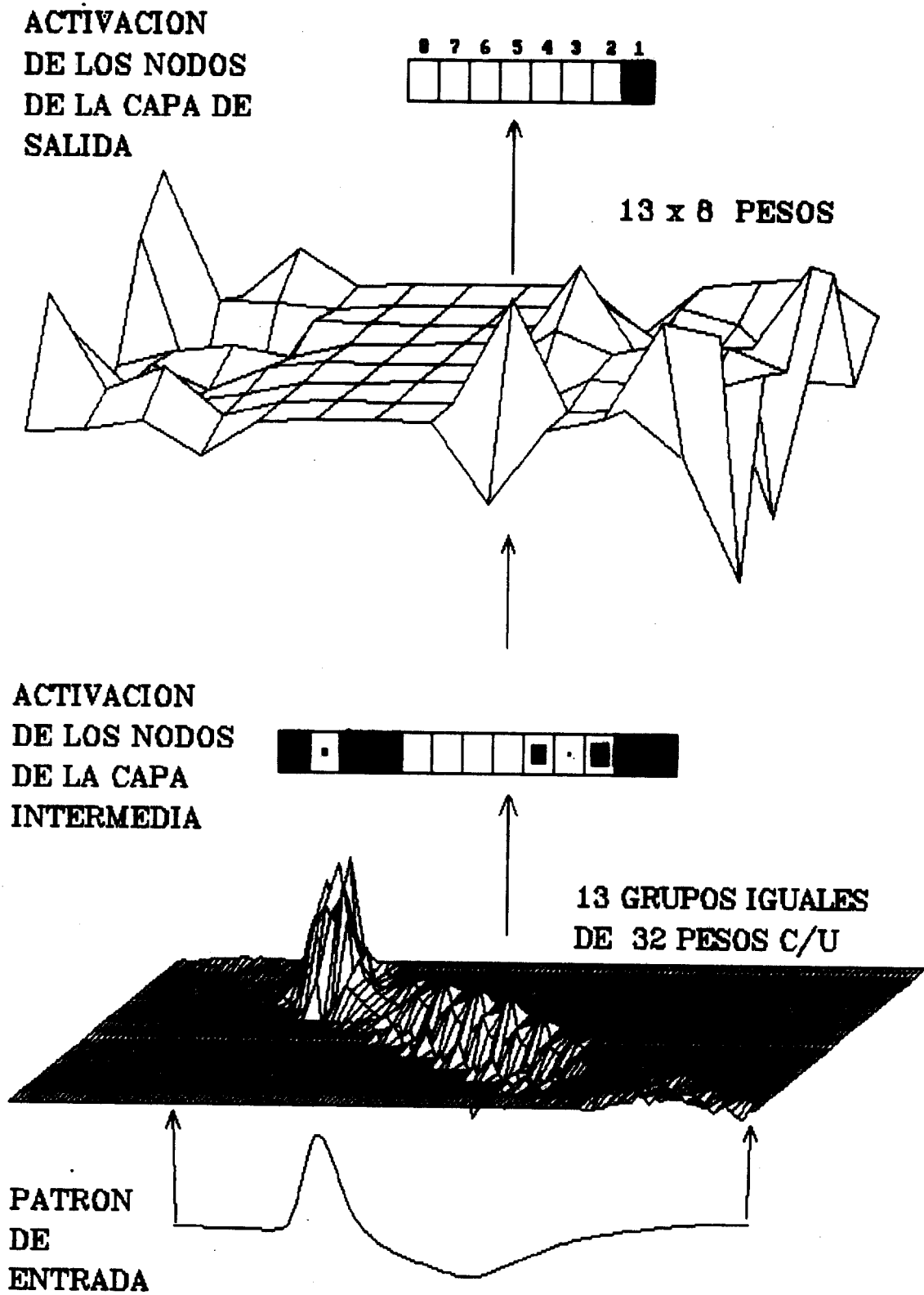
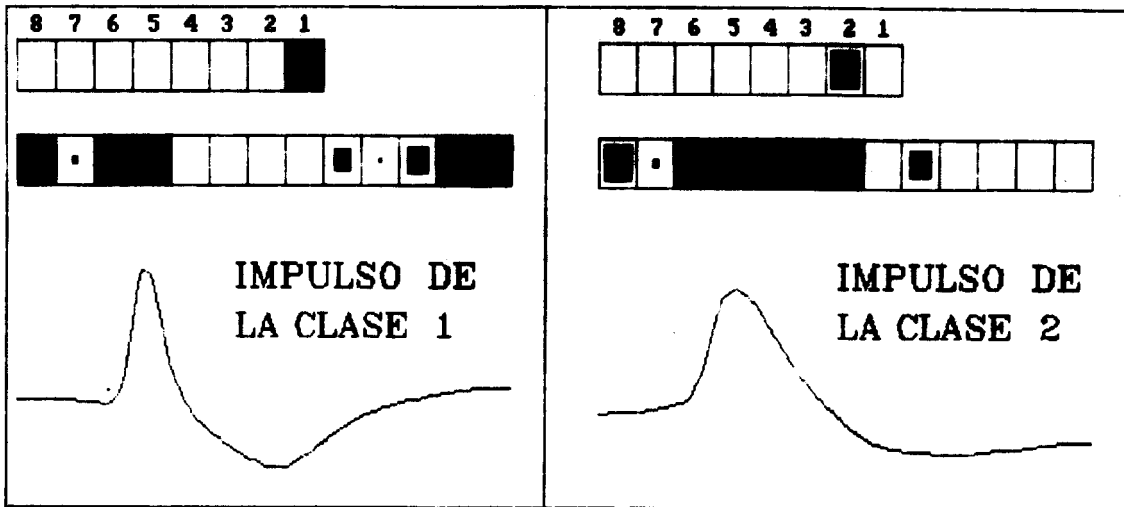
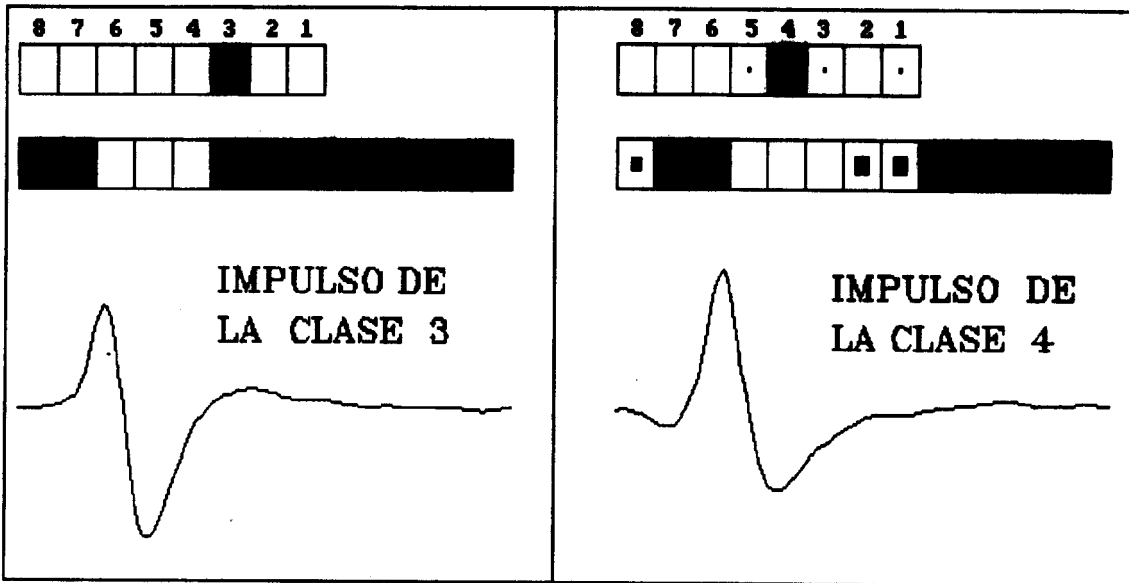


Fig. 3.7. Patrones de activación en una red con una capa intermedia. Véase el texto para la explicación.

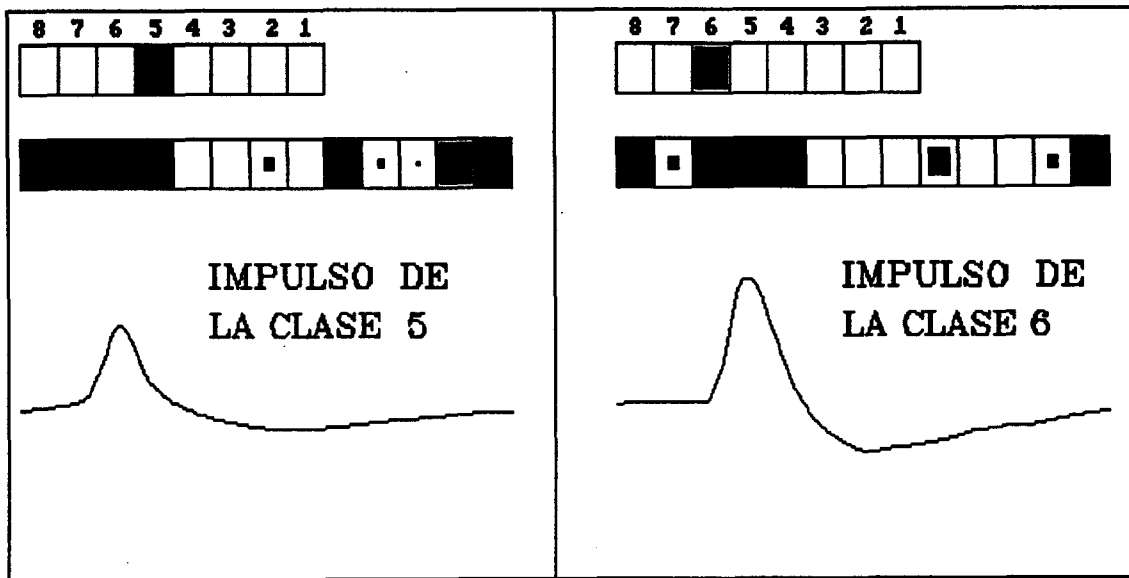


(a)

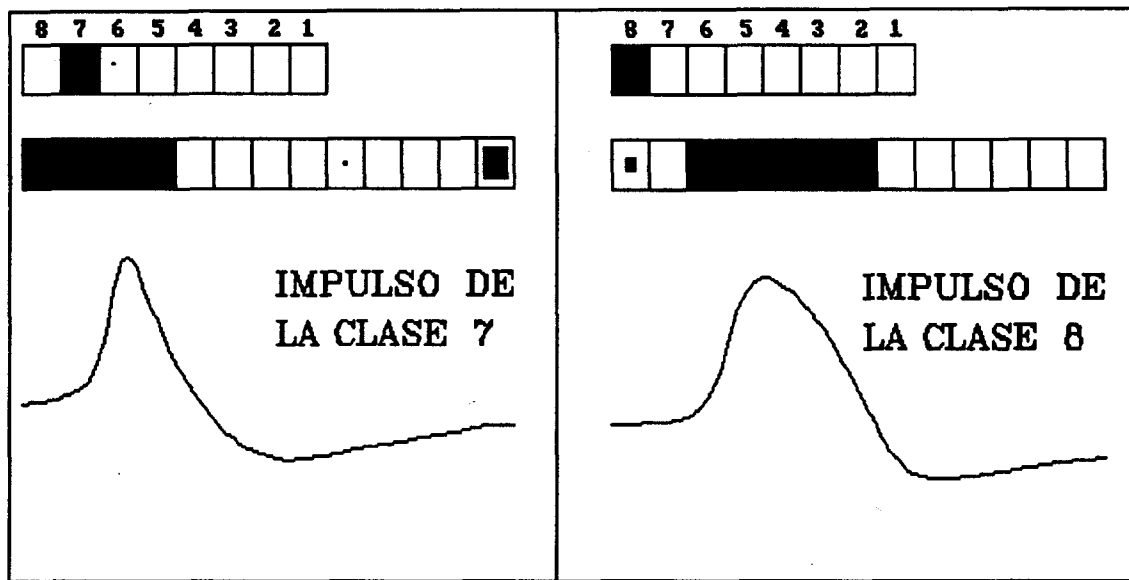


(b)

Fig. 3.8. Actividad en las capas intermedia y salida para clases diferentes de patrones de entrada: (a) 1 y 2, (b) 3 y 4.



(c)



(d)

Fig. 3.8 (cont.). Actividad en las capas intermedia y salida para clases diferentes de patrones de entrada: (c) 5 y 6, (d) 7 y 8.

La figura 3.8 ilustra la actividad en la capa intermedia, para la misma red que en la figura 3.7, cuando los patrones de entrada corresponden a los distintos prototipos de cada una de las clases, que aparecen en la figura 3.3. En los ejemplos mostrados, la señal de entrada se encuentra sin ruido. Se puede apreciar que la actividad en los nodos de la capa intermedia alcanza los valores extremos en la mayoría de los casos, bien sea cero o uno, y son pocos los casos en que se produce un valor intermedio. Aunque se pueden asociar activaciones cercanas a uno con aquellos nodos para los cuales la amplitud de la señal de entrada es positiva o con su fase de elevación, la representación interna está más bien distribuida en todos los nodos.

En la figura 3.9 se puede observar cómo otros dos patrones de la clase 1 son clasificados correctamente, a pesar del ruido presente, y cómo la actividad en la capa intermedia sigue siendo aproximadamente igual que en el caso sin ruido (fig. 3.8 (a)); aun cuando para la señal con mayor porcentaje de ruido, la activación del nodo de salida 4 empieza a ser considerable, la actividad en el correspondiente a la clase 1 sigue siendo mayor.

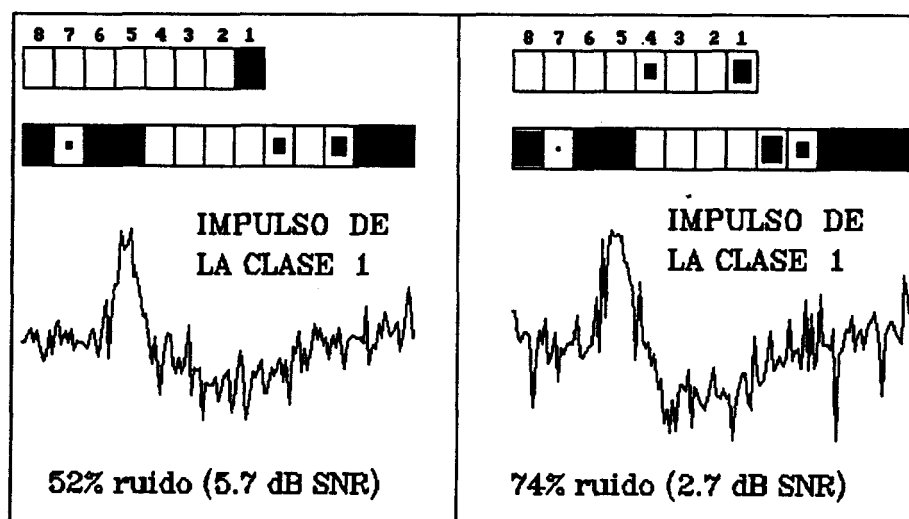


Fig. 3.9. Patrones de activación en las capas intermedia y de salida con señales ruidosas de entrada. En ambos casos la señal de entrada es de la clase 1 de la figura 3.3 (pág. 62).

La capacidad de generalizar puede apreciarse también al analizar la respuesta de la red ante señales que de antemano se sabe no corresponden a ninguna de las clases enseñadas a la red. Tomando como patrones de prueba a los impulsos que aparecen en la figura 2.1 (págs. 47-48) y como clases aprendidas a las de las espigas de la figura 3.3 (pág. 62), la tabla 3.6 resume la manera como fueron clasificadas las diferentes señales empleando el promedio obtenido con ocho redes distintas. Por ejemplo, los impulsos 22 y 63 de la figura 2.1 se asignaron a la misma clase que del impulso 64, que fue empleado como paradigma de la clase 8 (espiga 8 de la figura 3.3).

Tabla 3.6
Clasificación de las señales de la figura 2.1 (págs. 47-48)
en las 8 categorías de la figura 3.3 (pág. 62).

Clase asignada	Impulsos de la figura 2.1 considerados de esta clase
1	1, 2, 3, 5, 6, 7, 53.
2	<u>13</u> , 12, 16, 32, 50, 57, 58, 60, 61.
3	<u>24</u> , 27.
4	<u>25</u> , 4, 11, 15, 18, 21, 23, 26, 29, 30, 31, 62.
5	<u>42</u> , 10, 14, 17, 19, 20, 28, 33, 34, 35, 38, 39, 40, 41, 44, 45, 46, 47, 48, 49, 50, 55.
6	<u>43</u> .
7	<u>52</u> , 8, 9, 36, 37, 51, 54, 59.
8	<u>64</u> , 22, 63.

Los resultados se obtuvieron al evaluar la respuesta de 8 redes diferentes. La clase mostrada es la de mayor ocurrencia para las distintas redes. Los números subrayados corresponden a los prototipos de cada clase empleados en el aprendizaje (espigas de la figura 3.3 de la página 62).

La figura 3.10 (página 81) muestra algunos casos de clasificación de los impulsos de la figura 2.1 (págs. 47-48). La red usada es la misma que la de las figuras 3.7 a 3.9. En los casos de la parte superior, se puede comprobar que el patrón de actividad en la capa intermedia es muy parecido a aquel de los prototipos para la clase encontrada y que la asignación de clase ocurre sin ambigüedad. Por otro lado, en la parte inferior se muestra un caso en donde la respuesta de dos nodos de salida es comparable y otro en donde, a pesar de que se activa un solo nodo de salida, el patrón de activación en la capa intermedia no se parece al de la clase original.

3.4. CLASIFICACION DE IMPULSOS EN UNA TRAZA NERVIOSA

Con el propósito de probar el sistema integrado de detección y clasificación, se construyó una traza artificial con más de 500 impulsos, de cinco clases diferentes, correspondientes a las cinco primeras espigas de la figura 3.3 (página 62). Asumiendo que un tren de espigas nerviosas es un proceso en el que la probabilidad de ocurrencia de un impulso depende sólo del tiempo transcurrido desde la aparición del último y no de los eventos precedentes a éste, lo que es cierto apenas como una primera aproximación, si los tiempos de disparo de un grupo de neuronas no están correlacionados, el patrón múltiple de disparos tendrá una distribución aleatoria de Poisson. Tal idea está de acuerdo con la noción que la superposición de muchos procesos puntuales independientes produce dicha distribución (Griffith, 1971).

Como es de esperarse, la detección de los impulsos depende críticamente de la elección del umbral A de la ecuación (2.2). Si A es muy elevado habrá muy pocas detecciones y si tiene un valor muy bajo, el ruido presente puede dar origen a falsos negativos. El valor de A debe ser ajustado de manera experimental para cada tipo de prueba, de acuerdo con la amplitud de los impulsos esperados y de la del ruido presente. Un valor apropiado de A permite detectar correctamente todos los pulsos.

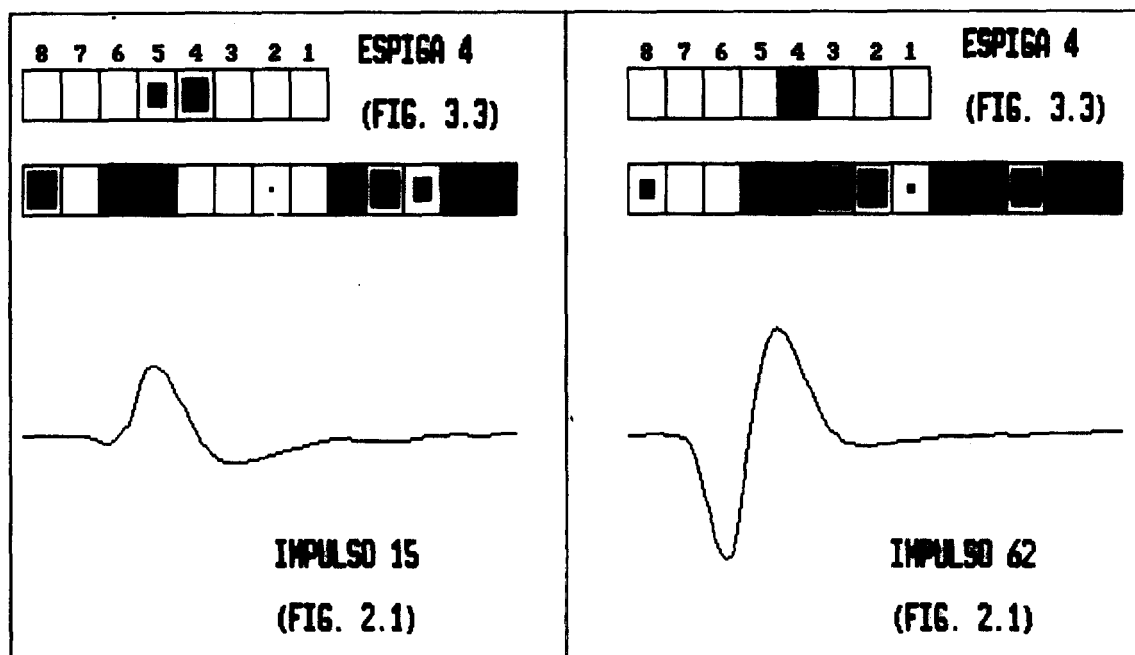
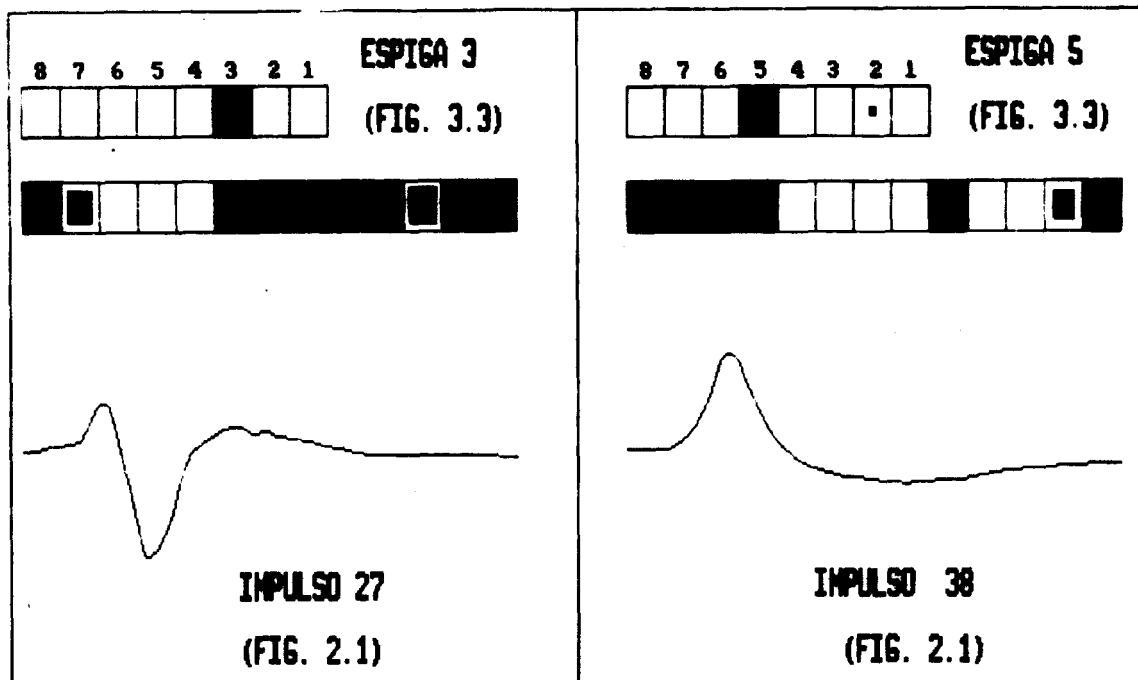


Fig. 3.10. Patrones de actividad en las capas intermedias y de salida al clasificar algunos de los impulsos de la figura 2.1.

Conociendo el tiempo de ocurrencia de un impulso, los datos necesarios para su identificación pueden ser pasados fácilmente al clasificador. Es necesario tener en cuenta, no obstante, que el momento de la detección no corresponde necesariamente con el comienzo de la señal con la que la red fue entrenada, por lo cual es preciso tomar unas cuantas muestras previas al momento de la detección en el arreglo que es entregado a la red de clasificación. También conviene recalcar que los datos que son entregados al clasificador corresponden a los de la señal original, sin el ruido removido.

La figura 3.11 muestra el diagrama en bloques del sistema empleado para analizar registros múltiples. En la figura 3.12 se muestra un tramo del registro analizado, luego de haber removido el ruido, que es del orden del 30%. Un histograma de la distribución de los tiempos entre impulsos, obtenidos luego de su detección, se ilustra en la figura 3.13, donde se observa que la distribución es aproximadamente de Poisson.

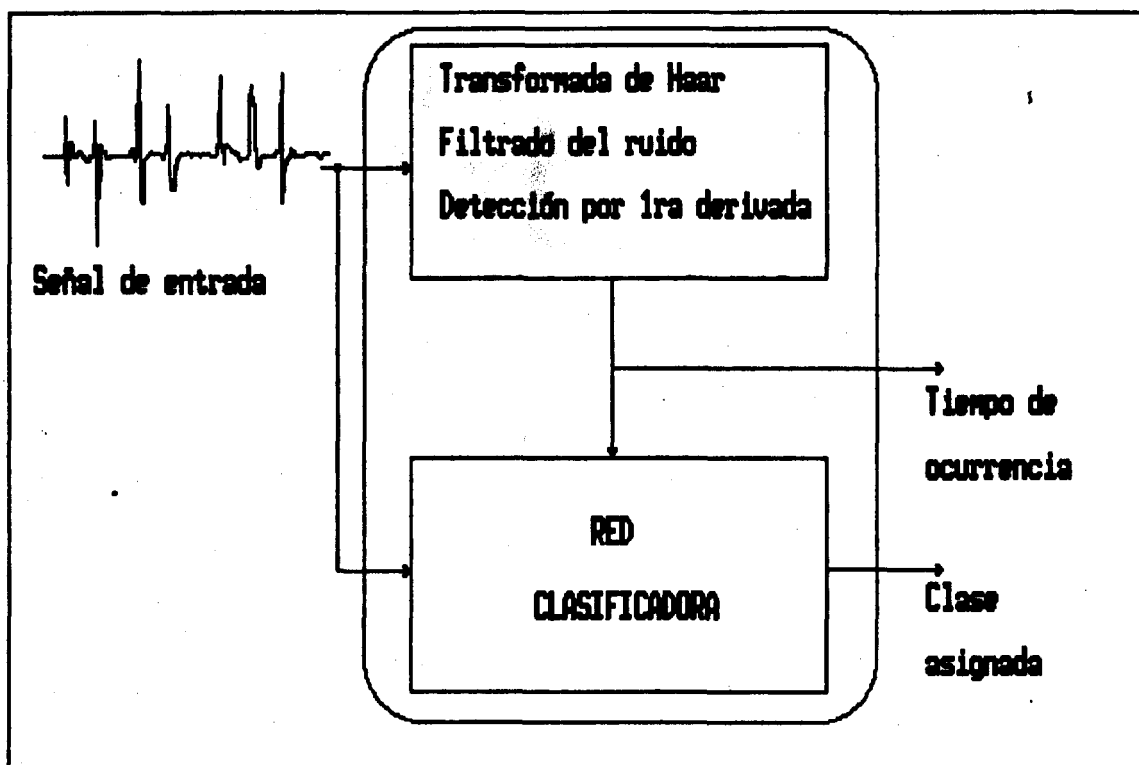


Fig. 3.11. Diagrama esquemático del sistema de clasificación y detección.

DETECCION Y CLASIFICACION DE IMPULSOS

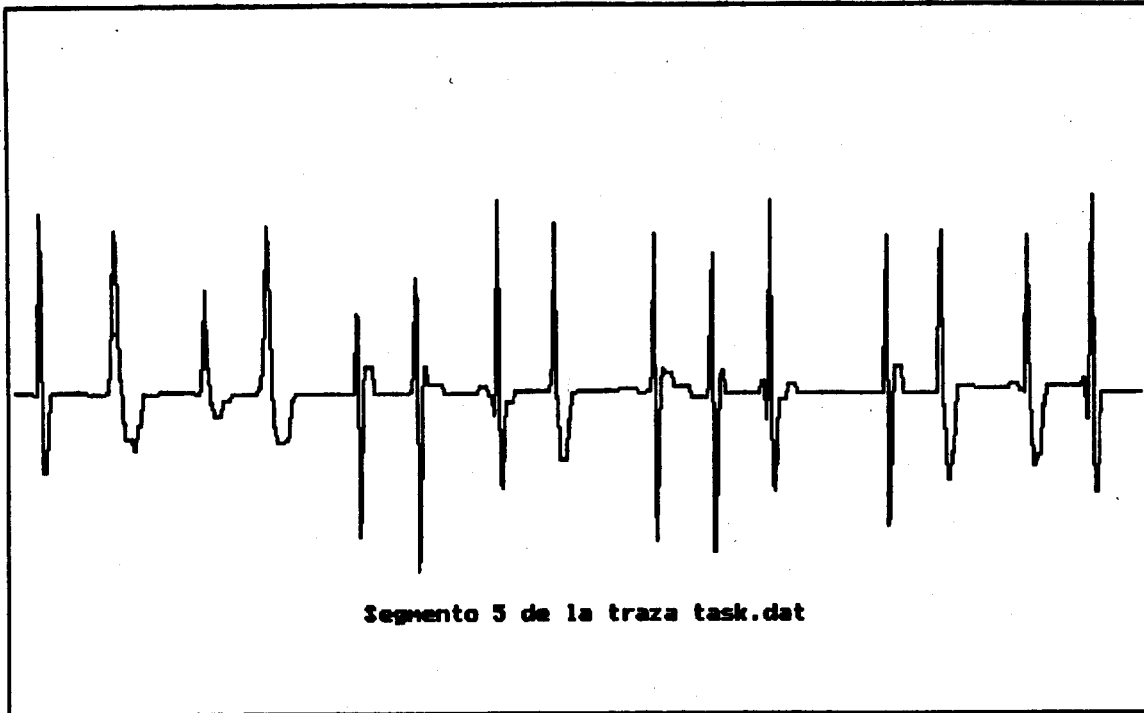


Fig. 3.12. Fragmento de la traza analizada para evaluar el sistema de detección y clasificación.

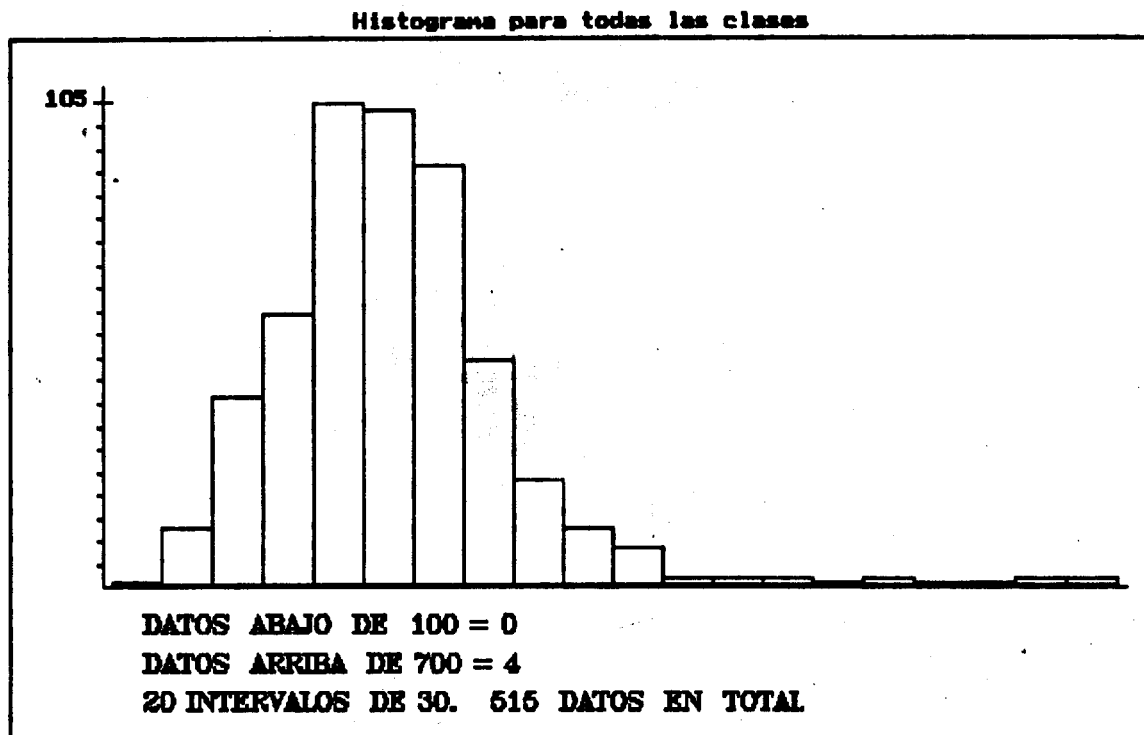


Fig. 3.13. Histograma de tiempos entre impulsos para la traza completa.

**RESULTADOS DEL PROCESAMIENTO DEL ARCHIVO task.dat.
516 pulsos fueron detectados.
Se clasificaron con la red e36.**

Pulso[1] del tipo 5 en 83.
Pulso[2] del tipo 1 en 335.
Pulso[3] del tipo 2 en 719.
Pulso[4] del tipo 4 en 979.
Pulso[5] del tipo 1 en 1295.
Pulso[6] del tipo 2 en 1487.
Pulso[7] del tipo 3 en 1763.
Pulso[8] del tipo 2 en 2051.
Pulso[9] del tipo 1 en 2331.
Pulso[10] del tipo 1 en 2565.
Pulso[11] del tipo 2 en 2799.
Pulso[12] del tipo 3 en 3035.
Pulso[13] del tipo 4 en 3339.
Pulso[14] del tipo 4 en 3679.
Pulso[15] del tipo 1 en 4060.
Pulso[16] del tipo 1 en 4356.
Pulso[17] del tipo 3 en 4560.
Pulso[18] del tipo 5 en 4872.
Pulso[19] del tipo 4 en 5032.
Pulso[20] del tipo 2 en 5304.
Oprima alguna tecla para continuar, <ESC> para cancelar.

Fig. 3.14. Listado parcial de los resultados obtenidos al analizar una traza simulada. Para cada impulso se indica la clase asignada y el lugar de ocurrencia dentro de la traza. La red empleada para la clasificación es una del tipo 5 de la tabla 3.2 (pág. 64), es decir, 17 grupos de dos nodos cada uno en la capa intermedia, con un barrido de 32 muestras y un traslape de 26.

En la figura 3.14 aparece parte del listado obtenido al finalizar el estudio de la traza artificial descrita arriba; para cada impulso detectado se indica su clase y lugar de ocurrencia dentro de la traza. En el ejemplo mostrado, el lugar de ocurrencia se indica con el número de la muestra del archivo que contiene la señal, aunque, por supuesto, sería posible señalar el tiempo a partir del inicio de la traza, si se conoce la frecuencia de muestreo. Los resultados pueden almacenarse en un archivo, con un formato particular, para ser analizados luego con otros programas (Avalos et al., 1990).

CAPITULO 4

DISCUSION Y CONCLUSIONES

Aunque la aparición de nuevos modelos teóricos y tecnologías surge, la mayoría de las veces, por la necesidad de explicar y resolver problemas no resueltos o que no lo han sido de manera satisfactoria por los métodos existentes, uno de los principales argumentos esgrimidos para demostrar su utilidad es el que se puedan utilizar para la solución de problemas ya resueltos con los métodos tradicionales, con un éxito al menos equivalente. Ateniéndonos a este criterio, si se comparan los porcentajes de clasificaciones correctas de señales ruidosas con los datos publicados para otros métodos, los resultados parecen ser alentadores, por ejemplo, para obtener un porcentaje del 90% de clasificaciones correctas con un clasificador que use componentes principales, el porcentaje de ruido de la señal no debe ser superior al 73% , o al 52% si se trata de un clasificador que compare las plantillas de las diferentes clases (Wheeler y Heetderks, 1982). Si el costo de incurrir en errores se considera elevado, una opción es la de tener un grupo de redes, cada una entrenada con la misma base de datos pero a partir de diferentes condiciones iniciales, y asignar la clasificación a partir de una especie de consenso entre las soluciones halladas por las diferentes redes.

Que el resultado obtenido sea satisfactorio no es difícil de explicar si se considera más detenidamente lo que ocurre al aplicar el algoritmo de retropropagación. Fundamentalmente, se está extrayendo información de la base de datos con el propósito de clasificar futuros nuevos patrones, es decir, generalizar por

medio de una representación estadística de los patrones del conjunto de entrenamiento. Al igual que en el caso de los clasificadores estadísticos tradicionales, el conjunto de entrenamiento está formado por un número finito de muestras en un espacio d-dimensional, donde la asignación de clases se ha hecho de una manera no probabilística. Al entrenar la red, minimizando el error cuadrático, se obtiene el valor de los pesos que permite que los nodos de la capa de salida asignen la clase correspondiente al patrón de entrada. Los pesos de la red son así una función estadística de los patrones de entrenamiento. Al contrario que en muchos métodos tradicionales, las estadísticas de los datos analizados no necesitan calcularse previamente, sino que son computadas por la misma red. Así, la extracción de características es efectuada de tal manera que se corresponde de manera óptima con el esquema de clasificación.

Pero esto no es otra cosa que encontrar, de manera aproximada, la probabilidad a posteriori de las clases. En muchos problemas, la distribución de patrones, en el conjunto de entrenamiento, entre las diferentes clases no es uniforme y, además, las probabilidades de las diferentes categorías se agrupan en hipersuperficies. Sin embargo, los nodos de la red pasan un producto escalar de señales de entrada por pesos a su función de activación, con lo cual las hipersuperficies de salida deben encontrarse a partir de hiperplanos en la entrada, por lo que podría ser necesario emplear combinaciones de orden superior de la señal de entrada (Shoemaker, 1991). Por otra parte, si las matrices de covariancia de las clases son idénticas, las hipersuperficies de igual probabilidad se convierten en hiperplanos y los nodos con entradas lineales y función de activación sigmoideal son suficientes para llevar a cabo la tarea de discriminación. Por esta razón, al utilizar nodos "normales" se está asumiendo implícitamente que se calcula, a partir del conjunto de entrenamiento, una matriz de covariancia promediada que caracteriza a cada clase por separado, es decir, puede considerarse el entrenamiento mediante el algoritmo de retro-

propagación como una generalización multidimensional no lineal de la función discriminante de Fisher (Lowe y Webb, 1991).

Usualmente se dice que los elementos procesadores de las capas intermedias funcionan como extractores de características. Sin embargo, cuando se aprecian sus valores de activación, es notorio que la estrategia de codificación no responde a una sola característica de la señal analizada sino que más bien es el resultado de la interacción de varias de éstas. La codificación interna que obtiene la red para los diferentes patrones no puede separarse de manera simple en un conjunto de particularidades determinadas y, aunque se supone que es un resultado de la estadística de los patrones de entrenamiento (White, 1989), hasta la fecha no existe un formalismo adecuado para interpretarla.

Se mencionó ya que en el conjunto de entrenamiento se incluyeron patrones desplazados temporalmente entre sí. La necesidad de hacer esto surge del hecho de que aunque el diseño de las conexiones entre la capa de entrada y la siguiente capa intermedia, así como aquél entre las capas intermedias, es invariante a la traslación de la señal de entrada, la capa de salida no lo es, puesto que funciona como un selector de uno entre N . Es debido a esto que la red, como un todo, sólo sea parcialmente invariante a la traslación, lo que conduce a pensar en procesar previamente la señal con alguna operación que sea invariante a la traslación, como ocurre, por ejemplo, con la componente en amplitud de la transformada de Fourier, debido a la ausencia de la información de fase, o, lo que es equivalente en el dominio temporal, la autocorrelación de la señal, ya que esta última es independiente de la posición absoluta de la señal (Glüber, 1988). Sin embargo, esto requiere que la señal se encuentre en un marco de referencia respecto al cual se produzca la traslación, lo cual no ocurre en el caso estudiado, pues aquí, cuando la señal se traslada sobre la capa de entrada sus componentes no son los mismos que los de la señal "original", ya que el número de elementos de la capa de entrada es fijo, en otras palabras, el arreglo de datos al que se le aplicaría la transformación requerida no es semejante en ambos casos. Otra

alternativa para obtener invariancia respecto a la traslación consiste en expandir los componentes de la señal de entrada con términos de segundo o más alto orden, pero en este caso la respuesta de la red se torna extremadamente sensible a la presencia de ruido en el patrón de entrada y a los cambios en su morfología (Pao, 1989; Lee, 1990).

Por otra parte, la reducción de tiempo conseguida por el uso de una red clasificadora más simple tal vez no sea una compensación suficiente por aquel empleado en el trabajo adicional de calcular algún tipo de transformación de la señal de entrada. La clase de normalización empleada, que consiste en ajustar la escala de la señal dependiendo de los valores máximo y mínimo esperados en el registro, permite compensar pequeños movimientos en el microelectrodo de registro, pues cuando estos ocurren la amplitud de la señal varía, pero no así su forma (Abeles, 1982).

Conviene anotar aquí que el desempeño de la red depende del valor del error promedio mínimo alcanzado al final de la sesión de entrenamiento. Al inicio del aprendizaje, el porcentaje de clasificaciones incorrectas, con patrones de un conjunto de prueba, disminuye al decrementarse el error promedio pero, a partir de cierto valor, el número de clasificaciones incorrectas empieza a aumentar. En otras palabras, el ajuste de la red a los datos del conjunto de entrenamiento se hace "demasiado bueno para ser verdad". Este sobre-entrenamiento disminuye la capacidad de generalización, por lo que es preciso evaluar el desempeño a intervalos regulares durante el entrenamiento y darlo por concluido cuando empiece a disminuir. Para las diferentes configuraciones usadas se encontró que el desempeño en la clasificación es óptimo cuando el error promedio total es de 0.01 y empieza a deteriorarse si se pretende disminuir más su valor. Al parecer, el valor crítico al que debe reducirse el error no depende del número de elementos de las capas intermedias sino más bien de la cantidad de patrones en el conjunto de entrenamiento, aunque es necesario investigar más detalladamente este comportamiento.

El uso de un conjunto de entrenamiento donde cada muestra está etiquetada con su clase, revela una de las limitaciones del método empleado. Es preciso conocer de antemano cuantas clases se encontrarán en un registro dado y tener algunos ejemplares característicos para cada una de ellas, lo que implica una labor previa antes de poder entrenar la red. Puesto que cuanto más ejemplares se tengan en el conjunto de entrenamiento, mejor será la estimación de los pesos para definir la pertenencia de clases, el número total de muestras requerido puede ser significativo, dependiendo de la cantidad de clases halladas. Además, si la ocurrencia de patrones de algunas clases es inferior a la de otras, será necesario simular algunas muestras. En el problema estudiado, se simularon otros impulsos agregando ruido de diferente magnitud a las muestras originales; además, en el conjunto de entrenamiento se representaron por igual todas las clases, lo que no puede ser cierto en general. Sin embargo, esta última consideración permite que la contribución de las características de cada clase a la optimización del valor de los pesos sea distribuida equitativamente durante el aprendizaje.

Si bien el requerimiento de un periodo de entrenamiento antes de poder usar la red como un clasificador es una restricción para su empleo en tiempo real, cuando se registra una zona de la que ya se conocen las posibles formas de onda de sus potenciales de acción, como ocurre con la corteza auditiva del gato, no es necesario estar reentrenando la red. Claro está que el entrenamiento inicial deberá realizarse para todas las clases esperadas de espigas nerviosas.

Una situación que no pudo ser resuelta con el enfoque presente es la de clasificar correctamente impulsos que se traslapan. Este es un problema crítico, que sólo ha podido ser tratado parcialmente mediante el uso de filtros óptimos, que requieren el empleo de un electrodo por cada neurona registrada y bajo nivel de ruido (27% de ruido para un 90% de clasificaciones correctas) (Wheeler y Heetderks, 1982). Para poder solucionar esta eventualidad con una red neuronal del tipo usado, es necesario representar a la señal de entrada mediante algunas de

sus características, de tal forma que estas sean recuperables aún en el caso de sobreposiciones.

No se ha hecho mucho énfasis en la labor de detección de los impulsos, puesto que se espera utilizar la red clasificadora como parte de un sistema más complejo de adquisición y procesamiento de espigas nerviosas, en el que la detección se efectúe con circuitos especiales, por ejemplo, con un procesador digital de señales (Ayhlón y Espinosa, 1990). Si se tiene en cuenta que, para el método aquí descrito, la tarea de detección de los impulsos toma unas dos terceras partes del tiempo total de computación, el ahorro de tiempo obtenido al utilizar un DSP para la detección es apreciable.

A pesar de que la aplicación particular investigada adolece de limitaciones que pueden restringir su uso en tiempo real, es un ejemplo claro de la factibilidad del empleo de las redes neuronales artificiales en la solución de muchos problemas, incluyendo los del campo de la ingeniería biomédica. Aun cuando mucho está todavía por realizarse, su aplicación es ya una realidad concreta en áreas como el procesamiento de imágenes, análisis de ECG, EEG y EMG, compresión de datos en monitores digitales Holter, análisis del estado de pacientes en salas de cirugía y de cuidado intensivo, ayudas que permiten a los parapléjicos el manejo de equipo por medio de la identificación y reconocimiento de voz, conversión de texto escrito a voz y lenguaje Braille, control de robots autónomos para asistir a minusválidos y de prótesis activas, ayudas auditivas que no sólo amplifiquen selectivamente el sonido sino que cambien sus componentes de frecuencia, integración de la información de sensores táctiles, auxiliares en el diagnóstico y pronóstico de enfermedades. (Una revisión reciente de diversas aplicaciones puede encontrarse en el número 3, vol. 9, del Engineering in Medicine and Biology Magazine de la IEEE, septiembre de 1990.)

Si bien una de las mayores restricciones en el empleo de las redes neuronales es el tiempo requerido para su entrenamiento, cuando este no es un factor crítico se pueden usar con equipo

económico, por ejemplo, computadoras personales, en la resolución de problemas complejos que de otra manera precisarían de equipo más costoso. Por otro lado, la disponibilidad creciente de equipo de cómputo más rápido y la posibilidad de instrumentar las redes neuronales en circuitos especiales, hacen que la consideración del tiempo de entrenamiento sea cada vez menos acuciosa. La última alternativa representa mejoras realmente dramáticas en los tiempos de entrenamiento y ejecución. Aunque simple en concepto, no es muy fácil de realizar pues requiere el control de gran cantidad de conexiones resistivas variables. Sin embargo, la manera de resolver estos obstáculos parece estar en la aplicación de conceptos neuromórficos, que involucran el uso de circuitos VLSI análogos para efectuar las operaciones matemáticas mientras que el control y la comunicación de datos se hace en forma digital (Mead, 1989).

Por supuesto, la redes neuronales no son una panacea universal y su uso debe integrarse y complementarse con otras tecnologías ya bien establecidas, como el procesamiento digital de señales, el control óptimo adaptivo y la inteligencia artificial, así como con otras que están emergiendo, por ejemplo, la lógica difusa y los algoritmos genéticos. Aun cuando serán precisos algunos años antes de que su empleo sea un procedimiento común, debemos ser conscientes de que no son una posibilidad fuera de nuestro alcance sino una herramienta que ya está disponible. Parafraseando el programa que Goethe, en una de sus conversaciones con Eckermann, proponía para la literatura, nos atrevemos a decir que si una tecnología puede emplearse en hacer más digna la vida de las personas, es nuestro deber encontrar la manera de incorporar el conocimiento de diferentes disciplinas con las ventajas que ofrece dicha tecnología en la solución de las necesidades existentes y la creación de nuevas oportunidades. Que esa posibilidad sea nuestro ideal.

REFERENCIAS

- Abeles, M. (1982): *Local Cortical Circuits*. Springer-Verlag. Nueva York.
- Abeles, M. y M.H. Goldstein (1977): Multispikes train analysis. *Proc. IEEE* 65: 762-773.
- Ahmed, N y K.R. Rao (1975): *Orthogonal Transforms for Digital Signal Processing*. Springer-Verlag. Nueva York.
- Almeida, L.B. (1989): Backpropagation in non-feedforward networks. En *Neural Computing Architectures*, págs. 74-91, editado por I. Aleksander. MIT Press. Cambridge, Mass.
- Amari, Sh. (1977): Neural theory of association and concept-formation. *Biol. Cybern.* 26: 175-185.
- Amari, Sh. (1988): Mathematical theory of self-organization in neural nets. En *Organization of Neural Networks*, págs. 399-413, editado por W. von Seelen, G. Shaw y U.M. Leinhos. VCH. Weinheim, Alemania.
- Andrews, H.C. y K.L. Caspari (1970): A generalized technique for spectral analysis. *IEEE Trans. Computers* 19: 16-25.
- Avalos, M., G. Leonetti, M. Alcántara e I. Espinosa (1990): Estudio comparativo de métodos para análisis de conectividad neuronal. *Rev. Mex. Ing. Bioméd.* 11: 109-124.
- Ayhlón, A. e I. Espinosa (1990): Resultados preliminares para un sistema de detección de señales neuroeléctricas. *Rev. Mex. Ing. Bioméd.* 11: 169-182.
- Barto, A.G. y P. Anandan (1985): Pattern recognizing stochastic learning automata. *IEEE Trans. Syst. Man Cybern.* 15: 360-375.
- Baum, E.B. y D. Haussler (1989): What size net gives valid generalization? *Neural Computation* 1: 151-160.
- Bottou, L., F. Fogelman Soulié, P. Blanchet y J. Liénard (1990): Speaker-independent isolated digit recognition: Multilayer perceptron vs. dynamic time warping. *Neural Networks* 3: 453-465.
- Bryson, A.E. y Y.-Ch. Ho (1969): *Applied Optimal Control*. Blaisdell. Waltham, Massachusetts.

- Cadzow, J.A. (1990): Signal processing via least squares error modeling. *IEEE ASSP Magazine* 7 (4): 12-31.
- Camp, C. y H. Pinsky (1979): Computer separation of unitary spikes from whole-nerve recordings. *Brain Res.* 169: 455-479.
- Chu, P.L. (1989): Fast gaussian noise generator. *IEEE Trans. Acoust., Speech, Signal Processing* 37: 1593-1597.
- D'Hollander, E.H. y A. Orban (1979): Spike recognition and on-line classification by unsupervised learning system. *IEEE Trans. Biom. Eng.* 26: 279-284.
- Denker, J.S. (1986): Neural network models of learning and adaptation. *Physica D* 22: 216-232.
- Dimopoulos, N. (1989): A study of the asymptotic behavior of neural networks. *IEEE Trans. Circuit Syst.* 36: 687-694.
- Dinning, G.J. y A. Sanderson (1981): Real-time classification of multiunit neural signals using reduced feature sets. *IEEE Trans. Biom. Eng.* 28: 804-811.
- Duda, R. y P. Hart (1973): *Pattern Classification and Scene Analysis*. John Wiley. Nueva York.
- Fahlman, S.E. (1989): Faster-learning variations of back-propagation. En *Proceedings of the 1988 Connectionist Models Summer School*, págs. 38-51, editado por D. Touretzky, G. Hinton y T. Sejnowski. Morgan Kaufmann. San Mateo, California. 1989.
- Felten, E.W., O. Martin, S.W. Otto y J. Hutchinson (1990): Multi-scale training of a large backpropagation net. *Biol. Cybern.* 62: 503-509.
- Fraden, J. y M.R. Neuman (1980): QRS wave detection. *Med. & Biol. Eng. & Comput.* 18: 125-132.
- Fukunaga, K. y R.R. Hayes (1989): Estimation of classifier performance. *IEEE Patt. Anal. Machine Intell.* 11: 1087-1101.
- Gerstein, G.L., M.J. Bloom, I. Espinosa, S. Evanczuck y M. Turner (1983): Design of a laboratory for multiunit studies. *IEEE Trans. Sys. Man Cybern.* 13: 668-676.
- Geszti, T. (1990): *Physical Models of Neural Networks*. World Scientific Press. Singapur.
- Glaser, E.M. y D.S. Ruchkin (1976): *Principles of Neurobiological Signal Analysis*. Academic Press. Nueva York.

- Glüber, H. (1988): Relational processing: A concept for the explanation of invariant visual pattern recognition. En *Processing Structures for Perception and Action*, págs. 39-77, editado por H. Marko. VCH. Weinheim, Alemania.
- Gorman, R.P. y T.J. Sejnowski (1988): Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks* 1: 75-90.
- Griffith, J.S. (1971): *Mathematical Neurobiology*. Academic Press. Londres.
- Grossberg, S. (1972): Neural expectation: Cerebellar and retinal analogs of cells fired by learnable or unlearned pattern classes. *Kybernetics* 10: 49-57. Republicado en *Studies of Mind and Brain*, págs. 296-331, editado por S. Grossberg. Reidel. Boston. 1982.
- Grossberg, S. (1974): Classical and instrumental learning by neural networks. *Prog. Theor. Biol.* 3: 51-141. Republicado en *Studies of Mind and Brain*, págs. 65-156, S. Grossberg (Editor). Reidel. Boston. 1982.
- Grossberg, S. (1980): How does the brain build a cognitive code? *Psychol. Rev.* 87: 1-51. Republicado en *Studies of Mind and Brain*, págs. 2-52, editado por S. Grossberg. Reidel. Boston. 1982.
- Grossberg, S. (1987): *The Adaptive Brain*. 2 vols. North-Holland. Amsterdam, Holanda.
- Guez, A., M. Kam y J. Eilbert (1989): Computational-complexity reduction for neural network algorithms. *IEEE Trans. Syst. Man Cybern.* 19: 409-413.
- Hampson, S.E. y D.J. Volper (1989): Feature handling in learning algorithms. En *Dynamic Interactions in Neural Networks: Models and Data*, págs. 87-106, editado por M. Arbib y Sh. Amari. Springer-Verlag. Nueva York.
- Hebb, D.O. (1949): *The Organization of Behavior*. John Wiley. Nueva York.
- Hecht-Nielsen, R. (1987): Neurocomputer applications. En *Neural Computers*, págs. 445-453, editado por R. Eckmiller y C. von der Malsburg. Springer-Verlag. Berlín.
- Hopfield, J.J. (1982): Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* 47: 2554-2558.
- Huang, S.-C y Y.-F. Huang (1991): Bounds on the number of hidden neurons in multilayer perceptrons. *IEEE Trans. Neural Networks* 2: 47-55.

- Jacobs, R. (1988): Increased rates of convergence through learning rate adaptation. *Neural Networks* 1: 295-308.
- Kandel, E.R. y J.H. Schwartz (1985): *Principles of Neural Science*. 2da. ed. Capítulo 10. Elsevier. Nueva York.
- Kohonen, T. (1987): *Content-addressable Memories*. 2da. edición. Springer-Verlag. Nueva York.
- Kohonen, T. (1988): *Self-organization and Associative Memory*. 2da. ed. Springer-Verlag. Nueva York.
- Kohonen, T. (1988a): An introduction to neural computing. *Neural Networks* 1: 3-16.
- Kosko, B. (1988): Bidirectional associative memories. *IEEE Trans. Syst. Man Cybern.* 18: 49-60.
- LeCun, Y. (1988): A theoretical framework for back-propagation. En *Proc. of the 1988 Connectionist Models Summer School*, págs. 21-28, editado por D. Touretzky, G. Hinton y T. Sejnowski. Morgan Kaufmann. San Mateo, California.
- Lee, S.C. (1990): Using a translation-invariant neural network to diagnose heart arrhythmia. En *Advances in Neuronal Information Processing 2*, págs. 240-247, editado por D.S. Touretzky. Morgan Kaufmann. San Mateo, California.
- Lippmann, R.P. (1987): An introduction to computing with neural nets. *IEEE ASSP Magazine* 4 (2): 4-22.
- Lisman, J. (1989): A mechanism for the Hebb and anti-Hebb processes underlying learning and memory. *Proc. Natl. Acad. Sci. USA* 86: 9574-9578.
- Lowe, D. y A.R. Webb (1991): Optimized feature extraction and the Bayes decision in feed-forward classifier networks. *IEEE Trans. Pattern Anal. Machine Intell.* 13: 355-364
- Maren, A.J., D. Jones y S. Franklin (1990): Configuring and optimizing the back-propagation network. En *Handbook of Neural Computing Applications*, págs. 233-250, editado por A. Maren, C. Harston y R. Pap. Academic Press. San Diego, California.
- McGillivray, R. y R. Wald (1984): Hybrid analog and digital system for the analysis of action potentials. *J. Clin. Eng.* 9: 333-340.
- Mead, C. (1989): *Analog VLSI and Neural Systems*. Addison-Wesley. Reading, Massachusetts.

- Minsky, M. y S. Papert (1969): *Perceptrons: An Introduction to Computational Geometry*. MIT Press. Cambridge, Mass. Edición aumentada publicada en 1988.
- Mirchandani, G. y W. Cao (1989): On hidden nodes for neural nets. *IEEE Trans. Circuits and Systems* 36: 661-664.
- Mitchinson, G.J. y R.M. Durbin (1989): Bounds on the learning capacity of some multi-layer networks. *Biological Cybernetics* 60: 345-356.
- Nilsson, N.J. (1965): *Learning Machines*. McGraw-Hill. New York. Republicado en 1990 por Morgan Kaufmann, San Mateo, Cal.
- O'Connell, R.J., W.A. Koesis y R.L. Schoenfeld (1973): Minicomputer identification and time of nerve impulses mixed in a single recording channel. *Proc. IEEE* 61: 1615-1621.
- Ohkita, M., Y. Kobayashi y M. Inoue (1983): Application of the Haar functions to solution of differential equations. *Math. and Computers in Simulation XXV*: 31-38.
- Pao, Y-H. (1989): *Adaptive Pattern Recognition and Neural Networks*. Addison-Wesley. Reading, Massachusetts.
- Pearlmutter, B.A. (1989): Learning state space trajectories in recurrent neural networks. *Neural Computation* 1: 263-269.
- Pierre, D.A. (1986): *Optimization Theory with Applications*. Dover Publ. Inc. Nueva York.
- Pineda, F.J. (1987): Generalization of back-propagation to recurrent neural networks. *Phys. Rev. Lett.* 59: 2229-2232.
- Pineda, F.J. (1989): Recurrent backpropagation and the dynamical approach to adaptive neural computation. *Neural Computation* 1: 161-172.
- Pitts, W. y W.S. McCulloch (1947): How we know universals: The perception of auditory and visual forms. *Bull. Math. Biophysics* 9: 127-147. Republicado en *Neurocomputing: Foundations of Research*, págs. 32-41, editado por J.A. Anderson y E. Rosenfeld. MIT Press. Cambridge, Mass. 1988.
- Reilly, D. y L. Cooper (1990): An overview of neural networks. En *An Introduction to Neural and Electronic Networks*, págs. 227-248, editado por C. Lau, J. Davis y S. Zornetzer. Academic Press. San Diego, California.
- Resnikoff, H.L. (1989): *The Illusion of Reality*. Springer-Verlag. Nueva York.

- Rosenblatt, R. (1962): *Principles of Neurodynamics*. Spartan Books. Nueva York.
- Rosenfield, I. (1988): *The Invention of Memory*. Basic Books. Nueva York.
- Rumelhart, D., G. Hinton y J. Williams (1986): Learning internal representations by error propagation. En *Parallel Distributed Processing*, capítulo 8, vol. 1., editado por D. Rumelhart y J. McClelland. MIT Press. Cambridge, Mass.
- Schmidt, E.M. (1984): Instruments for sorting neuroelectric data: A review. *J. Neuroscience Methods* 12: 1-24.
- Schmidt, E.M. (1984a): Computer separation of multiunit neuroelectric data: A review. *J. Neuros. Meth.* 12: 95-111.
- Serra, R. y G. Zanarini (1990): *Complex Systems and Cognitive Processes*. Springer-Verlag. Berlín.
- Shoemaker, P.A. (1991): A note on least-squares learning procedures and classification by neural networks models. *IEEE Trans. Neural Networks* 2: 158-160.
- Shvartsman, V., G. Barnes, L. Shvartsman y N. Flowers (1982): Multichannel signal processing based on logic averaging. *IEEE Trans. Biom. Eng.* 29: 531-536.
- Singer, W. (1988): Self-organization in biological systems. En *Processing Structures for Perception and Action*, págs. 1-18, editado por H. Marko. VCH. Weinheim, Alemania.
- Smith, A. (1986): *La mente*. Vol. II, página 415. Salvat Editores. Barcelona, España.
- Sontag, E. (1989): Sigmoids distinguish more efficiently than heavisides. *Neural Computation* 1: 470-472.
- Valiant, L.G. (1984): A theory of the learnable. *Comm. Ass. Comput. Mach.* 27: 1134-1142.
- Vibert, J-F., J.N. Albert y J. Costa (1987): Intelligent software for spike separation in multiunit recordings. *Med. & Biol. Eng. & Comput.* 25: 366-372.
- Waibel, A., T. Hanazawa, G. Hinton, K. Shikano y K. Lang (1989): Phoneme recognition using time-delay neural networks. *IEEE Trans. Acoust., Speech, Signal Processing* 37: 328-339.
- Wan, E.A. (1990): Neural network classification: A Bayesian interpretation. *IEEE Trans. Neural Networks* 1: 303-305.
- Watanabe, S. (1984): *Pattern Recognition: Human and Mechanical*. John Wiley. Nueva York.

- Waugh, F.R., C.M. Marcus y R.M. Westervelt (1990): Fixed-point attractors in analog neural computation. *Physical Review Letters* 64: 1986-1989.
- Weiss, Sh.M. y C.A. Kulikowski (1991): *Computer Systems that Learn*. Morgan Kaufmann. San Mateo, California.
- Werbos, P. (1974): *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Disertación doctoral. Harvard University, Cambridge, Mass.
- Wheeler, B.C. y W. Heetderks (1982): A comparison of techniques for classification of multiple neural signals. *IEEE Trans. Biom. Eng.* 29: 752-759.
- White, H. (1989): Learning in artificial neural networks: A statistical perspective. *Neural Computation* 1: 425-464.
- Widrow, B. y M.E. Hoff (1960): Adaptive switching circuits. 1960 *IRE Wescon Convention Record*, págs. 96-104, IRE, Nueva York. Republicado en *Neurocomputing: Foundations of Research*, págs. 32-41, editado por J.A. Anderson y E. Rosenfeld. MIT Press. Cambridge, Mass. 1988.
- Widrow, B., J.M. McCool, M.G. Larimore y C.R. Johnson (1976): Stationary and Nonstationary learning characteristics of the LMS adaptive filter. *Proc. IEEE* 64: 1151-1162.
- Wittgenstein, L. (1953): *Philosophical Investigations*. Basil, Blackwell and Mott. Oxford.
- Yang, X. y Sh. Shamma (1988): A totally automated system for the detection and classification of neural spikes. *IEEE Trans. Biom. Eng.* 35: 806-816.

APENDICE

PROGRAMAS DESARROLLADOS

El software desarrollado contiene los siguientes archivos:

```
Path: A:\AUXILIAR
Files:      AZAR      .C
            CES       .H
            MENU     .C

Path: A:\CLASIFIC
Files:      CLASIF    .C
            CLASIF    .EXE
            E36_V     .DAT
            E36_W     .DAT
            TRAZA     .DAT

Path: A:\ENTRENA
Files:      PAR3      .DAT
            PAR3_V    .DAT
            PAR3_W    .DAT
            PRUEBA    .DAT
            RETROP    .C
            RETROP    .EXE
```

Los programas ejecutables están escritos en Turbo C, versión 2. El programa para entrenar una red con el algoritmo de retropropagación (RETROP) está en el subdirectorio ENTRENA, mientras que el usado para detectar y clasificar los impulsos en un registro nervioso (CLASIF) aparece en el subdirectorio CLASIFIC. Los programas adicionales necesarios para compilar los archivos fuente, se encuentran en el subdirectorio AUXILIAR: CES.H contiene rutinas para la entrada y salida de datos y debe estar en el subdirectorio INCLUDE del compilador; MENU.C maneja menús y AZAR.C es un generador de números aleatorios. Los dos primeros se requieren tanto para RETROP.C como para CLASIF.C, mientras que el último sólo es indispensable para RETROP.C. A continuación se describe el uso de los programas.

A. PROGRAMA PARA ENTRENAR LA RED

Los programas empleados aparecen en el subdirectorío ENTRENA. El programa principal, RETROP.C, permite construir redes con una arquitectura convencional (figura 1.7) o con la utilizada en el presente trabajo (figura 3.2). El programa fue originalmente escrito para ejecutarse en una computadora Stardent Titan, con sistema operativo Unix, en la que la gran disponibilidad de memoria no impone muchas restricciones al tamaño de la red ni del conjunto de entrenamiento. La versión para computadora personal que se incluye aquí no optimiza el manejo de la memoria y permite el uso de redes con un máximo de 128 nodos en la capa de entrada, 10 en la de salida, 3 capas intermedias y un conjunto de entrenamiento con 70 patrones distintos, cada uno de hasta 138 elementos (128 datos para la entrada y 10 para la salida).

El programa se divide en tres módulos principales:

1. Inicio del entrenamiento.
2. Continuación del aprendizaje.
3. Reconocimiento de patrones.

que se describirán a continuación.

Para ilustrar el uso del programa se incluye el archivo par3.dat, que se usará como conjunto de entrenamiento para una red con 3 nodos en la capa de entrada, 1 capa intermedia con tres nodos y un nodo de salida, con una arquitectura convencional. Si los patrones son binarios, cuando el número de unos a la entrada de la red es par la respuesta de la red debe ser un uno y cero cuando el número de unos sea impar.

1. **Inicio del entrenamiento.** En la primera fase se especifica el conjunto de entrenamiento, la arquitectura de la red y, opcionalmente, la semilla del generador de números aleatorios. Ampliando un poco, se realizan los siguientes pasos:

i) El nombre de la tarea por ejecutar es el mismo que el del archivo con el conjunto de entrenamiento (par3 para el ejemplo). El archivo debe estar escrito en modo ascii, con un

patrón por renglón (un patrón incluye los datos para la capa de entrada, primero, y, luego, las respuestas esperadas en la capa de salida), con datos enteros o reales y su sufijo debe ser .dat.

ii) Para la arquitectura de la red se especifica el número de elementos para la capa de entrada (3 en el ejemplo), de salida (uno en el ejemplo) y el número de capas intermedias que se usarán (uno para par3). Si se quiere emplear una red como la de la figura 3.2, deberá indicarse el número de nodos por grupo, el barrido de cada grupo (cuántos nodos de la capa inferior se abarcan) y el traslape entre las medidas de los diferentes grupos. Puesto que no todas las combinaciones son posibles, es conveniente calcularlas de antemano; el programa no aceptará aquellas incorrectas. Si se desea utilizar una arquitectura convencional (véase la figura 1.7), el número de nodos por grupo en la capa intermedia será igual al número de nodos deseado para esa capa (3 para nuestro ejemplo) y el barrido igual al número de nodos de la capa precedente (también 3 para el ejemplo). Por supuesto, en esta última opción no se inquirirá por el traslape, ya que hay sólo un grupo de nodos en la capa intermedia.

iii) Se puede cambiar la semilla del generador de números aleatorios, con el propósito de variar los valores iniciales de los pesos al comenzar diferentes entrenamientos.

En la segunda fase se indica el valor de las variables que afectan la ejecución y resultados del programa, específicamente:

i) Velocidad de aprendizaje y variación del momentum.

ii) Condiciones para terminar el entrenamiento. Por una parte, si el error promedio total máximo tolerado es menor que el error obtenido por la red, se prosigue con el entrenamiento, mientras que por otro lado se compara el número de presentaciones del conjunto de entrenamiento con el número máximo de iteraciones especificado para determinar el fin del entrenamiento. La primera de estas dos condiciones en cumplirse determina la terminación del aprendizaje. Al finalizar, se generan dos archivos en modo ascii, uno con la arquitectura de la red y las variables empleadas y que tiene por sufijo _v.dat, y otro con el valor de

los pesos encontrados, con el sufijo `_w.dat`. Para la tarea `par3`, los archivos obtenidos al final son `par3_v.dat` y `par3_w.dat`.

iii) El error individual máximo se emplea para no calcular el cambio de los pesos cuando un determinado patrón produce un error en la salida menor que el indicado, es decir, cuando no contribuye de manera significativa al error total.

iv) Actualización de los pesos. Puede ser por épocas o por patrones. Si se hace de la última forma, existe la opción de presentar los patrones en el orden en que aparecen en el archivo del conjunto de entrenamiento o en un orden aleatorio.

v) Si se desea estudiar como cambia el error total durante el entrenamiento, se puede abrir un archivo para almacenar el valor del error total. El archivo está escrito en `ascii`, indica el número de la iteración y del error al finalizar esta y se denomina `error.dat`.

vi) Se debe indicar el número de patrones que se leerán del archivo con el conjunto de entrenamiento (8 en el ejemplo).

Con los valores que aparecen por defecto, para la red empleada con `par3`, el número de iteraciones necesario para reducir el error total a 0.01 debe ser de 635.

2. Continuación del aprendizaje. Básicamente, esta sección es igual a la segunda fase de la anterior, sólo que ahora el valor de los pesos, la arquitectura y otras variables se leen de los archivos generados durante una sesión de aprendizaje previa. Su utilidad estriba en que se puede proseguir el proceso de aprendizaje a partir del punto alcanzado en la sesión anterior, sin tener que recomenzar desde el principio. Claro está, ya no es posible modificar la arquitectura de la red.

3. Reconocimiento de patrones. Es posible comprobar el desempeño de la red con un conjunto de prueba. Para tal efecto, el archivo debe contener un patrón por renglón, en este caso el patrón tiene solamente los datos para la capa de entrada. El archivo puede estar escrito en modo `ascii` o binario, con datos tipo entero o real y se presupone que sufijo será `.dat`. Se ha

incluido el archivo prueba.dat, en modo ascii, para ser usado con este propósito. Antes de llevar a cabo el reconocimiento, es necesario indicar los archivos con los datos acerca de la arquitectura y valor de los pesos de la red, para lo cual no es preciso dar los sufijos _v.dat o _w.dat, sino el nombre de la tarea original (sólo par3 cuando se pregunte por la tarea, en el ejemplo usado).

B. PROGRAMA PARA DETECCION Y CLASIFICACION

El subdirectorio CLASIFIC contiene el programa empleado para la detección y clasificación de los impulsos presentes en una traza nerviosa, junto con un archivo con 4100 muestras de un registro simulado, TRAZA.DAT, y los datos de una red clasificadora, e36_v.dat y e36_w.dat, generados por el programa RETROP.C.

El programa CLASIF.C se puede subdividir en las siguientes partes:

1. **Datos de la red de clasificación.** Aquí se especifican los archivos en los que se encuentra la información pertinente a la red que se utilizará, y que han sido obtenidos con el programa RETRO.C. Solo se requiere suministrar el prefijo de los archivos, por ejemplo, e36 indica que se la información se encuentra en e36_v.dat y e36_w.dat, y estos deben estar en el directorio de trabajo.

2. **Archivo con la señal por procesar.** Es el archivo donde se encuentra la traza que se analizará. Debe estar escrito en binario, con datos enteros y su sufijo es .dat. Para el ejemplo suministrado debe indicarse traza. El archivo se leerá en tramos de 4096 datos, aunque es posible cambiar este valor por uno menor que sea potencia entera de dos. También se indicará en número de tramos que se leerán, dependiendo del valor elegido para cada lectura. Si se leen 4096 datos a la vez, el archivo traza se leerá una sola vez.

3. Procesamiento de los datos, con graficación. Una vez que los pasos 1. y 2. se han realizado, se puede proceder al análisis de la señal. El umbral para la remoción del ruido ha sido calculado a partir de los puntos de un tramo sin impulsos, pero puede variarse a voluntad. El umbral para limitar la amplitud debe calcularse con base en la menor amplitud esperada de alguna de las clases de espigas presentes en la traza y su elección es crítica para obtener un alto grado de confianza en la detección. Como ya se había anotado, el lugar de la detección ocurre en un sitio cercano al de la amplitud máxima y puesto que las señales usadas para el entrenamiento incluían datos antes de dicho punto, es conveniente tomar la señal que se le pasa al clasificador a partir de unas cuantas muestras antes. Los valores por defecto mostrados son adecuados para el registro simulado que se incluye. En esta opción, cada traza es dibujada, después de que el ruido ha sido removido mediante la transformada de Haar.

4. Procesamiento de los datos, sin graficación. Esta alternativa es semejante a 3., pero sin graficar los diferentes tramos analizados, por lo que es más rápida.

5. Listado de los resultados. Una vez concluido el análisis de la traza, los impulsos detectados pueden listarse, bien sea en orden cronológico dentro de la traza, o agrupados por clase.

6. Almacenamiento de los resultados. Los datos de la detección y clasificación pueden almacenarse en un archivo escrito en modo ascii, con ciertas características para poder ser utilizado posteriormente por otros programas. En la opción se describe la forma como escriben los datos, básicamente, la clase y el tiempo de detección de cada impulso en un renglón.

7. Otras variables. Dependiendo de los datos presentes en el archivo con el registro, se eligen los valores máximo y mínimo esperados para normalizar la señal que es presentada a la red clasificadora. Los valores mostrados son los adecuados para la señal simulada. En situaciones reales, estos valores dependerán de los valores máximo y mínimo que entregue el conversor analógico digital con el que se digitalice el registro y los utilizados para la normalización.