



13
20
UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES

“ARAGON”

ELEMENTOS DE UNA RED
NEURONAL ARTIFICIAL

T E S I S

Que para obtener el Título de:
INGENIERO EN COMPUTACION

P r e s e n t a n

CARLOS FLORES BECERRIL

EUGENIO VILLALOBOS SANCHEZ

TESIS CON
FALLA DE ORIGEN

San Juan de Aragón, Edo. de Méx.

1994



ENEP
ARAGON



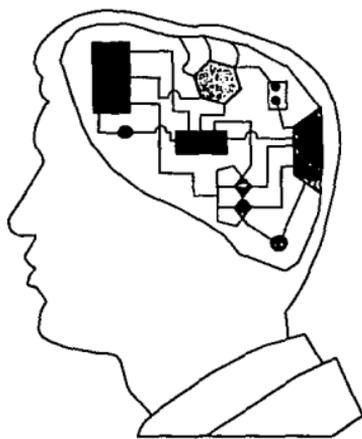
UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

ELEMENTOS DE UNA RED



NEURONAL ARTIFICIAL

Agradezco:

a mis padres:

Eugenio y Gudelia por el apoyo incondicional que dieron a mis estudios sabiendo que nunca podré pagar esa deuda que tengo con ellos.

a mis hermanos:

Antonio, José Luis con quienes compartí buenos y malos momentos lejos de nuestros padres; a Mauro, Francisca y Romelia que me guiaron por el camino correcto.

EUGENIO

a mis padres:

***Jorge y Teresa
que aunque con estas palabras sé que
no bastan para describir el apoyo y
estimulo que me brindaron durante toda
mi carrera profesional, les estaré siempre
agradecido, ya que sin ellos no podría
haberla culminado.***

a mis hermanos:

***Ernesto, Leticia, Armando, Lilia y Jorge
les doy las gracias por haber estado
conmigo en todo momento y que de una u
otra forma colaboraron para lograr
mis metas, esperando corresponderles en
su momento de la misma manera.***

CARLOS

***Agradecemos a nuestros profesores
quienes aportaron su valioso tiempo,
que con sus observaciones y
comentarios ayudaron a enriquecer este
trabajo.***

***Y a todas aquellas personas que siempre
estuvieron con nosotros sin esperar nada
a cambio.***

INDICE

INTRODUCCION	1
I.- Introducción a las redes neuronales artificiales.....	3
Introducción	3
1.1.- Antecedentes	4
1.2.- Principios	8
1.3.- Definición	13
II.- Elementos de un red artificial	18
Introducción	18
II.1.- Vectores de entrada, vectores ocultos y vectores de salida	21
II.2.- Conexiones	26
II.3.- Elementos de procesamiento (P.E.)	28
II.4.- Funciones P.E.	33
III.- Topología	40
Introducción	40
III.1.- Instars, Oustars y ADALINE	43
III.2.- Red de un nivel	45
III.3.- Red de dos niveles	51
III.4.- Red Multinivel	54
III.5.- Redes enlazadas aleatoriamente	56

IV.- Alimentación en las redes	58
Introducción	58
IV.1.- Alimentación hacia adelante	60
IV.2.- Alimentación hacia atrás	65
V.- Principios de los modelos ADALINE y MADALINE	69
Introducción	69
V.1.- ADALINE	71
V.2.- MADALINE	77
CONCLUSIONES	95
BIBLIOGRAFIA	97

INTRODUCCION

Durante la segunda mitad del presente siglo el desarrollo de los sistemas artificiales ha venido avanzando a grandes pasos, como consecuencia de las necesidades que van surgiendo en las tareas diarias del hombre, que cada vez son más difíciles de realizar. De entre las que destacan las labores complejas en las que se requiere de un alto grado de precisión y rapidez para llevarlas a cabo.

Dentro de estos sistemas artificiales debido a su importancia destacan principalmente: Los sistemas expertos y los sistemas basados en **Redes Neuronales Artificiales**, siendo este último el punto central en este trabajo.

Esta área de la inteligencia artificial debe su nombre a la intención de crear sistemas cuyo comportamiento fuera semejante al de las neuronas del cerebro humano, es decir, sistemas que fueran capaces de recibir patrones de entrada, procesarlos junto con la información que previamente es almacenada y producir resultados, que si bien no son los deseados, volver a ser tomados por los sistemas cuantas veces sea necesario, para finalmente proporcionar resultados satisfactorios.

La perspectiva de este trabajo tiene como objetivo principal proporcionar una visión amplia sobre las redes neuronales artificiales, desde el simple concepto hasta sus aplicaciones. Sobre sus inicios, principios, conceptos fundamentales así como su analogía con las neuronas biológicas se hace referencia en el capítulo uno. En tanto en el segundo capítulo se dá el siguiente paso, que consiste en conocer los componentes básicos que estructuran a las redes neuronales artificiales y sus características principales.

Durante el proceso del diseño de las redes neuronales artificiales se debe realizar una adecuada selección de la estructura que tendrá la misma, ya que esta debe ser acorde a las características de funcionamiento para cubrir las necesidades de aplicación que se le haya asignado; para esto, se ha destinado el capítulo número 3, donde se mencionan los tipos más comunes de topología.

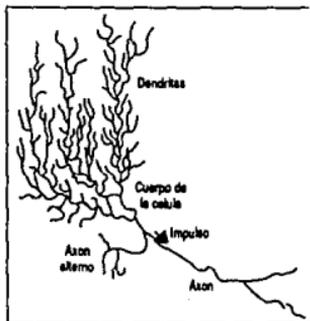
Otro aspecto a considerar en estos diseños es el método de conexión de los elementos internos de la red, así como la forma de alimentación que tendrá y que vayan de acuerdo con alguno de los algoritmos de entrenamiento que existen, que en este trabajo en particular se utiliza el algoritmo de retropropagación. Por lo que estos aspectos se contemplan en el capítulo 4.

Finalmente en el capítulo 5 se describen dos de los modelos de redes neuronales artificiales de estructura más sencilla, pero importante, debido a que estas sirven como base para la construcción de modelos más complejos, que para su manejo requieren de un estudio más profundo. Estos modelos son el Adaline y su extensión el Madaline.

Capítulo

I

INTRODUCCION A LAS REDES NEURONALES ARTIFICIALES



INTRODUCCION

Los fundamentos de las redes neuronales artificiales se basan en el estudio y la comprensión de la estructura y de los conceptos de carácter matemático que son utilizados para definir y analizar el procesamiento de dichas redes. En la mayoría de las investigaciones realizadas sobre las redes neuronales artificiales, se les ha atribuido tres propiedades principales que las caracterizan, las cuales se presentan a continuación:

TOPOLOGIA.

En esta propiedad se describe la organización y la interconexión de los niveles que forman parte de la red, esto sirve para tener una mejor visión acerca de la red que se desea trabajar y/o se esté investigando.

ENTRENAMIENTO.

Esta propiedad tiene como función describir la metodología que es utilizada para proporcionarle a la red la capacidad de recibir, procesar, almacenar y reutilizar la

información que le sea suministrada para finalmente enviar resultados, que en forma análoga a las neuronas biológicas sería: percibir, comprender y actuar.

RECUPERACION.

Esta propiedad esta muy relacionada con la anterior, debido a que en la etapa de recuperación se describe el proceso de extracción de la información que ha sido previamente almacenada en la red.

1.1. ANTECEDENTES.

Históricamente el estudio de las redes neuronales artificiales tiene principalmente dos objetivos: el deseo de comprender los principios sobre los cuales trabaja el cerebro humano, y el lograr la construcción de máquinas las cuales sean capaces de resolver tareas complejas.

Los primeros estudios llevados a cabo sobre redes neuronales artificiales datan de la década de los 40's, los cuales se le atribuyeron a los investigadores Warren McCulloch y a Walter Pitts quienes propusieron su teoría general acerca de los sistemas de procesamiento de información. Dicha teoría describe el modelo de redes construidas a base de interruptores binarios o también conocidos como elementos de decisión que eran capaces de realizar operaciones lógicas así como funciones de carácter aritmético-lógico, como se observa en la figura 1.1; debido a estas características se les acreditó el nombre de **Neuronas**.

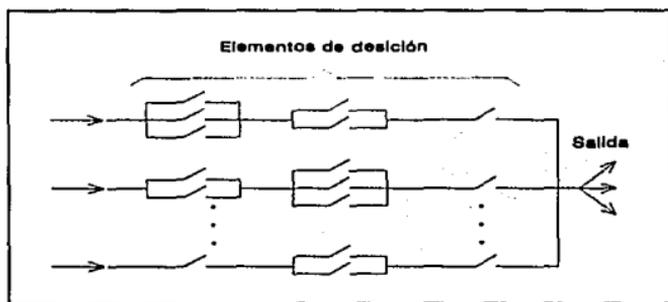


Fig.1.1.- Modelo de red neuronal artificial propuesto por McCulloch-Pitts.

Cada uno de estos elementos de decisión puede tomar un valor de salida de 0 o 1, de los cuales, cuando un elemento toma el valor 0 significa que se encuentra en estado inactivo, contrariamente si se encuentra en el estado 1 decimos que se encuentra en el estado activo. Mc Culloch y Pitts en su teoría manejan la diferencia que existe entre una red neuronal artificial y una computadora tradicional, consistente en que en la primera, las instrucciones de un programa no se ejecutan de forma secuencial como ocurre en las computadoras tradicionales, sino en forma paralela.

Para el año de 1949 Donald Hebb propone una regla para comprender la conexión que existe entre las neuronas, y que hasta la fecha se le conoce como la **Regla de Aprendizaje Hebbiana**¹, esta regla afirma que toda información proveniente del exterior puede almacenarse en el interior de las conexiones, además postula la técnica de aprendizaje que habría de servir como base para futuros desarrollos sobre este campo. La regla de aprendizaje de Hebb fue una de las primeras que aportó gran ayuda a la teoría de la red neuronal artificial.

En el transcurso de la década de los 50's, el investigador Marvin Minsky comenzó sus trabajos sobre las primeras neurocomputadoras, para finalmente en 1954 construir y probar modelos que había propuesto, en los que se adaptaron conexiones del tipo automático, y que servirían como base para el modelo denominado **PERCEPTRON**.

Alrededor de los años 60's Frank Rosenblatt y un grupo de colaboradores realizaron estudios sobre una clase específica de red neuronal artificial a la cual denominaron **PERCEPTRON**², que en un principio fue considerada como un modelo simplificado que trata de simular los mecanismos biológicos de procesamiento de información.

En forma general, un Perceptron está constituido por dos niveles que se encuentran separados y que representan al nivel de entrada y salida respectivamente, como se observa en la figura 1.2, en la que los elementos del nivel de salida reciben las señales provenientes de los elementos que se encuentran en la entrada, pero nunca en sentido contrario; los elementos que se encuentran en un sólo nivel no se

¹Hebb, D. "Organization of Behavior" New York: Wiley, 1949.

²Rosenblatt, F. "Principles of Neurodynamics". Washington, D.C: Spartan Books, 1962.

comunican entre sí, por lo que se dice que el flujo de información es estrictamente direccional.

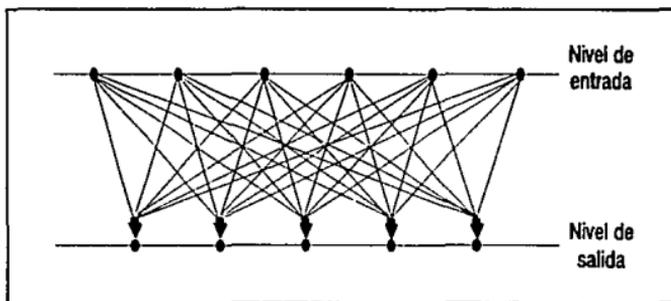


Fig.1.2.- PERCEPTRON: Modelo de red neuronal artificial propuesto por Frank Roseblatt.

En esta misma década un nuevo proyecto fue introducido, el cual fue llamado **ADALINE** que se deriva del inglés **AD**aptive **LINE**ar **CO**mbiner, junto con una nueva regla de aprendizaje que lleva como nombre **Regla de Aprendizaje Widrow-Hoff**³. Algunas de las primeras aplicaciones que se hicieron con los modelos ADALINE y su extensión **MADALINE** que significa varias ADALINE (Many ADALINE) incluyeron el reconocimiento de patrones, pronósticos del tiempo y controles adaptivos. Todo lo referente a estos modelos será estudiado en un capítulo posterior.

En tanto en Japón, Sun Ichi Amari (1972, 1977) propone un estudio acerca del aprendizaje en redes neuronales construidas de elementos preestablecidos. Por su parte, Kunihiro Fukushima desarrolla una arquitectura de red neural artificial a la cual llamó **NEOCOGNITRONS**⁴, esta arquitectura fue diseñada para el reconocimiento de patrones de imágenes; estas redes realizaban una simulación de una imagen tomada mediante un dispositivo que simulaba una retina artificial, para posteriormente procesarla, utilizando para ello, niveles bidimensionales de neuronas, como se muestra en la figura 1.3.

³Widrow, B. and M. Hoff. "Adaptive Switching Circuits", in WESCON Convention Records, vol. 4. 1960.

⁴Fukushima, K. "Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition", Neural Networks, vol. 1. 1988.

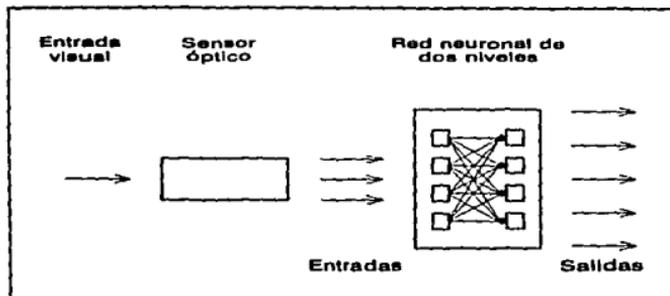


Fig.1.3.- NEOCOGNITRON: Modelo de red neuronal artificial propuesto por Kunihiko Fukushima.

Durante este período, Little y Gordon Shaw así como John Hopfield estudiaron el proceso en el que una red neuronal logra almacenar y extraer información (sistema recursivo); además, Little apunta las similitudes que existen entre la red neural del tipo propuesto por McCulloch-Pitts y los sistemas denominados **Spins** o **Sistemas de Instantes Magnéticos Elementales** o modelos Ising. Estos sistemas se pueden observar en la figura 1.4, en donde las localidades que se encuentran en el enrejado únicamente pueden tomar dos diferentes orientaciones, como pueden ser hacia arriba o hacia abajo. En analogía con una red neuronal biológica, se identifica a cada localidad como una neurona artificial y se asocia la orientación hacia arriba con el estado activo y la orientación hacia abajo con el estado inactivo.

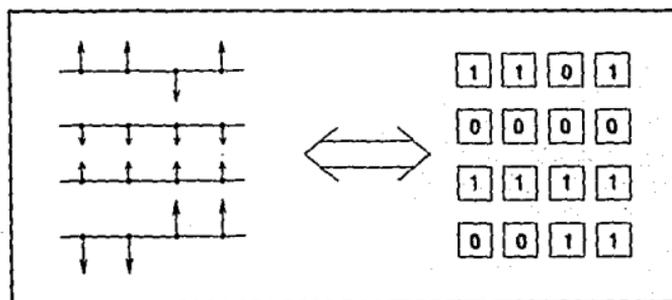


Fig.1.4.- Sistemas Spins o de Instantes Magnéticos Elementales propuestos por Little y Shaw.

En los años de 1986 y 1987 surgieron nuevamente programas dedicados a la investigación de redes neuronales artificiales a tal grado de que llegaron a propagarse un gran número de conferencias así como publicaciones de revistas enfocadas al campo de la neurocomputación.

A partir de 1986 cuando AT & T fabrica los primeros modelos de memorias neuronales en circuitos integrados, se observa el crecimiento que han tenido estos circuitos microelectrónicos; para esto, los sistemas neuronales artificiales fabricados con tecnología VLSI (Very Large Scale Integrated) han acaparado el mercado de la electrónica.

Durante este crecimiento electrónico se han creado expectativas acerca de las futuras aplicaciones que se le darán a los sistemas neuronales, dentro de las cuales destacan la visión electrónica, hablar, tomar decisiones, y razonar, así como la intervención en los procesadores de señales tales como los filtros, los detectores y los sistemas de control de calidad.

Actualmente la tecnología en las redes neuronales lleva un paso muy acelerado en su desarrollo, como es el caso de la expansión de los sistemas de redes neuronales artificiales, los cuales están basados esencialmente en máquinas inteligentes. Afortunadamente esta expansión hará que las tareas que hoy en día son difíciles, puedan resolverse de una manera más sencilla y esto como consecuencia, traer una mejor calidad de vida.

1.2. PRINCIPIOS.

Para tener una mejor comprensión sobre el concepto de redes neuronales artificiales, se tienen que analizar varios puntos acerca de las redes neuronales biológicas que a diferencia de las primeras, estas se encuentran alojadas en el cerebro, para esto, es necesario estudiar el funcionamiento general del cerebro humano.

Un cerebro humano esta construido de aproximadamente 2^{10} elementos inteligentes a los cuales se les denomina en términos biológicos Neuronas, capaces de comunicarse mediante una red de conexión compuesta de axones y sinápsis; una neurona contiene una cantidad de aproximadamente 2^{10} sinápsis. Dentro de la medicina, una hipótesis con respecto al modelo de un sistema nervioso biológico, dicta que las neuronas transmiten la información entre ellas mediante impulsos eléctricos.

La entrada de una red neuronal biológica esta compuesta por un conjunto de sensores que desempeñan la función de receptores, los cuales proporcionan los estímulos provenientes del exterior al interior de la red; dichos estímulos están representados por impulsos eléctricos que se encargan de transmitir la información hacia el sistema nervioso central, provocando así, efectos de respuesta humana, la cual puede estar expresada en una gran variedad de acciones. En la figura 1.5 se muestra el proceso que se lleva a cabo para enlazar la información al sistema nervioso central.

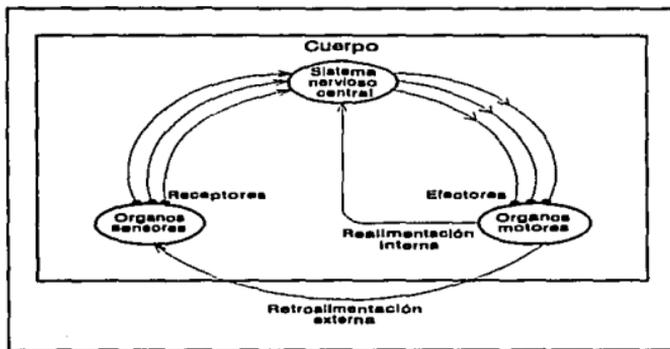


Fig.1.5.- Proceso para enlazar la información al sistema nervioso central.

Como se observa en la figura 1.5, la información es manejada en tres etapas, las cuales son: el procesamiento, la evaluación y la comparación con la información que se encontraba previamente almacenada dentro del sistema nervioso central. En los momentos en que son requeridas, el sistema nervioso central genera señales de control que son transmitidas hacia los órganos motores, con el fin de tener un monitoreo de los mismos. Este rastreo se lleva a cabo mediante los enlazadores de retroalimentación que se encargan de verificar esta acción.

En general, se observa que de un sistema nervioso, en su estructura, muchas de sus características son semejantes a las que posee un sistema de control infinito.

Como siguiente paso en el estudio de las redes neuronales artificiales, esta el de analizar la estructura y la fisiología de la neurona biológica, para posteriormente ver y establecer la analogía que existe entre estos dos modelos de neuronas.

Las neuronas son los componentes elementales en una célula nerviosa, a su vez éstas están constituidas de tres bloques principales, que son:

CUERPO.

Es la parte dilatada que posee la neurona, dentro del cual encontramos el núcleo y dos fibras nerviosas delgadas, parecidas a cabellos que se extienden a partir de dicho cuerpo. Estas fibras varían en su anchura, como pueden ser desde unos cuantos micrones hasta 30 o 40 micrones, y de longitud desde 1 o 2 milímetros hasta más de un metro.

AXONES.

Es el bloque especializado en distribuir o conducir excitación desde la zona dendrítica. Generalmente, es largo y liso, pero puede dar un colateral ocasional. Cuando un axon se separa del cuerpo celular debido a un corte, este sufre un ciclo de degeneración. Los axones conducen con normalidad el impulso nervioso en sentido opuesto al cuerpo celular, es decir, alejándose del mismo.

DENDRITAS.

Constituyen el bloque de la neurona especializado en recibir la excitación, tanto de los estímulos del medio ambiente como el de otra neurona. En conjunto forman un elemento denominado Arbol Dendrítico, el cual está constituido de fibras delgadas que rodean al cuerpo de la neurona. El impulso nervioso que conducen las dendritas es en sentido contrario al de los axones, es decir, hacia el interior del cuerpo celular.

La figura 1.6 muestra el esquema representativo de una neurona, donde se involucran todos sus elementos. En ella, existe un fenómeno que se presenta cuando se realiza la unión entre el axon de una neurona y la dendrita de la siguiente, al que se le conoce con el nombre de sinápsis (aunque en realidad, durante este fenómeno el axon y la dendrita no se tocan). El impulso que se provoca debido a esta "unión" sólo puede atravesar en el sentido axon-dendrita.

La sinápsis sirve de válvula para impedir el flujo inverso de impulsos. La transmisión a través de una sinápsis es considerablemente más lenta que la transmisión a lo largo de un nervio. Cabe mencionar que la conexión sináptica afecta únicamente a áreas limitantes de las neuronas que participan.

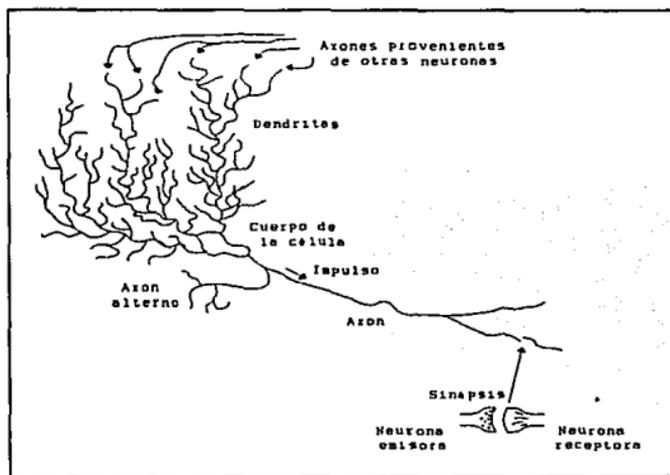


Fig.1.6.- Esquema de una neurona real y sus componentes.

Comenzando a analizar lo que son en sí las redes neuronales artificiales, tenemos que éstas se constituyen de patrones de entrada y salida, de elementos de procesamiento (EP), de conexiones de peso y operaciones preestablecidas.

La figura 1.7 muestra una red neuronal artificial típica, en ella se observa que cada uno de sus niveles se compone de un conjunto de elementos de procesamiento. cada uno de estos elementos reúne valores que provienen de los conectores de entrada, realizando así operaciones matemáticas predefinidas para finalmente producir un valor de salida. Cabe mencionar que cada uno de los elementos que componen a las redes serán descritos en capítulos posteriores.

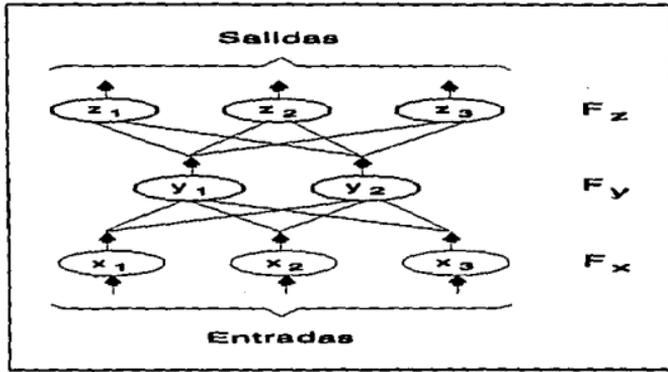


Fig.1.7.- Esquema de una red neuronal artificial y sus principales componentes.

Las funciones básicas de las redes neuronales artificiales son las siguientes:

CLASIFICACION.

Esta función consiste en que un vector de entrada es transmitido a la red, para el cual se produce una salida que es asociada a una clase representativa definida con anterioridad.

IGUALACION DE VECTORES.

En esta función al igual que en la anterior, se introduce un vector de entrada, para el cual se produce el vector de salida correspondiente.

COMPLEMENTACION DE VECTORES.

A la entrada de la red se introduce un vector incompleto, obteniendo a la salida un vector complementado con las partes que se carecían a la entrada.

ELIMINACION DE RUIDO.

Esta función consiste en eliminar el ruido que presenta un vector a la entrada

de la red, obteniendo a la salida un vector similar al de la entrada pero libre de ruido.

OPTIMIZACION.

A los vectores de entrada que presentan problemas en sus valores iniciales, la red proporciona como resultado un conjunto de variables que nos da una solución al problema.

CONTROL.

Teniéndose en la entrada vectores que representan el estado actual de un controlador, así como la respuesta deseada del mismo, la red da como salida una secuencia adecuada de señales las cuales crearán la respuesta correcta.

Entre las características importantes y representativas de las redes neuronales artificiales se encuentran las siguientes:

- Cada elemento de procesamiento que conforma a la red actúa de forma independiente.
- Necesariamente los elementos de procesamiento requieren de recibir información para realizar correctamente su función, además, un elemento de procesamiento no requiere conocer el estado de algún otro elemento.
- El trabajar con un gran número de conexiones trae como consecuencia un valor de salida redundante y elevado, pero también facilita la representación mediante una mejor distribución.

1.3. DEFINICION.

En el campo de los sistemas inteligentes se ha manejado un sinnúmero de definiciones referentes a las redes neuronales artificiales, de entre las cuales se encuentran las siguientes:

- " Las redes neuronales artificiales pueden ser vistas como una caja negra la

cual acepta entradas y produce salidas ⁵. Este concepto hace referencia de manera general de lo que es una red neuronal, a la que considera como un conjunto de elementos que se encuentran alojados en el interior de una caja, la cual recibe información que procesa para finalmente producir salidas, la figura 1.8 muestra la idea de este concepto.

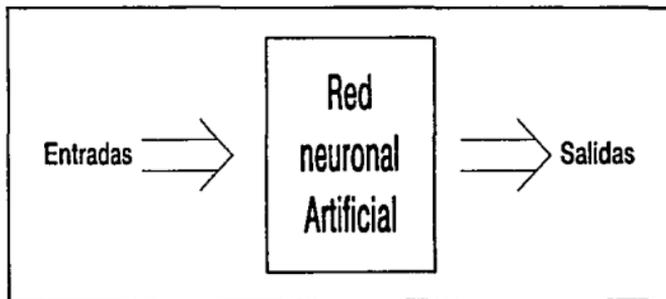


Fig.1.8.- Modelo de la caja negra, propuesto por Patrick K. Simpson.

* " Una red neuronal artificial es un conjunto de algoritmos que tienen como fin el de resolver tareas cognitivas, como lo es el aprendizaje y la optimización⁶. Este concepto se inclina hacia la relación que existe entre las funciones que realizan las redes neuronales artificiales y el cerebro.

Una de las definiciones más aceptables sobre las redes neuronales artificiales es la que maneja Drew Van Camp:

* " Una red neuronal artificial es un modelo de computadora inspirado en la estructura y estudio de las neuronas reales. Como el cerebro, estas redes pueden reconocer patrones, reorganizar datos y, lo más interesante, aprender ⁷.

⁵Patrick K. Simpson. "Foundations of Neural Networks", pag. 3.

⁶David S. Touretzky. "Advanced in Neural Information Processing system", vol. II.

⁷Drew Van Camp. "The Amateur Scientist", Scientific American, pag. 125.

Agregando a todos los conceptos, los investigadores llegan a un punto en común, que las redes neuronales artificiales están hechas de elementos llamados unidades, las cuales representan a los cuerpos de las neuronas biológicas y realizan las funciones correspondientes de los axones y las dendritas; estas unidades se conectan mediante enlazadores provocando el fenómeno sináptico.

Semejante a la fuerza de conexión que existe en una sinápsis, el enlazador multiplica la salida de una unidad por un **FACTOR DE PESO**⁸ para posteriormente pasarlo a otra unidad y sumarlo con los valores provenientes de otros enlazadores.

Un fenómeno importante es el disparo o excitación de una neurona, que análogamente, en las redes neuronales artificiales ocurre cuando el valor total de la suma de los enlazadores de entrada, excede a un valor ya preestablecido, para entonces, los factores de peso que se encuentran en los enlaces sufren modificaciones; y a todo este proceso se le conoce como aprendizaje de la red.

UNIDADES

Existen tres tipos diferentes de unidades, de las cuales pueden estar compuestas las redes neuronales artificiales: las unidades de entrada, que son las encargadas de llevar la información proveniente del exterior, al interior de la red. Las unidades de salida, cuya función es la de enviar las señales ya procesadas y visibles, a la salida. El tercer tipo de unidad no la contemplan todas las redes en su diseño (como es el caso de las redes de uno y dos niveles), estas son las unidades ocultas, que tienen como función la de actuar como intermediario, es decir, reciben las señales provenientes de las unidades de entrada, la procesan y finalmente producen un valor dirigido a las unidades de salida.

En la figura 1.9 se muestra un modelo de red sencilla, en donde los números que se encuentran en las unidades representan los valores de entrada preestablecidos. Los valores que se encuentran sobre la línea de conexión representan los pesos asignados a los enlazadores aunque algunos de estos utilizan un atajo, es decir, pueden desviarse o pasar por alto las unidades ocultas.

⁸ Para las redes neuronales artificiales, los pesos son físicamente magnitudes eléctricas, representadas por valores numéricos, que al ser implementados a los patrones de entrada, se produce un cambio en dicho patrón.

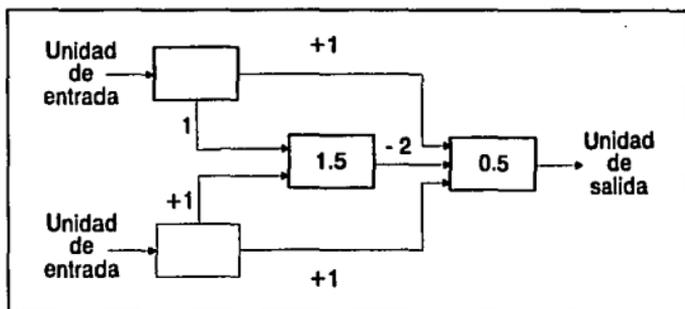


Fig.1.9.- Esquema que muestra una red neuronal artificial con implementación de valores en sus pesos y entradas.

En la figura 1.9 se observa el siguiente caso: si alguna de las unidades de entrada presenta un valor de 1 y la otra un valor de 0, el valor de entrada a la unidad oculta será de 1, eso se debe a que $(1 * 1) + (0 * 1) = 1$. Si este valor fuera menor que el valor preestablecido (0.5), la unidad oculta no se dispararía. En tanto, el valor que puede encontrarse en la unidad de salida puede ser: $(1 * 1) + (0 * (-2)) + (0 * 1) = 1$, el cual es mayor al valor preestablecido, por lo que se dice que la unidad de salida se dispara.

La lógica que se utiliza en este modelo de red neuronal es semejante al comportamiento de una compuerta **OR Exclusiva (EXOR)**. En resumen, si tan sólo una de las unidades de entrada contiene el valor de 1, la red siempre producirá en su salida un valor 1, de lo contrario, se presentara un valor 0. Esto significa que la unidad de salida no se disparará.

Finalmente podemos decir, que modelar una red neuronal artificial implementandola mediante una compuerta EXOR no es particularmente interesante, debido a que seleccionando cuidadosamente los pesos y entradas de la red se puede asemejar cualquier función lógica. Lo que es realmente interesante de tales redes es que no es necesariamente elegir los pesos y las entradas, ya que estos pueden comenzar con cualquier valor.

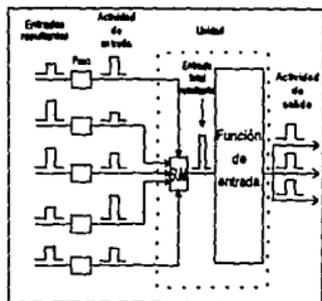
Posteriormente, la red comienza a mostrar de forma repetida los vectores de entrada y salida así como los pesos necesarios para implementarlos a la compuerta

EXOR; de esta manera la red se encuentra en proceso de aprendizaje. Por lo que, para un gran conjunto de datos, la red es capaz de reconocer vectores que no haya analizado anteriormente.

Capítulo



ELEMENTOS DE UNA RED ARTIFICIAL



INTRODUCCION

Las redes neuronales artificiales tienen en su estructura dos componentes básicos, los cuales son los **Conectores** y los **Elementos de Procesamiento**, que dependiendo de las necesidades resultantes a partir de la implementación y del diseño de la red, será la cantidad utilizada de estos componentes.

Además dependiendo de la aplicación que se le destine a la red, se presenta la necesidad de utilizar otros componentes además de los antes mencionados, como son el caso de las **Funciones Prestablecidas**, de los **Vectores de Entrada** y los **Vectores de Salida**.

Para comprender lo que son los componentes de una red neuronal artificial, haremos una analogía entre una gráfica dirigida y una red neuronal artificial, para esto tenemos que una gráfica dirigida contiene en su estructura trazos conocidos como **Conexiones**, que sirven para enlazar sus nodos o elementos de procesamiento, y que permiten el flujo de la información en una sola dirección, siendo esta representada por una flecha; de esta forma, el flujo de la información se lleva a cabo mediante los trazos y llegan hasta los elementos de procesamiento que es en donde se realizan las operaciones.

Acerca de los componentes de una red neuronal artificial es conveniente dar a conocer la terminología que se utilizará en la descripción posterior de cada elemento.

Para ilustrar esta terminología, considere la figura 2.1.

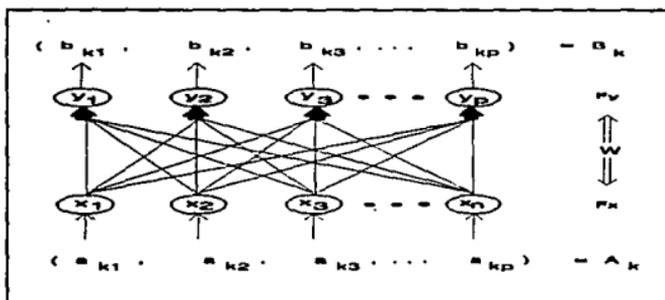


Fig.2.1.- Estructura de una red neuronal artificial de 2 niveles.

En la figura 2.1 se observa la estructura de una red neuronal artificial, la cual consta de dos niveles (nivel de entrada y nivel de salida). Dentro de cada nivel se encuentran los elementos de procesamiento que en conjunto forman vectores de entrada y salida, y cuya función es la de manejar la información proveniente del nivel de entrada y enviar una respuesta hacia la salida.

La nomenclatura para describir a los vectores de entrada y salida se denotará con las primeras letras del alfabeto (por ejemplo A, B, C , etc) acompañadas de un subíndice, el cual servirá para indicar la ubicación del vector dentro la red neuronal. De esta manera, los vectores de entrada se representarán por la siguiente ecuación:

$$A_k = (a_{k1}, a_{k2}, a_{k3}, \dots, a_{kn}) \quad \text{donde } k = 1, 2, 3, \dots, m$$

y los vectores de salida por la ecuación :

$$B_k = (b_{k1}, b_{k2}, b_{k3}, \dots, b_{kp}) \quad \text{donde } k = 1, 2, 3, \dots, m$$

Los Elementos de Procesamiento que se encuentren en un nivel cualquiera, se representan con la misma variable que le fue asignada a dicho nivel. Por su parte, al conjunto de elementos de procesamiento de un nivel se le conoce como **Vector de Elementos de Procesamiento (PE's)** al que se le interpreta con las últimas letras del alfabeto(por ejemplo **T,U,V**, etc), aunque en muchos de los casos, el utilizar 3 niveles de elementos de procesamiento, son los suficiente para formar una red neuronal artificial aceptable. De esta manera, los elementos de procesamiento que se encuentren localizados en el nivel de entrada se representarán mediante la siguiente ecuación:

$$F_x = (X_1, X_2, \dots, X_n)$$

De donde cada elemento x_i recibe información de su correspondiente entrada que se denotará como a_{ij} . El siguiente nivel de elementos de procesamiento será indicado como F_y , y así sucesivamente. Volviendo a la figura 2.1, se observa que el segundo nivel de la red es considerado como el nivel de salida, que se representa con la siguiente ecuación:

$$F_y = (y_1, y_2, \dots, y_n)$$

De donde cada elemento y_j esta correlacionado con el j -ésimo elemento de B_x .

Los pesos conectados entre los niveles de la red neuronal, que como se mencionó en el capítulo anterior sirven para modificar los valores que circulan en las conexiones, son almacenados en un arreglo matricial, denominado como **Matriz de Peso**, la cual puede estar representa por cualquier letra intermedia del alfabeto (**T, U, V**, etc).

Para el caso de la figura 2.1, que muestra una red neuronal de dos niveles, tenemos que su respectiva matriz de pesos es como la que se ilustra en la figura 2.2, que sirve para representar las correspondientes conexiones entre los elementos de procesamiento del nivel de entrada señalado como F_x y los del nivel F_y (nivel de salida). De acuerdo a la matriz de pesos de la figura 2.2, el valor de W_{11} , significa la conexión de los pesos de los elementos de procesamiento de x_1 con el de y_1 .

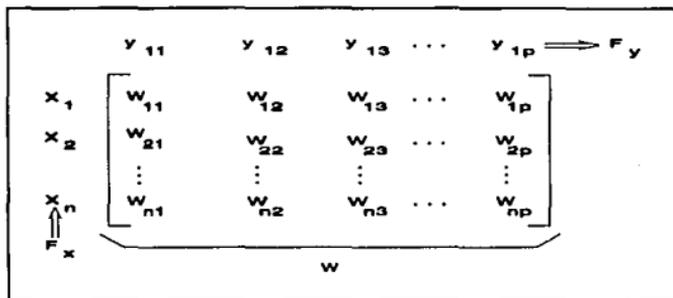


Fig.2.2.- Matriz de peso de una red neuronal de dos niveles.

2.1. VECTORES DE ENTRADA, VECTORES OCULTOS Y VECTORES DE SALIDA.

Como se había analizado anteriormente, las redes neuronales artificiales están compuestas por un gran número de unidades interconectadas, de manera que estas a su vez forman Vectores, que dependiendo de su ubicación o nivel en que se encuentren dentro de la red neuronal, se les denominan como **Vectores de Entrada**, **Vectores Ocultos** o **Vectores de Salida**.

Dependiendo del número de niveles que conformen a la red neuronal, tendrá esta un cantidad proporcional de niveles ocultos, y a su vez, un número determinado de vectores de salida así como los vectores de entrada.

Como se puede observar en la figura 2.3, el proceso de sinápsis se representa mediante un peso que puede ser modificado, el cual es asociado con cada una de las conexiones.

La mayoría de los modelos de redes neuronales artificiales presentan dos desventajas importantes y básicas: por un lado, no tienen la capacidad de proporcionar la geometría más acertada del funcionamiento de las dendritas y los axones, y por el otro, el de asemejar o interpretar las salida eléctrica, como la que se produce en una neurona biológica, como un simple valor numérico que representa la magnitud de disparo.

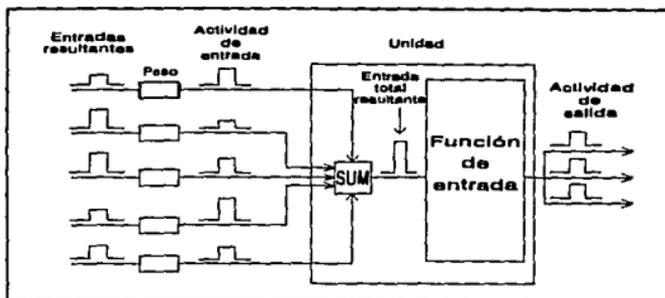


Fig.2.3.- Proceso de sinápsis en una red neuronal artificial.

De la figura 2.3 se puede observar que cada unidad transforma una actividad proveniente del exterior en un vector de entrada, siendo éste a su vez transmitido a otras unidades; este proceso se lleva a cabo durante dos fases: Primero, el vector de entrada es multiplicado por un factor de peso colocado sobre su conector, dando origen a un valor denominado **Entrada Resultante**, posteriormente se suma simultáneamente con las otras entradas, formándose así, un valor denominado **Entrada Total Resultante**. Segundo, una unidad hace uso de una **Función de Entrada-Salida** con el fin de transformar el valor de entrada total resultante a un valor de salida, llamado **Actividad de Salida**.

La **Función Entrada-Salida** y los pesos que se manejan en una red neuronal en perfecta combinación, determinan el desempeño que logre la red. Por su parte, la función **Entrada-Salida** puede encontrarse en cualquiera de las siguientes tres categorías: **Lineal**, **Umbral** o **Sigmoidal** (en función de Sigma).

En caso de que la función se encontrase en la categoría de unidades lineales, tendríamos que la actividad de salida de la red neuronal será proporcional al total de las entradas. Para el segundo caso, el de las unidades umbrales, la salida únicamente puede tomar uno de dos niveles, el que será determinado dependiendo si el valor de la entrada total resultante sea mayor o menor que alguno de los valores preestablecidos. Por último, para el caso de las unidades de tipo sigmoidal, el valor de salida, constantemente se encuentra en estado de variación, y esto trae como consecuencia ciertos cambios en la entrada; los casos anteriores se muestran en la figura 2.4.

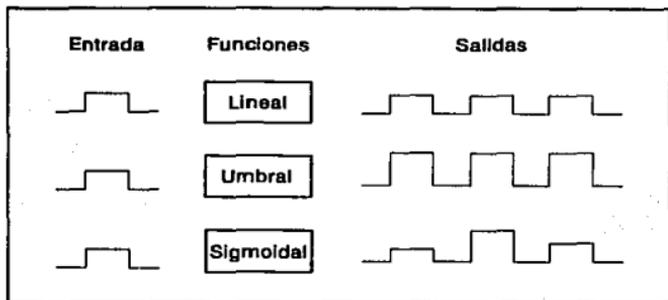


Fig.2.4.- Función Entrada-Salida en sus tres categorías: Lineal, Umbral y Sigmoidal.

De las tres categorías anteriores, la que se considera más próxima al comportamiento de las neuronas biológicas, es la de las unidades sigmoidales, aunque las tres son consideradas únicamente como aproximaciones al comportamiento real.

La clase de red neuronal artificial que con mayor frecuencia se diseña y se construye es la que se estructura de tres grupos o niveles, que son interconectados de la siguiente manera:

El Nivel que contiene a las unidades de entrada se conecta al nivel de las unidades ocultas, siendo éste a su vez conectado con el nivel formado con las unidades de salida. Por su parte, el funcionamiento de las unidades de entrada consiste en suministrar la información del exterior o de realimentación más reciente. En tanto, la función de las unidades ocultas se va componiendo de acuerdo a las actividades que realizan las unidades de entrada en combinación con los pesos que han sido colocados sobre las conexiones que se encuentran entre las unidades de entrada y las unidades ocultas.

De una forma muy similar, el comportamiento de las unidades de salida se deriva tanto de las actividades de las unidades ocultas como de los pesos colocados sobre las conexiones entre ambas unidades. Este tipo de red además de ser uno de los más sencillos, es uno de los más interesantes, debido a que las unidades ocultas tiene la capacidad de construir sus propias interpretaciones acerca de las actividades de entrada, de esta manera, mientras los pesos sufran ciertas modificaciones, darán otras interpretaciones diferentes sobre las actividades de entrada.

Una situación muy interesante, es que no todas las redes neuronales artificiales operan con algunos tipos comunes de vectores, es decir, algunos sistemas neuronales requieren para su operación de patrones simples, a estos sistemas se les conoce con el nombre de Red Neuronal Artificial Autoasociativa; pero también existen los sistemas denominados Redes Neuronales Artificiales Heteroasociativas, debido a que utilizan para su funcionamiento patrones pares.

Para las redes del tipo autoasociativa tenemos a un conjunto de patrones que se encuentran previamente almacenadas en la red, los cuales son comparados con el patrón de entrada y si alguno es similar, entonces ambos patrones se asocian, en la figura 2.5 se muestra este tipo de red, en la cual, un cuadrado incompleto, es asociado con el cuadrado que se encuentra en el interior de la red.

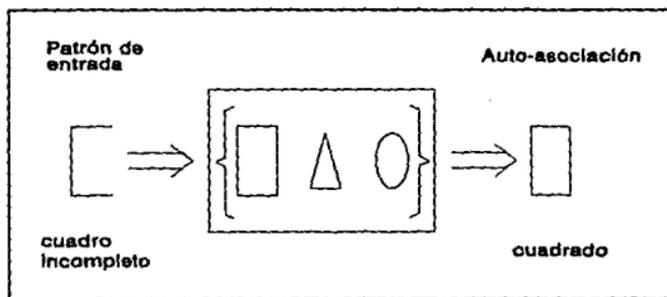


Fig.2.5.- Red neuronal artificial autoasociativa.

En las redes heteroasociativas, los patrones que se encuentran en su interior, están agrupados en pares, y de igual manera que en la red autoasociativa, se compara la entrada con los patrones almacenados, asociándole el par (romboide) que le fue asignado al patrón que se le asemeja (cuadrado). Esto se observa en la figura 2.6.

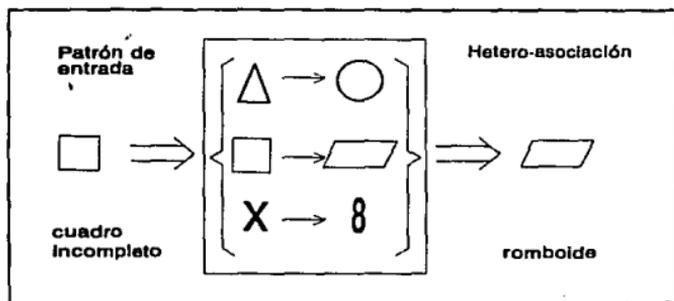


Fig.2.6.- Red neuronal artificial heteroasociativa.

Debido al tipo de procesamiento que realice la red neuronal, existen básicamente dos tipos de redes que son: las que únicamente tienen la capacidad de procesar datos en código binario; este tipo de máquinas se pueden ver tanto en el modelo de la Red Neuronal Artificial de Hopfield⁹ perfeccionado por Amari en 1972, como en el modelo llamado El Cerebro como una Simple Caja de Anderson¹⁰.

Por el contrario, otros modelos de redes neuronales artificiales, tienen la característica de procesar datos con valores reales, como es el caso del Método de Realimentación hacia Atrás de Williams¹¹, que será tema de análisis en un capítulo posterior, y del Vector de Cuantización para el Aprendizaje de Kohonen¹², cuyo propósito es el de producir un código obtenido de un patrón de entrada de n dimensiones, el cual es transmitido a través de un canal y utilizado para reconstruir la entrada original con un mínimo de distorsión.

⁹J.J. Hopfield. "Neural Network and Physical Systems with Emergent Collective Computational Abilities". Proc. Natl. Acad. Sci., vol. 79, Apr., 1982.

¹⁰Anderson, J. J., Silverstein, S., Ritz, and R. Jones. "Distinctive Features, Categorical Perception, and Probability Learning: Some Applications of a Neural Model". Psych. Rev., vol. 84, 1977.

¹¹Jack M. Zurada. "Introduction to Artificial Neural Systems". West Publishing Company, USA, 1992.

¹²Kohonen, T. "Learning Vector Quantization for Pattern Recognition". Helsinki University of Technology, 1986.

2.2. CONEXIONES

El propósito de las conexiones dentro de una gráfica dirigida es la de determinar la dirección del flujo de la información, como se muestra en la figura 2.7. El flujo de la información que se realiza del nivel F_x a el nivel F_y se hacen mediante las conexiones representadas con la letra W . En los digrafos de las redes neuronales se les asigna un peso a cada trazo (conexión), que representa el valor total de la señal de salida que se trasladará de un nodo (elemento de procesamiento) inferior a un nodo adyacente.

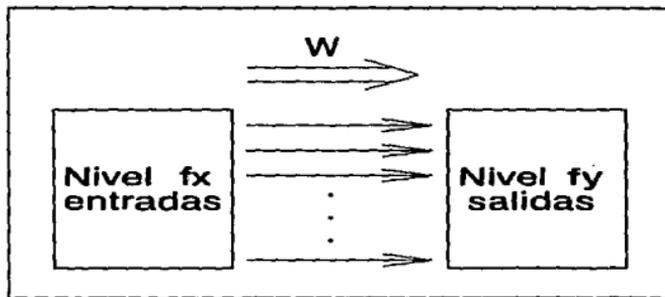


Fig.2.7.- Conjunto de conexiones de una red neuronal artificial.

Como se aprecia en la figura 2.7, se utiliza una flecha doble para indicar el sentido del flujo de la información. Una conexión tiene fundamentalmente dos funciones, por un lado la de indicar la dirección del flujo de la información, y por el otro, de modular la cantidad de información que fluye entre los elementos de procesamiento.

En tanto, las conexiones denominadas excitatorias son las que contienen pesos con valores positivos, y por el contrario, cuando estos pesos tienen un valor negativo se les denomina como conexiones inhibitorias. Cuando se presenta el caso de que no existe conexión alguna se debe a que el peso de esta conexión presenta un valor de cero o nulo.

Para el diseño de redes neuronales artificiales no es conveniente el uso continuo de conexiones que presenten el valor cero, a excepción de que estas conexiones se encuentren esparcidamente y en números pequeños. Durante el funcionamiento de una red neuronal artificial, una de las condiciones más deseables es la de lograr que los elementos de procesamiento mantengan valores parciales internos (valores preestablecidos).

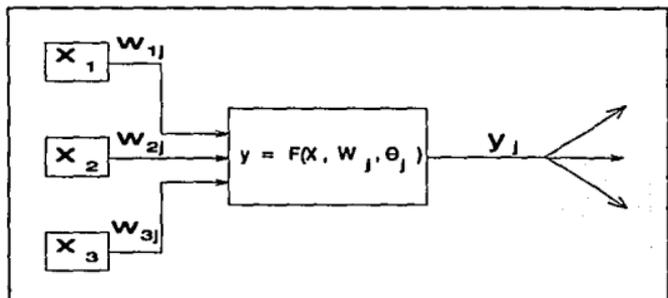


Fig.2.8.- Elemento de Procesamiento con tres conexiones.

En la figura 2.8 en donde se observa que el elemento de procesamiento que es señalado como y_j esta en función de tres conexiones representadas por la siguiente ecuación: $F_x (W_1, W_2, W_3)$ y por un valor preestablecido representado por θ_j .

Como se observa en la figura 2.9, para el mejor desempeño de la red es importante que el valor preestablecido θ_j sea utilizado como una conexión extra, marcada como w_0 y proveniente de la expresión F_x PE y x_0 , considerando que el valor de x_0 siempre tomará el valor de uno.

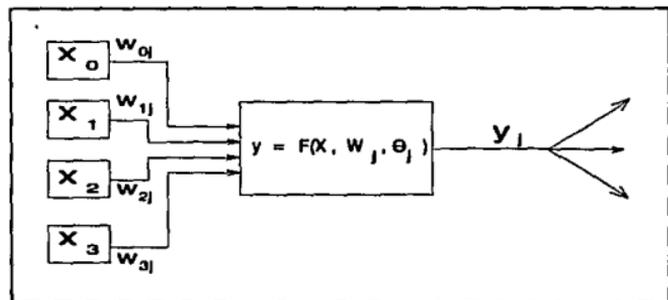


Fig.2.9.- Elemento de Procesamiento con conexión extra (w_0).

2.3. ELEMENTOS DE PROCESAMIENTO (P.E.)

Los elementos de procesamiento son los elementos más importantes y básicos que componen a una red neuronal artificial. Esta importancia se debe a que en el interior de estos elementos se ponen en funcionamiento los componentes restantes de la red, para realizar las operaciones que nos lleven a los resultados esperados. Uno de los elementos de procesamiento que se usa con mayor frecuencia se muestra en la figura 2.9.

Además, los elementos de procesamiento pueden encontrarse en dos situaciones diferentes: la primera de ellas, se presenta cuando un elemento de procesamiento se encuentra en el nivel de entrada y por consiguiente recibirá un solo patrón de entrada, como se observa en la figura 2.10(a). La segunda ocurre cuando cada elemento de procesamiento que se encuentre en un nivel que no sea el de entrada, tiene varias conexiones de pesos provenientes de otros elementos de procesamiento de diferente nivel, como se ilustra en la figura 2.10(b).

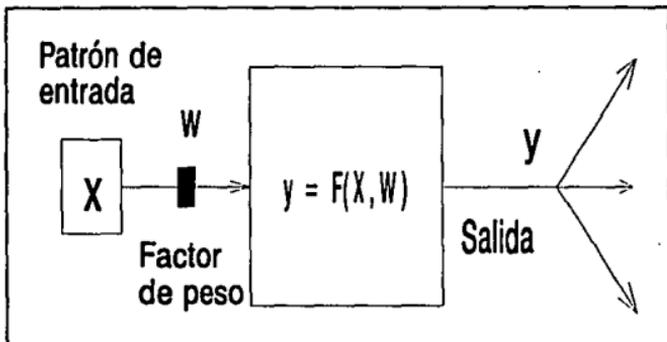


Fig.2.10(a).- Elemento de Procesamiento con una sola conexión de peso.

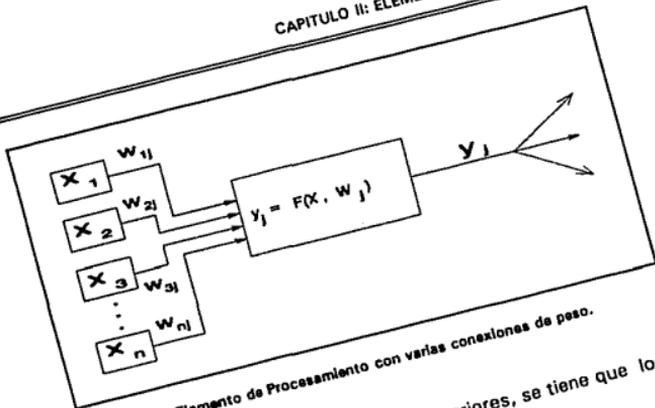


Fig.2.10(b).- Elemento de Procesamiento con varias conexiones de peso.

Como consecuencia de las situaciones anteriores, se tiene que los elementos de procesamiento tienen dos cualidades importantes:

- 1.- Los elementos de procesamiento necesitan únicamente de información local ubicada en las entradas de estos, para posteriormente almacenarlas en el interior de dicho elemento y finalmente producir un valor de salida. Una ventaja muy importante de esta característica es que ya no se requiere de una información extra para un mejor desempeño.
- 2.- Durante su funcionamiento, los elementos de procesamiento utilizan los valores de entrada para procesarlos y producir un sólo valor de salida, transmitiéndose mediante las conexiones hacia un elemento de procesamiento receptor o bien puede ser utilizado como valor de salida de la red.

Como se hizo anteriormente con las conexiones, el valor que se le asigna a cada elemento de procesamiento así como a la etiqueta que lo representará, serán denotados con una misma variable, por ejemplo, para representar el j -ésimo elemento de procesamiento que se localiza en el nivel F_j se utiliza la expresión y_j y por consiguiente el valor que tenga este elemento será también de la forma y_j .

El valor de salida de un elemento de procesamiento esta en función de las entradas y los pesos que lleguen a él, como se puede observar claramente en la figura 2.9, en donde y_j representa la función de las salidas producidas en un nivel precedente, denotada por la ecuación $F_x = X = (x_1, x_2, \dots, x_n)$ y de los pesos localizados desde el nivel F_x hasta el elemento y_j , es decir, $W = (w_1, w_2, \dots, w_n)$; de acuerdo a lo anterior, la ecuación que representa al elemento y_j es:

$$y_j = F(X, W_j). \quad (1)$$

Existen básicamente tres tipos de funciones que se utilizan para el diseño de los elementos de procesamiento, de acuerdo a la aplicación que se les vaya a asignar a estos, las cuales son las siguientes :

FUNCION POR COMBINACION LINEAL.-De los métodos utilizados para el diseño de los elementos de procesamiento el más común es el conocido como **Combinación Lineal** llamado también **Producto Punto**. Este método consiste en realizar el producto punto con los valores de entrada X y los pesos de conexión W_j , para ilustrar este método utilizaremos el elemento de procesamiento mostrado en la figura 2.9 en donde la salida y_j está expresada por la siguiente ecuación:

$$y_j = f \left[\sum_{i=0}^n x_i W_{ij} \right] = f(X, W_j) \quad (2)$$

De la ecuación 2 tenemos que $W_j = (w_{1j}, w_{2j}, \dots, w_{nj})$, y f es una de las funciones no lineal del elemento de procesamiento. Una de las características más importantes del producto punto es que este es muy usado en la computación.

FUNCION CON CONEXIONES DOBLES (MEDIA-VARIANZA).-En algunas ocasiones, en la representación de un elemento de procesamiento se utilizan dos conexiones interconectadas a dicho elemento como se aprecia en la figura 2.11. Esta doble conexión tiene dos funciones, por un lado la de representar el

valor de la clase, y por el otro, la varianza que existe en dicha clase¹³, entendiéndose por varianza, al valor medio de las desviaciones cuadráticas de cada una de las variables.

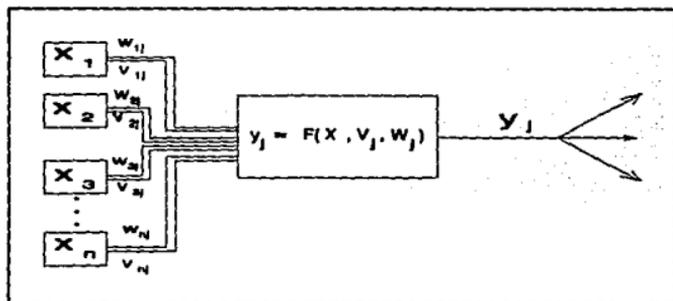


Fig.2.11.- Elemento de Procesamiento con conexiones dobles (medio-varianza).

Para este tipo de conexiones, tanto el valor de las entradas como el valor de las conexiones determinan el valor de la salida del elemento de procesamiento, es decir, $y_j = F(X, V_j, W_j)$, de donde las conexiones medias se representan con la expresión $W_j = (w_{1j}, w_{2j}, \dots, w_{nj})$ y las conexiones que llevan el valor de la varianza se denota con la expresión $V_j = (v_{1j}, v_{2j}, \dots, v_{nj})$. El valor de la salida y_j se determina calculando la diferencia entre el valor de la entrada X y el valor de peso W_j , dividiéndola entre la varianza V_j y elevar el resultado al cuadrado, esto se hace para transformar el valor resultante mediante la función Gaussiana no lineal del elemento de procesamiento, y obtener finalmente el valor de salida representado como :

$$y_j = g \left[\sum_{i=1}^n \left(\frac{w_{ij} - x_i}{v_{ij}} \right)^2 \right] \quad (3)$$

¹³ Robinson, A., M. Niranjan, and P. Fallside. "Generalizing the Nodes of the Error Propagation Network". Cambridge University Engineering Department, 1988.

De la ecuación anterior se obtiene que la función no lineal Gaussiana del elemento de procesamiento está dada por:

$$g(x) = \exp\left(\frac{-x^2}{2}\right) \quad (4)$$

FUNCION CON CONEXIONES MIN/MAX.-Este tipo de conexiones son usadas con menos frecuencia, a pesar de que utilizan también conexiones dobles. El objetivo de estas funciones, es la de asignar como límite inferior de la clase a uno de los vectores pertenecientes al conjunto V_i , y de forma similar asignar como límite máximo de la clase, a uno de los vectores pertenecientes a W_j . Los patrones de entrada cuyo valor se encuentre dentro de estos límites, producirán un valor conocido como **Valor de Activación Min/Max**¹⁴. En la figura 2.12 se muestra la gráfica de los puntos mínimo y máximo antes mencionados.

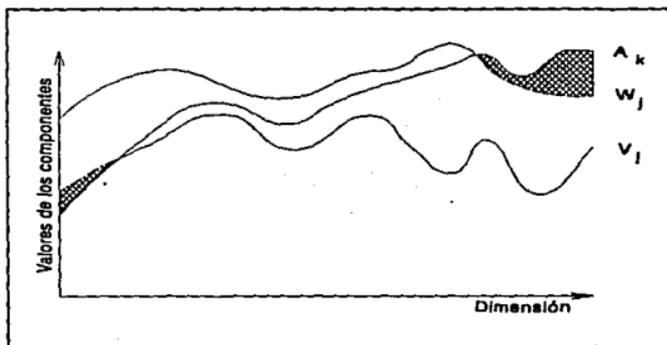


Fig.2.12.- Elemento de Procesamiento con conexiones Min/Max.

¹⁴ Simpson, P. "Fuzzy Min-Max Neural Networks: Classification". IEEE Trans. Neural Networks, in Press, 1991.

En el eje de las ordenadas se localizan los valores de cada elemento que pertenecen a su correspondiente vector, mínimo o máximo, mientras que en el eje de las abscisas se encuentran los valores que representan la dimensionalidad del espacio de ambos vectores. La región sombreada en la gráfica representa la desviación que existe entre los límites de clase de los vectores V_j y W_j con respecto al valor de entrada. Esta diferencia se realiza mediante la comparación entre los límites de clase y el patrón de entrada. Además de representar la desviación V_j , la región sombreada muestra los momentos en los que los elementos de procesamientos producen valores de activación y_j .

Como se observa en la figura 2.12, los puntos W_j y V_j representan respectivamente al límite máximo y límite mínimo permisibles en la clase j . La magnitud relativa del patrón de entrada X que cae fuera de la clase j indica el grado con el cual ese valor no cae entre los límites máximo y mínimo.

2.4. FUNCIONES P.E.

Existe una gran variedad de funciones P.E., cuyo número posiblemente sea infinito, pero básicamente son cinco las funciones más utilizadas en las redes neuronales artificiales, las cuales son:

- 1) Función P.E. Lineal.
- 2) Función P.E. Escalón.
- 3) Función P.E. Rampa.
- 4) Función P.E. Sigmoidal.
- 5) Función P.E. Gaussiana.

De las funciones anteriores, a excepción de la primera, todas presentan un comportamiento no lineal durante el desempeño de la red neuronal, dependiendo del grado de limitación que se les da a los valores de salida por medio de un rango preestablecido. A continuación se dará una breve descripción de estas funciones:

FUNCION P.E. LINEAL.-Teniendo un valor x descrito por la ecuación:

$$f(x) = \alpha x \quad (5)$$

se obtiene a la salida una señal modulada linealmente, producida por la función P.E lineal. De la ecuación (5) es importante considerar que el valor de entrada x se encuentra en el dominio de los números reales, en tanto, el valor de α sólo puede tomar valores escalares positivos. Una gráfica de esta función se muestra en la figura 2.13.

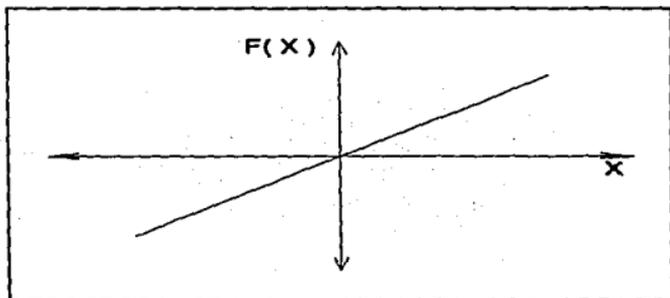


Fig.2.13.- Representación gráfica de la función P.E. Lineal.

FUNCION P.E. ESCALON.-Este tipo de función esta representada por la siguiente ecuación:

$$f(x) = \begin{cases} \beta & \text{si } x \geq \theta \\ -\delta & \text{si } x < \theta \end{cases} \quad (6)$$

que gráficamente se representa en la figura 2.14.

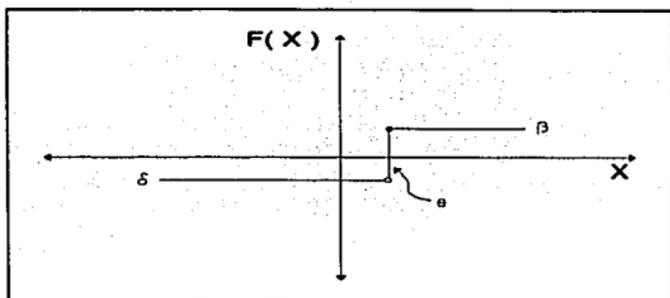


Fig.2.14.- Representación gráfica de la función P.E. Escalón.

La ecuación (6) indica que la función P.E. escalón únicamente puede producir dos valores, si la entrada x es igual o mayor que un valor preestablecido Θ entonces la función P.E produce un valor β , de lo contrario, la función P.E producirá un valor de salida $-\delta$; siendo β y δ valores escalares positivos.

Estas funciones producen salidas propias de un sistema binario, es decir, en respuesta a las señales de entrada, envían un 1 si el valor x es positivo, y se obtiene una salida 0 si no lo es; si se le asigna a β un valor de 1, y 0 tanto a δ como a Θ , se obtiene que la función P.E escalón en términos binarios queda de la siguiente forma:

$$f(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{otro valor} \end{cases} \quad (7)$$

Esta función es la más utilizada para el diseño de las redes neuronales artificiales, tal es el caso del modelo de red neuronal realizado por Hopfield.

Sustituyendo en la ecuación (7) el valor de 0 por un -1 se obtiene la siguiente función P.E llamada Bipolar:

$$f(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ -1 & \text{otro valor} \end{cases} \quad (8)$$

El valor negativo -1 se utiliza para asegurar cambios en los valores de salida, no siendo esto posible utilizando el valor 0.

FUNCION P.E RAMPA.-Esta función es una consecuencia de combinar la función P.E Lineal y la función P.E Escalón. Esta función tiene un límite inferior y un límite superior, que indican cuando la función se encuentra en estado de linealidad, esto ocurre cuando los valores de salida se encuentran entre los límites y normalmente cuando están en las aproximaciones al origen; los puntos de saturación son simétricos alrededor del origen y son discontinuos antes o después de dichos puntos. La función P.E Rampa tiene la siguiente forma:

$$f(x) = \begin{cases} \tau & \text{si } x \geq \tau \\ x & \text{si } |x| < \tau \\ -\tau & \text{si } x \leq -\tau \end{cases} \quad (9)$$

De la ecuación 9, τ representa el valor de saturación de la función, y tanto el punto $x = \tau$ como el de $x = -\tau$ indican los momentos de discontinuidad de la función. La gráfica de esta función es la siguiente:

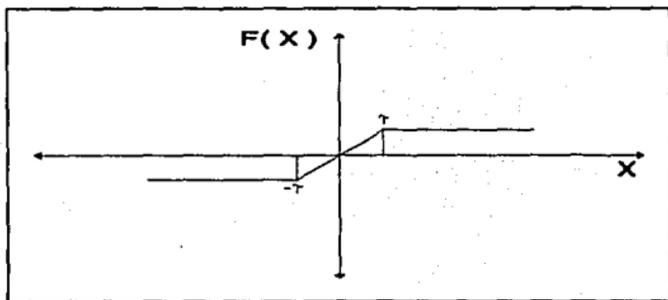


Fig.2.15.- Representación gráfica de la función P.E. Rampa.

FUNCION P.E. SIGMOIDAL.-La función P.E. Sigmoideal, llamada así por tener forma de S, es semejante a la función P.E. Rampa, a excepción de que la segunda tiene puntos de discontinuidad y la primera es una función continua, además es limitada, monótona, no decreciente, y proporciona una respuesta no lineal dentro de un rango preestablecido. Dentro de las funciones sigmoideales existe una que es aplicada con más frecuencia, la cual es conocida como **Función Logística**, que tiene la siguiente forma:

$$f(x) = \frac{1}{1 + e^{-\alpha x}} \quad (10)$$

Gráficamente equivale a:

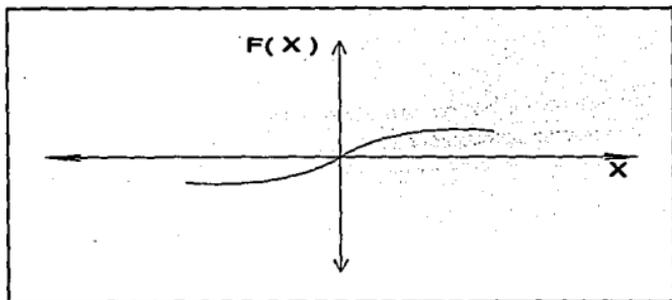


Fig.2.16.- Representación gráfica de la función P.E. Sigmoideal.

De donde $\alpha > 0$, aunque comúnmente α toma el valor de 1; es decir, α proporciona un valor de salida que puede ser desde 0 hasta 1. Algunas de las aplicaciones más comunes que se le da a esta función, es en la Estadística (Función de Distribución Gaussiana), en la Química (en la descripción de reacciones catalíticas), y en la Sociología (para describir el crecimiento de la población).

Observando la relación que existe entre las ecuaciones (10) y (7), tenemos que cuando α es igual a ∞ , en la primera, la pendiente de la señal sigmoideal comienza a irse infinitamente en declive entre los puntos 0 y 1, dando por resultado una función semejante a la Escalón representada por la ecuación (7).

FUNCIÓN P.E. SIGMOIDAL.-La función P.E. Sigmoïdal, llamada así por tener forma de S, es semejante a la función P.E. Rampa, a excepción de que la segunda tiene puntos de discontinuidad y la primera es una función continua, además es limitada, monótona, no decreciente, y proporciona una respuesta no lineal dentro de un rango preestablecido. Dentro de las funciones sigmoïdales existe una que es aplicada con más frecuencia, la cual es conocida como **Función Logística**, que tiene la siguiente forma:

$$f(x) = \frac{1}{1 + e^{-\alpha x}} \quad (10)$$

Gráficamente equivale a:

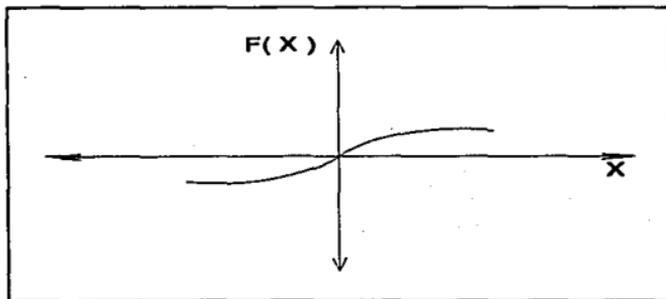


Fig.2.16.- Representación gráfica de la función P.E. Sigmoïdal.

De donde $\alpha > 0$, aunque comúnmente α toma el valor de 1; es decir, α proporciona un valor de salida que puede ser desde 0 hasta 1. Algunas de las aplicaciones más comunes que se le da a esta función, es en la Estadística (Función de Distribución Gaussiana), en la Química (en la descripción de reacciones catalíticas), y en la Sociología (para describir el crecimiento de la población).

Observando la relación que existe entre las ecuaciones (10) y (7), tenemos que cuando α es igual a ∞ , en la primera, la pendiente de la señal sigmoïdal comienza a irse infinitamente en declive entre los puntos 0 y 1, dando por resultado una función semejante a la Escalón representada por la ecuación (7).

En tanto, la función sigmoideal logística tienen otras dos alternativas, la primera, es la de la **Tangente Hiperbólica** representada por:

$$f(x) = \tanh(x) \quad (11)$$

para el rango que va desde -1 a 1; y por el otro, La **Relación Aumentada de los Cuadrados** dada por la expresión:

$$f(x) = \begin{cases} \frac{x^2}{1+x^2} & \text{si } x > 0 \\ 0 & \text{otro valor} \end{cases} \quad (12)$$

con un rango que va desde 0 hasta 1.

FUNCION P.E GAUSSIANA.-Esta función se puede considerar como radial debido a que tiene una simetría en los puntos cercanos al origen. La gráfica de esta función se muestra en la figura 2.17.

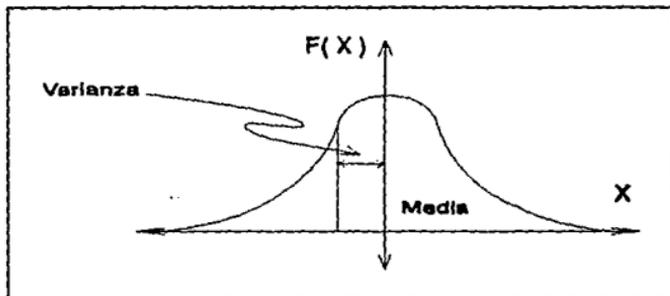


Fig.2.17.- Representación gráfica de la función P.E. Gaussiana.

Para el correcto desempeño de la función P.E Gaussiana se requiere de la utilización de la **varianza** v que debe ser mayor que 0, la cual sirve para formar la función Gaussiana. En algunos casos, en las **redes neuronales artificiales** la función Gaussiana se utiliza en combinación con un conjunto de conexiones dobles, y en otros

casos, la varianza esta preestablecida. En el último de los casos la función P.E Gaussiana está definida por la expresión:

$$f(x) = \exp \left[\frac{-x^2}{v} \right] \quad (13)$$

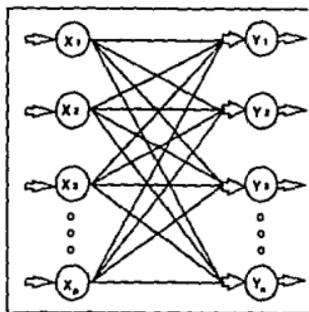
de donde x representa la media, y v la varianza preestablecida.

En resumen, una regla que hay que seguir cuidadosamente para tener el éxito en cualquier aplicación que se le destine a la red neuronal artificial con la que se vaya a trabajar, es la de tener una correcta visión de las características en las que se basará nuestra red, para tener la selección más adecuada de los elementos que compondrán a la misma.

Capítulo



TOPOLOGIA



INTRODUCCION

Las redes neuronales artificiales se componen de niveles, que a su vez contienen elementos de procesamiento interconectados mediante pesos de conexión. La forma en que son conectados dentro de la red estos elementos en combinación con los patrones de entrada, define lo que se conoce como topología.

En principio es conveniente tener conocimiento de algunos términos que se manejarán a lo largo de este capítulo para posteriormente proporcionar una breve descripción de seis tipos diferentes de topologías de redes neuronales artificiales.

Niveles.

Forman parte de las redes neuronales artificiales y contienen a los elementos de procesamiento (PE's). Dentro de cada nivel, los elementos de procesamiento tienen dos propiedades importantes:

- 1) Las conexiones que proporcionan la información a los elementos de procesamiento de un mismo nivel, provienen de un mismo origen, como se puede apreciar en la figura 3.1, en la que los elementos de procesamiento del nivel marcado como F_x reciben la información proveniente de los patrones de entrada, y de la misma forma los elementos de procesamiento del nivel F_y reciben información de los elementos de procesamiento del nivel F_x .
- 2) Los elementos de procesamiento pertenecientes a un mismo nivel manejan la misma información debido a que utilizan tanto la misma función PE como el tipo de conexiones, como se muestra en la figura 3.1.

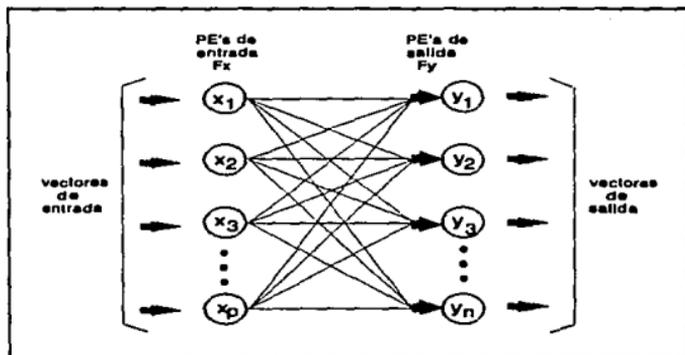


Fig.3.1.- Red neuronal de dos niveles.

Conexiones intraniveles e interniveles.

Existen dos tipos de interconexiones que emplean las redes neuronales artificiales, que son: **Intraniveles** (del latín *intra* "dentro") las cuales conectan los elementos de procesamiento de un mismo nivel; mientras que las conexiones **Interniveles** (del latín *inter* "entre") conectan elementos de procesamiento de diferentes niveles. En la actualidad, es posible que algunos diseños de redes neuronales artificiales contemplen uno o ambos tipos de conexiones.

En la figura 3.2 se observa el caso de una red neuronal con los dos tipos de conexiones.

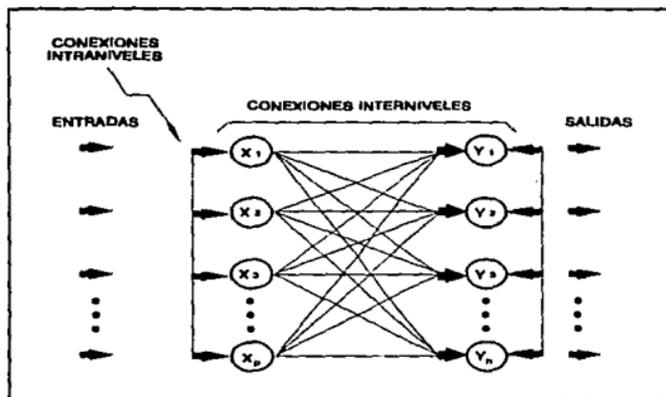


Fig.3.2.- Red neuronal con conexiones intraniveles e interniveles.

Redes de alimentación hacia adelante y hacia atrás.

Las redes neuronales artificiales que tienen alimentación hacia adelante son las que poseen conexiones para trasladar la información en una sola dirección, es decir, de entrada a salida, y que no poseen lazos de retroalimentación¹⁵. En la figura 3.3(a) se muestra la arquitectura de este tipo de red neuronal.

Por el contrario, si durante la trayectoria de la información en una red neuronal existe alguna retroalimentación conectada a uno de los niveles anteriores o a un elemento de procesamiento en particular, entonces se trata de una red de alimentación hacia atrás.

En la figura 3.3(b) se ilustra un red neuronal con este tipo de alimentación.

¹⁵Jack M. Zurada, "Introduction to Artificial Neural Systems", West Publishing Company, USA, 1992.

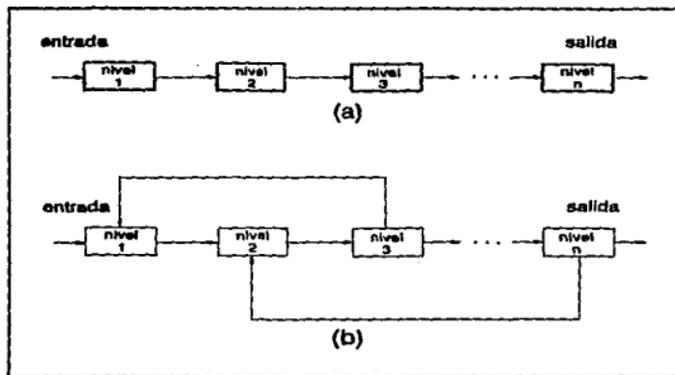


Fig.3.3.- (a)Red de alimentación hacia adelante (b)Red de alimentación hacia atrás.

3.1. INSTARS, OUTSTARS Y ADALINE.

Dos de las topologías básicas de las redes neuronales artificiales, son el *Instar* y el *Outstar*¹⁶. A la primera de ellas, se le considera como Red de Patrón Codificado. Un patrón codificado es el que se obtiene utilizando los valores de los patrones representados por la expresión $A_k = \{a_{k1}, a_{k2}, \dots, a_{kn}\}$ y normalizar cada uno de estos valores, para finalmente interpretarlos y utilizarlos como los pesos de conexión, representados por $W_j = \{w_{j1}, w_{j2}, \dots, w_{jn}\}$. Ahora bien, el vector resultante y que representa al patrón codificado es el siguiente:

$$V_j = \frac{a_{kj}}{\sum_{i=1}^n a_{ki}} \quad (1)$$

donde $i = 1, 2, 3, \dots, n$

La figura 3.4 muestra una red neuronal de tipo Instar.

¹⁶ Grossberg S. "Studies of Mind and Brain". Boston: Reidel, 1982.

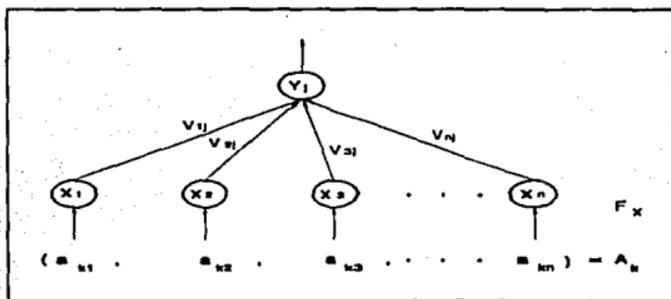


Fig.3.4.- Red neuronal de tipo INSTAR.

Por el contrario, tenemos las redes neuronales artificiales de tipo Outstar, en la cual el patrón de salida que se genera, se expresa con la siguiente ecuación:

$$Z_i = y_i w_i \quad (2)$$

donde $i = 1, 2, 3, \dots, n$

De la expresión anterior tenemos que los pesos indicados por w_i se pueden obtener utilizando la ecuación 1. Debido a esto, decimos que este tipo de red puede ser utilizado como complemento para la red instar, ya que el valor de salida y_i producido por esta red se puede utilizar como valor de entrada y_i para la red outstar.

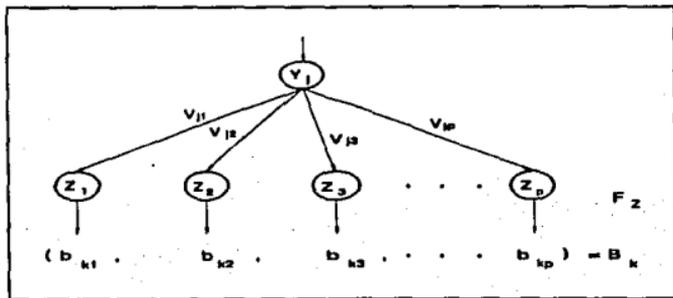


Fig.3.5.- Red neuronal de tipo OUTSTAR.

Por otra parte, tenemos que el tipo de red **ADALINE**¹⁷ (ADaptive Linear Neuron "Neurona Lineal Adaptiva") y el Instars tienen muchas características topológicas semejantes, a excepción de que en la ADALINE los pesos representados por V_j tienen la propiedad de ser modificables.

También tenemos que dentro de la topología de los sistemas de procesamiento adaptivos de señales, se encuentran los **Filtros de Respuesta de Impulso Finito (FIR)**¹⁸, que entre sus aplicaciones básicas se encuentran la cancelación de ruido, la cancelación de ecos, el control de antenas adaptativas, etc. La figura 3.6 nos muestra una red neuronal del tipo ADALINE.

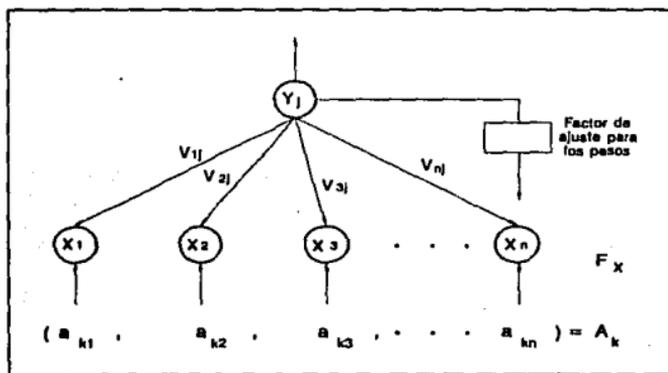


Fig.3.6.- Red neuronal de tipo ADALINE.

3.2. RED DE UN NIVEL

Las redes neuronales artificiales pueden estar construidas de varios niveles, de las cuales la básica es la de un solo nivel, el cual está formado por un conjunto de intraconecciones de sus elementos de procesamiento, como se observa en la figura 3.7.

¹⁷ Widrow B. and M. Hoff. "Adaptive switching circuits". WESCON Convention Record: Part IV, 1960.

¹⁸ Widrow B. and R. Winter. "Neural nets for adaptive filtering and adaptive pattern recognition". IEEE Computer Mag. 1988

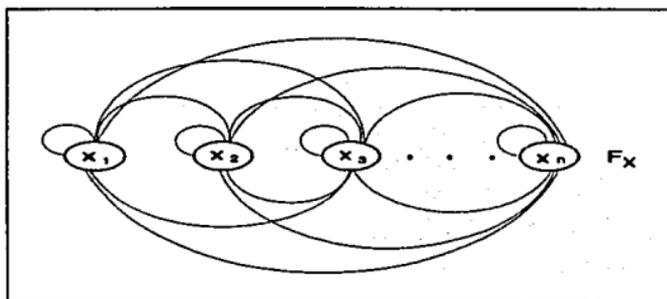


Fig.3.7.- Red neuronal de un solo nivel.

De acuerdo a la figura 3.7 tenemos que este tipo de redes consisten de n elementos de procesamiento en un nivel F_x , en donde además de que las conexiones de cada elemento de procesamiento se hacen con los elementos restantes, pueden reconectarse así mismos, dando origen a una matriz de conexiones con n entradas, por lo que tienen la capacidad de aceptar un patrón de entrada de n^2 dimensiones. Este tipo de redes se pueden encontrar en una de tres diferentes situaciones:

- 1) **Únicamente inicializar a los elementos de procesamiento.**- En este caso, el patrón de entrada se utiliza para inicializar a los elementos de procesamiento únicamente, de manera que no influya en el resto del proceso.
- 2) **Inicializar a los elementos de procesamiento e intervenir como una constante preestablecida.**- En esta situación, el patrón de entrada es utilizado para dos funciones: inicializar a los elementos de procesamiento y crear una constante preestablecida que intervenga durante todo el proceso.
- 3) **Únicamente intervenir como una constante preestablecida.**- Aquí, los elementos de procesamiento son inicializados con un valor de cero previamente a la llegada del patrón de entrada, para permitir que dicho patrón intervenga únicamente como constante preestablecida.

Entre los tipos de procesamiento de patrones que llevan a cabo las redes neuronales de un solo nivel se encuentran las siguientes: para **Complementación de Patrones**, para **Eliminación de Ruido**, para **Optimización Neuronal** y para **Mejoramiento de Contraste**. Los dos tipos primeros tipos de procesamiento funcionan utilizando patrones codificados autoasociables, los cuales utilizan al patrón de entrada para únicamente inicializar a los elementos de procesamiento. En tanto, las redes optimizadas son los sistemas dinámicos cuya función es la de mantener estable el estado de la red, el cual representa la solución del problema, además, utilizan a los patrones de entrada tanto para inicializar a los elementos de procesamiento como para actuar como la constante preestablecida.

Para el caso referente al mejoramiento de contrastes, las redes utilizan a los patrones de entrada para inicializar a los elementos de procesamiento únicamente. A continuación, daremos una descripción del funcionamiento de estos tipos de procesamiento, así como un ejemplo representativo:

Complementación de patrones.

En este tipo de procesamiento, a las redes neuronales artificiales, inicialmente se les presenta a la entrada una parte del patrón a analizar, asumiendo que en base a esta porción, la red sea capaz de complementar al patrón con las partes restantes y así obtener a la salida el patrón completo. Un ejemplo de esto, se da en las redes neuronales diseñadas para almacenar imágenes de cuerpos humanos, como se muestra en la figura 3.8, en donde a la red se le aplica a la entrada una parte o la mitad de un cuerpo, obteniendo a la salida la imagen completa, siendo que previamente la red durante su proceso busca las partes faltantes y las complementa al cuerpo.

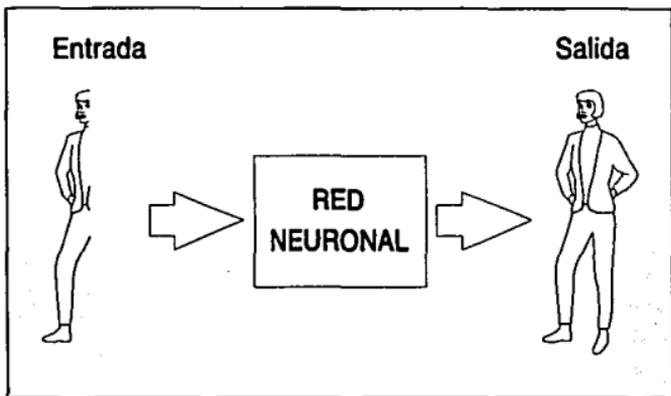


Fig.3.8.- Proceso de complementación de patrones.

Eliminación de ruido.

Este tipo de proceso es similar al anterior, sólo que aquí se desea una respuesta libre de todo ruido, de un patrón que ha sido afectado por tal; la diferencia entre estos dos tipos de procesos está básicamente en el modo de operación. Como en el ejemplo anterior, si la imagen que se le presenta a la red neuronal a la entrada, presenta en esta ocasión áreas borrosas o manchadas, la red se encarga de limpiar la imagen para quedar lo más nítida posible y enviarla a la salida, como se observa en la figura 3.9.

Es importante mencionar que las redes neuronales para la complementación de patrones y eliminación de ruido, se basan en el modelo de Red Discreta¹⁹ y el modelo El Cerebro como una Caja Negra²⁰, analizados en el capítulo 1.

¹⁹ Hopfield J. "Neural Networks and Physical Systems with Emergent Collective Computational Abilities". Proc. Nat. Acad. Sci. USA, vol.79, 1982.

²⁰ Anderson J., J. Silverstein, S. Ritz and R. Jones. "Distinctive Features, Categorical Perception, and Probability Learning: Some Applications of a Neural Model". Psych. Rev., vol. 84, 1977.

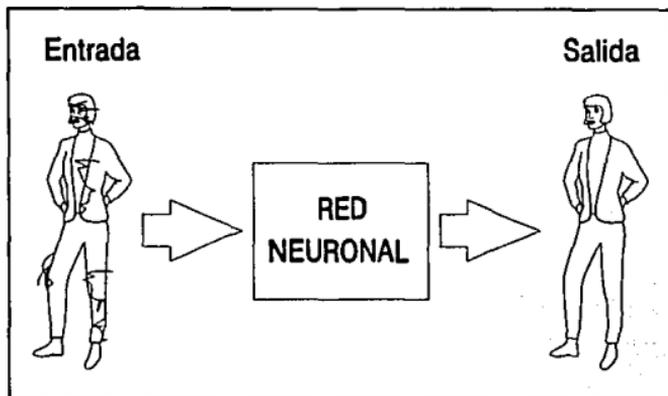


Fig.3.9.- Proceso de eliminación de ruido.

Optimización neuronal.

Este proceso consiste en una técnica para resolver problemas representados mediante una ecuación matemática, la cual debe ser maximizada o minimizada, según sea el caso. Algunos ejemplos representativos de la Técnica de Optimización²¹ son los que involucran inventarios, rutinas, distribución de recursos, etc.

En este tipo de comportamiento, al problema se le interpreta como una función de energía, la cual describe el comportamiento del sistema neuronal, por lo cual, si la función de energía desciende a un valor mínimo, la red buscará siempre mantenerla en un valor estable para poder encontrar la solución. Para esto, las entradas a la red neuronal representan el estado inicial del problema, y los valores finales que produzcan los elementos de procesamiento serán los parámetros de la solución. La figura 3.10 ilustra este proceso.

²¹Hopfield J. and Tank. "Neural" Computation of Decision in Optimization Problems". Biol. Cybernet., vol. 52, 1985.

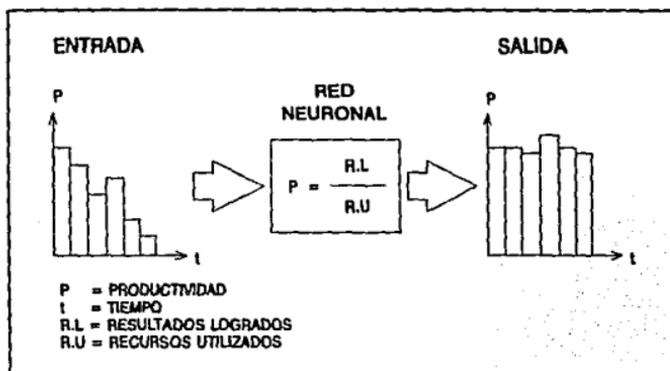


Fig.3.10.- Proceso de optimización neuronal.

Mejoramiento de contrastes.

Este tipo de procesamiento se caracteriza por utilizar conexiones denominadas "encendido-centro/apagado-alrededor". Las conexiones encendido-centro conocidas también como autoconexiones positivas representadas por $\{w_{ii} = \alpha (\alpha > 0)\}$ para $i = 1, 2, \dots, n$, son las que permiten que el valor de activación de un patrón vaya incrementándose por sí mismo. Por su parte, las conexiones apagado-alrededor llamadas también conexiones contiguas negativas expresadas por $\{w_{ij} = \beta (\beta > 0)\}$ para $i = j$, compiten con las conexiones encendido-centro.

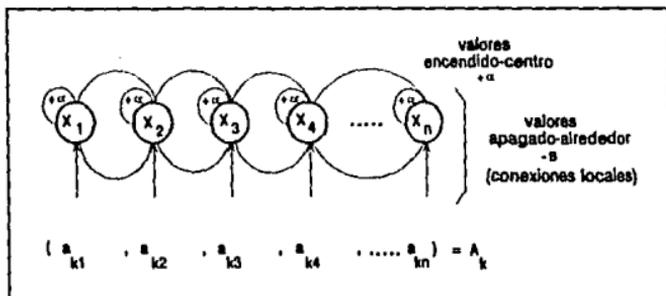


Fig.3.11.- Proceso de mejoramiento de contrastes con conexiones locales.

Las redes neuronales con este tipo de proceso pueden estar estructuradas de dos formas: con conexiones locales o conexiones globales, como se observa en la figura 3.11; esto es, si los elementos de procesamiento del nivel F_x son conectados con algunos de los elementos de procesamiento contiguos, se obtendrá una competencia local que dará como resultado valores de activación de diferentes tamaños. Ahora bien, si las conexiones apagado-alrededor se interconectan alrededor del nivel F_x , como se ilustra en la figura 3.12, habrá una competencia global y sólo habrá un ganador.

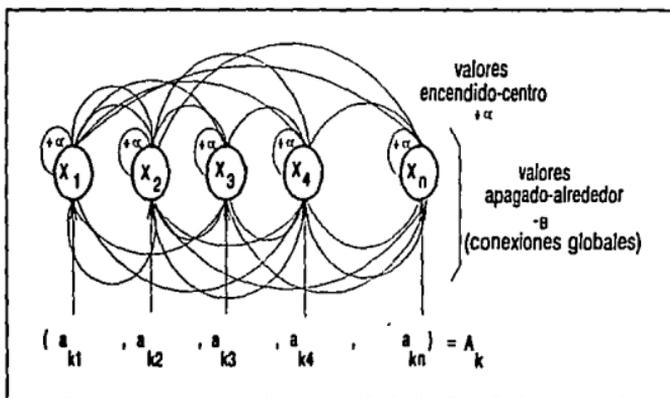


Fig.3.12.- Proceso de mejoramiento de contrastes con conexiones globales.

3.3. RED DE DOS NIVELES

Las Redes Neuronales Artificiales de dos niveles se conforman de la siguiente manera: un nivel F_x formado por n elementos de procesamiento y un nivel F_y de p elementos de procesamientos interconectados en su totalidad como se muestra en la figura 3.13, dando origen así a una matriz de peso W de dimensión $n \times p$, en la cual las entradas w_{ij} representan los pesos de las conexiones del i -ésimo elemento del nivel $F_x(x)$ hasta el j -ésimo elemento de procesamiento del nivel $F_y(y_j)$.

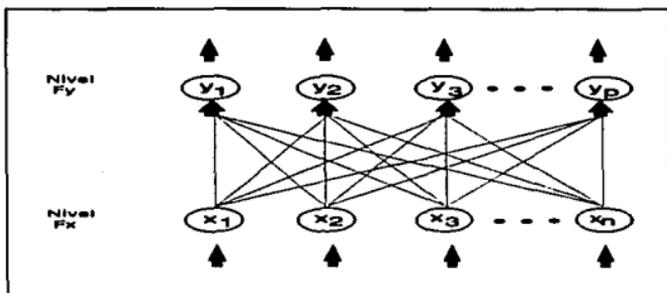


Fig.3.13.- Red de dos niveles.

A continuación daremos tres tipos de redes neuronales de dos niveles:

Igualación de patrones con alimentación hacia adelante .- Este tipo de red neuronal asigna el patrón de entrada A_k a su correspondiente patrón de salida B_k para $k = 1, 2, \dots, m$ como se muestra en la figura 3.14. Este tipo de red acepta el patrón de entrada A_k produciendo así un patrón de salida $Y = (y_1, y_2, \dots, y_p)$. Una asignación óptima entre las entradas y las salidas se obtiene cuando se ha alcanzado la mejor respuesta B_k .

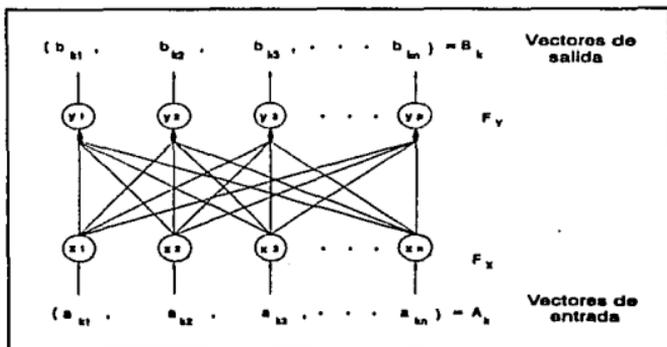


Fig.3.14.- Red para igualación de patrones con alimentación hacia adelante.

Aunque la mayoría de las redes de dos niveles son aplicadas para la búsqueda de asignaciones lineales óptimas entre patrones de entrada y salida, existen también otros tipos de redes de alimentación hacia delante que puede funcionar con asignaciones no lineales, esto lo logran ampliando el área que se le asigna a los patrones de entrada para incluir en ella una serie de multiplicaciones con las entradas originales.

Igualación de patrones con alimentación hacia atrás .- Una red de este tipo se muestra en la figura 3.15, en ella se observa que se aceptan entradas de cualquiera de los dos niveles de la red, es decir, del nivel F_x o del nivel F_y , para finalmente producir una salida que se enviará a alguno de los dos niveles, según sea el caso.

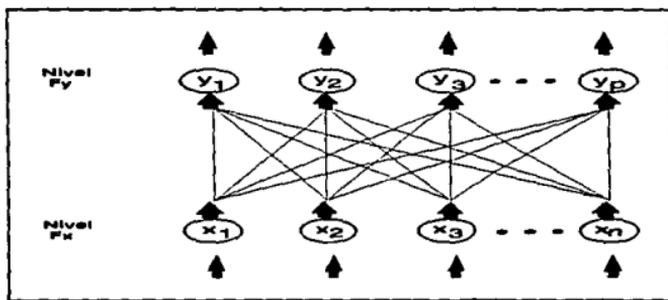


Fig.3.15.- Red para igualación de patrones con alimentación hacia atrás.

Clasificación de patrones con alimentación hacia adelante .- En este tipo de redes neuronales artificiales se crean una serie de clases para lo cual se asigna un patrón de entrada A_k a cualquiera de las p clases, como se observa en la figura 3.16. En este proceso se interpreta a cada clase como a un elemento de procesamiento del nivel F_y independiente a los demás, esto trae como consecuencia la reducción de la tarea para clasificar a los patrones y seleccionar el elemento de procesamiento del nivel F_y que mejor responda al patrón de entrada. Muchos de los sistemas cuya función es la de clasificar patrones utilizan la competencia entre las conexiones encendido-centro y apagado-alrededor para llevar a cabo esta tarea.

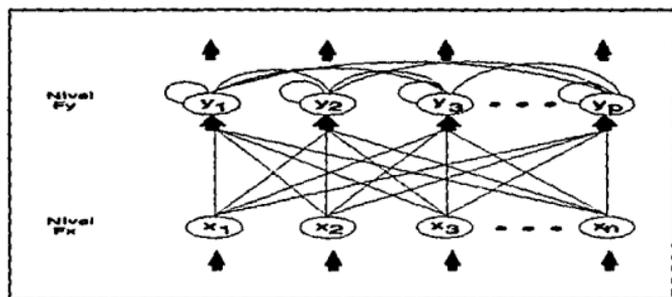


Fig.3.16.- Red para clasificación de patrones con alimentación hacia atrás.

3.4. RED MULTINIVEL

Las redes neuronales artificiales que en su estructura tienen más de dos niveles, son consideradas como redes multiniveles como se observa en la figura 3.17. Tenemos un modelo de red multinivel en la cual existe un nivel con elementos de procesamiento de entrada marcado como F_x , también tiene L niveles ocultos de elementos de procesamiento representados por F_y (y_1, y_2, \dots, y_L) y finalmente un nivel de salida F_z .

A los niveles F_y se les llama ocultos debido a que intervienen en las conexiones entre los niveles de entrada y salida, es decir, los elementos de procesamiento del nivel de entrada F_x son conectados a los elementos del nivel oculto F_{y1} , éste a su vez es conectado al nivel oculto F_{y2} y así sucesivamente, hasta que el último nivel oculto F_{yL} es conectado con el nivel de salida F_z .

Aunque las conexiones que se presentan en la figura 3.15 se realizan entre niveles contiguos, se puede dar el caso en el que las conexiones de un nivel pasen por encima de alguno o algunos niveles como puede ser cuando el nivel de entrada es conectada directamente con el nivel de salida; pero también puede suceder que algunas conexiones se realicen entre elementos de procesamientos de un mismo nivel.

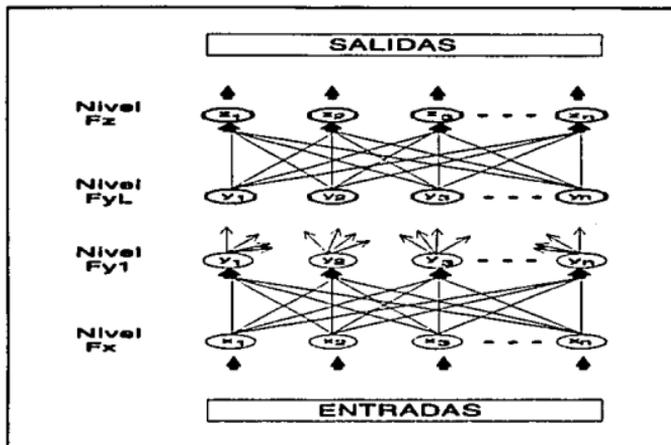


Fig.3.17.- Red de tipo multinivel.

Entre algunas de las aplicaciones de las redes neuronales artificiales en estudio se encuentran: la clasificación de patrones, igualación de patrones, las aproximaciones de funciones lineales; pero agregándole una función diferenciable continua a la red, tal como una función sigmoideal Gaussiana, es posible lograr que la red neuronal aprenda prácticamente cualquier tipo de rastreo de patrones y asignaciones no lineales.

El mecanismo que permite a las redes neuronales multiniveles realizar rastreos completos consiste en particionar en regiones el área que se le asigna a la entrada para posteriormente ir realizando rastreos de patrones en cada región particionada sucesivamente, para así producir eventualmente respuestas de salida. Debido a este mecanismo ahora las redes neuronales tienen la capacidad de resolver problemas en donde intervengan momentos de desición muy complejos.

Parece difícil determinar el número de niveles así como el número de elementos de procesamiento por cada uno, que se necesitan para la solución de un problema, pero de acuerdo a varias investigaciones se ha comprobado que el empleo de tan sólo tres niveles son lo suficiente para resolver cualquier problema donde se necesite de rastreos de patrones no lineales, es decir, para cualquier grado de dificultad que se

presente, tan sólo basta un nivel oculto.

Algunas de las técnicas más comunes utilizadas en las redes neuronales multinivel, donde intervienen los pesos de conexión modificables para realizar rastreo de patrones lineales o no lineales, son : el **Algoritmo de Retropropagación**²² (objeto de estudio en el capítulo 4) y el modelo **Neocognitron**²³ (analizado en el capítulo 1).

3.5. REDES ENLAZADAS ALEATORIAMENTE

En las redes neuronales artificiales conectadas aleatoriamente, la conexión de los pesos se realiza de forma aleatoria, dentro de un rango específico. Algunas de las redes de este tipo contienen conexiones de valor binario, es decir, la información que manejan es discreta. En este caso tenemos que cuando un peso de conexión tiene valor de cero equivale a decir que no hay presencia de conexión; debido a este tipo de conexiones aleatorias de valores binarios, las redes se simplifican en cuanto a número de conexiones. Este tipo de redes son frecuentemente utilizadas de las siguientes tres formas:

1. Pesos iniciales.- En este caso los valores iniciales de conexión que son presentados a la entrada de la red son previamente elegidos al azar siempre y cuando se encuentren dentro de un rango preestablecido. Esta técnica que se utiliza para seleccionar los valores iniciales que tendrán los pesos tienen su mayor aplicación en algunos sistemas de aprendizaje.

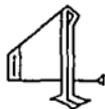
2. Procesamiento de patrones.- En este caso en los dos primeros niveles de una red neuronal multinivel se colocan un conjunto de conexiones de valor binario aleatorio para simular la operación de un patrón preprocesador; este patrón es utilizado para aumentar la dimensión del área que ha sido utilizada para rastrear patrones, este incremento se utiliza para mejorar dicho rastreo.

²² Rosenblatt P. "Principles of Neurodynamics". Washington, D.C.: Spartan Books, 1962.

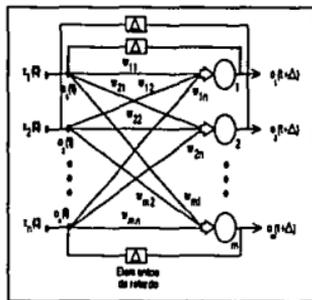
²³ Fukushima K. "Neocognitron: A hierarchical neural network capable of visual pattern recognition". Neural Networks, vol. 1, 1988.

3. "Conocimiento" como causa de la aleatoriedad.- A cerca de este caso, tenemos que las primeras investigaciones realizadas sobre las redes neuronales artificiales se dedicaron en gran parte al estudio de análisis de sistemas de valores binarios conectados aleatoriamente; como por ejemplo el modelo del Cerebro como **Una Red de Neuronas Conectadas Aleatoriamente**, que sirvió como base para el desarrollo de dichas investigaciones.

Capítulo



ALIMENTACION EN LAS REDES



INTRODUCCION

Una etapa importante en el diseño de las redes neuronales artificiales, es aquella en la cual se define el método de alimentación y manejo de la información para el aprendizaje de la red. Para esto, se requiere primeramente analizar el método de Retropropagación²⁴, que sirve para analizar el aprendizaje de las redes neuronales. Este método es uno de los más fáciles de actualización de información tienen como base: si la respuesta de salida no es satisfactoria, entonces los pesos sufren un proceso de corrección para que el error en la salida disminuya notablemente y por consecuencia las respuestas siguientes sean más acertadas.

El método de retropropagación fue propuesto primeramente por Paul Werbos en 1974, posteriormente fue revisado y modificado totalmente por David Parker en 1982, para que finalmente fuera publicado por Rumelhart y McClelland. Dicha publicación fue sin duda la que más tuvo influencia en el desarrollo de las redes

²⁴P. Rosenblatt, "Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanism". Washington, D.C: Sperry Books, 1962.

neuronales artificiales. Este método tomó muchas de sus características del modelo de red neuronal **Perceptron**²⁵, aunque existen dos diferencias importantes entre ambos:

- 1.- El modelo perceptron únicamente se estructura de dos niveles de elementos de procesamiento, entrada y salida, y un nivel oculto. Que por el contrario, el método de retropropagación puede ser aplicado a cualquier tipo de red neuronal artificial.
- 2.- El modelo perceptron sólo puede aprender de patrones que son linealmente separables, no siendo así para el modelo de retropropagación, ya que es capaz de resolver problemas en donde se requiera del mapeo de patrones lineales o no lineales.

Es muy común que en la estructura de un modelo de retropropagación este constituido de redes neuronales de tres o más niveles, ya que en la arquitectura de estas redes es frecuente que el nivel de entrada reciba los patrones externos, y que el número de niveles ocultos que pueda haber en la red sea ilimitado, para finalmente pasar hacia el nivel de salida.

Todos los niveles ocultos de la red se encuentran conectados mediante todos los elementos de procesamiento de los niveles tanto inferiores como superiores, no siendo posible que se conecten elementos de procesamiento de un mismo nivel, pero sí puede darse el caso que los elementos de procesamiento de un nivel se conecten con los elementos de procesamiento de niveles superiores o inferiores pudiendo ser brincados algunos de ellos. En la figura 4.1 se observa una red neuronal artificial con la arquitectura de retropropagación.

²⁵ D.E. Rumelhart and J.L. McClelland, Eds., "Parallel Distributed Processing", Cambridge, M.A: M.I.T. Press, 1986.

neuronales artificiales. Este método tomó muchas de sus características del modelo de red neuronal **Perceptron**²⁵, aunque existen dos diferencias importantes entre ambos:

- 1.- El modelo perceptron únicamente se estructura de dos niveles de elementos de procesamiento, entrada y salida, y un nivel oculto. Que por el contrario, el método de retropropagación puede ser aplicado a cualquier tipo de red neuronal artificial.
- 2.- El modelo perceptron sólo puede aprender de patrones que son linealmente separables, no siendo así para el modelo de retropropagación, ya que es capaz de resolver problemas en donde se requiera del mapeo de patrones lineales o no lineales.

Es muy común que en la estructura de un modelo de retropropagación este constituido de redes neuronales de tres o más niveles, ya que en la arquitectura de estas redes es frecuente que el nivel de entrada reciba los patrones externos, y que el número de niveles ocultos que pueda haber en la red sea ilimitado, para finalmente pasar hacia el nivel de salida.

Todos los niveles ocultos de la red se encuentran conectados mediante todos los elementos de procesamiento de los niveles tanto inferiores como superiores, no siendo posible que se conecten elementos de procesamiento de un mismo nivel, pero si puede darse el caso que los elementos de procesamiento de un nivel se conecten con los elementos de procesamiento de niveles superiores o inferiores pudiendo ser brincados algunos de ellos. En la figura 4.1 se observa una red neuronal artificial con la arquitectura de retropropagación.

²⁵D.E. Rumelhart and J.L. McClelland, Eds., "Parallel Distributed Processing". Cambridge, MA: M.I.T. Press, 1986.

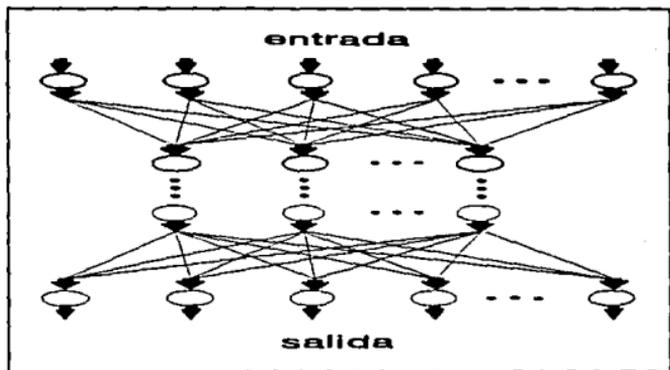


Fig.4.1.- Red Neuronal con método de retropropagación.

Entre las aplicaciones que se les ha dado a las redes con este tipo de proceso, se encuentran los diagnósticos médicos que asocian un cuadro clínico o una serie de síntomas a enfermedades particulares, además para la detección de arritmias, taquicardias y otras enfermedades cardiovasculares, es decir, en donde se requiera la lectura de electrocardiogramas; en el área de la robótica se ha empleado principalmente en los sistemas de control, como es el caso del manejo de movimiento de brazos mecánicos y/o dar seguimiento a trayectorias temporales. Otras aplicaciones que se le ha dado, es en el ámbito económico, como lo son las predicciones en los mercados bursátiles, o también en el desarrollo de apuestas en algunos deportes, en el reconocimiento de códigos postales, etc.

El algoritmo de entrenamiento por retropropagación puede aplicarse en dos tipos de redes neuronales artificiales: con **Alimentación Hacia Adelante** y con **Alimentación Hacia Atrás**²⁰. A continuación analizaremos la forma en que es aplicado este algoritmo en cada una de estas redes.

4.1. ALIMENTACION HACIA ADELANTE.

Para las redes neuronales artificiales que son entrenadas mediante el algoritmo

²⁰Jacek M. Zurada. "Introduction to Artificial Neural Systems". West Publishing Company. USA, 1992.

de retropropagación con alimentación hacia adelante, el proceso da comienzo con la presencia de un patrón de entrada a la red, el cual es detectado y recibido por el o los elementos del nivel de entrada, que únicamente adquieren el valor del vector de entrada y lo dirigen al siguiente nivel, en donde comienza a ser procesado por los elementos de dicho nivel, para ser enviado a su vez a otros niveles de procesamiento o activación, hasta llegar finalmente al nivel de salida.

Un punto muy importante durante este proceso es el que realizan los elementos de procesamiento, ya que se encargan de sumar todas sus entradas para unificar estos valores en uno sólo y pasarlo por una función no lineal y ser transmitido a otros niveles.

Para comprender mejor este proceso, tomaremos como referencia la arquitectura de una red neuronal con alimentación hacia adelante, que consta de m elementos de procesamiento del nivel de entrada, aceptando n patrones de entrada como se observa en la figura 4.2.

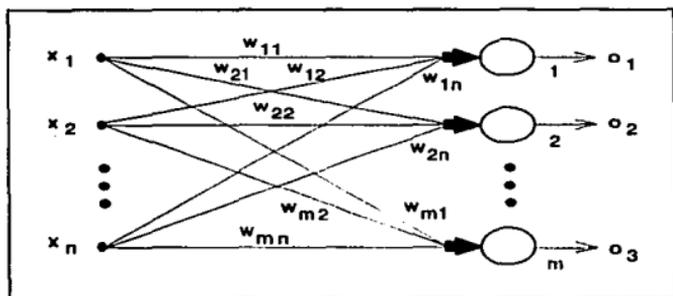


Fig.4.2.- Red Neuronal con alimentación hacia adelante.

De acuerdo a la figura obtenemos que los vectores respectivos al nivel de entrada y de salida son:

$$O = [o_1 , o_2 , \dots , o_m] \quad \text{entrada} \quad (1)$$

$$X = [x_1 , x_2 , \dots , x_n] \quad \text{salida}$$

Los pesos de conexión representados por w_{ij} conectan al i -ésimo nivel de procesamiento del nivel de entrada con el j -ésimo patrón de entrada. Cabe mencionar que la notación que utilizaremos para los subíndices es la siguiente: el primer valor corresponde al nodo destino y el segundo valor al nodo fuente. De esta manera denotaremos el valor de activación para el i -ésimo elemento de procesamiento (PE_{*i*}) mediante la expresión:

$$PE_i = \sum_{j=1}^n w_{ij} x_j \quad \text{para } i = 1, 2, \dots, m \quad (2)$$

El siguiente paso consiste en utilizar un proceso de transformación no lineal, la cual involucra a la función de activación $f(PE_i)$ para $i = 1, 2, \dots, n$, completando de esta forma el procesamiento para el vector de entrada X . Este proceso de transformación se activa de forma independiente para cada uno de los m elementos de procesamiento del nivel de entrada en la red, además, representa un mapeo no lineal expresado de la siguiente forma:

$$o_i = f(W_i X) \quad \text{para } i = 1, 2, \dots, n \quad (3)$$

De donde W representa el vector de peso, que esta formado por los principales pesos del i -ésimo nodo de salida que definiremos en la siguiente ecuación:

$$W_i = (w_{i1}, w_{i2}, \dots, w_{in}) \quad (4)$$

El mapeo que se lleva a cabo entre el vector X y el vector de salida O dentro de la red, lo expresaremos utilizando el operador matricial no lineal Γ , quedando de la siguiente forma:

$$O = \Gamma [W X] \quad (5)$$

De donde W representa la matriz de peso, también conocida como matriz de conexiones cuya nomenclatura es:

$$W = \begin{bmatrix} W_{11} & W_{12} & \dots & W_{1n} \\ W_{21} & W_{22} & \dots & W_{2n} \\ \cdot & \cdot & \dots & \cdot \\ W_{m1} & W_{m2} & \dots & W_{mn} \end{bmatrix} \quad (6)$$

que utilizando el operador Γ queda como:

$$G [\cdot] = \begin{bmatrix} f(\cdot) & 0 & \dots & 0 \\ 0 & f(\cdot) & \dots & 0 \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ 0 & 0 & \dots & f(\cdot) \end{bmatrix} \quad (7)$$

En la ecuación (7) es importante mencionar que las funciones de activación no lineales f que se encuentran sobre la diagonal principal de dicha matriz, operan básicamente sobre los valores de activación PE de cada elemento de procesamiento.

Por lo tanto, tenemos que cada valor de activación representa a un producto escalar entre un patrón de entrada y su respectivo vector de peso.

Para las redes que utilizan este tipo de alimentación, el mapeo que se realiza de un patrón de entrada hacia un patrón de salida, como se muestra en la figura 4.3, es de tipo instantáneo y con alimentación hacia adelante, debido a que no utiliza un tiempo de retardo entre el patrón de entrada X y el patrón de salida O .

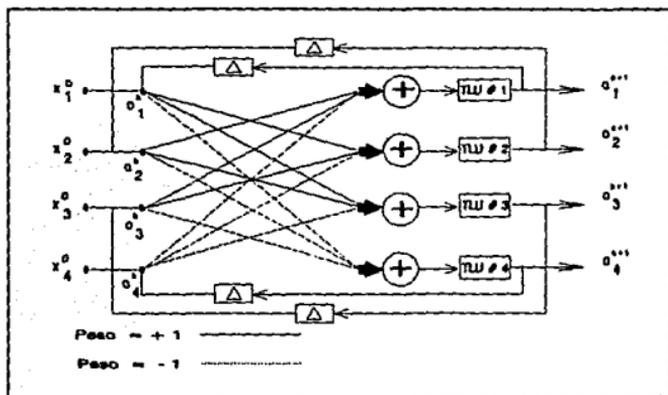


Fig.4.3.- Red neuronal con alimentación hacia adelante y mapeo de tipo instantáneo.

Por lo tanto tenemos que la ecuación (5) puede tomar la siguiente forma explícita en donde involucramos el tiempo t:

$$O(t) = \Gamma \{ W X(t) \} \quad (8)$$

De esta manera tenemos que las redes neuronales artificiales con alimentación hacia adelante se caracterizan principalmente por la ausencia de lazos de retroalimentación, como se aprecia en la figura 4.4, en donde tenemos un diagrama a bloques de dichas redes.

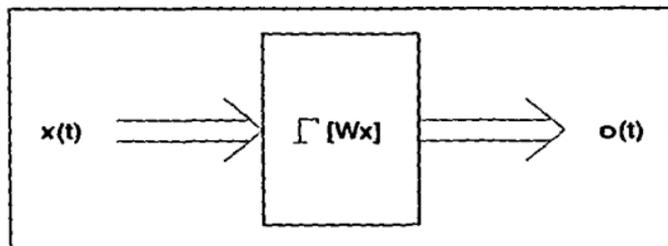


Fig.4.4.- Diagrama a bloques de una red neuronal de alimentación hacia adelante.

Finalmente, este tipo de redes pueden ser enlazadas en forma de cascada dando origen así a una red de tipo multinivel como se muestra en la figura 4.5, en donde la salida de un nivel se convierte en la entrada del nivel siguiente; aunque las redes neuronales artificiales de este tipo no poseen conexiones explícitas de alimentación hacia atrás, cuando el vector de entrada $x(t)$ es mapeado hacia el vector de salida $o(t)$, los valores de salida son constantemente comparados con la información circundante, proporcionándose así el valor de salida como también una señal de error que puede ser utilizada para la adaptación de los pesos de la red.

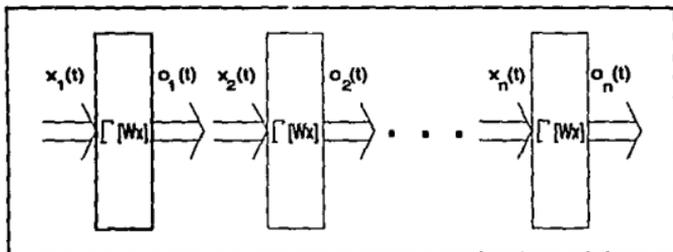


Fig.4.5.- Red Neuronal con alimentación hacia adelante de tipo multinivel.

4.2. ALIMENTACION HACIA ATRAS

Para el diseño de una red neuronal artificial con alimentación hacia atrás podemos tomar como base la estructura de una red neuronal con alimentación hacia adelante, agregándole conexiones entre las salidas y las entradas mediante lazos de retroalimentación, como se muestra en la figura 4.6.

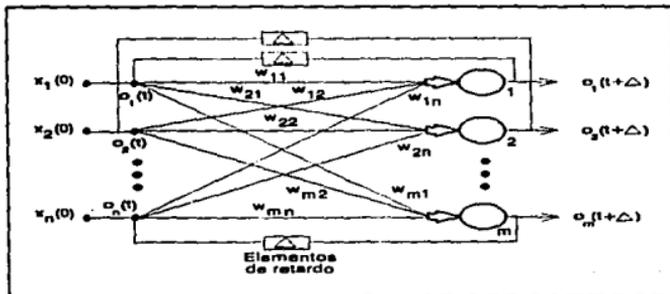


Fig.4.6.- Red Neuronal con alimentación hacia atrás.

La función principal de esta retroalimentación es la de proporcionar un control de la salida o_j , mediante las salidas representadas por la variable o_j , para $j = 1, 2, \dots, m$. Dicho proceso de control se lleva a cabo cuando la salida presente $o(t)$ controla a la salida que se presenta en el instante siguiente marcado como $o(t + \Delta)$. Este lapso de tiempo (Δ) entre t y el instante $(t + \Delta)$ es producido y proporcionado por los elementos de retardo, los cuales se encuentran ubicados en el lazo de retroalimentación, como se muestra en la figura 4.6.

Si utilizamos la notación empleada en las redes neuronales artificiales con alimentación hacia adelante, el mapeo que se realiza entre las salidas $o(t)$ y $o(t + \Delta)$ se expresa como:

$$o(t + \Delta) = \Gamma [W o(t)] \quad (9)$$

Dicha fórmula se observa en el diagrama a bloques de la figura 4.7.

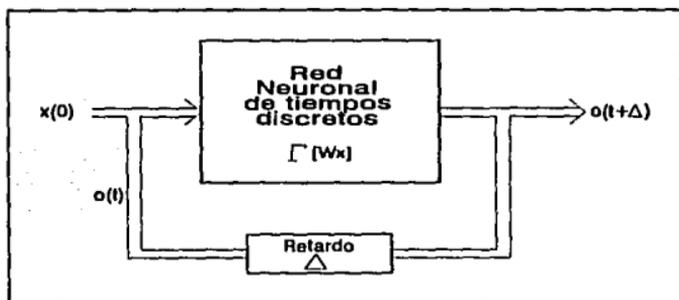


Fig.4.7.- Red Neuronal con alimentación hacia atrás con tiempos de retardo.

De la figura tenemos que el patrón de entrada $x(t)$ se utiliza únicamente para inicializar a la red, esto es, que el valor $o(0) = x(0)$. Por lo tanto tenemos que la entrada es eliminada y el sistema se vuelve autónomo para los tiempos $t > 0$, por lo que consideramos a este un caso especial de entre las configuraciones de redes neuronales artificiales con alimentación hacia atrás que corresponde a las de un sólo nivel.

Dentro de este tipo de redes existe uno especial denominado **Sistema de Tiempo Discreto**²⁷, el cual consiste en considerar al tiempo como una variable discreta para observar el funcionamiento de las redes neuronales mediante instantes de tiempos discretos $\Delta_1, \Delta_2, \Delta_3, \dots$. Para las redes de tiempo discreto es conveniente utilizar enteros positivos de manera indexada para los instantes de tiempo, para lo cual el símbolo Δ significa únicamente la unidad de retardo. Para estos sistemas neuronales artificiales, tenemos que la ecuación (9) para tiempos discretos queda de la siguiente forma:

$$o^{k+1} = \Gamma [W O^k] \quad \text{para } k = 1, 2, \dots \quad (10)$$

de donde k representa cada instante. A esta red se le conoce como **recurrente**, esto debido a que la respuesta en el instante $k + 1$ depende directamente de los instantes anteriores, es decir, desde que la red inicia su funcionamiento en el instante $k + 1$. A partir de lo anterior logramos obtener una serie de soluciones derivadas de la ecuación (10), como son:

$$\begin{aligned} o^1 &= \Gamma [W x^0] \\ o^2 &= \Gamma [W \Gamma [W x^0]] \\ &\vdots \\ &\vdots \\ o^{k+1} &= \Gamma [W \Gamma [\dots \Gamma [W x^0] \dots]] \end{aligned} \quad (11)$$

Las redes de proceso recurrentes como la anterior operan con una representación escrita de datos y con elementos de procesamiento con funciones de activación ilimitada. A este tipo de redes neuronales artificiales se les considera como autómatas debido a que utilizan para su funcionamiento entradas de tiempo discreto y una representación de datos discretos como ya se había mencionado.

Cuando estas redes neuronales se encuentran en los instantes o^k para $k = 1, 2, 3, \dots$ comienzan a producir una secuencia de estados de transición, es decir, la red comienza los estados de transición una vez que se ha inicializado en el instante 0 con x^0 y continúan sus estados de transición o^k para $k = 1, 2, 3, \dots$, hasta que sea posible encontrar un estado de equilibrio, este estado comúnmente es llamado

²⁷ Jacob M. Zurada. "Introduction to Artificial Neural Systems". West Publishing Company. USA 1992.

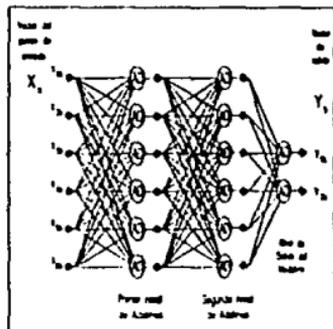
atractor, el cual consiste únicamente de un sólo estado o de un número limitado de estados.

Finalmente tenemos que la secuencia de un estado recurrente es generalmente no determinístico. Además, estos estados pueden ser potencialmente extendidos por la red después de un número de transiciones no determinísticas también.

Capítulo



PRINCIPIOS DE LOS MODELOS ADALINE Y MADALINE



INTRODUCCION

Los métodos de procesamientos de señales tuvieron su desarrollo dentro de la rama de la ingeniería con la llegada de una nueva rama en la electrónica, la comunicación. En sus inicios, para llevar a cabo este tipo de comunicación se requería de filtros analógicos, los cuales consistían de circuitos RLC (resistor-inductor-capacitor) que fueron diseñados con el propósito de eliminar el ruido que existía en las señales de comunicación. Hoy en día, los sistemas de procesamientos de señales junto con sus avances, se han involucrado dentro de una tecnología multifacética, haciendo énfasis principalmente en la evolución que hubo desde la implementación de los primeros circuitos sintonizadores (RLC) hasta la llegada de los procesadores de señales digitales (DSP's).

Las bases para las industrias de la comunicación residen en el diseño e implementación de filtros, cuya función es la de eliminar el ruido existente en las señales de información generadas y/o recibidas por ellas.

Para el desarrollo de los procesadores de señales digitales se pensó en varios modelos de redes neuronales artificiales, de los cuales se tomó de acuerdo a sus características el primer perceptrón, que consistía de dos niveles, por lo que las

investigaciones no se detuvieron con el desarrollo de este perceptron y optaron por el diseño de un perceptron más preciso, el cual consistió de múltiples niveles. Dentro de este conjunto de perceptrones, Bernard Widrow y un grupo de investigadores tomaron principalmente dos modelos para su análisis y desarrollo, a los cuales denominaron perceptrones **Adaline** y **Madaline**. El Adaline (ADaptive LINear Element "Elemento Lineal ADaptivo") también es conocido como **Unidad Umbral Lineal** y constituye el bloque básico de la estructura de una red Madaline (Many Adaline "Muchas Adaline"). De acuerdo al modelo Adaline, su estructura se constituye de una sola neurona localizada en el nivel F2, en tanto el modelo Madaline posee cualquier número mayor de una neurona en el mismo nivel, como se aprecia en la figura 5.1.

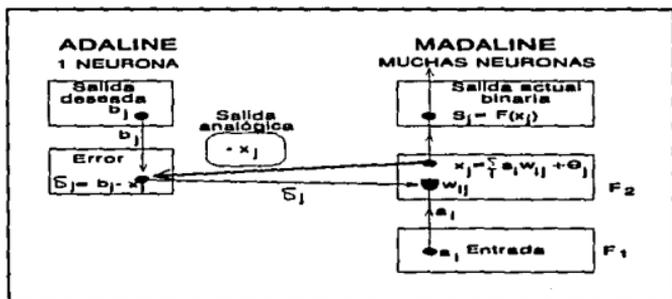


Fig.5.1.- Comparación entre el modelo ADALINE/MADALINE y el Perceptron.

En la figura 5.1, se puede observar que las áreas resaltadas muestran la diferencia principal entre los modelos Adaline/Madaline y el Perceptron de alimentación hacia adelante de Rosemblatt, que consiste en que: el modelo Adaline / Madaline compara la salida analógica x_j que genera la red con la salida b_j que se pretende obtener. Esta comparación proporciona una indexación de errores derivada de la relación entre una salida binaria y una salida deseada. De acuerdo a la figura 5.1 el error $b_j - x_j = \delta_j$ representa la retroalimentación que sirve para ajustar los pesos, que utilizando la **Regla de Corrección de Error de Retroacoplamiento de Rosemblatt**²⁸ queda de la siguiente forma:

$$\frac{dw_{ij}}{dt} = \alpha \delta_j \frac{a_i}{|a_i|^2}$$

²⁸ P. Rosemblatt. "Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanism". Washington, D.C.: Spartan Books, 1962.

esta regla minimiza el error cuadrático medio representado por:

$$\sum_j \delta_j^2$$

con la regla anterior obtenemos el promedio de todas las entradas.

Los modelos Adaline y Madaline nos proporcionan muchos ejemplos sobre los avances tecnológicos generados por las investigaciones de las redes neuronales artificiales.

A continuación daremos una descripción más amplia acerca de los modelos básicos del Adaline y Madaline.

5.1. ADALINE

COMBINADOR LINEAL ADAPTIVO

La red neuronal artificial denominada Adaline, es un dispositivo conformado únicamente por un elemento de procesamiento, por lo tanto, en términos técnicos, el Adaline no es considerado como una red neuronal artificial, sin embargo, aunque pequeña, su estructura es muy importante, debido a que se forma de un **Combinador Lineal Adaptivo**, por lo que el estudio de este tipo de redes, requiere previamente de un análisis de estos combinadores.

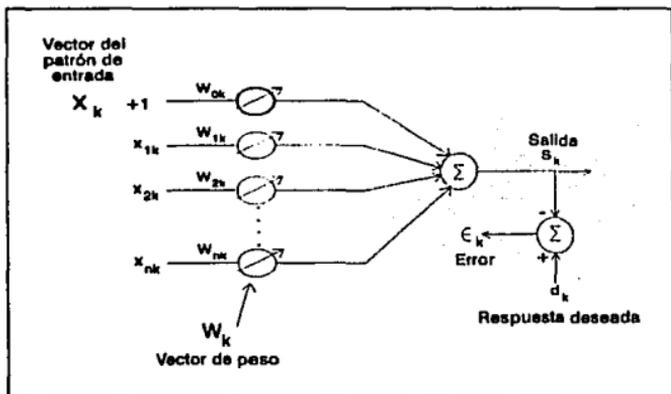


Fig.5.2.- Esquema de un Combinador Lineal Adaptivo.

En la figura 5.2 se observa un Combinador lineal adaptivo, en el que la salida representa la combinación lineal de las entradas. Ahora bien, si implementamos este combinador a un sistema de tiempos discretos, recibirá en el tiempo marcado como k un vector con la señal de entrada, representado por la expresión:

$$X_k = [x_{0k}, x_{1k}, x_{2k}, \dots, x_{rk}]$$

y una respuesta deseada d_k ; en tanto, una entrada especial se utiliza para efectos de entrenamiento y aprendizaje. El vector de entrada está compuesto principalmente por un conjunto de pesos marcados con su respectivo coeficiente, que se expresa por:

$$W_k = [w_{0k}, w_{1k}, w_{2k}, \dots, w_{rk}]$$

entonces la suma de las entradas y los pesos es procesada, produciendo así una salida lineal representada por el producto:

$$S_k = X_k W_k$$

de donde los componentes de X_k pueden tomar cualquier valor analógico continuo o en su defecto valores binarios. Por su parte, los pesos son variables, esencialmente, son consideradas como variables continuas y pueden tomar valores ya sean positivos o negativos.

Durante el proceso de entrenamiento de la red, tanto los patrones de entradas como las correspondientes respuestas deseadas, se presentan ante el combinador lineal. Después, un algoritmo de entrenamiento ajusta de manera automática los pesos, de manera que las respuestas de salida puedan ser consideradas como las respuestas deseadas.

Para esto, tenemos que en las aplicaciones en donde intervengan procesamientos de señales, el método más común para la adaptación de pesos es el **Algoritmo LMS (Least Mean Square)**²⁹, que frecuentemente es llamada **Regla Delta Widrow-Hoff**³⁰. Dicho algoritmo tiene como tarea minimizar el total de la suma de los cuadrados de los errores lineales, para esto, el error lineal ϵ_k se encuentra definido

²⁹B. Widrow and M.E. Hoff, Jr. "Adaptive Switching Circuits". IRE Western Electric Show and Convention Record, Part IV, 1960.

³⁰D.H. Rumelhart and J.L. McClelland. "Learning Internal Representation by Error Propagation". In Parallel Distributed Processing, Vol. 1, ch. VIII, Eds. Cambridge, Ma: M.I.T. Press, 1986.

por la diferencia entre la respuesta deseada d_k y la salida lineal S_k durante el tiempo k , siendo esto necesario para la adaptación de los pesos.

Cuando el combinador lineal adaptivo esta implantado en una red neuronal artificial de arquitectura multinivel, por lo menos una señal de error está disponible, aunque no de manera directa, para cada combinador lineal individual, y muchos de los procedimientos deberán ser proporcionados para adaptar sus vectores de pesos.

ADALINE

El bloque básico de construcción que frecuentemente es usado para el diseño de las redes neuronales artificiales, es el Elemento Lineal Adaptivo mejor conocido como **ADALINE**³¹, que en sus principios fue llamado *Neurona Lineal Adaptiva*, como se observa en la figura 5.3.

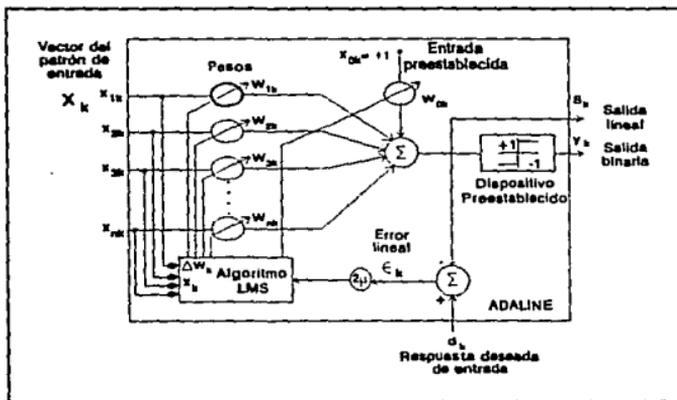


Fig. 5.3.- Red neuronal artificial con arquitectura ADALINE.

Muchos de los investigadores coinciden en que la estructura del ADALINE es semejante a la descripción general de un elemento de procesamiento, para esto, existen dos modificaciones que se requieren para hacer que la estructura de dicho elemento de procesamiento cumpla con las características de una red ADALINE, las cuales son:

³¹ B. Widrow and M.E. Hoff, Jr. "Adaptive Switching Circuits". IRE Western Electric Show and Convention Record, Part IV, 1960.

- 1.- Adicionar al conjunto de conexiones que llegan al combinador lineal adaptivo, una conexión más, con un peso marcado como w_0 , el cual siempre tendrá un valor de entrada igual a uno.
- 2.- Esta consiste, en la colocación de una condición bipolar sobre la salida; dicha condición, funciona de la siguiente forma: si la salida del combinador lineal adaptivo es positivo, la salida de la red tendrá un valor igual a +1, si por el contrario, la salida del combinador lineal adaptivo es negativo, la salida al final de la red será -1.

Estas características se aprecian en la figura 5.4.

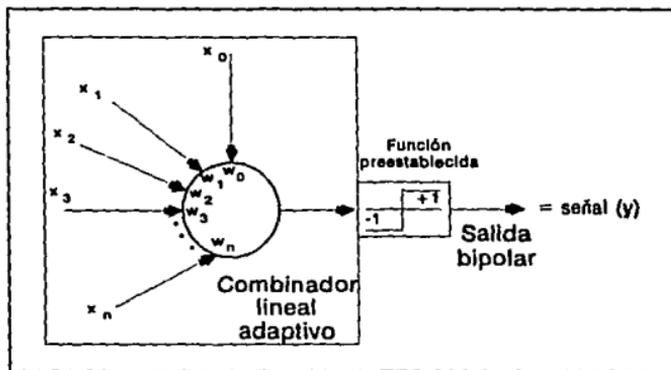


Fig.5.4.- Elemento de procesamiento modificado para funcionar como una estructura ADALINE.

Para comprender mejor el funcionamiento de una red neuronal de tipo ADALINE, tomaremos como referencia la red neuronal artificial que se observa en la figura 5.3, la cual es capaz de implementar ciertas funciones lógicas.

En ella tenemos 2^n posibles patrones de entrada; para esto, se pretende que una implementación lógica general sea capaz de clasificar cada patrón, cualquiera que sea su valor, +1 o -1, de acuerdo a la respuesta deseada. De esta manera tenemos 2^{2^n} posibles funciones lógicas que conectan las n entradas a una sola salida binaria. Debido a lo anterior, una red neuronal de tipo ADALINE tiene la capacidad de procesar únicamente un pequeño subconjunto de estas funciones, las cuales son conocidas

como **Funciones Lógicas Separables Linealmente** o **Funciones Lógicas Preestablecidas**³², que pueden ser obtenidas mediante todas las variaciones posibles sobre los pesos.

Por otro lado, tenemos un caso particular, en el cual el elemento llamado **ADALINE** posee únicamente dos entradas, como se observa en la figura 5.5. Así también, en la figura 5.6 podemos apreciar todas las posibles entradas binarias a dicho elemento, representadas por cuatro puntos extensos, localizados dentro del espacio correspondiente al vector del patrón de entrada. En este espacio, los componentes de dicho vector se encuentran situados a lo largo de los ejes de coordenadas.

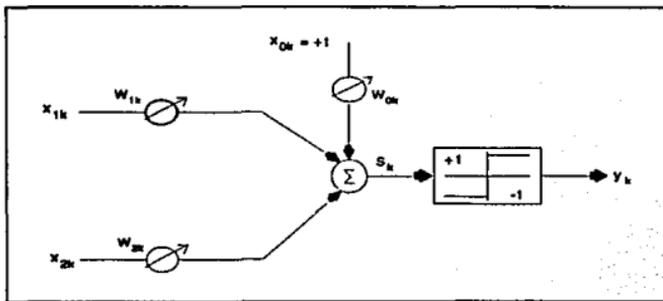


Fig.5.5.- Elemento ADALINE con dos entradas.

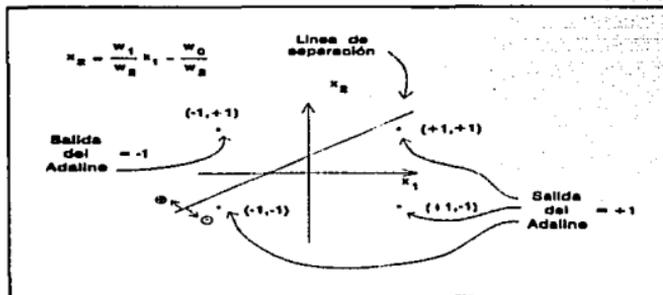


Fig.5.6. Número posible de entradas binarias para el elemento ADALINE de 2 entradas.

³² P.M. Lewis II and C. Conca, "Threshold Logic". New York: Wiley, 1967.

De acuerdo a las figuras anteriores, durante su funcionamiento, el elemento ADALINE lleva a cabo una tarea muy importante, que consiste en separar a los patrones de entrada en dos categorías que depende de los valores que tengan los pesos. Para esto, tenemos que una condición preestablecida en estado crítico se presenta cuando el valor de la salida lineal es igual a cero, es decir:

$$s = x_1 w_1 + x_2 w_2 + w_0 = 0$$

de la ecuación anterior tenemos que:

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

siendo esta relación lineal, apreciada en la figura 5.6, la cual contempla tanto una línea inclinada de separación como una intersección, representadas ambas por:

$$\text{pendiente} = - \frac{w_1}{w_2}$$

$$\text{intersección} = - \frac{w_0}{w_2}$$

Los pesos marcados como w_0 , w_1 y w_2 son los que determinan la pendiente, la intersección y el lado de la línea de separación que corresponde a la salida positiva, en tanto en el lado opuesto, la línea de separación corresponde a la salida negativa.

De acuerdo a las expresiones anteriores, decimos que en las redes ADALINE que contemplan en su estructura cuatro pesos, su límite de separación será siempre un plano, no siendo así para las ADALINE que contemplan más de cuatro pesos, ya que en este caso su límite de separación será un hiperplano. Como lo señalamos anteriormente, cuando el peso adicional marcado como w_0 toma el valor de cero, tenemos que la separación hiperplana será homogénea, es decir, pasará dentro del espacio del vector del patrón y por el origen.

De la figura 5.6 obtenemos que los patrones binarios de entrada están clasificados de la siguiente manera:

$(+1, +1)$	—————▶	$+1$
$(+1, -1)$	—————▶	$+1$
$(-1, -1)$	—————▶	$+1$
$(-1, +1)$	—————▶	$+1$

De acuerdo a los resultados anteriores, observamos que la red ADALINE cumple con la función de separabilidad lineal. Por el contrario, si no cumpliera con esta función, estaríamos hablando de que lleva a cabo una función NOR exclusiva con dos entradas, y en la cual los patrones binarios de entrada estarían clasificados de la siguiente forma:

$(+1, +1)$	—————▶	$+1$
$(+1, -1)$	—————▶	-1
$(-1, -1)$	—————▶	$+1$
$(-1, +1)$	—————▶	-1

De esta manera, observamos que una red ADALINE de estructura simple sin una fase de preprocesamiento, no tiene la capacidad de implementar una función del tipo NOR exclusiva.

En tanto una red ADALINE sencilla pero con dos entradas, es capaz de realizar 14 de las 16 posibles funciones lógicas; que si por el contrario, la red posee un número mayor de entradas, únicamente una pequeña fracción de las funciones lógicas se podrán realizar, pero siempre siendo éstas linealmente separables.

Finalmente tenemos que las combinaciones de los elementos o también llamados redes de elementos, tienen su mayor aplicación en las redes cuyas funciones no son linealmente separables.

5.2. MADALINE

Como lo analizamos en el tema anterior, uno de los principales problemas en el desarrollo de las redes, se presenta en las redes ADALINE de dos niveles, debido a que estas no pueden procesar funciones como es el caso de la OR exclusiva.

Para solucionar este problema surgieron los primeros modelos de redes neuronales artificiales cuyos niveles se encontraban compuestos por elementos adaptivos múltiples, es decir, una combinación de redes ADALINE, como se observa

Algunos análisis matemáticos enfocados al estudio de la estructura del MADALINE fueron desarrollados en las tesis de Ridway³⁵, Hoff y Glanz³⁶, en las que coincidieron en que las arquitecturas multinivel MADALINE eran las más aproximadas a lo que es la estructura real de una red neuronal artificial, como la que se aprecia en la figura 5.8.

De acuerdo a la figura anterior, tenemos que la red MADALINE podría ser presentada con un vector de entrada de una dimensión extensa, como por ejemplo, los valores de los pixeles de una imagen rastreada pueden ser tomados como los valores de los patrones de entrada. En este caso, con un entrenamiento adecuado, la red podría enviar como respuesta un 1 binario a cualquiera de los diferentes nodos de salida, en donde cada respuesta corresponde a una de las porciones de la imagen de entrada. Para un patrón de entrada, un nodo deberá tener una salida con valor +1, si el patrón de entrada corresponde al de la clase representada por dicho nodo, de manera que los nodos restantes que forman a la red deberán tener un valor de -1.

Cuando una red neuronal de este tipo ha sido entrenada con una identificación previa de los patrones de entrada, el vector de salida deseado es ya conocido; lo que realmente es desconocido, es el valor de la salida deseada para un nodo conocido el cual se encuentra localizado en uno de los niveles ocultos.

Durante el desarrollo de estas redes, tenemos que al principio de la década de los 60's se construyó una red con arquitectura MADALINE formada por 1000 conexiones de peso; la cual fue utilizada en el área de reconocimiento de patrones; en ella, los pesos que se utilizaron para su elaboración fueron fabricados con memistores, que son resistencias eléctricamente variables; este tipo de redes fueron desarrolladas por Widrow y Hoff, y cuya configuración fue la siguiente:

Las entradas que son introducidas por la retina se conectan a un nivel formado por elementos Adaline adaptivos, en tanto, las salidas eran conectadas a un

³⁵W.C. Ridway III. "An Adaptive Logic System with Generalizing Properties". Ph.D. Thesis, Tech. Rep., Stanford Electron Labs., Stanford, CA. April 1962.

³⁶F.H. Glanz. "Statistical Extrapolation in Certain Adaptive Pattern-Recognition Systems". Ph.D. Thesis, Tech. Rep., Stanford Electron Labs., Stanford, CA., May., 1965.

dispositivo lógico estable, el cual finalmente generaba la salida del sistema. Un ejemplo de este tipo de redes se muestra en la figura 5.9 en donde se conectan dos redes Adaline a un dispositivo lógico del tipo AND para así proporcionar una salida.

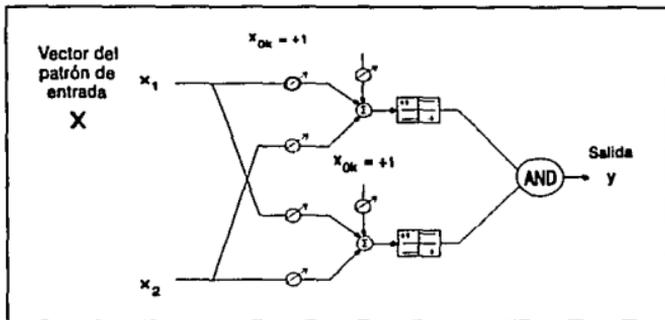


Fig.5.9.- Red neuronal MADALINE formada con 2 elementos ADALINE conectadas a un dispositivo lógico de tipo AND.

Con una adecuada selección de los pesos, obtenemos que el límite de separación dentro del espacio del patrón de entrada para la red neuronal ilustrada en la figura 5.9, deberá ser similar a la que se observa en la figura 5.10.

Es importante mencionar que este límite de separación es capaz de implementar la función NOR exclusiva.

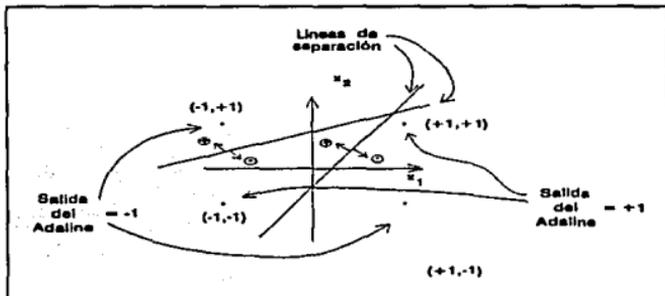


Fig.5.10.- Gráfica donde se observa el límite de separación para la red de la figura 5.9.

Muchas de estas redes fueron construidas con un número mayor de entradas, así como de elementos Adaline en el primer nivel, y un número variado de dispositivos lógicos estables, de entre los que destacan el dispositivo AND, el OR y el de la Mayoría de Elementos Tomados del segundo nivel. Estos tres dispositivos o funciones son consideradas como funciones lógicas preestablecidas.

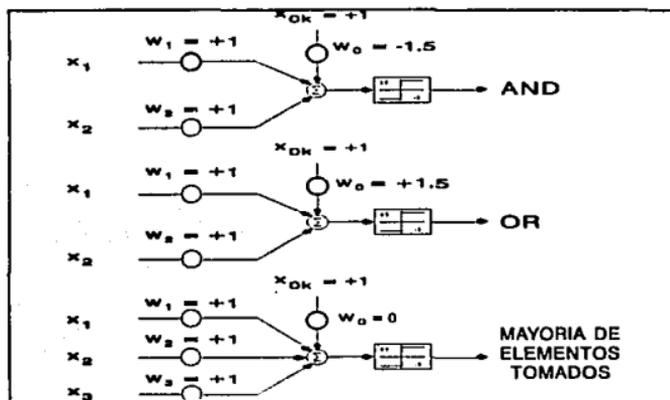


Fig.5.11.- Dispositivos lógicos estables: AND, OR y el de la Mayoría de Elementos Tomados.

Para las redes neuronales artificiales que involucran los valores de pesos conocidos, se relacionan con cualquiera de las tres funciones anteriores, pero con un factor muy importante, que las selecciones de los pesos no son las únicas; esto se puede apreciar en la figura 5.11. También tenemos que durante los años 60's las redes de tipo MADALINE tenían una estructura que consistía de elementos adaptivos en el primer nivel (de entrada), y de funciones preestablecidas estables en el segundo (de salida), aunque hoy en día las redes neuronales con alimentación hacia adelante, frecuentemente involucran en su estructura muchos niveles, y regularmente la mayoría de estos niveles son del tipo adaptivo.

En la figura 5.12 se observa una red del tipo MADALINE, la cual está formada

de tres niveles, que a su vez involucran diversos elementos Adaline, los cuales se encuentran conectados en su totalidad, es decir, cada elemento Adaline recibe en sus entradas todas las salidas provenientes del nivel precedente.

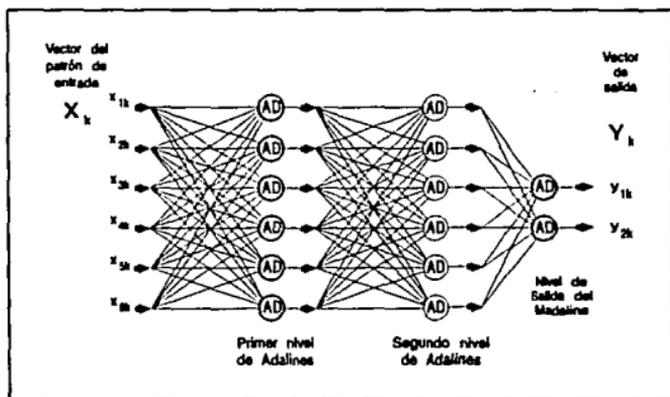


Fig.5.12.- Red neuronal artificial de tipo MADALINE, con alimentación hacia adelante y formada por dos niveles ADALINE.

Durante su funcionamiento, la respuesta de cada elemento de salida es comparado con su correspondiente respuesta deseada; en tanto, las señales de error asociadas con cada uno de estos elementos son parcialmente procesadas, de esta manera, la adaptación del nivel de salida es directa. Además, uno de los problemas fundamentales de las redes multiniveles con arquitectura MADALINE, se encuentra en la obtención de las señales de error para los elementos de las redes Adaline que se encuentran en los niveles ocultos, pero esto no es un factor determinante para decir que una red con alimentación hacia adelante no deba tener una estructura multinivel.

Finalmente, retomando el problema que se presenta frecuentemente en los sistemas de reconocimiento de patrones, tenemos que la invariancia que existe en la salida de la red, ante los cambios presentes tanto en la posición como en el tamaño

del patrón de entrada o de la imagen, es mínima. Para esto, existen varias técnicas que han sido utilizadas para solucionar problemas, entre los que se encuentran: el movimiento de traslación, rotación y escala de imágenes, así como la eliminación de los efectos de invariancia en el tiempo. Estas cuatro propiedades de las redes con estructura MADALINE, son esenciales para cualquier sistema que se desee aplicar dentro del campo de la visión artificial como es el reconocimiento de patrones o los sensores infrarrojos; esto se debe, a que los humanos no siempre tienen la capacidad de reconocer al momento ciertos objetos, sino hasta que han sido rotados u orientado, o en su defecto han sufrido una reducción o ampliación en su escala.

Además, dentro de los métodos de entrenamiento de redes, el ya mencionado algoritmo LMS tiene una característica muy importante, que dependiendo de ésta, sabemos a qué tipo de redes es aplicable, ya que se basa en que sólo puede operar sobre las salidas analógicas provenientes de los combinadores lineales adaptivos, y no siendo así, operable para los valores de salida de tipo bipolar que proporcionan las redes de estructura ADALINE; debido a esto, con una estrategia diferente de entrenamiento, puede ser desarrollada, entrenada y aplicada una red con arquitectura MADALINE.

Finalmente daremos un ejemplo representativo del modelo de redes neuronales artificiales más frecuentemente usado, el ADALINE. Este ejemplo se basa en un programa codificado en lenguaje C escrito por Maureen Caudill, el cual simula un entrenamiento de una unidad Adaline utilizando la Regla Delta. Básicamente este sistema artificial es entrenado para aprender a reconocer dos tipos diferentes de patrones visuales, que consisten en los caracteres V y V invertida. Estos patrones son simulados mediante una matriz de 5x5 plasmada sobre una "retina" simulada, la cual consta de 25 entradas en total.

Esta simple simulación almacena la respuesta correcta en un arreglo, al que se le consulta cada vez que los pesos van a ajustarse. La simulación del entrenamiento comienza con la inicialización de los pesos con valores aleatorios; después se presenta el primer patrón de entrada (el carácter V) dando comienzo el ajuste de los pesos

mediante la Regla Delta hasta que se reconozca la salida correcta. Los valores de las salidas fueron elegidos de la siguiente manera: el patrón **V** tiene valor de -1 y el patrón **V** invertido de +1. El siguiente paso es la introducción del segundo patrón (**V** invertida) y de igual forma, los pesos son ajustados hasta que se reconoce la salida deseada.

Un punto importante es el de verificar que el reconocimiento del segundo patrón no deba alterar el reconocimiento del primer patrón. Para evitar esto, se reajustan los pesos del primer patrón mientras se reconoce el segundo, para que ambos generen la salida correcta.

Cuando ambos patrones han sido reconocidos, decimos entonces que la red ha aprendido a distinguir entre una **V** y una **V** invertida, esto mediante la inspección de los valores de los pesos que se generan durante el proceso de entrenamiento, con la cual podemos observar el ajuste preciso del vector de pesos que se genera para el reconocimiento de los patrones.

Para lograr que una Adaline logre reconocer el caracter **V** y/o **V** invertida sin importar su ubicación sobre la retina simulada, podemos colocar varias Adalines que reconozcan el caracter **V** de forma similar, es decir, tanto en la parte superior como inferior de la retina. Estos sistemas son considerados como Madaline.

```

/***** */
/*      A D A L I N E      */
/*      */
/* Simulación de aprendizaje de una red neuronal artificial */
/* utilizando la Regla Delta (Widrow-Hoff)      */
/*      */
/* Escrito por Maureen Caudill en lenguaje C      */
/***** */

#include <math.h> /* incluir funciones matemáticas */
#include <stdio.h> /* incluir funciones de entrada-salida */

#define NUMRENS 5
#define NUMCOLS 5
#define NUMPATS .2
#define UMBRALE 0
#define BETA 0.1

/*-----*/
/* Almacenamiento global de variables y arreglos */
/*-----*/
/* El ADALINE tiene un total de NUMRENS*NUMCOLS pesos, */
/* almacenados en una matriz */
/*-----*/

double adaline[NUMRENS][NUMCOLS];
double entra1[NUMRENS][NUMCOLS];
double entra2[NUMRENS][NUMCOLS];
double respues[NUMPATS];
double interior1, interior2; /* magnitud de cada patrón de entrada */
FILE *fopen(), *outfile;
char *path = "widscrip1";
int maxren, maxcol;

```

```

/*-----*/
/*          printpesos()          */
/* Imprime el valor actual de los pesos de la ADALINE */
/*-----*/
printpesos()
{
int i,j;
for (i=0; i<NUMRENS; i++)
{
printf("\n");
for (j=0; j<NUMCOLS; j++);
printf(" %8.3lf", adaline[i][j] );
}
printf("\n");
return;
} /* fin de printpesos */

/*-----*/
/*          init_entras()          */
/* Inicializa los patrones de entrada con las entradas */
/* deseadas, así como calcula las magnitudes de las */
/* entradas y las almacena en las variables globales */
/* interior1 e interior2 para ser usadas posteriormente */
/* por el procedimiento adjust_pes */
/*-----*/
init_entras()
{
int row,col;
int i,j,k;
double size;

/* inicialización de la escritura de salida al archivo */
outfile = fopen(path,"W");

```

```
/* Inicialización del tamaño de maxren, maxcol */
maxren = NUMRENS;
maxcol = NUMCOLS;

/* primer conjunto de señales de salida a -1 */
for (row=0; row<NUMRENS; row++)
{
    for (col=0; col<NUMCOLS; col++)
    {
        entra1[row][col] = 0;
        entra2[row][col] = 0;
    }
}

/* elaboración del patrón V con los valores de entra1 */
entra1[0][0] = 1.0; /* X X */
entra1[1][1] = 1.0; /* X X */
entra1[2][2] = 1.0; /* X */
entra1[1][3] = 1.0; /* _____ */
entra1[0][4] = 1.0; /* _____ */

/* elaboración del patrón V invertida con los valores de entra2 */
entra2[2][0] = 1.0; /* X */
entra2[1][1] = 1.0; /* X X */
entra2[0][2] = 1.0; /* X X */
entra2[1][3] = 1.0; /* _____ */
entra2[2][4] = 1.0; /* _____ */
printf("\n Muestra de salida del sistema ADALINE: \n");
printf("\n Patrón de entrada 1:");
for (i=0; i<NUMRENS; i++)
{
    printf("\n");
    for (j=0; j<NUMCOLS; j++);
}
```

```

    printf( " %8.3lf", entra1[i][j] );
}
printf("\n Patrón de entrada 2:");
for (i=0; i<NUMRENS; i +)
{
    printf("\n");
    for (j=0; j<NUMCOLS; j +)
        printf( " %8.3lf", entra2[i][j] );
}
printf("\n Oprime ENTER para continuar....");
scanf("%c",&i);

/* asignación al cuadrado de la magnitud como valor 5 */
interior1 = 5.0;
interior2 = 5.0;
return;
} /* fin de init_entras */

/*-----*/
/*          init_pes()          */
/* inicialización de los pesos aleatoriamente entre -1 y */
/* + 1. Observese que los valores de los pesos no son */
/* necesariamente normalizados          */
/*-----*/
init_pes()
{
    int  ranvalue;
    unsigned int seed;
    double weighth;
    int  row,col;
    printf("Introduce un valor entero aleatorio: ");
    scanf("%d",&seed);
    ranvalue = srand(seed);

```

```

for (row=0; row<NUMRENS; row+ +)
{
for (col=0; col<NUMCOLS; col+ +)
{
ranvalue = rand(); /* selección del número aleatorio entero */
weigth = (16384. - (double)ranvalue)/16384.;
/* para lograr que los pesos sean "+" o "-" */
adaline[row][col] = weigth;
} /* fin para cada columna */
} /* fin para cada renglón */
return;
} /* fin de init_pes */

/*-----*/
/*          init_respues()          */
/* Inicialización del arreglo de respues para */
/* proporcionar las salidas correctas      */
/*-----*/
init_respues()
{
/* colocacion de respues utilizando un método primitivo */
respues[0] = -1;
respues[1] = 1;
return;
} /* fin de init_respues */

/*-----*/
/*          out_señal(pattnum)      */
/* Cálculo de la señal de salida del ADALINE para el patrón */
/* de entrada "pattnum". El valor regresado tiene un valor */
/* de doble punto flotante.          */
/*-----*/
double out_señal (pattnum)

```

```

int pattnum;
{
int row,col;
double result;
result = 0;
for(row=0; row<NUMRENS; row++)
{
for (col=0; col<NUMCOLS; col++)
{
switch (pattnum)
{
case 0:
result += entra1[row][col]*adaline[row][col];
break;
case 1:
result += entra2[row][col]*adaline[row][col];
break;
} /* fin del switch */
} /* fin para cada columna */
} /* fin para cada renglón */
if (result >= (double) UMBRALE)
result = 1.0;
else
result = -1.0;
return (result);
} /* fin de out_senaal */

/*-----*/
/*          adjust_pes(pattnum,res)          */
/* Ajuste de los pesos del ADALINE de acuerdo a la Ley de */
/* Aprendizaje Widrow-Hoff:                          */
/* w_act = w_ant + (BETA * Error * Input) / ((Input^2)) */
/*-----*/

```

```
adjust_pes(patt,res)
int  patt; /* número de patrones de entrada */
double res; /* salida actual */
{
double error, w_ant, w_act;
double in, insize;
int  row,col;
double beta;
beta = BETA;
error = respues[patt] - res; /* Error = salida ideal-salida actual */
for (row=0; row<NUMRENS; row + +)
{
for (col=0; col<NUMCOLS; col + +)
{
switch (patt)
{
case 0:
in  = entra1[row][col];
insize = interior1;
break;
case 1:
in  = entra2[row][col];
insize = interior2;
break;
}
w_ant = adaline[row][col];
w_act = w_ant + beta*error*in/insize;
adaline[row][col] = w_act;
} /* fin para cada columna */
} /* fin para cada renglón */
return;
} /* fin de adjust_pes */
```

```
/*-----*/
/*  PROGRAMA PRINCIPAL  */
/*-----*/
main()
{
int  patt, prev_patt;
int  ok;
double ans,res;
double out_señal();
int  in;
int  i,j,k,row,col;
clrscr();

/* Inicialización del sistema artificial */
init_pes();
init_entras();
init_respues();
maxren = NUMRENS;
maxcol = NUMCOLS;

/* Visualización de los pesos iniciales del ADALINE */
printf("\n\n Pesos iniciales:");
printpesos();
printf("\n");

/* Respuestas del ADALINE para cada patrón de entrada */
for (patt = 0; patt < NUMPATS; patt++)
{
printf("\n\n Comienza la adaptación del patrón %d", patt);
res = out_señal(patt);
printf("\n La respuesta es %8.3lf y debe ser %8.3lf.", res,
respues[patt]);
while (res != respues[patt])
```

```
{
adjust_pes(patt,res);
printf("\n Ajustando los pesos del patrón %d",patt);
printf("\n Valores actuales de los pesos: ");
printpesos();
res = out_señal(patt);
printf("\n La respuesta actual es %8.3lf y debe ser %8.3lf.",
respues[patt]);
if (res == respues[patt])
printf("\n Patrón identificado correctamente");
else
{
printf("\n Ajusta otra vez? (Oprima '1' para continuar)");
scanf("%d",&in);
if (in != 1)
res = respues[patt];
} /* fin del if-else */
} /* fin del while */
/* Proceso para hacer un reajuste en los valores de los pesos */
/* en caso de que el patrón no haya sido identificado. */
for (prev_patt=0; prev_patt<patt; prev_patt++)
{
while (res == out_señal(prev_patt) /* agregue un signo = */
! = respues[prev_patt])
{
adjust_pes(prev_patt,res);
printf("\n Re-ajustando los pesos del patrón %d",prev_patt);
printf("\n Valores actuales de los pesos: ");
printpesos();
} /* fin del while */
} /* fin del proceso prev_patt */
printf("\n El sistema ADALINE reconoció el patrón %d",patt);
} /* fin del proceso de reconocimiento de cada patrón */
```

```
/* Proceso para checar que todos los patrones fueron reconocidos */
printf("\n\n El sistema Adaline está capacitado para reconocer todos los patrones.");
printf("\n Los valores finales de los pesos son: ");
printpesos();
printf("\n\n Patrón #      respues      Salida");
for (patt = 0; patt < NUMPATS; patt + +)
{
  res = out_señal(patt);
  printf("\n  %d      %3.0lf      %3.0lf",patt,respues[patt],res);
}
printf("\n\n Todos los patrones han sido aprendidos....");
printf("\n\n El programa ADALINE ha terminado....");
return;
} /* fin del programa principal */
```

CONCLUSIONES

De acuerdo al contenido del presente trabajo se pudo observar la importancia que tuvieron, tienen y tendrán los sistemas artificiales, en especial los **Sistemas Neuronales Artificiales** dentro de la vida cotidiana, debido a que con estos sistemas las tareas complejas pueden desarrollarse de una forma más rápida y sencilla, además de proporcionar en otros aspectos una comodidad con los sistemas instalados en el hogar y los destinados a la recreatividad.

Debido a esto se han creado varias expectativas acerca de estos sistemas ya que un gran número de la población no puede entender el concepto de las máquinas "casi humanas", es decir que su comportamiento se asemeje cada vez más al de un ser humano, como por ejemplo, tener la capacidad de percibir, aprender, razonar y dar soluciones a los problemas que se le planteen.

Por esto, podemos estar completamente seguros de que los sistemas neuronales artificiales no podrán reemplazar en su totalidad a las computadoras digitales convencionales, ya que éstas últimas se utilizan principalmente para la ejecución de subrutinas y cálculos matemáticos, así como para el procesamiento de textos, diseños por computadora, la transferencia y procesamiento de datos entre otras tareas.

En tanto, los sistemas neuronales artificiales nos proporcionan la posibilidad de simular sistemas físicos, como por ejemplo, tareas en donde se involucren la clasificación, asociación y tal vez el razonamiento de una secuencia aritmética, o también en aplicaciones en donde se requiera de un preciso control en tiempo real para sistemas complejos como lo pueden ser los procesadores de señales.

Además estos sistemas artificiales pueden ser aplicados junto con los sistemas expertos, es decir, sirven como auxiliares para los diagnósticos médicos, servicios financieros, predicciones en las bolsas de valores, entre otras aplicaciones.

Por todo lo anterior, este trabajo de investigación fue realizado con el objetivo de dar a conocer las bases sobre las cuales se encuentran las redes neuronales artificiales, con el propósito de que pueda servir para estudios posteriores con aplicaciones específicas tanto en software como en hardware; entre las que se encuentran: el diseño de nuevas redes, el estudio de sus comportamientos, el análisis de sus resultados y la creación de ideas para poder implementarlas adecuadamente.

BIBLIOGRAFIA

Anderson J.J. Silverstein, S. Ritz and R. Jones.

Distinctive Features, Categorical Perception and Probability Learning: Some Application a Aeural Model.

Pysch. Rev., Vol. 84, 1977.

Arun V. Holden and Vitaly I. Kryukov

Neuro Computers and Attention: Connectionism and Neurocomputers.

Manchester University Press., U.S.A., 1991.

Bart Kosko.

Neural Networks and Fuzzy Systems.

Ed. Prentice Hall, Englewood Cliffs, N.J. 1992.

Branko Soucek and Iris Group.

Neural and Intelligent System Integration: Fifth and sixth generation. Integrated Reasoning: Information Systems.

Ed. John Wiley and Sons., Inc., US.A. 1991.

Branko Soucek and Marina Soucek.

Neural and Massively Parallel Computers: The sixth generation.

Ed. Wiley-Interscience Publication, U.S.A. 1988.

Clifford Lau.
Neural Networks.
Theoretical Foundations and Analysis.
Ieee Press., Piscataway, N.J. 1991.

David S. Touretzky.
Advanced in Neural Information Processing System.
Vol. 2.

Drew Van Camp.
The Amateur Scientist.
Scientific American.

Edgar Sánchez - Sinencio and Clifford Lau.
Artificial Neural Networks.
Ieee Press., Piscataway, N.J. 1992.

Fukushima K.
Analysis of the process of visual pattern recognition by the neocognitron.
Neural Networks Vol. 2.

Fukushima K.
Neocognitron: A self-organizing Neural Network model for a
mechanism of pattern recognition unaffected by shift in
position.
Biol. Cybern., Vol 36, 1980.

Fukushima K.

Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition.
Neural Networks, Vol. 1, 1988.

Hebb D.

Organization of Behavior.
Wiley, New York, 1949.

Hopfield J.J.

Neural Networks and Physical System whit Emergent Collective Computation Abilities.
Proc. Natl. Acad. Sci., Vol. 79, April 1982.

Hopfield J.J.

Neuron with grade response have collective computational properties likes those of two-state neurons.
Proc. Nat. Acad. Sci. U.S.A., Vol. 81, May 1984.

Jacek M. Zurada.

Introduction to Artificial Neural Systems.
West Publishing Company, U.S.A., 1992.

James A. Freeman and David M. Skapura.

Neural Networks.
Algorithms, Applications and Programming Techniques.
Ed. Addison - Wesley Publishing Company, U.S.A. 1992.

Kohonen T.

Learning Vector Quantization for Pattern Recognition.
Helsinki University of technology, 1986.

McKenna Thomas, Joel Davis and Steven F. Zornetzer.

Single Neuron Computation.
Academic Press. Inc., U.S.A., 1992.

Patrick Simpson.

Foundation of Neural Networks.

Roseblatt F.

Principles of Neurodynamic.
Spartan Books, Washington, D.C., 1962.

Stephen Grossberg.

Nonlinear Neural Networks: Principles, Mechanism and Architectures.
Neural Networks Vol. 1, No 1, 1988.

Thomas McKenna, Joel Davis and Steven F. Zornetzer.

Single Neuron Computation.
Academic Press, Inc., U.S.A., 1992.

Transaction On.

IEEE Neural Networks.
Vol. 2, No 3, May 1991.

Transaction On.
IEEE Neural Networks.
Vol 1, No 1, March 1990.

Widrow B. and M. Hoff.
Adaptive Switching Circuits.
Wescon Convention Records, Vol. 4, 1960.