UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE CIENCIAS

LAS REDES NEURONALES ARTIFICIALES TIPO CASCADA EN EL CONTEXTO DE LA MECANICA ESTADISTICA

T E S I S

QUE PARA OBTENER EL TITULO DE

F I S I C O

P R E S E N T A :

FIDEL SANTAMARIA PEREZ

DIRECTOR: DR. JOSE ISMAEL ESPINOSA ESPINOSA

MEXICO, D. F. 1994

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

VNIVERSIDAD NACIONAL
AVFNOMA DE
MEXICO

M. EN C. VIRGINIA ABRIN BATULE
Jefe de la División de Estudios Profesionales
Facultad de Ciencias
Presente

Los abajo firmantes, comunicamos a Usted, que habiendo revisado el trabajo de Tesis que realiz(ó)ron el pasante(s) SANTAVARIA PEREZ FIDEL

con número de cuenta 9052258-9 con el Título:

"Las redes neuronales artificiales tipo cascada en el contexto de la mecánica estadística".

Otorgamos nuestro **Voto Aprobatorio** y consideramos que a la brevedad deberá presentar su Examen Profesional para obtener el título de Físico

GRADO	NOMBRE(S)	APELLIDOS COMPLETOS	FIRMA
DR	JOSE ISMAEL	ESPINOSA ESPINOSA	
Director de Tesis DR.	FEDERICO	ZERTUCHE MONES	
DR.	HENRI WAELBROECK	GOGNEAUX	
DRA.	GERTRUDIS HORTENSIA	GONZALEZ GOMEZ	
Suplente DR.	RAMON	PERALTA FABI	
Suplente			

D. O. M.
A mis padres, mi hermano, Xomalin y Jose María Escrivá de Balguer.

Agradecimientos

Dr. José Ismael Espinosa Espinosa, por darme un lugar en el laboratorio de Cibernética e introducirme a la interdisciplina.

Jorge Quiza Tomich, sin su ayuda me hubiera tardado otro año en entender lo que tenía que hacer.

J. Jesús González Fernández, por ser el que me introdujo a estos temas.

Roberto Serna, por la colaboración para avanzar más rápido en beneficio del laboratorio.

Irma Domínguez, Luis Felipe Rivera, José Guevara y Rubén Arroyo Angeles, por aguantar mis constantes ataques de organización.

Rafael Cabello y Jordi Iñaki, por su compañerismo.

A los sinodales Henri Waelbroek, Federico Zertuche, Horntensia Gonzalez y Ramón Peralta. Por sus anotaciones y recomendaciones.

Este trabajo fué apoyado en parte por el proyecto DGAPA IN100593 en Ciencias Exactas denominado: "Dinámica de la interacción neuronal en sistemas complejos del cerebro de la rata".

INDICE

RESUMEN

Primero se presentan las partes y funcionamiento de una neurona biológica. Después, con un esbozo histórico de las redes neronales artificiales (RNA) se recalca la participación de físicos en el tema. Por otro lado, se ve el núcleo principal desde el punto de vista físico, en cuanto a la modelación del funcionamiento del cerebro.

Se revisan los conceptos básicos que definen a una RNA como son la arquitectura, el algoritmo de aprendizaje y la interpretación geométrica. En particular se analizan todos estos aspectos en la arquitectura de Cascada. Se estudian los teoremas de Funahashi [Funahashi, 1989], que aseguran que una red en cascada es un mapeador universal y se analizan sus consecuencias geométricas y sus relaciones con el espacio de Hilbert L^2. Con la ayuda de las propuesta de Amit [Amit, 1989] y Peretto [Peretto, 1992] para las funciones de activación de los nodos y una arquitectura permitida por Funahashi, se reinterpretan este tipo de RNA como sistemas estadísticos.

Se ve la comparación de la función de error con la función logística en el entrenamiento de redes neuronales artificiales tipo cascada. Los resultados de esta comparación demuestran que la función de error es mejor, en el sentido de aprender más rápido, que la función logística. Además se comprueba computacionalmente que las redes tipo cascada son sistemas estadísticos.

Se introduce el concepto de redes con invariancia a la translación, para después hacer una aplicación a reconocimiento de señales eléctricas provenientes del hipocampo de rata.

Finalmente, se discuten posibles caminos ha seguir en el campo de análisis teórico de las RNA y se proponen posibles aplicaciones de estas para el reconocimeinto de señales eléctricas provenientes del hipocampo de rata.

INTRODUCCION

La mecánica estadística se encarga del estudio de conjuntos grandes de partículas, ejemplo de ello son los vidrios de espines, el templado, los gases y líquidos entre otros. En este contexto se definen las redes neuronales artificiales (RNA) que se entienden como conjuntos de partículas de dos estados que interactúan ferro y antiferromagnéticamente y que tienen capacidades computacionales emergentes [Hopfield, 1982]. Esto es, se pasa de un estado inicial del sistema con cierta energía a otro final con otra energía y se asocia cada estado de energía a cierto tipo de información. Varios métodos de la teoría de la probabilidad que se usan en mecánica estadística han sido aplicados a estos modelos como son el método Montecarlo de probabilidades y las cadenas de Markov. Al mismo tiempo, los modelos de redes neuronales han motivado nuevos problemas a la mecánica estadística como son el estudio de interacciones de matrices asimétricas [Gutfreund, 1990].

Una RNA puede ser pensada como un conjunto de espines. El estado de cada espin depende de su estado de energía y de su relación distancia y energía -con los demás espines. Desde la mecánica estadísitca, se puede plantear una función de energía del sistema, la cual, se pueda maximizar, minimizar o simplemente llegar a un estado deseado de energía. Es claro que al hablar de estos temas se está hablando de sistemas probabilísticos.

Varios físicos se han adentrado en el tema, como L. Cooper [Cooper, 1973] -la C en la teoría BCS de la superconductividad-, que propone que el aprendizaje es la ortogonalización -en un cierto espacio- de los patrones que se quieran aprender. Otro de los físicos es Little [Little, 1974], quien supone que el cerebro es un sistema estadístico y propone que los modelos de éste deben de contemplar esta característica.

Actualmente, además de ser objeto de estudio de la física, las RNA son una herramienta útil en la misma, ya que tienen la cualidad de aprender y generalizar distintos tipos de señales y de categorías. En experimentos donde el ruido y la variedad de variables que se quieren investigar es grande las RNA pueden simplificar el trabajo. La identificación de partículas por sus trazas marcadas al salir de un acelerador

[Humpert, 1990; Denby, 1990], en análisis y predicción del comportamiento de sistemas caóticos [Selvam, 1989], la solución a la ecuación de Schrödinger [Darsey, et al, 1991] son algunos ejemplos.

0.1 OBJETIVO.

En la sección anterior se ha visto que las RNA han sido tratadas estadísticamente, sin embargo, existen teorías sobre las RNA que no consideran el carácter estadístico del sistema [Rumelhart, 1986; Kohonen, 1984]. El objetivo de la tesis es demostrar heurísticamente el carácter probabilístico y estadístico de un tipo especial de RNA, que son llamadas en cascada y que tradicionalmente han tenido una representación determinista [Rumelhart, et al, 1986]. Este tipo de redes son definidas como un conjunto de partículas, también llamadas nodos, que están dispuestas en capas. Estos sistemas tienen un conjunto de partículas, llamadas de entrada, que actúan como el receptor de las perturbaciones externas y otro conjunto que actúa como la salida del sistema; entre ambas pueden existir varias capas, inclusive ninguna, llamadas ocultas. En este tipo de arquitectura, las interacciones sólo ocurren entre nodos de capas contiguas, nunca entre nodos de la misma capa. Usando los Teoremas de Funahashi [Funahashi, 1989], que aseguran que toda función en los reales puede ser aproximada por una RNA de este tipo, y una función de probabilidad entre partículas de dos estados propuesta por Amit [Amit, 1989] y Peretto [Peretto, 1992]. Con esto se reinterpretará a las RNA tipo cascada como un sistema estadístico, esto es, un sistema de partículas que tienen dos estados: disparando o en el estado excitado, y silente o en el estado base. Esta característica resultará en que la red globalmente dará la probabilidad de reaccionar de cierta manera a una perturbación determinada.

Usando lo anterior, se hará una aplicación a reconocimiento de señales eléctricas provenientes de neuronas de rata anestesiada. La necesidad de esta aplicación es que cada neurona tiene una señal eléctrica característica (potencial de acción), distinta de las demás. A partir del comportamiento temporal de estas señales, se pueden determinar las conexiones que existen entre las neuronas de la región que se está analizando. Así se puede

determinar el tipo de redes neuronales biológicas que existen en el cerebro de rata anestesiada.

0.2 CONTENIDO.

El presente trabajo se encuentra dividido en cinco partes:

Capítulo Primero. Las redes neuronales en el contexto de la física.
Se presenta información general sobre la forma, partes y estructura de la neurona, redes de éstas y el cerebro. Se da una visión histórica del desarrollo de esta disciplina, resaltando su interdisciplina y la crucial intervención de la física. Se expone el núcleo de las ideas de los físicos que han aportado más a este campo, como son Cooper, Hopfield, Little y otros.

Capítulo Segundo. Las redes neuronales en el contexto de la estadística descriptiva.
Sobre las Redes neuronales artificiales se dan definiciones, terminología y formas gráficas y matemáticas de representarlas. Se analiza el caso de redes en cascada, que tradicionalmente se han interpretado como sistemas deterministas. Además, se usan los teoremas de Funahashi [Funahashi, 1989] para justificar la disposición de las partículas que forman la red, y se analizan sus consecuencias geométricas y su relación con las funciones estadísticas. Por otro lado, usando la propuesta de Amit y Peretto, se reinterpreta este tipo de RNA como un sistema estadístico, sustituyendo la función logística (FL) con una función que mide la probabilidad de que una neurona real dispare un pulso o no (FE). Se codifica una RNA tipo cascada que cumpla las condiciones de Funahashi.

Capítulo Tercero. Visión estadística de las redes en cascada y ejemplo de aplicación.
Se corrobora experimentalmente que las redes en cascada son sistemas estadísticos. Además, que la función sigmoidal de error del capítulo 2 acelera el aprendizaje en comparación con la función logística. Para comprobar la afirmación de que una RNA tipo cascada es un sistema estadístico, se miden las probabilidades de las partículas que se usarán de

salida del sistema. Ya que estas no interactúan entre ellas, la probabilidad total de la capa de salida será la suma de las probabilidades de cada una de las partículas que componen esta capa. Para evaluar el desempeño de la red propuesta, ésta se entrena con los problemas más usados para evaluar redes; estos son el DECODIFICADOR y el XOR. El primero consiste en que la señal de entrada sea igual a la de salida y el segundo es la función lógica. Además, se comparan resultados con redes con la misma arquitectura pero que en lugar de la FL la usen FE, la cual es la más usada en este tipo de redes. Por su rapidez, se usa el algoritmo de aprendizaje de RNA llamado Retropropagación.

Capítulo Cuarto. Redes estadísticas invariantes a translación del patrón de entrada y ejemplo de aplicación. Se usa una red en cascada, que tiene la característica de ser invariante a la translación de los patrones presentados a la red, y los resultados de los capítulos III y IV para demostrar que este tipo de redes cumple todas las restricciones de los teoremas de Funahashi. Así, se obtienen redes estadísticas invariantes a translación. Se discute sobre la aplicación de este tipo de redes en el reconocimiento de patrones provenientes del cerebro de rata anestesiada. Se hacen pruebas en patrones sintéticos que tienen las formas que se encuentran en el cerebro y se trata de aplicar a señales reales.

Conclusiones:
Se revisan los conceptos vistos en los cinco capítulos y los resultados obtenidos. Se discute sobre posibles aplicaciones y caminos teóricos que se pueden seguir investigando.

Apéndice A. Código fuente de los programas creados.

CAPITULO PRIMERO

LAS REDES NEURONALES ARTIFICIALES EN EL CONTEXTO DE LA FISICA

Primero se presentarán las partes y funcionamiento de una neurona biológica. Después, con un esbozo histórico de las RNA se recalcará la participación de físicos en el tema. Por otro lado, se verá el núcleo principal del punto de vista físico en cuanto a la modelación del funcionamiento del cerebro.

1.1 LA NEURONA BIOLOGICA.

Se desarrollan brevemente las características generales de las neuronas y del potencial de membrana.

1.1.1 Partes de una neuronay potencial de reposo.

La neurona es una célula. Esta célula se puede entender como la mínima unidad para comunicación de información. La forma de la neurona es singular y su axón se ramifica y va a contactar a otras neuronas en una estructura llamada *sinápsis*. Una forma típica de la neurona es como en la figura 1.1. Las partes principales de esta célula son el soma o cuerpo, las dendritas, el axón y el botón sináptico. La neurona mantiene una diferencia de potencial constante respecto al medio que lo rodea, esta diferencia es llamada potecial de reposo, que es de alrededor de -70 mV.

1.1.1.1 Soma o Cuerpo.

Es la parte central de la neurona, ahí se encuentran el núcleo y demás organelos celulares.

1.1.1.2 Dendritas y Botón Sináptico.

Las dendritas son ramificaciones que se originan en el soma. Estas, acaban en el botón sináptico. La mayoría de las veces el botón sináptico queda muy cerca de otra neurona (puede tambier quedar cerca de un músculo), en promedio el espacio que existe entre el botón sináptico y la membrana de la siguiente neurona es de 20Å. A esta estructura que forma el

botón sináptico de una neurona con otra se le llama sinapsis. Las dendritas reciben el mayor número de cantactos siápticos. Es en este lugar donde se lleva a cabo la transmisión de la información, ya que cuando el potencial de acción alcanza el final del axón se genera una serie de procesos electroquímicos y físicos que causan la liberación de moléculas llamadas neurotransmisores hacia el espacio sináptico (ver la Fig. 1.2), donde se encuentra el fluido extracelular. Esto es, salen de la neurona (presináptica) e interaccionan con los receptores de la neurona (postsináptica) con que se está haciendo contacto. Al llegar a ella, por otra serie de mecanismos bioquímicos, se modifica el potencial de reposo de la neurona postsináptica. Se puede decir que la información de la neurona presináptica es transmitida a la postsináptica por medio de una combinación de mecanismos bioquímicos.

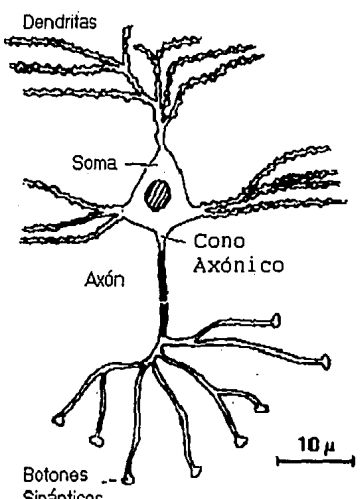


Fig. 1.1 Representación esquemática de una neurona [Peretto, 1992].

1.1.1.3 Axón.
Es una especie de cable submarino, ya que dentro y fuera de él existe medio conductor, la membrana hace la vez de aislante. Tiene su origen en el soma y que en su parte terminal se ramifica. Cada ramificación va a contactar ya sea, a las dendritas de otras neuronas, o al soma, al axón o a una fibra muscular.

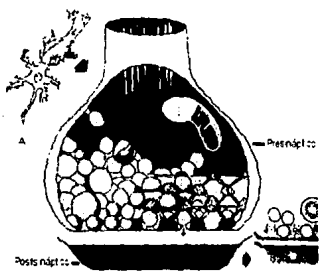


Fig 1.2 Unión sináptica. (A) Conjunto de conexiones entre axones y dendritas. (B) Una sinápsis donde se muestra el contenedor vesicular y vesículas de neurotransmisores incorporados a la membrana presináptica. Una de estas está abierta al espacio sináptico y libera el neurotransmisor que viajará a la membrana postsináptica [Amit, 1989].

1.1.2 Potencial de Acción Axónico.
Es un cambio temporal en la diferencia de potencial que existe entre el exterior e interior celular. La generación de este potencial depende de la cantidad de neurotransmisores recibidos de las neuronas con las que se hace sinápsis. Para generar un pulso se ha visto que es necesario rebasar un umbral de potencial. Por otro lado, como la forma del potencial de acción es siempre igual para la misma neurona, la amplitud de este potencial no es una variable importante, mas bien, es la frecuencia con la que llegan a la neurona postsináptica. Otra variable es el lugar donde se hace la sinápsis recpecto al soma, si es muy lejos tendrá menos influencia y si es más cerca tendrá más. Esto

es, se puede pensar a la neurona como un integrador - sumador- espaciotemporal, ya que en un intervalo de tiempo determinado y en una área dada se reciben varios potenciales de otras neuronas que al integrarse de manera no lineal pueden sobrepasar el umbral de activación generando con esto un potencial de acción.

La forma de este potencial es como se muesta en la figura 1.3,

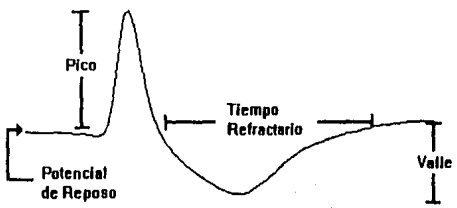


Fig. 1.3 Potencial de acción, también llamado espiga.

Sin embargo, en general estos potenciales no son tan limpios como en la figura 1.3, sino que tienen ruido (Fig. 1.4). El ruido es causado porque en el momento del experimento también se estan registrando potenciales de otras neuronas y existen variables que no se pueden controlar.

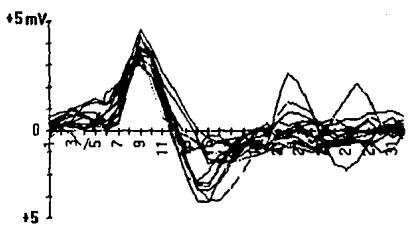


Fig. 1.4 Potenciales de acción celular de la reción CA3 del hipocampo de rata. La amplitud del pico es de 5mV.

Al parecer, todo lo que se ha explicado es parte de un proceso determinista. Sin embargo, las variables que rigen este sistema se pueden considerar aleatorias, ya que aunque el sistema pueda ser determinista, este es muy complejo. Por lo tanto, es mejor que se hable de la probabilidad de generar un potencial de acción.

Los potenciales pueden ser excitatorios (inducen a la neurona postsináptica a generar un potencial de acción) o inhibitorios (bloquean la generación de un potencial de acción en la neurona postsináptica).

1.2 HISTORIA DEL MODELADO NEURONAL.

El estudio de las interacciones de los seres vivos con el medio que les rodea ha tenido muchos expositores a lo largo de la historia. Sin embargo, un estudio científico formal se empieza a fraguar hasta en el siglo XVII. Este desarrollo se puede visualizar como un árbol con cuatro ramas principales: Biología, Matemática, Física y Psicología. A continuación se dará una serie de breves notas para recalcar la crucial participación de físicos y presentar las teorías físicas modernas para el entendimiento y simulación de la actividad neuronal. Este grupo de notas trata de mostrar el desarrollo interdisciplinario que se ha dado en varias ramas de la ciencia (entre ellas la física) para llegar a lo que ahora se conocen como Neurociencia Computacional y Redes Neuronales Artificiales (RNA).

1.2.1 Descartes.

Uno de los primeros intentos por dilucidar la relación del cerebro con las actividades de los seres vivos fue hecho por Descartes, quien plantea que las actividades de los seres está basada en reflejos condicionados. Esto es, un estímulo externo captado por los sentidos (resequedad en la garganta) será mandado al cerebro por los nervios y este lo interpretará de una única manera mandando una respuesta (tomar agua) [Rothschuh, 1973].

Para Descartes, pequeñas partículas viajan en la sangre hacia el cerebro y penetran a los ventrículos cerebrales y a la glándula pineal por una pequeña abertura. Una vez en estas estructuras, las partículas son convertidas en líquidos animales; una especie de aire fino o líquido que fluye a través de la substancia nerviosa y entra a los

nervios, que son huecos. Por otro lado, estos líquidos circulan a través de los nervios hacia los músculos del cuerpo, donde estos son inflados provocando así el movimiento. El hombre, como el resto de la naturaleza, trabaja determinísticamente bajo las leyes de la causa y el efecto.

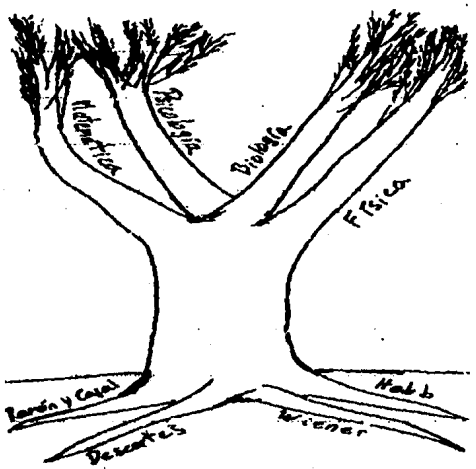
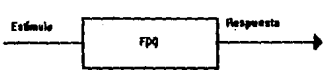


Fig. 1.5 El árbol del estudio de los procesos cognitivos.

Viendo estas propuestas bajo la teoría de control es como si tuviéramos una función de lazo abierto. En símbolos:



La teoría de Descartes ha sido superada desde hace mucho. Una de las muchas cosas que deja esta teoría de lado, es que no toma en cuenta la capacidad autoregulativa de los seres.

1.2.2 Ramón y Cajal.

A finales del siglo pasado se editaron los estudios clásicos de Santiago Ramón y Cajal. En estos, demuestra la existencia de las neuronas como entidades independientes y discretas, además de los árboles dendríticos (Fig. 1.6). Estos trabajos tienen gran importancia ya que gracias a ellos se supo que el cerebro es un conjunto de estos elementos y no un gel. También dio paso a la neurobiología y neurofisiología modernas.

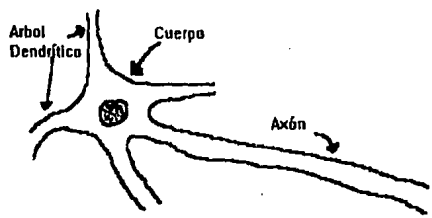


Fig. 1.6 La neurona según Ramón y Cajal.

1.2.3 Hebb.

Donald O. Hebb nació a principios de este siglo en Nova Scottia, Canadá. Primero estudió para escritor pero finalmente hizo estudios de posgrado en psicología. Después de algunos años de experimentar con condicionamiento pavloviano, editó un libro titulado "The Organization of Behavior" [Hebb, 1949]. Con éste, culminó una serie de investigaciones y marca un parteaguas en el estudio de la neurobiología, psicología y pone un precedente para la Neurociencia Computacional.

La idea de Hebb para el aprendizaje es que está íntimamente ligado con la relación sináptica entre las neuronas y que un cambio estable en ella va a resultar en un conocimiento nuevo adquirido. En otras palabras, si dos neuronas tienen actividad, la relación o fuerza que existe entre ellas va a intensificarse (mayor excitación). Esto implica que el conocimiento se da en un conjunto de neuronas y que el aprendizaje se da por las interacciones entre ellas. Se puede idealizar a una red de neuronas tal que

puedan tener cualquier distribución espacial y cualquier tipo de conexiones entre ellas (Fig. 1.7), como una matriz en la que sus elementos son las sinapsis, entendiendo por esto una frecuencia de disparo que ejerce la neurona i hacia la j, pudiendo ser esta excitatoria o inhibitoria.

Sea J la matriz de sinapsis. Entonces el cambio, para cada elemento de matriz, estará dado por

$$J_{ij} = J_{ij} + \eta x_i x_j \qquad 1.1$$

donde la x_i es la actividad de la neurona i y la η se le puede llamar el porcentaje de la conexión sináptica entre la neurona j y la i. Esta forma se la llama ahora aprendizaje tipo Hebb. Esta regla ha sido usada, con algunas variantes y con interpretaciones particulares, por gran parte de las personas que se dedican a las redes neuronales artificiales.

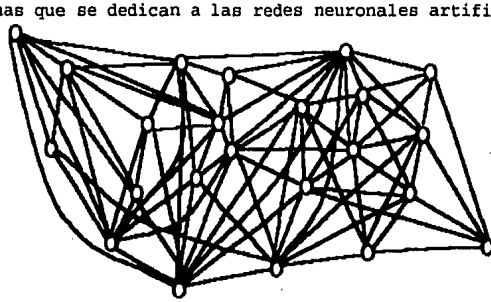


Fig. 1.7 Una red neuronal. Los nodos (circulos) tienen una cierta actividad, la cual es transmitida por los lazos, que la incrementan o disminuyen, según sea la interacción entre el par de neuronas. No existe una disposición predeterminada de los elementos.

1.2.4 McCulloch y Pitts.

"A logical Calculus of the Ideas Inmanent in Nervous Activity" por Warren S. McCulloch y Walter Pitts [McCulloch y Pitts, 1947] se puede tomar como el primer intento de modelar neuronas y conjuntos de ellas.

La hipótesis principal es que basta definir una neurona como un elemento todo-o-nada, es decir, que dispara o no dispara; el tipo de pulso puede ser excitatorio o inhibitorio. Cuando el pulso sea excitatorio serán de la

misma intensidad y cuando sea inhibitorio anulará todos los pulsos excitatorios causando a la neurona a la que se conecta que no dispare en absoluto. Se resumen estas hipótesis en los siguientes cinco postulados:

a) La actividad de una neurona es un proceso todo-o-nada, es decir una función escalón.

b) Un cierto número fijo de neuronas deben estar excitadas dentro del período de latencia para así excitar a otra neurona en cualquier momento, y su número es independiente de la actividad previa y posición de la neurona.

c) El único retraso considerado es el retraso sináptico.

d) La actividad de cualquier sinapsis inhibitoria tiene como consecuencia la no excitación de la neurona en ese instante.

e) La estructura del arreglo de neuronas no cambia con el tiempo.

La demostración más fuerte que existe en el trabajo de McCullog y Pitts es que toda expresión lógica finita puede ser realizada por este tipo de redes de neuronas artificiales.

La salida de una neurona es -1 (inhibitoria), 0 (neurona inhibida) ó 1 (excitada).

La dinámica del sistema consiste en que cada neurona artificial (NA) tiene un cierto número de conexiones, dependiendo de las reglas antes mencionadas; cada NA reaccionará al impulso que le llegue.

Se hacen notar algunos problemas:

a) No hay aprendizaje: nunca se declara la forma en que la red aprende. Por el contrario, el que implanta el sistema tiene que fijar las conexiones.

b) Es un proceso determinista (las neuronas están en -1, 0 ó 1): En la naturaleza las variables que rigen las relaciones entre las neuronas son tan complejas que es mejor dar un tratamiento estadistico, es por eso que es mejor hablar de la probabilidad de disparo.

c) Si se actualizan los pesos es en paralelo (todos al mismo tiempo): Para poder contruir otra función se tienen que cambiar todas o algunas conexiones y después reiniciar el sistema.

A pesar de lo anterior, tenemos que recordar que es 1943. Todavía no existían técnicas para medir la actividad interneuronal, ni tampoco las ecuaciones de Hodgkin y Huxley sobre los potenciales de acción.

Este trabajo influenció a von Neumann en el momento en que él estaba pensando en computadoras digitales.

1.2.5 von Neumann.

En el año de su muerte, 1957, John von Neumann dio una conferencia en la universidad de Yale titulada La Computadora y el Cerebro [von Neumann, 1957]. En esta plática von Neumann hizo una reflexión sobre las analogías que se pueden hacer entre una computadora digital y una analógica, con el sistema nervioso. También analizó y propuso que el funcionamiento del cerebro se basa en medios estadísticos y no en algo determinado absolutamente por la conectividad del mismo, también planteó algunos problemas.

Un problema es el de la precisión con la que las neuronas deben de trabajar, i. e., el error que cometen al pasar la información. Se sabe que las neuronas biológicas no tienen una precisión mayor a 2 o 3 cifras. Con esta información, supóngase que se necesitan 10^4 neuronas para que una persona tome un vaso con la mano y que la propagación de errores es su suma con una precisión de 10^{-3}. Entonces resulta que en la posición tenemos 10 cm. de incertidumbre, lo que es más que el diámetro del vaso.

Otro de los problemas planteados fue el de la arquitectura y comunicación del cerebro y su implantación en máquinas electrónicas. Aquí, se usa la analogía del cerebro con partes digitales y analógicas; así pues, dice que la comunicación entre neuronas es analógica, ya que son corrientes las que pasan por los axones. Por otro lado, se ha visto que el disparo de una neurona no depende de la intensidad del pulso, sino de su frecuencia, i. e., suma de los pulsos en un período dado de tiempo, por lo tanto tenemos aquí un sistema digital. Así mismo sugiere que los genes son parte de un sistema digital.

Otra forma de ver el funcionamiento del cerebro, continúa von Neumann, es verlo como un código que se le da a una máquina para ejecutar una tarea específica, pero recalca que a diferencia de una computadora *normal* en el cerebro podemos quitar un pedazo y éste seguirá haciendo la tarea

correctamente. Esto conduce a que el funcionamiento de un conjunto de neuronas no se rige por un código en el que el flujo de la información se encuentra determinado sino que ésta, vista como un conjunto, fluye de un lugar a otro corrigiendo sus errores. Así pues, podemos decir que la información circula con un carácter estadístico.

1.2.6 Rosenblatt.

Un año después de la conferencia de von Neumann, un psicólogo de la universidad de Cornell, Frank Rosenblatt, dio a conocer el perceptrón, un tipo de RNA en cascada. Esta fue la primera máquina neuronal y en un principio tuvo mucho éxito. Lo que hizo, fue agregar un proceso para ajustar el peso de las conexiones al modelo de McCulloch-Pitts. Rosenblatt analizó perceptrones (Fig. 1.8) de dos o tres capas de procesadores, pero sólo pudo probar que el de dos capas podía separar dos entradas a dos clases si estas eran linealmente separables, es decir, si se grafican las entradas estas pueden separarse por un plano. Este resultado se llamó el teorema de convergencia del perceptrón, se ajustan los pesos entre las capas de entrada y salida proporcionalmente al error entre la salida computada y la deseada. Generalmente la primera capa recibía la información del mundo exterior que en el experimento original era una matriz fotosensible, por eso recibió el nombre de retina, la segunda, procesaba la información creando una imagen interna de lo que la retina *observaba* y la tercera era la de salida (Fig. 1.8).

La meta principal de Rosenblatt era demostrar, analítica y experimentalmente, que .las redes neuronales adaptativas con una interconexión rica y pseudosinapsis no-lineal podían imitar algunas de las funciones cognitivas y que la existencia de estas estructuras no entraba en conflicto con la evidencia biológica de la época. El éxito obtenido por su propuesta fue tal que se hizo una computadora en 1959, la Mark I alpha-perceptron [Rosenblatt, 1957].

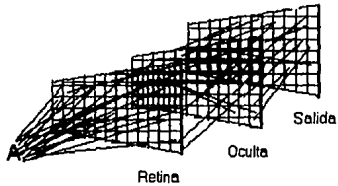


Fig. 1.8 El perceptrón de Rosenblatt. La retina es una matriz fotosensible y la salida pueden ser focos que indiquen la asociación encontrada. Puede existir una rejilla oculta.

1.2.7 Minsky y Papert.

Sin embargo, las redes neuronales artificiales declinaron debido a que no hubiera mayores avances que los del inicio y además sus problemas. Estos fueron los de la falta de grandes generalizaciones y el no poder realizar ciertas tareas. El golpe mas duro y que dejó a las RNA latentes, fue dado por Minsky y Papert en 1969 en su libro "Perceptrons" [Minsky y Papert, 1969]. En él analizan a los perceptrones, con un formalismo matemático claro y demuestran que no pueden realizar ciertas tareas. Sin embargo, no se puede dejar toda la culpa a Minsky y Papert sino que fue una combinación de la falta de resultados nuevos, de pensamientos encontrados y la muerte prematura de Rosenblatt.

1.2.8 Modelación entre los años 50 y 60.

En esta década varios neurofisiólogos y algunos físicos (Gerstein y Caianiello) se dedicaron a la modelación de neuronas unitarias y redes, obteniendo resultados interesantes; la mayoría de ellos se basaron en las ecuaciones de Hodgkin y Huxley, agregando geometría al árbol dendrítico, tomando en cuenta sus propiedades resistivas. Una discusión detallada de las técnicas usadas hasta ese tiempo se encuentra en "Neural Modeling" [Harmon y Lewis, 1966].

a) Rall: Tomó en cuenta la contribución de las dendritas en las propiedades eléctricas de toda la neurona. Propuso como modelo, el llamado cilindro equivalente, que se

usa para analizar los efectos de la forma del árbol dendrítico sobre la conducción de los potenciales postsinápticos. En estos casos, todo el árbol se puede reducir a un cilindro equivalente (Fig. 1.9).

b) Lewis: Estudió el fenómeno de excitación de membrana por debajo del umbral de excitación [Lewis, 1964]. Modeló los fenómenos previos a la generación de un potencial de acción.

c) Gerstein y Mandelbrot: Procesos estocásticos. En la corteza auditiva de gato se encontraban patrones de potenciales de acción no periódicos al ser expuesto el sistema a impulsos periódicos. Modelaron estos patrones por medios estadísticos [Gerstein y Mandelbrot, 1962].

d) Caianiello: Estudió grandes conjuntos de neuronas a nivel de la corteza cerebral, propuso ecuaciones que describían la relación instantánea en una red de partículas que simulaban neuronas [Caianiello, 1961].

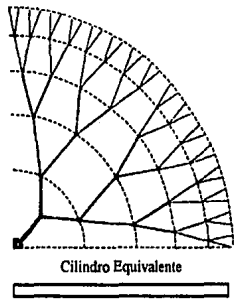


Fig. 1.9 Cilindro equivalente de Rall. Las secciones de un árbol dendrítico se proyectan sobre un cilindro.

1.2.9 Cooper.

Leon Cooper (la C en la teoría BCS de superconductividad) en 1973 se interesa por el tema y publica "A Possible Organization of Animal Memory and Learning" [Cooper, 1973].

Cooper propone un modelo más abstracto comparado con el de Rosenblatt; su meta es simular el comportamiento mental asociado con la memoria y aprendizaje animal. El modelo es

consistente con los conocimientos neurofisiológicos de la época. Sin embargo, la hipótesis más importante es que se debe de suponer que hay una comunicación entre el cuerpo celular y la terminal de la dendrita en dirección opuesta al flujo de las señales eléctricas. Aquí el autor también introduce un elemento muy usado en física: las transformaciones lineales. Para justificar el uso de éstas, en sistemas no lineales, argumenta que existen modelos en que los potenciales neuronales son promediados en períodos cortos y se obtienen muy buenos resultados.

Para usar las transformaciones lineales se deben definir espacios donde estén los vectores de eventos y donde se quieran mandar. Sea E el espacio de eventos que interesan al sistema y $\{e^i\}$ el conjunto de vectores que describen cada evento. Por una transformación (lineal o no lineal) se debe de pasar del espacio E a un espacio F por medio de los sensores, con la condición de que la distancia entre los mismos se mantenga; falta aún definir la distancia. Ya dentro de F, que bajo la analogía son las neuronas de entrada, se debe de hacer otra transformación al mismo conjunto de neuronas o a otro G, por medio de A; esta transformación A puede ser variable con el tiempo y puede depender de las acciones pasadas. Es aquí donde se dice que A es una transformación lineal, ya que físicamente lo que está pasando es que la sinapsis entre un conjunto de neuronas y el otro está cambiando. Así, se puede definir un conjunto G y uno F de neuronas que están interactuando y que ese mapeo sobre G está dado por A.

1.2.10 Teuvo Kohonen.

En 1984 Teuvo Kohonen, un pionero en el estudio del diseño electrónico y diseño de memorias asociativas, publica un libro titulado "Self-Organization and Associative Memory" [Kohonen, 1984]. En él se propone una nueva dinámica para la corrección de los pesos basada en la competencia de los nodos en la capa de salida.

El sistema consiste de una sola capa de nodos que actúan como la entrada y la salida del sistema (Fig. 1.10). Estos están conectados al vector de entrenamiento por una matriz de pesos; la respuesta de cada uno de los nodos es el valor de la proyección del vector de pesos sobre el vector de entrada, el producto interno (una función que se define).

Por sencillez se usa la métrica Euclidiana, que define el producto interno de w (pesos) con x (vector de entrenamiento) como

$$\vec{S} = \vec{w} \cdot \vec{x} = \sum_{i=1}^{N} w_i x_i \qquad 1.2$$

Si los pesos y eventos están normalizados tendremos que el nodo con máxima respuesta (intensidad) será precisamente el que tenga la mayor proyección sobre el evento, por lo que el criterio para elegir al ganador será el que sea el mayor, numéricamente.

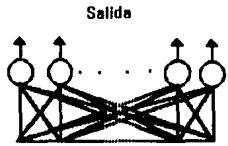


Fig. 1.10 El aprendizaje tipo Kohonen requiere sólo de una sola capa de procesadores.

Este nodo tendrá el derecho de cambiar sus pesos para poder acercarse más al vector de eventos de la siguiente forma

$$w'_{i+1} = w_i + \alpha\left(x - w'_i\right) z^I \quad , \qquad 1.3$$

donde α es un valor entre 0 y 1 que puede variar en el transcurso del entrenamiento, z una función que puede ser de dos maneras: 1 para el nodo que gana y 0 para todos los demás o una función que es máxima para el nodo que gana e inhibe a todas o algunas de los nodos que lo rodean (Fig. 1.11).

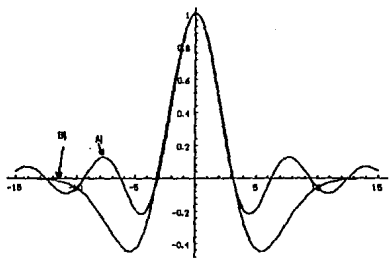


Fig. 1.11 Si las marcas son nodos estos serán excitados o inhibidos según la posición relativa con respecto al nodo central pudiendo ser las funciones de distintas formas: A) La función $Sen(x)/x$, B) la segunda derivada de $Exp[-x^2/20]$.

1.2.11 Hopfield.

En 1982, Hopfield con "Neural Networks and Physical Systems with Emerging Collective Computational Abilities" [Hopfield, 1982], genera un nuevo marco teórico al problema, es entonces que las RNA adquieren un segundo aire que todavía se esta viviendo.

Hopfield propone un sistema capaz de aprender una serie de estados $\{V^s\}$ por medio de elementos individuales V_i. Su punto de vista es que existen estados estables que una vez alcanzados por el sistema, éste se quedará ahí. Cada elemento de proceso i -llamado neurona- tiene 2 estados $V_i=0$ (no dispara) y $V_i=1$ (dispara a frecuencia máxima). Este estado cambia en el tiempo dependiendo del siguiente algoritmo: Para cada neurona i existe un umbral U_i; cada neurona i reajusta su estado al azar, en el tiempo, pero con un promedio de intento W. Haciendo

$$\left.\begin{matrix} V_i \to 1 \\ V_i \to 0 \end{matrix}\right\} \text{si} \quad \sum_{j \neq i} T_{ij} V_j \begin{cases} > U_i \\ < U_i \end{cases}. \qquad 1.4$$

La forma de T_{ij} se propone como

$$T_{ij} = \sum_{\epsilon} (2V_i^{\epsilon} - 1)(2V_j^{\epsilon} - 1). \qquad 1.5$$

Así que cada neurona se evaluará asíncrona y aleatoriamente si está por arriba o abajo del umbral.

En este modelo también se tiene la hipótesis de Cooper de que debe de existir una retroalimentación.

La matriz T_{ij} es simétrica y se define una función de energía E

$$E = -\frac{1}{2} \sum_{i \neq j} \sum T_{ij} V_i V_j . \qquad 1.6$$

Así pues

$$\Delta E = -\Delta V_i \sum_{j \neq i} T_{ij} V_j \ . \qquad 1.7$$

Entonces, el algoritmo que modifica a V_i causa que E sea monótonamente decreciente (ver 1.4.2).

Es aquí donde se hace una analogía, las neuronas artificiales son como espines que van cambiando su estado, de +1 a -1 o al revés, hasta encontrar una E-mínima (posiblemente local), este caso es isomorfo con un modelo de Ising. T_{ij} ocupa el lugar de la pareja variante y hay también un campo externo local en cada punto (neurona artificial). Cuando el estado inicial de T_{ij} es simétrico pero aleatorio -el vidrio de espín- se sabe que hay muchos puntos estables locales, en otras palabras, la curva de energía tiene muchos mínimos locales y si se pone al sistema lejos de uno de ellos el proceso de interacción entre las neuronas artificiales (algoritmo que modifica a V_i) causa que el sistema se vaya acercando gradualmente a un estado estable, así como los espines que se alinean con el campo magnético presente cuando el material se enfría. Claramente el problema es llegar al mínimo total en un tiempo rápido.

1.2.12 Rumelhart et al.

En 1986 Rumelhart y el grupo PSP -Parallel Distributed Processing - publicó un libro en dos volúmenes titulado "Parallel Distributed Processing: Explorations in the Microstructure of Cognition", [Rumelhart et al, 1986]. En estos libros se condensan ideas y hechos acerca de los procesos cognitivos de los seres vivos explicando que esto se puede dar sólo con procesamiento en paralelo o llamado

también distribuido. Por otro lado, nos muestran resultados computacionales que desarrollaron a las RNA enormemente.

Dentro de los resultados computacionales que se muestran está el de un algoritmo que aprende representaciones internas por medio de la propagación del error, comúnmente llamado Backpropagation o en español Retropropagación (BP) -aunque el algoritmo fue inventado por Paul Werbos [Werbos, 1974] en los setentas y separadamente por Parker [Parker, 1987] y le Cun [le Cun, 1987]. Este algoritmo usa una ley de aprendizaje llamada ley delta y cálculo diferencial y de diferencias.

Se necesita que la red esté en capas, como en la figura 2.2, y los pesos que existen entre cada nodo se les dá un valor inicial aleatorio.

Se ecoge un conjunto de pares de ejemplos que se dividirán en dos subconjutos. El primero será el conjunto de entrenamiento que consiste de pares de asociaciones (el ejemplo y su asociación). Después de realizado el entrenamiento se ve su desempeño con el otro subconujunto llamado de prueba. Esto es, se toma un conjunto de entrenamiento, este conjunto de entrenamiento se puede representar en forma de un vector (una gráfica bidimensional consiste de puntos y cada punto es una componente del vector), cada componente del vector se asocia con un nodo de la capa de entrada. Análogamente, la asociación (la salida deseada de la red) también se puede representar como un vector. Este algoritmo consiste de dos partes.

La primera parte del algoritmo consiste (Fig. 1.12) en que el ejemplo pasa por la capa de entrada, de ahí cada nodo de esta capa se conecta con los de la capa siguiente multiplicando el valor de la componente del vector por el peso respectivo, a su vez, por medio de la función sigmoidal se calcula la salida de los nodos de esa capa y se repite el proceso hasta llegar a la capa de salida.

La segunda parte del aprendizaje (Fig. 1.13) consiste en comparar el valor de los nodos de la capa de salida con la salida deseada, por medio de una función positiva definida se calcula el error y se aplica ahora el proceso en reversa, esto es, partiendo de la capa de salida se van cambiando los pesos de las capas desde esta hasta la de entrada. Este proceso se repite varias veces (presentando de nuevo el conjunto de entrenamiento y calculando un nuevo

error a la salida) hasta minimizar el error hasta el nivel que el programador desee. Esta forma de corregir se puede dar de distintas maneras, a) aleatoria, presentar los ejemplos del conjunto de entrenamiento de manera aleatoria y actualizar los pesos; b) cíclica, se presenta todo el conjunto y se calcula un error general.

Finalmente, se usa un conjunto de prueba para verificar el buen desempeño de la red.

Así pues, si se define la función de error como

$$E_p = \frac{1}{2}\sum_j (t_{pj} - o_{pj})^2 \quad , \qquad 1.14$$

el subíndice p significa que es el error medido para el p-ésimo tipo de patrón que se quiera enseñar, se hace la suma de las diferencias cuadradas para todos los nodos de la capa de salida (por eso la suma sobre j). Además, se define a las funciones de los nodos acotadas, continuas, derivables y monótonamente crecientes.

El cambio de los pesos estará dado por

$$\Delta_p w_{ji} = \eta \delta_{pj} o_{pi} \quad , \qquad 1.8$$

siendo η un valor entre 0 y 1, llamado la velocidad de aprendizaje (tiene semejanza con el porcentaje de la conexión sináptica de la regla de Hebb), o_{pi} es la función de activación de la neurona i y δ_{pj} está definida para capas de salida como:

$$\delta_{pj} = (t_{pj} - o_{pj}) f'(x_{pj}) \quad , \qquad 1.9$$

siendo t el patrón a aprender, f la función de activación y x es la suma de todas las interacciones que llegan a la neurona j. Y para capas intermedias:

$$\delta_{pj} = f_j'(x_{pj}) \sum_k \delta_{pk} w_{kj} \quad , \qquad 1.10$$

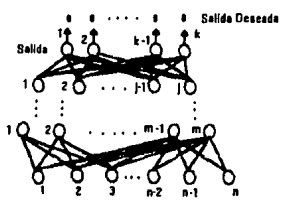


Fig. 1.12 La primera parte de la Retropropagación. Se obtienen valores en los nodos de la capa de salida y se comparan con los valores deseados.

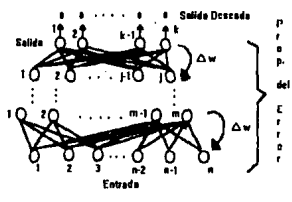


Fig. 1.13 La segunda parte de la Retropropagación. Se calcula la corrección que se debe de hacer a cada peso en dirección opuesta al flujo de la información.

Este tratamiento asegura un gradiente descendente de la función de error (Fig. 1.14). La ecuación 1.15 es llamada la regla delta.

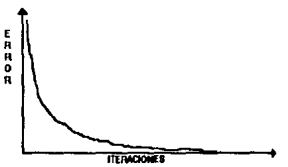


Fig. 1.14 Gráfica típica de una curva de la disminución del error con relación al número de iteraciones que se hacen para corregir los pesos entre las conexiones.

1.3 MODELOS DE REDES SUPERVISADAS Y AUTO-ORGANIZADAS.

La mayoría de los modelos de redes que se han presentado pueden ser separados de dos maneras. Aquellos que se les presenta una entrada y se quiere que aprendan una salida -Rumelhart- y los que se les presentan ejemplos y por los algoritmos de interacción entre los elementos estos mismos aprenden -Kohonen, Hopfield [Simpson, 1990].

1.3.1 Redes Supervisadas.

En las redes supervisadas el aprendizaje requiere de la presentación de los patrones que se van a aprender (entrada) junto con los patrones de salida deseados (salida), esto es, presentar el patrón y su asociación. Esto se hace hasta alcanzar un error mínimo entre la salida real del sistema y la salida deseada. Es claro que se debe de definir una función de error que dependa de las conexiones entre las neuronas y una forma de variar los pesos entre ellas. Esta forma de atacar el problema tiene su mayor representante en el algoritmo de retropropagación. Este algoritmo usa como función de error la diferencia cuadrada de la salida deseada y la real, y calcula una corrección a los pesos basada en la llamada ley delta de aprendizaje. A partir de esta ley se calcula la corrección de la capa de salida hacia la inmediata inferior, de esta última se vuelve a calcular el error y así hasta llegar a la capa de entrada (Figs. 1.12 y

1.13). Este procedimiento se repite hasta alcanzar un mínimo en la función de error. Existen variantes de esta regla para hacerla más eficiente, generalmente se le suman un par de términos lineales más.

1.3.2 Redes Auto-organizadas.

Los sistemas de redes auto-organizadas consisten principalmente de un conjunto de nodos interconectados, los cuales *competirán* bajo ciertas leyes que medirán el desempeño de los mismos; los que obtengan mejor desempeño tendrán el derecho de ajustar sus conexiones con el evento. Los perdedores no podrán cambiar sus conexiones. Como ejemplos tenemos el modelo de Kohonen y el de Hopfield.

1.4 LAS RNA COMO OBJETO DE ESTUDIO DE LA FÍSICA.

A lo largo de este capítulo se ha mencionado -indirectamente- que el estudio del funcionamiento del cerebro ha tenido un desarrollo igual a cualquier otra ciencia, desde la observación y experimentación, hasta las abstracciones. Es en ésta última parte donde surgen los modelos de RNA.

La diferencia entre la propuesta de von Neumann y la de Hopfield recae firmemente en que la primera hace una analogía de un objeto real con otro igualmente real y el segundo la hace de un objeto real a un objeto descrito en un espacio abstracto.

1.4.1 El modelo de neurona.

El modelo aceptado en la actualidad se basa en la hipótesis de que la información -la variable que se quiere analizar, que en este trabajo significa la forma de interactuar entre un par de neuronas- es transportada exclusivamente por las frecuencias de disparo de las neuronas que forman la red.

Si z_i es el potencial de membrana instantáneo y z_0 el potencial de reposo, la interacción entre la neurona i y las demás que están en contacto con ella es de la forma

$$z_i = z_0 + \sum_j J_{ij} z_j \ , \qquad 1.11$$

donde z_0 la podemos poner a cero y J_{ij} es llamada la fuerza sináptica, que se define como el producto de neurotransmisor liberado por la parte pre-sináptica y la eficiencia

sináptica, esto es, la cantidad de neurotransmisor recibido por la parte post-sináptica.

Se ha visto que la frecuencia de disparo -f- es cero cuando la suma de las entradas es muy pequeña y tiene un máximo cuando es muy grande. Esto significa que la función que determina la frecuencia de generación de pulsos en la neurona i es no-lineal.

$$V = f(z_i) = f\left(\sum_j J_{ij} z_j - z_0\right) . \qquad 1.12$$

Definiendo a V como el valor de la función al tiempo t. El caso límite es la función escalón. Debe de tenerse en cuenta que esta ecuación corresponde a una neurona a un tiempo t.

Supóngase ahora una red en la cual la salida de cada neurona va a hacer sinapsis en otras de la misma red. Entonces las salidas de unas se hacen inmediatamente entradas de otras, las cuales determinan las salidas en un tiempo posterior t+1. Sustituyendo la z por la V del tiempo anterior, la dinámica queda determinada por:

$$V_i(t+1) = f\left(\sum_{j=1}^{N} J_{ij} V_j(t) - U\right) . \qquad 1.13$$

1.4.2 Las interpretaciones de Little y Hopfield.

Little, en 1974 [Little, 1974] propone que para estudiar las interacciones neuronales, se debe simplificar el funcionamiento de las mismas a partículas de dos estados energéticos, él propone espines que tengan estados +1 y -1. El estado +1 significa que la neurona genera un impulso excitatorio, el -1 significa que genera un impulso inhibitorio. Por otro lado, afirma que no se puede tratar a la actividad neuronal como un sistema Newtoniano, por el contrario, el tratamiento que se debe dar es el de la mecánica estadística. Se basa en los resultados experimentales de la biología que al parecer, indican que si bien el cerebro no es determinista es un sistema muy complejo para tratarlo deterministamente. Por eso mismo, afirma que la función 1.25 se debe de interpretar como la probabilidad de que la neurona i genere un pulso al tiempo t.

Supóngase que se tiene un cristal de espines a alta temperatura y que la orientación de los espines es aleatoria al tiempo inicial. Un tiempo después se verá otra

configuración de los espines y así sucesivamente. A cada configuración se le puede asociar un valor de una función de energía, ya que se sabe como interactúan los espines; así se pueden buscar mínimos y máximos locales y globales. A cada estado estable se le denominará estado de aprendizaje. Es claro que pueden existir varios estados estables. Por lo tanto, se pueden asociar varios patrones de aprendizaje a un mismo cristal. Es por esto que se propone que la función que determina la generación de pulsos es más bien una función de probabilidad de que se genere un pulso.

Usando la ecuación 1.25 se propone el siguiente cambio de variable,

$$S_i = 2V_i - 1 = \pm 1 \qquad 1.14$$

Si se rescala J para que

$$\frac{1}{2} J_{ij} \to J_{ij} \qquad 1.15$$

se tiene que el argumento se puede expresar como

$$\sum_j J_{ij} S_j(t) + \sum_j \left(J_{ij} - U \right) \qquad 1.16$$

Así, se puede reconocer que la primera suma es la energía de las interacciones de las neuronas y la segunda, por no depender de S_i, la podemos interpretar como un campo magnético externo.

Así pues, la probabilidad de generar un pulso en t+1 queda como

$$S_i(t+1) = +1 \text{ con probabilidad f } (h_i(t))$$

donde

$$h_i(t) = \sum_j J_{ij} S_j(t) + h_i^{ext} \qquad 1.17$$

Se puede ver que $f(h_i(t))$ se puede escribir como [Geszti, 1990],

$$f\left(h_i(t)\right) = \frac{e^{\beta h_i(t)}}{e^{+\beta h_i(t)} + e^{-\beta h_i(t)}} \qquad 1.18$$

para $S_i = \pm 1$ respectivamente, donde β es el inverso de un parámetro que se nombrará, por pura analogía a la termodinámica, temperatura.

Quedando todo resumido de la siguiente manera,

$$S_i(t+1) = \pm 1 \text{ con probabilidad } \frac{1}{1+e^{\mp 2\beta h_i(t)}}. \qquad 1.19$$

Para temperatura cero el modelo se vuelve determinista, entonces la función queda como

$$S_i(t+1) = f\left(\sum_{j=1}^{N} J_{ij} S_j(t) + h_i\right) \qquad 1.20$$

En la misma línea de pensamiento que Little, Hopfield [Hopfield, 1982] piensa en un sistema de espines que usan las ecuaciones dinámicas 1.31 para sistemas ruidosos y la 1.32 para no ruidosos, con las siguientes restricciones:

1) El cambio de estado de cada espín - neurona - en lugar de efectuarse al mismo tiempo se hará de uno en uno, escogiendo el espín a actualizar al azar.

2) Para actualizar la fuerza sináptica se usa la regla de Hebb.

Las fuerzas J_{ii} son cero, esto es porque los espines no generan fuerzas sobre sí mismos y $J_{ij}=J_{ji}$ que es tener en cuenta la tercera ley de Newton; esto es, que la fuerza generada por la partícula i sobre la j es la misma que la generada por la j sobre la i.

1.4.3 El modelo de Cooper.

Cooper hace un análisis más general y más centrado en las transformaciones lineales que en su forma general son:

$$g = \sum_{i=-\infty}^{\infty} C_i f_i, \qquad 1.21$$

donde C es un coeficiente que es el producto interno de la función con un elemento de la base, y f es un elemento de la base. En general el símbolo S puede ser una suma, una integral o ambas y hablar de infinitos significa sobre todos los valores posibles - pueden ser finitos.

En física, los espacios naturales para las transformaciones lineales son los espacios de Hilbert que están definidos como los espacios que tienen producto interno definido y su base es completa - toda sucesión de Cauchy de elementos del espacio convergen a un elemento que está en el mismo espacio -, a los coeficientes de transformación se les llama en general Coeficientes de Fourier y son el producto interno de la base con la función a transformar. Las bases de estos espacios son un conjunto de funciones, las cuales son ortonormales entre sí.

1.5 RECAPITULACIÓN.

Lo que se ha visto es que la aplicación de teorías físicas ha dado como resultado una visión totalmente nueva del problema que consiste en abstraer el funcionamiento del cerebro a redes de espines y pensar en el aprendizaje como funciones que mapean un espacio de eventos a un espacio de asociaciones además de que han aportado soluciones a problemas prácticos [Simpson, 1993].

CAPITULO SEGUNDO

LAS REDES NEURONALES EN EL CONTEXTO DE LA ESTADISTICA DESCRIPTIVA

Se revisarán los conceptos básicos que definen a una RNA como son la arquitectura, el algoritmo de aprendizaje y la interpretación geométrica. En particular se analizarán todos estos aspectos en la arquitectura de Cascada (ver sección 1.1.12). Se estudiarán los teoremas de Funahashi [Funahashi, 1989] que aseguran que una red en cascada es un mapeador universal, se analizarán sus consecuencias geométricas y sus relaciones con el espacio de Hilbert L^2. Finalmente, con la ayuda de las propuesta de Amit [Amit, 1989] y Peretto [Peretto, 1992] para las funciones de activación de los nodos y una arquitectura permitida por Funahashi, se reinterpretará este tipo de RNA como sistemas estadísticos.

2.1 REDES NEURONALES ARTIFICIALES (RNA).

Primero se definirá la neurona artificial y luego los conjuntos de ellas llamados las Redes Neuronales Artificiales (RNA). Después se verán sus funciones de activación, características, formas de aprender y de recuperar la información.

2.1.1 La Neurona Artificial y las RNA.

La neurona artificial (también llamada nodo, procesador) es un sistema que consta de dos partes, un sumador y una función de activación (Fig. 2.1).

Cada nodo, por medio de la función de activación, genera una salida real. La salida del nodo se multiplica por un valor real llamado peso, que generalmente se designa con la letra w. Si se quiere simular un potencial inhibitorio generalmente el peso será negativo y si se quiere un potencial excitatorio el peso será positivo.

Las conexiones entre nodos se crearan por lazos, en estos lazos fluirá la información. Así, se puede simular una red neuronal por una representación de nodos y lazos, esto

es, una gráfica (ver Fig. 1.7).

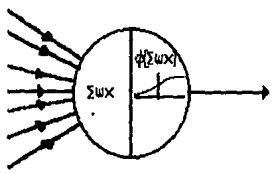


Fig. 2.1 Una neurona artificial. Ejerce dos funciones, calcular una combinación lineal de las salidas de otras neuronas y hacer un mapeo a una función de activación.

Por lo anterior se puede hacer la siguiente definición de Red Neuronal Artificial: conjunto de nodos y lazos. Cada nodo puede recibir información (estar conectado) de todos los demás nodos (inclusive de sí mismo). En cada nodo, se llevan a cabo dos acciones: una combinación lineal de los valores provenientes de todos los nodos que tienen conexión con el nodo específico y el mapeo de esa combinación por medio de una función de activación. La combinación lineal consta en multiplicar el valor de la salida de cada nodo por un número particular (peso) y después sumar todos los resultados de todas las conexiones. La salida del nodo será el valor de la función, esto es, lo que se transmitirá ese nodo a los demás nodos.

2.1.2 Las funciones de activación.

Las funciones de activación, mapean la entrada en una salida. Como ejemplo de funciones de activación se tienen Lineal (Fig. 2.2):

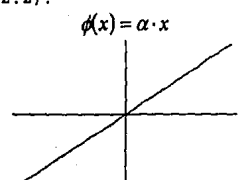


Fig. 2.2 La función lineal.

Rampa (Fig. 2.3):

$$\phi(x)=\begin{cases}+\gamma \text{ si } x\geq\gamma\\ x \text{ si } |x|<\gamma\\ -\gamma \text{ si } x\leq-\gamma\end{cases}$$

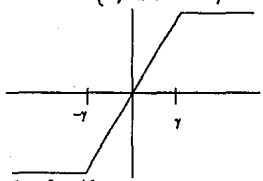


Fig. 2.3 La función rampa, como caso límites tiene la función lineal y la escalón.

Escalón (Fig. 2.4):

$$\phi(x)=\begin{cases}+\gamma \text{ si } x>0\\ -\delta \text{ de cualquier otra manera}\end{cases}$$

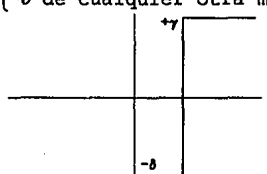


Fig. 2.4 La función escalón, el escalón puede estar centrado en cualquier parte del eje de las abscisas.

Sigmoidal: Son funciones continuas, acotadas, derivables y monótonamente crecientes. De este tipo de funciones se usa comúnmente la llamada función logística (ver Fig. 2.6)

$$\phi(x)=\frac{1}{1+e^{-x}},$$

otro ejemplo de funciones sigmoidales es la tanh.

2.1.3 Características topológicas.

Como topología de una red neuronal artificial se entenderá su arquitectura o la disposición de los nodos en un espacio

hipotético. Se toma en cuenta para esta discusión que toda RNA puede representarse en forma de capas de nodos [Simpson, 1990].

Hay dos tipos principales de conexiones, excitatorias e inhibitorias. Las conexiones excitatorias incrementan la activación y son usualmente representadas por señales positivas. Las conexiones inhibitorias decrementan la activación y son normalmente representadas por señales negativas.

Las capas que reciben información del exterior se llaman capas de entrada, las que mandan información al exterior se llaman capas de salida. Las que están entre éstas dos son llamadas capas ocultas (Fig. 2.5).

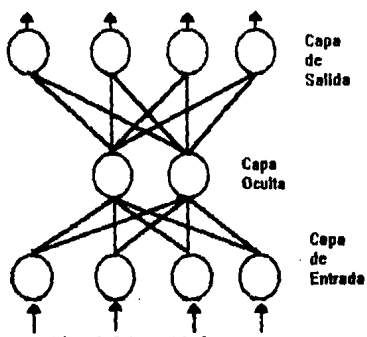


Fig. 2.5 Una RNA de tres capas.

2.1.4 Representación Matemática de una RNA.

Tomando en cuenta las figuras 2.1 y 2.5 y la función de activación de un nodo, se puede escribir que la salida de uno de estos es

$$f_i(x) = \phi\left(\sum_{j=1}^{a} w_{ij}x_j - \theta_i\right), \qquad 2.0$$

con a siendo el numero de nodos de la capa de entrada, x_j el valor del j-ésimo nodo de la capa de entrada, ϕ la función sigmoidal, y θ es un valor de umbral.

El valor de los nodos de la capa de entrada queda

determinado por el patrón de entrada, esto es, la información que recibe la red en forma de números.

Tomando la red de la Fig. 2.5, se puede pensar un patrón de entrada como X=(0,1,0,5). La primera entrada de este vector corresponde al primer nodo de la capa de entrada, la segunda entrada al segundo nodo y así sucesivamente. Ahora viendo la Ec. 2.0, cada nodo -i- de la capa intermedia recive el valor de la multiplicación de la señal de cada nodo de la capa de entrada por el peso de conexion entre los nodos $-w_{ij}x_j$. Se puede afectar esta suma por un valor de umbral θ_i.

Es claro que se pueden hacer composiciones de estas funciones y así se obtiene el procesamiento de la señal de entrada capa por capa.

2.1.5 Aprendizaje.

En el contexto de las RNA, aprendizaje se debe de entender como el poder encontrar una matriz de pesos -W- que satisfaga las asociaciones que se quieran hacer. Como en el caso de la red en la Fig. 2.5, se puede crear la asociación del patrón de entrada (0,0,0,1) con la salida (1,0,0,0).

Para realizar el aprendizaje se toma un grupo de datos, el cual se divide en dos conjuntos. El primer conjunto es el de entrenamiento y otro el de prueba. El conjunto de entrenamiento sirve hacer que la red cree las asociaciones, en el caso de retropropagación el est conjunto consta de los ejemplos que se quieren aprender y sus asociaciones respectivas (salida deseada). El conjunto de prueba son ejemplos que no se usaron para la etapa de entrenamiento, con ellos se verifica que la red aprendió.

Uno de los puntos que se deben de recalcar es el que una RNA se puede entrenar para que aprenda primero un patrón y después otro, pero esto no significa que haya olvidado el primer patrón, este es el punto fundamental del poder que tienen las RNA.

Los métodos de aprendizaje pueden ser clasificados en dos ramas: aprendizaje supervisado y no-supervisado, aunque características de ambos pueden coexistir. El aprendizaje supervisado es un proceso que incorpora un maestro externo y/o información global, esto es, que existe un agente externo al sistema y al algoritmo que indica a aquel que tan bien va su actuación. Técnicas de supervisado incluyen

decidir cuando terminar el aprendizaje, decidir cuanto y que tan seguido presentar cada asociación para entrenar, y proporcionar un error de actuación de la red. El no-supervisado se subdivide en otras dos categorías: aprendizaje estructural y aprendizaje temporal. El aprendizaje estructural codifica el mapeo autoasociativo o heteroasociativo a W, que es la matriz de pesos. El temporal codifica una secuencia de patrones necesarios para adquirir una salida final. Ejemplos de algoritmos supervisados son el de corrección del error, retropropagación, contrapropagación, reforzamiento, estocástico. No supervisados también se refieren a auto-organizado.

2.1.6 Generalización.

Una vez entrenada una RNA se le pueden presentar patrones difusos. Se definirá como patrón difuso a un patrón que sea parecido a alguno de los patrones que sirvieron de entrenamiento. Esto es, si un patrón de entrenamiento fué (1,0,0,0), un patrón difuso será (0.8,0.2,0.4,0.23). Ahora, si la asociación para el patrón de entrenamiento fué (1,0,0,0), o sea, el mismo patrón (que no es en general) la salida para el patrón difuso será algo parecido a esto (0.8,0.1,0.1,0). Esto es, los patrones difusos son clasificados a la asociación más cercana a ellos. A esta propiedad se le llama Generalización.

2.2 LA ARQUITECTURA Y LA GEOMETRÍA DE LAS RNA TIPO CASCADA.

Se verán las características principales que definen una RNA tipo cascada que son la arquitectura y la geometría inducida por el mapeo que realizan. Sin embargo estos mismos principios se aplican a todas la RNA.

2.2.1 La arquitectura.

La disposición espacial de los elementos de una RNA y las conexiones entre los elementos de esta se le llama arquitectura de la red. Una arquitectura y el algoritmo de aprendizaje definen completamente el comportamiento y mapeo ejercido por una RNA.

La arquitectura de las redes en cascada consiste usualmente de una capa de entrada y otra de salida, con un número variable de capas intermedias - inclusive ninguna. La

información fluye de capa en capa, de la entrada hacia la salida.

Las funciones de activación que se usan en los nodos son sigmoidales, que se definen como aquellas que son acotadas, continuas, derivables y monótonamente crecientes. En especial se usa la función logística cuya forma es

$$f(x)=\frac{1}{1+e^{-x}}, \qquad 2.1$$

y su derivada

$$f'(x)=f(x)(1-f(x)), \qquad 2.2$$

obteniendo así una gráfica de la siguiente forma (Fig. 2.6),

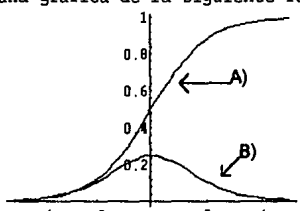


Fig. 2.6 A) La función sigmoidal, B) Su derivada.

por lo que es una función fácil de computar y de codificar en la mayoría de los lenguajes de programación. Una de las justificaciones de usar funcines logísticas se puede ver en 1.4.1. A esta función logística se le pueden agregar otros factores en el exponente

$$f(x) = \frac{1}{1+e^{\left(\frac{x}{\varepsilon}+\theta\right)}}, \quad \varepsilon > 0, \quad \theta \in \Re \qquad 2.3$$

estos harán que se desplace sobre el eje x o que se incremente la derivada hasta que en el caso límite la función quedará como

$$Lim_{\varepsilon \to 0} f(x) = \begin{cases} 0 \ si \ x<0 \\ 1 \ si \ x>0 \end{cases}, \qquad 2.4$$

y la derivada

$$Lim_{\varepsilon \to 0} f'(x) = \delta(0), \qquad 2.5$$

obteniendo así una delta centrada en 0.

El cambio de los pesos se dará de acuerdo con la ley de aprendizaje que se defina.

2.2.2 La geometría.

Recordar que en el capítulo uno, la propuesta de Cooper era una combinación de las entradas. De cierta manera, esta combinación lineal, modifica a la red, así obteniendo una salida generalmente distinta al patrón de entrada. Esto es un mapeo, así pues, es factible analizar a las RNA geométricamente ya que se puede considerar a un evento como un vector - gráfica - o una matriz - imagen. Todos estos elementos pertenecen a espacios con geometrías determinadas.

Una RNA mapea un evento de un espacio E - eventos - a un espacio A - abstracciones. La pregunta que surge es la de qué tipo de espacio se está usando. En principio se debe de tomar lo más general, un espacio de Banach, que es un conjunto de elementos en el cual toda sucesión de Cauchy converge a un elemento en el mismo espacio. Ahora bien, se debe hacer notar, que si un sistema sensa a sus alrededores dos eventos distintos al mismo tiempo, v.g. oír y ver, estos también deberán ser distintos en su abstracción. Esto es, los eventos deben de estar separados de cierta forma. Por lo tanto, se puede afirmar que también es necesaria una topología - un espacio con norma definida. Por otro lado, también se desea caracterizar un conjunto de objetos por sus propiedades generales, esto es, agruparlos por la semejanza con respecto a otros - como un gato persa es un gato. Así que también se pueden utilizar una base - las abstracciones - y si se propone la condición de que esta base también sea completa, esto es, que todo elemento del espacio puede ser representado por una combinación de la base. Se tienen pues, todos los elementos para afirmar que el espacio que se está buscando es un espacio de Hilbert cuya definición formal es el de un espacio con producto interno definido y con una base completa. Decir que la base debe de ser completa es afirmar que el lema de Reiz se cumple, esto es, las funciones definidas sobre el espacio son separables sobre la base, v.g. una serie de Fourier.

Un par de características arquitectónicas y geométricas que diferencían a una RNA de otra es la forma del producto interno definido o inducido por el mapeo creado por la RNA y la flexibilidad de la dimensión que se está usando, esto es, si la dimensión del espacio de abstracciones puede cambiar en el tiempo de aprendizaje.

El producto interno es una función C a C. Sean u, v, w

funciones complejas y a una constante compleja, entonces el producto interno se define como ($\bar{}$ significa complejo conjugado),

$$\begin{aligned} &(u,v)=\overline{(v,u)} \\ &(u,u)\geq 0 \\ &(u,u)=0 \Leftrightarrow u=0 \\ &(au,v)=\bar{a}(u,v) \\ &(u,v+w)=(u,v)+(u,w) \end{aligned} \qquad 2.6$$

Ejemplos de productos internos son

a) El producto punto de $\Re^3$

Sean $x_1,\ x_2 \in \Re^3$, $(x_1,x_2)=\sum_i x_1^i x_2^i$

b) El espacio L^p.

Sean f, g funciones medibles tales que $\int|f|^p<\infty$, $\int|g|^p<\infty$, entonces $(f,g)=\int|fg|^p$.

c) La distancia de Hamming.

Sean $x_1,\ x_2 \in (0,1)^n$, $(x_1,x_2)=\sum_i x_1^i x_2^i$, que es el número de unos en el que se diferencían los dos vectores.

El concepto de distancia se da con base en el producto interno. Generalmente se usa la idea Euclidiana de que entre dos puntos se toma la hipotenusa que forman la intersección de las coordenadas de estos y se designa por el siguiente símbolo.

$$\|\|_g$$

quedando los ejemplos anteriores como

$$\|x_1,x_2\|_{\Re^3}=\sqrt{\sum_i\left(x_1^i-x_2^i\right)^2},$$

$$\|f,g\|_{L^p}=\left(\int|fg|^p\right)^{1/p},$$

$$\|x_1,x_2\|_{H^n}=\sqrt[1/n]{\sum_i x_1^i x_2^i},$$

2.2.3 Las redes en cascada con algoritmo de retropropagación.

Un algoritmo de entrenamiento para las redes en cascada es el llamado de Retropropagación (en inglés Backporpagation, BP) y que se mencionó en el Capítulo Primero. Actualmente es uno de los algortimos más usados [Simpson, 1990].

Los algoritmos BP conllevan un problema en el mapeo, ya que lo que se crea es un mapeo rígido, esto es, que si el mapeo está definido como

$$f:\Gamma^m \to \Omega^n,$$

a m y la n están fijas. Por lo tanto, aunque se haya enseñado el conjunto de elementos que tiene su mapeo como

$$\{e_i\} \in \Gamma^m \xrightarrow{f} \{a_i\} \in \Omega^n,$$

se puede insertar un e_{i+1} que activará el sistema y sufrirá una transformación al espacio Ω aún y cuando no tenga representación en este espacio, esto es en símbolos

$$e_{i+1} \in \Gamma^m \xrightarrow{f} a_i \in \Omega^n \text{ para alguna i}.$$

Como ejemplo, se usará la red de la Fig. 2.5 que se le enseñaron a hacer la asociacion de patrones de entrada (e) con los de salida (s),

$$e_1 = (1,0,0,0) \quad s_1 = (1,0,0,0)$$

$$e_2 = (0,0,0,1)' \, s_2 = (0,0,0,1)$$

La red para el patrón 1 será

$$f(e_1) = \phi\left(\sum_{i=1}^{2} C_i \phi\left(\sum_{j=1}^{2} w_{ij} - \theta_i\right)\right),$$

y para el patrón 2

$$f(e_2) = \phi\left(\sum_{i=1}^{2} C_i \phi\left(\sum_{j=3}^{4} w_{ij} - \theta_i\right)\right),$$

Si se presenta a la misma red el siguiente patrón

$$e_3 = (0,1,1,0),$$

que es totalmente distinto a los otros dos se obtendrá

$$f(e_3) = \phi\left(\sum_{i=1}^{2} C_i \phi\left(\sum_{j=2}^{3} w_{ij} - \theta_i\right)\right),$$

esto es, se tiene una respuesta aunque el patrón no pertenezca a ninguna de las clases predeterminadas.

En las redes auto-organizadas este problema es menor ya que el tamaño de la n (del espacio de asociaciones) puede cambiar.

2.3 LOS TEOREMAS DE FUNAHASHI.

La idea de un mapeo ha sido bastante tratada por las personas interesadas en RNA. Es así que surgen un par de teoremas que demuestran que todo mapeo continuo se puede aproximar por una RNA con arquitectura en Cascada - y posiblemente creada con el algoritmo BP. Las condiciones para esto son que la capa de entrada sea lineal, las intermedias - ocultas -, sigmoidales y de salida lineales o sigmoidales, estos son los Teoremas de Funahashi [Funahashi, 1988].

2.3.1 Los Teoremas.

Teorema 1. Sea $\phi(x)$ una función no constante, acotada y monótonamente creciente. Sea K un subconjunto compacto en $\Re^n$ y $f(x_1,\ldots,x_n)$ una función de valores reales continua en K. Entonces dada $\varepsilon>0$, existe un entero N y constantes reales c_i, $\theta_i (i=1,\ldots,N)$, $w_{ij} (i=1,\ldots,N,\ j=1,\ldots,n)$ tal que

$$\bar{f}(x_1,\ldots,x_n)=\sum_{i=1}^{N} c_i \phi\left(\sum_{j=1}^{n} w_{ij} x_j-\theta_i\right) , \qquad 2.7$$

satisface que

$$max_{x\in K}\left|f(x_1,\ldots,x_n)-\bar{f}(x_1,\ldots,x_n)\right|<\varepsilon . \qquad 2.8$$

En otras palabras, para una $\varepsilon>0$, existe una red de tres capas cuyas funciones de salida para la capa oculta son $\phi(x)$ y cuyas funciones para las capas de entrada y salida son lineales, siendo la función de transferencia de la red $\bar{f}(x_1,\ldots,x_n)$ tal que $max_{x\in K}\left|f(x_1,\ldots,x_n)-\bar{f}(x_1,\ldots,x_n)\right|<\varepsilon$.

Este Teorema 1 se generaliza en el teorema 2.

Teorema 2. Sea $\phi(x)$ una función no constante, acotada y monótonamente creciente. Sea K un subconjunto compacto en $\Re^n$ y un entero fijo $k\geq 3$. Entonces para cualquier mapeo continuo $f:K\rightarrow\Re^m$ definido por $x=(x_1,\ldots,x_n)\rightarrow(f_1(x),\ldots,f_m(x))$ puede aproximarse en el sentido de una topología uniforme sobre K por mapeos de entrada-salida de redes de k-capas cuyas funciones para las capas ocultas son $\phi(x)$, y cuyas funciones de salida para capas de entrada y salida son lineales. En otras palabras, para cualquier mapeo continuo

$f:K\to\Re^m$ y una $\varepsilon>0$ arbitraria, existe una red de k-capas cuyo mapeo de entrada-salida está dado por $\tilde{f}:K\to\Re^m$ tal que $max_{x\in K}d(f(x),\tilde{f}(x))<\varepsilon$, donde $d(,)$ es una métrica que induce la topología usual de $\Re^m$.

Corolario 1. Sea $\phi(x)$, K como definida arriba y un entero $k\geq 3$. Entonces para cualquier mapeo $f:\bar{x}\in K\to(f_1(\bar{x}),...,f_m(\bar{x}))\in\Re^m$ donde $f_i(\bar{x})(i=1,...,m)$ son sumables sobre K, puede ser aproximado en el sentido de la topología L^2 sobre K por mapeos entrada-salida de redes k-capas (capas ocultas k-2) cuyas funciones de salida para las capas ocultas son $\phi(x)$ y cuyas funciones de salida para capas entrada y salida son lineales. En otras palabras, para cualquier $\varepsilon>0$, existe una red de k-capas cuyo mapeo entrada-salida está dado por $\tilde{f}:\bar{x}\in K\to(\tilde{f}_1(\bar{x}),...,\tilde{f}_m(\bar{x}))\in\Re^m$ tal que

$$d_{L^2(K)}(f,\tilde{f})=\left(\sum_{i=1}^{m}\int_K\left|f_i(x_1,...,x_n)-\tilde{f}_i(x_1,...,x_n)\right|^2 d\bar{x}\right)^{1/2}<\varepsilon\ . \qquad 2.9$$

Corolario 2. Sea K definida como arriba y $k\geq 3$. Sea $\phi(x)$ una función estrictamente continua y creciente tal que $\phi((-\infty,\infty))=(0,1)$. Entonces cualquier mapeo continuo $f:K\to(0,1)^m$ puede ser aproximado en el sentido de topología uniforme sobre K por mapeos de entrada-salida de $k(\geq 3)$-capas de redes neuronales cuyas funciones de salida para capas ocultas y de salida sean $\phi(x)$.

2.3.2 Esbozo de la demostración de los teoremas de Funahashi.

En esta sección no se dará una demostración detallada de la demostración de los teoremas de Funahashi [Funahashi, 1989], sino que se mostrarán los lemas y teoremas en que se basó y los pasos cruciales de la demostración.

Lema 1. Sea $\phi(x)$ una función acotada, continua y monótonamente creciente. Para $\alpha>0$ tenemos que

$$g(x)=\phi(x/\delta+\alpha)-\phi(x/\delta-\alpha), \qquad 2.10$$

tenemos que $g(x)\in L^1(\Re)$ y el valor de su transformada de

Fourier $G(\xi)$ en $\xi=1$ es distinto de cero. En otras palabras g es absolutamente integrable.

Teorema de Irie-Miyake. Sea $\psi(x)\in L^1$ y $f(x_1,...,x_n)\in L^2(\Re^n)$. Sean $\Psi(\xi)$ y $F(w_1,...,w_n)$ sus transformadas de Fourier respectivamente y si $\Psi(1)\neq 0$ entonces

$$f(x_1,...,x_n)=\int_{-\infty}^{\infty}...\int_{-\infty}^{\infty}\psi\left(\sum_{i=1}^{n}x_iw_i-w_0\right)\frac{1}{(2\pi)^n\Psi(1)}F(w_1,...,w_n)exp(iw_0)dw_0dw_1...dw_n \qquad 2.11$$

a) Sean

$$I_A(x_1,...,x_n)=\int_{-A}^{A}...\int_{-A}^{A}\psi\left(\sum_{i=1}^{n}x_iw_i-w_0\right)\frac{1}{(2\pi)^n\Psi(1)}F(w_1,..,w_n)exp(iw_0)dw_0dw_1...dw_n \qquad 2.12$$

$$I_{\infty,A}(x_1,...,x_n)=\int...\int\left[\int\psi\left(\sum x_iw_i-w_0\right)\frac{1}{(2\pi)^n\Psi(1)}F(w_1,...,w_n)exp(iw_0)dw_0\right]dw_1...dw_n \qquad 2.13$$

$$J_A(x_1,...,x_n)=\frac{1}{(2\pi)^n}\int\int F(w_1,...,w_n)exp\left(i\sum x_iw_i\right)dw_1...dw_n, \qquad 2.14$$

donde $\psi(x)\in L^1$ es definida por

$$\psi(x)=\phi(x/\delta+\alpha)-\phi(x/\delta-\alpha), \qquad 2.15$$

para alguna α y δ tal que satisfagan el lema 1.
Obs:

$$\int\psi\left(\sum x_iw_i-w_0\right)exp(iw_0)dw_0=exp\left(i\sum x_iw_i\right)\Psi(1). \qquad 2.16$$

Se puede demostrar que

$$lim_{A\to\infty}J_A(x_1,...,x_n)=f(x_1,...,x_n), \qquad 2.17$$

y que

$$lim_{A\to\infty}I_{\infty,A}(x_1,...,x_n)=f(x_1,...,x_n), \qquad 2.18$$

esto es, que para $\varepsilon>0$ existe $A>0$ tales que

$$max_{x\in\Re^n}\left|I_{\infty,A}(x_1,...,x_n)-f(x_1,...,x_n)\right|<\varepsilon/2, \qquad 2.19$$

b) Para $A'>0$ se define $I_{A',A}$

$$I_{A',A}(x_1,...,x_n)=\int...\int\left[\int\psi\left(\sum x_iw_i-w_0\right)\frac{1}{(2\pi)^n\Psi(1)}F(w_1,...,w_n)exp(iw_0)dw_0\right]dw_1...dw_n$$

2.20

Se aproxima $I_{\omega,A}$ por estas integrales finitas sobre K, esto es que

$$max_{x\in K}\left|I_{A',A}(x_1,...,x_n)-I_{\omega,A}(x_1,...,x_n)\right|<\varepsilon/2, \qquad 2.21$$

y por lo anterior,

$$max_{x\in K}\left|f(x_1,...,x_n)-I_{A',A}(x_1,...,x_n)\right|<\varepsilon. \qquad 2.22$$

c) En otras palabras f(x) puede ser aproximada por una integral finita $I_{A',A}(x)$ uniformemente en K. El integrando de $I_{A',A}(x)$ puede ser reemplazado por la parte real

$$\Re\left(I_{A',A}\right)=\int...\int\left[\int\psi\left(\sum x_i w_i-w_0\right)\frac{1}{(2\pi)^n\psi(1)}F(w_1,...,w_n)\cos(w_0)dw_0\right]dw_1...dw_n, \qquad 2.23$$

por ser una función continua sobre K se puede aproximar uniformemente sobre K por una suma de Riemann.

$$R^N_{I_{A,A}}=\frac{(2A)^n}{N^{n+1}}\frac{2A'}{(2\pi)^n\Psi(1)}\sum\sum\left[\psi\left(\sum x_i\left(-A+\frac{k_1 2A}{N}\right)-\left(-A'+\frac{k_0 2A'}{N}\right)\right)\right.$$
$$\left.F\left(-A+\frac{k_1 2A}{N},...,-A+\frac{k_n 2A}{N}\right)Cos\left(-A+\frac{k_0 2A'}{N}\right)\right]. \qquad 2.24$$

Tenemos

$$R^N_{I_{A,A}}=C_1\sum\left[\psi\left(\sum x_i\left(-A+\frac{k_1 2A}{N}\right)-\left(-A'+\frac{k_0 2A'}{N}\right)\right)\right.$$
$$\left.F\left(-A+\frac{k_1 2A}{N},...,-A+\frac{k_n 2A}{N}\right)Cos\left(-A+\frac{k_0 2A'}{N}\right)\right]. \qquad 2.25$$

Como

$$\psi\left(\sum x_i w_i-w_0\right)=\phi\left(\sum w_i x_i/\delta-w_0+\alpha\right)-\phi\left(\sum w_i x_i/\delta-w_0-\alpha\right) \qquad 2.26$$

$\Rightarrow$

$$R^N_{I_{A,A}}=C_1\sum\left[\phi_{+\alpha}\left(\sum x_i\left(-A+\frac{k_1 2A}{N}\right)-\left(-A'+\frac{k_0 2A'}{N}\right)+\alpha\right)\right.$$
$$\left.-\phi_{-\alpha}\left(\sum x_i\left(-A+\frac{k_1 2A}{N}\right)-\left(-A'+\frac{k_0 2A'}{N}\right)-\alpha\right)F\left(-A+\frac{k_1 2A}{N},...,-A+\frac{k_n 2A}{N}\right)Cos\left(-A+\frac{k_0 2A'}{N}\right)\right] \qquad 2.27$$

En notación simplificada

$$R^N_{1_{A,A}} = C_1 \sum \phi_{+\alpha} F(\bar{k}) Cos(k_0) - C_1 \sum \phi_{-\alpha} F(\bar{k}) Cos(k_0), \qquad 2.28$$

se puede expandir la sumatoria y luego renumerar constantes obteniendo

$$R^N_{1_{A,A}} = \sum_j B_j \phi_{+\alpha}\left(\sum_i x_i w_i - w_j\right) - \sum_j B_j \phi_{-\alpha}\left(\sum_i x_i w_i - w_j\right), \qquad 2.29$$

que por lo anterior se afirma que

$$f(\bar{x}) = \sum_j B_j \phi_{+\alpha}\left(\sum_i x_i w_i - w_j\right) - \sum_j B_j \phi_{-\alpha}\left(\sum_i x_i w_i - w_j\right), \qquad 2.30$$

esto quiere decir, que cualquier función puede ser aproximada por RNAs tipo cascada de 3-capas con funciones de activación en las capas de entrada y salida lineales y en la capa intermedia sigmoidal.

Además, como $\exists\, g(x), h(x)$ tales que $f(\bar{x}) = g(\bar{x}) + h(\bar{x})$, que heredan las propiedades de $f(\bar{x})$, entonces se puede decir que

$$f(\bar{x}) = g(\bar{x}) + h(\bar{x}) \approx \sum_j B_j \phi_{+\alpha}\left(\sum_i x_i w_i - w_j\right) - \sum_j B_j \phi_{-\alpha}\left(\sum_i x_i w_i - w_j\right) \qquad 2.31$$

entonces, sin pérdida de generalidad se puede definir,

$$\begin{aligned} g(\bar{x}) &\approx \sum_j B_j \phi_{+\alpha}\left(\sum_i x_i w_i - w_j\right), \\ h(\bar{x}) &\approx -\sum_j B_j \phi_{-\alpha}\left(\sum_i x_i w_i - w_j\right) \end{aligned} \qquad 2.32$$

∴ Cualquier función continua, acotada y con soporte compacto puede ser aproximada por una RNA de 3-capas.

El teorema 2 se demuestra por inducción: se aplica a una RNA de varias capas el teorema 1 repetidamente.

El corolario 1 y 2 hablan del espacio donde las funciones se encuentran y sus características métricas (la topología). Así pues, no sólo se está en un espacio de Hilbert sino que las funciones que crean el mapeo se encuentran en un espacio L^2. El corolario 2 es una particularidad que extiende las posibilidades de hacer una RNA con capa de salida sigmoidal.

Obs1:

Teoría de la medida.

Afirmar que las funciones que se usan en RNA pertenecen al espacio L^2 - corolario 1 - permite usar todos los

teoremas de teoría de la medida e integral de Lesbegue. El espacio L^2 se define como el conjunto de relaciones de equivalencia de funciones que son cuadrado integrable, en símbolos

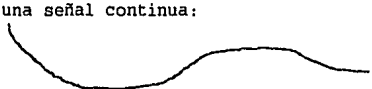
$$\int f < \infty.$$

Un teorema en especial de la teoría de la integral de Lesbegue nos puede ayudar para entender las RNA:

Teorema: Sean f y g funciones iguales para casi todo punto (p.p.) entonces

$$\int f = \int g.$$

Decir que son iguales p.p. es decir que el conjunto de puntos que difieren entre f y g tiene medida cero. Así que si se tiene una señal continua:



y se le suma ruido de medida cero las integrales serán las mismas.

Ya se ha visto que una RNA es un sistema que debe de tolerar errores - ruido -, se puede afirmar que ésta actúa como un eliminador de los errores encontrando el patrón partiendo de la señal de entrada. También se puede entender como un sistema que elimina el conjunto de puntos de medida cero que difieren entre el paradigma y la señal presentada. Esto es claro para funciones continuas, sin embargo, para señales discretas - las que inevitablemente se tienen que usar en una computadora - se tendrá que proponer un criterio para afirmar cuando un ruido es de medida cero o no.

Obs2:

Todo lo anterior, sólo se refiere al estado final que se obtiene de entrenar una RNA de capas - cascada -, no se está hablando del algoritmo de aprendizaje, este puede ser cualquiera. Lo único que se aseguran estos teoremas es que existe la matriz de pesos W para cualquier función que se quiera que el sistema aprenda. Tampoco habla del tiempo requerido para realizar este mapeo.

2.4 LA INTERPRETACIÓN FÍSICA-ESTADÍSTICA.

En los análisis antes descritos no se encuentran justificaciones biológicas o analogías físicas que sustenten

las hipótesis planteadas, a excepción de la regla de Hebb. La única base es la empírica, esto es, es sabido que las funciones sigmoidales sirven. En el trabajo original de Rumelhart et al. es una proposición basada en la propuesta de McCulloch y Pitts en el sentido en que un disparo está o no está - función escalón - que demostró sus limitaciones.

Antes de hacer cualquier interpretación se tiene que construir un modelo. Para hacer un modelo se tienen que asumir ciertas hipótesis y utilizar resultados anteriores. En este trabajo se harán las siguientes hipótesis y tomaremos en cuenta algunos hechos biológicos.

hipótesis

arquitectura → Cascada-Funahashi.

aprendizaje → Hebb

hechos

probabilidad → Se sabe que los procesos de redes neuronales son probabilísticos.

En otras palabras, se usará una RNA tipo cascada, con la capa de entrada lineal, las intermedias y de salida sigmoidal. Se tendrá en cuenta el carácter probabilístico de la actividad neuronal.

2.4.1 De los nodos.

Se puede considerar a los nodos como partículas que tienen dos estados de energía

$$S_i = \begin{cases} 1 \text{ cuando dispara} \\ 0 \text{ cuando no dispara} \end{cases}$$

En otras palabras una partícula tiene dos estados: el base y el excitado.

Si la interacción de estas partículas contiene ruido desembocará en un proceso estocástico, entonces se hace necesario hablar de la probabilidad de una partícula de estar en un estado o en el otro.

Se usará la discusión de Amit y Peretto sobre modelado de redes neuronales biológicas simplificando la neurona hasta el punto de ser una partícula de dos estados - como los ya descritos. Los procesos que, bajo ciertas condiciones experimentales especiales, influyen para que el fenómeno biológico sea estocástico son:

I. El número de vesículas descargadas en la llegada de un potencial de acción varía aleatoriamente bajo una

probabilidad de Poisson con un valor medio J_{ij}(de la neurona i a la j).

II. Aún en la ausencia de un potencial de acción existen liberación aleatorias espontáneas de neurotransmisores químicos en el espacio sináptico (cuanta).

III. El tamaño de los cuantos puede variar sobre un intervalo largo, pero la media y variancia de sus distribuciones es igual en todas las sinapsis. Las contribuciones de cada cuantum al potencial postsináptico está dado por una probabilidad de densidad Gaussiana, independientemente de i y j.

Por estas afirmaciones, la función densidad de probabilidad (P) para que el potencial (voltaje) obtenga el valor U es [Amit, 1989]

$$P(U_i = U) = \frac{1}{\sqrt{2\pi\delta^2}} exp\left[-\frac{(U-\bar{U}_i)^2}{2\delta^2}\right] \qquad 2.33$$

donde $\bar{U}_i$ es la media de U_i, y δ es el ancho de la distribución que depende de los parámetros asociados con las diferentes fuentes de ruido mencionadas. La probabilidad de que la neurona i dispare un potencial de acción es igual a la probabilidad de que su potencial de membrana sea mayor que el valor de umbral T_i, esto es,

$$P(S_i = 1) = \int P(U_i = U) = \frac{1}{2}\left[1 + erf\left(\frac{\bar{U}_i - T_i}{\delta\sqrt{2}}\right)\right] \qquad 2.34$$

donde S_i, es el estado de la partícula (+1 o 0), erf(x) queda definida como la función de error la cual tiene la forma

$$erf(x) \equiv \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \qquad 2.35$$

así se obtiene una función sigmoidal mucho más difícil de computar que la logística.

Automáticamente queda definida la probabilidad de que no dispare,

$$\Pr(S_i = 0) = 1 - \Pr(S_i = 1) = \frac{1}{2}\left[1 - erf\left(\frac{\bar{U}_i - T_i}{\delta\sqrt{2}}\right)\right] \qquad 2.36$$

Recapitulando, se tiene una partícula en la que su estado de actividad está determinado por las interacciones que tiene con las demás partículas, los estados que puede tomar es 1 y 0. El primero dice que la partícula reacciona a

los impulsos que recibe de otras partículas y el segundo dice lo contrario.

2.4.2 De la red.

Ahora se verá el funcionamiento de varios nodos interconectados bajo la arquitectura BP.

En una RNA-Cascada, lo que se mide es la respuesta de los nodos de la capa de salida. En general, uno se encuentra en el caso de no saber lo que se esta presentando como patrón de entrada a la red. Por otro lado, se ha visto que cada nodo es una partícula, la cual tiene una cierta probabilidad de encontrarse en cierto estado debido a las interacciones con otras partículas. Supóngase que se está usando de nuevo la red de la Fig. 2.5 y que ha sido entrenada para las siguientes asociaciones (las e son entrada y las s salida)

e1=(1,0,0,0)	s1=(1,0,0,0)
e2=(0,1,0,0)	s2=(0,1,0,0)
e3=(0,0,1,0)	s3=(0,0,1,0)
e4=(0,0,0,1)	s4=(0,0,0,1)

Entonces, cuando se presente el patrón e1 se obtendrá s1, y esto es siempre. Se puede decir se ha entrenado a la red para que la probabilidad de que clasifique a e1 como s1 es 1 y la probabilidad de clasificarlo en cualquier otra clase (los demás nodos) es cero.

Se ha afirmado que las funciones sigmoidales miden la probabilidad de que el nodo genere un pulso, si se crea una red con estos nodos la probabilidad se conservará. Los nodos de la capa de salida medirán la probabilidad de clasificar a un patrón determinado en alguna clase previamente aprendida.

Como ya se vió en la sección 2.2.3, la red independientemente del tipo de patrón de entrada, generará una salida. Esto significa, que la probabilidad total de clasificar un patrón de entrada es 1 (o alguna normalización). Por lo tanto, como los nodos de la capa de salida no interactúan entre ellos, la probabilidad total de la capa de salida es la suma de las funciones de cada nodo que deben de sumar 1.

Se puede presentar un patrón difuso, por ejemplo (0.8,0.2,0.05,0,0.1) que se parece a e1, entonces se obtendrá una salida del tipo (0.8,0.1,0.05,0.05), esto es, la probabilidad de que perteneza a la clase que corresponde

al primer nodo de la capa de salida es 0.8 y así sucesivamente. Además, si se suman los componentes de la salida debe de dar 1.

Estas afirmaciones concuerdan en el resultado con la propuesta de White [White, 1989] , sin embargo el punto de vista es que aquí se usa la probabilidad de reacción de un conjunto de partículas a un impulso dado y él asevera que las RNA emulan la experiencia del observador del fenómeno.

En el siguiente capítulo se demostrará que las RNA-Cascada son sistemas estadísticos y que la probabilidad de reacción de la capa de salida está normalizada.

Así pues, se tiene que la matriz de eventos es afectada por la capa de entrada en el sentido que sus componentes van a ser objeto de una transformación lineal siendo esto lo que se presentará a la capas intermedias. Se puede pues entender a la capa de entrada como el transductor o los sensores del sistema que miden el ambiente que lo rodea, la linealidad de la transformación nos habla de que los eventos se van a transformar isomórficamente y que el orden entre estos se va a conservar.

En las capas intermedias actúan las funciones sigmoidales descritas arriba y surge la interpretación probabilística. Se puede decir que lo que pasa es que los nodos tienen una cierta probabilidad de reaccionar de cierta manera al presentarse una entrada, esto estará directamente relacionado con la fuerza sináptica, o sea el peso que se le den a las entradas. Es así que la suma de los impulsos que llegan a un nodo tienen una posibilidad de que la información fluya por ese camino, sin embargo, si es menor, esta inhibirá. Es claro que la probabilidad de que un nodo reaccione a una señal dada es independiente de los demás nodos de su capa, ya que los nodos de la capa k+1 procesan la suma de las probabilidades de los nodos de la capa k.

Por otro lado, si se sigue la recomendación de Funahashi, en el corolario 2, se pueden implantar redes con capa de salida sigmoidal. Entonces se puede interpretar esta capa como la medida de la probabilidad de clasificar a una señal en una clase predefinida - que el sistema aprendió.

CAPITULO TERCERO

VISION ESTADISTICA DE LAS REDES EN CASCADA

Se verá la comparación de la función de error con la función logística. Los resultados de esta comparación demuestran que la función de error es mejor que la función logística. Además se comprueba computacionalmente que las redes tipo cascada son sistemas estadísticos.

3.1 COMPARACIÓN DE REDES CON FUNCIÓN LOGÍSTICA Y CON FUNCIÓN DE ERROR.

Se comparan RNA-Cascada entrenadas con el algoritmo de retropropagación, una red implantada con la función sigmoidal de error (FE) Erf[x/2] - vista en el capítulo tercero -, la otra con la función logística, 1/(1+exp[-x]) (FL). Se demuestra que se requieren menos iteraciones para llegar al mismo nivel de error. La tolerancia o generalización a patrones difusos, es casi igual. Se demuestra que las RNA-Cascada son sistemas estadísticos.

3.1.1 Introducción.

Como se vió en el capítulo anterior, el usar la FE ayuda a interpretar las redes en cascada como sistemas estadísticos. Se comparará esta FE con la FL. También se vio que las redes en cascada siempre reaccionan a cualquier impulso, independientemente si fueron entrenadas para identificar esos impulsos o no, esto quiere decir que la probabilidad total del sistema a reaccionar a un impulso dado es siempre 1.

Se pueden tomar los valores de la capa de salida como un vector y entrenar la red para que cada conjunto de patrones que se quiera que aprenda pertenezca a un vector de salida ortonormal a los demás vectores de salida. Lo que resulta en una matriz diagonal. Como los nodos de la capa de salida no interactúan entre sí, la probabilidad de cada uno de ellos se suma. Así se obtiene la probabilidad total del sistema que como se dijo debe de ser 1. Si se da el caso que

las salidas no sean ortogonales es fácil pasar a la forma ortonormalizada. Tener las salidas ortonormalizadas significa crear un mapeo en el que el espacio de entradas de m dimensiones es proyectado al de asociaciones de n dimensiones.

Los problemas que se van a usar para comparar el desempeño de las redes son los que usualmente se usan para este propósito: DECODIFICADOR y XOR, el primero trata de que la salida sea igual a la entrada y el segundo es la función lógica.

3.1.2 Arquitectura y algoritmo de aprendizaje.

El argumento de la FE es x/2, de otra manera su derivada sería muy grande y se acercaría al caso determinista, y el de la FL es x. Ambas funciones se normalizaron entre (-1,1), los patrones son conjuntos binarios que si por ejemplo son (-1,-1) con salida 1 tomarán los valores (-0.9,-0.9) y (0.9) o que es lo mismo entre (0,1) serán 0.05 y 0.95, por lo tanto se obtendrá que la probabilidad total no será 1 sino 0.95, esto se hace por motivos computacionales, ya que si obligáramos a la red a alcanzar el 1 o el -1 ésta tendría que tener un infinito en su argumento por la naturaleza de las funciones probabilísticas. Se utilizó un nodo bias [Rumelhart el al., 1986] para la capa de entrada y otro para la intermedia. Se modificó la regla delta como lo propone Rumelhart para acelerar, esto significa menos interaciones para llegar al error deseado, la convergencia de aprendizaje

$$\Delta_p w_{ji} = \eta \delta_{pj} o_{pi} + \alpha w_{ji}$$

donde w es el peso entre el nodo j y el i, o es la salida real del nodo i para el p-ésimo patrón, δ es la función delta (definida en 1.2.12), α es un valor entre 0 y 1 llamado momentum.

Por definición una época es la presentación de una serie de patrones igual al numero de patrones que se quiera que la red aprenda, generalmente esta presentación será aleatoria.

Se hicieron 4 series con 40 condiciones iniciales distintas para cada problema, cada serie tiene una arquitectura distinta. La etapa de entrenamiento consistió de 200 épocas y el criterio de convergencia para ambas redes fue la obtención de un error estable abajo del 0.01 y

en menos de 200 iteraciones. La función de error se define como

$$\frac{1}{2N}\sum_{i=1}^{N}\|t_i - o_i\|^2$$

donde t es la asociación a aprender, O es la salida de la red y N el número de elementos del patrón de entrada.

No todas las series cumplieron las condiciones planteadas, se tuvo que tomar un criterio en el que si más de quince resultados de cada serie no convergía la serie se descartaba, esto se tuvo que hacer por motivos de la estadística, ya que lo que se está comparando es la convergencia de los dos tipos de redes.

3.1.3 Entrenamiento de las redes.

Problema del DECODIFICADOR: El problema es hacer que la salida sea igual a la entrada, con el número de nodos de la entrada igual a una potencia de 2, el número de nodos de la capa intermedia igual a la potencia y el número de nodos de la salida igual que la entrada (Fig. 3.1). Los vectores de entrenamiento son (p de entrenamiento y o de asociación)

p1=(-0.95,-0.95,-0.95,0.95)	o1=(-0.95,-0.95,-0.95,0.95)
p2=(-0.95,-0.95,0.95,-0.95)	o2=(-0.95,-0.95,0.95,-0.95)
p3=(-0.95,0.95,-0.95,-0.95)	o3=(-0.95,0.95,-0.95,-0.95)
p4=(0.95,-0.95,-0.95,-0.95)	o4=(0.95,-0.95,-0.95,-0.95)

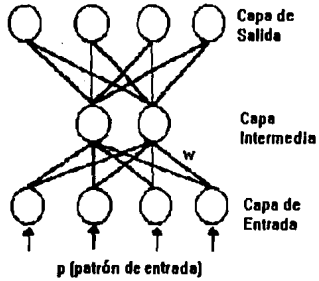


Fig. 3.1 Arquitectura del problema decodificador. La entrada es igual que la salida.

Los parámetros de cada serie son

DA η=0.9 y α=0.0

DB η=0.9 y α=0.2

DD η=0.2 y α=0.5

Los resultados del entrenamiento se encuentra en la tabla 3.1. Se representan las series DA, DB y DD, la DC se descartó. En cada conjunto de cuatro columnas se encuentra el número del experimento, el número de iteraciones que requirió la FE en converger, el número de épocas que requirió la FL y la semilla de azar que se utilizó para la inicialización de los pesos (ver Apéndice A).

Tabla 3.1

Serie: DA				Serie: DB				Serie: DD			
No	FE	FL	Az	No	FE	FL	Az	No	FE	FL	Az
1	60	90	91	1	34	62	86	1	113	192	9
2	50	78	182	2	43	57	172	3	117	191	27
3	42	82	273	3	33	52	258	4	136	187	36
4	39	75	364	4	29	52	344	6	158	164	54
6	42	70	546	5	33	55	430	7	88	162	63
7	37	80	637	6	32	52	516	8	97	150	72
8	117	18	728	7	24	50	602	9	109	175	81
9	90	99	819	8	30	57	688	10	77	148	90
10	39	67	910	10	32	58	860	12	84	144	108
12	43	72	1092	11	33	59	946	13	108	163	117
13	40	68	1183	12	29	49	1032	14	77	138	126
14	114	159	1274	14	32	65	1204	15	80	137	135
15	45	83	1365	15	53	67	1290	16	95	152	144
16	46	76	1456	16	49	84	1376	18	83	159	162
17	37	69	1547	17	33	65	1462	19	100	144	171
18	37	73	1638	18	35	62	1548	20	82	158	180
19	40	72	1729	19	30	70	1634	22	81	152	198
20	46	72	1820	20	50	84	1720	23	89	141	207
21	57	82	1911	21	38	71	1806	24	86	151	216
23	44	122	2093	22	48	85	1892	27	77	145	243
24	47	125	2184	24	28	50	2064	28	70	146	252
25	144	120	2275	25	27	49	2150	29	135	146	261
26	38	67	2366	27	37	60	2322	34	87	152	306
28	182	184	2548	28	34	52	2408	35	80	142	315
29	82	199	2634	29	26	49	2494	36	86	152	324
30	42	71	2730	30	80	70	2580	38	119	197	342
31	34	62	2821	31	42	55	2666	39	85	152	351
33	42	72	3003	32	33	56	2756	40	92	187	360
34	190	82	3094	34	35	58	2924				
35	41	64	3185	36	37	68	2096				
36	67	94	3276	37	27	51	3182				
37	40	68	3367	38	29	52	3268				
38	33	61	3458	39	42	67	3354				
39	37	66	3549	40	35	55	3440				

De la serie DA: 33 de las 36 muestras fue mejor la FE, se anularon 6 muestras, de las cuales 2 ninguna de las dos redes convergieron y 3 sólo convergió la FL.

De la serie DB 35 de las 35 muestras, se anularon 5 muestras, de las cuales 1 ninguna de las dos redes convergieron y 4 sólo convergió la FL.

De la serie DD 28 de las 28 muestras, se anularon 11 muestras, de las cuales 3 ninguna de las dos redes convergieron y 3 sólo convergió la FL.

Las gráficas de 3.2a a 3.2h muestran como varía el error promedio cuadrado entre la red implementada con la FE y la implementada con la FL contra el número de épocas.

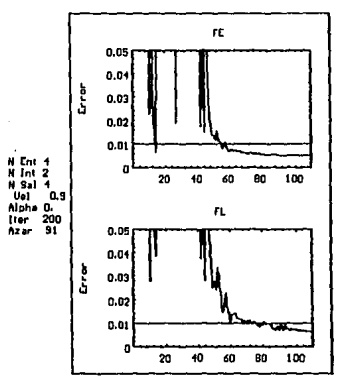


Fig. 3.2a Figura de Da1. Aunque la FE entra en la zona menor a 0.01 en menos de 20 iteraciones no lo hace establemente. Lo que se busca es estabilidad como ocurre después de 60 iteraciones.

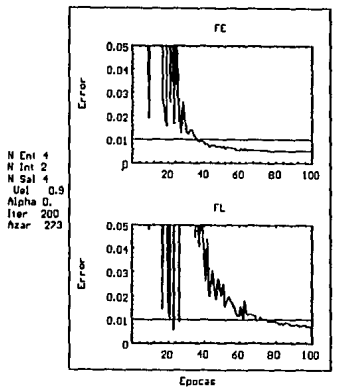


Fig. 3.2b Figura de Da3. Ambas son parecidas, sin embargo es más suave la FE y casi la mitad de iteraciones que la FL.

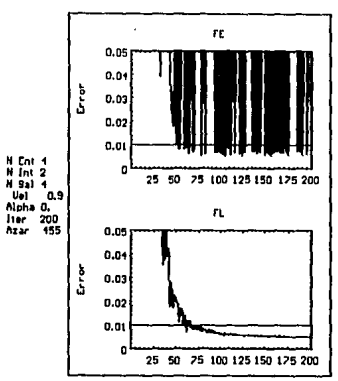


Fig. 3.2c Figura de Da5. En algunos pocos caso la FE no converge y la FL sí. No se usó para la estadística ninguna de las dos.

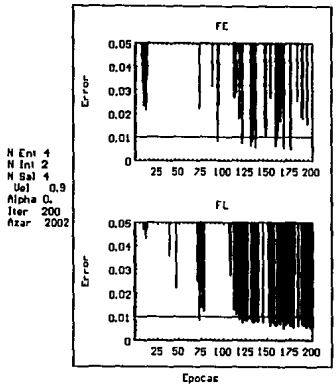


Fig. 3.2d Figura de Da22. Ninguna de las dos converge. No se usó para la estadística.

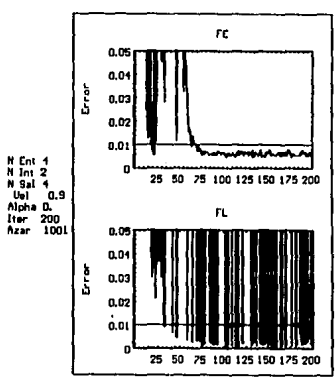


Fig. 3.2e Figura de Da11. La PL no converge, en cambio la PE lo hace en menos de 80 iteraciones. No se usó para la estadística.

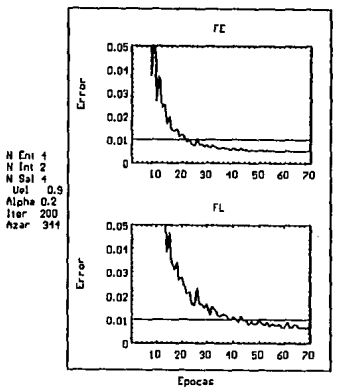


Fig. 3.2f Figura de Db4. Las dos gráficas son muy parecidas, sin embargo, es más suave la FE.

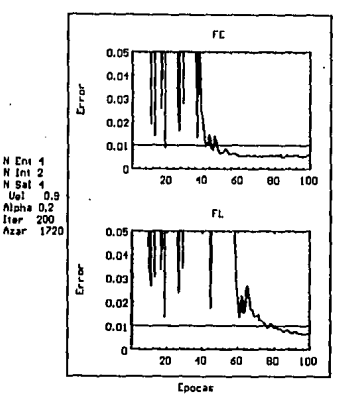


Fig. 3.2g Figura de Db20. La convergencia en la FE es más abrupta que en la FL.

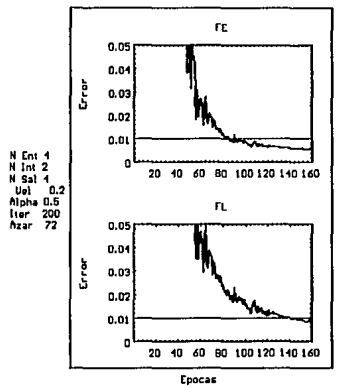


Fig. 3.2h Figura de Dd8. La entrada a la zona de convergencia es casi la mitad más grande en la FL que en la FE.

En las gráficas 3.2 se aprecia que algunas redes no convergen, esto significa que no para cualquier conjunto inicial de pesos la red converge, además la convergencia depende de la arquitectura (número de nodos en cada capa y el valor de la velocidad y el momentum).

Problema del XOR: Dadas las variables binarias, entrenar la red para que en la salida se obtenga un 0 si los números son distintos y un 1 si son iguales. Por motivos de graficación se hizo que cuando las variables tuvieran el mismo signo la salida fuera -1 y 1 de lo contrario. También se normalizó entre (-1,1). La arquitectura usada fueron dos nodos de entrada, dos en la capa intermedia y uno de salida [Rumelhart et al, 1986]. Los resultados se encuentran en la Tabla 3.2

p1=(-0.95,-0.95)	s1=-0.95
p2=(-0.95, 0.95)	s2= 0.95
p3=(0.95,-0.95)	s3= 0.95
p4=(0.95, 0.95)	s4=-0.95

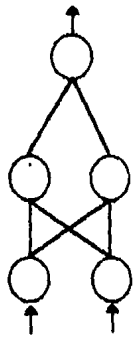


Fig. 3.3 Arquitectura usada en el problema XOR.

Los parámetros para las series son los siguientes:

XA η=0.9 y α=0.0

XB η=0.9 y α=0.3

Tabla 3.2

Serie XA				Serie XA				Serie XB				Serie XB			
No	FE	FL	Az	No	FE	FL	Az	No	FE	FL	Az	No	FE	FL	Az
1	16	31	345	22	50	62	7590	1	20	30	856	27	22	33	23112
2	30	47	690	23	19	32	7935	2	54	76	1712	28	25	52	23968
3	37	60	1035	24	28	42	8280	3	34	53	2568	29	46	53	24824
5	40	55	1725	26	40	46	8970	5	46	69	4280	30	39	51	25680
6	22	34	2070	27	31	48	9315	6	20	27	5136	34	38	170	29104
7	18	32	2415	28	42	59	9660	11	12	21	9416	35	41	59	29960
8	20	32	2760	29	64	82	10005	12	25	30	10272	36	37	43	30816
9	33	46	3105	30	33	52	10350	13	43	48	11128	37	69	64	31672
12	25	40	4140	32	48	64	11040	14	24	32	11984	38	37	105	32528
13	42	53	4485	33	25	46	11385	17	61	71	14552	39	25	34	33384
14	20	33	4830	34	32	55	11730	18	29	36	15408				
15	31	50	5175	35	44	66	12075	19	96	59	16264				
17	36	55	5865	37	56	91	12765	20	40	52	17120				
18	25	40	6210	38	40	54	13110	24	65	98	20544				
20	55	65	6900	39	37	52	13455	25	24	35	21400				
22	22	38	7245					26	24	32	22256				

De la XA 31 de las 31 muestras fue mejor la FE, se anularon 9 muestras, de las cuales 7 ninguna de las dos redes convergieron y 2 sólo convergió la FE.

De la XB 24 de las 26 muestras fue mejor la FE, se

anularon 14 muestras, de las cuales 6 ninguna de las dos redes convergieron y 2 sólo convergió la FE.

Las Figuras 3.4a a 3.4h nos muestran algunas comparaciones entre la redes implementadas con la FE y las implementadas con la FL.

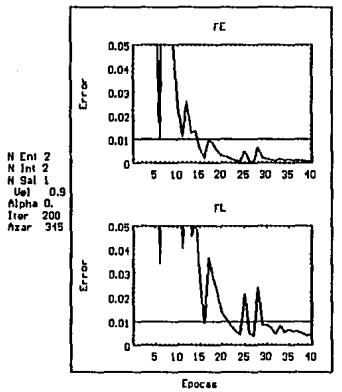


Fig. 3.4a Figura de Xa1. La convergencia de la red con FE es la mitad de iteraciones que la red con FL.

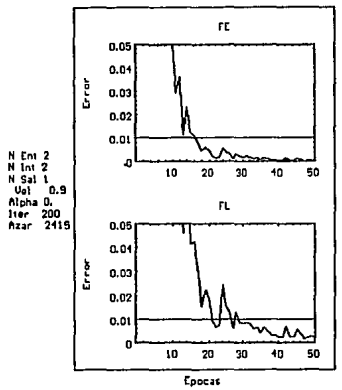


Fig. 3.4b Figura de Xa7. Ambas figuras son parecidas, sin embargo la FE es más rápida y más suave.

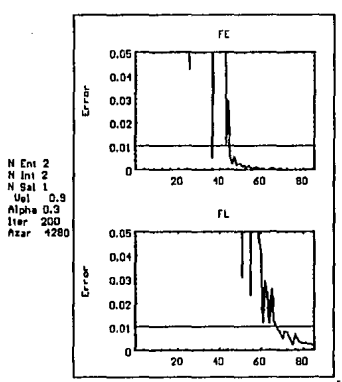


Fig. 3.4c Figura de Xb5. Esta gráfica muestra dos figuras distintas, en este caso la red con FE converge abruptamente, en cambio la red con FL lo hace de una manera suave.

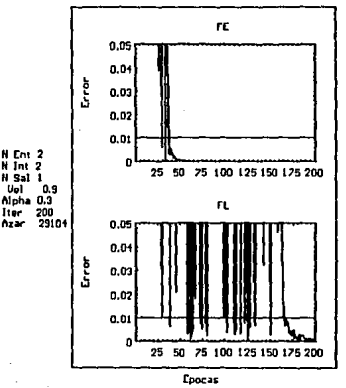


Fig. 3.4d Figura de Xb34. En este caso la convergencia de la red con FE es casi cuatro veces más rápida que la de la red con FL.

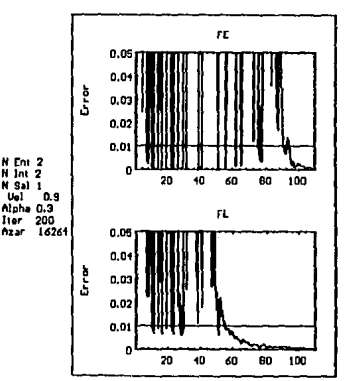


Fig. 3.4e Figura de Xb19. Las dos convergen pero es mejor la red con FL.

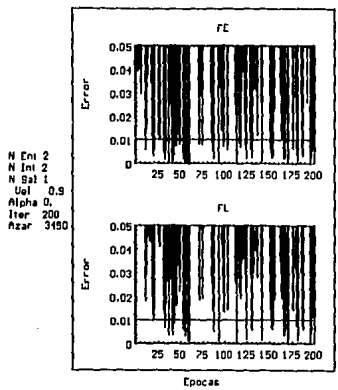


Fig. 3.4f Figura de Xa10. Ninguna de las dos converge. No se tomó en cuenta para la estadística.

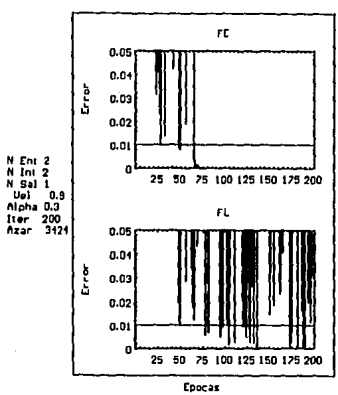


Fig. 3.4g Figura de Xb4. Sólo la FE converge, es interesante que su convergencia es muy rápida, en cambio la FL no converge. No se tomó en cuenta para la estadística.

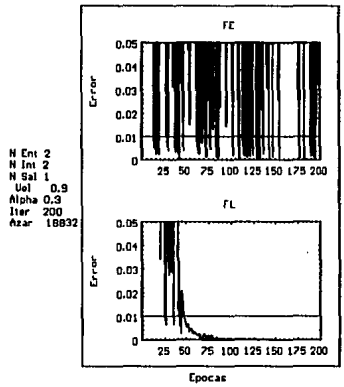


Fig. 3.4h Figura de Xb22. Sólo la FL converge, es la inversa de la figura anterior. No se tomó en cuenta para la estadística.

En este conjunto de figuras se nota los mismo que en la Fig. 3.3, algunas redes no convergen en absoluto, esto nos puede hablar de un carácter probabilístico del comportamiento de la red.

3.1.4 Comprobación experimental de que la red es un sistema estadístico.

En el capítulo dos se explicó que una RNA tipo cascada es un sistema estadístico. Por otro lado, en las secciones 2.3.1 y 3.1.1 se afirma que la capa de salida es una medida de la probabilidad de que la red clasifique a un patrón presentado como perteneciente a una clase aprendida. En la sección anterior se analizó el desempeño de dos tipos de funciones de activación, se usará la FE, dado su mejor desempeño a comparación con la FL, para demostrar las afirmaciones hechas en 3.1.1.

Ya White ha asegurado que todas las RNA son sistemas estadísticos [White, 1989] pero en el sentido que simulan la experiencia del experimentador, no como se está viendo en este trabajo.

En el caso de que las salidas de las asociaciones aprendidas sean ortogonales (como en los problemas del DECODIFICADOR y el XOR). Las partículas de la capa de salida

no interactúan entre ellas (ya que es una arquitectura de cascada), por tanto, la probabilidad total de encontrar ese conjunto de partículas en un estado es la suma directa de las pobabilidades individuales de cada partícula. Por lo visto en 2.3.1 y 3.1.1 la suma de esta capa debe de ser igual para cualquier patrón presentado, en estos casos 0.95 dada la forma de entrenar a las redes.

En el caso del problema XOR tenemos sólo dos tipos de salidas, uno o menos uno, entonces la red tiene una cierta probabilidad de clasificar a un patrón dentro de una clase predeterminada. Si se suma la probabilidad del nodo de salida y la probabilidad de no clasificarlo (1-probabilidad de clasificar) se tendrá siempre un uno (en este caso 0.95).

En el problema DECODIFICADOR se presentaron 50 patrones difusos a la red y se hizo un promedio. Se espera que la salida sea también difusa pero que clasifique correctamente. En la tabla 3.3 se muestran los resultados del experimento.

Tabla 3.3

Red	media50 FE	media50 FL	Red	media50 FE	media50 FL
da1	0.930932	0.932651	db1	0.942561	0.947674
da2	0.929935	0.939933	db2	0.952017	0.942991
da3	0.918203	0.953701	db3	0.962953	0.94362
da4	0.913658	0.928461	db4	0.917618	0.929275
da6	0.950314	0.956775	db5	0.931792	0.968073
da7	0.901332	0.925728	db6	0.937404	0.922118
da8	0.886959	1.07634	db7	0.919667	0.935787
da9	0.898097	1.04432	db8	0.926844	0.965775
da10	0.937858	0.944015	db10	0.944268	0.938735
da12	0.938867	0.970981	db11	0.963092	0.951883
da13	0.928797	0.942803	db12	0.931178	0.927193
da14	0.986742	1.17519	db14	0.932182	0.978438
da15	0.928553	0.928146	db15	0.910827	1.02716
da16	0.94579	0.931577	db16	0.922015	0.958793
da17	0.931601	0.926896	db17	0.961718	1.01015
da18	0.956138	0.928957	db18	0.92635	0.922409
da19	0.92619	0.908452	db19	0.93443	1.03366
da20	0.938647	0.964992	db20	0.927825	0.91023
da21	0.978299	1.05061	db21	0.962311	0.96126
da23	0.9146	0.931645	db22	0.947461	1.15139
da24	0.957748	1.24547	db24	0.937982	0.93014
da25	0.876748	0.906892	db25	0.928933	0.937825
da26	0.928261	0.942935	db27	0.944541	0.934703
da28	0.935022	0.929248	db28	0.959087	0.95412
da29	1.05624	0.932005	db29	0.951292	0.967915
da30	0.918321	0.948168	db30	0.972417	1.01196
da31	0.942262	0.933904	db31	0.921026	0.924524
da33	0.936171	0.939042	db32	0.919868	0.938018
da34	0.908731	0.93136	db34	0.94965	0.972365
da35	0.926342	0.953221	db36	0.927221	0.932978

da36	0.964355	1.01231	db38	0.923714	0.951417
da37	0.929944	0.943359	db39	0.931357	0.99717
da38	0.924734	0.935	db40	0.941393	0.945374
da39	0.912845	0.936045			
Prom	0.934087	0.96620976	Prom	0.962827	0.96143
StDv	3.15x10-2	7.34x10-2	StDv	1.586x10-2	4.6015x10-2
Red	media FE	media FL	Red	media FE	media FL
dd1	1.05673	1.0468	dd19	0.908401	0.910935
dd3	0.860107	0.914119	dd20	0.934588	0.927203
dd4	1.0475	0.992894	dd22	0.928153	0.92876
dd6	0.848608	0.896743	dd23	0.959545	0.977217
dd7	0.919174	0.942359	dd24	0.91062	0.926144
dd8	0.9442	0.960063	dd27	0.941533	0.941837
dd9	1.05243	1.01405	dd28	0.93845	0.949614
dd10	0.939774	0.934084	dd29	0.945315	0.951538
dd12	0.917788	0.952665	dd34	0.93872	0.927009
dd13	0.985034	0.976151	dd35	0.941689	0.95233
dd14	0.917979	0.937616	dd36	0.93923	0.974263
dd15	0.926793	0.927143	dd38	1.0835	1.04207
dd16	0.94321	0.958299	dd39	0.932696	0.912865
dd18	0.918532	0.925403	dd40	0.986902	0.991115
			Prom	0.9488285	0.95326032
			StDv	5.4357×10^{-2}	3.7583×10^{-2}

Todas las desviaciones estándar incluyen el 0.95. Cada una de las pruebas tuvo un error promedio cuadrado menor a 0.01, esto es, en promedio el reconocimiento de patrones fue satisfactorio. Se puede pues afirmar, que las RNA tipo cascada, en estos casos (DECODIFICADOR y XOR), sí son sistemas estadísticos.

3.1.5 Pruebas de generalización.

Una vez entrenada la red se prueba su desempeño en la clasificación de patrones difusos. Para esta prueba es más sencillo usar el problema XOR, ya que se puede representar como dos variables independientes y la salida como la variable dependiente. El problema DECODIFICADOR por su naturaleza es mucho más difícil de visualizar. En el problema del XOR se tiene el patrón (0.95,-.95) con salida 0.95, un patrón difuso sería (0.8,-0.5) con salida 0.90.

Los que se obtuvieron concuerdan con los ya obtenidos por Pao [Pao, 1989] que graficó las curvas de nivel a intervalos iguales para el problema XOR obteniendo que el gradiente no era constante (Fig. 3.5). El entrenamiento se hizo con la salida entre 0 y 1, y con el siguiente conjunto de asociaciones

(0,0) con salida 1
(1,1) con salida 1
(0,1) con salida 0
(1,0) con salida 0

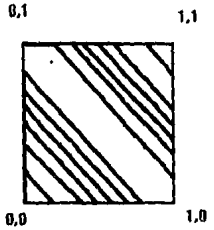


Fig. 3.5 La figura que hizo Pao es basada en un entrenamiento con la salida contraria a la que se utilizó aquí (-1,-1 y 1,1 -> 1), es por eso que la figura tiene distinta orientación a las que se presentan aquí.

El resultado no parece el esperado, intuitivamente las curvas de nivel deberían de formar un símbolo de +, ya que se puede pensar que (0.5, 0.6) debería de ir al 0 y (0.6, 0.6) al 1. Sin embargo, se tiene que recordar que la red entrenó bajo el criterio de mantener un cierto valor de error debajo de un umbral, así que se puede tomar como una aproximación.

Generando una cuadrícula con las dos variables entre -1 y 1 se obtiene una superficie que describe la generalización de la red. Las figuras 3.6a a 3.6e se presentan dos formas de ver los datos. La primera es la graficación de las dos redes y la segunda es el cáculo de las curvas de nivel (la implementada con la FE y la FL). Es interesante que nunca se cruzan la superficies. También se muestran las curvas de nivel de ambas superficies, son curvas de nivel a cada 0.163 unidades entre -1 y 1. Todas las pruebas tuvieron un error promedio cuadrado menor a 0.01.

Dado que las asociaciones se hicieron al revés, esto es, cuando las entradas eran iguales la salida era -1 y cuando eran distintas era 1, las gráficas estarán rotadas, exceptuando esto toda la discusión anterior sigue siendo válida.

Fig. 3.6a: Se puede ver que las superficies están cerca una de la otra, sin embargo en las curvas de nivel, la red FL converge más rápido a la zona de (-1,1) que en la misma zona de la FE, pero en general están iguales. Esto indica que localmente la FL se comporta mejor que la FE. Las Fig. 3.6a es una respuesta típica en todos los experimentos.

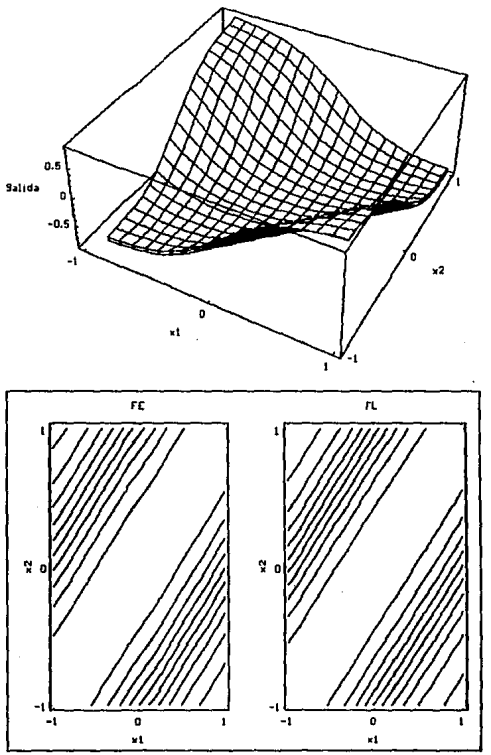


Fig. 3.6a Figuras de Xa1.

Fig. 3.6b: Ambas superficies son casi iguales en el valle - alrededor de la diagonal que va de (1,1) a (-1,-1). En cambio en la región de (-1,1) y (1,-1) se separan, aunque no simétricamente, esto puede ser debido a que la red aprendió muy bien sólo tres patrones.

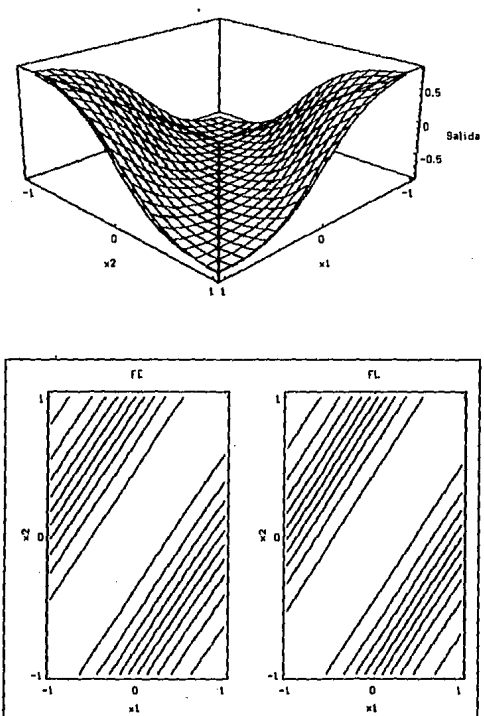


Fig. 3.6b Figuras de Xa2. Al igual que la figura anterior la semejanza es mucha.

3.6c: Se observa una gran separación entre ambas superficies. Esto se ve reflejado en la curva de nivel de la FE en la región (-1,1).

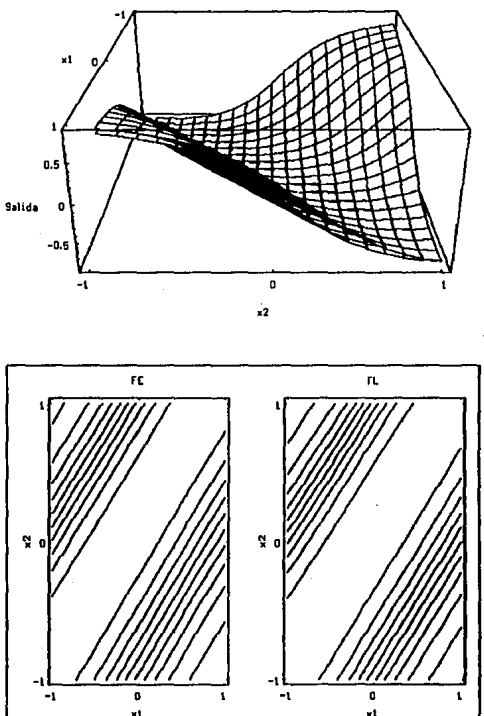


Fig. 3.6c Figuras de Xal3. En la figura tridimensional se nota una diferencia que no es fácil ver en las curvas de nivel. La gráfica que está abajo es la de la red con FE.

Fig. 3.6d: Aunque se nota una separación entre las superficies las curvas de nivel son muy parecidas.

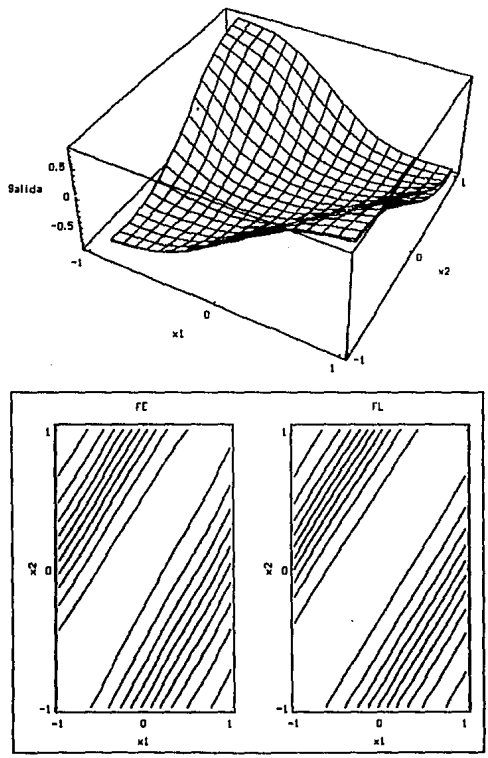


Fig. 3.6d Figuras de Xa28. En la esquina superior derecha de la red con FE se nota un adelgazamiento de la franja que está entre -1 y -0.8.

Fig. 3.6e: Las curvas de nivel de la FE en su parte central son más angostas que ls de la FL.

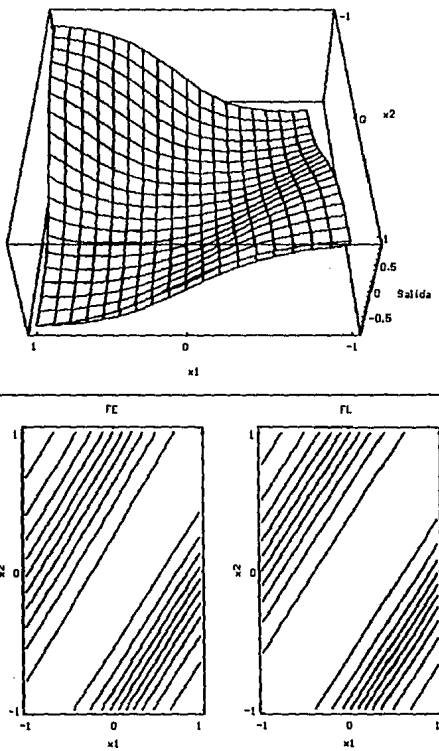


Fig. 3.6e Figuras de Xb38. En este caso la diferencia es grande cuando nos acercamos a la región de (-1,-1).

Las gráficas 3.6 dicen que el sistema es estable ante la presentación de patrones difusos -ruidosos-, esto es, no hay cambios abruptos en la respuesta del sistema, además, hay que recordar que las redes entrenadas con la FE aprenden en menos épocas que las entrenadas con la FL.

CAPITULO CUARTO

REDES ESTADISITICAS INVARIANTES A TRANSLACION DEL PATRON DE ENTRADA Y EJEMPLO DE APLICACION

Se introducirá el concepto de redes con invariancia a la translación para después hacer una aplicación a reconocimiento de señales eléctricas provenientes de hipocampo de rata.

4.1 REDES EN CASCADA CON INVARIANCIA A LA TRANSLACION.

Después de la invención del algoritmo BP varios investigadores hasta la fecha se han abocado a modificarlo y analizarlo. Una de las modificaciones que se ha realizado es el de la invariancia a la translación de la señal de entrada. Se presentará el uso de una propuesta de invariancia a la translación [Espinosa y Quiza, 1987], implantándola con la FE, en lugar de con la FL como ellos hicieron.

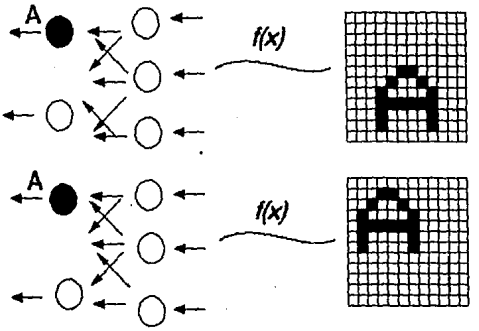
4.1.1 Invariancia a la translación (IT).

Invariancia a la translación significa que si se entrena la red con el patrón A en el centro de la pantalla, al presentarlo a la izquierda de ésta la red lo seguirá clasificando como el patrón tipo A (véase la Fig. 4.1).

A continuación se verán las diferencias entre la arquitectura clásica y la que causa invariancia bajo translación (IT) de redes BP.

4.1.2 Arquitectura estándar versus arquitectura particular de IT (EQ)).

Una de las propiedades arquitectónicas de una RNA-BP clásica es que no hay restricción en las conexiones entre nodos de capas contiguas. Esto es, el nodo i de la capa k puede estar conectado con pesos distintos de cero a cualquier nodo - j - de la capa k+1 o k-1, (vease la Fig. 4.2).



Salida Entrada

Fig. 4.1 Una de las metas de las RNA es presentar invariancia a la translación. El patrón A es presentado a una red e independientemente de su posición en la malla la probabilidad de que la red siga clasificándolo como patrón A es mayor que clasificarlo en algún otra clase aprendida.

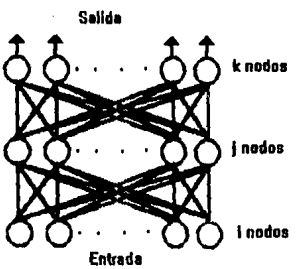


Fig. 4.2 Las RNA clásica no tienen restricciones para que un nodo de una capa se comunique con todos los demás de la siguiente.

Actualmente, no existe una red que sea invariante a cualquier tipo de patrón, por tanto, se han creado redes partículares para cada problema específico. La propuesta que se presenta, sirve para clasificar patrones continuos (como una señal). Esta propuesta para la invariancia [Espinosa y Quiza, 1987] se basa en la restricción de las conexiones entre nodos de capas contiguas, esto es, no todos los nodos de la capa k van a estar conectados con cada uno de los de la capa k+1 (o k-1), llamaremos a esta red la red EQ. Como hay restricciones, es necesario que surjan nuevas variables que definan el sistema, así que se pueden definir nuevas variables como el barrido y traslape. El barrido es el número de nodos conectados a cada nodo de la capa inmediata superior, y el traslape es el número de nodos que están conectados a dos de esa capa (ver Fig. 4.3).

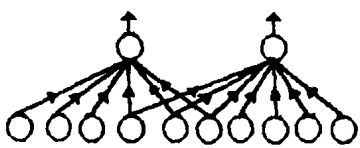


Fig. 4.3 Una propuesta para lograr RNA-BP-IT es restringir las conexiones entre capas contiguas.

Se puede definir grupo como el conjunto de nodos de la capa inmediata que están conectados al mismo conjunto de nodos de la capa anterior. El número de grupos queda determinado por

$$\frac{N-B}{B-T}+1=G$$

excepto para B=T, donde N, número de nodos de entrada; B, número de nodos barridos; T, nodos de traslape y G, número de grupos. Es claro que no toda combinación dará un número entero de grupos.

Es fácil ver que la red EQ cumple todos y cada uno de los teoremas de Funahashi, por lo tanto esta arquitectura también garantiza mapeos universales. Podemos pues, postular este tipo de arquitectura como un corolario más a los teoremas de Funahashi -el tercero. Formalizando:

Corolario 3:

Sea $f_o(x) \in K \subset \Re$, K compacto, $\phi(x)$ sigmoidal y $\varepsilon > 0$, entonces $\exists \quad \{c_i\}_1^n, \quad \{\theta_i\}_1^n, \quad \{w_{ji}\}_{1,1}^{m,n}$ tales que $f(x) = \phi\left(\sum_{i=1}^{n} c_i \phi\left(\sum_{j=1}^{a_1} w_{ji} x_j + .. + \sum_{j=a_p}^{m} w_{ji} x_j\right)\right)$ y $max_{x \in K} |f - f_o| < \varepsilon$, con $a_1 < a_2 < .. < a_m$. En otras palabras, toda función definida sobre un compacto sobre los reales puede ser aproximada por una RNA de 3 capas, con la capa en entrada lineal y la intermedia y de salida sigmoidal.

Demostración:

Del teorema 1 (sección 2.3)

$$\tilde{f}(x_1, ..., x_n) = \sum_{i=1}^{N} c_i \phi\left(\sum_{j=1}^{n} w_{ij} x_j - \theta_i\right) \qquad 2.7$$

que por el corolario 2

$$\phi(\tilde{F}(x_1, \ldots, x_n)) = \phi\left(\sum_{i=1}^{N} c_i \phi\left(\sum_{j=1}^{n} w_{ij} x_j - \theta_i\right)\right)$$

si se tienen las restricciones de conexiones entre la capa de entrada y la intermedia, éstas se pueden simular con pesos igual a cero, por lo tanto

$$\phi(\tilde{F}(x_1, \ldots, x_n)) = \phi\left(\sum_{i=1}^{N} c_i \phi\left(\sum_{j=1}^{n} w_{ij} x_j - \theta_i\right)\right) = \phi\left(\sum_{i=1}^{n} c_i \phi\left(\sum_{j=1}^{a_1} w_{ji} x_j + .. + \sum_{j=a_p}^{m} w_{ji} x_j\right)\right)$$

4.2 APLICACION DE LA RED EQ CON FUNCION DE ERROR A CLASIFICACION INVARIANTE A TRANSLACION DE LOS PATRONES DE ENTRADA.

Como ya se vió en la sección 3.1, las redes en cascada con FE aprenden en ménos épocas que las redes que usan la FL. Ahora se utilizará ésta en el caso particular de la red EQ y en específico en un problema de clasificación de señales provenientes de la actividad eléctrica de neuronas.

4.2.1 Necesidad de la aplicación.

Como se explicó en la sección 1.3.1 el modelo aceptado de transferencia de información es en el que ésta se transmite exclusivamente por medio de impulsos eléctricos entre neuronas. Si se quiere corroborar esta hipótesis experimentalmente, es necesario clasificar estos impulsos

eléctricos (espigas) provenientes de distintas fuentes (neuronas) para así poder poner al conjunto de neuronas bajo estímulos externos y ver si las frecuencias de disparo de las neuronas cambian o si las conexiones entre ellas cambian.

Muchas variables intervienen en este proceso (como se verá en la siguiente sección), entre ellas, el ruido inherente a la medición y la aleatoriedad de la generación de pulsos debido al comportamiento probabilístico de las neuronas. Para clasificar las espigas existen métodos estadísticos y de análisis de Fourier, pero son lentos y esto influye cuando se está haciendo un experimento, ya que la zona que se está midiendo pertenece a un animal o si está *in vitro* no se puede mantener funcionando permanentemente. Es por esto, que se hace necesario un sistema de clasificación automática de la señal que no requiera tanto tiempo y que de información valiosa. El objetivo de la aplicación es reconocer espigas sin necesidad de pre-procesar la información, esto es, en el momento en que se esté llevando el experimento poder clasifcar las señales, para así, poder determinar si se está en una zona de interés para el experimentador. Esto se puede lograr usando una RNA.

Para hacer esto, primero se tiene que generar una base de datos de espigas en las que se tengan las formas de espigas más comunes de la región que se quiere estudiar. Se debe de entrenar una red con un conjunto de entrenamiento y después ver su desempeño por medio de un conjunto de prueba. Finalmente, se debe de probar con datos reales.

4.2.2 Metodología.

Para generar la base de datos requerida primero se tiene que procesar la señal proveniente del cerebro de rata, de tal manera que se pueden obtener ejemplos de los distintos tipos de espigas, usar algunos como conjunto de entrenamiento y otros como conjunto de prueba. La forma de adquirir adquirir y procesar la señal es la siguiente: (ver Fig. 4.4) [Gómez, Quiza y Espinosa, 1992; Serna, Austrich y Espinosa, 1992].

a) Los trenes de potenciales de acción (espigas) se adquieren del hipocampo de rata anestesiada, para ello se usan microelectrodos (25 micras de diámetro por 2.5 cm de largo).

b) Se preprocesa la señal con pre-amplificadores y amplificadores.

c) Se guarda la señal en videocasetes y al mismo tiempo se puede grabar en forma digital (en una computadora) por medio de una grabadora digital.

d) Se analiza la señal por las características inherentes a la forma temporal de la espiga, v.g. ancho de valle vs. altura de pico. Se pueden separar espigas por medio de tales características -lo que es una alternativa a lo propuesto aquí- (ver Fig. 4.5 y Fig. 4.6).

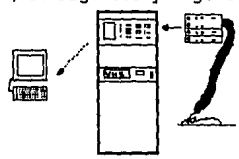


Fig. 4.4 Dispositivo experimental para adquirir espigas de rata anestesiada. De la rata se pasa a los preamplificadores, amplificadores, grabadora digital, almacenamiento en videocasetera y análisis en computadora. La utilización de la RNA-BP-EQ se hace en la computadora personal.

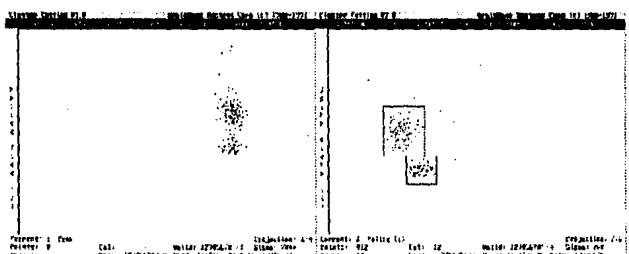


Fig. 4.5 Si graficamos las espigas por sus características podemos obtener un conjunto de puntos en un plano (en este caso, sobre el eje de las abscisa es ancho de valle y en el de las ordenadas el alto de pico). Se nota la formación de cúmulos, con esto, se pueden tomar criterios para separar las espigas.

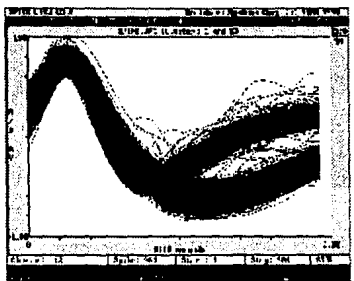


Fig. 4.6 Una vez separadas las espigas en cúmulos se pueden visualizar y se puede ver que son distintas.

Después de clasificar, se toma un conjunto de espigas de cada clase para servir como conjunto de entrenamiento y otro como conjunto de prueba.

4.2.3 Diseño y entrenamiento de la red.

Como primer paso para la solución de un problema es usual probar el sistema en condiciones controladas. Para el problema controlado se propuso clasificar ocho patrones distintos (ver Fig. 4.7) sin importar su variancia en translación (como en el trabajo original de Espinosa y Quiza).

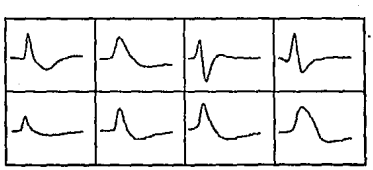


Fig. 4.7 Espigas usadas. Se armó un tren de 240 espigas agregando ruido aleatorio.

Los resultados del entrenamiento se encuentran en la tabla 4.1. Los números que están en el renglón de Arquitectura - Arq. - definen la arquitectura de cada red, el significado de estos números es: Nodos de Entrada, Nodos de Salida; Nodos por Grupo, Nodos Barridos, Nodos

Traslapados; Velocidad, Momentum. Se compara el error final y el número de iteraciones para cada programa. El programa Retro2 usa la FE definida en el capítulo segundo.

Tabla 4.1

Prog/Arq	Err Final	Epoca	Err Final	Epoca
Arq.	128,8;1,32,24;0.2,0.0		128,8;2,32,24;0.2,0.0	
Retro2	0.021507	100	0.009995	30
Arq.	128,8;1,32,20;0.15,0.0		128,8;2,32,20;0.15,0	
Retro2	0.018141	500	0.009997	92
Arq.	128,8;1,32,26;0.3,0.0		128,8;2,32,26;0.3,0.0	
Retro2	0.009885	68	0.009713	49

Para probar la red con patrones difusos y transladados se empleó un programa en el que se detectan espigas que se encuentran sobre un tren de pulsos, el cual contiene 240 espigas basado en los 8 originales agregando una señal que simula ruido sináptico, pero no comparable a la amplitud de las espigas (ver Fig. 4.8). El algoritmo de detección aplica una transformada de Haar para eliminar ruido después por medio de la amplitud y la derivada de la señal determina si ese tramo es o no una espiga y la presenta a la red [Espinosa y Quiza, 1991]. El punto de la detección puede variar dependiendo del nivel de ruido. Los resultados obtenidos fueron los mismos que los de los autores originales que reportaron haber clasificado correctamente toda la traza. En este programa se cambió el uso de la FL por la FE.

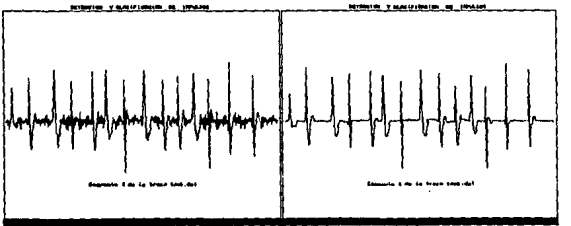


Fig. 4.8 Tren de impulsos (traza) con ruido no comparable a la magnitud de las espigas. La figura de la izquierda es una traza sin eliminar ruido, la de la derecha es la misma traza pero con el ruido eliminado.

4.2.4 Intentos por solucionar el problema real.

Como primer paso se analizó un experimento en el que con la metodología anterior se discernían claramente 2 clases de espigas que fueron separadas (ver sección 4.2.2), de este conjunto se toman algunas para entrenar la red y otras para ver su desempeño de clasificación. El número de épocas para aprender los dos patrones fué siempre menor a 5 y el porcentaje de clasificación al presentarle el conjunto de prueba fué del 100%.

Utilizando la traza completa (incluyendo el ruido, que era comparable en amplitud a las espigas) se trataron de clasificar los patrones, el resultado fúe que el desempeño de la red fué muy bajo (menor al 50%). Esto es debido a que el algoritmo de detección para que el programa diga si el tramo de señal es un pulso o no depende de la derivada y de la amplitud de la señal, como el ruido es comparable (en amplitud y derivada) al tamaño de la espiga, muchas de estas señales eran procesadas por la red. Por lo visto en la sección 2.2.3 las redes en cascada entrenadas con sólo los patrones que se quieren aprender tiene la desventaja de clasificar cualquier otro tipo de señal.

Para tratar de corregir este problema se trató de entrenar a la red para que aprendiera el ruido, sin embargo, no se pudo lograr que el error promedio cuadrado fuerá menor al 0.01. También se eliminó el criterio de la derivada para la clasificación, pero tampoco se logró obtener un buen resultado.

CONCLUSIONES

En este trabajo se ha visto:

a) El estudio de las redes neuronales, tanto biológicas como artificiales, no ha sido ajeno a la física. Por el contrario, la abstracción y el estudio formal de la física de las neuronas ha revolucionado el entendimiento de éstas.

b) Las redes neuronales artificiales se encuentran situadas dentro de la física de muchas partículas y esto resulta en sistemas estadísticos. Se demostró heurísticamente que las RNA tipo cascada son sistemas estadísticos y que la actividad de los nodos de la capa de salida representa la probabilidad de clasificar un patrón de entrada en alguna clase previamente aprendida.

c) Las RNA están apoyadas con teoremas que nos aseguran su utilidad para cualquier tipo de problema en el que el mapeo sea continuo. En este punto se aportó un nuevo corolario a los teoremas de Funahashi que nos da la posibilidad de hacer RNA invariantes a la translación del patrón de entrada.

d) Las RNA tienen aplicaciones importantes en el estudio de problemas actuales, como es el estudio de la dinámica de las interacciones de las neuronas en el cerebro de rata. En este punto, se propuso el uso de RNA para identificar espigas provenientes de distintas neuronas, se obtuvo un buen resultado cuando las señales eran sintéticas, sin embargo, la aplicación real se topó con muchos problemas.

A pesar de todo esto, el problema de la modelación de los fenómenos que se llevan a cabo en el cerebro no está resuelto. No existe una teoría general para la formación de las conexiones sinápticas, ni tampoco se han podido modelar completamente sistemas nerviosos de insectos o moluscos (supuestamente más sencillos que el de los mamíferos). Por el lado de las aplicaciones de RNA, no se han encontrado arquitecturas que resuelvan completamente los problemas planteados en este trabajo, pero las RNA compiten exitosamente contra otros métodos tradicionales, como son la transformada répida de Fourier (FFT).

Por otro lado, con lo relacionado a la parte teórica de este trabajo, no se pudo relacionar este tipo de redes neuronales con alguna de las estadísticas conocidads

(Maxwell-Boltzman, Fermi-Dirac y Bose-Einstein) [Alonso, 1986; Fowler, 1980; Zemansky, 1984]. Sin embargo, sí se obtuvieron resultados estadísticos provenientes de las interacciones de este tipo de redes neuronales.

El problema de pasar de un análisis teórico a aplicar una RNA para identificación de espigas se creía que podía ser resuelto con una red invariante a translación. Sin embargo, los problemas de la adquisición de señales se complicó por el ruido que se encuentra en la región del cerebro de rata que se analizó. Además, los tiempos de entrenamiento, se hacen largos en tiempo real (varias decenas de horas), esto pasa porque se llevan a cabo en una computadora en la que las operaciones se realizan de manera secuencial y no paralela, esto es, en la red en cascada que se usó las operaciones de los nodos se deberían de llevar a cabo al mismo tiempo en todos los nodos de cada capa.

Considerando la metodología que surgió al realizar este trabajo se puede decir que es relativamente fácil programar una RNA en Mathemática, esto ayuda a obtener resultados rápidos para saber si una RNA es buena o no. Sin embargo, para hacer una aplicación se tiene que programar en algún lenguaje como Pascal, C, C++ o Fortran para hacer más rápido el proceso de aprendizaje y prueba. Por otro lado, mundialmente se están trabajando con prototipos de RNA en VLSI (very large scale integration) ya que se minimizan los errores de sincronización que se obtienen en circuitos integrados. Es importante realizar estos proyectos en hardware, ya que se explotan todas las capacidades (de velocidad y generalización) de las RNA, además la aplicación última de una RNA se hace en este tipo de circuitos.

Finalmente se quiere proponer lo que podrían ser los siguientes pasos para el estudio teórico y de aplicaciones de las RNA.

a) Estudio de sistemas físicos: Análisis de las redes de Hopfield y cualquier sistema físico que tenga características emergentes (varios estados estables).

b) Estudio de la integral de Lesbegue: La integral de dos funciones es la misma si las funciones difieren por un conjunto de medida cero. Si se crea una red con esta característica, dos patrones que difieran por un conjunto de medida cero serían clasificados como de la misma clase. Esto podría ayudar para las redes invariantes a la translación,

ya que esta sería otra variable de clasificación. El problema, es que en el momento de la aplicación computacional se tendría que hacer una definición de medida cero de un conjunto.

c) Para clasificar trazas de espigas con ruido comparable a ellas: Usar una red auto-organizada de Kohonen, ya que tiene la capacidad de hacer cúmulos de los eventos que se parecen (en un espacio imaginario). Como el ruido es distinto y es aleatorio, no formará cúmulos tan definidos como el de las espigas.

d) Estudio de redes biológicas: Partiendo de las características físicas de las neuronas, ver que se puede simplificar, proponer modelos y corroborarlos con la experimentación.

APENDICE A

LISTADO DE LOS PROGRAMAS CODIFICADOS

En este apéndice se listan unicamente los programas codificados para la realización de este trabajo en el lenguaje de matemáticas simbólicas Mathematica. Las instrucciones para el uso del programa Retro2 se encuentran en la tesis de maestría de Jorge Quiza.

Los programas tienen el objetivo de comparar la FE vs. la FL.

Se utilizó el lenguaje de matemáticas simbólicas Matemática versión 2.0 para PC para implementar las redes propuestas. Se utilizó una PC-386 con coprocesador matemático.

Cada programa tiene en el encabezado una sección que explica la forma de uso de cada programa, la bibliografía base de estos programas es J. Fremman, Simulating Neural Networks with Mathematica, Addison Wesley, 1993. La figura A.1 contiene el diagrama cde flujo delos programas.

a) En bpnComp se comparan 200 épocas, se eligen los que convergen, se pasa a bpnReCom.

b) En bpnReComp se comparan con distino número de épocas, pero que tengan el mismo error promedio cuadrado.

En los programas bpnComp, y bpnReComp se usan los programas bpnBio, y el bpnPrueba en el primero se comparan las redes y en segundo se calcula el error promedio cuadrado al presentarse patrones de entrenamiento o difusos.

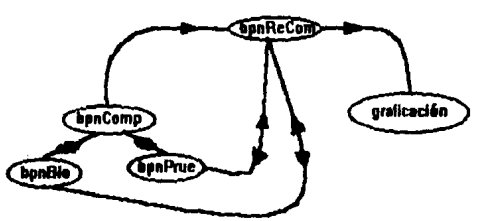


Fig. A.1 Diagrama de uso de los programas de entrenamiento de RNA-BP.

A.1 PROGRAMA BPNCOMP.MA

```
(* este programa va a comparar las redes de retropropagacion
con la funcion logistica (FL) y la de error (FE) la
funcion de error se define como Erf[x] y la logistica
1/(1+exp(-x))
Los parametros que se usan son el numero de nodos de
entrada, los nodos en la capa intermedia, la velocidad, el
momentum, el numero de iteraciones, el numero de
comparaciones por iteracion, el archivo donde estan los
pares de aprendizajes, i.e. xor.dat, dec.dat; el sufijo es
una cadena la cual es el sufijo con el que se van a guardar
los resultados, azar es la semilla de azar que se usara.
Las comparaciones se generan multiplicando la semilla por el
numero de la comparacion (1*azar, 2*azar,..., comp*azar)
Este programa se compone de otros: bpnBio y bpn Prueba, el
primero hace las comparaciones y el segundo calcula el error
promedio total al presentarle el conjunto de patrones de
entrenamiento.
Abril 1994.
Este es un programa para Mathematica 2.0 DOS
*)

ClearAll[bpnComp,bpnBio,bpnPrueba];


Print["Comparador de BPN  FE y logística"];
Print["bpnComp[Nodos Ent, Nod Inter, Nod Sal, vel, alpha,
iter, comp,\"pares\", \n \"sufijo\", azar];"];
bpnComp[nodEnt_,nodInter_,nodSal_,vel_,alp_,iter_,ito_,nombr
e_,num_,az_:195]:=
Module [{sbioSt,w,j},
  (*SetDirectory["f:\math\redneuro.nal"];*)
  Get[nombre];
  Print[ioPairs];
  <<bpnbio.ma;
  <<bpnprue.ma;

  SetDirectory["f:\math"];
  outsST={0,0,0};
```

```
  outsBIO={0,0,0};

  arqui={nodEnt,nodInter,nodSal,vel,alp};


  For[ w=0, w<ito, w++, (* for de comparaciones *)
        {
        azar=az * (w+1) ;

        Print["Comparacion: ",w+1,", Arq: ",nodEnt,"
",nodInter," ",
                  nodSal," ",vel," ",alp," ",iter," ",ito,",
Azar: ",azar];

        f=1;
        For [j=0,j<2,j++,  (* switch *)

          If[f<2,outsBIO=
bpnBio[nodEnt,nodInter,nodSal,ioPairs,vel,alp,
                          iter,f,azar],
                    outsST=
bpnBio[nodEnt,nodInter,nodSal,ioPairs,vel,alp,
                          iter,f,azar]
            ]; (* fin del if f<2 *)


          f=2];(* fin del for de j *)


bioArc=StringJoin["redneuro.nal\bio",num,ToString[w+1],".dat
"];

stArc=StringJoin["redneuro.nal\std",num,ToString[w+1],".dat"
];
      OpenWrite[bioArc];
      OpenWrite[stArc];
      Save[bioArc,outsBIO,azar,arqui];
      Save[stArc,outsST,azar,arqui];
      Close[bioArc];
      Close[stArc];
      Archivos[bioArc,", ",stArc];
         )
```

```
    ]; (* fin del for de w - comparaciones - *)
  ResetDirectory[];

  ]; (* fin del modulo *)
```

A.2 PROGRAMA BPNBIO.MA

```
(*Este programa compara redes neuronales artificiales con un
nodo bias en la capa de entrada y otro en la capa
intermedia. Los parametros de la funcion son el numero de
nodos de entrada, el numero de nodos de la capa intermedia,
el numero de nodos de la capa de salida, la variable que
almazena los pares de entrenamiento, la velocidad, el
momento, el numero de iteraciones, la f que es un switchque
si vale 1 se usa la  FE y si es 2 se usa la sigmoidal, la
semilla de azar. Al final de cada proceso llama a la funcion
bpnPrueba la cual calcula el error promedio cuadrado de la
red al presentarle los patrones de entrenamiento.
Abril 1994
Este es un programa para Mathematica 2.0 DOS
*)

bpnBio[inNumber_,hidNumber_,outNumber_,ioPairs_,eta_,
        alpha_,numIters_,f_,azar_:195]:=

Module[{errors,hidWts,outWts,ioP,inputs,outDesired,hidOuts,
          outputs,
outErrors,hidLastDelta,outDelta,hidDelta,i,long},

    SeedRandom[azar];
    long=Length[ioPairs];
    If[f<2,{sigmoid[x_]:=  Erf[x/2];
            derivada[x_]:=  0.56419 E^(-(x)^2/4)
            },
           {sigmoid[x_]=2/(1 + E^(-x))-1;
            derivada[x_]:= 2 E^(-x)/(1+E^(-x))^2
            }];
      (* definicion de la funcion de trans *)

    hidWts= Table[Table[Random[Real,{-
2.0,2.0}],{inNumber+1}],{hidNumber}];
```

```
    outWts= Table[Table[Random[Real,{-
2.0,2.0}],{hidNumber+1}],{outNumber}];
    hidLastDelta = Table[Table[0,{inNumber+1}],{hidNumber}];
    outLastDelta =
Table[Table[0,{hidNumber+1}],{outNumber}];

errorList= Table[
    For[errores=0;i=1, i<(long+1),i++,
     {
     ioP=ioPairs[[Random[Integer,{1,long}]]];
     inputs=Append[ioP[[1]],1.0];
     outDesired=ioP[[2]];
     hidOuts= sigmoid[hidWts.inputs];
     outInputs= Append[hidOuts,1.0];
     outputs= sigmoid[outWts.outInputs];
     outErrors = (outDesired-outputs);
     errores += outErrors.outErrors;

     If [First[Abs[outErrors]]>0.01,
       outDer= derivada[outWts.outInputs];
       outDelta= outErrors outDer;

       hidDer= Append[derivada[hidWts.inputs],0];
       hidDelta= hidDer * Transpose[outWts].outDelta;

       outLastDelta = eta *
                  Outer[Times,outDelta,outInputs] + alpha
outLastDelta;
       outWts += outLastDelta;
       hidLastDelta = eta *
          Drop[Outer[Times,hidDelta,inputs],-1] + alpha
hidLastDelta;
       hidWts += hidLastDelta, Continue]; (* fin del if
outErrors *)
     }];  (* fin del for de errores *)
     outErrors=(1 / ( 2 long)) * errores;
     outErrors, {numIters}]; (* fin de la tabla *)
     bpnPrueba[hidWts,outWts,ioPairs];
     Return[{hidWts,outWts,errorList}];
    ];
```

A.3 PROGRAMA BPNPRUE.MA

```
(* Este programa calcula el error promedio cuadrado de una
red neuronal, los parametros son los pesos de la capa
oculta, los pesos de la capa intermedia y los patrones de
prueba, estos patrones pueden ser los mismos con los que se
entreno la red o tambien pueden ser patrones difusos.
El formato debe de ser una lista de la siguiente manera
ioPairs={{{patron prueba},{salida}},...,
         {{patron prueba},{salida}} };
            |__patron original    |__ salida esperadoa
                  o difuso
este nombre  (ioPairs) de variable siempre se debe de usar
*)

bpnPrueba[hidWts_,outWts_,ioPairs_]:=
Module[{prueba,uno,unomas,error,erroCuad,i,j,long,pesosEsc,p
esosSal},
pesosEsc=hidWts;
pesosSal=outWts;
outTest=ioPairs;
long1=Length[ioPairs];

(* Print["************************************************"];
*)
For[j=0,j<long1,j++,
long=Length[ioPairs[[j+1]]];
For[error=0;i=0,i<long,i++,
  ioP=ioPairs[[j+1,i+1]];
  prueba=Append[ioP[[1]],1];
  uno=sigmoid[pesosEsc.prueba]; (* salida de la capa
intermedia *)
  unomas=Append[uno,1];
  dos=sigmoid[pesosSal.unomas]; (* salida final *)

  error += (ioP[[2]] - dos).(ioP[[2]] - dos); (* error
cuadrado *)

(*  Print["Entrada: ",ioP[[1]]];
  Print["Salida: ",ioP[[2]] ];
  Print["Salida real: ",dos];*) (* opcional*)
  outTest[[j+1,i+1,2]]= dos;
```

```
(*  Print["Error: ",ioP[[2]] - dos];
  Print[""];*)  (* opcional *)
  ]; (* fin del for i *)
  ]; (* fin del for j *)
  errorCuad= 1/(2 long long1) * error;
(*  Print["                    Error promedio cuadrado:
",errorCuad]; *)
    (* opcional *)
  Return[outTest];
]; (* fin del modulo *)
```

A.4 PROGRAMA BPNRECOM.MA

```
(* Este programa va a comparar las redes de retropropagacion
con la funcion logistica y la  de error.
La diferencia de este programa con el bpnCom.ma es que se
pueden dar parametros diferentes (numero de épocas) para
cada red. El nuevo parametro que se usa es el nombre del
archivo donde se encuentra la lista que incluye el numero de
iteraciones para cada red y su semilla de azar, el formato
que se us para esta lista es la siguiente

iter_Bio iter_Std Azar Num
    |        |      |   |____Numero de prueba generada en
    |        |      |        el bpncomp.ma
    |        |      |_______Azar de la comparacion
    |        |______________Numero de iteraciones de la red
    |                       con FL
    |______________________Numero de iteraciones de la red
                            con  FE
Este archivo debe de estar en ASCII.
El proposito de este programa es despues de entrenar ambas
redes con un numero de iteraciones fijo se busca cuando
llegan a un error promedio cuadrado menor a un numero
fijado, i.e. 0.01, despues se vuelven a entrenar solo hasta
la iteracion donde se consiguio llegar establemente a ese
error.
La salida de este programa son archivos que tiene un prefijo
bio o std, un sufijo que se le agrega una r y extension dat.
Los datos ahi almacenados son una lista que tiene el nombre
outsBio o outsStd con los pesos en los nodos intermedios,
```

```
los pesos en los nodos de salida y la lista de error
promedio cuadrado, otra lista llamada arqui con la
arquitectura de la red
arqui={nodEnt,nodInter,nodSal,vel,alp};
y la variable azar que contiene el valor de azar.
Abril 1994
Este es un programa para Mathematica 2.0 DOS
*)

ClearAll[bpnReCom,bpnBio,bpnPrueba];

Print["Comparador de BPN  de error y logística"];
Print["bpnReCom[Nodos Ent, Nod Inter, Nod Sal, vel, alpha,
\"pares\", \n \"sufijo\", \"lista\"];"];

bpnReCom[nodEnt_,nodInter_,nodSal_,vel_,alp_,nombre_,num_,no
mlist_]:={
Module [{w,j,long,lista},
  Get[nombre];
      (* archivos de pares de ejemplos *)
  Print[ioPairs];
  lista=ReadList[nomlist,Number,RecordLists -> True];
      (* informacion de  entrenamiento *)
  long=Length[lista];
  <<bpnbio.ma;
  <<bpnprue.ma;

  SetDirectory["f:\math"];
  arqui={nodEnt,nodInter,nodSal,vel,alp};

 For[w=1,w<(long+1),w++,{
  azar=lista[[w,3]];
  outsST={0,0,0};
  outsBIO={0,0,0};
  Print["Comparacion: ",lista[[w,4]],", Arq: ",nodEnt,"
",nodInter," ",
          nodSal," ",vel," ",alp," ",lista[[w,1]],", Azar:
",azar];

  outsBIO= bpnBio[nodEnt,nodInter,nodSal,ioPairs,vel,alp,
                  lista[[w,1]],1,azar];
```

```
  Print["Comparacion: ",lista[[w,4]],", Arq: ",nodEnt,"
",nodInter," ",
          nodSal," ",vel," ",alp," ",lista[[w,2]],", Azar:
",azar];

  outsST=  bpnBio[nodEnt,nodInter,nodSal,ioPairs,vel,alp,
                  lista[[w,2]],2,azar];


bioArc=StringJoin["redneuro.nal\bio",num,ToString[lista[[w,4
]]],"r.dat"];

stArc=StringJoin["redneuro.nal\std",num,ToString[lista[[w,4]
]],"r.dat"];
  OpenWrite[bioArc];
  OpenWrite[stArc];
  Save[bioArc,outsBIO,azar,arqui];
  Save[stArc,outsST,azar,arqui];
  Close[bioArc];
  Close[stArc];
  (* Archivos[bioArc,", ",stArc]; *)
}]; (* fin del for *)
  ResetDirectory[];

]}; (* fin del modulo *)
```

REFERENCIAS

1. Alonso, M. y Finn, E. J. **Física, vol. 3: Fundamentos cuánticos y estadísticos.** Addison-Wesley Iberoamericana, 1986.

2. Anderson, J. **A simple neural network generating an interactive memory.** Math. Bios. 14, 197-220 (1972).

3. Amit, D. J. **Modeling brain function.** Cambridge University Press, 1989.

4. Caianiello, E. R. **Outline of a theory of thought processes and thinking machines.** J. Theoret. Biol. 1, 204-235, 1961.

5. Cooper, L. **A possible organization of animal memory and learning.** Proceedings of the Nobel Symposium on Collective Properties of Physical Systems. Academic Press, 252-264, 1973.

6. Darsey, H. **Application of neural network computing to the solution for the ground state eigenenergy of two-dimensional harmonic oscillator.** Chem. Phys. Lett., 177:2, 189-194 (1991).

7. Denby, B. **Spatial pattern recognition in a high-energy particle detector using a neural network algorithm.** Comput. Phys. Commun., 56:3, 293-297 (1990).

8. Descartes, R. y Spinoza, A. **Discourse on Method y Meditations on First Philosophy.** Britannica Great Books, 1988.

9. Espinosa E., I. y Quiza T., J. **Classification of noisy action potentials (APs) by means of a neural network employing back-propagation.** Soc. Neurosci. Abstr., 16, 1092 (1990).

10. Espinosa E., I. y Quiza T., J. **Off-line sorting of spikes using an artificial neural network.** Soc. Neurosci. Abstr., 17, 124 (1991).

11. Erkki O. **Neural networks, principal components, and subspaces.** Int. Jour. of Neur. Sys., 1:1, 61-68 (1989).

12. Freeman, J. A. **Simulating neural networks with Mathematica.** Addison-Wesley, 1994.

13. Fowler, R. H. **Statistical mechanics.** Cambridge University Press, 1980.

14. Funahashi, K. **On the approximate realization of continuous mappings by neural networks.** Neural Networks, 2:3, 183-192 (1989).

15. Gerstein, G., Bloom, M., Espinosa, I., Evanczuk, S., Turner, M. **Design of a laboratory for multineuron studies.** IEEE Transactions on Systems, Man, and Cybernetics, SMC-13:5, 668-676 (1983).

16. Geszti, T. **Physical models of neural networks.** World Scientific, 1990.

17. Gómez, J., Quiza, J., Espinosa, I. **Experimentos piloto sobre la actividad neuronal distribuida en el cerebro de la rata anestesiada.** Rev. Mex. Ing. Biomed., 13:2, 379-386 (1992).

18. Gutfreund, H. **From statistical mechanics to neural networks and back.** Physica A, 163, 373-385 (1990).

19. Harmon, L. D. y Lewis, E. R. **Neural modeling.** Phys. Rev. 46(513), 513-591 (1966).

20. Hebb, D. O. **The organization of behavior.** Wiley, 1949.

21. Hetcht-Nielsen, R. **Neurocomputing.** Addison-Wesley. (1990).

22. Hopfield, J. J. **Neural networks and physical systems with emergent collective computational abilities.** Proceedings of the National Academy of Sciences (USA), 79, 2554-2558 (1982).

23. Humpert, B. **On the use of neural networks in high-energy physics experiments.** Comput. Phys. Commun., 56:3, 299-311 (1990).

24. Kohonen, T. **Self-organization and Associative memory.** Springer-Verlag, 1984.

25. le Cun, Y. **Modeles conexionnistes de l'apprentissage.** Disertación doctoral., Universidad de Pierre y Marie Curie, Paris, Francia, 1987.

26. Xu, L., Krzyzak, A., Oja, E. **Neural nets for dual subspace pattern recognition method**. Int. Jour. of Neur. Sys., 2:3, 169-184 (1991).

27. Levin, E, Tishby, N. y Solla, S. **A statistical approach to learning and generalization in layered neural networks**. Proceedings IEEE, 78:10, 1568-1574, Octubre (1990).

28. Little, W. A. **The existence of persistent states in the brain.** Math. Biosci. 19, 101-120 (1974).

29. Little, W. A. **The evolution of non-newtonian views of brain function.** Concepts in Neuroscience 1:1, 149-164 (1990).

30. Milner, M. P. **The mind and Donald O. Hebb**. Sci. Am., Enero (1993).

31. Pao, Y. **Adaptive pattern recognition and neural networks.** Adisson-Wesley, 1989.

32. Parker, D. B. **A comparison of algorithms for neuron-like cells.** en Denker, J. [Ed.], Proc. Second Annual conf. on Neural Networks for Computing, 151, 327-332, Am. Inst. of Physics, New York, 1986.

33. Peretto, P. **An introduction to the modeling of neural networks**. Cambridge University Press, 1992.

34. Quiza, J. **Clasificacion de potenciales de acción con una red neuronal que emplea retropropagación.** Tesis de maestría en ingeniería biomédica. Universidad Autonoma Metropolitana (1991).

35. Rosenblatt, F. **The perceptron: A probabilistic model for information storage and organization in the brain.** Phys. Rev. 65, 386-408, 1958.

36. Rothschuh, K. E. **History of Physiology**. R. E. Krieger Publishing Company, 1973.

37. Rumelhart, D. E., et al. **Parallel distributed processing vol. 2**. MIT Press, 1986.

38. Selvam, A. M., **Deterministic chaos model for self-organized adaptive networks in atmospheric flows**. Proceedings of the

IEEE 1989 Nat. Aerospace and Electronics Conference, NAECON 1989, 3, 1145-1152 (1989).

39. Serna, R., Austrich, J., Espinosa, I. **Reconocimiento, mediante programa, de espigas en registros neurofisiologicos compuestos**. Rev. Mex. Ing. Biomed., 13:2, 353-356 (1992).

40. Simpson, P. K. **Artificial neural systems**. Pergamon Press, 1990.

41. Stein, D. L.. **Spin glasses**, Sci. Am., Julio, 36-42 (1989).

42. Waserman, P. **Neural computing**. Van Nostrand Reinhold, 1989.

43. Werbos, P. J. **Beyond regression: New tools for prediction and analysis in the behavioral sciences**. Doctoral Dissertation, Appl. Math. Harvard University, Noviembre 1974.

44. Widrow B. and Lehr M. **30 years of adaptive neural networks: Perceptron, madaline and backpropagation**. Proceedings IEEE, 73:9, 1415-1442 (Sept 1990).

45. White, H. **Learning in artificial neural networks: A statistical perspective**. Neural Computation 1:4, 425-464 (1989).

46. Zemansky, M. W. **Calor y termodinámica**. McGrawHill, 1984.