

UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA



**“DETECCIÓN DE CARACTERES MEDIANTE REDES
NEURONALES CON ENTRADA DE DATOS CON EXCEL”**

**INFORME DE SUFICIENCIA
PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO ELECTRÓNICO**

**PRESENTADO POR:
EDWIN TELLO GODOY**

**PROMOCIÓN
1980 - I**

**LIMA - PERU
2005**

DEDICATORIA

Dedico este trabajo a mi hijo quien me
alentó a culminar éste Informe.

**“DETECCIÓN DE CARACTERES MEDIANTE REDES
NEURONALES CON ENTRADA DE DATOS CON EXCEL”**

SUMARIO

En las aplicaciones la ventaja de las redes neuronales artificiales frente a otras técnicas está en el procesamiento paralelo, adaptivo y no lineal. Las técnicas desarrolladas mediante asociadores de patrones y algoritmos de retropropagación para muchos fines con RNA.

La estructura del cerebro es radicalmente diferente a la de la computadora convencionalmente no está compuesto por un único microprocesador complejo sino por miles de millones de ellos. En el reconocimiento de patrones que ni las super computadoras son capaces de resolver de un modo eficiente, mientras que el cerebro humano lo viene haciendo con facilidad y eficiencia.

Las redes neuronales para aplicaciones concretas no están ligadas a las redes neuronales biológicas. Utilizan algoritmos de aprendizaje adaptivo y autoorganizativos como el procesamiento no lineal aumenta la capacidad de la red de aproximar y su inmunidad frente al ruido y un alto nivel de interconectividad. Estas características juegan un papel importante en las redes RNA aplicadas en el procesamiento de señal e imagen y caracteres.

El conjunto de técnicas de carácter heurístico más o menos estándar producen resultados satisfactorios en muchos campos con éxito. Se apoyan en paradigmas válidos en todos los problemas de diseño.

En los sistemas neuronales de procesamiento de información las entradas deben ser tratadas de manera que produzcan a la salida resultados útiles de acuerdo con el problema, un ejemplo del sistema de procesamiento de información donde la entrada que es imagen binaria que contiene texto manuscrito debe ser procesado de manera que en la salida del sistema produzca caracteres en el código .ASCII .

Las redes neuronales están solucionando problemas en muchos campos con éxito y con error aceptable en el que los métodos tradicionales (Técnicas heurístico) tienen dificultades.

INDICE

	<u>Pág</u>
PROLOGO	
CAPITULO I	
SISTEMA DE TIEMPO REAL DE CONMUTACIÓN REMOTA	
1.1. SISTEMAS EN TIEMPO REAL	3
1.2. SISTEMAS MILITARES	4
1.3. SISTEMAS DE CONTROL DE TRÁFICO AÉREO	5
1.4. TIEMPO COMPARTIDO	5
1.5. TIEMPO REAL EN TELE-EDUCACIÓN	6
CAPITULO II	
NEURONA BIOLÓGICA Y ASPECTOS GENERALES	
2.1. INTRODUCCIÓN	11
2.2. NEURONAS BIOLÓGICAS	13
2.3. HISTORIA DE REDES NEURONALES ARTIFICIALES	16
2.4. SISTEMA EXPERTO	20
2.5. LA NEURONA ARTIFICIAL	21
2.6. ENTRENAMIENTO DE REDES NEURONALES	23
2.7. MÉTODO DE TRANSMISIÓN DE LA INFORMACIÓN EN EL CEREBRO	27
2.8. DIFERENCIAS ENTRE EL CEREBRO Y UN ORDENADOR	29

CAPITULO III**REDES NEURONALES**

3.1.	FUNDAMENTOS DE REDES NEURONALES	35
3.2.	VENTAJAS DE LOS ANS	36
3.3.	FUNCIONAMIENTO	38

CAPITULO IV**TIPOS DE REDES NEURONALES MÁS IMPORTANTES**

4.1.	REDE NEURONAL MULTICAPA	41
4.2.	RESUMEN DE LAS REDES NEURONALES	43
4.3.	APRENDIZAJE DE LAS REDES NEURONALES	44
4.4.	EL PERCEPTRON	47
4.5.	REGLA DE APRENDIZAJE DEL PERCEPTRON	49
4.6.	EL PERCEPTRON MULTINIVEL	50
4.7.	RED BACKPROPAGATION	51
4.8.	APLICACIÓN DE LA RED BACPROPAGATION	53
4.9.	REDE KOHONEN	54

CAPITULO V**RECONOCIMIENTO DE CARACTERES UTILIZANDO EXCEL**

5.1.	USO DE LA RED BACKPROPAGATION EN EL RECONOCIMIENTO DE PATRONES	59
5.2.	DESCRIPCIÓN DEL PROYECTO	64
5.3.	ESTRUCTURA DE LA RED	65

5.4.	FORMATO DE DATOS ENTRADAS Y SALIDAS	66
5.5.	PROCESAMIENTO DE DATOS	70
5.6.	DESCRIPCIÓN DEL PROCEDIMIENTO UTILIZANDO NEUROSHELL2	71
5.7.	RESULTADOS	82

CONCLUSIONES

BIBLIOGRAFÍA

PRÓLOGO

En este trabajo se estudia el problema de la clasificación automática de patrones de caracteres escritos a mano. Su campo de aplicación es muy amplio, por ejemplo, reconocimiento de Códigos Postales de cartas y cheques de banco. Los patrones pueden estar desplazados, escalados, distorsionados e incluso con cierta inclinación, o sobrescritos. El método usado se basa en la utilización de una red neuronal multicapa que ha sido entrenada con el algoritmo de Backpropagation. La entrada a la red neuronal son un conjunto de vectores de características obtenidos mediante un preprocesado, que lleva consigo la captura, segmentación, normalización, escalado y compresión de los patrones originales. A este preprocesado se le añade una extracción de vectores o mapas de características, basados en el algoritmo de detección de esta manera se consigue introducir en la red neuronal información de más alto nivel y prescindir de información redundante o de tipo ruidoso.

Para el entrenamiento de la red neuronal se ha creado una base de datos escritos a mano. El método de estimación del error de generalización de la red, para el reconocimiento de patrones nunca antes “vistos” por la red, es el de Validación Cruzada; en el que se particiona toda la base de datos, repetidas veces, entrenando con una partición y validando con el resto.

Los resultados obtenidos mediante el método estudiado son bastante alentadores pues nos permiten alcanzar unos porcentajes de reconocimiento correcto con un porcentaje bastante alto de caracteres manuscritos.

CAPITULO I

SISTEMAS DE TIEMPO REAL DE COMPUTACIÓN REMOTA

1.1. SISTEMAS EN TIEMPO REAL

Un sistema de tiempo real tiene una relación temporal paralela con una actividad que se lleva a cabo continuamente, por lo que produce información con la rapidez necesaria para ayudar a controlar dicha actividad dinámica. Así, las palabras "tiempo real" describen un sistema de procesamiento interactivo con severas limitaciones. Un sistema de tiempo real utiliza el procesamiento interactivo, pero no es forzoso que un sistema interactivo opere en tiempo real. La diferencia es que el procesamiento en tiempo real requiere la introducción inmediata de las transacciones provenientes de todas las terminales en las que se originan entradas. Se conectan en forma directa muchas estaciones por medio de líneas de telecomunicación de alta velocidad a uno o más procesadores. Pueden operar varias estaciones al mismo tiempo. Los archivos se actualizan a cada minuto y las consultas se atienden por medio del acceso, en fracciones de segundo, a archivos completamente actualizados. En cambio, es posible tener un sistema interactivo que combine el acceso inmediato

a los registros para atender consultas con la introducción periódica (quizá diaria) de transacciones provenientes de una fuente central recolectora para actualizar los registros. Un sistema así podría satisfacer muchas necesidades y ser más sencillo y económico que un sistema en tiempo real.

Los sistemas de reservaciones que utilizan las líneas aéreas, hoteles y agencias de alquiler de vehículos son ejemplos de sistemas de tiempo real que ya se han mencionado antes. Otro ejemplo de transmitir en tiempo real de video y audio es la red ATM.

1.2. SISTEMAS MILITARES

En los Sistemas militares se ha desarrollado un sistema de mando y control militar mundial para los comandantes militares de Estados Unidos, que comienza con el presidente, continúa descendiendo en la jerarquía. El sistema conecta 35 computadoras grandes en 26 puestos de mando en todo el mundo. Asimismo, más de una docena de computadoras en el mando de Defensa Aérea de Estados Unidos (North American Air Defense Command) aceptan, almacenan y actualizan constantemente montañas de datos provenientes de instalaciones de radar en todo el globo. Se sigue la trayectoria de todos los objetos en órbita terrestre producidos por el hombre. Si se lanza un cohete, las computadoras calculan con rapidez su trayectoria.

1.3. SISTEMAS DE CONTROL DE TRÁFICO AÉREO

Los controladores de tráfico aéreo siguen las trayectorias de millones de vuelos en todo Estados Unidos. Las computadoras supervisan los vuelos y los transfieren a diferentes jurisdicciones de control, conforme se desplazan sobre el continente. Cuando un vuelo se acerca al Aeropuerto Internacional de Chicago, por ejemplo, un sistema de control transmite una señal "guía" especial. El avión responde con señales que proporcionan su identidad, altitud y velocidad. Las computadoras procesan estos datos y los exhiben instantáneamente en la pantalla de un controlador, junto con un punto luminoso que representa al avión. El sistema es capaz de controlar y seguir, en forma simultánea, a más de 100 aviones.

1.4. TIEMPO COMPARTIDO

Es un término general que se emplea para describir un sistema de procesamiento que cuenta con varias estaciones independientes, de relativamente baja velocidad, en línea y que se pueden utilizar simultáneamente. Por supuesto, todas las estaciones tienen acceso directo al procesador. El uso de programas especiales permite al procesador conmutar de una estación a otra y realizar una parte de cada trabajo en la "porción" de tiempo que tiene asignada hasta terminar el trabajo. En muchas ocasiones el proceso es tan rápido que el usuario tiene la ilusión de que nadie más está utilizando la computadora.

Existen varias organizaciones que venden servicios de tiempo compartido y de cómputo remoto a sus clientes. En algunos casos estas organizaciones instalan terminales o computadoras personales en las oficinas de los clientes y después conectan estas estaciones de trabajo a sus procesadores centrales por medio de canales de telecomunicación. En otros casos conectan sus procesadores a las computadoras personales que ya poseen los clientes. Es posible procesar una gama muy amplia de tareas, pero algunas organizaciones de servicio se especializan en las necesidades de un grupo específico. Otros servicios de computación remota (llamados a veces SERVICE BUREAUS, oficinas de servicio) aceptan los datos de entrada del cliente a través de líneas de telecomunicación, efectúan un proceso por lotes especial para el cliente y más tarde transmiten los resultados a la terminal del cliente. Las organizaciones de servicio de tiempo compartido y cómputo remoto suelen ofrecer una biblioteca en línea de programas de aplicación a sus clientes, quienes sólo necesitan proporcionar los datos de entrada y llamar a los programas para obtener la información deseada. Los clientes pagan cargos por transacción que varían (como las llamadas telefónicas de larga distancia).

1.5. TIEMPO REAL EN TELE-EDUCACIÓN

El tan promocionado sistema de tele-educación a distancia a través de Internet, denominado Plan Huascarán, sin lugar a dudas tiene muy

buenas intenciones por parte de los entes gubernamentales promotores.

Sin embargo, utilizando Tecnologías de la Información y sin enfocar este tema políticamente, vamos a tocarlo de un modo netamente técnico.

Hasta la fecha, no se han definido los aspectos fundamentales, como son su infraestructura tecnológica y la capacidad instalada a emplearse en este importante proyecto.

No es posible hablar de un sistema de tele-educación a distancia, sin definir en primer lugar, la disponibilidad de los anchos de banda necesarios y la velocidad de la transmisión de datos, voz, sonido e imágenes vía satelital, en tiempo real, vale decir la capacidad de las comunicaciones en esta red vial.

Se estima que en todo el Perú, un país que bordea los 26 millones de habitantes, el total de ancho de banda disponible de las empresas establecidas como Proveedores de Servicios de Internet (ISP), a Noviembre del 2001 no supera los 600 Mbps, en limitadas conexiones análogas T1 (1.544 Mbps) que emplean desde el obsoleto par de alambres cobre, en algunos casos y en otras, cuentan como un máximo de conexiones OC-3 (155.52 Mbps).

La mayoría de corporaciones de servicios de Internet en los Estados Unidos, Canadá y Europa, cuentan individualmente con conexiones digitales OC-48, que permiten una velocidad de transmisión de 2.488 Gbps. Para ilustrar al interesado, el ancho de banda equivaldría a la capacidad del circuito de acceso y navegación en Internet y el tipo de conexiones denotan las velocidades de transmisión de datos.

En segundo lugar, cualquier computadora dotada de una cámara web (Web Cam), tal como fuera demostrado recientemente en algunos noticieros de la televisión, no está en la capacidad transmitir y mostrar imágenes en tiempo real. Estos equipos de bajo costo tienen una velocidad máxima de 15 cuadros por segundo, en una resolución de 640 x 480 pixels, cuando lo mínimo requerido son 30 cuadros por segundo a una resolución de 1024 por 768 pixels.

En caso contrario, los alumnos no podrán coordinar mentalmente la voz en tiempo real, con imágenes de baja resolución y en cámara lenta de los profesores. Para este propósito se requieren costosos sistemas de video conferencia. Por otro lado, las computadoras empleadas tanto por los profesores como los alumnos, deben estar equipadas con potente microprocesadores Pentium III de 600 Mhz en adelante, memoria RAM de 128 o 256 MB y tarjetas de video de 32 o 64 MB.

Cada centro de estudios de Tele-educación a distancia debe estar dotado de un Ruteador digital y un servidor de red apropiado, con un Firewall seguro.

Los componentes de cada computadora deberán tener por lo menos doble ventilación y enfriamiento, ya que el video en uso continuo, en tiempo real, tiende a recalentar. Por si fuera poco, en un ancho de banda de 512 kbps, por ejemplo, el más alto porcentaje equivalente a 384 kbps será consumido por la transmisión de imágenes, quedando 128 kbps para la transmisión de voz.

Tanto el servidor de la red así como las computadoras personales deben contar con los correspondientes software licenciados.

En Bolivia y Chile algunas Universidades imparten tele-educación a distancia pero a través de videos pre-grabados y no en tiempo real, debido a las implicancias antes expuestas.

Muchos hablan de Internet, desconociendo sus expectantes posibilidades, limitaciones y grandes vulnerabilidades, como son la congestión o saturación de las transmisiones en las horas punta, caídas de los servidores de Red por fallas o falta de mantenimiento, incursiones no autorizadas realizadas por intrusos, conocidos como hackers y crackers, proliferación de virus informáticos, etc.

Se sabe que Arpanet es la precursora de Internet, en la década de los 60 en los EE. UU. y se puede afirmar que la frase "Internet es la supercarretera de la información" acuñada por el ex Vice presidente norteamericano Al Gore, 10 años atrás, se le puede agregar que por cierto "es una supercarretera de la información, pero todavía con muchos baches, que la ciencia hasta ahora está intentando reparar".

Se debe asesorar a los gestores del Plan con técnicos, altamente capacitados y con experiencia en dicho campo, pues todos los peruanos deseamos que la educación en el Perú definitivamente se expanda y mejore, siempre y cuando se empleen los medios apropiados.

CAPITULO II

NEURONA BIOLÓGICA Y ASPECTOS GENERALES

2.1. INTRODUCCIÓN

A finales del siglo 19 se logró una mayor claridad sobre el trabajo del cerebro debido a los trabajos de Ramón y Cajal en España y Harrington en Inglaterra. El primero trabajó en la anatomía de las neuronas y el segundo en los puntos de conexión de las mismas o sinápsis.

El tejido nervioso es el más diferenciado del organismo y está constituido por células nerviosas, fibras nerviosas y la neuroglia, que está formada por varias clases de células. La célula nerviosa se denomina neurona, que es la unidad funcional del sistema nervioso. Hay neuronas bipolares, con dos prolongaciones de fibras y multipolares, con numerosas prolongaciones. Pueden ser neuronas sensoriales, motoras y de asociación.

La estructura de una neurona se muestra en la figura 2. 1.

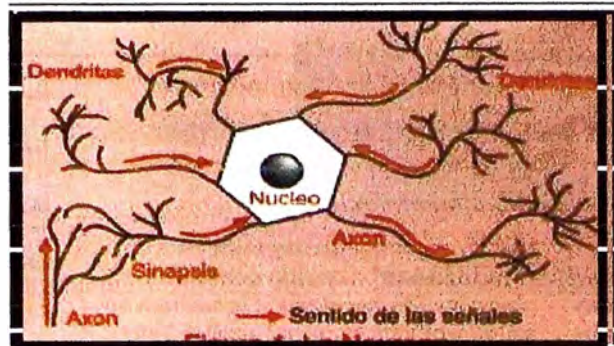


Figura 2.1. La Neurona

Se estima que en cada milímetro del cerebro hay cerca de 50,000 neuronas. El tamaño y la forma de las neuronas es variable, pero con las mismas subdivisiones que muestra la figura. El cuerpo de la neurona o Soma contiene el núcleo. Se encarga de todas las actividades metabólicas de la neurona y recibe la información de otras neuronas vecinas a través de las conexiones sinápticas.

Las dendritas son las conexiones de entrada de la neurona. Por su parte el axón es la "salida" de la neurona y se utiliza para enviar impulsos o señales a otras células nerviosas. Cuando el axón está cerca de sus células destino se divide en muchas ramificaciones que forman sinápsis con el soma o axones de otras células. Esta unión puede ser "inhibidora" o "excitadora" según el transmisor que las libere. Cada neurona recibe de 10.000 a 100.000 sinápsis y el axón realiza una cantidad de conexiones similar.

La transmisión de una señal de una célula a otra por medio de la sinápsis es un proceso químico. En él se liberan sustancias transmisoras en el lado del emisor de la unión. El efecto es elevar o disminuir el potencial eléctrico dentro del cuerpo de la célula receptora. Si su potencial alcanza el umbral se envía un pulso o potencial de acción por el axón. Se dice, entonces, que la célula se disparó. Este pulso alcanza otras neuronas a través de las distribuciones de los axones.

Las redes neuronales (RN) surgieron después de la Inteligencia Artificial (IA) simbólica o tradicional, los RN artificiales son el resultado de investigación académica que usa una metodología con conceptos matemáticos. Las redes neuronales se utilizan para aprender patrones y relaciones de datos.

2.2. NEURONAS BIOLÓGICAS

El cerebro humano está formado por células denominadas neuronas. Generalmente una neurona recibe información de miles de otras neuronas y a su vez, envía información a otras miles de neuronas. El poder del cerebro se mide por la cantidad de neuronas y las múltiples conexiones entre ellos. Hay 3 tipos de neuronas: Sensoriales, motoras y de asociación o Interneuronas cada una de ellas desempeñan funciones distintas.

- **Neuronas Sensoriales.-** Se encargan de conducir los impulsos desde todos los receptores sensoriales del cuerpo (ubicado en los distintos órganos de los sentidos) al sistema nervioso sensaciones y percepciones como el dolor y la luz.
- **Neuronas Motoras.-** Conducen los impulsos nerviosos desde el sistema nervioso central hacia los músculos o las glándulas. El cuerpo celular de las neuronas motoras se ubica en el sistema nervioso central.
- **Neuronas de Asociación.-** Vinculan a las neuronas motoras con las sensoriales reciben, procesan y envían a cada rincón del organismo. Se encuentra en el sistema nervioso central.

En la estructura de las neuronas están compuestas por el cuerpo o soma neuronal, las dendritas y el axón. El cuerpo celular aloja al núcleo y el citoplasma.

- **Las dendritas .-** Son conexiones de entrada de las neuronas. Son fibras nerviosas que transportan el impulso nervioso hacia el cuerpo celular.
- **El Axón.-** Son también fibras nerviosas son más largas y es la salida de la neurona y transmite impulsos nerviosos vitales. La

mielina que recubre los axones de algunas células nerviosas. La cubierta de mielina contribuye a mantener aislado a los axones para evitar que los impulsos en uno de ellos interfieran con los de los otros.

- **La sinapsis.-** Se llama así el espacio que separa el axón de una neurona sensorial de una neurona de asociación o interneuronas. Las partes de la neurona lo puede apreciar en la Figura 2.2.

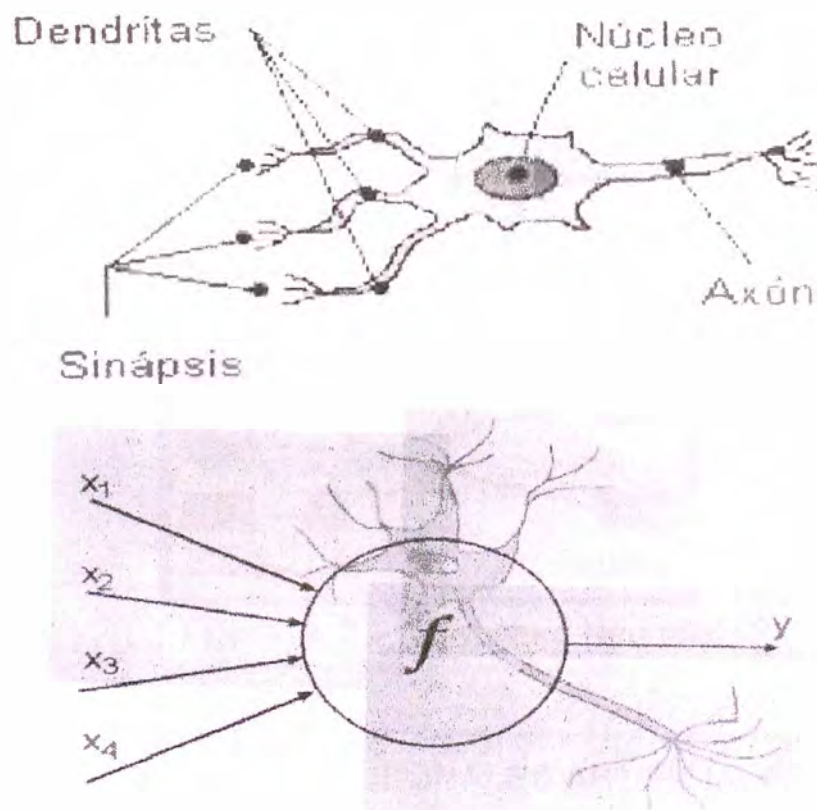


Figura 2.2. Estructura Básica de una neurona Biológica

El sistema de neuronas biológicas esta compuesto por neuronas de entrada (sensores) conectados a una compleja red de neuronas "calculadoras" (neuronas ocultas), las cuales, a su vez, están conectadas a las neuronas de salidas que controlan, por ejemplo, los músculos. La figura 2.3 muestra un esquema conceptual. Los sensores pueden ser señales de los oídos, ojos, etc. las respuestas de las neuronas de salida activan los músculos correspondientes. En el cerebro hay una gigantesca red de neuronas "calculadoras" ocultas que realizan la computación necesaria. De esta manera similar, una red neuronal artificial debe ser compuesta por sensores del tipo mecánico o eléctrico.

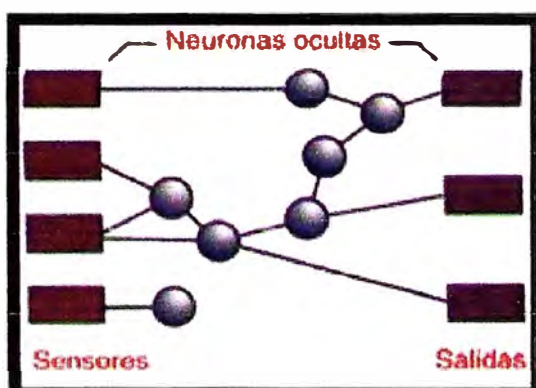


Figura 2.3. Estructura Neuronal

2.3. HISTORIA DE REDES NEURONALES ARTIFICIALES

Los intentos por imitar el funcionamiento del cerebro han seguido la evolución del estado de la tecnología. Por ejemplo, al finalizar el siglo 19 se le comparó con la operación de la bomba hidráulica; durante la década de 1920 a 1930 se intento utilizar la teoría de la conmutación

telefónica como punto de partida de un sistema de conocimiento similar al del cerebro. Entre 1940 y 1950 los científicos comenzaron a pensar seriamente en las redes neuronales utilizando como concepto la noción de que las neuronas del cerebro funcionan como interruptores digitales (on - off) de manera también similar al recién desarrollado computador digital. Así nace la idea de "revolución cibernética" que maneja la analogía entre el cerebro y el computador digital.

- **1943 Teoría de las Redes Neuronales Artificiales**

Walter Pitts junto a Bertran Russell y Warren McCulloch intentaron explicar el funcionamiento del cerebro humano, por medio de una red de células conectadas entre sí, para experimentar ejecutando operaciones lógicas. Partiendo del menor suceso psíquico (estimado por ellos): el impulso todo / nada, generado por una célula nerviosa.

El bucle "sentidos - cerebro - músculos", mediante la retroalimentación produciría una reacción positiva si los músculos reducen la diferencia entre una condición percibida por los sentidos y un estado físico impuesto por el cerebro.

También definieron la memoria como un conjunto de ondas que reverberan en un circuito cerrado de neuronas.

- **1949 Conductividad de la sinápsis en las Redes Neuronales.**

Seis años después de que McCulloch y Pitts mostraran sus Redes Neuronales, el fisiólogo Donald O. Hebb (de la Mc Hill University) expuso que estas (las redes neuronales) podían aprender. Su propuesta tenía que ver con la conductividad de la sinápsis, es decir, con las conexiones entre neuronas. Hebb expuso que la repetida activación de una neurona por otra a través de una sinápsis determinada, aumenta su conductividad, y la hace más propensa a ser activada sucesivamente, induciendo a la formación de un circuito de neuronas estrechamente conectadas entre sí.

- **1951 Primera Red Neuronal**

El extraordinario estudiante de Harvard, Marvin Minsky conoció al científico Burrhus Frederic Skinner, con el que trabajó algún tiempo ayudándole en el diseño y creación de máquinas para sus experimentos. Por aquel entonces entabló amistad con otro brillante estudiante, Dean Edmonds, el cual estaba interesado en el estudio de una nueva ciencia llamada Electrónica.

Durante el verano de 1951, Minsky y Edmonds montaron la primera máquina de redes neuronales, compuesta básicamente de 300 tubos de vacío y un piloto automático de un bombardero B-24. Llamaron a su creación "Sharc", se trataba nada menos que de una red de 40 neuronas artificiales que imitaban el cerebro de una rata.

Cada neurona hacia el papel de una posición del laberinto y cuando se activaba daba a entender que la "rata" sabía en que punto del laberinto estaba. Las neuronas que estaban conectadas alrededor de la activada, hacían la función de alternativas que seguir por el cerebro, la activación de la siguiente neurona, es decir, la elección entre "derecha" o "izquierda" en este caso estaría dada por la fuerza de sus conexiones con la neurona activada. Por ejemplo, la "rata" completaba bien el recorrido eligiendo a partir de la quinta neurona la opción "izquierda" (que correspondería a la sexta), es entonces cuando las conexiones entre la quinta y sexta se hacen más fuertes (dicha conexión era realizada por el piloto automático), haciendo desde este momento más propensa esta decisión en un futuro. Pero las técnicas Skinnerianas (que eran las que se habían puesto en funcionamiento en esta red neuronal) no podrían llevar muy lejos a este nuevo engendro, la razón pasa porque esto, en sí, no es inteligencia, pues la red neuronal nunca llegaría a trazar un plan.

Después de su Red Neuronal, Minsky escribió su tesis doctoral acerca de esta, en ella describía "cerebros mucho mayores", exponiendo que si se realizaba este proyecto a gran escala, con miles o millones de neuronas más y con diferentes sensores y tipos de retroalimentación la máquina podría ser capaz de razonar, más

el sabía que la realización de esta Red Neuronal era imposible y decidió buscar otra forma de crear inteligencia.

2.4. SISTEMA EXPERTO

Un método más avanzado para representar el conocimiento, es el sistema experto. Típicamente está compuesto por varias clases de información almacenada: Las reglas If - Then le dicen al sistema como se debe reaccionar ante los estados del "mundo". Una regla del sistema experto puede ser If Y es un hombre, Then Y es mortal. Los hechos describen el estado del "mundo". Por ejemplo: Juan es mortal. Por último, una máquina de inferencia relaciona los hechos conocidos con las reglas If - Then y genera una conclusión. En el ejemplo: Juan es mortal. Esta nueva conclusión se añade a la colección de hechos que se almacena en los medios ópticos o magnéticos del computador digital. De esta forma, un sistema experto sintetiza nuevo conocimiento a partir de su "entendimiento" del mundo que le rodea. De esta forma, un sistema experto es un método de representación y procesamiento del conocimiento, mucho más rico y poderoso que un simple programa de computador. Sin embargo, con respecto a la manera en que opera el cerebro humano, las limitaciones son múltiples.

- **Interpretación De La Neurona Por Computadora**

Una neurona se puede comparar con una caja negra compuesta por varias entradas y una salida. La relación de activación entre la salida y la entrada, o en términos circuitales o de teoría de control, la función de transferencia se encuentra en la figura 2.4.

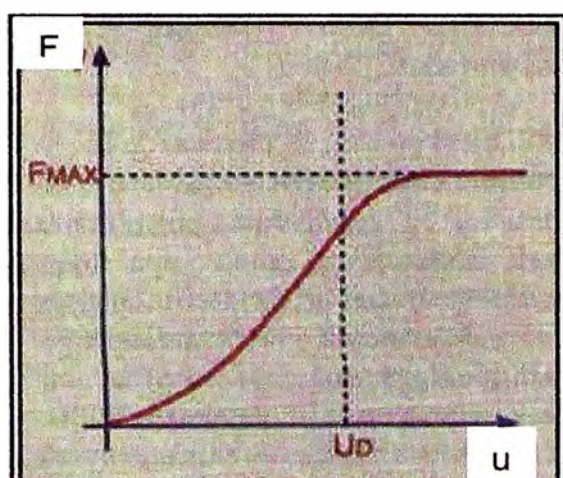


Figura 2.4. - Función de transferencia de una Neurona

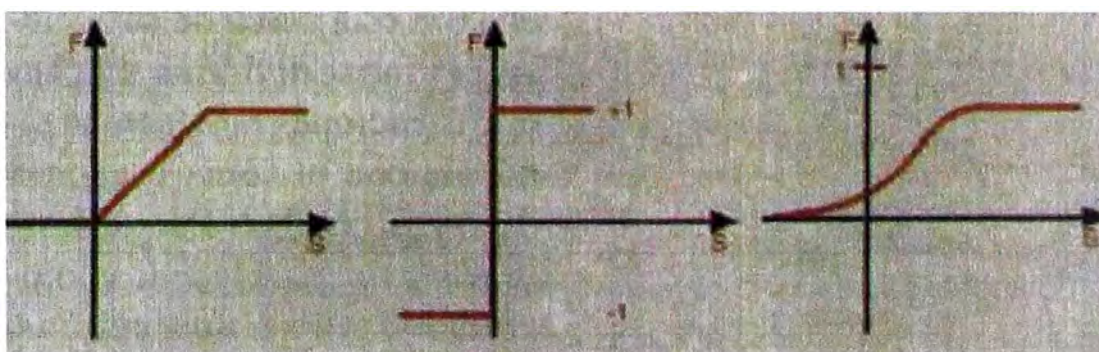
La variable F es la frecuencia de activación o emisión de potenciales u es la intensidad del estímulo del soma.

2.5. LA NEURONA ARTIFICIAL

Un circuito eléctrico que realice la suma ponderada de las diferentes señales que recibe de otras unidades iguales y produzca en la salida un uno o un cero según el resultado de la suma con relación al umbral o nivel de disparo, conforma una buena representación de lo que es una neurona artificial. La función de transferencia para la activación o

disparo de la neurona puede ser de umbral lógico, limitación dura o función tipo S. (Ver Figura 2.5). W representa el peso o ponderación de la conexión a través de una entrada.

La neurona artificial es un dispositivo eléctrico que responde a señales eléctricas. La respuesta la produce el circuito activo o función de transferencia que forma parte del cuerpo de la neurona. Las "dendritas" llevan las señales eléctricas al cuerpo de la misma.



Umbral lógico

Limitación dura

Función tipo S

Figura 2.5

- **Función transferencia o activación de una neurona artificial**

Estas señales provienen de sensores o son salidas de neuronas vecinas. Las señales por las dendritas pueden ser voltajes positivos o negativos; los voltajes positivos contribuyen a la excitación del cuerpo y los voltajes negativos contribuyen a inhibir la respuesta de la neurona.

Esto se aprecia en la Figura 2.6.

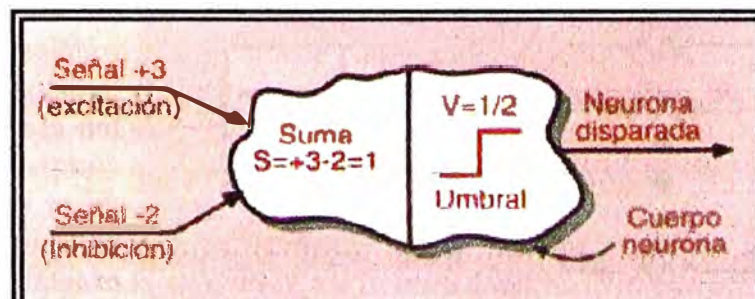


Figura 2.6. Excitación, inhibición, disparo

2.6. ENTRENAMIENTO DE REDES NEURONALES

Cuando el sistema humano de neuronas, los ojos captan un objeto A en la Figura 2.7, por ejemplo, algunos de los sensores de la visión se activan y envían señales a las neuronas ocultas. Las neuronas que se disparan con la señal de entrada aumentan el grado de conexión de ellas.

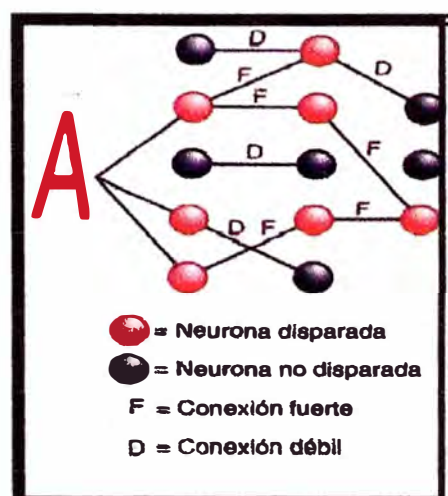


Figura 2.7. Entrenamiento para reconocer la A

Si el mismo objeto A se presenta una y otra vez, la interconexión de neuronas se refuerza y, por lo tanto, el conocimiento del objeto.

Si se le presenta a la red anterior el objeto A modificado en la Figura 2.8, la unión de las neuronas para el conocimiento de tal objeto, es débil. Las neuronas deben entrenarse para reconocer el objeto A en esta nueva presentación. Luego de algunas sesiones de entrenamiento, el sistema neuronal es capaz de reconocer el objeto A en todas sus formas. Si el objeto cambia nuevamente la red de neuronas y el conocimiento se actualizan.

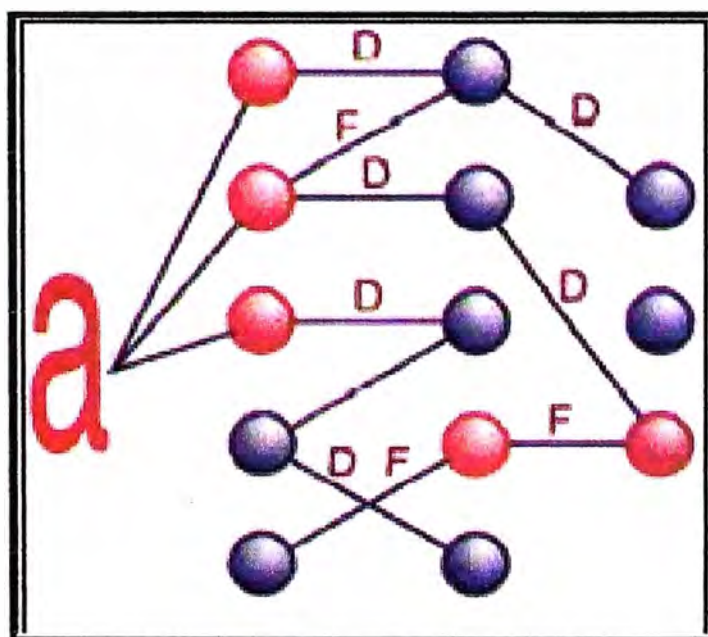


Figura 2.8. Entrenamiento para reconocer la a

Este entrenamiento, repetido para todos los valores de entrada y salida que se quiera, origina una representación interna del objeto en la red, que considera todas las irregularidades y generalidades del mismo.

En la figura 2.9 se presenta el esquema de una neurona artificial durante la etapa de aprendizaje. Una vez establecidos los pesos definitivos de interconexión, la neurona adquiere su forma tradicional.

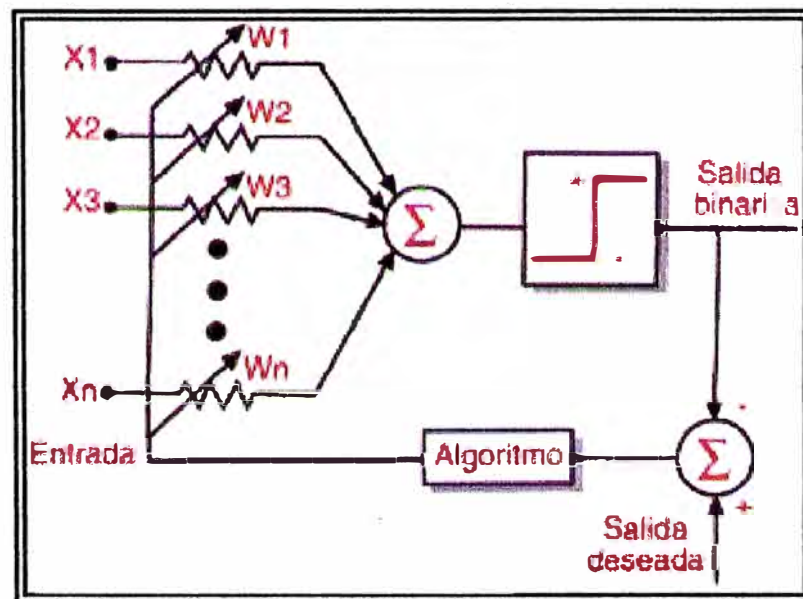


Figura 2.9. Neurona artificial con sistema de aprendizaje

En la figura 2.10 se presenta una red neuronal sin entrenar y una red entrenada.

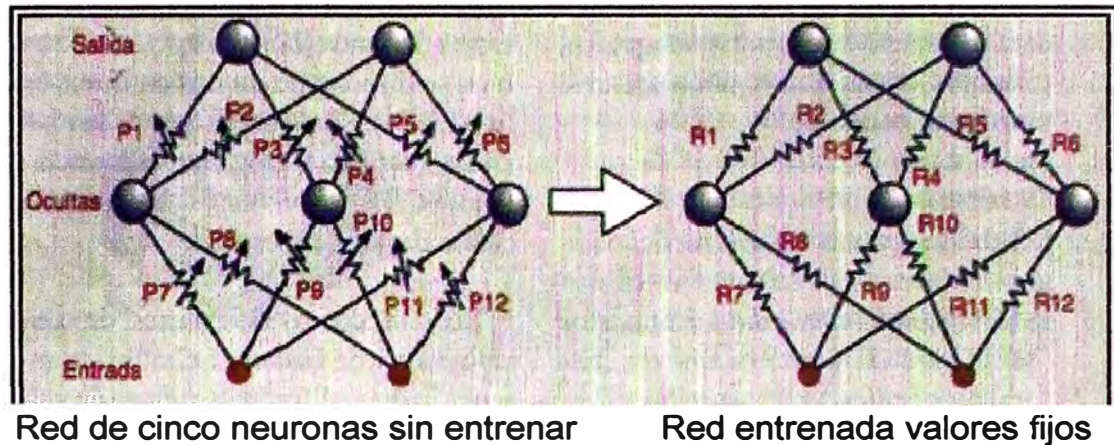


Figura 2.10

La red neuronal artificial sin entrenar con dos entradas, tres neuronas ocultas y dos salidas. El estado de no entrenamiento se representa por las resistencias variables que indican los pesos de las conexiones. Después de aplicar un algoritmo de entrenamiento y de aplicar repetidamente todas las parejas de entrada y salida necesarias, la red queda entrenada y con el peso de conexiones definido.

2.7. MÉTODO DE TRANSMISIÓN DE LA INFORMACIÓN EN EL CEREBRO

Antes conviene saber que en los primeros tiempos de la informática a los ordenadores se los llamaba calculadoras de cifras electrónicas o simplemente calculadoras digitales. Los sistemas digitales trabajan con cifras en código binario que se transmiten en formas de impulsos (bits). Los sistemas analógicos procesan señales continuamente cambiantes, como música o palabra hablada.

Por suerte para nuestro propósito de imitar con un ordenador el cerebro este también codifica la información en impulsos digitales. En los humanos las sensaciones se generan digitalmente y se transmiten así a través del sistema nervioso. Con otras palabras cuando la luz se hace más intensa, el sonido más alto o la presión más fuerte, entonces no es que fluya más corriente a través de los nervios, sino que la frecuencia de los impulsos digitales aumenta. En principio los ordenadores trabajan de manera semejante. Así una sensación más fuerte corresponde en un equipo informático a una cifra más alta (o en una palabra más larga). Sin embargo en un ordenador los datos se transmiten siempre a un mismo ritmo; la frecuencia base es inalterable. Por eso las cifras más altas tardan mas tiempo en ser transmitidas. Como por lo general el ordenador o trabajan en tiempo real, esto no tiene mayor importancia, pero cuando se trata de un procesador en tiempo real, como son los empleados en proceso

industrial, hace falta ampliar el número de canales de transmisión para que en el mismo espacio de tiempo pueda fluir mayor cantidad de datos.

Sabemos que los elementos básicos de un ordenador son las compuertas lógicas, en el cerebro también existen aunque no son idénticas a las de un ordenador. En un ordenador las compuertas And, Or etc. tiene una función perfectamente determinada e inalterable. En el cerebro también hay elementos de conexión parecidos, las llamadas sinapsis, donde confluyen en gran número las fibras nerviosas.

Cientos de datos fluyen por los nervios hasta cada sinapsis, donde son procesados. Una vez analizada y tratada la información esta sale ya transformada por los canales nerviosos en un ser vivo. Por eso cada sinapsis es simultáneamente una compuerta And, Or, Not etc.

Una sinapsis suma las tensiones de los impulsos entrantes. Cuando se sobrepasa un determinado nivel de tensión; el llamado umbral de indicación; esta se enciende, esto es deja libre el camino para que pasen los impulsos. Si el umbral de indicación de tensión es demasiado bajo, la sinapsis actúa como una puerta lógica del tipo Or, pues en tal caso pocos impulsos bastan para que tenga lugar la conexión. En cambio cuando el umbral de indicación es alto, la

sinapsis actúa como una puerta And, ya que en ese caso hace falta que lleguen la totalidad de los impulsos para que el camino quede libre. También existen conducciones nerviosas que tienen la particularidad de bloquear el paso apenas llegan los impulsos. Entonces la sinapsis hace la función de una compuerta inversora. Esto demuestra lo flexible del sistema nervioso.

2.8. DIFERENCIAS ENTRE EL CEREBRO Y UN ORDENADOR

La diferencia más importante y decisiva es el cómo se produce el almacenamiento de información en el cerebro y en el ordenador.

En el ordenador los datos se guardan en posiciones de memoria que son celdillas aisladas entre sí. Así cuando se quiere acceder a una posición de memoria se obtiene el dato de esta celdilla. Sin que las posiciones de memoria adyacentes se vean afectadas. En el Cerebro La gestión es totalmente diferente. Cuando buscamos una información no hace falta que sepamos donde se encuentra almacenada y en realidad no lo podemos saber ya que nadie sabe donde guarda hasta hoy en día el cerebro los datos. Pero tampoco es necesario ya que basta con que pensemos en el contenido o significado de la información para que un mecanismo, cuyo funcionamiento nadie conoce, nos proporcione automáticamente no solo la información deseada sino que también las informaciones vecinas, es decir, datos que de una u otra manera hacen referencia a lo buscado.

Los expertos han concebido una serie de tecnicismos para que lo incomprensible resulte algo más comprensible. Así a nuestro sistema para almacenar información se lo llama memoria asociativa. Esta expresión quiere dar a entender que los humanos no memorizan los datos diseccionándolos en celdillas, sino por asociación de ideas; esto es, interrelacionando contenidos, significados, modelos. En todo el mundo pero sobre todo en Estados Unidos y Japón, científicos expertos tratan de dar con la clave de la memoria asociativa. Si se consiguiera construir un chip de memoria según el modelo humano, la ciencia daría un paso gigante en la fascinante carrera hacia la inteligencia artificial. Y además el bagaje del saber humano quedaría automáticamente enriquecido.

Según todos los indicios el cerebro dispone de dos mecanismos de almacenamiento de datos: la memoria intermedia acepta de cinco a diez unidades de información, aunque solo las mantiene durante algunos minutos. La memoria definitiva guarda las informaciones para toda la vida, lo que no significa que nos podamos acordar siempre de todo. La memoria inmediata trabaja como una espac de cinta continua: la información circula rotativamente en forma de impulsos eléctricos por los registros. El sistema es comparable a la memoria dinámica de un ordenador, en la que la información tiene que ser refrescada continuamente para que no se pierda. En cambio, la memoria definitiva parece asemejarse mas bien a las conocidas

memorias de celdillas de los ordenadores. Se cree que esta memoria funciona gracias a formaciones químicas de las proteínas presentes en el cerebro humano.

Las principales diferencias entre el cerebro y una computadora se pueden apreciar en la Tabla N° 2.1.

Cerebro	Computadora
Sistema de datos de múltiple propósito capaz de tratar gran cantidad de información en poco tiempo pero no necesariamente con exactitud.	El Sistema está altamente especializados con capacidad para procesar información muy concreta, siguiendo unas instrucciones dadas.
La frecuencia de los impulsos nerviosos puede variar.	La frecuencia de transmisión es inalterable y esta dada por el reloj interno de la maquina.
Las llamadas sinapsis cumple en el cerebro la función simultánea de varias compuertas (and, or, not etc.)	Las compuertas lógicas tienen una función perfectamente determinada e inalterable.
La memoria es del tipo asociativo y no se sabe dónde quedara almacenada.	La información se guarda en posiciones de memoria de acceso directo por su dirección.
Los impulsos fluyen a 30 metros por segundo.	En el interior de una computadora los impulsos fluyen a la velocidad de la luz.

Tabla N° 2.1. Diferencias entre el cerebro y una computadora.

- **Similitudes entre el cerebro y una computadora**

Ambos codifican la información en impulsos digitales. Tanto el cerebro como la computadora tienen compuertas lógicas. Existen distintos tipos de memoria. Los dos tienen aproximadamente el mismo consumo de energía.

- **Control de Ordenadores por Señales Neurales**

La búsqueda de controlar el ordenador mediante impulsos bio-eléctricos, es una idea que ha dejado de ser parte de las novelas de ciencia-ficción, para integrarse a las filas de temas de estudio e investigación serios. Hoy en día, estamos un paso más cerca de lograr tales sueños, puesto que la tecnología y la teoría científica, al respecto, son cada vez más cercanos.

Una computadora puede ser controlada por impulsos electromiográficos (EMG), también puede ser controlado por impulsos Electroculográficos (EOG).

Lógica difusa o Lógica fuzzy, en informática, forma de lógica utilizada en algunos sistemas expertos y en otras aplicaciones de inteligencia artificial, en la que las variables pueden tener varios niveles de verdad o falsedad representados por rangos de valores entre el 1 (verdadero) y el 0 (falso). Con la lógica fuzzy, el resultado de una operación se puede expresar como una probabilidad y no

necesariamente como una certeza. Por ejemplo, además de los valores verdadero o falso, un resultado puede adoptar valores tales como probablemente verdadero, posiblemente verdadero, posiblemente falso y probablemente falso.

- **El caso EEG (electroencefalográfico)**

Los neurólogos creen que el origen de las tensiones EEG está en las *células piramidales* de la corteza cerebral. Cada célula constituye un diminuto dipolo eléctrico, cuya polaridad depende de que la entrada de la red a la célula sea una inhibición o excitación. Los electrodos son altamente sensibles, de modo que, puede colocarse los electrodos sobre la piel y por encima del área que se desea estudiar.

Se ha intentado aislar ciertas señales que el sujeto pueda controlar a su voluntad. Por desgracia, las señales captadas se resisten al control. Lo que se hace, es medir continuamente una diversidad de señales EEG y eliminar por filtrado las partes indeseadas. Las diferentes ondas, se caracterizan por la frecuencia de sus emisiones, hay cinco tipos: ALFA, se crean por acciones sencillas; BETA, se las asocia a un estado de alerta; TETHA, se originan por tensión emocional, como la frustración; MU, al parecer asociadas con la corteza motora (disminuyen con el movimiento o la intención de moverse).

CAPITULO III

REDES NEURONALES

3.1. FUNDAMENTOS DE REDES NEURONALES

En las redes neuronales biológicas, las células neuronales (neuronas) corresponden a los elementos de proceso anteriores. Las interconexiones se realizan por medio de las ramas de salida (axones) que producen un número variable de conexiones (sinapsis) con otras neuronas. Es decir son sistemas de simples elementos de proceso (neuronas) muy interconectados. Y su compleja operación es el resultado de abundantes lazos de realimentación junto con no linealidades de los elementos de proceso y cambios adaptivos de sus parámetros, que pueden definir incluso fenómenos dinámicos muy complicados.

Además esta enorme red es programada aún durante su funcionamiento. Programar una red de este tipo significa dos cosas: Las estructuras de interconexión entre células son alteradas; y las fuerzas de estas interconexiones son cambiadas. Parece que existen bastantes estrategias claras de cómo cambiar las fuerzas en la dirección correcta, mientras que cambios en las interconexiones son

mas difíciles de definir, porque suelen tener efectos radicales en el comportamiento de la red, especialmente en lo concerniente a la operación secuencial y las funciones jerárquicas.

Las redes neuronales artificiales (ANS) surgen de tomar algunas ideas de la neurobiología y así construir estructuras idealizadas que tratan de emular el funcionamiento de las neuronas biológicas. Por lo tanto una definición que podríamos dar de los modelos ANS es que son un conjunto de simples elementos de procesamiento (PE's) muy interconectados entre sí, cuyos pesos de sus interconexiones cambian durante el entrenamiento hasta alcanzar valores que se ajusten a la respuesta esperada del problema que planteemos.

3.2. VENTAJAS DE LOS ANS

Debido a su constitución y a sus fundamentos, los ANS presentan un gran número de características semejantes a las del cerebro. Una de ellas es que son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante, etc. Esto hace que ofrezcan numerosas ventajas y que este tipo de tecnología se esté aplicando en múltiples áreas. Estas ventajas incluyen:

- **Aprendizaje Adaptivo;** capacidad de aprender a realizar tareas basadas en un entrenamiento o una experiencia inicial.

- **Autoorganización;** una red neuronal puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje.

- **Tolerancia a Fallos;** la destrucción parcial de una red conduce a una degradación de su estructura; sin embargo, algunas capacidades de la red se pueden retener, incluso sufriendo un gran daño.

- **Operación en tiempo real;** los computadores neuronales pueden ser realizados en paralelo, y se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad.

- **Fácil inserción de la tecnología existente;** se pueden obtener chips especializados para redes neuronales que mejoran la capacidad en ciertas tareas. Ello facilitará la integración modular en los sistemas existentes.

3.3. FUNCIONAMIENTO

Como ya dijimos las redes neuronales son un conjunto de elementos de procesamiento (PE) con múltiples interconexiones entre sí. Entonces ahora vamos a ver a estas unidades de procesamiento. Cada PE está numerado, siendo el j -ésimo el que aparece en la figura 3.1.

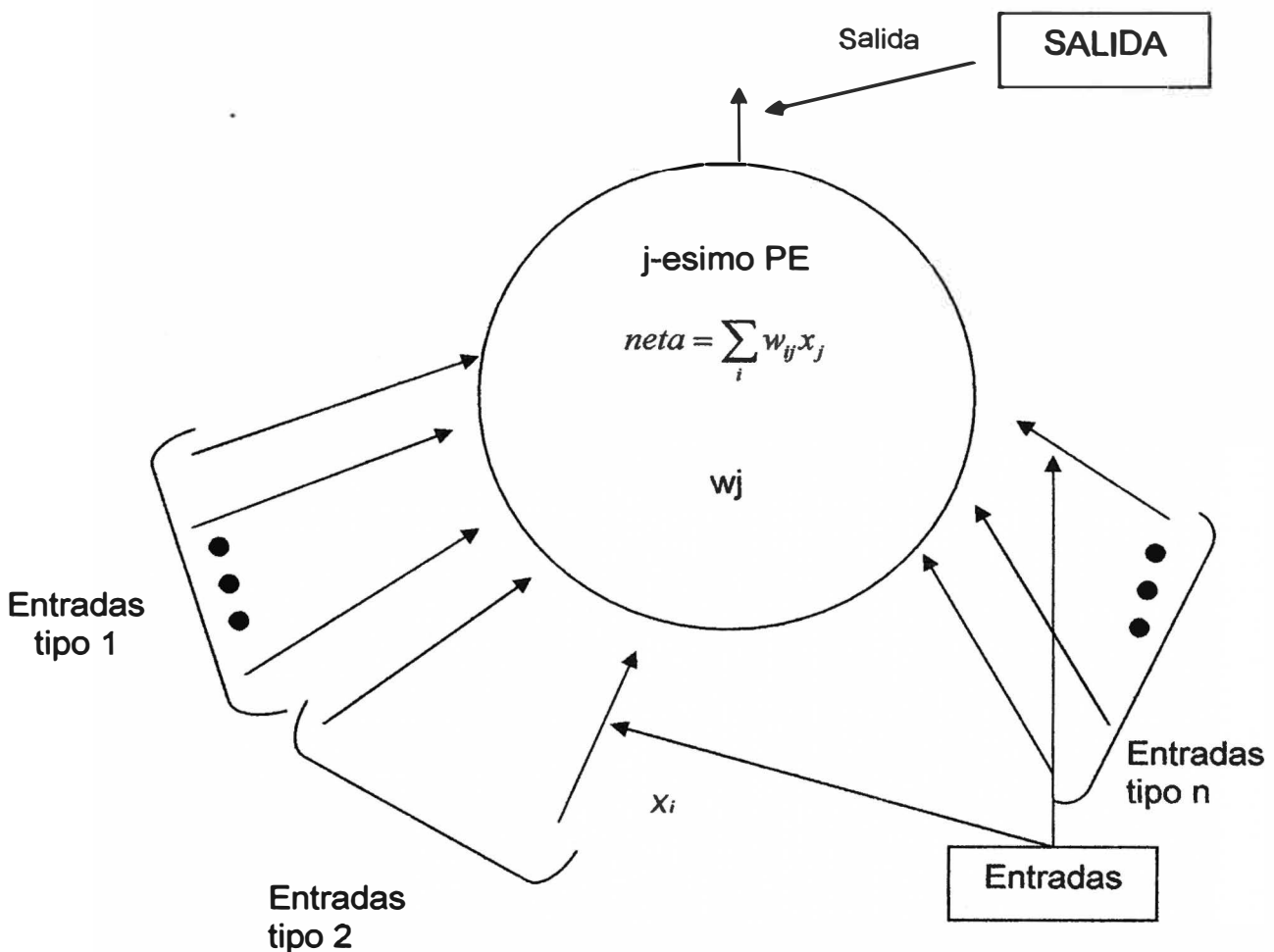


Figura 3.1. Esquema de una Neurona

Al igual que una neurona verdadera, el Pe tiene muchas entradas pero solo una salida, que se puede aplicar a muchos otros PE's de la red. La entrada que recibe el j-esimo PE procedente del i-esimo PE se indica en la forma x_i . Cada conexión con el j-esimo PE tiene asociada a él una magnitud llamada peso o intensidad de conexión. El peso de la conexión procedente del i-ésimo PE y que llega al j-ésimo PE se denota por W_{ji} . Cada PE determina un valor de entrada neto basándose en todas las conexiones de entrada. En ausencia de conexiones especiales, lo típico es calcular el valor de entrada neto sumando los valores de entrada, ponderados mediante sus pesos correspondientes.

$$\text{neta}_j = \sum_i x_i w_{ji}$$

Luego de calcular la entrada neta se determina la salida x_j

$$X_j = f(\text{neta}_j)$$

Donde f es la función de activación. Las más comunes son la función escalón, rampa o sigmoideal y Gauseano.

Los pesos w_{ji} deben modificarse durante el proceso de aprendizaje hasta alcanzar los valores que codifican el conocimiento que deseamos que aprenda el sistema.

La unión de varios de estos PE's y sus interconexiones da lugar a un modelo ANS que puede estar formado por:

- Una sola capa de PE's
- Dos capas de PE's; una de entrada y una de salida.
- Tres capas de PE's; una capa de entrada, una capa oculta y una capa de salida.
- O mas capas de PE's; una capa de entrada, varias capas ocultas y una de salida. Todo lo anterior se puede apreciar en la Figura 3.2.

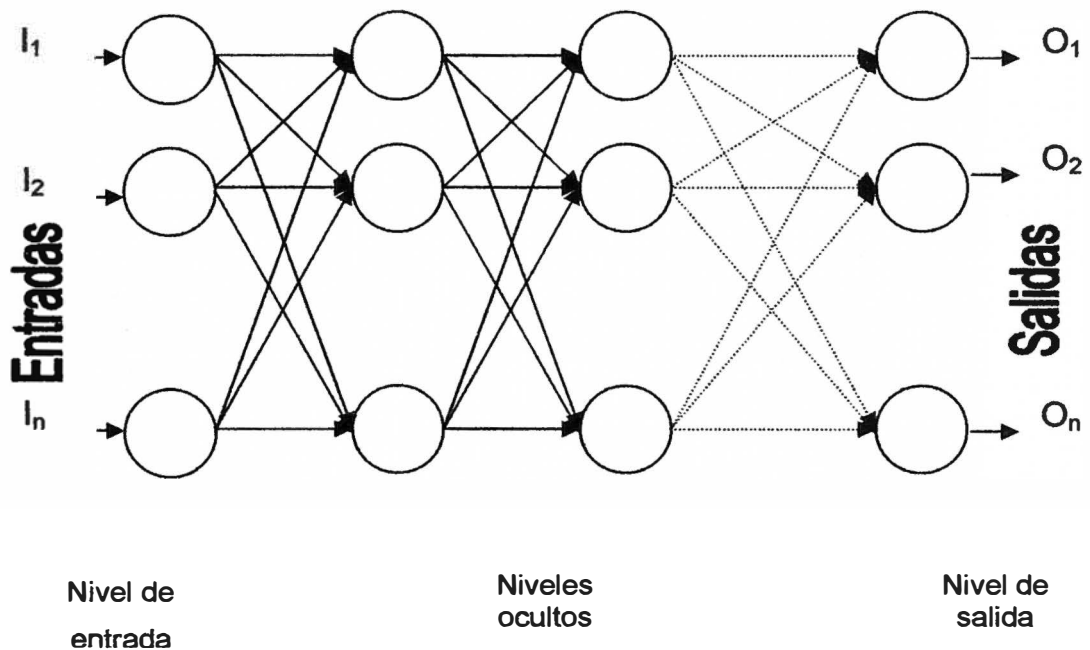


Figura 3.2. Una red neuronal con sus capas

CAPITULO IV

TIPOS DE REDES NEURONALES MÁS IMPORTANTES

4.1. RED NEURONAL MULTICAPA

Uno de los objetivos buscados en las redes neuronales es la minimización del error cuadrático medio (MSE) entre la salida actual de la red y la salida enseñada. Esta minimización se lleva a cabo mediante un procedimiento de gradiente descendente donde se requiere el cálculo del gradiente del MSE respecto a los pesos de los enlaces.

Backpropagation es un método eficiente de calcular ese gradiente. En la red neuronal se ha de llegar a un compromiso en el número de conexiones o pesos que la conforma. Si el número de pesos se mantiene demasiado bajo, la red no llega a ser capaz de aprender el conjunto de entrenamiento; por otra parte, si ese número de pesos es excesivo, los parámetros pueden causar sobreentrenamiento. Cada una de las subredes entre la capa de entrada y la capa oculta es inicializada con pesos aleatorios y entrenada con diferentes mapas de características, respectivamente.

Todas las conexiones en la red son de tipo adaptativo, y son entrenadas con el algoritmo de retropropagación o "Backpropagation". El coeficiente de aprendizaje que es el único parámetro libre en el algoritmo de entrenamiento, es fijado antes del entrenamiento y no se varía posteriormente.

La arquitectura más común en las aplicaciones comerciales es una RN Multi-capa supervisada, llamada PERCEPTRON. Mediante un algoritmo llamado Retro-Propagación (Back Propagation) se minimiza el error cuadrático medio entre el valor real del output y el valor deseado, a través de un proceso iterativo. Además existen las redes de Kohonen, este modelo es apropiado solo para los problemas de conglomerados. Se trata de un modelo de RN con capacidad para formar mapas de características de manera similar a como ocurre en el cerebro. EL objetivo de Kohonen era demostrar que en un estímulo externo (información de entrada) por sí solo, suponiendo una estructura propia y una descripción funcional del comportamiento de la red, era suficiente para forzar la formación de mapas. La función de base radial es un método desarrollado más recientemente que se puede utilizar para las mismas tareas que el modelo multicapa, pero que trabaja de una manera diferente.

- **Nodos**

Un nodo es una unidad de procesamiento autocontenido que actúa en paralelo con otros nodos.

4.2. RESUMEN DE LAS REDES NEURONALES

El nombre de las redes neuronales y la persona que lo desarrolló se muestra en la Tabla N° 4.1.

NOMBRE DE LA RED	APLICACIONES MÁS IMPORTANTES	INVENTADA / DESARROLLADA POR
Avalancha	<ul style="list-style-type: none"> - Reconocimiento de habla continua - Control de Brazos robot. 	Stephen Grossberg
Teoría resonancia adaptiva (ART)	<ul style="list-style-type: none"> - Reconocimiento de patrones (radar, sonar etc.) 	Gail Carpenter / Stephen Grossberg
ADELINE / MADALINE	<ul style="list-style-type: none"> - Filtrado de señales - Ecuador adaptivo - Modems 	Bernard Widrow
Backpropagation	<ul style="list-style-type: none"> - Síntesis de voz desde texto - Control de robots - Predicción - Reconocimiento de patrones 	Paul Werbos / David Parker / David Rumelhart
Memoria Asociativa Bidireccional	<ul style="list-style-type: none"> - Memoria heteroasociativa de acceso por contenido 	Bart Kosko
Hopfield	<ul style="list-style-type: none"> - Reconstrucción de patrones y optimización 	John Hopfield
Máquinas de Boltzman y Cauchy	<ul style="list-style-type: none"> - Reconocimiento de patrones - Optimización 	Jeffrey Hinton / Terry Sejnowski Harold Sizu
Neocognitron	<ul style="list-style-type: none"> - Reconocimiento de caracteres manuscritos 	K Fukushima
Perceptrón	<ul style="list-style-type: none"> - Reconocimiento de caracteres impresos 	Frank Rosenblatt
Self Organizing-Map Topology-Preserving-Map	<ul style="list-style-type: none"> - Reconocimiento de patrones - Codificación de datos - Optimización de datos 	Teuvo Kohonen

Tabla N° 4.1. Tipos De Redes Neuronales

4.3. APRENDIZAJE DE LAS REDES NEURONALES

La característica de una RN, que verdaderamente la aparte del resto de las técnicas, es su capacidad de aprender, basándose generalmente en sus errores.

El aprendizaje consiste en la presentación de patrones a la red, y la subsiguiente modificación de los pesos de las conexiones siguiendo un algoritmo de aprendizaje que trata de optimizar su respuesta, generalmente mediante la minimización del error o la optimización de alguna función de energía.

- **Aprendizaje Supervisado;** el modo de aprendizaje más sencillo consiste en la presentación de patrones de entrada junto a los patrones de salida deseados (targets) para cada patrón de entrada, por eso se llama aprendizaje supervisado.
- **Aprendizaje No Supervisado;** si no se le presentan a la red los patrones de salida deseados, diremos que se trata de aprendizaje no supervisado, ya que no se le indica a la red que resultados debe dar, sino que se le deja seguir alguna regla de auto-organización.

- **Aprendizaje Reforzado;** un tercer tipo de aprendizaje, a medio camino entre los anteriores, es el llamado aprendizaje reforzado, en este caso el supervisor se limita a indicar si la salida ofrecida por la red es correcta o incorrecta, pero no indica que respuesta debe dar.

Cualquiera que sea el tipo de aprendizaje usado, una característica esencial de la red es la regla de aprendizaje usada, que indica como se modifican los pesos de las conexiones en función de los datos usados en la entrada, es decir, de la historia de aprendizaje de la red. Por ejemplo, entre los algoritmos de aprendizaje supervisado, la regla delta generalizada, modifica los pesos realizando en cada ciclo de aprendizaje un incremento los pesos proporcionales a la tasa de variación del error respecto al peso, en sentido negativo.

El aprendizaje requiere la presentación repetida de un número relativamente amplio de ejemplos de entrenamiento, hasta lograr una respuesta satisfactoria del sistema (según la medida de error o función de energía utilizada).

Es usual disponer de un conjunto de datos distintos a los usados para el entrenamiento, para los cuales se conoce la respuesta correcta, y se usan como test, evaluando con ello si la red

responde adecuadamente frente a datos distintos a los usados durante el aprendizaje (generalización), si es así se considera que la red funciona bien y se puede pasar a la fase de aplicación, es decir, se puede usar para obtener un resultado frente a datos totalmente nuevos, para los que no se conoce la respuesta correcta.

Aunque la capacidad de aprendizaje es esencial y casi definitoria de lo que sería una RNA, se han desarrollado modelos que no requieren modificar los pesos de las conexiones, sino que son precalculados y pre-establecidos antes de presentarle patrones a la red, como en las redes de Hamming y en redes de Hopfield, aún cuando en estos casos también son necesarios procesos iterativos para la resolución de las dificultades encontradas.

Otro criterio que se puede utilizar para diferenciar las reglas de aprendizaje se basa en considerar si la red puede aprender durante su funcionamiento habitual o si el aprendizaje supone la desconexión de la red. En el primer caso se trataría de un aprendizaje ON LINE, mientras que el segundo es lo que se conoce como aprendizaje OFF LINE.

Cuando el aprendizaje es OFF LINE se distingue entre una fase de aprendizaje o entrenamiento, una fase de testeo y una fase de

operación o funcionamiento, existiendo un conjunto de datos de entrenamiento y un conjunto de datos de prueba. En estas redes los pesos de las conexiones permanecen fijos después que termina la etapa de entrenamiento.

Como ya se mencionó, un circuito eléctrico que realice la suma ponderada de las diferentes señales que recibe de otras unidades iguales y produzca en la salida un uno o un cero (según el resultado de la suma con relación al umbral o nivel de disparo) conforma una buena representación de lo que es una neurona artificial. La función de transferencia para la activación (disparo) de la neurona puede ser de umbral lógico, de limitación dura o de función de tipos. W representa el peso o ponderación de la conexión a través de una entrada.

4.4. EL PERCEPTRON

Las características de la red Perceptron se muestran en la Tabla N° 4.2.

Modelo de Red	Topología	Aprendizaje			Asociación Auto/Hetero	Informac. De Entrada y Salida	Autor
		ON-OFF LINE	supervisado y no supervisado	Regla			
Perceptron	2 capas Feedforward	OFF	Supervisado	Corrección Error	Heterasociativa	E: Analogica S: Binaria	Roseblatt 1958

Tabla N° 4.2. Topología y Aprendizaje del Perceptron

Desarrollado por Roseblatt en 1958. Tiene la capacidad para aprender a reconocer patrones sencillos: está formado por varias neuronas lineales para recibir las entradas a la red y una neurona de salida, es capaz de decidir cuándo una entrada presentada a la red pertenece a una de las dos clases que es capaz de reconocer.

Su única neurona de salida realiza la suma ponderada de las entradas, resta el umbral y pasa el resultado a una función de transferencia de tipo escalón. Responde + 1 si el patrón presentado pertenece a una clase A, o -1 si pertenece a la clase B. La salida depende de la entrada neta y del valor umbral.

Sin embargo, al constar solo de una capa de entrada y otra de salida con una única neurona, tiene una capacidad de representación bastante limitada. Solo es capaz de discriminar patrones muy sencillos, linealmente separables. Estos detalles de la red Perceptron se muestran en la Figura 4.1.

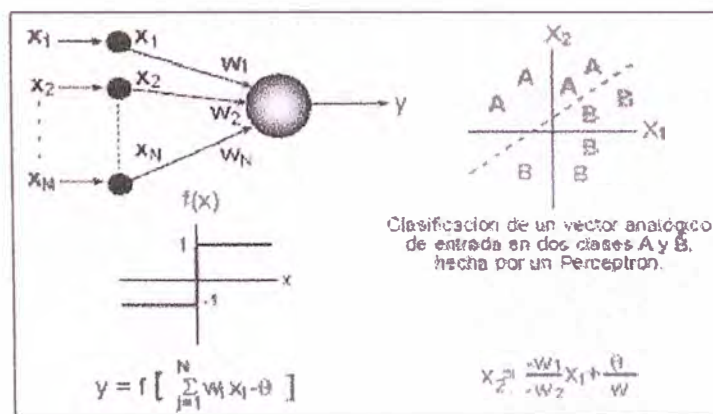


Figura 4.1 El Perceptron

4.5. REGLA DE APRENDIZAJE DEL PERCEPTRON

Es de tipo supervisado, lo que requiere que sus resultados sean evaluados y se realicen las oportunas modificaciones del sistema si se necesita.

Solo se pueden aprender clasificaciones fáciles, debido a que un Perceptron usa un separador lineal como célula de decisión, con lo que solo se puede realizar una sola separación lineal (por medio de un hiperplano).

Los pasos de la regla de aprendizaje del Perceptron son los siguientes:

1. Iniciación de los pesos y del umbral.

Se asignan valores aleatorios a los pesos de las conexiones y al umbral.

2. Presentación de un nuevo par (Entrada, Salida esperada). Se pasa la entrada y la salida deseada.

3. Cálculo de la salida actual. Cálculo de la función de activación.

4. Adaptación de los pesos.

5. Volver al paso 2.

4.6. EL PERCEPTRON MULTINIVEL

Es una red de tipo feedforward compuesta por varias capas de neuronas entre la entrada y la salida de la misma. Permite establecer regiones de decisión mucho más complejas que las de dos semiplanos, como el Perceptron de un solo nivel, que sólo puede establecer dos regiones separadas por una frontera lineal en el espacio de patrones de entrada.

Un Perceptron con tres niveles de neuronas puede formar cualquier región convexa (o cerrada) en ese espacio. Las regiones convexas se forman mediante la intersección entre las regiones formadas por cada neurona de la segunda capa. Cada elemento se comporta como un Perceptron simple.

El esquema del Perceptrón multinivel está en la Figura 4.3.

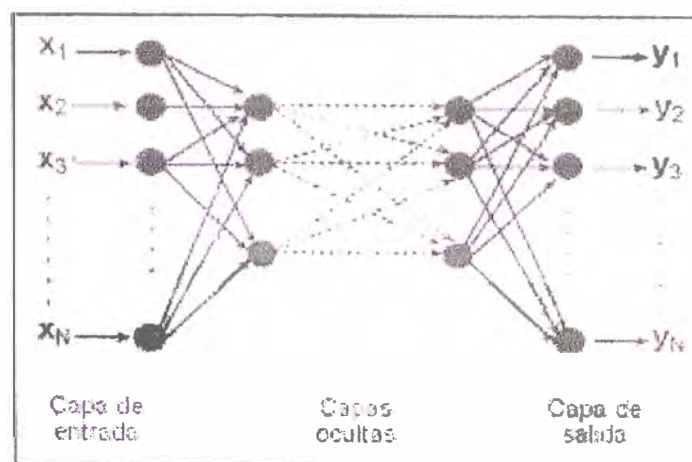


Figura 4.3. El Perceptron Multinivel

Un Perceptron de cuatro capas puede formar regiones de decisión arbitrariamente complejas. El proceso de separación en clases que se lleva a cabo consiste en la partición de la región deseada en pequeños hipercubos (cuadrados para dos entradas de la red).

4.7. RED BACKPROPAGATION

El algoritmo de backpropagation se caracteriza, principalmente, por la representación interna del conocimiento que es capaz de organizar en la capa intermedia de las células, para conseguir cualquier correspondencia entre la entrada y la salida de la red.

El funcionamiento de una red backpropagation consiste en un aprendizaje de un conjunto predefinido de pares de entradas-salidas dados como ejemplo, empleando un ciclo propagación adaptación de dos fases:

- Primero se aplica un patrón de entrada como estímulo para la primera capa de las neuronas de la red, se va propagando a través de todas las capas superiores hasta generar una salida, se compara el resultado obtenido en las neuronas de salida con la salida que se desea obtener y se calcula un valor del error para cada neurona de salida.

- Luego, se transmiten estos errores hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de la capa intermedia que contribuyan directamente a la salida, recibiendo el porcentaje de error aproximado a la participación de la neurona intermedia en la salida original. Repitiéndose este proceso capa por capa, hasta que todas las neuronas de la red hayan recibido un error que describa su aportación relativa al error total. Basándose en el valor del error recibido, se reajustan los pesos de conexión de cada neurona, de manera que en la siguiente vez que se presente el mismo patrón, la salida se encuentre más cercana a la deseada; es decir, el error disminuye.

- **La Regla Delta Generalizada**

Fue propuesta por Widrow en 1960 se ha extendido a redes con capas intermedias (regla delta generalizada) con conexiones hacia delante (feedforward) y cuyas células tienen funciones de activación continuas (lineales o sigmoideas), dando lugar al algoritmo de retropropagación (backpropagation). Estas funciones continuas son no decrecientes y derivables. Una función de este tipo es la sigmoidea, a diferencia de la función escalón utilizada en el Perceptron, ya que la última no es derivable en el punto donde se encuentra la discontinuidad.

Este algoritmo utiliza también una función o superficie de error asociada a la red, en la que se va buscando el estado estable de mínimo error a través del camino descendiente de la superficie del error. Por esa razón, el error del sistema es realimentado para realizar la modificación de los pesos, en un valor proporcional al gradiente decreciente de dicha función de error.

La red propagation tiene la capacidad de autoadaptar los pesos de las neuronas de las capas intermedias para aprender la relación que existe entre un conjunto de patrones dados como ejemplo y sus salidas correspondientes.

Esta característica, es llamada capacidad de *generalización*, que se entiende como la facilidad de dar salidas satisfactorias a entradas nunca antes vistas por el sistema en su fase de entrenamiento.

4.8. APLICACIONES DE LA RED BACKPROPAGATION

Las Redes Backpropagation son unas de las más aplicadas en los últimos tiempos. Pueden ser aplicadas a una amplia variedad de problemas, por lo que quedaría un poco difícil describir o citar a todas sus aplicaciones. Esto se da a causa de la gran versatilidad de este tipo de red, debido a la naturaleza de su proceso de aprendizaje, que solamente necesita de dos ecuaciones para propagar las señales de

error hacia atrás. Se elige entre una u otra ecuación dependiendo si la unidad de proceso es o no de salida.

A continuación citamos algunas de las aplicaciones de las Redes Backpropagation:

- Codificación de la Información.
- Traducción de texto en lenguaje hablado.
- Compresión de Datos.
- Inspección de calidad de pintura.
- Reconocimiento de lenguaje hablado.
- Reconocimiento óptico de caracteres (OCR).
- En cardiología.
- Clasificación de señales electrocardiográficas (ECG).
- Detección de taquicardias ventriculares y supraventriculares.
- Reconocimiento de formas anormales en señales ECG (electrocardiográficas).
- Cancelación de ruido en señales ECG.

4.9. RED DE KOHONEN

Existen evidencias que demuestran que en el cerebro hay neuronas que se organizan en muchas zonas, de forma que las informaciones captadas del entorno a través de los órganos sensoriales se representan internamente en forma de mapas bidimensionales.

Aunque en gran medida esta organización neuronal está predeterminada genéticamente, es probable que parte de ella se origine mediante el aprendizaje, esto sugiere que el cerebro podría poseer la capacidad inherente de formar mapas topológicos de las informaciones recibidas del exterior, de hecho esta teoría podría explicar su poder de operar con elementos lingüísticos, algunas áreas del cerebro simplemente podrían crear y ordenar neuronas especializadas o grupos con características de alto nivel y sus combinaciones, en definitiva se construirían mapas especiales para atributos y características.

A partir de estas ideas que tuvo Kohonen presenta un sistema con un comportamiento semejante, se trata de un modelo de red neuronal con capacidad para formar mapas de características de manera similar a como ocurre en el cerebro; el objetivo de Kohonen era demostrar que un estímulo externo (información de entrada) por sí solo, suponiendo una estructura propia y una descripción funcional del comportamiento de la red, era suficiente para forzar la formación de los mapas.

Este modelo pertenece a la categoría de las redes competitivas o mapas de autoorganización, es decir, con aprendizaje no supervisado de tipo competitivo. Poseen una arquitectura de dos capas (entrada-salida) (una sola capa de conexiones), funciones de activación

lineales y flujo de información unidireccional (son redes en cascada). Las unidades de entrada reciben datos continuos normalizados, se normalizan así mismo los pesos de las conexiones con la capa de salida. Tras el aprendizaje de la red, cada patrón de entrada activará una única unidad de salida. En algunos modelos cada entrada puede provocar la activación de un conjunto de unidades de salida, en ese caso estaríamos usando un código demográfico. El objetivo de este tipo de redes es clasificar los patrones de entrada en grupos de características similares, de manera que cada grupo activará siempre la(s) misma(s) salida(s). Cada grupo de entradas queda representado en los pesos de las conexiones de la unidad de salida triunfante. la unidad de salida ganadora para cada grupo de entradas no se conoce a priori, es necesario averiguarlo después de entrenar a la red.

Una de las cualidades de este tipo de redes es la incorporación a la regla de aprendizaje de cierto grado de sensibilidad con respecto al vecindario o historia. Esto hace que el número de neuronas que no aprenden desaparezca, aumentando así su capacidad de extraer o mapear características topológicas de los datos.

La red mapea el espacio de entrada hacia un espacio de salida con cierto orden topológico, en gráfico se muestra las salidas agrupadas por zonas, Kohonen propone un método para que este orden se conserve al entrenar la red, la clave está en reducir el tamaño del

vecindario de la unidad ganadora en cada iteración.

Este modelo tiene dos variantes denominadas LVQ (learning Vector Quantization) y TPM (Topology Preserving Map) o SOM (Self Organizing Map), ambas se basan en el principio de formación de mapas topológicos para establecer características comunes entre las informaciones (vectores) de entrada a la red, aunque difieren en las dimensiones de éstos, siendo de una sola dimensión en el caso de LVQ y bidimensional o tridimensional en la red SOM. El aprendizaje en el modelo de Kohonen es de tipo Off-line, por lo que se distingue una etapa de aprendizaje y otra de funcionamiento.

El modelo de Kohonen se usa bastante en computación neuronal a pesar de sus limitaciones en la duración del proceso de aprendizaje, los mapas de Kohonen se aplican en : Reconocimiento de patrones, reconocimiento de voz, reconocimiento de textos manuscritos, codificación de datos, comprensión de imágenes, resolución de problemas de optimización y segmentación.

La red Kohone se puede apreciar en la figura 4.4.

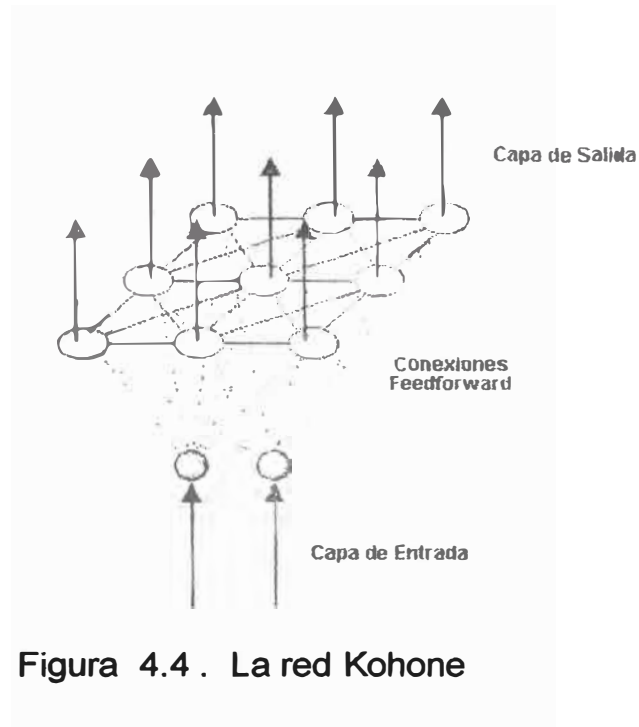


Figura 4.4 . La red Kohone

CAPITULO V

RECONOCIMIENTO DE CARACTERES UTILIZANDO EXCEL

5.1. USO DE LA RED BACKPROPAGATION EN EL RECONOCIMIENTO DE PATRONES

La red Backpropagation BPN (Propagación hacia atrás), este algoritmo de propagación hacia atrás es una regla de aprendizaje que se puede usar en modelos de redes con mas de dos capas de células. Una característica de este algoritmo es la representación interna del conocimiento que es capaz de organizar en la capa intermedia de las células para conseguir cualquier correspondencia entre la entrada y la salida de red.

Su funcionamiento es como sigue. Primero la red aprende un conjunto predefinido de entradas y salidas dados como ejemplo, empleando un ciclo propagación-adaptación de dos fases. Una vez que se ha aplicado una trama de entrada como estímulo para la primera capa de unidades de la red, esta se va propagando a través de todas las capas superiores hasta generar una salida.

La señal de salida se compara entonces con la salida deseada, y se calcula una señal de error para cada unidad de salida. Las señales de error son entonces transmitidas hacia atrás, partiendo de la capa de salida, hacia todos los nodos de la capa intermedia que contribuyen directamente a la salida. Sin embargo las capas intermedias solo reciben una fracción de la señal total de error, basándose aproximadamente en la contribución relativa que haya aportado la unidad a la salida original. Este proceso se repita capa por capa, hasta que todos los nodos de la red hayan recibido una señal de error que describa su contribución relativa al error total. Basándose en la señal de error recibida, se actualizan los pesos de conexión de cada unidad, para hacer que la red converja hacia un estado que permita codificar todas las tramas de entrenamiento.

La importancia de este proceso consiste en que, a medida que se entrena la red, los nodos de las capas intermedias se organizan así mismos de tal modo que los distintos nodos aprenden a reconocer distintas características del espacio total de entradas. Después del entrenamiento cuando se les presente una trama arbitraria que contenga ruido o este incompleta, las unidades de las capas ocultas de la red responderán con una salida activa si la nueva entrada contiene una trama que se asemeje a aquella característica que las unidades individuales hayan aprendido a reconocer durante su entrenamiento.

En esta Red utilizamos la Regla Delta generalizada como algoritmo para la convergencia de la red. La Red Backpropagation es una red formada por capas, con propagación hacia adelante, que esta completamente interconectada entre capas; por tanto no hay conexiones de realimentación ni conexiones que salten de una capa para ir a una capa anterior. En esta red la convergencia de la red es buscar un mínimo absoluto o local (esto depende de nuestra aplicación y de cuanto error estamos dispuestos a admitir) en el espacio de pesos. Ver figura 5.1 y 5.2.

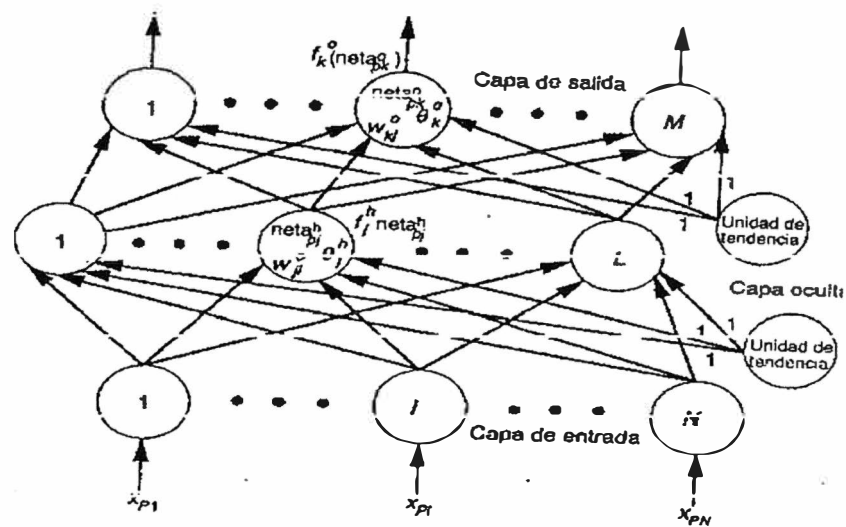


Figura 5.1. Red Backpropagation

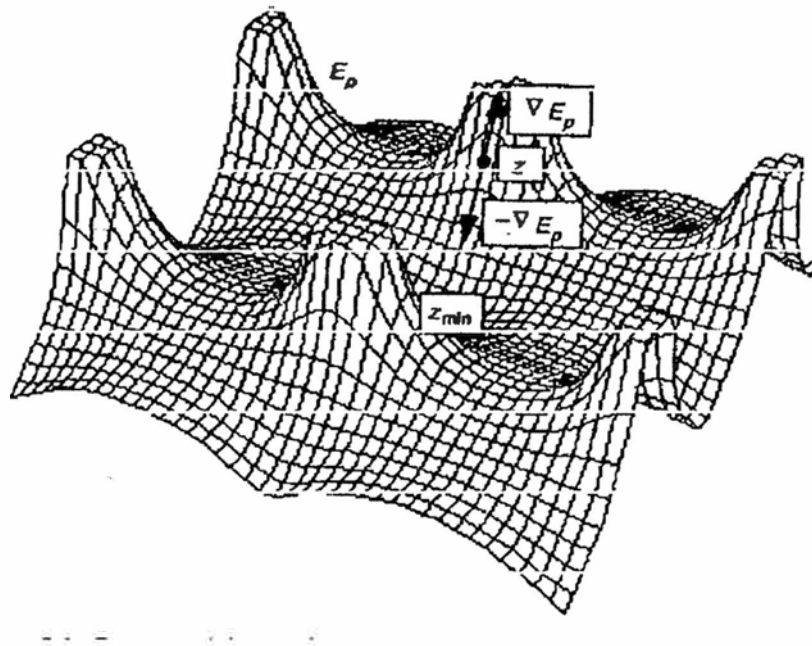


Figura 5.2 Superficie de convergencia de la BPN

Dadas las características descritas las redes BPN una aplicación típica es el reconocimiento de patrones, siendo este caso los caracteres de un texto los patrones a reconocer. Se puede utilizar esta red para aprender las diferentes formas de caracteres que se presentaran como imágenes compuestas por cierto número de píxeles, generando la red como salida el código ASCII. El objetivo es que la red tenga capacidad para generalizar; es decir, que conozca las formas de caracteres no utilizadas en la fase de aprendizaje, siempre que no difieran excesivamente de los patrones de entrenamiento.

Otra aplicación en telecomunicaciones es en la transmisión de información utilizando redes neuronales, dado que se generan errores y este sistema permite detectarlos, este sistema en diagrama de bloques se puede apreciar en la Figura 5.3.

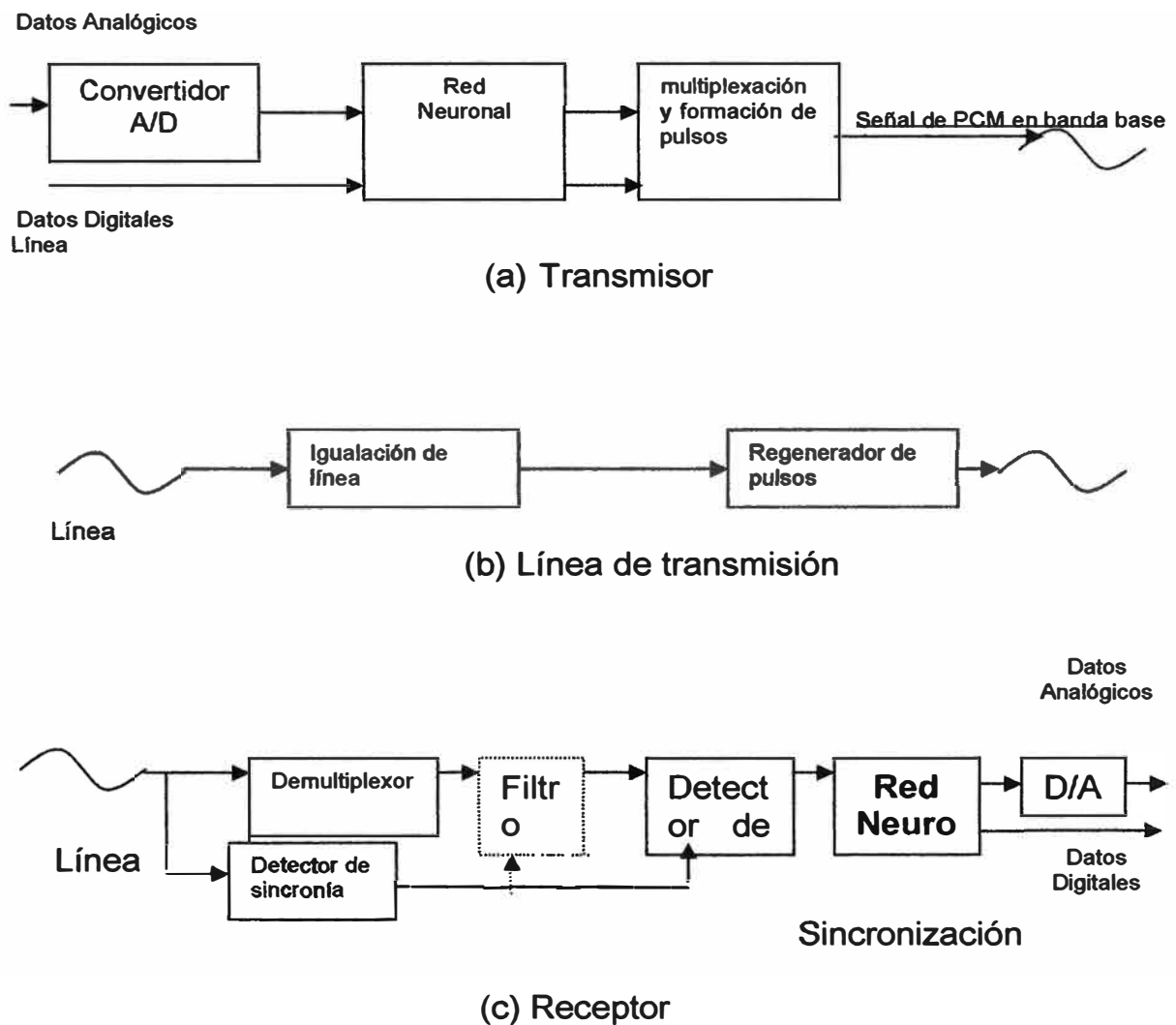


Figura 5.3. Transmisión de información con red Neuronal

5.2. DESCRIPCIÓN DEL PROYECTO

El proyecto consiste en llevar a cabo el reconocimiento de patrones de letras mayúsculas y minúsculas para su conversión posterior en el código ASCII que le corresponde. Para este proyecto se han utilizado solo las vocales tanto mayúsculas y minúsculas escritas en una matriz de 14x10 mm.

La red backpropagation es entrenada con patrones de estas letras y su correspondiente código ASCII, para luego ser probada con diferentes entradas simulando letras mal escritas y con ruido.

El código ASCII correspondiente a cada vocal es:

A → 65 → 1000001

E → 69 → 1000101

I → 73 → 1001001

O → 79 → 1001111

U → 85 → 1010101

a → 97 → 1100001

e → 101 → 1100101

i → 105 → 1101001

o → 111 → 1101111

u → 117 → 1110101

5.3. ESTRUCTURA DE LA RED

La red esta formada por neuronas agrupadas en tres capas de la siguiente manera:

En la capa de entrada existen 140 neuronas que corresponden a las 140 entradas de la red. Su función de activación es lineal $[0,1]$.

En la capa oculta existen 78 neuronas y su función de activación es sigmoideal (Tanh).

En la capa de salida existen 7 neuronas que corresponden a las 7 salidas de la red. Su función de activación es sigmoideal.

Los pesos de las conexiones entre las capas son iniciados ***todos en 0,3***.

Además los parámetros de: Learning rate = 0.1 y el de momentum = 0.1 son ajustados a estos valores para evitar una rápida y poco exacta convergencia de la red.

5.4. FORMATO DE DATOS ENTRADAS Y SALIDAS

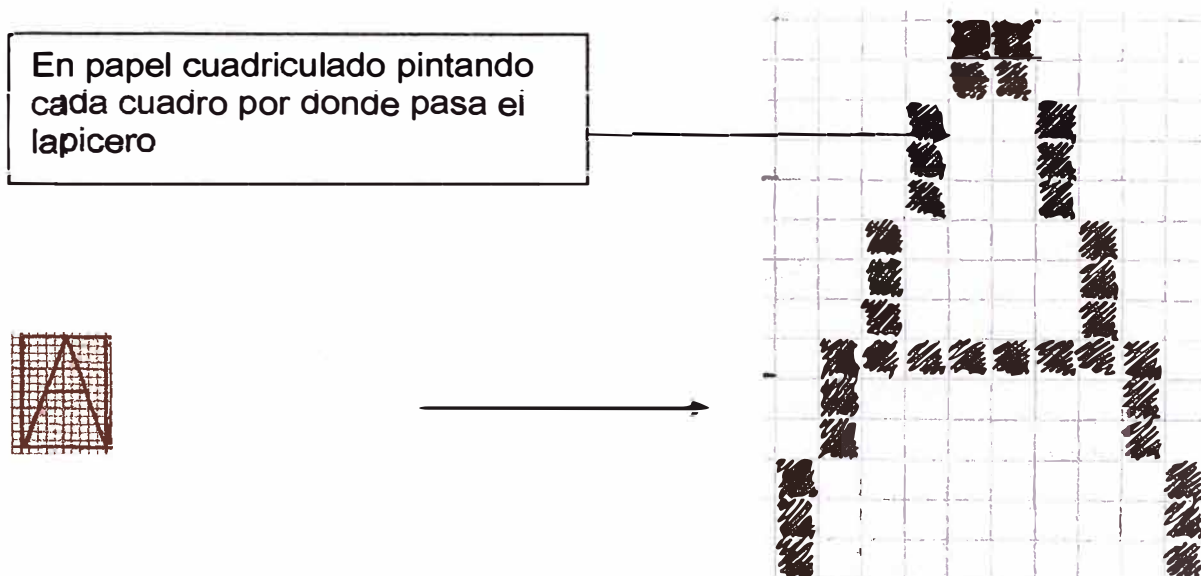
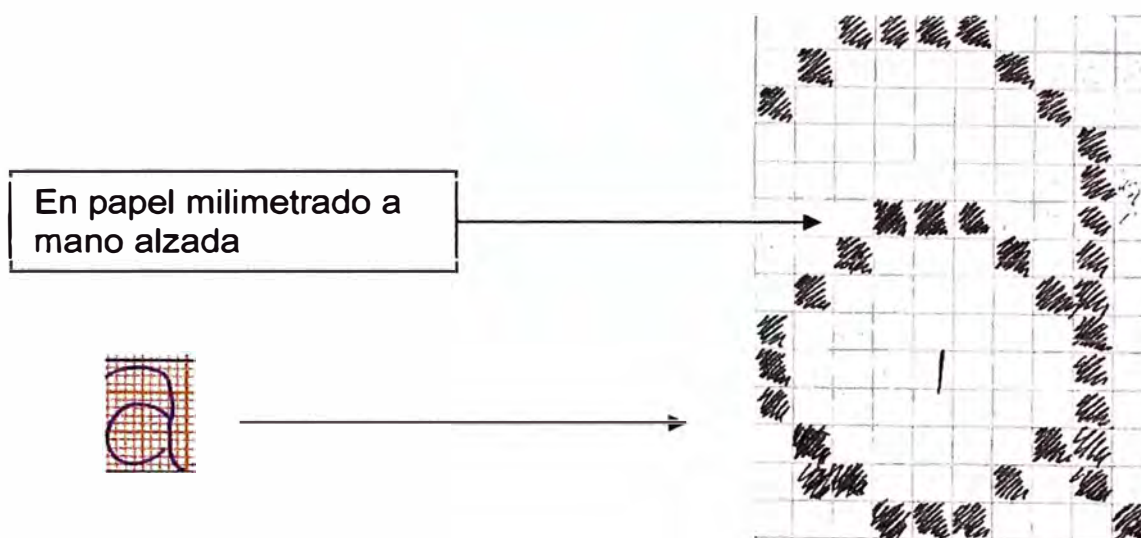
- **Para la entrada**

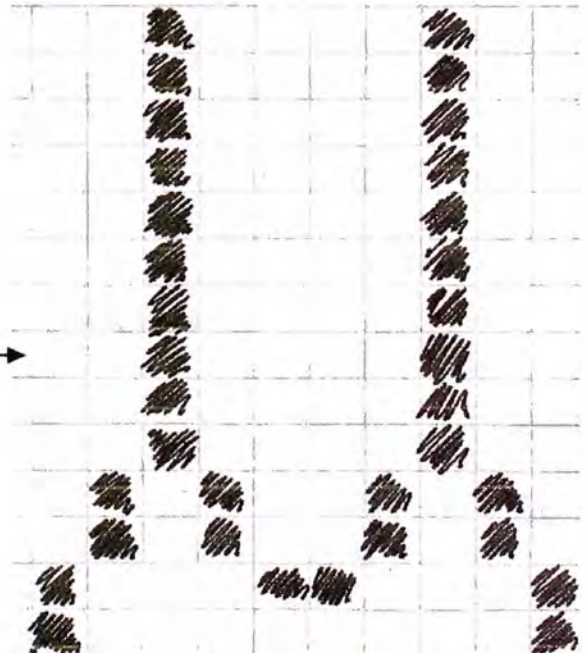
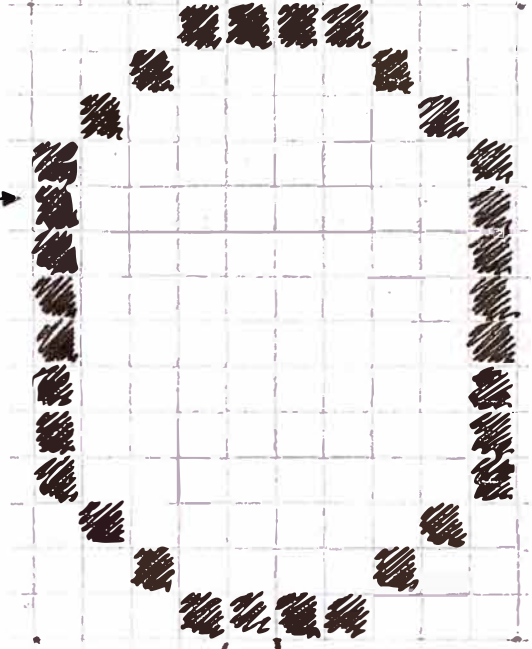
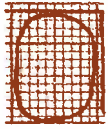
Los datos son ingresados en una matriz de 10mm x 14mm en papel milimetrado a mano alzada con un lapicero. Luego son copiados en una matriz de las mismas dimensiones en una tabla de Excel colocando en cada campo de la matriz un 1 si es que el trazo del lapicero pasa por ese campo en el papel milimetrado, y un 0 (cero) si el lapicero no pasa por ese campo. Este proceso se repite hasta agrupar la cantidad deseada de patrones que con los que se desea entrenar a la red. Cuando tenemos listos los patrones convertimos este archivo de Excel a un archivo separado por tabulaciones (ASCII) de extensión *dat*.

Hemos utilizado MATLAB para acomodar esta matriz de valores (*xxx.dat*) en vectores de 140 elementos que van a representar cada uno a una de las vocales (patrones de entrada). Luego son devueltos a Excel donde le agregamos a cada patrón su correspondiente valor ASCII de salida y otra vez convertimos este archivo a un archivo de tipo ASCII *.dat* para que sea importado al NEUROSHELL2.

Para nuestro proyecto vamos a entrenar a la red con 23 patrones de entrada entre vocales mayúsculas y minúsculas.

Aquí vemos algunos de los patrones que se han usado para el entrenamiento.





- **Para la Salida**

Los datos de salida son exportados del Neuroshell2 a un archivo Patx.txt, es decir a un archivo de texto separado por espacios. Este archivo contiene Las salidas esperadas por red, las salidas reales de la red y una comparación de entre las salidas esperadas y las reales mostrándonos el error cometido por la red al realizar la estimación.

Esta salida en formato de texto separado por espacios es leída |posteriormente con MATLAB para separar los vectores que contienen las salidas reales. Dada que la función de activación de las neuronas de salida es sigmoideal la salida no es discreta (1 y 0). Entonces esta salida después de ser importada a MATLAB pasa por una rutina que convierte todos los valores mayores a 0.5 en "1" y los menores a 0.5 en "0". De esta manera obtenemos una salida discreta con los códigos ASCII correspondientes a cada letra.

Aquí mostramos la rutina de *matlab*:

Vovec.m

```
vorecp=PAT1(:,8:14);
```

```
vocomp=PAT1(:,1:7);
```

```
for i=1:23
```

```
    for j=1:7;
```

```
        if(vorecp(i,j)<0.5);
```

```
        vorecp(i,j)=0;
    end
    if(vorecp(i,j)>0.5);
        vorecp(i,j)=1;
    end
end
end
end
vocb=vorecp;
```

Como vemos la matriz de salida conteniendo los valores binarios que representan el código ASCII de cada letra lo obtenemos en la variable **vorecp**; para ser utilizada posteriormente. Las dimensiones de esta matriz dependen de la cantidad de patrones analizados, es decir en nuestro caso es una matriz de 23x7 elementos.

5.5. PROCESAMIENTO DE DATOS

En este proyecto hemos utilizado dos tipos distintos de entrenamiento para la misma estructura de red descrita anteriormente.

La primera red PAT1.DSC fue entrenada escogiendo un a cierta cantidad de patrones (20%) del total de patrones ingresados, para formar un grupo de prueba. Este grupo de prueba nos sirve para calibrar el entrenamiento de nuestra red y así lograr un mejor

resultado. El entrenamiento duro 21 seg. Y el mínimo error promedio fue de 0.0000042.

La segunda red PAT2.DSC fue entrenada sin escoger ningún grupo de prueba y su entrenamiento fue detenido manualmente debido a que el error seguía disminuyendo. El entrenamiento duró 2 min y el mínimo error promedio fue de 0.0000014.

5.6. DESCRIPCIÓN DEL PROCEDIMIENTO UTILIZANDO EL NEUROHELL2

Primero abrimos nuestro proyecto en el modo avanzado y tendremos estas opciones que se muestra en la Figura 5.4.

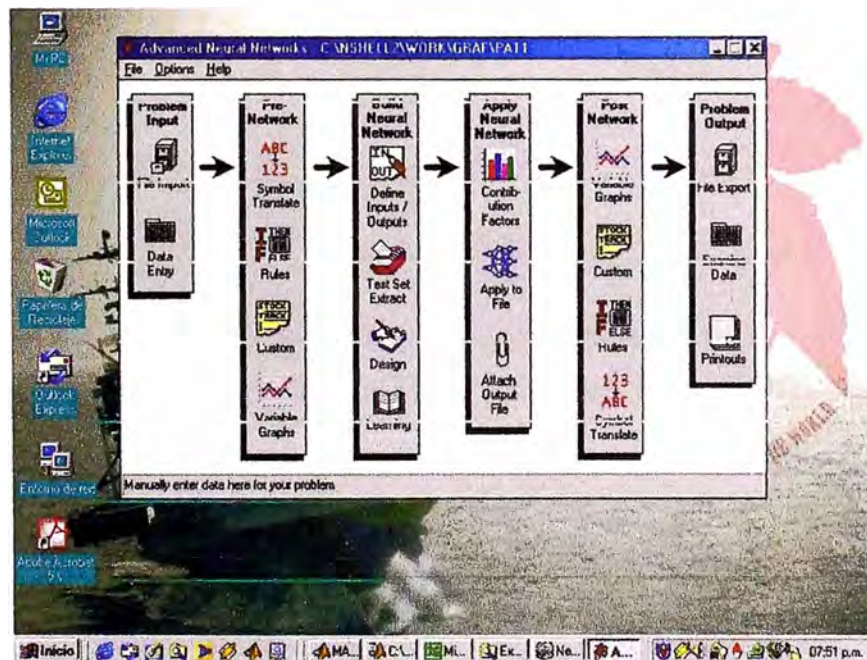


Figura 5.4. Simulador de Redes Neuronales con Neuroshell 2

Importamos los datos en formatos xxx.dat que contienen los patrones a entrenar. Se puede ver en la Figura 5.5.

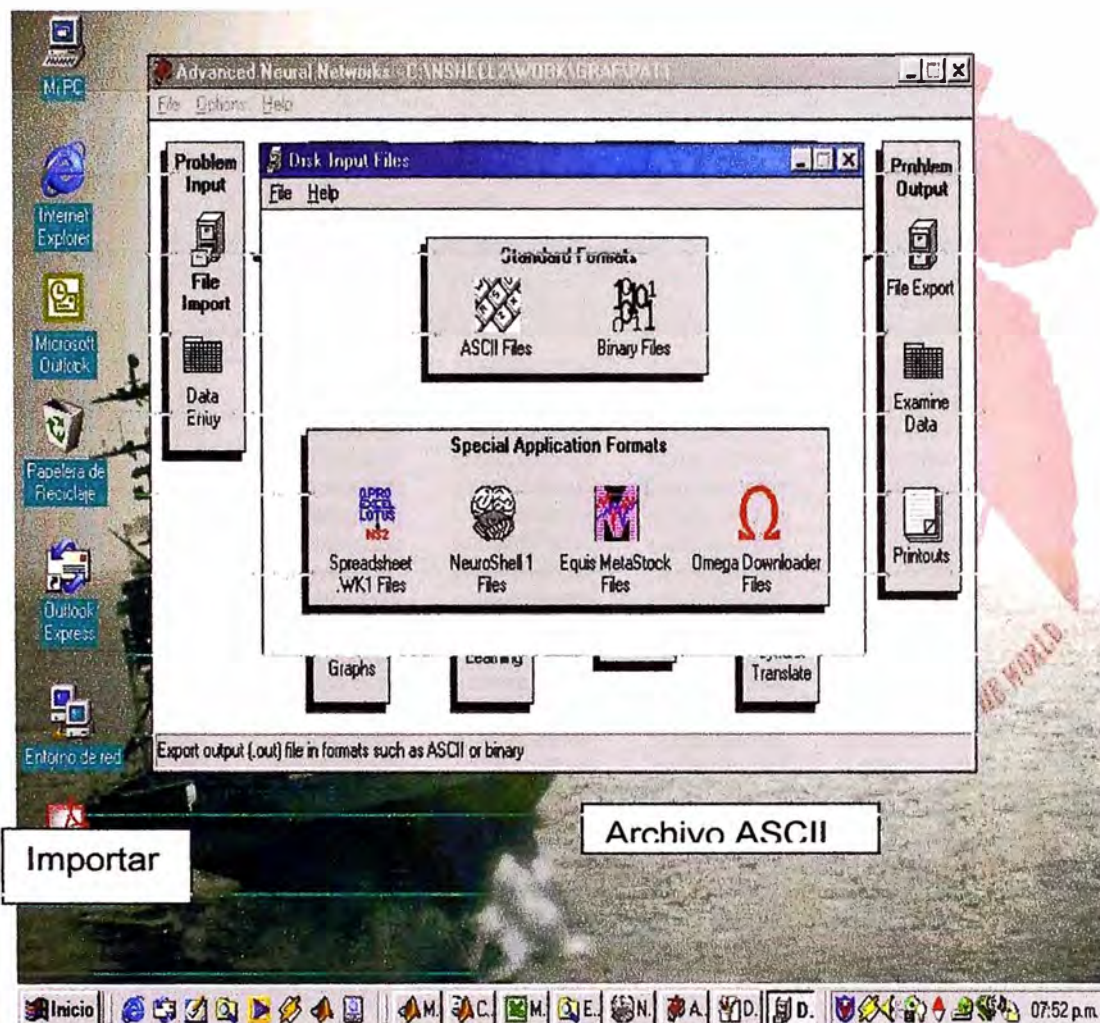


Figura 5.5. Importación de archivos en formato ASCII

Ahora abrimos la entrada de datos para ajustar los parámetros de entrada, que se aprecia en la Figura 5.6.

Microsoft Excel

Advanced Neural Networks - C:\NSHELL2\WORK\GRAF\PAT1

File Options Help

Datagrid: C:\NSHELL2\WORK\GRAF\PAT1

File Edit Format Help

Number of row with variable names (blank if none): left/right arrow keys end edit

First row containing actual training data: Size: 100 rows 147 columns

Note: This is not a commercial spreadsheet and may not load fast enough for large files. The NeuroShell 2 Options menu allows you to change the datagrid call to your own spreadsheet. Search help file for "datagrid" for details.

	A	B	C	D
1	IN1	IN2	IN3	IN4
2	0.000000000000	0.000000000000	0.000000000000	0.000000000000
3	1.000000000000	1.000000000000	1.000000000000	1.000000000000
4	1.000000000000	1.000000000000	1.000000000000	1.000000000000
5	0.000000000000	0.000000000000	1.000000000000	1.000000000000
6	1.000000000000	1.000000000000	0.000000000000	0.000000000000
7	0.000000000000	0.000000000000	0.000000000000	0.000000000000
8	0.000000000000	0.000000000000	0.000000000000	0.000000000000
9	1.000000000000	1.000000000000	1.000000000000	1.000000000000
10	1.000000000000	1.000000000000	1.000000000000	1.000000000000
11	1.000000000000	1.000000000000	1.000000000000	1.000000000000
12	1.000000000000	1.000000000000	1.000000000000	1.000000000000
13	0.000000000000	0.000000000000	0.000000000000	1.000000000000
14	0.000000000000	0.000000000000	1.000000000000	1.000000000000
15	1.000000000000	0.000000000000	0.000000000000	0.000000000000
16	1.000000000000	0.000000000000	0.000000000000	0.000000000000

Listo

Advanced Neural Networks - C:\NSHELL2\WORK\GRAF\PAT1

09:25 p.m.

Figura 5.6. Preparando Datos de Entrada para el entrenamiento de la Red

Luego definimos las entradas y las salidas. En esta pantalla definimos 140 entradas que van a corresponder a los puntos que conforman la matriz de 14 por 10 con la que hemos modelado cada vocal. Además también definimos las salidas que están en la Figura 5.7, que para este caso son 7 y corresponden a la representación del código ASCII en binario de cada vocal en mayúsculas o minúsculas.

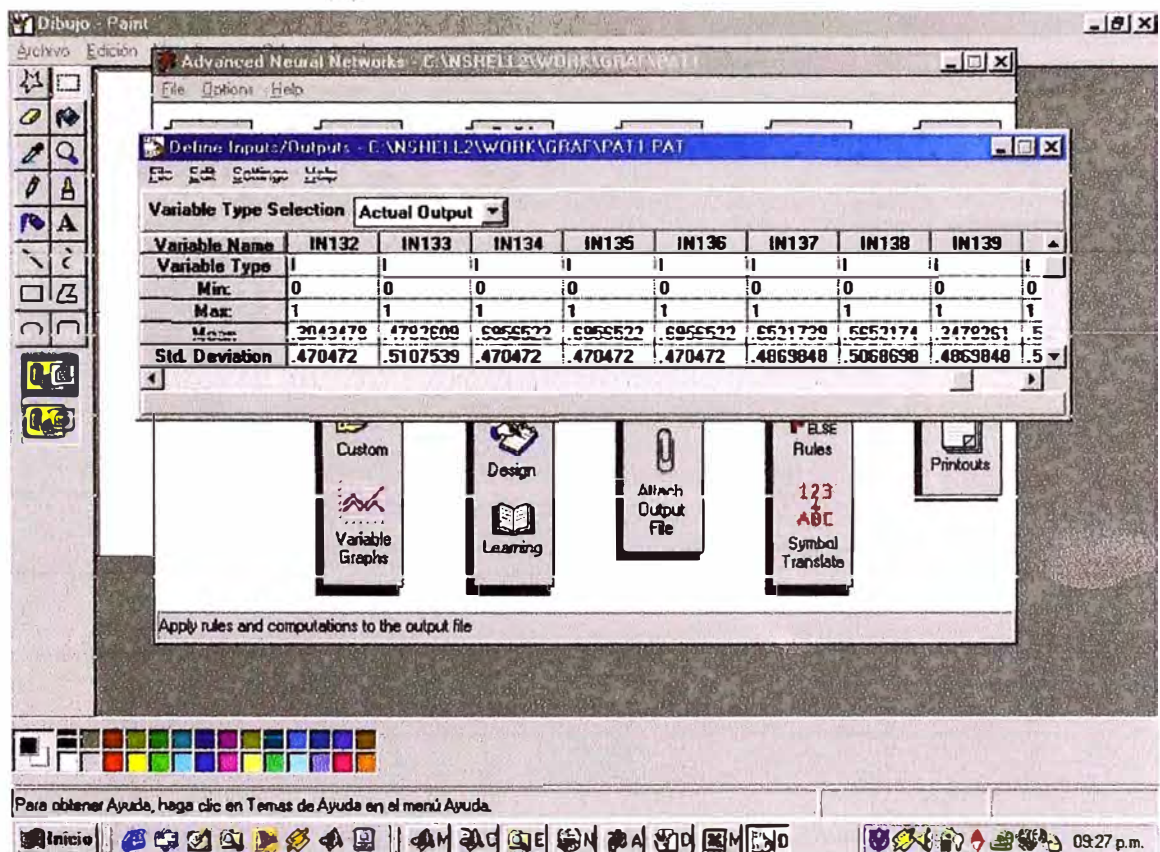


Figura 5.7. Preparando el Vector de salida para el entrenamiento de la Red

Ahora veremos como escogemos el grupo de prueba que como ya dijimos nos va a servir para realizar una calibración de nuestra red durante el entrenamiento. En nuestro caso escogemos la opción del 20% de los patrones totales. Para PATV1 Para PATV2 no escogemos nada.

La extracción de muestras se muestra en la Figura 5.8.

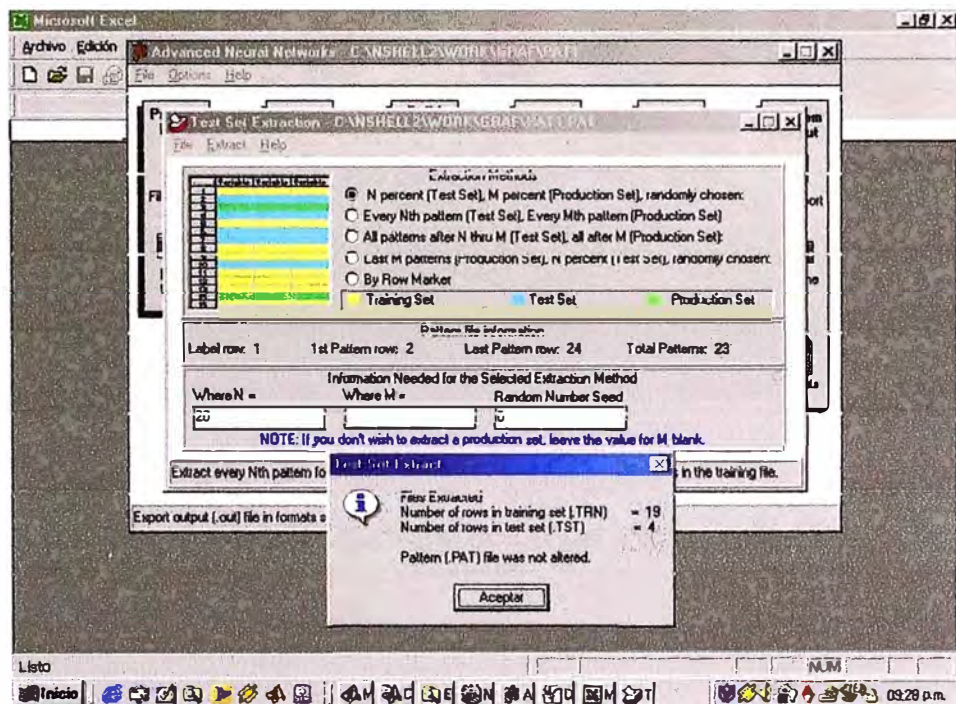


Figura 5.8. Extracción de muestra para el Test de Inicio

Definido esto es momento de diseñar la estructura de nuestra red. Como ya dijimos debe tener tres capas 1 de entrada con 140 neuronas, 1 capa oculta con 78 neuronas y una capa de salida con 7 neuronas. Estas capas se pueden ver en la Figura 5.9.

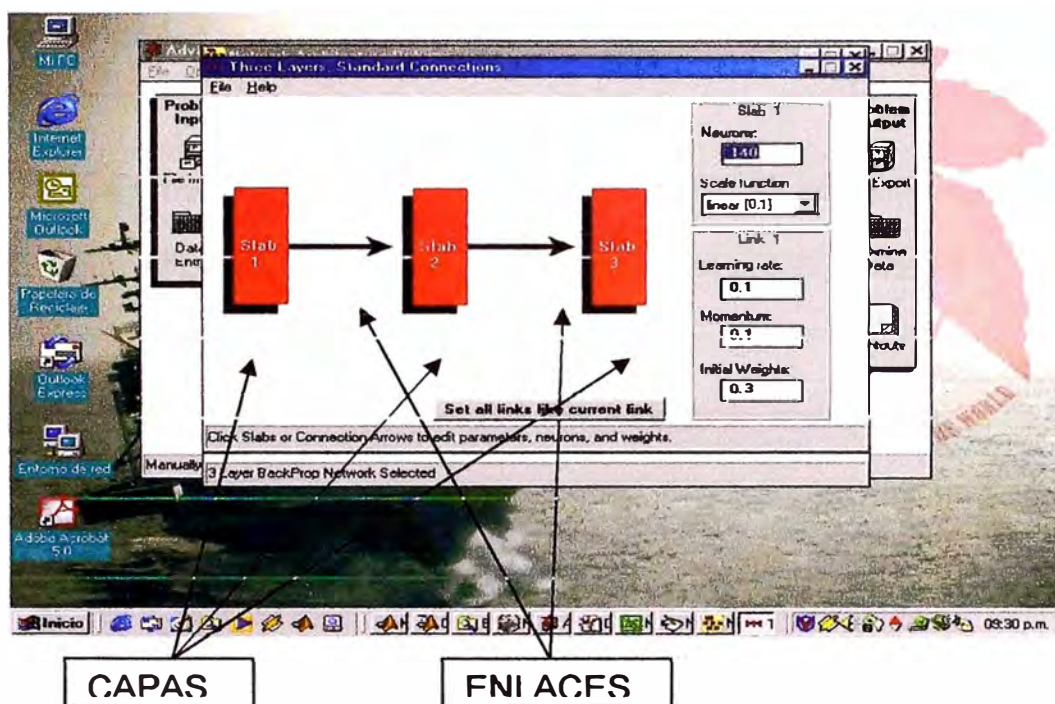


Figura 5.9. Preparando la configuración de inicio de la Red

BackPropagation

La cantidad de neuronas de la capa oculta la escogemos de acuerdo a la siguiente formula:

de neuronas ocultas = $1/2$ (Entradas + Salidas) + la raíz cuadrada de la cantidad de patrones de entrenamiento. La elección del criterio de entrenamiento se aprecia en la Figura 5.10.

Los parámetros se ajustan como se describió en estructura de la red. Luego definimos los criterios para detener el entrenamiento.

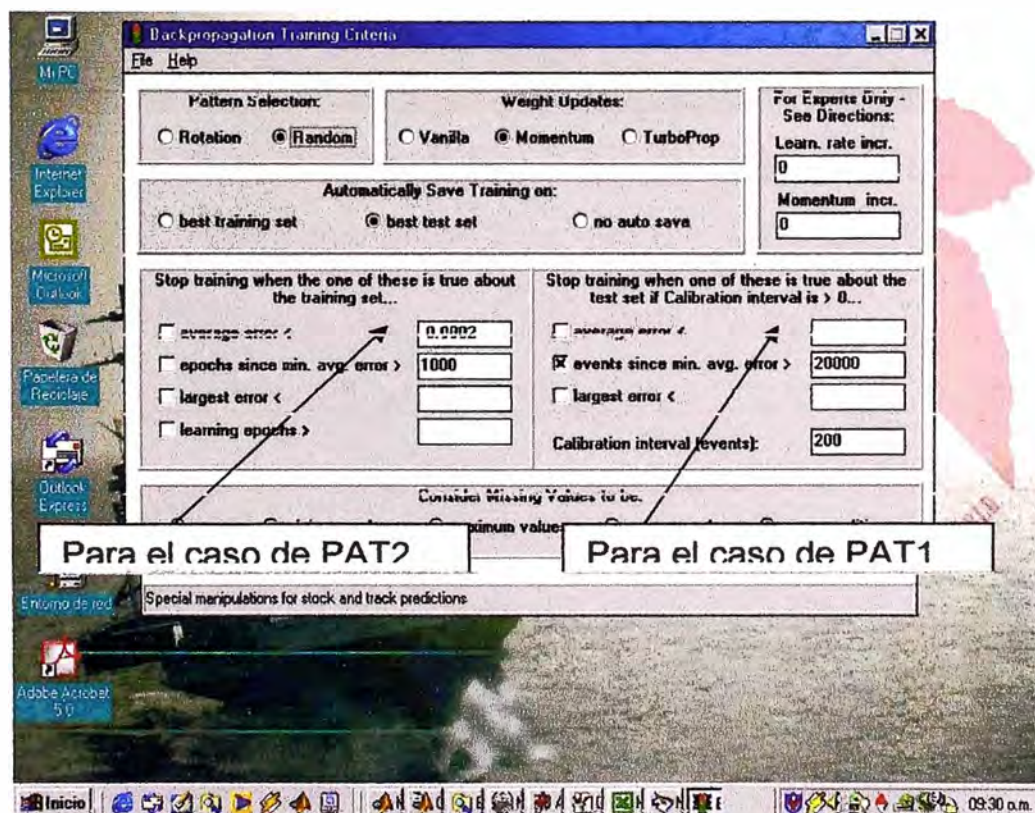


Figura 5.10. Elección de Criterios de Entrenamiento.

Ahora comenzamos a entrenar a la red. El aprendizaje de la red se ve en la Figura 5.11.

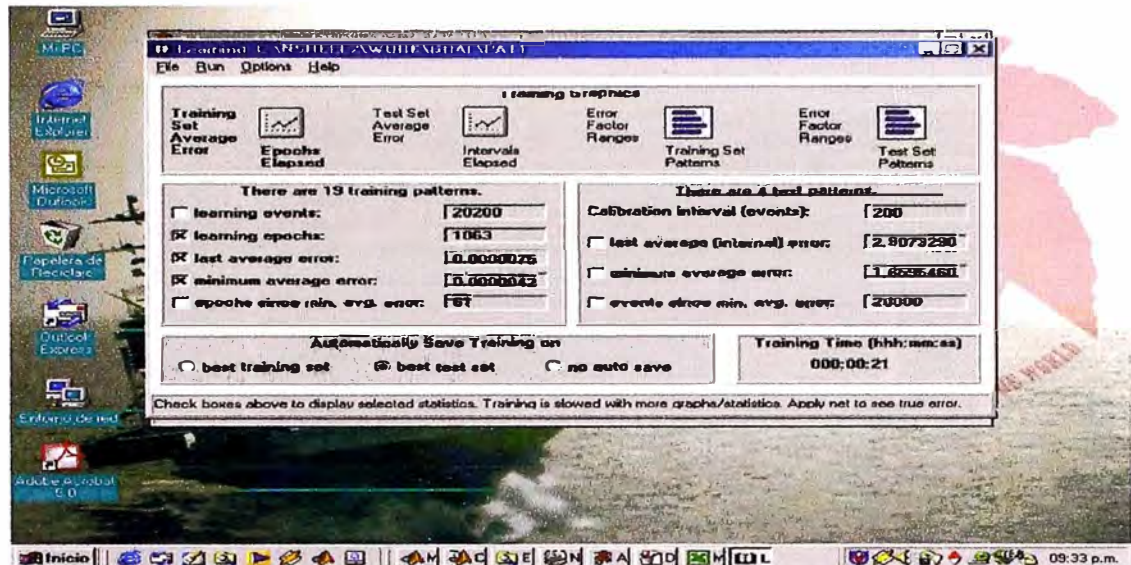


Figura 5.11. Observando aprendizaje de la red.

Después de entrenarla por cualquiera de los 2 métodos descritos. Probamos la red con los mismos patrones que utilizamos para el entrenamiento. La prueba de la red entrenada está en la Figura 5.12.

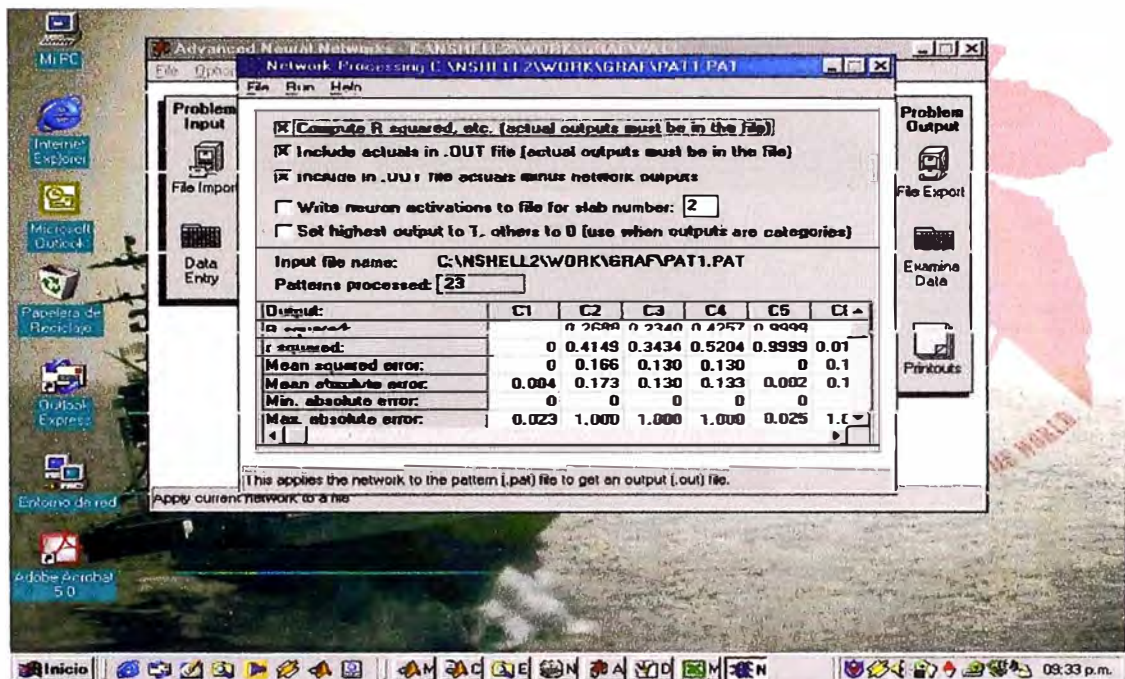


Figura 5.12. Probando la red que está entrenada.

Finalmente exportamos nuestros resultados en formato de texto separado por espacios .txt, este archivo es importado a MATLAB para realizar el procedimiento descrito anteriormente. Finalmente exportamos nuestro resultados que se aprecia en la Figura 5.13.

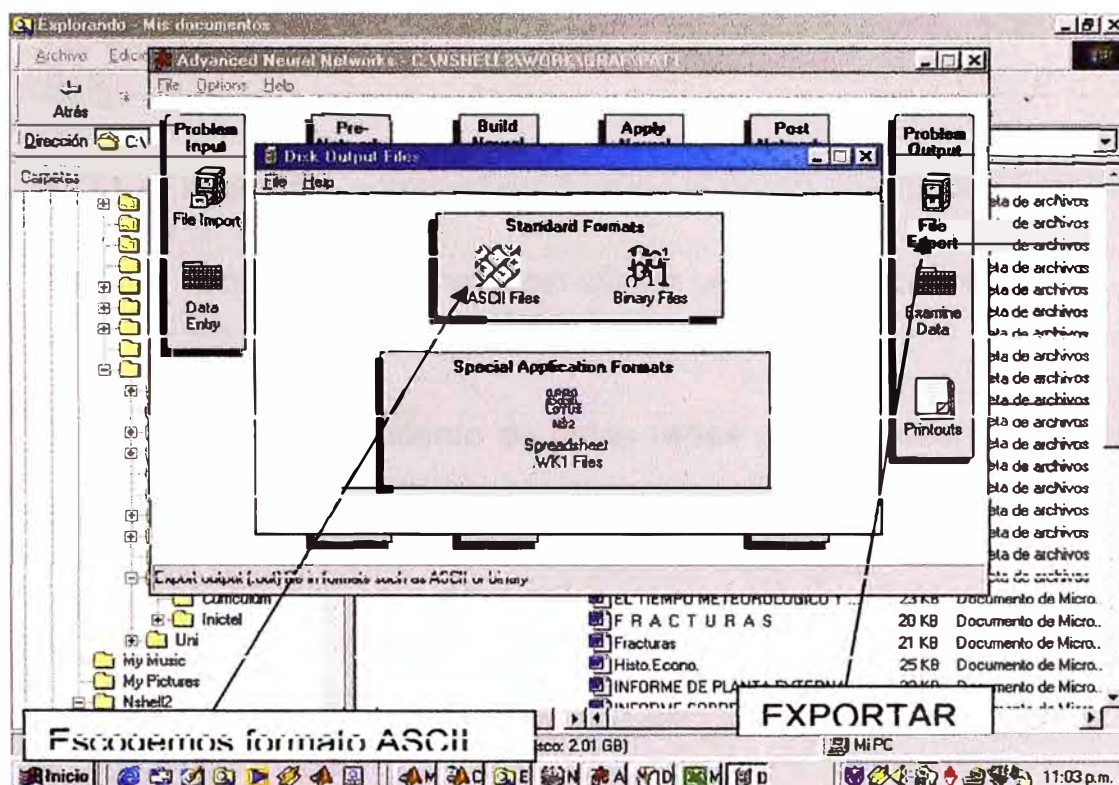


Figura 5.13. Exportando los resultados para analizarlos y presentarlos en una Hoja de Cálculo

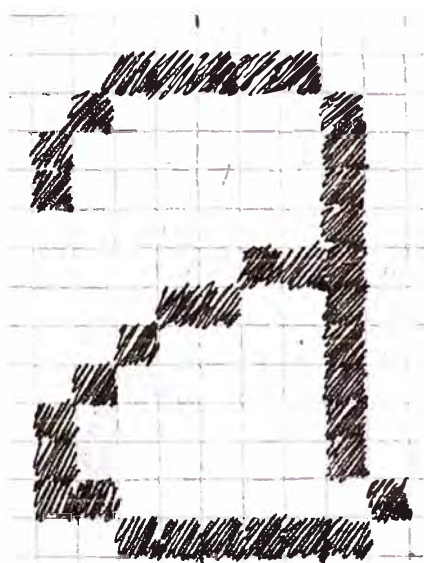
5.7. RESULTADOS

Después del entrenamiento de nuestra Red y de su prueba utilizando los mismos patrones de entrenamiento esta lista para ser utilizada con patrones que representan letras mal escritas o con mucho ruido. Para ver los resultados hemos nombrado a nuestras redes según su entrenamiento:

PAT1 → la que fue entrenada utilizando el grupo de prueba (Test Set)

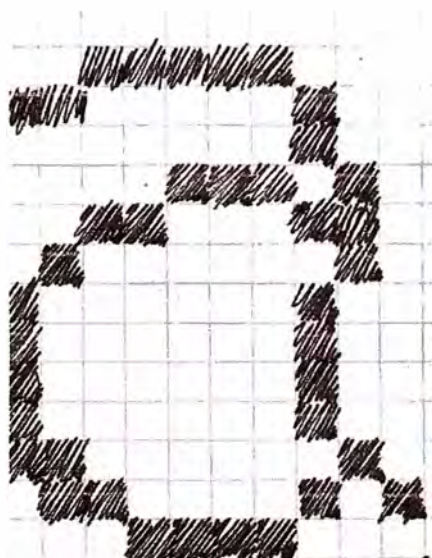
PAT2 → la que fue entrenada sin utilizar un grupo de prueba.

Después del entrenamiento de estas redes se ingresaran 8 patrones como sigue y se verificaran los resultados:



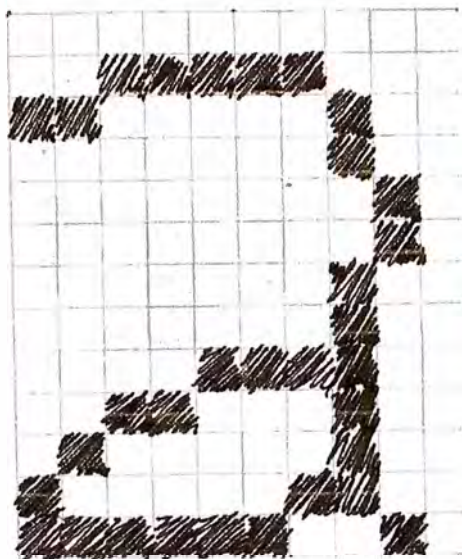
Patrón 1

a



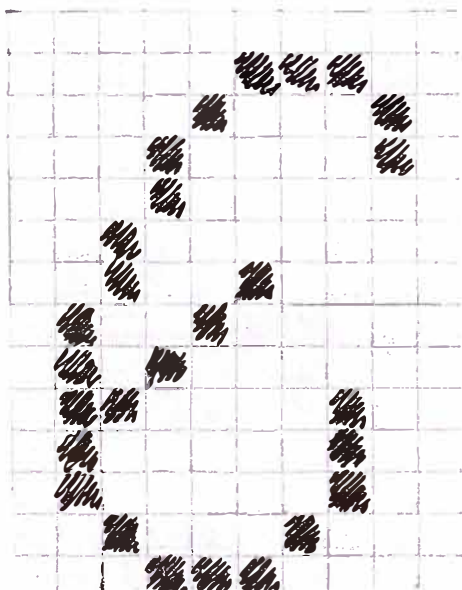
Patrón 2

a



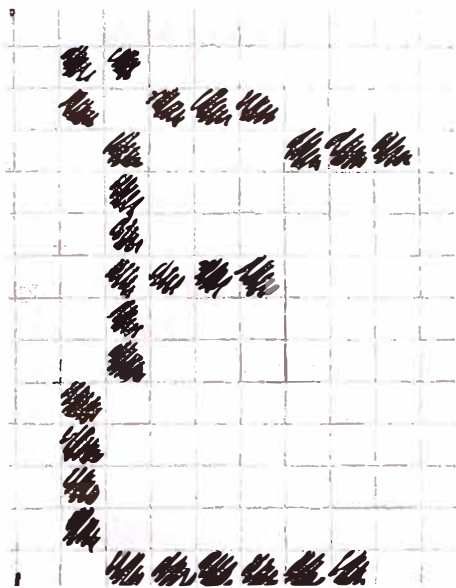
Patrón 3

a

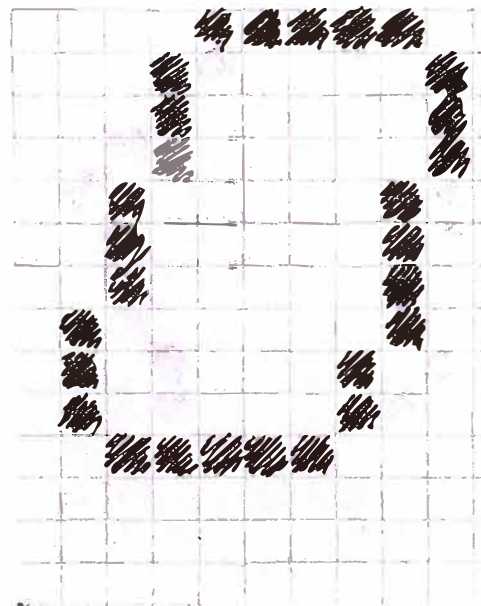


Patrón 4

e



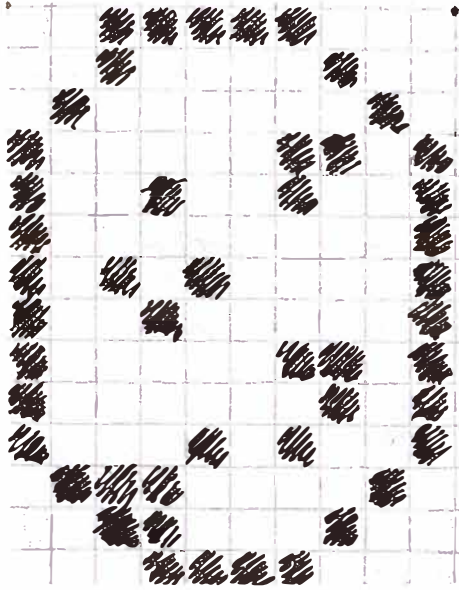
Patrón 5



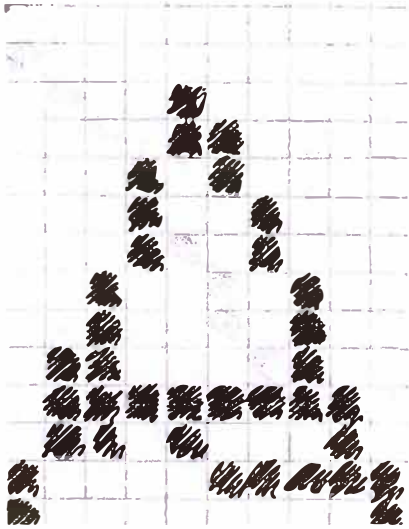
Patrón 6

E

O



Patrón 7



Patrón 8



Ahora veremos los resultados de cada una de ellas individualmente:

PAT1:

- Para entrenar esta red se escogió aleatoriamente un 20% de los patrones de entrenamiento como grupo de prueba (Test Set).

- El entrenamiento se detuvo según los parámetros de calibración utilizando el grupo de prueba. El entrenamiento debía detenerse cuando el número de eventos después de haber alcanzado el mínimo error promedio sea de 20000.

- El entrenamiento duró 21 seg. Y el mínimo error promedio conseguido fue de 0.0000042.

- Los pesos que se escogen son los de la **Mejor Red**. Y han sido archivados en PAT1.WTS, esto se puede apreciar en la Figura 5.14.

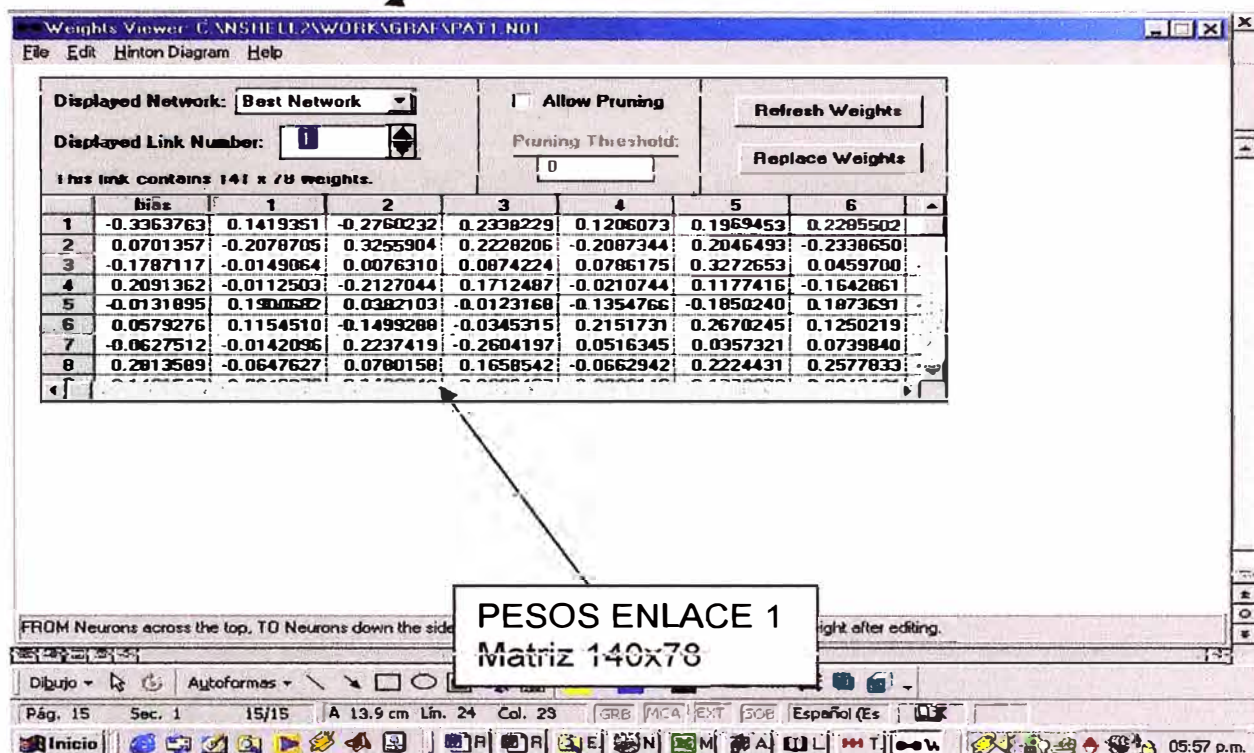


Figura 5.14. Examinando la matriz de pesos en la Red Entrenada

Ahora probamos primero el funcionamiento de la red utilizando los mismos patrones de entrenamiento. El resultado es exportado a un archivo de texto separado por tabulaciones con nombre PAT1.TXT, se observa los resultados en la tabla N° 5.1.

El error cometido es casi cero en la mayoría de las respuestas. Hemos pintado de amarillo el valor de los errores y de rojo las respuestas que han cambiado debido a este error. Como observamos las salidas que debían ser "1" se han convertido en "0" a la salida de la RED:

Esto esta afectando a los siguientes caracteres del patrón de entrenamiento.

O → 1001111 → 1000101 → E

e → 1100101 → 1000101 → E

Salida Esperada							Salidas de la RED PAT ¹							Error =(Salidas Esperadas - Salidas de la Red)						
1	0	0	0	0	0	1	1.000035	2.38E-07	0	2.21E-06	7.75E-07	6.29E-06	1.000013	-3.54E-05	-2.38E-07	0	-2.21E-06	-7.75E-07	-6.29E-06	-1.25E-05
1	0	0	0	1	0	1	1.000017	3.83E-07	0	1.80E-04	0.986006	5.90E-06	1.000001	-1.74E-05	-9.83E-07	0	-1.80E-04	0.013394	-5.90E-06	-1.43E-06
1	0	0	1	0	0	1	1.000348	3.20E-03	0	0.969636	2.42E-05	1.41E-05	1.000028	-3.48E-04	-3.20E-03	0	3.04E-02	-2.42E-05	-1.41E-05	-2.79E-05
1	0	0	1	1	1	1	1.000134	0	5.96E-08	2.94E-05	0.999959	1.01E-05	1.000004	-1.34E-04	0	-5.96E-08	0.999971	4.11E-05	0.999999	-4.05E-06
1	0	1	0	1	0	1	1.000089	0	2.20E-05	1.94E-05	0.999998	1.28E-06	1.000015	-8.90E-04	0	0.999978	-1.94E-05	1.91E-06	-1.28E-06	-1.55E-05
1	0	0	0	0	0	1	1.000009	2.46E-02	0	9.18E-06	0	6.65E-05	1.000215	-8.94E-05	-2.46E-02	0	-9.18E-06	0	-6.65E-05	-2.15E-04
1	0	0	0	0	0	1	1.000057	0	0	1.36E-05	2.69E-05	3.04E-06	1.000011	-5.66E-05	0	0	-1.36E-05	-2.69E-05	-3.04E-06	-1.10E-05
1	0	0	0	1	0	1	1.000905	2.38E-07	2.98E-08	6.21E-04	0.999165	1.17E-05	1.000008	-9.05E-04	-2.38E-07	-2.98E-08	-6.21E-04	8.35E-04	-1.17E-05	-8.46E-06
1	0	0	0	1	0	1	1.000478	3.58E-07	0	1.55E-02	0.97468	1.03E-05	1.000005	-4.78E-04	-3.58E-07	0	-1.55E-02	2.53E-02	-1.03E-05	-5.48E-06
1	0	0	1	0	0	1	1.014874	4.17E-06	0	0.999476	2.89E-06	3.18E-05	1.000275	-1.49E-02	-4.17E-06	0	5.24E-04	-2.89E-06	-3.18E-05	-2.75E-04
1	0	0	1	0	0	1	1.001526	1.37E-02	0	0.988075	2.34E-03	5.44E-05	1.000033	-1.53E-03	-1.37E-02	0	1.19E-02	-2.34E-03	-5.44E-05	-5.32E-05
1	0	0	1	1	1	1	1.002539	0	1.01E-06	2.41E-04	0.999998	1.16E-05	1.000033	-2.54E-03	0	-1.01E-06	0.999759	1.79E-06	0.999999	-3.27E-05
1	0	0	1	1	1	1	1.001859	0	6.56E-07	2.32E-03	0.999999	7.36E-05	1.000013	-1.86E-03	0	-6.56E-07	0.997583	1.43E-06	0.999993	-1.34E-05
1	0	1	0	1	0	1	1.009332	0	8.66E-05	1.54E-04	1	2.41E-06	1.000143	-9.33E-03	0	0.999914	-1.54E-04	1.19E-07	-2.41E-06	-1.43E-04
1	0	1	0	1	0	1	1.008696	0	1.26E-03	1.57E-04	1	1.06E-05	1.000076	-8.70E-03	0	0.998737	-1.57E-04	1.19E-07	-1.06E-05	-7.58E-05
1	1	0	0	0	0	1	1.000001	0.967417	0	5.07E-06	2.09E-07	9.54E-07	1.000001	-1.07E-05	3.26E-02	0	-5.07E-06	-2.09E-07	-9.54E-07	-1.07E-06
1	1	0	0	1	0	1	1.000266	1.97E-06	2.68E-07	1.37E-03	0.999767	7.72E-06	1.000009	-2.66E-04	0.999998	-2.68E-07	-1.37E-03	2.33E-04	-7.72E-06	-9.30E-06
1	1	0	1	0	0	1	1.003027	7.42E-02	1.01E-04	0.993942	2.82E-04	1.56E-03	1.006036	-2.30E-02	0.925811	-1.01E-04	6.06E-03	-2.82E-04	-1.56E-03	-6.09E-03
1	1	1	0	1	0	1	1.001872	0.999722	0.99738	2.16E-04	0.998983	5.67E-04	1	-1.87E-03	2.78E-04	2.62E-03	-2.16E-04	1.02E-03	-5.67E-04	-2.38E-07
1	1	0	0	0	0	1	1.000013	2.23E-02	0	6.18E-05	4.42E-04	2.32E-06	1.000001	-1.29E-05	0.977728	0	-6.18E-05	-4.42E-04	-2.32E-06	-9.54E-07
1	1	0	0	1	0	1	1.000281	1.28E-06	2.09E-07	8.03E-04	0.999011	4.65E-06	1.000011	-2.81E-04	0.999999	-2.09E-07	-8.03E-04	1.89E-04	-4.65E-06	-1.06E-05
1	1	0	1	0	0	1	1.006835	0.998233	3.25E-06	0.999916	8.34E-07	1.11E-03	1.002629	-6.83E-03	0.01177	-3.25E-06	8.44E-05	-8.34E-07	-1.11E-03	-2.63E-03
1	1	1	0	1	0	1	1.012081	0.999997	0.999981	5.14E-05	1	8.40E-03	1.000006	-1.21E-02	3.04E-06	1.91E-05	-5.14E-05	2.38E-07	-8.40E-03	-5.72E-06

Tabla Nº 5.1. Funcionamiento de Red PAT1

Después de introducir los patrones mencionados observamos en la Tabla N° 5.2. donde se muestran los resultados. Observamos que esta red presenta errores en los patrones:

Patron 1 → 1100001 → 1000001

Patrón 2 → 1100001 → 1000101

Patron 4 → 1100101 → 1100011

Patron 6 → 1001111 → 1000001

Patron 7 → 1001111 → 1000101

Patron 8 → 1000001 → 1100001

- Solo pudo reconocer correctamente los patrones 3 y 5.

Salidas Esperadas	Salidas de la RED PAT2						Error=(Salidas esperadas - Salidas de la Red)							
1 1 0 0 0 1	1.000476	0.962612	1.61E-06	0.999415	1.03E-04	4.70E-02	1.00031	-4.76E-04	3.74E-02	-1.61E-06	-0.999415	-1.03E-04	-4.70E-02	-3.10E-04
1 1 0 0 0 1	1.000023	0.7089127	2.16E-03	5.45E-04	0.3835077	1.73E-06	1.000079	-2.25E-05	0.2910873	-2.16E-03	-5.45E-04	-0.383508	-1.73E-06	-7.89E-05
1 1 0 0 0 1	1.000689	0.1774885	0.3889562	0.6401737	3.93E-04	4.42E-05	1.002356	-6.89E-04	0.822515	-0.388956	-0.640174	-3.93E-04	-4.42E-05	-2.36E-03
1 1 0 0 1 0 1	1.000015	0.9993903	2.03E-03	5.51E-05	0.5599773	2.68E-07	1.000607	-1.45E-05	9.66E-06	-2.03E-03	-5.51E-05	4.00E-02	-2.68E-07	-6.07E-04
1 0 0 0 1 0 1	1.000011	0.6623531	2.91E-03	1.24E-02	0.4692288	3.70E-06	1.000036	-1.06E-05	-0.662353	-2.91E-03	-1.24E-02	0.5307712	-3.70E-06	-3.61E-05
1 0 0 1 1 1 1	1.000279	0.9319888	6.09E-02	8.67E-06	0.2824742	4.07E-04	1.000473	-2.79E-04	-0.931989	-6.09E-02	0.9999913	0.7175258	0.9995934	-4.73E-04
1 0 0 1 1 1 1	1.000056	1.19E-07	2.14E-04	0.9794832	0.5999906	0.9999378	1.000023	-5.61E-05	-1.19E-07	-2.14E-04	2.05E-02	9.42E-06	6.22E-05	-2.29E-05
1 0 0 0 0 1	1.000864	0.9991952	0.4659927	6.44E-04	6.91E-05	0	1.001597	-8.64E-04	-0.999195	-0.465993	-6.44E-04	-6.91E-05	0	-1.60E-03

Tabla N° 5.4. Resultados de Red entrenada PAT2.

PAT2:

- Para entrenar esta red no se escogió ningún grupo de prueba (Test Set).
- El entrenamiento se detuvo manualmente cuando el mínimo error promedio se consideró suficientemente bajo.
- El entrenamiento duró 2 min. y el mínimo error promedio conseguido fue de 0.0000014.
- Los pesos que se escogen son los de **la Mejor Red**. Y han sido archivados en PAT2.WTS. Esto se aprecia en la Figura 5.15.

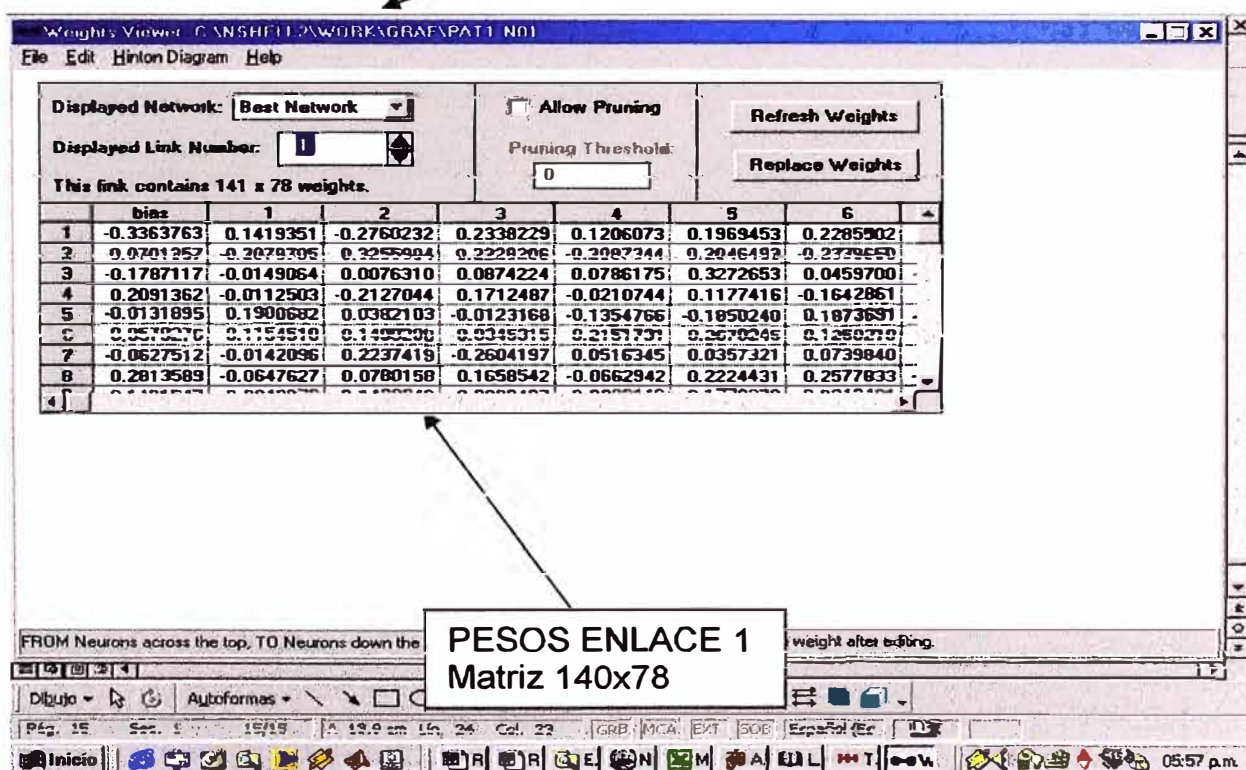


Figura 5.15. Escogiendo los pesos de la mejor red.

- Ahora probamos primero el funcionamiento de la red utilizando los mismos patrones de entrenamiento. El resultado es exportado a un archivo de texto separado por tabulaciones con nombre PAT2.TXT, cuyos resultados se aprecia en la Tabla N° 5.3.

Como observamos en este todos los errores son muy pequeños y no se presenta ninguna confusión en el reconocimiento de patrones.

Salida Esperada						
1	0	0	0	0	0	1
1	0	0	0	1	0	1
1	0	0	1	0	0	1
1	0	0	1	1	1	1
1	0	1	0	1	0	1
1	0	0	0	0	0	1
1	0	0	0	0	0	1
1	0	0	0	1	0	1
1	0	0	0	1	0	1
1	0	0	1	0	0	1
1	0	0	1	0	0	1
1	0	0	1	1	1	1
1	0	0	1	1	1	1
1	0	1	0	1	0	1
1	0	1	0	1	0	1
1	1	0	0	0	0	1
1	1	0	0	1	0	1
1	1	1	0	1	0	1
1	1	0	0	0	0	1
1	1	0	0	1	0	1
1	1	1	0	1	0	1

Salida de la Red PAT2						
1.000065	7.57E-05	3.61E-04	1.14E-05	0	0	1.000683
1.000006	1.10E-06	3.17E-05	8.51E-04	0.9999728	2.44E-06	1.000006
1.000161	9.83E-07	1.79E-07	0.9999967	2.98E-08	0	1.000089
1.000023	8.94E-07	6.20E-06	0.9988445	0.9999955	1	1.000029
1.0002	0	0.9992682	1.11E-03	0.9998778	2.60E-05	1.000098
1.000036	7.97E-04	3.88E-04	2.98E-08	1.40E-06	0	1.000414
1.000484	6.74E-05	9.22E-04	7.72E-04	0	1.19E-05	1.000327
1.000049	6.47E-06	3.05E-04	2.13E-04	0.9998853	5.66E-07	1.000012
1.000031	6.23E-06	6.44E-05	2.27E-04	0.9996206	2.98E-08	1.000008
1.00025	1.07E-06	8.94E-08	0.9999949	6.85E-07	0	1.000309
1.000258	1.49E-07	7.15E-07	0.9999943	4.80E-06	3.87E-07	1.00007
1.000032	2.68E-07	1.39E-05	0.9988126	0.9999995	1	1.000014
1.000048	0	3.26E-05	0.9994694	0.9999983	0.9999999	1.00001
1.000466	0	0.9999168	1.00E-04	1	2.05E-04	1.000059
1.000417	0	0.9995595	5.47E-05	1	2.16E-04	1.000073
1.000003	1	2.98E-08	9.08E-05	2.38E-07	8.94E-08	1.000018
1	1	0	1.39E-05	0.9999636	1.84E-04	1.000001
1.000459	0.9999806	1.76E-06	0.9999946	2.95E-06	0	1.000151
1.000716	0.9999998	0.9997776	0	0.9995192	0	1.000073
1.000004	1	8.94E-08	9.90E-04	4.82E-05	1.07E-04	1.000009
1	0.9999999	0	8.37E-06	0.9999895	3.44E-04	1.000003
1.000817	0.9998746	9.54E-07	0.9999995	2.98E-08	2.98E-08	1.000765
1.000251	1	0.999066	0	0.999923	0	1.000197

Error=(Salida Esperada - Salida de la Red)						
-6.52E-05	-7.57E-05	-3.61E-04	-1.14E-05	0	0	-6.83E-04
-6.08E-06	-1.10E-06	-3.17E-05	-8.51E-04	2.72E-05	-2.44E-06	-6.08E-06
-1.61E-04	-9.83E-07	-1.79E-07	3.34E-06	-2.98E-08	0	-8.94E-05
-2.32E-05	-8.94E-07	-6.20E-06	1.16E-03	4.53E-06	0	-2.91E-05
-2.00E-04	0	7.32E-04	-1.11E-03	1.22E-04	-2.60E-05	-9.82E-05
-3.58E-05	-7.97E-04	-3.88E-04	-2.98E-08	-1.40E-06	0	-4.14E-04
-4.84E-04	-6.74E-05	-9.22E-04	-7.72E-04	0	-1.19E-05	-3.27E-04
-4.94E-05	-6.47E-06	-3.05E-04	-2.13E-04	1.15E-04	-5.66E-07	-1.24E-05
-3.06E-05	-6.23E-06	-6.44E-05	-2.27E-04	3.79E-04	-2.98E-08	-7.63E-06
-2.50E-04	-1.07E-06	-8.94E-08	5.07E-06	-6.85E-07	0	-3.09E-04
-2.58E-04	-1.49E-07	-7.15E-07	5.72E-06	-4.80E-06	-3.87E-07	-6.97E-05
-3.21E-05	-2.68E-07	-1.39E-05	1.19E-03	4.77E-07	0	-1.41E-05
-4.84E-05	0	-3.26E-05	5.31E-04	1.67E-06	1.19E-07	-1.05E-05
-4.66E-04	0	8.32E-05	-1.00E-04	0	-2.05E-04	-5.88E-05
-4.17E-04	0	4.40E-04	-5.47E-05	0	-2.16E-04	-7.32E-05
-3.10E-06	0	-2.98E-08	-9.08E-05	-2.38E-07	-8.94E-08	-1.84E-05
-2.38E-07	0	0	-1.39E-05	3.64E-05	-1.84E-04	-1.19E-06
-4.59E-04	1.94E-05	-1.76E-06	5.36E-06	-2.95E-06	0	-1.51E-04
-7.16E-04	2.38E-07	2.22E-04	0	4.81E-04	0	-7.33E-05
-4.41E-06	0	-8.94E-08	-9.90E-04	-4.82E-05	-1.07E-04	-8.94E-06
-3.58E-07	5.96E-08	0	-8.37E-06	1.05E-05	-3.44E-04	-3.22E-06
-8.17E-04	1.25E-04	-9.54E-07	5.36E-07	-2.98E-08	-2.98E-08	-7.65E-04
-2.51E-04	0	9.34E-04	0	7.70E-05	0	-1.97E-04

Tabla N° 5.3. Funcionamiento Red PAT2

- Ya entrenada nuestra red observamos los resultados en la Tabla N° 5.4 y vemos que hay errores en los siguientes patrones:

Patrón 1 → 1100001 → 1101001

Patrón 3 → 1100001 → 1000001

Patrón 5 → 1000101 → 1100101

Patrón 7 → 1001111 → 1100001

Patrón 8 → 1000001 → 1100001

- Respondió correctamente a los patrones 2, 4 y 7. Y el error cometido no es grande con excepción del patrón 6.

Salidas Esperadas	Salidas de la RED PAT2							Error=(Salidas esperadas - Salidas de la Red)						
1 1 0 0 0 0 1	1.000476	0.962612	1.61E-06	0.999415	1.03E-04	4.70E-02	1.00031	-4.76E-04	3.74E-02	-1.61E-06	-0.999415	-1.03E-04	-4.70E-02	-3.10E-04
1 1 0 0 0 0 1	1.000023	0.7089127	2.16E-03	5.45E-04	0.3835077	1.73E-06	1.000079	-2.25E-05	0.2910873	-2.16E-03	-5.45E-04	-0.383508	-1.73E-06	-7.89E-05
1 1 0 0 0 0 1	1.000689	0.1773885	0.3889562	0.6401737	3.93E-04	4.42E-05	1.002356	-6.89E-04	0.822515	-0.388956	-0.640174	-3.93E-04	-4.42E-05	-2.36E-03
1 1 0 0 1 0 1	1.000015	0.9999903	2.03E-03	5.51E-05	0.5599773	2.68E-07	1.000607	-1.45E-05	9.66E-06	-2.03E-03	-5.51E-05	4.00E-02	-2.68E-07	-6.07E-04
1 0 0 0 1 0 1	1.000011	0.6626531	2.91E-03	1.24E-02	0.4692288	3.70E-06	1.000036	-1.06E-05	-0.662653	-2.91E-03	-1.24E-02	0.5307712	-3.70E-06	-3.61E-05
1 0 0 0 1 0 1	1.000279	0.9319888	6.09E-02	8.67E-06	0.2824742	4.07E-04	1.000473	-2.79E-04	-0.931989	-6.09E-02	0.9999913	0.7175258	0.9995934	-4.73E-04
1 0 0 1 1 1 1	1.000056	1.19E-07	2.14E-04	0.9794832	0.9999906	0.9999378	1.000023	-5.61E-05	-1.19E-07	-2.14E-04	2.05E-02	9.42E-06	6.22E-05	-2.29E-05
1 0 0 0 0 0 1	1.000864	0.4659927	6.44E-04	6.91E-05	0	1.001597	-8.64E-04	-0.999995	-0.465993	-6.44E-04	-6.91E-05	0	-1.60E-03	

Tabla N° 5.4. Resultados de Red entrenada PAT2.

CONCLUSIONES

1. La red Backpropagation presenta una excelente capacidad para reconocer patrones. Después del entrenamiento cuando se le ingresa un patrón con distorsionado la red logra reconocerlo.
2. Para reconocimiento de patrones con poca distorsión (patrones de entrenamiento) la red PAT2 (no presenta errores grandes) se desempeña muchísimo mejor que la red PAT1 (presenta errores muy elevados).
3. Para reconocer patrones con gran distorsión como los del archivo PATx.PRO ambas redes cometen errores, siendo la red PAT 2 la que presenta menor cantidad de errores.
4. Durante el entrenamiento la RED PAT1 converge en 21 seg. Al utilizar un grupo previo de entrenamiento con el cual se calibra el error dando como resultado un mínimo error promedio de 0.0000042.
5. Durante el entrenamiento de la RED PAT2 esta no converge y error sigue disminuyendo hasta que fue detenida manualmente cuando se alcanzo un mínimo error promedio de 0.0000014 después de transcurridos 2 minutos de entrenamiento.

6. Como vemos el entrenamiento de la Red PAT1 se lleva a cabo en menor tiempo que la Red PAT2. pero tiene una respuesta de menor calidad.
7. Para mejorar la respuesta de ambas Redes el primer paso sería aumentar el tiempo de entrenamiento y/o aumentar la cantidad de patrones de entrenamiento.
8. El segundo paso si esto no funciona sería el de agregar neuronas a la capa oculta, con consecuencia de que estaríamos agregando tiempo de procesamiento a la red.
9. Y Tercero agregar mas capas ocultas lo que causaría un aumento de las interconexiones y por lo tanto tiempo de procesamiento.
10. En este trabajo se ha mostrado un esquema para el reconocimiento de caracteres escritos, usando una red neuronal multicapa y entrenada con el algoritmo Backpropagation. Los resultados finales obtenidos por el este reconocedor son de un error de clasificación con un porcentaje aceptable sin rechazo de patrones.

BIBLIOGRAFÍA

1. UNIVERSITAS Enciclopedia cultural, Salvat, 1971.
2. Marr, David. "Artificial Intelligence – A personal view". Vol. 9, 1977 (2)
3. W.K. Pratt, Digital Image Processing. New York: Wiley, 1978.
4. CÓMO FUNCIONA, Enciclopedia Salvat de la técnica, 1981.
5. Stefan Knerr, Leon Personnaz, Gerard Dreyfus, Handwritten Digit Recognition by Neural networks with Single-Layer Training, IEEE Trans. on Neural Networks Vol.3. 1992.
6. GUYTON Tratado de Fisiología Médica, 8ª Edic. Interamericana - McGraw –Hill. 1993.
7. Anderson, James A. "An Introduction to Neural Networks". MIT press, 1995
8. Seong-Whan Lee, Off-Line Recognition of totally unconstrained handwritten numerals using multiplayer cluster neural networks, IEEE Trans. on P.A.M.I. Vol 18, 1996.
9. Enciclopedia Multimedia en CD-ROM (Los seres vivos) . Artículo: Control de Ordenadores por Señales Neurales. Investigación y Ciencia, diciembre, 1996.
10. Ripley, B.D. "Can Statistical Theory Us use Neural Networks, Betters", 1997