

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

Redes neurales
aplicadas al pronóstico del índice de precios al consumidor

TESIS

PRESENTADA A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

Virginia Victoria Tala Ayerdi

AL CONFERÍRSELE EL TÍTULO DE
INGENIERO EN CIENCIAS Y SISTEMAS

Guatemala, septiembre de 1,997

PROPIEDAD DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
Biblioteca Central



08
T(4098)
C.4

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de tesis titulado:

**Redes neurales
aplicadas al pronóstico del índice de precios al consumidor**

tema que me fuera asignado por la Dirección de la Escuela de Ciencias y Sistemas con fecha 24 de agosto de 1,992.

Virginia Victoria Tala Ayerdi

Guatemala, septiembre de 1,997



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

MIEMBROS DE JUNTA DIRECTIVA

DECANO	Ing. Herbert René Miranda Barrios
Vocal 1o.	Ing. Miguel Ángel Sánchez Guerra
Vocal 2o.	Ing. Jack Douglas Ibarra Solórzano
Vocal 3o.	Ing. Juan Adolfo Echeverría Méndez
Vocal 4o.	Br. Víctor Rafael Lobos Aldana
Vocal 5o.	Br. Wagner Gustavo López Cáceres
Secretario	Ing. Gilda Marina Castellanos de Illescas

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Julio Ismael González Podszueck
Examinador	Ing. José Christian Bradna Villanueva
Examinador	Ing. Mario Enrique Sosa Castillo
Examinador	Ing. Roberto Marroquín Duarte
Secretario	Ing. Francisco Javier González López

Guatemala,
28 de Junio de 1996

Ing.
Jorge Luis Alvarez Mejia
Coordinador de la Comisión de Tesis
Ingenieria en Ciencias y Sistemas
Universidad de San Carlos de Guatemala
Ciudad de Guatemala

Estimado Ingeniero:

Me permito presentarle el trabajo de tesis de la estudiante **Virginia Victoria Tala Ayerdi**, el cual se titula: **Redes Neurales Aplicadas al Pronóstico de Indice de Precios al Consumidor**.

En mi calidad de asesor, he analizado el contenido, así como las conclusiones y recomendaciones expuestas en el trabajo, después de haber discutido a través de su elaboración en reuniones conjuntas con la interesada, habiendo hecho las modificaciones pertinentes y considerando que dicho trabajo es de gran interés para la Facultad de Ingeniería, dejo constancia de mi consideración del tribunal que sea designado para el examen de tesis.

Basándome en la aprobación del protocolo de tesis otorgada por la Coordinación de la Carrera de Ingeniería en Ciencias y Sistemas, me responsabilizo conjuntamente con el autor por el contenido, conclusiones y recomendaciones de dicho trabajo.

Sin otro particular, me es grato suscribirme de usted.

Atentamente,



Ing. Carlos Armando Amado Avila
Ingeniero en Sistemas

Guatemala, agosto 2, 1996

Ingeniero
Calixto Monzón
Director Escuela de Ciencias y Sistemas
Facultad de Ingeniería
Universidad de San Carlos de Guatemala

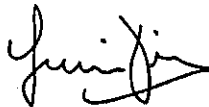
Estimado Ing. Monzón:

Por este medio, me permito informarle que he procedido a revisar el trabajo de tesis titulado **Redes Neuronales Aplicadas al Pronóstico del Índice de Precios al Consumidor**, elaborado por la estudiante Virginia Victoria Tala Ayerdi, y a mi juicio, el mismo cumple con los objetivos propuestos para su desarrollo.

Por lo tanto, el autor de este trabajo de tesis, su asesor Ing. Armando Amado y este servidor, nos hacemos responsables por el contenido, recomendaciones y conclusiones del mismo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme.

Atentamente,



Ing. Francisco Guevara
REVISOR



FACULTAD DE INGENIERIA

Escuelas de Ingeniería Civil, Ingeniería
Mecánica Industrial, Ingeniería Química,
Ingeniería Mecánica Eléctrica, Técnica
y Regional de Post-grado de Ingeniería
Sanitaria.

Ciudad Universitaria, zona 12
Guatemala, Centroamérica

Guatemala,
19 de septiembre de 1,997
REF.: CS. 080.97

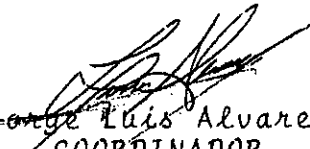
Ingeniero
Herbert René Miranda Barrios
Decano, Facultad de Ingeniería

señor Decano:

Atentamente me dirijo a usted, para informarle que
después de conocer el dictamen del Asesor del trabajo de
tesis del estudiante VIRGINIA VICTORIA TALA AVERDI, titu-
lado REDES NEURALES APLICADAS AL PRONOSTICO DEL INDICE DE
PRECIOS AL CONSUMIDOR, procedo a la autorización del mismo.

Atentamente,

"ID Y ENSEÑAD A TODOS"


Ing. Jorge Luis Alvarez Mejía
COORDINADOR
INGENIERIA EN CIENCIAS Y SISTEMAS

JLAM/edj

c.c. Archivo






FACULTAD DE INGENIERIA

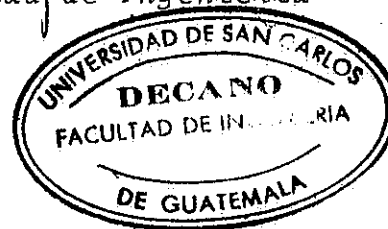
Escuelas de Ingeniería Civil, Ingeniería Mecánica Industrial, Ingeniería Química, Ingeniería Mecánica Eléctrica, Técnica y Regional de Post-grado de Ingeniería Sanitaria.

Ciudad Universitaria, zona 12
Guatemala, Centroamérica

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la autorización por parte del Coordinador de la Carrera de Ingeniería en Ciencias y Sistemas, al trabajo de tesis titulado REDES NEURALES APLICADAS AL PRONOSTICO DEL INDICE DE PRECIOS AL CONSUMIDOR, presentado por el estudiante universitario VIRGINIA VICTORIA TALA AVERDI, procede a la autorización para la impresión de la misma.

IMPRIMASE;


Ing. Herbert René Moranda Barrios
Decano de la Facultad de Ingeniería



Guatemala, septiembre de 1, 997

AGRADECIMIENTO

A Dios

Mi amigo más fiel.

A la Universidad de San Carlos de Guatemala

En cuyas aulas se han formado ilustres guatemaltecos, y a la cual honraré toda mi vida.

A mis catedráticos

Por sus sabias enseñanzas.

A mis padres

A quienes les debo el ser y que con su amor y esfuerzo han permanecido a mi lado, aún cuando la distancia nos separe.

A mis abuelas y abuelos

Cuya herencia ha sido valorar el conocimiento. Especialmente a mi abuelita Clara, a quien considero una segunda madre.

A mi tío Antonio

Quien me ha brindado lo mejor de sí.

A Sergio Alemán

Por su apoyo incondicional en la culminación de mi carrera, pero especialmente, por ser el portador de un mañana mejor.

A mis asesores y revisores de tesis

Cuyo esfuerzo desinteresado ha mejorado el contenido de este documento.

Al Licenciado Hilcías Aguilar

Cuyos aportes contribuyeron a realizar la aplicación de este trabajo de tesis.

A mis compañeros de estudio

Con quienes compartimos largas horas durante nuestra preparación académica.

Al lector

Por permitirme hacer un aporte a sus conocimientos.

Indice general

Indice de figurasv

Indice de tablas..... vii

Glosario ix

Introducción xii

Objetivosxiii

1. CEREBRO Y MODELOS..... 1

 1.1 Introducción a la neurofisiología 1

 1.2 Modelo de McCulloch-Pitts3

 1.3 Autómatas finitos y redes neurales.....4

 1.4 El perceptrón4

 1.4.1 Reglas de refuerzo5

 1.4.2 Conexiones aleatorias6

 1.5 Retroalimentación.....8

 1.6 Modelo de Peter H. Green.....9

 1.7 Homeóstasis..... 11

2. TEORÍA GENERAL DE REDES NEURALES 12

 2.1 Introducción 12

 2.2 Biológicamente inspiradas..... 12

 2.3 Neuronas 13

 2.4 Conexiones..... 13

 2.5 Estratos..... 15

 2.6 Red neural 16

 2.7 Modelo de neurona..... 16

 2.8 Modelo de red..... 17

2.9 Matriz de pesos	18
2.10 Funciones de activación y transferencias.....	18
2.10.1 Función de transferencia lineal.....	20
2.10.2 Función de transferencia umbral.....	20
2.10.3 Función de transferencia step.....	20
2.10.4 Función de transferencia Sigmoid.....	21
2.10.5 Función de transferencia de Gauss.....	22
2.11 Entrenando una red neural.....	22
2.11.1 Reglas de aprendizaje.....	23
2.11.1.1 Regla de Hebb.....	23
2.11.1.2 Regla delta.....	23
2.11.2 Formas de aprendizaje.....	24
2.11.2.1 Supervisada.....	24
2.11.2.2 No supervisada.....	24
2.11.2.3 Auto supervisada.....	24
2.11.3 Algoritmos de aprendizaje.....	24
2.11.3.1 Algoritmo de propagación hacia atrás.....	24
2.11.3.2 Algoritmo ATNN.....	24
2.11.3.3 Algoritmo de entrenamiento rápido para redes neuronales con múltiples estratos.....	25
2.11.3.4 Entrenamiento iterativo.....	25
2.11.3.5 Entrenamiento forma óptima.....	26
2.12 Número de ejemplos requeridos para el entrenamiento.....	27
3. CLASIFICACIÓN DE REDES NEURALES.....	28
3.1 Introducción.....	28
3.2 Desarrollo histórico de las teorías de redes neuronales.....	29
3.3 Clasificación de redes neuronales.....	30
3.3.1 Retroalimentación.....	31
3.3.1.1 Red Hopfield.....	32

3.3.1.2 Red BAM	33
3.3.1.3 Red ART	34
3.3.2 Alimentación hacia adelante (BPN)	34
3.3.2.1 Perceptrón	35
3.3.2.2 Asociador lineal	36
3.3.2.3 Red de Kohonen.....	36
3.3.2.4 Neocognitrón	37
3.3.3 Probabilística (PNN)	37
3.3.3.1 Red neural de regresión general (GRNN)	38
3.3.3.2 Otros Modelos PNN.....	39
3.4 Comparaciones entre PNN y BPN.....	40
3.4.1 Precisión	41
3.4.2 Velocidad	41
3.4.3 Conclusiones	41
4. SOFTWARE Y HARDWARE PARA REDES NEURALES	42
4.1 Introducción	42
4.2 Software disponible	43
4.2.1 Autonet	43
4.2.2 Net Talk	43
4.2.3 Neuroshell.....	44
4.2.4 Brainmaker	44
4.2.5 Neuralworks Professional II.....	46
4.3 Máquinas neurales	47
5. APLICACIÓN	48
5.1 Introducción	48
5.2 Definición del caso de estudio	48
5.2.1 Hardware y software utilizado.....	49
5.3 Identificación de la información	50

5.4 Reuniendo los datos	52
5.4 Preparación de los datos	52
5.4.1 Definición de la matriz	52
5.4.2 Datos representados como cambios	53
5.4.3 Tendencia	55
5.4.4 Datos preparados	56
5.5 Creación archivos de entrenamiento y pruebas	57
5.6 Construcción	59
5.6.1 Tipo de red	59
5.6.2 Función de transferencia	60
5.6.3 Neuronas escondidas	60
5.6.3.1 Menos de la mitad del número de entradas y salidas	61
5.6.3.2 Cinco al diez por ciento del número de los casos de entrenamiento	61
5.6.3.3 Número de ejemplos dividido el número de entradas y multiplicado por la tolerancia	61
5.7 Entrenando la red neural	61
5.8 Pruebas de la red neural	62
5.9 Ejecución de la red	66
5.10 Análisis de los resultados	68
Conclusiones	xiv
Recomendaciones	xv
Bibliografía	xvi

Indice de figuras

Figura 1..... 1
El Sistema nervioso humano considerado como un sistema de tres etapas

Figura 2..... 2
Dibujo esquemático de una neurona

Figura 3..... 5
Esquema de un perceptrón

Figura 4..... 6
Esquema de un perceptrón sencillo

Figura 5..... 7
Letra E percibida por un perceptrón

Figura 6..... 9
Sistema de retroalimentación negativa

Figura 7..... 14
Modelo de neurona simplificado

Figura 8..... 14
Red neural simple

Figura 9..... 17
Red con retroalimentación y competencia

Figura 10..... 18
Matriz de pesos

 Figura 10.1..... 18

 Figura 10.2..... 18

 Figura 10.3..... 18

 Figura 10.4..... 18

Figura 11..... 19
Funciones de una neurona

Figura 12..... 20
Función de transferencia lineal

Figura 13..... 20
Función de transferencia umbral

Figura 14.....21
 Función de transferencia step

Figura 15.....21
 Función de transferencia Sigmoid

Figura 16.....22
 Función de transferencia de Gauss

Figura 17.....32
 Red con retroalimentación

Figura 18.....33
 Red Hopfield

Figura 19.....33
 Red BAM

Figura 20.....34
 Red ART

Figura 21.....35
 Red BPN

Figura 22.....37
 Red de Kohonen

Figura 23.....69
 Red umbral con 1 neurona escondida

Figura 24.....69
 Red umbral con 2 neuronas escondidas

Figura 25.....70
 Red Sigmoid con 1 neurona escondida

Figura 26.....70
 Red Sigmoid con 2 neuronas escondidas

Figura 27.....71
 Red Gauss con 2 neuronas escondidas

Índice de tablas

Tabla 130
Resumen histórico

Tabla 231
Clasificación de las redes neurales

Tabla 340
Características de BPN y de PNN

Tabla 450
Cálculo del índice de precios al consumidor

Tabla 551
Ponderaciones del índice de precios al consumidor. Producto: azúcar

Tabla 652-53
Datos etiquetados

Tabla 754-55
Valores de cambio

Tabla 855-56
Indicadores de la tendencia

Tabla 957
Archivo de datos

Tabla 1058-59
Datos de entrenamiento

Tabla 1159
Datos para pruebas

Tabla 1260
Número de conexiones con una neurona escondida

Tabla 1360
Número de conexiones con dos neuronas escondidas

Tabla 1462
Resultados del entrenamiento

Tabla 1563
Pruebas red lineal con una neurona

Tabla 1663
Pruebas red lineal con dos neuronas

Tabla 1764
Pruebas red umbral con una neurona

Tabla 1864
Pruebas red umbral con dos neuronas

Tabla 1964
Pruebas red step con una neurona

Tabla 2064
Pruebas red step con dos neuronas

Tabla 2165
Pruebas red Sigmoid con una neurona

Tabla 2265
Pruebas red Sigmoid con dos neuronas

Tabla 2365
Pruebas red Gauss con una neurona

Tabla 2465
Pruebas red Gauss con dos neuronas

Tabla 2566-67
Datos para la Ejecución

Tabla 2667
Pruebas red umbral con una neurona

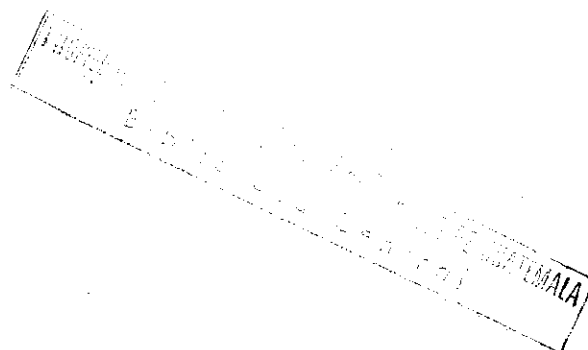
Tabla 2767
Pruebas red umbral con dos neuronas

Tabla 2868
Pruebas red Sigmoid con una neurona

Tabla 2968
Pruebas red Sigmoid con dos neuronas

Tabla 3068
Pruebas red Gauss con dos neuronas

Tabla 3171
Resultados del Entrenamiento de las cinco redes analizadas



Glosario

aprendizaje.

Fase en una red neural cuando nuevos datos son introducidos a la red, causando que los pesos de elementos de procesamiento sean ajustados.

arquitectura.

Estructura de la organización de un sistema de computación.

ataxia.

Disfunción del sistema nervioso.

axón.

Parte de la neurona. Es un cilindro estrecho que transporta los impulsos desde el soma a otras células. Parte eléctrica activa de una célula nerviosa.

back-propagation.

Algoritmo de aprendizaje para una red de múltiples estratos en la cual los pesos son modificados mediante la propagación de una señal de error desde las salidas a las entradas (backward).

bit.

Abreviatura de dígito binario. Unidad de información igual a una decisión binaria o a uno de dos posibles estados como uno o cero; encendido o apagado. El menor elemento de información en un sistema de notación binaria.

byte.

Número de bits (unos y ceros binarios) que una computadora necesita para representar un solo carácter o palabra. Una palabra es la menor unidad direccionable del almacenamiento principal de una computadora.

célula glial.

Las células gliales forman parte del sistema nervioso y sirven para soportar y nutrir las neuronas, otra de sus funciones es memorizar.

cibemética.

Ciencia que estudia comparativamente los sistemas de comunicación y regulación automática de los seres vivos con sistemas electrónicos y mecánicos semejantes a aquellos.

conexión.

Camino entre los elementos de procesamiento, tanto positivos o negativos, que une los elementos de procesamiento en la red.

dendrita.

Parte terminal de una célula nerviosa que lleva impulsos hacia el cuerpo de la célula. Partes eléctricas pasivas de una célula nerviosa.

efector.

Dícese del órgano o la parte orgánica en que esa acción se manifiesta.

entrenamiento.

Proceso en el cual una red aprende a asociar un patrón de entrada con una respuesta correcta.

estímulo.

Incitación para actuar.

estocástico.

Envuelve cambio, probabilidad o variable aleatoria.

estrato escondido.

Un tercer estrato de unidades entre los estratos de entrada y salida que provee poder adicional de cómputo.

función de activación.

Función mediante la cual cada salida de la unidad básica es derivada de una combinación de las entradas de la red y del estado actual de la unidad (la entrada total).

función de sumación.

Función que combina las varias entradas de activación en una única activación.

grafo dirigido.

Representación de la variación y dirección del flujo para los elementos de procesamiento con respecto a otros elementos de procesamiento.

líneas de entrada.

En una red, son aquellas entradas de módulos de la red que no están conectadas a salidas modulares.

líneas de salida.

En una red, son aquellas líneas de salidas modulares que no están conectadas a entradas modulares.

lógica confusa.

Información incompleta o contradictoria.

maquina neural.

Hardware que modela el paralelismo de las neuronas (neurocomputers).

neurona.

Célula nerviosa cuyas características morfológicas y funcionales la hacen distinta a las demás células. Consta de un cuerpo celular llamado soma del que parten diversas prolongaciones llamadas dendritas.

netware.

Software que emula las neuronas y sus interconexiones en computadoras convencionales.

paradigma de la red.

Una arquitectura de red que especifica la estructura de interconexión de una red.

período de sumación latente.

Período de tiempo en que los bulbos terminales de las dendritas reciben impulsos, antes de ser emitido un impulso a lo largo del axón.

perceptrón.

Dispositivo de reconocimiento de formas capaz de aprender a reconocer los modelos de un conjunto después de un número finito de pruebas. Una larga clase de redes parecidas a una sola neurona con sólo un estrato de entrada y uno de salida.

perceptrón simple.

Perceptrón en el cual las unidades asociadoras no están conectadas entre sí, lo que significa que no posee memoria de corto plazo.

perceptrón acoplado.

Perceptrón en el cual existen conexiones entre las unidades asociadoras.

peso.

Fuerza de una conexión de entrada expresada por un número real. Elementos de procesamiento reciben datos por medio de las interconexiones. Cada interconexión tiene un peso asignado a ella. La suma de los pesos hace un valor que actualiza el elemento de procesamiento. El valor de salida de un elemento de procesamiento es descrito por un nivel de excitación que causa que las interconexiones estén tanto on (salida excitatoria) u off (salida inhibitoria).

receptor.

Aparato que recibe un estímulo.

red.

Interconexión eléctrica de puntos como las estaciones de energía, equipo de comunicaciones o terminales. También un grupo de computadoras que se comunican a través de líneas telefónicas o que emplean ondas de radio, infrarrojas o microondas.

red modular.

Colección de módulos, todos ellos con la misma escala de tiempos, interconectados a base de ramificar la salida de cada módulo en un cierto número de líneas y conectar algunas de ellas, o todas, a las entradas de otros módulos.

Sigmoid.

Función de activación.

sinapsis.

Puntos de cuasi-contacto, en los cuales los pequeños bulbos en que termina el axón, casi tocan las dendritas de otra neurona, transmitiendo así los impulsos nerviosos de una a otra neurona.

umbral.

Nivel mínimo de energía de excitación.

Introducción

Las redes neuronales son usadas diariamente para hacer predicciones e interpretar información en casi todos los campos explorados por el ser humano: investigación, medicina, ciencia, ingeniería, mercadeo, manufactura, administración, apuestas y otros. Al igual que las personas, las redes neuronales aprenden a través de la experiencia, no de la programación. Se caracterizan por su buen desempeño para reconocer patrones, generalizar y predecir tendencias; son rápidas, tolerantes a información incompleta o confusa y no necesitan fórmulas o reglas.

Una red neural es un tipo de sistema que simula muchas de las habilidades del cerebro humano. El cerebro está compuesto de billones de células llamadas neuronas, cada una de las cuales es una pequeña computadora. Estas neuronas se comunican entre sí y como resultado se obtiene el sistema más inteligente que hasta ahora se conoce ... el ser humano. Usando la misma arquitectura que el cerebro, pero en una escala menor, las redes artificiales trabajan en una forma similar.

Las habilidades de un humano, tales como el reconocimiento de patrones y tendencias, o bien, el aprendizaje a través de la experiencia, pueden ser ahora ejecutados en una computadora. Las redes neuronales combinan la habilidad del humano para analizar y aprender, con la habilidad de la computadora para procesar muchos datos en una forma fácil y rápida.

En este trabajo de tesis se presentan los principales conceptos concernientes a las redes neuronales. Se simplifican las explicaciones técnicas, describiéndolas en términos sencillos que permitan al lector descubrir las ventajas en la utilización de las redes neuronales.

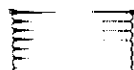
El capítulo I hace referencia a los estudios iniciales en el campo de las redes neuronales: los primeros modelos inspirados en el cerebro y ciertos conceptos en los cuales se fundamentan las redes neuronales.

El capítulo II explica la similitud de las redes neuronales con el cerebro humano, cómo operan las neuronas y cómo los estratos de neuronas interconectadas pueden aprender a resolver problemas.

En el capítulo III se clasifican las redes neuronales, tomando como parámetro los siguientes aspectos: el comportamiento de sus neuronas, el esquema de interconexión y el algoritmo de aprendizaje. También se describen las conclusiones de un estudio comparativo de los dos tipos de red más utilizados.

El capítulo IV describe las formas en las cuales las redes neuronales son utilizadas y producidas: "netware" y máquinas neuronales.

Finalmente, el capítulo V hace referencia al desempeño que tuvo el "netware" Brainmaker en la predicción y el pronóstico de ventas.



Objetivos

El objetivo principal de este trabajo de tesis es proporcionar información sobre el novedoso campo de las redes neuronales, exponiendo al lector de una forma sencilla el complicado funcionamiento de las redes neuronales artificiales.

Objetivos secundarios:

- Determinar los distintos componentes de las redes neuronales artificiales.
- Enumerar los aspectos concernientes al aprendizaje de las redes neuronales.
- Clasificar las redes neuronales artificiales, actualmente conocidas.
- Enumerar los trabajos de hardware y software que se han desarrollado para redes neuronales artificiales.
- Incluir un caso de estudio sobre el pronóstico de ventas.
 - Definir datos de entrada, patrones y datos de salida para la red neural.
 - Construir la red neural.
 - Reunir el conjunto de datos para entrenamiento, prueba y ejecución de la red.
 - Entrenar la red presentando información real.
 - Probar el aprendizaje de la red con ejemplos diferentes de los usados durante el entrenamiento.
 - Alimentar a la red con información del índice de precios al consumidor, para obtener un pronóstico.
 - Determinar las ventajas y desventajas de esta aplicación.
- Especificar recomendaciones y conclusiones para el uso de redes neuronales artificiales.

1. CEREBRO Y MODELOS

1.1 Introducción a la neurofisiología

El modelo matemático del cerebro se basa en la hipótesis fundamental que todo el funcionamiento del sistema nervioso está regido únicamente por el paso de impulsos eléctricos a través de células llamadas neuronas. En el sistema nervioso existen otros tipos de células, las cuales no son relevantes para el tema tratado por esta tesis; por eso, este documento se limita a una discusión de los tejidos neuronales.

Se considera al sistema nervioso como una vasta red de neuronas, distribuidas en estructuras de interconexiones extremadamente complejas. Esta red recibe entradas desde un gran número de receptores: los ojos, la piel, el olfato, etc., los cuales transforman los estímulos en patrones de impulsos eléctricos que proporcionan información a la red. Estos impulsos interactúan a su vez con los patrones de impulsos eléctricos que viajan por las neuronas y dan lugar a la emisión de impulsos que controlan a los efectores, tales como nuestros músculos y glándulas, para dar las respuestas. Así pues, se tiene un sistema nervioso de tres etapas: receptores, red neural y efectores el cual se ilustra en la figura 1.

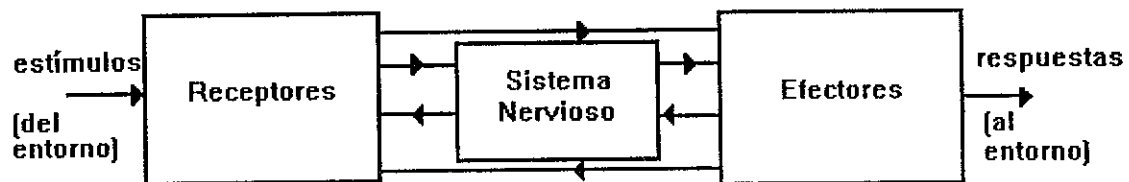


FIGURA 1
El sistema nervioso humano
considerado como un sistema de 3 etapas

Para formular un modelo de la red neural, es necesario formular primero uno de la neurona. Aunque las neuronas del sistema nervioso tienen multitud de formas, este estudio se basa en una representación neuronal como la figura 2.

La neurona tiene un núcleo contenido en el soma o cuerpo de la célula. Las dendritas son como unos filamentos muy finos, más delgados que el axón. El axón se asemeja a un cilindro estrecho que transporta los impulsos desde el soma a otras células y se divide en una delicada estructura en forma de árbol, cuyas ramas terminan en un pequeño bulbo que casi toca las dendritas de otra neurona. Estos puntos de cuasi-contacto se denominan sinapsis. Los impulsos que llegan a una sinapsis establecen señales eléctricas, las cuales tienen diferentes magnitudes y que se transmiten a otras neuronas.

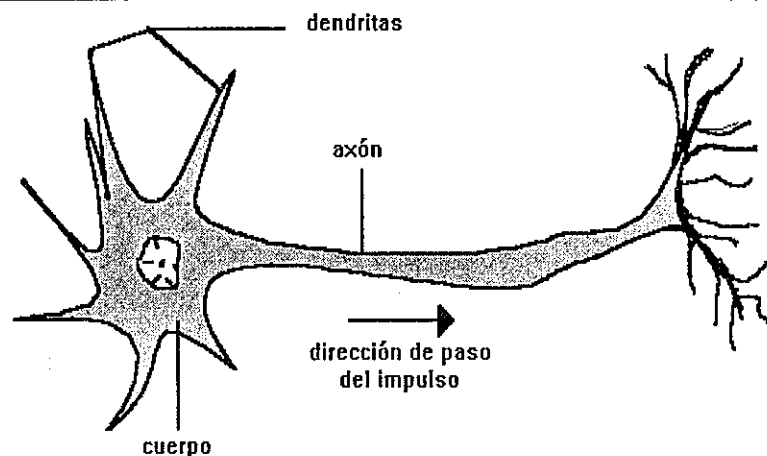


FIGURA 2
Dibujo esquemático de una neurona

Una determinada neurona sólo dispara un impulso eléctrico a lo largo de su axón si el número de impulsos que llegan a sus dendritas, durante un corto período de tiempo (denominado período de sumación latente), es suficiente. Estos impulsos pueden ayudar o impedir el disparo de un impulso y se llaman, respectivamente, excitadores e inhibidores. La condición para la activación de una neurona es que la excitación supere a la inhibición en una cantidad crítica, llamada umbral de la neurona. Si se asigna un peso positivo a cada sinapsis excitadora y un peso negativo a cada sinapsis inhibidora, se puede decir que:

"Una neurona se activa sólo si el peso total de las sinapsis que reciben impulsos durante el período de sumación latente es superior al valor del umbral."¹

El modelo de sumación lineal simple representa una gran simplificación del funcionamiento de una neurona y aunque el umbral es un parámetro que varía con el tiempo, en los modelos formales de neuronas rara vez se tiene en cuenta este hecho.

Entre un período de sumación latente y el paso del impulso hay un pequeño retardo, de modo que la llegada de los impulsos a las dendritas de una neurona no determina la activación de su axón, sino un corto período después. Luego que un impulso pasa por el axón, hay un tiempo (denominado período refractario), durante el cual el axón es incapaz de transmitir otro impulso; por lo tanto, durante un período refractario sólo un impulso puede atravesar el axón. Si se elige como unidad de tiempo el período refractario de la neurona, se puede determinar su comportamiento especificando si ha disparado durante los diferentes intervalos de tiempo t_i ($i=1,2,3,\dots$). Se llega así al supuesto que la neurona sólo puede dispararse en los instantes t_i y que la misma escala de tiempos puede ser utilizada para todas las neuronas de la red; es decir, el comportamiento de la red queda completamente determinado en los instantes t_i .

¹

Arbib, Michael A. **Cerebros, Máquinas y Matemáticas**. s.d.e. 19 pp.

1.2 Modelo de McCulloch-Pitts

Tomando en cuenta las simplificaciones mencionadas en la sección 1.1, se define el modelo de McCulloch-Pitts:

"Un módulo (o neurona formal) es un elemento con m entradas x_1, \dots, x_m ($m \geq 1$) y una sola salida d . Está caracterizado por $m+1$ números, su umbral θ y los pesos w_1, \dots, w_m , donde w_i está asociado a x_i . La neurona opera dentro de una escala discreta de tiempo t_i ($i=1,2,3,\dots$) y la activación de su salida en el instante $n+1$ queda determinada por la activación de sus entradas en el instante n de acuerdo con la siguiente regla: la neurona envía un impulso a lo largo de su axón en el instante $n+1$ si y sólo si el peso total de las entradas estimuladas en el instante n supera el valor del umbral de la neurona."²

DEFINICION 1.2.1

Si se introduce la siguiente notación:

$$\begin{aligned} m(t_i) &= 0 && m \text{ no dispara en el instante } t_i \\ m(t_i) &= 1 && m \text{ sí dispara en el instante } t_i \end{aligned}$$

(donde m puede ser una salida o una entrada neuronal), se puede expresar simbólicamente que la salida de la neurona en el instante $n+1$ es:

$$d(n+1) = 1 \quad \text{si y sólo si} \quad \sum w_i x_i(n) \geq \theta$$

REGLA 1.2.1

Debe observarse que los pesos positivos ($w_i > 0$) corresponden a entradas excitadoras, mientras que los pesos negativos ($w_i < 0$) corresponden a entradas inhibitorias.

Partiendo de este simple modelo de una neurona, se puede definir el primer modelo de red neural:

"Una red neural es una colección de neuronas, todas ellas con la misma escala de tiempos, interconectadas a base de ramificar la salida de cada neurona en una o más líneas y conectar algunas de ellas, o todas, a las entradas de otras neuronas. Así pues, una salida puede conducir a varias entradas, pero una entrada sólo puede proceder de una salida como máximo."³

DEFINICION 1.2.2

Las líneas de entrada de una red son aquellas que no vienen de otra neurona. Las líneas de salida de la red son aquellas que no están conectadas a otra neurona.

El concepto de red neural tiene una definición matemática precisa y es en este sentido matemático en el que se considera como un modelo del cerebro. Este modelo se basa en definiciones y axiomas, explicando así las propiedades conocidas del sistema original y prediciendo otras nuevas.

² Ibídem, 20-21 pp.

³ Ibídem, 21 pp.

El modelo fue construido asumiendo algunas simplificaciones, pero es necesario recordar que este primer modelo es sólo un punto de partida para el estudio de las redes neurales y no un fin en sí mismo. Sin embargo, todas estas simplificaciones no han inutilizado al modelo.

1.3 Autómatas finitos y redes neurales

A continuación se introduce el concepto de autómata finito y se demostrará que toda red neural es uno de ellos. El objetivo es probar que, recíprocamente, el comportamiento de entrada-salida de un autómata finito puede reproducirse siempre mediante una red neural apropiada. Puesto que es más fácil diseñar un autómata finito para un trabajo determinado, que la correspondiente red neural, su resultado explicará qué trabajos son capaces de realizar las redes neurales.

Sea la red neural N , con g neuronas, m líneas de entrada y n líneas de salida. Se dice que conocemos la entrada de la red cuando se sabe qué líneas están activadas (on) y cuáles no (off); así pues, hay 2^m combinaciones de entradas y 2^n combinaciones de salidas.

Se llama S al conjunto de los estados posibles, I al conjunto de entradas y O al de salidas de la red.

El disparo de una neurona en el instante $t+1$ se determina por las condiciones del instante t : el estado de la red, la entrada a la neurona y el patrón de activación de esta entrada. Esto significa que la entrada y el estado de la red, determinan la salida y el estado de la red en el instante $t+1$. Por lo tanto, se puede definir un autómata finito como una red neural, considerando un autómata como una caja negra que puede aceptar cualquiera de las 2^m entradas. Estas entradas tienen un número finito de estados internos y pueden producir cualquiera de las 2^n salidas. Matemáticamente se expresan estas condiciones de la siguiente forma:

Un autómata finito es una quintupla $A = (I, O, S, \delta, \gamma)$ donde

- I es un conjunto finito de entradas
- O es un conjunto finito de salidas
- S es un conjunto finito de estados internos

$\delta : S \times I \rightarrow S$ representa la función del estado siguiente
 $\gamma : S \times I \rightarrow O$ representa la función de la salida siguiente

DEFINICION 1.3.1

El autómata A trabaja según una escala discreta de tiempos, de forma que si en el instante t se encuentra en el estado q y recibe la entrada a , entonces, en el instante $t+1$ cambiará al estado (q,a) y emitirá la salida (q,a) .

De esta forma, se aclara el hecho que una red neural es un autómata finito y que cualquier autómata finito puede ser reemplazado por una red neural. Se puede afirmar también que el comportamiento de entrada-salida de un autómata finito puede sustituirse siempre por una restricción del comportamiento de entrada-salida de una red neural.

1.4 El perceptrón

En la sección 1.1 se dieron algunos conceptos introductorios a la neurofisiología y se obtuvo el primer modelo de cerebro. En la sección 1.2 se obtuvo el modelo de red neural o modelo de McCulloch-Pitts.

El grupo perceptrón de la Universidad Cornell (U.S.A.) ha trabajado en un modelo ligeramente diferente de red neural. La principal diferencia estriba en que ellos no han supuesto que la función de una neurona es constante en el tiempo. Por el contrario, suponen que los pesos de cada neurona cambian, con el fin de conseguir que la red de neuronas se modifique y aprenda.

Un perceptrón es similar a un dispositivo de reconocimiento de formas que no está construido para reconocer un conjunto específico de modelos, sino que es capaz de aprender a reconocer ciertos modelos después de un número finito de pruebas.

El patrón o modelo incide en el perceptrón sobre una retina de unidades sensoriales (p.e. fotocélulas). Aunque es posible codificar cualquier entrada sensorial de forma que sirva de entrada a un banco de estas unidades, lo más natural (como sugiere la palabra retina) es concebir el modelo como una entrada visual de luz y sombra. Una fotocélula que recibe una porción relativamente iluminada del modelo es activada, mientras que si recibe una porción relativamente oscura, no es activada.

La retina, compuesta de unidades sensoriales, tiene como función fundamental la transformación de la imagen mediante el área de proyección. Estas unidades sensoriales están conectadas a unidades asociadoras (neuronas formales), que a su vez pueden conectarse a otras del mismo tipo, o bien, a unidades de respuesta.

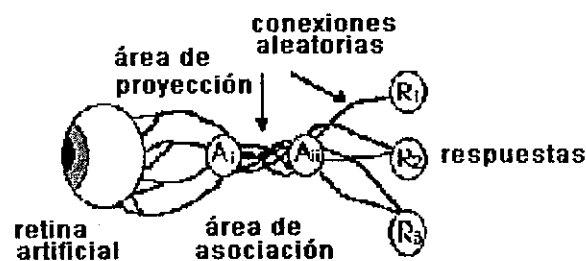


FIGURA 3
Esquema de un perceptrón

La retina constituye los receptores del perceptrón, las unidades asociadoras comprenden al propio sistema nervioso y las unidades de respuesta son los efectores. Cuando un estímulo actúa sobre la retina del perceptrón, los impulsos son conducidos desde las unidades sensoriales activadas hasta las unidades asociadoras; si la señal total que llega a la unidad asociadora es superior a su umbral, entonces el asociador se activa y envía impulsos a las unidades de respuesta a las que está conectado.

El perceptrón es un conglomerado de datos neurofisiológicos muy simplificados, que actúa sobre un sistema nervioso con receptores exclusivamente visuales. Sin embargo, el grupo perceptrón ha ido más lejos y las propiedades adicionales de la red merecen ser mencionadas.

1.4.1 Reglas de refuerzo

Los seres humanos tenemos dos clases de memoria: de corto plazo y de largo plazo. Una idea debe ser retenida en la memoria de corto plazo, durante un buen rato, antes de ser transferida a la de largo plazo. El tiempo necesario para esta transferencia varía según las estimaciones: una de ellas lo calcula en 20 minutos. Al entrar en estado de coma, los recuerdos de los 20 minutos anteriores se pierden para siempre, es decir, son transferidos a la memoria de largo plazo.

El tipo de memoria que se proporciona a la red neural es de corto plazo, ésto, mediante el paso de complicados patrones de impulsos eléctricos a través de la red. Si tal actividad transitoria persiste el tiempo suficiente, acaba por modificar la red.

Uno de los mecanismos que se han postulado para la memoria de largo plazo consiste en la formación de proteínas específicas dentro de las neuronas, cambiando así sus umbrales en respuesta a los patrones de memoria de corto plazo. Otro mecanismo consiste en que los bulbos terminales aumentan con el uso repetido, incrementando el peso de la correspondiente entrada sináptica, lo cual haría más fácil restablecer los patrones de impulso usando esa sinapsis y como consecuencia acceder a la correspondiente memoria. El mecanismo exacto no es conocido todavía, puesto que la fisiología, la histología y la anatomía no han dado aún un veredicto.

El perceptrón está equipado con una memoria de largo plazo, lo cual se ha conseguido cambiando los pesos de las neuronas o, para decirlo en el lenguaje utilizado por el grupo perceptrón, cambiando la magnitud del impulso transmitido por una conexión. Estos cambios dependen de la actividad transmitida por dicha conexión. La regla que rige esta dependencia se llama regla de refuerzo, porque está pensada para reforzar las respuestas correctas del perceptrón a los estímulos que le son presentados. Siendo la evidencia fisiológica tan oscura, el grupo perceptrón eligió reglas de refuerzo, teórica o experimentalmente, más convenientes. Tal elección ha permitido al perceptrón exhibir ciertos aspectos de aprendizaje.

1.4.2 Conexiones aleatorias

El grupo perceptrón tuvo que decidir cómo conectar las unidades sensoriales a las unidades asociadoras. Un primer intento consistió en hacer conexiones múltiples y aleatorias. Más tarde, introdujeron restricciones dentro de la red, es decir, subredes para reconocimiento de líneas.

La red neural presentaba cierta capacidad de memoria y cómputo, a pesar de las grandes simplificaciones hechas en su obtención. Los primeros perceptrones exhibían propiedades de aprendizaje sencillas.

El grupo perceptrón ha seguido tres modos principales de investigación: análisis matemático, simulación mediante un computador digital y construcción de una máquina real. Cada uno de ellos tiene sus propias ventajas. Un resultado importante de la utilización de una máquina real es que ni la precisión ni la fiabilidad de los componentes son factores importantes y que las conexiones no tienen por qué ser precisas; otro resultado interesante es que el perceptrón puede aprender, a pesar que el instructor cometa errores.

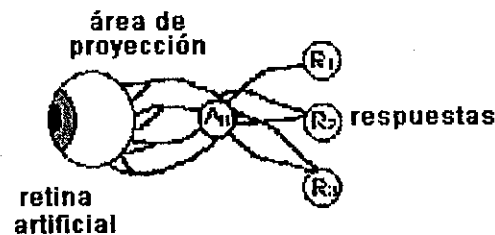


FIGURA 4
Esquema de un perceptrón sencillo

Un perceptrón acoplado es aquél en el que las unidades asociadoras están conectadas entre sí. Si las unidades no están conectadas entre sí, significa que no posee memoria de corto plazo y se denomina perceptrón sencillo.

El perceptrón Mark I, una encarnación en hardware de un perceptrón simple, tiene una retina de 20*20 fotocélulas. Si se considera el caso de restringir los estímulos a las 26 letras del alfabeto, cada una en una posición estándar (e.d. hay sólo 26 estímulos) y se toma la salida de cinco elementos de respuesta binaria ($2^5 = 32 > 26$). Según los autores de este hardware, en uno de los experimentos la máquina aprendió a identificarlas correctamente después de 15 pruebas con cada letra (en total, 390 pruebas). Ahora bien, a un sistema de reconocimiento de modelos se le exige más que la simple distinción de estímulos en posición estándar. Lo que se quiere es una máquina que pueda reconocer cada letra en cualquier posición en la retina, ya esté ligeramente ladeada, distorsionada o expuesta sobre un fondo de puntos. Tomando en cuenta estas consideraciones, el número de estímulos crece alarmantemente.

En la figura 5 se representa la letra E, tal como podría ser percibida por las unidades sensitivas de un perceptrón.

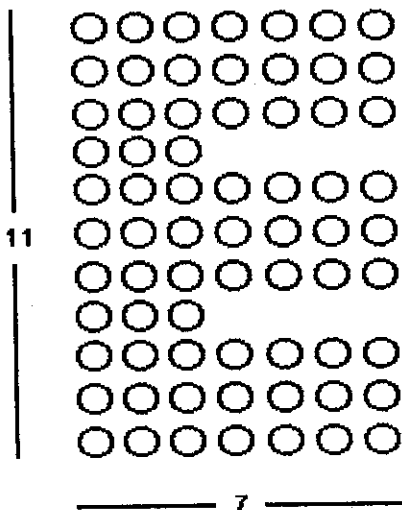


FIGURA 5
Letra E percibida por un perceptrón

El perceptrón simple es demasiado sencillo para ser utilizado en ciertas aplicaciones. Sin embargo, el grupo perceptrón ha realizado trabajos más complicados. A medida que establecían redes más elaboradas, construían más estructuras dentro de ellas. Los trabajos en curso incluyen investigaciones neuroquímicas sobre la memoria y la formulación de modelos concomitantes.

Los trabajos relativos al perceptrón han recibido críticas muy adversas. Las primeras publicaciones del grupo perceptrón se caracterizan por ser muy pretenciosas y deben leerse con precaución, agregando que muchos investigadores se vieron perjudicados por culpa de una prensa desafortunada.

Rosenblatt inició el estudio del perceptrón, al cual se hace referencia ahora como una red neural de un solo estrato; luego Minsky y Papert lo continuaron en 1969 cuando demostraron, en su libro *Perceptrons*, que el perceptrón no presentaba ningún interés práctico. Luego de alrededor de 20 años, con la percepción postperceptrón, se retomó el perceptrón pero agregándole estratos y completando

significativamente la capacidad de aprendizaje y de expansión en la solución de problemas durante la década de los ochenta. Esto fue vigorizado por desarrollos importantes realizados por Hopfield, Barto y, más recientemente, por Sejnowski (ésto será examinado con mayor precisión en el capítulo 2).

Es así como el estudio del perceptrón ha aportado importantes conocimientos en el desarrollo de la teoría de redes neurales.

1.5 Retroalimentación

La palabra cibernética fue acuñada en 1947 por Norbert Wiener y sus colegas para denominar el estudio (comparativo) del control y la comunicación en el animal y en la máquina. En este sentido, todo lo antes expuesto puede incluirse bajo esta denominación.

Hasta ahora se ha tratado exclusivamente lo relativo a la estructura del propio sistema nervioso (las redes neurales de McCulloch-Pitts y el perceptrón). Sin embargo, no se ha tratado cómo interacciona el cerebro con los efectores. Si a un autómata finito se le da el poder de comunicarse con el medio que le rodea, se le darán también nuevas e importantes propiedades. Así pues, si se extiende la visión del sistema nervioso para dar cabida a las interacciones efectores-receptores, se podrá llegar a ideas nuevas e importantes. Con este fin, ahora se expondrá la relación efectores --> receptores.

A la red nerviosa se le puede enviar (retroalimentar) información acerca de la eficacia con que está controlando la actividad de los efectores. A continuación se describen dos ejemplos ilustrativos de esta situación:

1.5.1. Si quiero tomar un lápiz, muevo una mano hacia él. Mis ojos me dicen qué distancia tiene que recorrer la mano. El cerebro está continuamente recibiendo información de mis receptores (los ojos) para comunicarme la posición de mi efector (la mano), con lo cual puede computar las instrucciones apropiadas para ordenar a los músculos del brazo que muevan mi mano de tal forma que reduzcan la diferencia entre la posición actual de mi mano y la deseada.

1.5.2. Al caminar, levanto un pie y luego lo dejo caer. Controlo las flexiones musculares de mi pierna para cambiar de pie cuando la presión ejercida sobre el pie por el suelo excede de un valor crítico. Así pues, la posición de mi efector (pie) con respecto al suelo es retroalimentada al cerebro por mis receptores (los nervios sensibles a la presión). De nuevo es una diferencia, entre la posición de mi pie y el suelo, la que determina las instrucciones que mi cerebro ordena a los músculos de mi pierna. La importancia de la retroalimentación desde los receptores de presión de los pies es ilustrada por la siguiente experiencia: después de permanecer sentados con las piernas cruzadas durante mucho tiempo, al levantarnos comprobamos que se nos han dormido y que tenemos que caminar sin utilizar la retroalimentación de la presión; como resultado de ello, los movimientos de la pierna son torpes y faltos de coordinación.

La conclusión que obtenemos de estos dos ejemplos es que la relación efectores --> receptores juega un papel crucial en la determinación de nuestras respuestas. El concepto de retroalimentación debe desempeñar un papel esencial en el estudio de los cerebros y las máquinas, donde se dice que un organismo o una máquina tiene retroalimentación si su actividad está controlada, hasta cierto punto, por una comparación entre su estado real y un cierto estado preestablecido. En particular, como se demostró en el ejemplo 1.5.1, interesa la retroalimentación negativa, en la cual la máquina trata de disminuir la diferencia entre el estado real y el deseado.

Un ejemplo muy sencillo de retroalimentación negativa en las máquinas es el termostato, que sirve para disminuir la diferencia entre el estado real (la temperatura de la habitación) y el estado deseado (el ajuste de la temperatura en el control del termostato) controlando adecuadamente la producción de calor de una caldera.

El diagrama para un sistema sencillo de retroalimentación negativa es ilustrado por la figura 6.

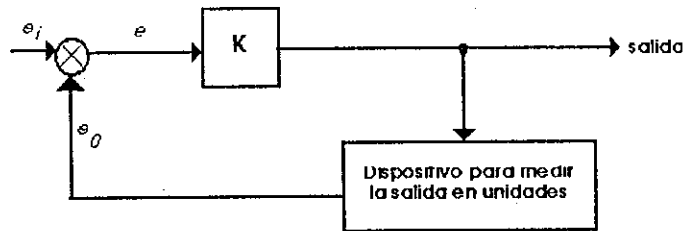


FIGURA 6
Sistema de retroalimentación negativa

Continuamente se introduce al sistema una cantidad i que indica la medida de la salida deseada. Se considera ahora la salida real o y se retroalimenta al detector de errores, el cual encuentra la diferencia entre los estados real y deseado, calculando la señal de error.

$$e = i - o$$

ECUACION 1.5.1

Esta señal de error es la que controla realmente el sistema K y determina su salida. Para una situación determinada, el problema consiste en diseñar la caja negra K de forma que la señal de error 'e' tienda a cero.

Un sistema de retroalimentación negativa del tipo dado en la figura 6 también se llama servomecanismo. Otro ejemplo de servomecanismo lo constituye el timón de un barco, ejemplo que proporciona la etimología de la palabra cibernética, porque en griego se traduce como timonel.

Las nociones adquiridas sobre la retroalimentación proporcionan datos sobre una enfermedad del sistema nervioso llamada ataxia. A continuación se dan dos breves ejemplos:

1.5.1 Un paciente gravemente incapacitado para caminar. Camina con paso incierto, mirando al suelo y a sus piernas. Si se le vendan los ojos no puede mantenerse en pie y cae al suelo.

1.5.2 Otro paciente al estar sentado parece normal. Sin embargo, al sostener un cigarrillo balancea la mano; si fuese un vaso lo vaciaría antes de poder beber su contenido.

Los pacientes de estos ejemplos no tienen atrofiaciones musculares, es más, son fuertes y sanos pero incapaces de organizar sus acciones. El primer paciente con ataxia es causada por la pérdida de retroalimentación. El segundo, que también sufre de ataxia, es originada por un K inestable.

1.6 Modelo de Peter H. Greene

Este modelo es radicalmente diferente al modelo de McCulloch-Pitts. Es un modelo en el cual las frecuencias juegan un papel muy importante. Fue presentado por Peter H. Greene en 1962, en Chicago, en una conferencia sobre sistemas auto-organizados bajo el título **On the Representation of Information by Neural Nets Models**.

Hasta ahora se ha supuesto que una neurona no se puede activar más de una vez en cada período refractario; no obstante, es posible conseguir cualquier frecuencia de impulsos, ésto si y sólo si el período entre impulsos sucesivos excede del período refractario. Así pues, se modifica una acción refleja, mediante la frecuencia de impulsos activadores a lo largo de los axones de los nervios sensibles del receptor.

Greene propone un modelo de actividad neural de un caballo, basado en algunas de las propiedades de los sistemas lineales:

Supone que las entradas sensoriales activan sistemas lineales, los cuales se supone que están precedidos, seguidos y controlados por sistemas que no tienen que ser lineales y sobre los cuales no se hace ninguna suposición. Se supone que las redes lineales están tanto en los mecanismos sensoriales como en los motrices. Los componentes de los vectores de salida representan niveles de excitación en distintos lugares de la red y estas excitaciones controlan las actividades de los numerosos nervios y músculos que dirigen el comportamiento del animal. Se supone que los sistemas, sensorial y motor, se han desarrollado de tal forma que algunas de las entradas sensoriales excitan modos naturales de las redes neurales y que estos modos constituyen algunos de los elementos importantes de la información sensorial, cuya excitación es una condición para la excitación de los modos normales de las redes motoras. Los modos motores constituyen patrones de acción de la lista de rasgos del comportamiento.

Se desea averiguar cómo son controladas las cuatro formas de caminar del caballo (al paso, trotar, galopar, etc.). Para cada forma se busca un sistema de neuronas que haga funcionar a los cuatro músculos al ritmo prescrito. Para cada uno de ellos cabría esperar que la pata A estuviese conectada a la pata B, que la retroalimentación de ambas inhiba a la pata C y que ésta esté conectada por medio de un determinado número de retardos sinápticos a D, o algún esquema parecido. Luego se busca algún elemento conector que seleccione la red que controla las extremidades en un momento particular. Los cuatro sistemas quizás no estuviesen físicamente separados; un ajuste de umbrales podría permitir a una misma red ejecutar más de un ritmo. En una red complicada resulta muy difícil identificar todas las estructuras y ajustes, teniéndolos que correlacionar con los detalles del comportamiento exhibido, pero, en principio, eso sería entender completamente el sistema nervioso del ser. Este modelo de animal tiene un solo sistema de cuatro neuronas y vías de interconexión, no cuatro sistemas con un elemento selector, careciendo de partes que puedan interpretarse como conexiones inhibitorias, unidades de retardo, etc., en analogía con los rasgos de su comportamiento exhibido, que cuando camina lentamente va al paso, cuando va aprisa trote, cuando va aún más aprisa galopa, etc.

Para concluir con esta sección, algunas observaciones sobre el modelo son realizadas. La primera de ellas se refiere a que no es que el cerebro sea una red lineal (de hecho, cuanto más se estudia de sus características, menos lineal parece), sino que las frecuencias de resonancia en el cerebro pueden jugar un papel importante en el comportamiento del ser; la elección de una red lineal sirvió para simplificar la descripción matemática de Green de las frecuencias de resonancia relevantes.

La segunda observación es de vital importancia y es que, independientemente de lo objetivo que pueda ser un fisiólogo o un psicólogo, su elección de los experimentos y de las formas de experimentación está siempre presidida por alguna hipótesis o modelo. Es necesario, entonces, estar muy alerta para descubrir defectos en los antiguos modelos e ideas que conduzcan a otros nuevos, sean matemáticos o no. El modelo de Greene es uno de ellos, ésto se puede constatar pues los avances sobre el modelo no tienen relevancia; sin embargo, ha formado parte del desarrollo de la teoría general de las redes neurales y por ello ha sido expuesto.

1.7 Homeóstasis

El estudio de la retroalimentación en el sistema nervioso humano ha influido enormemente sobre el diseño de miembros artificiales.

El tipo más antiguo de miembro artificial, la pata de palo, suple sólo una de las funciones de la pierna perdida: brindar apoyo. En secciones anteriores se ha mostrado la importancia de la retroalimentación para una utilización completa de nuestros miembros.

Existen importantes aplicaciones fisiológicas del principio de la retroalimentación que no se relacionan con el sistema nervioso. Un gran número de casos en los cuales la retroalimentación no sólo se manifiesta en los fenómenos fisiológicos sino que es absolutamente esencial para la continuación de la vida, se agrupan bajo el nombre de homeóstasis.

La homeóstasis se da en un sistema (fisiológico) si, al introducir una alteración moderada que tiende a desplazar el sistema de sus valores normales, sus partes reaccionan e interaccionan de tal forma que los efectos perjudiciales de la alteración quedan reducidos. A continuación se dan dos ejemplos breves de homeóstasis:

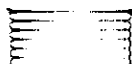
1.7.1 Cuando una persona tiene frío, éste estimula un mecanismo en la base del cerebro que le hace tiritar. La actividad muscular genera calor, contrarrestando el frío.

1.7.2 Una hemorragia repentina origina una bajada drástica de la presión sanguínea. La baja presión hace que las arterias se compriman, disminuyendo así la hemorragia.

Lars Løfgren⁴ ha hecho un estudio teórico de redes con elementos lógicos, semejantes a las neuronas formales, que pueden detectar y sustituir elementos defectuosos. Løfgren encontró que el máximo de vida para un autómata bien localizado (un autómata cuyo crecimiento, si es que lo hay, tiene que suceder siempre dentro de un volumen de espacio fijo), se obtiene si el autómata contiene una estructura para la localización de errores. Este autor consideró el problema de la localización de errores aún cuando éstos puedan ocurrir en los elementos de localización, determinando que el máximo de vida es finito. La vida del autómata completo puede ser mucho mayor que la de sus componentes.

En un estudio posterior Løfgren elimina la condición que el autómata esté bien localizado y halla que un autómata no bien localizado y autorreparador puede tener vida ilimitada si crece y se repara él mismo de una forma adecuada. En su esquema, tales autómatas contienen subautómatas que se reproducen por sí mismo. Løfgren da así una condición suficiente para la inmortalidad en términos de la probabilidad de error, la complejidad de los subautómatas autorreproductores y la forma geométrica del autómata autorreparador durante su expansión.

Subraya Løfgren también, que la vida finita de los autómatas bien localizados y autorreparadores se puede comparar a la de cualquier planta o animal en la naturaleza. La vida ilimitada y en continua expansión, de los autómatas no bien localizados y autorreparadores, puede compararse a la vida ilimitada de la población humana, considerada como una unidad. Los resultados de Løfgren parecen implicar que esta población tendría que extenderse más allá de los límites terrestres para poder tener una vida ilimitada.



2. TEORÍA GENERAL DE REDES NEURALES

2.1 Introducción

Este capítulo explica la similitud de las redes neurales artificiales con el cerebro humano, cómo operan las neuronas y cómo los estratos de neuronas interconectadas pueden aprender a resolver problemas.

Las redes neurales pueden hacer algunas cosas impresionantes, pero no pueden replicar todos los aspectos del cerebro humano; el cerebro es muy complejo para ser simulado aún por una supercomputadora. Las redes neurales pueden ser utilizadas para simular ciertas actividades o funciones del cerebro. Existen dos tipos de simulación que una red neural puede hacer: modelar procesos del cerebro y modelar las capacidades del cerebro.

El propósito de modelar 'procesos' del cerebro es probar nuevas teorías sobre el funcionamiento cerebral. Este modelo depende en un alto grado de la precisión en el modelo de la neurona individual. Por ejemplo, el cerebro tiene la capacidad de reconocer un diálogo, pero su habilidad para hacerlo depende de su temperatura. Si la temperatura del cerebro baja de 90 grados o sube de 105 grados, pierde un porcentaje en su habilidad de reconocimiento. Por lo tanto, una red neural que modela procesos del cerebro debe incluir un factor de temperatura.

El propósito de modelar las 'capacidades' del cerebro es ejecutar las funciones cerebrales, pero no necesariamente en la misma forma que lo hace el cerebro. Así, una red neural que es usada para reconocimiento de diálogo es diseñada sin el factor de temperatura y las interconexiones de las neuronas, o el método de aprendizaje, pueden simplificarse.

La mayoría de las redes neurales artificiales deben ser clasificadas como modelos de las capacidades cerebrales, ya que modelan las capacidades del cerebro y no precisamente su funcionamiento. Todas las redes neurales artificiales modelan alguna estructura biológica o función, por eso se hace necesario aprender acerca del cerebro humano antes de iniciar la teoría de las redes.

2.2 Biológicamente inspiradas

Tanto las redes neurales biológicas como las artificiales contienen neuronas, reales o simuladas, respectivamente. Estas neuronas tienen muchas conexiones a las otras neuronas, mediante las cuales se transfiere información. El conocimiento de una red está distribuido a través de las interconexiones de las neuronas, no como bits de inteligencia almacenados en las neuronas como se pensaba.

Las neuronas artificiales también son llamadas elementos de procesamiento, nodos, unidades o células. Cada neurona recibe las salidas de muchas otras neuronas. Una neurona obtiene su propia salida calculando la sumatoria de sus entradas y pesos, generando un nivel de activación y pasándolo a través de una salida o función de transferencia. El punto en el cual dos neuronas se comunican es llamado conexión (análogo a sinapsis). La fuerza de la conexión entre dos neuronas es llamado peso. La colección de pesos arreglada en filas y columnas es llamada matriz de pesos.

Una neurona artificial emula los axones y las dendritas con cables y emula las sinapsis usando resistores con valores de peso. Las neuronas interactúan entre sí, dependiendo de cómo están interconectadas: completamente (opuesto a parcialmente), o bien, con o sin un lazo de retroalimentación. La naturaleza de estos lazos de retroalimentación determina la capacidad de entrenamiento de la red; el grado de interconexión determina su paralelismo.

Mientras que la memoria de una computadora digital está medida en bytes, una memoria de red neural es juzgada por sus interconexiones. De forma similar, mientras la velocidad de una computadora digital es expresada en instrucciones por segundo, la velocidad de la red neural es medida en interconexiones por segundo.

Muchos procesos de percepción toman a los humanos unos pocos cientos de milisegundos, mientras que las neuronas individuales en el cerebro humano computan operaciones a una tasa tan lenta como una instrucción de una computadora digital lo hace. El cerebro ejecuta su hazaña de procesamiento a través de paralelismo masivo, usando 10 billones de neuronas y más de 1000 veces ese número de interconexiones.

2.3 Neuronas

Una neurona en el cerebro tiene cuatro partes básicas: el cuerpo, los canales de entrada, el canal de salida y los puntos de conexión entre las neuronas, que son llamados sinapsis. La neurona recibe muchas señales de otras neuronas en las sinapsis. Luego que la señal es recibida en el canal de entrada, se procesa antes de ser enviada al cuerpo de la célula.

La sinapsis asigna pesos a las señales de entrada para que cada una de ellas tenga un efecto diferente sobre la neurona. Una sinapsis puede apagar o encender el volumen de una señal para que tenga un efecto más fuerte o más débil en la neurona receptora que otras señales. Una sinapsis puede causar una señal que tienda a encender (excitar) o apagar (inhibir) la neurona. Una neurona altamente excitada envía una señal de salida, una neurona inhibida no lo hace.

El trabajo del cuerpo de la neurona es sumar todas las señales que entran y decidir si el total es suficiente para enviar una salida. Cada neurona detecta y envía una salida solamente sobre cierta información (oscuridad, claridad, frío, calor, etc.); no envía muchas salidas o reconoce diferente información. La sinapsis permite que la salida tenga diferentes efectos sobre las neuronas receptoras, por ejemplo, en un área del cerebro la oscuridad puede ser deseada (en el centro del sueño) y en otro puede ser indeseable (centro de reconocimiento de imágenes).

2.4 Conexiones

Una conexión es una línea única de comunicación que va de una neurona que envía información, hacia otra que la recibe. Existen dos tipos posibles de conexión: excitadora e inhibidora. Las conexiones inhibitorias previenen que la neurona se dispare. Las conexiones excitatorias causan el disparo de la neurona.

La estructura de la red puede incluir conexiones inhibidoras de una neurona al resto de neuronas en el mismo estrato, éste recibe el nombre de inhibición lateral. Algunas veces una red tiene tanta inhibición lateral que sólo una neurona del estrato, comúnmente el estrato de salida, puede estar activa en un período determinado. Este tipo de efecto, el de minimizar el número de neuronas activas, es un tipo de competencia.

La forma en que las neuronas están conectadas a las demás tiene una gran incidencia en la capacidad de entrenamiento de la red. El tipo de conexión determina el procesamiento que se llevará a cabo.

Una red neural artificial, construida con la tecnología actual, tiene muy pocas conexiones comparado con el número existente en el cerebro. El cerebro humano tiene alrededor de 10 billones de neuronas y más de 1000 veces ese número de interconexiones. Tal parece que el cerebro también tiene



muchas redes neurales pequeñas que se encargan de resolver porciones de un problema y redes de mayor nivel encargadas de unir estas porciones.

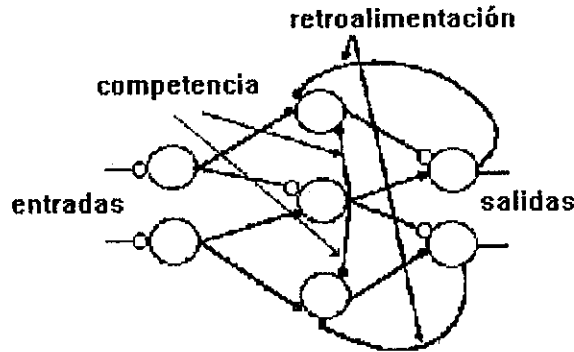


FIGURA 7
Red con retroalimentación y competencia

2.5 Estratos

Una red neural consiste de estratos de neuronas que están conectados entre sí. Los detalles de cómo las neuronas se interconectan representa una de las selecciones más importantes cuando se construye una red neural. Algunas de las neuronas se usan para comunicarse con nosotros, el mundo exterior y otras se comunican sólo con ciertas neuronas, éstas se llaman neuronas escondidas.

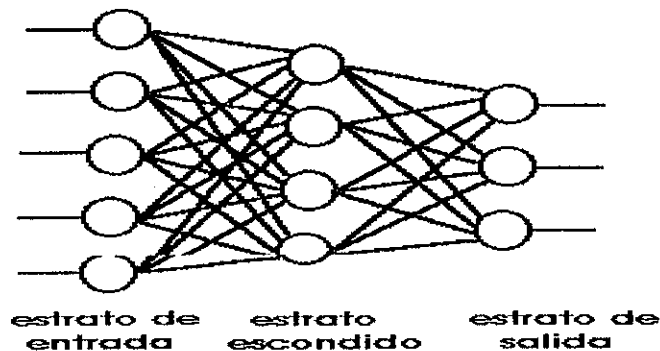


FIGURA 8
Red neural simple

Las neuronas se localizan en alguno de los tres tipos de lugares: estrato de entrada, estrato de salida o estratos escondidos. Las neuronas de la entrada reciben datos del mundo exterior (p.e., teclado, archivo de datos, etc.) y envían información a las neuronas del estrato escondido.

Las neuronas del estrato escondido son todas aquellas que se localizan entre el estrato de entrada y el estrato de salida. En el exterior no miramos sus entradas o salidas porque se conectan únicamente a otra neurona. Estas neuronas son parte del patrón interno y abstracto que representa la solución del problema mediante el uso de la red neural.

Las neuronas del estrato de salida son las que proveen las respuestas al mundo exterior (p.e., monitor, impresora, archivo de datos, etc.)

Dos tipos de representaciones deben tomarse en cuenta en el estudio de las neuronas escondidas: primero, debe entenderse lo que significan los pesos; segundo, los patrones de activación de estas neuronas en respuesta a entradas particulares.

El patrón de actividad en el estrato escondido es una codificación de lo que la red piensa que son los rasgos significantes de la entrada. Las dos representaciones, pesos y patrones de actividad, están muy relacionados.

Las neuronas escondidas actúan como detectores de rasgos, o filtros, para algunos tipos de entradas. Combinando estos rasgos (es decir, agregando estratos escondidos), la neurona de salida puede ejecutar más clasificaciones.

En general, agregar estratos escondidos le permite a las redes neurales artificiales particionar eficientemente el espacio de entrada en regiones arbitrarias, ejecutando así, tareas más complejas. Las redes con un sólo estrato se hicieron populares con Frank Rosenblatt, perdiendo popularidad con Marvin Minsky y Seymour Papert.

A la medida que el número de neuronas escondidas se aproxima al número de ejemplos, el peligro de la memorización se incrementa. Si existen muy pocas neuronas escondidas, la red no puede aprender suficiente. Desafortunadamente, no existe una fórmula precisa para determinar el número ideal de neuronas escondidas necesarias para una aplicación.

Existen muchas formas de determinar un buen número de neuronas escondidas. Una solución es entrenar varias redes con diferentes números de neuronas escondidas y seleccionar aquella que obtenga los mejores resultados. Una segunda solución es empezar con un número pequeño de neuronas y agregar más durante el entrenamiento si la red no está aprendiendo.

Cuando una red neural está dando problemas en el aprendizaje debido a insuficiente número de neuronas escondidas, solamente encontrará solución correcta a ciertas cuestiones y dará problemas al aprender más. Si hay demasiadas neuronas, tomará un gran número de cuestiones correctas, pero seguirá sin aprender suficiente. En ambos casos, continuar el entrenamiento no ayuda porque en cierto momento la red dejará de mejorar su aprendizaje. Si esto sucede y la persona que utiliza la red está segura que los datos son correctos, probablemente no se tienen suficientes neuronas escondidas.

Una mejor solución a este problema es empezar con un pequeño número de neuronas escondidas y agregar neuronas durante el entrenamiento, si son necesarias. El número inicial de neuronas escondidas puede ser:

- menos de la mitad del número de entradas y salidas
- del 5% al 10% del número de los casos de entrenamiento
- el producto de la tolerancia por el cociente del número de casos de entrenamiento y el número de neuronas de entrada

$$N = (\#_casos_entrenamiento / \#_neuronas_entrada) * tolerancia$$

ECUACION 2.5.1

Algunos investigadores despliegan los pesos de una única unidad escondida cuando tratan de analizar la red. El trabajo de Lang y Witbrock⁵, ambos de Mellon University, muestra que para problemas geométricos puede ser de más ayuda desplegar la respuesta de la neurona en un ejemplo sistemático de puntos en la región de entrada, especialmente cuando la red tiene más de un estrato escondido.

Esta práctica es también común en investigaciones de neurociencia clásica del sistema visual. No se pueden medir los pesos entre las neuronas vivientes de la corteza del cerebro, pero se pueden medir sus respuestas a varias entradas. Muchos estudios del sistema visual han sido realizados graficando la tasa de neuronas corticales que se descargan mientras varía un patrón de estímulo presentado a la retina.

2.6 Red neural

Existen muchas definiciones de lo que es una red neural, a continuación se muestran algunas de ellas:

"Una nueva forma de computación, inspirada en modelos biológicos."⁶

"Un modelo matemático compuesto por un gran número de elementos procesales organizados en niveles."⁷

"...un sistema de computación hecho por un gran número de elementos simples, elementos de proceso muy interconectados, los cuales procesan información por medio de su estado dinámico como respuesta a entradas externas."⁸

"Redes neuronales artificiales son redes interconectadas masivamente en paralelo de elementos simples (usualmente adaptativos) y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico."⁹

En términos generales y con palabras sencillas, una red neural es una colección de nodos conectados entre sí en la forma de grafo dirigido, organizado de tal modo que la estructura de la red sea la adecuada para solucionar el problema que se esté considerando.

2.7 Modelo de neurona

El modelo de neurona es el componente más básico de una red neural. Cierta número de modelos han sido propuestos para describir el comportamiento de las neuronas en los sistemas nerviosos actuales. Es importante distinguir entre dos clases básicas de modelos.

-
- 5 Lang, K.J. y M.J. Witbrock. **Learning to Tell Two Spirals Apart**. 1988
 - 6 Hilera, José R. y Víctor J. Martínez. **Redes Neuronales Artificiales. Fundamentos, Modelos y Aplicaciones**. Addison-Wesley Iberoamericana, 1995. 9 pp.
 - 7 Ibídem, 9 pp.
 - 8 Hecht-Nielsen, R. **Neurocomputing: Picking the Human Brain**. IEEE Spectrum, No. 25, marzo 1988. 36-41 pp.
 - 9 Kohonen, T. **An Introduction to Neural Computing**. Neural Networks, vol.1, 1988. 3-16 pp.

- Modelos detallados

Esta clase de modelos representa el comportamiento de las células nerviosas en el más grande detalle matemático, usualmente envuelven sistemas de ecuaciones diferenciales.

- Neuromimos

Asume un número de simplificaciones. Estos modelos son diseñados específicamente para implementaciones sencillas. Capturan la esencia del comportamiento de la neurona actual, mientras mantienen requerimientos de computación simples.

La mayoría de modelos de red neural usan neuromimos en lugar de modelos detallados.

La figura 9 muestra un ejemplo del modelo de neurona simplificado. La neurona recibe un conjunto de señales de entrada x_i , cada una de las cuales es multiplicada por un factor de peso w_i . Todas las señales de entrada multiplicadas por el factor de peso son sumadas para producir n , la entrada de la red a la neurona. Una función de activación F transforma la entrada de la red a la neurona en una señal de salida que es transmitida a otras neuronas.

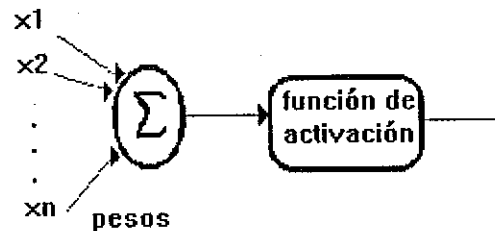


FIGURA 9
Modelo de Neurona Simplificado

2.8 Modelo de red

En este modelo las neuronas son arregladas en estratos y el número de neuronas en cada estrato puede ser diferente (ver figura 8).

Un estrato de entrada recibe el conjunto de entradas únicas de la red y las transfiere. La transferencia la realiza directamente a través de un conjunto de conexiones de peso, las cuales se encuentran en el medio o estrato escondido. Cada neurona en el estrato escondido opera sobre sus propias señales de entrada y pasa un valor de salida a través de un segundo estrato de pesos para cada neurona en el estrato de salida. El estrato de salida, entonces, ejecuta una última sumatoria y transformación. Finalmente, este último estrato produce la salida, la cual es la respuesta al mundo exterior. El significado de la respuesta, proveniente de cada neurona del estrato de salida, se interpreta de la forma en que se halla definido la red inicialmente.

Las funciones de activación pueden diferir de un estrato a otro, pero deben ser iguales dentro de un mismo estrato. También es posible tener más de un estrato escondido de neuronas.

2.9 Matriz de pesos

La figura 10.1 identifica las neuronas y los pesos de una red neural. Las señales de la 'a' a la 'f' son las entradas de la red, es decir, vienen del mundo exterior. Los valores actuales de las neuronas no se indican porque dependen de la información presentada a la red. Los valores que se indican en la figura 10.2 son los pesos de las conexiones neurales, dentro del estrato. Los pesos ejecutan una función similar a las sinapsis, ellos controlan la intensidad de las entradas a la neurona. El peso de cada conexión es representado por w_{ij} , donde i es la neurona que recibe y j es la neurona que envía.

En la figura 10.1 las conexiones excitatorias (peso positivo) aparecen con un pequeño círculo negro, el resto, son conexiones inhibitorias (peso negativo). Por ejemplo, el peso w_{kg} (-2), muestra la cantidad por la cual la señal de la neurona g es multiplicada previo a ser sumada con las otras señales en la neurona k.

En representación matricial, a cada estrato de conexiones le corresponde una matriz diferente. Los valores de los pesos son ordenados así: los elementos de las columnas representan las neuronas que envían y los elementos de las filas representan las neuronas que reciben. La figura 10.3 representa la matriz de pesos del estrato 1, la figura 10.4 representa la matriz de pesos del estrato 2.

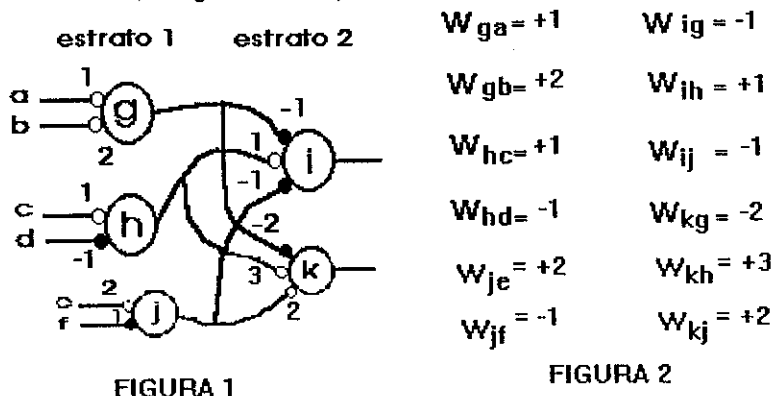


FIGURA 1

FIGURA 2

	a	b	c	d	e	f	
g	+	+					estrato 1
h			+	-			
j					+	-	

FIGURA 3

	g	h	j	
i	-	-	-	estrato 2
k	-	+	+	

FIGURA 4

FIGURAS 10.1, 10.2, 10.3, 10.4
Matriz de pesos

2.10 Funciones de activación y transferencia

Las neuronas procesan las entradas que reciben para producir una salida. Estas entradas corresponden a las salidas de muchas otras neuronas; una vez dentro de la neurona, estas señales y los pesos son sumados para obtener la salida, es decir, el valor que la neurona aporta a la red. En la mayoría de modelos se utiliza la ecuación 2.10.1:

$$net_i = (w_{ij} * o_j)$$

ECUACION 2.10.1

Esta ecuación indica que la salida de la neurona i es igual a la sumatoria del producto de todas las entradas a la neurona i , procedentes de la neurona j , por el peso de la salida de la neurona j .

La neurona, como se muestra en la figura 11, calcula su salida mediante la sumatoria de sus entradas (net_j) y la aplicación de la función de activación. Esta función produce un nivel de activación (a_j) dentro de la neurona. La activación es transmitida a una salida, o función de transferencia f_j , la cual produce la salida de la neurona i durante el período t , $o_j(t)$.

En los modelos más sencillos, la función de activación es la sumatoria net_j , el estado anterior no se toma en cuenta. En modelos más complicados, la función de activación también utiliza la salida de la neurona en el período anterior ($t-1$), de tal forma que la neurona puede ser autoexcitada. Estas funciones de activación son decrecientes en el tiempo; un estado de excitación bajo lleva a un nivel de inactividad. Algunas veces la función de activación es estocástica, es decir, incluye un factor de variación aleatoria.

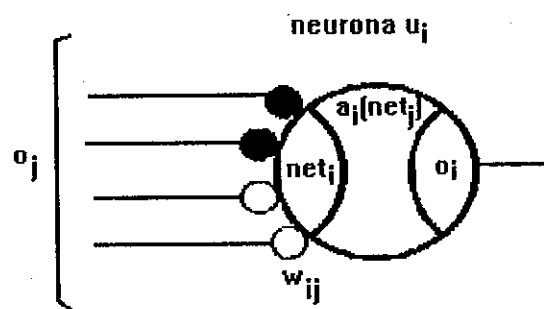


FIGURA 11
Funciones de una neurona

El estado de activación es una forma de referirse al estado de la red neural en un período determinado. Cada neurona tiene un valor de activación individual, el cual puede denominarse $a_i(t)$, donde a significa activación, i es la neurona y t es el período. La función de activación específica lo que la neurona hará con las entradas, luego que los pesos han realizado su efecto.

En la mayoría de modelos, el valor de activación puede ser cualquier número en un rango permitido de valores continuos, incluyendo cantidades no enteras. En otros modelos, el valor de activación es determinado y discreto, es decir, sólo permite valores enteros. Los valores discretos más utilizados son -1 y $+1$, donde -1 significa inactivo y $+1$ significa activo. En algunos otros modelos el valor de activación no tiene un rango establecido, lo cual se denomina valor de activación ilimitado y continuo.

La función de activación puede ser usada para realizar algún ordenamiento en el tiempo de las entradas, de forma que la neurona y la red tienen una conducta dependiente del tiempo.

2.10.1 Función de transferencia lineal

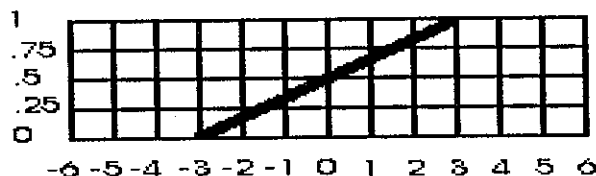


FIGURA 12
Función de transferencia lineal

Una función de transferencia lineal es aquella cuya salida tiene una inclinación constante. La función se compone del valor central b , el cual representa el punto en el que la salida es cero y el valor de inclinación m . En la figura 12, $b=0.5$ y $m=1/6$.

2.10.2 Función de transferencia umbral

La función de transferencia umbral es aquella cuya salida se caracteriza por ser el múltiplo constante de la entrada sobre algún rango, posiblemente desplazado hacia la derecha o la izquierda un valor constante. Si el valor que se obtiene está por debajo del límite inferior del rango, la función lo toma como BAJO; si el valor está sobre el límite superior del rango, la función lo toma como ALTO. En otras palabras, este tipo de función se comporta como una función de transferencia lineal con un valor central $b=(ALTO+BAJO)/2$ y con una inclinación m , obtenida mediante el comportamiento de la función.

Debido al valor de umbral, las redes construidas con este tipo de función tienen un comportamiento de mayor interés que aquellas construidas con funciones lineales. La función de umbral fue usada en la red perceptrón. La desventaja que presenta esta función es la limitación de su aprendizaje.

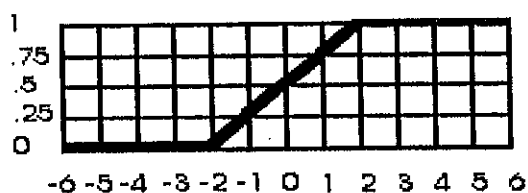


FIGURA 13
Función de transferencia umbral

2.10.3 Función de transferencia step

Una función de transferencia step es aquella cuya salida se limita a dos posibles valores. Por debajo de un umbral, la salida es siempre BAJO y por arriba del umbral, es siempre ALTO. Se comporta igual que un chip de lógica digital (0 y 1). Se denomina valor central al punto en el cual la salida salta de BAJO hacia ALTO (si el step es de incremento) o de ALTO a BAJO (si el step es de decremento).

Debido a la discontinuidad en la función las redes construidas con esta función presentan un comportamiento mucho más interesante que aquellas construidas con funciones lineales o con funciones de umbral. Este tipo de función se utilizó en algunos de los trabajos iniciales de las redes neurales, tales como la red McCulloch-Pitts (sección 1.2). Además, es el tipo de función más utilizado en las redes de Hopfield.

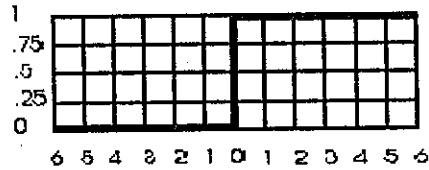


FIGURA 14
Función de transferencia step

2.10.4 Función de transferencia Sigmoid

Este tipo de función es conocida también como función en forma de S "s-shaped", función semilineal o función squashing. La salida es una función monótona y continua de la entrada. Tanto la función en sí como su derivada son continuas todo el tiempo, es decir, son asintóticas en los límites ALTO y BAJO.

El valor central es aquel donde la salida es igual a $(ALTO+BAJO)/2$. La inclinación es directamente proporcional a la derivada de la función en el punto central.

Debido a que la función es no-lineal, continua y continuamente diferenciable, presenta muchas propiedades deseables cuando se construye una red neural.

Si la inclinación tiende a 1, la función Sigmoid se aproxima a una función step; si la inclinación tiende a 0, se aproxima a una función lineal. Una inclinación de 1 es comúnmente una buena elección.

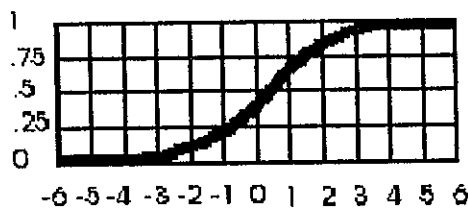


FIGURA 15
Función de transferencia Sigmoid

2.10.5 Función de transferencia de Gauss

Esta función también es conocida como la campana de Gauss y rara vez se elige como función de transferencia. La función no ha sido utilizada en trabajos anteriores, el único producto, actualmente conocido, que utiliza la función de Gauss es BrainMaker.

El valor central es aquel en el cual la salida es igual a ALTO. La inclinación es proporcional a la desviación estándar de la campana.

La función de Gauss es monótona en ambos sentidos, continua y continuamente diferenciable.

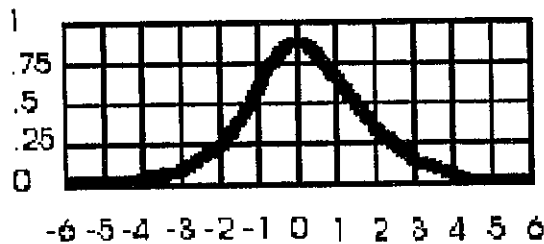


FIGURA 16
Función de transferencia de Gauss

2.11 Entrenando una red neural

El cerebro ejecuta su hazaña de procesamiento a través de paralelismo masivo (sección 2.2). Para simular este paralelismo, la red neural artificial consta de una red de elementos de procesamiento, cada uno de éstos es la analogía electrónica de una neurona. Cada elemento de procesamiento tiene un número de entradas, un pequeño conjunto de posibles estados y una salida que es una función de las entradas. Cada entrada al elemento de procesamiento tiene un valor de peso.

Cuando un elemento de proceso es activado, evalúa todas sus entradas y computa sus respectivos valores de peso. Si el valor de peso cumple cierto patrón, la unidad de cómputo genera un valor de salida que es usado como entrada por otros elementos de proceso. Sólo los valores de peso de las entradas cambian durante el aprendizaje.

Entrenar una red neural es cuestión de ajustar pesos, ya sea manual o automáticamente. Una red neural consiste de un número de nodos o elementos de procesamiento. Cada elemento de procesamiento tiene una única señal de salida, la cual se interconecta con otros elementos de procesamiento. Cada nodo procesa la señal que viene, basada en los valores de las constantes almacenadas en él. Las reglas de aprendizaje determinan la variación de los pesos.

2.11.1 Reglas de aprendizaje

2.11.1.1 Regla de Hebb

La regla de Hebb define los cambios en los pesos:

$$w_{ij} = \Delta a_i \cdot o_j$$

ECUACION 2.11.1.1.1

Δw_{ij}	cambio en el peso de la conexión entre la neurona j y la neurona i
a_i	activación para la neurona i
o_j	salida de la neurona j
Δ	tasa de aprendizaje

La tasa de aprendizaje especifica qué tan significativo es el cambio para la red, o bien, qué tan rápido se adapta la red. Si las conexiones son cambiadas dramáticamente durante el entrenamiento, la red se comporta extrañamente: puede ser que aprenda en un tiempo mayor o que no aprenda bien.

2.11.1.2 Regla delta

La regla delta establece que si existe una diferencia (durante el entrenamiento), entre el actual patrón de salida y el patrón de salida deseado, los pesos son cambiados para reducir esta diferencia. La variación de los pesos se representa por la ecuación 2.11.1.2.1:

$$\Delta w_{ij} = \Delta (T_i - a_i(t)) \cdot o_j(t)$$

ECUACION 2.11.1.2.1

Δw_{ij}	cambio en el peso de la conexión entre la neurona j y la neurona i
T_i	valor de entrada o respuesta correcta
t	período de tiempo
a_i	activación de la neurona i
o_j	salida de la neurona j
Δ	tasa de aprendizaje

La regla delta, también llamada "least mean squared" (LMS), es una variación de la regla de Hebb (sección 2.11.1.1). Fue desarrollada por Bernard Widrow y Ted Hoff, ambos de la Universidad Stanford, cuyas publicaciones se realizaron en 1,960. La red que desarrollaron se llama ADALINE, una abreviación de elemento de adaptación lineal "adaptive linear element".

La regla delta generalizada, llamada también regla de propagación hacia atrás "backpropagation", es una variación de la regla delta.



2.11.2 Formas de aprendizaje

2.11.2.1 Supervisada

Un conjunto de ejemplos de entrenamiento son presentados, uno a uno, a la red. La red entonces calcula las salidas basadas en su actual entrada. La salida resultante es entonces comparada con una salida deseada para ese ejemplo en particular. El error entre las dos salidas es usado para modificar los pesos en la red, de forma que el error se reduce.

2.11.2.2 No supervisada

Los datos son presentados como un vector de entrada, sin la intervención del humano. La regla de aprendizaje ajusta los pesos de forma tal que los ejemplos de entrada son agrupados en clases, basado en sus propiedades estadísticas.

2.11.2.3 Auto supervisada

Ocurre cuando la red se monitorea a sí misma y corrige los errores en la interpretación de datos mediante retroalimentación.

2.11.3 Algoritmos de aprendizaje

A continuación se exponen brevemente algunos de los algoritmos existentes para el aprendizaje de una red neural.

2.11.3.1 Algoritmo de propagación hacia atrás "backpropagation"

En 1985-1986, este algoritmo fue simultáneamente desarrollado por tres grupos de personas:

- Rumelhart, G.E. Hinton y R.J. Williams
- Le Cun
- Parker

La propagación hacia atrás es un método de aprendizaje supervisado en el cual los errores calculados a la salida de la red, como resultado de un ejemplo de entrenamiento, son propagados hacia atrás, de estrato a estrato hacia la entrada. En cada estrato dentro de la red, los errores son usados para actualizar los pesos. Los errores para el estrato de salida se obtienen sustrayendo el vector de salida actual del vector de salida deseado. Alterando los pesos durante su paso, el algoritmo previene que el mismo error ocurra nuevamente.

Una desventaja de la propagación hacia atrás es que puede tomar un largo número de iteraciones para converger a la solución deseada, a pesar de ello, es el algoritmo más ampliamente usado.

2.11.3.2 Algoritmo ATNN

Un algoritmo adaptativo de entrenamiento "adaptatively trained neural network" (ATNN) para perceptrones de múltiples estratos ha sido propuesto por algunos investigadores. Este algoritmo usa un perceptrón de un estrato, el cual ha sido previamente entrenado para adaptar los pesos de los nuevos datos. Los algoritmos de aprendizaje convencionales, tales como propagación hacia atrás, pueden dar

un resultado promedio para datos inconsistentes, lo cual puede ocurrir frecuentemente en un ambiente no estacionario.

Este algoritmo utiliza métodos de programación no lineal para adaptar datos de entrenamiento temporalmente no estacionarios. Una de las ventajas del algoritmo ATNN es que tiene la flexibilidad adicional de permitir priorizar los pesos a los datos de entrenamiento.

Con el propósito de adaptar los pesos del estrato del perceptrón a una variación no estacionaria lenta, el procedimiento debiera cumplir con:

- Seguir respondiendo apropiadamente a datos de entrenamiento previo, si esos datos no están en conflicto con los nuevos datos de entrenamiento.
- Adaptarse a los nuevos datos de entrenamiento aún cuando estén en conflicto con porciones de los antiguos datos.

Este algoritmo fue propuesto por Dong Chul Park; El-Sharkawi, A. Mohamed y Robert J. Marks II ¹⁰. Su propuesta es un procedimiento que cumple con las restricciones anteriores y que es aplicable cuando los datos de entrenamiento estacionarios varían suficientemente despacio. Este procedimiento asegura una respuesta apropiada a los datos de entrenamiento previos, mediante la búsqueda de la minimización de una función de costo sensible al peso, mientras que se minimiza el error cuadrático.

2.11.3.3 Algoritmo de entrenamiento rápido para redes neurales con múltiples estratos

Entrenar una red neural de múltiples estratos con propagación hacia atrás es lento y requiere de elecciones arbitrarias del número de unidades escondidas y de estratos. El algoritmo descrito es mucho más rápido que propagación hacia atrás (aunque este último es muy popular) y no es necesario especificar inicialmente el número de unidades escondidas.

Existe una relación entre este algoritmo y algunos otros algoritmos de reconocimiento rápido de patrones, como aquellos basados en árboles k-d. La idea principal es entrenar un perceptrón de múltiples estratos construyendo primero un árbol de decisión y derivando después el perceptrón del árbol.

Generalmente el algoritmo alcanza precisión para el reconocimiento, igual o mejor, que los perceptrones de múltiples estratos entrenados usando propagación hacia atrás y el proceso de entrenamiento es mucho más rápido que el de éste último. La precisión de reconocimiento alcanzada por el algoritmo es comparable al algoritmo del el vecino más cercano "nearest_neighbor", el cual es más lento y requiere más espacio de almacenamiento. Se concluye que el algoritmo debiera ser útil para reconocimiento de patrón práctico; también sirve para demostrar una conexión más fuerte entre las redes neurales y ciertas clasificaciones y métodos de recuperación de datos usados por los estadísticos.

Este tema ha sido desarrollado por Richard P. Brent¹¹.

10 Park, Dong Chul; El-Sharkawi, A. Mohamed y Robert J. Marks II. **An Adaptively Trained Neural Network**. IEEE Transactions on Neural Networks, vol.2, No.3, mayo 1991. 334-335 pp.

11 Brent, Richard P. **Fast Training Algorithms for Multilayer Neural Networks**. IEEE Transactions on Neural Networks, vol.2, No.3, mayo 1991. 346-354 pp.

2.11.3.4 Entrenamiento iterativo

Este tipo de entrenamiento comúnmente involucra el ajuste incremental de los pesos en la red, de acuerdo a un conjunto de ejemplos de entrenamiento, que son repetidamente presentados a la red hasta que algún criterio de convergencia es logrado. El entrenamiento iterativo ha sido empleado para enseñar al neocognitrón (inciso 4, sección 3.3.2), los pasos a seguir son los siguientes:

- Todos los pesos son valores seleccionados puestos a cero.
- Cada patrón de entrenamiento es presentado a la red y la neurona correspondiente a la clase a la cual ese patrón pertenece es modificada sumándole un pequeño incremento, el cual es igual a 'n' veces el patrón 'x' de los pesos correspondientes 'a'.
- El valor de inhibición 'b' es modificado sumándole un pequeño incremento por las veces de la magnitud del patrón ($\sum U_j$).
- El valor de selección es también modificado sumándole un pequeño incremento (diferente al anterior).

Este proceso es repetido hasta que el porcentaje de no clasificación decae por debajo de un umbral presentado, hasta que otra mejora es hecha o hasta que la tasa de clasificación correcta empieza a decaer. Sin embargo, este esquema de entrenamiento sufre de ciertos defectos:

- No hay satisfacción intuitiva para el criterio de terminar con el entrenamiento. Si el entrenamiento continúa indefinidamente, entonces el valor de selectividad de cualquier neurona puede ser incrementado al punto donde aún los patrones de entrenamiento serán rechazados por ella misma.
- Aún, si el entrenamiento prosigue a un punto en el cual no hay clasificaciones ambiguas (es decir, que dos o más neuronas responden simultáneamente al mismo patrón de entrada) de los datos de entrenamiento, un nuevo ejemplo de entrenamiento puede generar una respuesta ambigua.

En otras palabras, el clasificador resultante es sub-óptimo, puesto que la clasificación está incompleta. Finalmente, es fácil de construir casos para los cuales el procedimiento de entrenamiento no terminará. Esto significa que el tiempo de entrenamiento es dependiente de los datos y no se garantiza que converja.

2.11.3.4 Entrenamiento de forma óptima "optimal closed-form"

Este entrenamiento es realizado en dos etapas: en la primera etapa el aprendizaje supervisado es usado para establecer la dirección de cada plantilla. El patrón para una clase será la suma del vector de todos los patrones de entrenamiento asignados a esa clase y la dirección de cada plantilla es aquella de su promedio no pesado. En la segunda etapa, el umbral, o valores de selección r , son determinados.

Puesto que este proceso garantiza que las regiones de aceptación de todas las clases están mutuamente no traslapadas, garantiza también la clasificación completa. Por la construcción, las regiones de aceptación son tan largas como puedan sin que se traslapen. Entonces, la generalización ha sido maximizada bajo la condición de exclusión mutua.

Es fácil considerar modificaciones a este proceso, con el propósito de generar clasificadores parciales que obedezcan a criterios específicos.

2.12 Número de ejemplos requeridos para el entrenamiento

A la vez que hay muchos resultados empíricos que prueban la utilidad de las técnicas de aprendizaje de las redes neurales, aspectos tales como las capacidades, limitaciones y requerimientos de las mismas son relativamente menos comprendidos. Muchos problemas, como el número de ejemplos de entrenamiento requeridos para lograr un aprendizaje exitoso y el tamaño de la red neural para una tarea específica, son resueltos en la práctica mediante prueba y error. Algunos resultados han sido alcanzados recientemente en el intento de resolver estos inquietantes problemas, pero el área es extensamente amplia para la investigación. Estas interrogantes son difíciles de resolver porque existe una considerable dependencia de la aplicación y la red neural que se utiliza.

Para que el aprendizaje neural sea considerado exitoso, es esencial que el sistema ejecute correctamente la clasificación de ejemplos de prueba en los cuales el sistema no ha sido entrenado. Se enfatiza en las capacidades de una red en el sentido de generalizar con los ejemplos de entrenamiento, no en el sentido de memorizarlos.

Una red con muy pocas neuronas escondidas, o entrenada con muy pocos ejemplos, tiende a memorizar el conjunto de ejemplos y producir resultados pobres durante las pruebas. Si una red aprende rápidamente los ejemplos, pero obtiene resultados pobres, puede ser que esté pasada de peso "overweight"; este tipo de red tiene muy pocas conexiones para el problema y no generaliza. En vez de aprender, memoriza algunas reglas que satisfacen los requerimientos de los ejemplos.

El número mínimo de ejemplos necesarios depende del tamaño de la red. Mientras mayor sea la red (especialmente, entre más neuronas escondidas tenga), más ejemplos son necesarios.

$$N = (\#_neuronas_escondidas / tolerancia) * \#_neuronas_entrada$$

ECUACION 2.12.1

Kichan G Mehrotra, Chilukuri K. Mohany y Sanjay Ranka han desarrollado el tema del número límite de ejemplos necesarios para el aprendizaje de la red neural¹². Otros intentos, según los autores, han sido inadecuados por las suposiciones poco realistas que han tomado.

La utilidad de conocer el número de ejemplos requerido es obvia cuando se desea colocar una gran confianza en que el aprendizaje de la red neural haya sido completado. Con el interés de reducir el tiempo de entrenamiento, es de importancia entrenar a las redes neurales con un número limitado de ejemplos, pero si muy pocos ejemplos han sido usados, la capacidad de generalización de la red será pobre y no se podrá esperar que funcione bien en los casos de prueba que no ha sido entrenada.

12

Mehrotra, Kichan G; Chilukuri K. Mohany y Sanjay Ranka. **Bounds on the Number of Samples Needed for Neural Learning**. IEEE Transactions on Neural Networks, vol.2, No.6, noviembre 1991. 548-558 pp.8

3. CLASIFICACIÓN DE REDES NEURALES

3.1 Introducción

En lo referente a tareas que requieren conjuntos de datos incompletos, información confusa o contradictoria, las redes neurales tienen mejores resultados que las computadoras convencionales, incluyendo los procesadores en paralelo. Sin embargo, las redes neurales no son tan buenas manejando determinadas tareas como pueden serlo las computadoras comunes, por ejemplo, realizando cálculos que requieren cierta precisión. Los cómputos de las redes neurales tienen limitantes de I/O al igual que los cómputos convencionales las tienen.

El paralelismo masivo da a las redes neurales un alto grado de tolerancia a fallas, llamadas asociativas y la habilidad para recuperarse elegantemente si ocurre alguna falla con el procesador. Estas propiedades hacen de las redes neurales una opción atractiva para muchas aplicaciones comerciales, militares e industriales.

La teoría de las redes neurales está siendo estudiada en dos aspectos principales: primero, la eficiencia de la arquitectura electrónica basada neuronalmente y segundo, el logro de un entendimiento de las funciones biológicas de las redes neurales.

A medida que se acerca el siglo XXI, necesitamos una nueva aproximación a la informática, describiendo las máquinas de redes neurales. Al igual que existe una estructura formal para la red neural biológica, las redes neurales artificiales más eficientes necesitarán una estructura y un orden para determinar cómo aprender, procesar y seleccionar la información de entrada. Necesitarán también saber cuántas partes diferentes de un sistema inteligente ejecutan funciones específicas.

Una de las áreas de estudio que están siendo exploradas por David Rumelhart, profesor de psicología en la Universidad de Stanford, es la referente al desarrollo de las redes que pueden elegir sus propias arquitecturas. Aún en su más rudimentaria etapa, esta ciencia usa una clase de lazo de retroalimentación para la construcción "built_in_feedback_loop", similar al uso de modelos neurales para resolver problemas relativamente especializados y aprender de sus experiencias.

La ejecución de una red neural depende de muchos elementos. Algunos de los más importantes, actualmente, son los siguientes:

- ¿Cuántas estratos y neuronas son suficientes?
- ¿Qué tan creativo debe ser el sistema?
- ¿Qué tipo de red es el adecuado?

Robert Hecht-Nielsen, cofundador de la corporación de neurocomputadoras Hecht_Nielsen en San Diego, sugiere que se revisen al menos seis criterios cuando se elige una configuración de red neural, éstos se listan a continuación:

1. Formato óptimo de I/O
2. Tiempo de entrenamiento
3. Requerimientos de preprocesamiento de datos
4. Optimización matemática
5. Estimados de ejecución
6. Requerimientos de prueba/ejecución

Muchas personas se atreven a predecir que las redes neurales no reemplazarán los métodos convencionales de cómputo, especialmente aquellos relacionados con procesamiento numérico de alta

velocidad, pero los complementarán. La combinación de las computadoras tradicionales y el poder único de redes neurales resolverán problemas que, de otra forma, podrían seguir sin solución.

3.2 Desarrollo histórico de las teorías de redes neurales

La investigación sobre redes neurales tuvo sus primeros resultados importantes hace cuarenta años, cuando Warren McCulloch y Walter Pitts demostraron que una red compuesta de neuronas de valor binario eran capaces de ejecutar cálculos simples con lógica de umbral (sección 1.2).

En 1,949 Donald Hebb, en su libro **Organization of Behavior** (Organización de la Conducta), propuso un mecanismo biológico mediante el cual el aprendizaje se podía llevar a cabo. La mayoría de reglas de aprendizaje tienen su origen en la regla de Hebb o una variación de ella.

En la década de los 50's, la figura dominante en la investigación de redes neurales fue el psicólogo Frank Rosenblatt. El diseñó una clase de red llamada perceptrón, en Cornell Aeronautical Lab y escribió ampliamente sobre su potencial. Sus afirmaciones indicaban que esta red era capaz de realizar el reconocimiento de patrones, aprendizaje asociativo y atención selectiva; sin embargo, las habilidades del perceptrón eran limitadas. El perceptrón, diseñado para simular un modelo de sensor biológico, usó una combinación de diferentes estratos de neuronas de umbral y una variación de la regla de Hebb.

En la década de los 60's, Marvin Minsky y Seymour Papert realizaron un estudio profundo de las capacidades y limitaciones del perceptrón. Este estudio terminó en 1,969, cuando publicaron el libro **Perceptrons**; ellos concluyeron que una gran cantidad de interesantes problemas no podían ser resueltos por las redes perceptrón. Este libro redujo notablemente el desarrollo de las investigaciones sobre redes neurales, durante la década de los 70's, mientras que las reglas de inteligencia artificial, establecidas por Minsky y Papert, dieron origen a un nuevo y atractivo campo de investigación.

Al final de la década de los 60's y al inicio de la década de los 70's, el procesamiento serial y la computadora de Von Neumann dominaron tanto la psicología como la inteligencia artificial. Pocos investigadores continuaron trabajando con redes neurales, entre ellos se conoce a Steven Grossberg, Geoffery Hinton, Teuvo Kohonen, Kunihiko Fukushima y J.A. Anderson. Sus esfuerzos originaron conocimientos fundamentales en la teoría de redes neurales.

Anderson y Kohonen (1,972) desarrollaron simultáneamente un asociador lineal que utilizaba neuronas que podían ser disparadas en frecuencias variables como respuesta a las señales de entrada, en lugar de simplemente encenderse o apagarse como lo hacía el perceptrón. Stephen Grossberg fue de los primeros en analizar algunas de las propiedades del aprendizaje competitivo (1,976), su análisis matemático le llevó a conocer ciertos aspectos que podían ser apreciados únicamente mediante muchas simulaciones. El percibió la relevancia de mecanismos neuralmente inspirados en áreas de la percepción y la memoria, antes que el campo de las redes neurales estuviese listo para tales ideas (1,978).

El trabajo de J.A. Anderson influenció grandemente la investigación de James McClelland y David Rumelhart, referente al desarrollo de las teorías de procesamiento paralelo distribuido (Parallel Distributed Processing, PDP). Anderson insistió en una representación de datos distribuidos y demostró la relevancia de modelos inspirados neuralmente para teorías del aprendizaje, lo cual condujo a las teorías de PDP.

En 1,982 John Hopfield, profesor de química y biología en California Institute of Technology, publicó un trabajo sobre redes neurales, el cual demostró que éstas son capaces de comportarse de una forma interesante. La aportación más importante de Hopfield fue darle credibilidad al campo de las redes neurales, estimulando a una nueva generación de investigadores. Las dos ideas principales descritas por Hopfield en sus publicaciones son: agregar el concepto de sistemas no lineales y el concepto de la función de energía global que caracteriza al sistema estable.

En los siguientes nueve años, a partir de 1,982, muchos modelos de red y reglas de aprendizaje se desarrollaron; éstos han demostrado tener habilidades interesantes, tales como reconocimiento de patrones, tolerancia a fallas y predicción.

La tabla 1 resume el desarrollo histórico de las investigaciones sobre redes neurales.

AÑO	INVESTIGADOR	TRABAJO	RELEVANCIA
1,943	McCulloch & Pitts	red de McCulloch-Pitts	primer modelo de red neural
1,949	Hebb	aprendizaje sináptico	bases biológicas
1,958	Rosenblatt	perceptrón	lógica de umbral
1,960	Widrow & Hoff	ADALINE	regla delta
1,969	Minsky & Papert	trabajo "Perceptrons"	libro de crítica
1,972	Anderson	asociador lineal	salida variable
1,972	Kohonen	asociador lineal	salida variable
1,973	Von der Malsburg	modelo de la corteza visual	pruebas de teorías fisiológicas
1,976	Grossberg	patrones de adaptación	psicología y matemática
1,976	Marr & Poggio	visión	cooperación
1,977	Amari	piscina neural	competencia
1,980	Grossberg	resonancia de adaptación	modelo psicológico
1,981	McClelland	percepción de las letras	detección de formas
1,982	Bienenstock	corteza visual	neurofisiología
1,982	Kohonen	mapeo de formas	auto-organización
1,982	Hopfield	autoasociación	estados estables
1,982	Feldman	conexionismo	procesamiento paralelo distribuido
1,983	Fukushima	neocognitrón	reconocimiento
1,983	Grossberg	ART	complejo modelo de retroalimentación
1,984	Hinton	máquina de Boltzman	templar
1,985	Rumelhart, Parker	propagación hacia atrás "backpropagation"	regla delta con estratos escondidos
1,986	Rumelhart	PDP	libros sobre redes neurales
1,986	Sejnowski	NetTalk	aplicación de la propagación hacia atrás
1,987	Kosko	BAM	asociación par
1,988	IEEE	IJCNN	conferencia de redes neurales
1,988	Specht	PNN	modelo introductorio
1,990	Intel	estándar i80170NX	chip de neurona comercial
1,992	Loral	tableros NET	hardware de aprendizaje rápido
1,993	Adaptive	CNAPS	hardware de aprendizaje rápido

TABLA 1
Resumen histórico

3.3 Clasificación de las redes neurales

Las redes neurales son clasificadas por el comportamiento de sus neuronas, el esquema de interconexión y el algoritmo de aprendizaje. Principalmente se clasifican como:

- i. retroalimentación "feedback"
- ii. alimentación hacia adelante "feedforward"

iii. probabilística "probabilistic"

RETROALIMENTACIÓN	ALIMENTACIÓN HACIA ADELANTE (BPN)	PROBABILÍSTICA (PNN)
red Hopfield	perceptrón	red de regresión general
bidirectional associative network	asociador lineal	radial basis function
adaptive resonance theory	red de Kohonen	gaussian basis functions
	neocognitrón	RCE
		weighted probabilistic

TABLA 2
Clasificación de las redes neurales

3.3.1 Retroalimentación

En una red de este tipo, las salidas de las neuronas retroalimentan "feedback" a las neuronas del mismo estrato o del estrato anterior, ver figura 17. La retroalimentación se refiere a la forma en que las neuronas se conectan. Este tipo de red envía señales de regreso, como entradas a otras neuronas; ésto no debe confundirse con la propagación hacia atrás, que describe un método de entrenamiento en el que el error obtenido en el estrato de salida es usado para corrección. Las redes de retroalimentación no usan el algoritmo de propagación hacia atrás para entrenarse y las redes de propagación hacia atrás no son modelos de retroalimentación.

Los modelos de retroalimentación pueden ser construidos o entrenados. En una red construida se forma la matriz de pesos. Luego de la construcción, un patrón parcial o poco preciso puede ser presentado a la red y después de un tiempo, se espera que la red converja a uno de los patrones de entrada originales. Las redes Hopfield y BAM son dos populares modelos de retroalimentación construidos.

En las redes con retroalimentación, la salida de una neurona puede ser conectada, mediante algún camino, con sus propias entradas o con la entrada de una neurona previa. Las salidas de la red con retroalimentación dependen siempre de las salidas previas de la red. Estas redes, que incluyen las de Hopfield, modelos Brain-State-In-A-Box y BAM, tienen la presencia de algún elemento aleatorio (tales como los estados iniciales de las entradas o salidas). Las redes con retroalimentación no encuentran siempre la misma solución a un problema, a pesar que las entradas sean iguales; sin embargo, representan, en general, una buena elección.

PROPIEDAD DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
Biblioteca Central

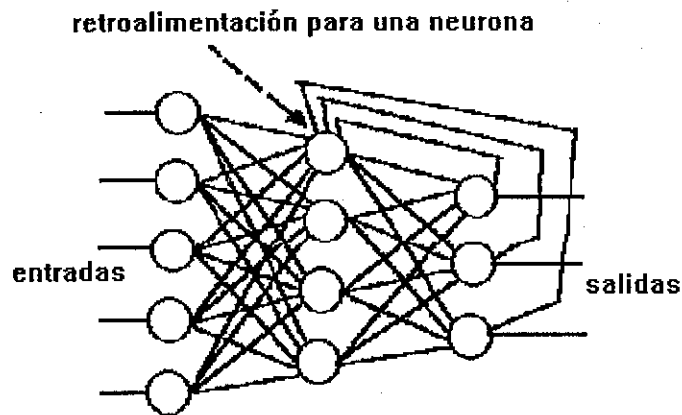


FIGURA 17
Red con retroalimentación

3.3.1.1 Red Hopfield

Una red Hopfield usa auto-organización y memoria asociativa. Está compuesta de un único estrato de neuronas que actúan tanto de entrada como de salida. Las neuronas están simétricamente conectadas ($w_{ij} = w_{ji}$), como lo muestra la figura 18. Cada neurona se conecta a todas las otras neuronas y tiene la característica de ser no-lineal y de asumir dos valores de salida: -1 (apagada) y +1 (encendida). Las redes Hopfield utilizan masivamente los ciclos de retroalimentación. Otra característica es que la salida de cada neurona depende de los valores previos de su propia activación, de forma que las neuronas individuales tienen una conducta dependiente del tiempo.

En el modelo original de Hopfield, las neuronas tenían la función de transferencia Step y se actualizaban asincrónicamente. La función de salida de la neurona era no-lineal. A pesar de su aparente simplicidad, una red Hopfield tiene un alto grado de ejecución. Los modelos más recientes usan actualizaciones sincrónicas para las neuronas con valores continuos.

Un proceso de reacción-estimulación-reacción entre las neuronas ocurre hasta que la red llega a un estado estable. De esta forma, la red resulta en una respuesta directa de la entrada.

Las redes Hopfield pueden reconocer patrones, comparando nuevos patrones de entrada con patrones almacenados previamente. Cuando se aplica un patrón, se determina cuál de los patrones almacenados es el más parecido. Las redes Hopfield son especialmente buenas para encontrar la mejor respuesta entre muchas posibilidades, así como para recordar todo acerca de algún dato almacenado cuando se les provee de información incompleta. Frecuentemente se utilizan como una forma de memoria contenida direccionable (Content Addressable Memory, CAM).

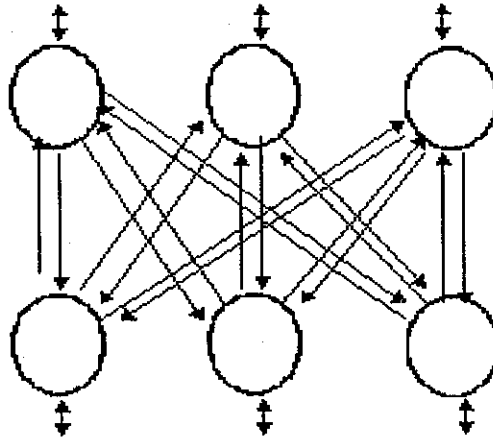


FIGURA 18
Red Hopfield

3.3.1.2 Red BAM

La red Hopfield es una red con retroalimentación canónica. Otras redes con retroalimentación, tales como red de memoria asociativa bidireccional "bidirectional associative network" (BAM), son familiares cercanas de la red Hopfield. La red BAM tiene un estrato de entrada y uno de salida, como lo muestra la figura 19. Las neuronas son del tipo de umbral lineal y utilizan la regla de aprendizaje de Hebb. Para crear la matriz de pesos, BAM utiliza pares de patrones, o asociación de patrones "pair association". Luego de construir la matriz de pesos, cada patrón en un par puede utilizarse para recordar al otro.

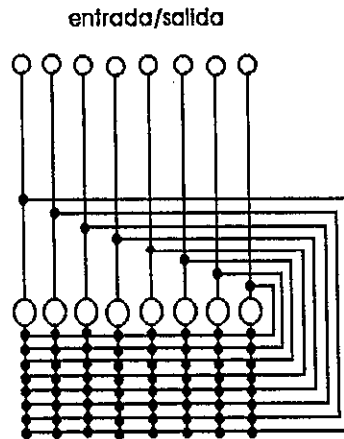


FIGURA 19
Red BAM

3.3.1.3 Red ART

Otro tipo de red con retroalimentación es conocida como teoría de resonancia de adaptación "Adaptive Resonance Theory" (ART), desarrollada por Stephen Grossberg y Gail Carpenter de la Universidad de Boston, mostrada en la figura 20. Este es un modelo altamente complejo, basado en teorías psicológicas y matemáticas. Las neuronas tienen funciones de activación y esquemas de interconexión más complicados que otros modelos. Gran parte del modelo ART no se basa en fisiología, de forma que no es una red neural en este sentido.

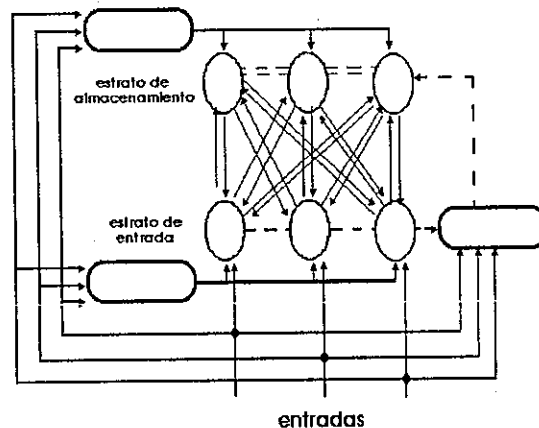


FIGURA 20
Red ART

La red ART utiliza un único método de entrenamiento no supervisado. Un patrón de entrada es transmitido al estrato de almacenamiento, a través de los pesos de las conexiones. Debido a que el estrato de almacenamiento tiene inhibición lateral, sólo el nodo con la respuesta mayor puede generar una salida. Esa salida es mandada hacia atrás al estrato de entrada, pasando por otro conjunto de pesos. En este punto, si el patrón de actividad es igual al patrón de entrada original, se dice que los dos patrones son resonantes.

3.3.2 Alimentación hacia adelante (BPN)

En una red BPN, las señales van sólo en una dirección, no existen ciclos en el sistema, ver figura 21.

Los primeros modelos de red neural fueron del tipo BPN. Las redes BPN son menos frecuentemente relacionadas con memoria asociativa que las redes con retroalimentación; sin embargo, son capaces de dar la misma funcionalidad. Matemáticamente se puede demostrar que cualquier red con retroalimentación tiene una red BPN equivalente, la cual ejecuta la misma tarea.

Según las investigaciones realizadas, las actuales arquitecturas de red BPN trabajan mejor que las arquitecturas de retroalimentación. Durante la ejecución, los modelos BPN son más rápidos debido a que necesitan hacer una pasada a través del sistema para encontrar una solución, mientras que las redes con retroalimentación deben realizar ciclos repetitivamente hasta que la salida de la red deja de cambiar, lo cual toma de 3 a 1,000 ciclos.

Actualmente, las redes neurales más utilizadas son los modelos no lineales de BPN.

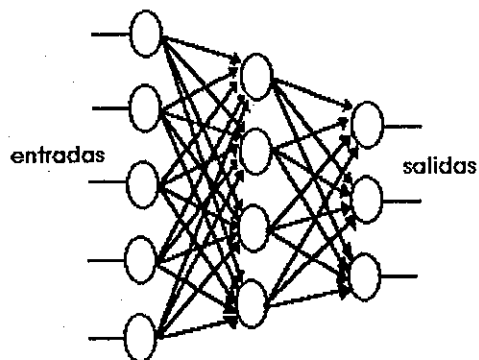


FIGURA 21
Red BPN

3.3.2.1 Perceptrón

En 1958, Frank Rosenblatt desarrolló el procedimiento de convergencia perceptrón, el cual mejoró la regla de Hebb utilizada en ese momento. A pesar que sus modelos no fueron capaces de todo lo que él esperaba, su visión acerca de ciertos tópicos importantes (p.e., la dinámica del procesamiento humano de la información, interacción y del modelo auto-organizado), fundamenta las teorías actuales.

El perceptrón fue diseñado para simular un modelo de sensor biológico. Utiliza una combinación de diferentes estratos de neuronas. El primer estrato, el área de proyección A_i , está compuesto de sensores. El segundo estrato, el área de asociación A_{ii} , está compuesto de decodificadores. El tercer estrato es el de respuestas. Las conexiones entre los estratos son aleatorias. La única retroalimentación en la red se presenta entre el estrato de respuesta y el área de asociación.

La regla de aprendizaje usada es una variante de la regla de Hebb. El cambio de los pesos de las conexiones son proporcionales al producto de las actividades de conexión entre las neuronas que envían y las que reciben.

Si la suma de los impulsos es igual o mayor que el umbral de A_i , entonces A_i dispara a A_{ii} , la cual responde de igual manera. A_i y A_{ii} son llamadas unidades lógicas de umbral, las cuales calculan un resultado lógico de sus entradas y su salida puede ser uno de los dos estados, on y off (ver figura 3).

El aprendizaje del perceptrón es limitado, ésto originó el pesimismo de Marvin Minsky y Seymour Papert (1969) en su libro **Perceptrons**, en el cual demostraron que todos los perceptrones eran equivalentes al perceptrón Sencillo y que no podrían procesar ciertas propiedades básicas de una imagen, tales como la falta de interconexión (una figura interconectada es aquella que puede ser dibujada sin levantar el lápiz del papel). Ellos indicaron que sus propios algoritmos de programación sí eran capaces de hacer tales distinciones y que el método top-down era mejor. El libro **Perceptrons** convenció a muchos que la arquitectura de un único procesador era más prometedora, por ello las investigaciones de redes neurales se redujeron considerablemente durante los últimos años de la década de los 60's y los primeros de la década de los 70's.

3.3.2.2 Asociador Lineal

En 1,972, el trabajo simultáneo e independiente de los investigadores James A. Anderson, neurofisiólogo y Teuvo Kohonen, ingeniero eléctrico, les condujo a proponer el mismo modelo de alimentación hacia adelante de memoria asociativa. El asociador lineal cuenta con muchas neuronas, las cuales responden a los cambios en sus entradas modificando la tasa de disparo de sus salidas. Las unidades lógicas de umbral, del tipo usado en perceptrón, no se consideran apropiadas.

El modelo lineal, con la regla simple de Hebb, es llamado asociador lineal simple. El modelo se prueba presentándole entradas cuyo patrón de salida no se compara con el patrón de salida de las entradas originales, con las cuales el modelo fue entrenado. Todo patrón que pueda aprender un perceptrón, puede ser aprendido por un asociador lineal, pero no al revés.

Los patrones de actividad grandes tienen muchas entradas y salidas activas al mismo tiempo. La asociación entre los patrones de actividad grandes permite la correlación múltiple entre diferentes elementos en un nivel sináptico (de conexión), puesto que cualquier conexión puede estar activa en más de un patrón. Esto fortalece las habilidades de la red para generalizar, eliminar el ruido y formar prototipos.

Si un conjunto incompleto de datos de entrada puede hacer que la red recuerde una versión completa de un patrón almacenado, el sistema se llama autoasociativo. En dicho sistema, todas las neuronas representan entradas y salidas, retroalimentándose así mismas como a cada una de sus vecinas. Este tipo de red es la base de muchos modelos con retroalimentación.

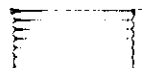
Si un conjunto completo de datos de entrada es requerido para que la red recuerde un patrón almacenado, el sistema es heteroasociativo.

3.3.2.3 Red de Kohonen

Este modelo es del tipo BPN no supervisado y utiliza neuronas adaptativas sencillas para recibir las señales. El mecanismo de este esquema causa que el sistema se modifique así mismo, provocando que las neuronas cercanas respondan de forma similar. Las neuronas compiten y sólo una de ellas emite la salida.

Las unidades inician apagadas y responden en forma aleatoria a los eventos. Al principio se provee a la red de una señal de entrada con un valor de parámetro, una neurona (la ganadora) responde mejor a esa entrada provocando que ella y todas sus vecinas cambien los pesos de sus conexiones. A partir de ese momento esta neurona y sus vecinas responden en forma similar.

La capacidad de la red Kohonen para transformar de diálogo a texto en los idiomas Finlandés y Japonés, ha sido demostrada por Kohonen, investigador en Helsinki University of Technology, Finlandia. El sistema construye el texto captando el diálogo a través de unidades de reconocimiento fonético, las cuales utilizan el chip TMS32010.



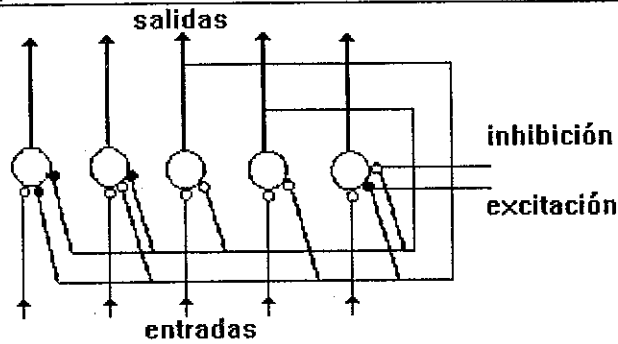


FIGURA 22
Red de Kohonen

3.3.2.4 Neocognitrón

El neocognitrón fue diseñado por Kunihiko Fukushima, del NHK Science and Technical Research Lab de Tokyo. Es un caso especial del modelo BPN. El modelo original es no supervisado, pero modelos más recientes (a partir de 1,983) tienen variaciones. Este sistema de siete a nueve estratos asume que el constructor de la red conoce ampliamente el tipo de resultado deseado. Todas las neuronas son de tipo análogo. Las entradas y las salidas toman valores positivos proporcionales a las frecuencias de disparo instantáneas de las neuronas biológicas. Se utiliza una variación de la regla de Hebb.

El neocognitrón fue construido como un modelo del sistema visual humano y es una red especializada con una complicada topología. Luego que el aprendizaje está completo, el sistema final es capaz de reconocer números manuscritos en varios estilos, aún cuando se presenten en cualquier posición de un archivo y con una distorsión considerable.

3.3.3 Probabilística (PNN)

En 1,987 varios grupos de personas introdujeron redes neurales probabilísticas "probabilistic neural networks" (PNN), a pesar que sus orígenes se remontan a la década de los 60's. La arquitectura de PNN está basada en memoria. Estas redes tienen una arquitectura similar a las de propagación hacia atrás, pero no son entrenadas con un proceso repetitivo y las neuronas internas tienen una función de activación diferente. La red es entrenada colocando los pesos de cada neurona interna para que reconozca los patrones de entrenamiento. Se necesita de una neurona de patrón separada para cada ejemplo del conjunto de entrenamiento.

Donald Specht¹³, de Lockheed Missile & Space Company, fue uno de los pioneros en premiar públicamente a las redes PNN. Las aplicaciones de PNN son relativamente pocas porque este modelo es muy nuevo, pero su uso como clasificador es prometedor. PNN es un modelo de múltiples estratos que usa neuronas no lineales. Se dice que es una red neural por su arquitectura, no por el método de aprendizaje. Calcula una función de probabilidad para cada salida cuando la red es creada.

13

Specht, Donald. **Probabilistic Neural Networks**. Neural Network magazine, Vol. 3. 1990. 109-118 pp.

El modelo PNN contiene cuatro estratos de neuronas: de entrada, de patrón, de sumación y de salida. Los estratos de entrada y de salida son similares a los de cualquier red BPN, mientras que los dos estratos interiores son diferentes. El estrato de patrón es usado para almacenar patrones. El número de neuronas en el estrato de patrón es igual al número de patrones de entrenamiento en la red. El estrato de sumación contiene neuronas que pueden servir como neuronas para categorizar.

Los modelos PNN construyen un patrón almacenando la matriz de pesos. El tamaño de la matriz está definido por el número de patrones de entrenamiento y el número de datos en cada patrón. Si hubieran 200 patrones y 45 datos en cada patrón, la matriz es de dimensiones $45 * 200$.

Las redes PNN pueden reconocer patrones individuales o categorizar patrones de entrada. Los patrones agrupados son conectados a través del estrato de sumación a la misma neurona en el estrato de salida. De esta forma, los grupos o neuronas similares pueden estar asociados con las categorías de salida. Si se reconocen patrones individuales, cada patrón almacenado está conectado a diferentes neuronas en el estrato de sumación, de forma que existe una correspondencia de uno a uno entre las entradas y salidas.

Las redes neurales probabilísticas tienen las propiedades siguientes:

- se aproximan a la regla de decisión óptima de Bayes (la estrategia de Bayes minimiza el riesgo esperado de la respuesta)
- se entrenan rápido en una pasada
- los datos del entrenamiento son almacenados
- es fácil de agregar nuevos datos en tiempo real
- la decisión está basada en los pesos de ejemplos almacenados en las vecindades
- la implementación del software sigue progresando

De igual forma, pueden ser descritas sus ventajas y desventajas:

- aprenden instantáneamente en tiempo real
- la prueba de velocidad puede ser lenta, a menos que la partición o hardware paralelo sea usado
- los requerimientos de memoria son muy altos
- puede calcular probabilidad además de tomar decisiones
- proveen un significado de retenimiento instantáneo, con lo cual pueden permitir a los patrones de clasificación reponer los datos perdidos

3.3.3.1 Red Neural de regresión general (GRNN)

Un algoritmo de aprendizaje modificado ha dado lugar a la utilización de una red basada en memoria que provee estimados de variables continuas y converge a la superficie de regresión que interesa (lineal o no lineal). La red tiene un algoritmo de aprendizaje de una pasada con una estructura altamente paralela. Aún con datos esparcidos en un espacio multidimensional, el algoritmo provee transiciones suaves de un valor observado a otro. La forma algorítmica puede ser usada para cualquier problema de regresión en el cual no se justifica asumir linealidad. La forma de red paralela puede ser usada para predicción, modelación, mapeo e interpolación o como un controlador.

La red neural de regresión general, mejor conocida como GRNN "general regression neural network", es similar en su forma a las PNN, pero tiene más capacidades. Pueden asociar entradas con salidas continuas o binarias. El modelo GRNN genera transiciones suaves de un patrón almacenado a otro, a través del uso de estimadores no paramétricos de funciones de densidad probabilísticas. El

particionamiento puede ser agregado al modelo GRNN; ésto se refiere al agrupamiento de patrones similares para requerir menos neuronas de dicho patrón.

Las ventajas relativas de GRNN, en comparación de otras técnicas de regresión no lineal, son las siguientes:

- la red aprende en una pasada a través de los datos y puede generalizar de ejemplos tan rápido como éstos son almacenados
- el estimado converge al significado de la superficie de regresión con mayor rapidez cuando más ejemplos son observados; forma superficies razonables de regresión basadas en sólo unos pocos ejemplos
- el estimado es limitado por el mínimo y máximo de las observaciones
- el estimado no puede converger a soluciones pobres que correspondan a mínimos locales del criterio de error (como sucede algunas veces con las técnicas iterativas)
- una simulación de software es fácil de escribir y de usar
- la red puede proveer un mapeo del conjunto de puntos de un ejemplo a otro. Si el mapeo es uno a uno, un mapeo inverso puede fácilmente ser generado del mismo ejemplo de puntos
- la versión particionada de GRNN limita el número de neuronas y provee (opcionalmente) de un mecanismo para olvidar los datos viejos

La principal desventaja de GRNN (sin particiones) en relación a otras técnicas, es que requiere cómputos grandes para evaluar nuevos puntos. Hay muchas formas de cubrir esta desventaja: una es usar las versiones particionadas de GRNN, otra es tomar ventaja de una estructura paralela inherente de esta red y diseñar circuitos semiconductores para el cómputo. Ambas, combinadas, proveen una adaptación rápida.

3.3.3.2 Otros modelos PNN

El modelo función básica radial "radial basis function" (RBF), también fue introducido en 1,987. Este modelo difiere de PNN en que usa otra fórmula para combinar las salidas de las neuronas de patrón en las salidas finales de la red.

Tanto RBF como las funciones básicas de Gauss "gaussian basis functions" (GBF) se asemejan a PNN en las funciones de densidad y arquitecturas que utilizan. Se diferencian en la forma que los patrones son almacenados y comparados. Los patrones son almacenados como puntos, en lugar de vectores. Las funciones RBF generalmente no calculan las fórmulas de igual forma y requieren información global para determinar sus parámetros. El estrato de salida de las RBF es entrenado usando un gradiente decreciente.

GBF usa una función de transferencia gaussiana para las neuronas. La salida cero de una neurona indica que el patrón es muy similar, cualquier otro número indica que el patrón es muy diferente. GBF permite el aprendizaje de cada patrón sin degradar la ejecución de patrones similares.

La red RCE fue introducida inicialmente en 1,980 por los doctores Leon Cooper, Charles Elbaum y Douglas Reilly; ha sido reconocida como nueva y muy distinta. La red RCE tiene tres estratos: uno de entrada, uno de salida y un estrato escondido. Cada neurona en el estrato escondido es un detector de formas. Las formas similares en los datos de entrada causan que se active una neurona. El entrenamiento se realiza por la modificación de los umbrales de las neuronas internas, usando un método de aprendizaje supervisado similar al de la propagación hacia atrás.

David Montana¹⁴ ha desarrollado una variante del modelo PNN, red neural probabilística con pesos (Weighted Probabilistic Neural Network), la cual utiliza una fórmula diferente para estimar la densidad de la probabilidad. Este modelo mejora la ejecución con patrones almacenados similares.

3.4 Comparaciones entre PNN y BPN

Los datos que a continuación son presentados fueron analizados por Lockheed Missiles & Space Company¹⁵. El proyecto de estudio se titula **Neural Network Technology**.

La implementación de las redes neurales se realiza de la siguiente forma:

1. BPN fue ejecutado en dos sistemas:
 Hecht-Nielsen Neurocomputer, coprocesador Anza Plus y con Neurosoft Software Brainmaker de California Scientific Software y ejecutado en IBM PC
2. PNN fue codificado en C por Lockheed y ejecutado en varias IBM PC

En BPN, la mitad del conjunto de entrenamiento fue usado para determinar la configuración del estrato escondido y para determinar el punto de finalización del entrenamiento.

Conjuntos idénticos de prueba fueron usados tanto para PNN como para BPN. En la siguiente tabla se ilustran las características de ambas redes.

Redes BPN	Redes PNN
basadas en la estrategia de Bayes	es la red mejor conocida y más ampliamente usada
mapea a una estructura típica de red neural	
clasificador resultante <ul style="list-style-type: none"> • se aproxima al óptimo de Bayes • se entrena rápido en una pasada • estima la probabilidad de decisión tanto como hacer la decisión • fácil de agregar nuevos datos en tiempo real 	clasificador resultante <ul style="list-style-type: none"> • adapta incrementalmente un largo tiempo de entrenamiento • puede quedarse bloqueado en unas pocas conexiones locales mínimas • requerimientos pequeños de memoria • tiempo de evaluación rápido
puede ser implementada en neuronas análogas o digitales, enteramente en paralelo	puede ser implementada en neuronas análogas o digitales, enteramente en paralelo

TABLA 3
 Características de BPN y de PNN

14 Montana, David. **A Weighted Probabilistic Neural Network**. Advances in Neural Information Processing Systems, Vol.4. 1110-1117 pp.

15 Rauch, Herbert E. **Introduction to Neural Networks and their Application to Control Systems**.

3.4.1 Precisión

PNN es ideal para explorar nuevas bases de datos y para técnicas de preprocesamiento que determinan precisiones de clasificación para nuevas aplicaciones; ésto se atribuye a que las PNN tienen una ventaja de velocidad en el tiempo de evaluación.

Las conclusiones de este estudio se pueden resumir de la siguiente forma:

- con un número pequeño de ejemplos presentados, la red PNN es más precisa
- con un número mayor de dimensiones, la red BPN tiene un rendimiento de precisión mayor

3.4.2 Velocidad

La información de esta sección está basada en la presentación realizada por Specht & Shapiro en la Conferencia Internacional de Redes Neuronales, ésta se llevó a cabo en París durante julio de 1,990. Ellos compararon los tiempos de entrenamiento de PNN versus BPN.

Las comparaciones fueron realizadas sobre las bases del número de sumas, restas y multiplicaciones en los algoritmos.

Se utilizó una base de datos de n patrones (n variada de 100 a 10,000) separada en dos subconjuntos, una mitad para entrenamiento, otra mitad para evaluación. Se configuró BPN con uno y dos estratos escondidos y los resultados obtenidos indican que PNN varía de 1 a 7 órdenes de magnitud más rápido que BPN.

Para bases de datos muy grandes, BPN es más rápido que la red básica PNN.

3.4.3 Conclusiones

La red neural probabilística (PNN) es ideal para explorar nuevas bases de datos:

- usualmente es más rápida que BPN
- es fácil de usar
- la generalización de la precisión es aproximadamente tan buena como BPN y a menudo mejor, aún en espacios altamente dimensionados con entradas irrelevantes

Para producción, PNN tiene tanto ventajas como desventajas:

- aprende instantáneamente en tiempo real,
- la velocidad probada puede ser lenta a menos que el particionamiento o paralelismo de hardware sea usado,
- puede calcular probabilidad, además de tomar decisiones,
- provee un significado para el entrenamiento instantáneo y permite la clasificación de patrones con datos extraviados.

4. SOFTWARE Y HARDWARE PARA RN

4.1 Introducción

Las redes neurales están siendo utilizadas y producidas en la forma de netware o de máquinas neurales:

- netware (software disponible):
es el software que emula las neuronas y sus interconexiones. Un aspecto importante de este software es que puede ser simulado en computadoras convencionales.
- máquina neural:
es hardware que modela el paralelismo de las neuronas. También se les conoce como neurocomputadoras (neurocomputers).

Las máquinas neurales han sido configuradas al nivel de chip, al nivel de tablero y al nivel de un sistema completo. Las máquinas neurales de uso general están disponibles para conectarse como coprocesadores para computadoras digitales. En este caso, se accesa la red neural como si fuera una subrutina que puede ser llamada cuando se desee. Las máquinas neurales están disponibles a operar lado a lado con computadoras de tecnología convencional.

En el verano de 1,988 la compañía NEC anunció que había desarrollado una computadora personal de red neural que usa el algoritmo de aprendizaje con retroalimentación. Los planes de NEC incluyeron el mercadeo y la venta de su producto sólo en Japón; calculaban que este producto se podría cotizar a 11,000 dólares.

En 1,988 alrededor de 10,000 paquetes de red neural para computadoras personales fueron vendidos en los Estados Unidos, muchos de ellos provenían del disco incluido en un manual publicado por James L. McClelland y David Rumelhart¹⁶.

En general, los programas de redes neurales que están comercialmente disponibles son aquellos que realizan simulaciones en escalas muy reales (basadas en software o en un uso específico).

James A. Anderson, profesor de Psicología, Ciencias del Conocimiento y Lingüística en la Universidad de Brown, hizo notar que no es la cantidad de MIPS (millones de instrucciones por segundo) de un dispositivo lo que determina si puede o no manejar una red neural. El efectivo manejo de memorias grandes y buenos compiladores, son mucho más importantes que la velocidad del CPU en la ejecución de cómputos rápidos en la red neural.

Las estaciones de trabajo como VAX, Sun y otras son ideales para la tarea, pero aún en estaciones de trabajo rápidas los trabajos pueden correr por horas. Muchas computadoras personales tienen que trabajar arduamente cuando se encaran con un sistema de 150,000 conexiones y 400 arreglos dimensionales, mientras que las estaciones de trabajo están diseñadas para trabajos largos. De nuevo, es importante qué tan bueno es el compilador. Según Anderson, las Sun y las VAX (especialmente éstas últimas) tienen maravillosos compiladores, pero también se puede realizar un trabajo de útil desarrollo en las computadoras personales aprendiendo, experimentando y tomando el tiempo para correr proyectos propios en lenguaje ensamblador.

16

McClelland, James y David E. Rumelhart. **Explorations in Parallel Distributed Processing. A Handbook of Models, Programs and Exercises.** Massachusetts, Institute of Technology, 1988

Sin embargo, no todos comparten la opinión de Anderson de utilizar sistemas pequeños debido a que tienen problemas corriendo ciertas aplicaciones en tiempo real. Las máquinas digitales simulan lo que está intrínseco en los sistemas paralelos y están limitadas por su propia velocidad y poder de procesamiento. Por ello, mientras los problemas largos son difíciles de correr en computadoras personales o estaciones de trabajo, tareas menos complejas se pueden ejecutar mejor. Actualmente existen alrededor de 300 compañías envueltas en la tecnología de redes neurales, muchas de ellas haciendo software de redes neurales para computadoras personales o estaciones de trabajo.

Entre los productos desarrollados con redes neurales están: software, shells, herramientas de desarrollo, chips y tableros de aceleración. Algunas compañías están en el proceso de desarrollo de equipo y chips de uso específico para aplicaciones de gran escala. En el año 1,990 se introdujeron muchos productos referentes a simulaciones.

4.2 Software disponible

A continuación se expondrán algunos de los paquetes de software existentes para redes neurales, para cada uno de ellos se describe la fuente consultada. Es necesario hacer mención que muchos otros trabajos sobre redes neurales se están llevando a cabo, por lo tanto, al momento que este trabajo sea consultado debe tomarse en cuenta que pueden existir nuevas y mejores opciones.

4.2.1 Autonet¹⁷

Es un tipo especial de software que automáticamente construye redes neurales a partir de ejemplos de entrenamiento, que incluyen las variables de entrada y los resultados esperados. Autonet determina la arquitectura de red más apropiada para calcular el resultado esperado. Luego de ésto, la red construida puede ser usada para calcular nuevos resultados a partir de otras variables de entrada.

El desarrollo y producción de Autonet fue posible por la colaboración del profesor Kenneth O. Cogger, Universidad de Kansas en Lawrence, con Peak Software Corporation. El profesor Cogger es el diseñador de la red y elaboró los algoritmos estadísticos usados en el programa. Peak Software desarrolló las interfaces con el usuario, las subrutinas de error y las gráficas.

Autonet está basado en las investigaciones realizadas por A.G. Ivakhnenko, un cibernético ucraniano, quien examinó lo que él denominara un método de grupo para el manejo de la información (Group Method of Data Handling, GMDH). Autonet incorpora muchos de estos descubrimientos, los cuales permiten la construcción de redes para problemas que manipulan una amplia variedad de tipos de datos.

4.2.2 Net Talk

Es una simulación de red neural de alimentación hacia adelante (BPN) de tres estratos. Fue desarrollada en 1,986 por Terry Sejnowski y Charles Rosenberg, ambos de Princeton. Net Talk ha sido entrenada para recibir un texto y convertirlo a sonidos "text to speech", en una Digital Equipment Corp., VAX. Crear un discurso a partir de palabras escritas es un problema que ha sido difícil de resolver con las técnicas de programación tradicional, sin embargo, Net Talk ha demostrado ser extremadamente precisa. Este software no utiliza reglas o un diccionario para pronunciar las palabras, lo que sabe es el

17

Autonet Versión 2.01. Peak Software Corporation, 1990

producto de su propio aprendizaje, conseguido a través de 500 palabras utilizadas como ejemplo. Se inicializa con un texto escrito, lo cual produce una secuencia de fonemas que son enviados a una bocina.

Lo sorprendente sobre Net Talk es el pequeño número de neuronas simuladas que fueron requeridas para su desarrollo; trabaja con un poco más de 300 neuronas, de las cuales 203 son entradas, 26 son salidas y 80 están en el estrato escondido.

El éxito que obtuvo este modelo dio al campo de las redes neurales gran credibilidad.

4.2.3 Neuroshell¹⁸

Neuroshell (versión 3.1) fue desarrollado por el Grupo Ward Systems y se ha catalogado como un software poderoso y efectivo. Fishman y Gallizzi, ambos de Eckerd College, lo recomiendan como un producto superior que refleja extraordinaria pericia en lo referente a la tecnología y la inusual competencia en el diseño e implementación.

Los datos cuantitativos no son restringidos a enteros, Neuroshell tiene la capacidad de manejar números de punto flotante. Se provee una versión separada de este programa, denominada (F)ANALOG.EXE para utilizarlo con problemas que requieren datos cuantitativos (o mixtos), opuesto a los problemas en los cuales todas las entradas y salidas son binarias, para lo cual utiliza el programa (F)BINARY.EXE; esta distinción provee una ejecución más rápida.

Neuroshell tiene la capacidad de graficar las relaciones entre las características e imprimir reportes. Específicamente, Neuroshell consiste de un intérprete de red neural y un programa utilitario. Los requerimientos mínimos para Neuroshell incluyen:

- una computadora compatible IBM
- sistema operativo DOS 3.0 o mayor
- 256K bytes de memoria RAM (problemas mayores requieren más de 640K)
- disco duro o discos flotantes
- ratón es optativo

El programa utilitario permite a las funciones graficar los datos de entrada, importar y exportar archivos de Lotus y archivos en ASCII, despliega los pesos de la red y elige al azar datos del archivo de entrenamiento para conformar el conjunto de prueba.

Productos adicionales del grupo Ward Systems incluyen una interfase utilitaria para importar datos de sistemas de base de datos, un sistema de tiempo de corrida que permite a los desarrolladores de software agregar funciones de la red neural a sus productos y un tablero de aceleración.

4.2.4 Brainmaker¹⁹

Brainmaker fue desarrollado por California Scientific Software, la versión 2.1 es la más reciente y ha sido comercializada a partir de 1,993.

18 Fishman, Mark B. y Edmund L. Gallizzi. **A Pair of Neural Packages for Research.** IEEE Computer, vol.24, No.1, enero 1991, págs. 107-108

19 *Ibidem*, págs. 108-110

Los datos cuantitativos no son restringidos a enteros en Brainmaker. Su uso es muy fácil, en el sentido que se puede evitar reformatear el conjunto de entrenamiento para agregar columnas binarias separas para cada uno de los valores categóricos. No se requiere indicar explícitamente los rangos de datos numéricos en el archivo de configuración. Es el único software que utiliza la función de transferencia de Gauss.

Para usar Brainmaker se invoca la utilidad Netmaker, la cual permite importar archivos ASCII u hojas electrónicas de trabajo. Se puede entonces designar las características como bases, resultados o simples columnas. Los datos pueden ser manipulados en forma general, como en una hoja de trabajo. Luego que el paquete genera automáticamente un archivo de definición de red, se puede proceder directamente a Brainmaker.

Brainmaker despliega cada una de las neuronas, demostrando su aprendizaje en una forma atractiva y visual (se puede evitar este paso para ejecutar más rápido el proceso). Este software puede desplegar histogramas, graficar las relaciones entre las características e imprimir reportes.

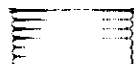
A continuación se describen las ventajas de Brainmaker²⁰:

- incluye Netmaker, un programa con el estilo de una hoja electrónica, que ayuda a la recolección de datos y a la creación de redes neurales
- no es necesario tener experiencia en programación. El usuario decide qué aprenderá la red mediante el uso de Netmaker y Brainmaker hace el resto
- incluye 3 tutores y 8 redes de ejemplo
- las opciones se presentan al usuario en forma de menú
- se permite la migración de imágenes, símbolos o números desde otro software, p.e. Lotus, dBase, etc.
- rapidez en el algoritmo de entrenamiento y la ejecución. Ejecuta más de 500,000 conexiones neurales por segundo en una IBM PC

Los requerimientos de este software son:

- computadora IBM PC, XT, AT, PS/2 o compatible
- memoria RAM de 512K bytes
- sistema operativo PC/DOS, MS/DOS 3.0, Windows 3.1 o mayor
- monitor monocromo o de color Hercules, CGA, EGA, VGA o MCGA

California Scientific Software tiene muchos productos de red neural, incluyendo Brainmaker versión 2.1 y Brainmaker Professional versión 1.5. Productos de soporte incluyen el programa de entrenamiento Hypersonic para hacer más rápido el proceso de entrenamiento; el programa de gráficas Net Plotter para presentar gráficamente la salida de Brainmaker; para asistir en el desarrollo de redes neurales Network Tools; el Intel i80170 Neural Network Support, el cual entrena redes para ejecutar en un chip de red neural; un tablero de aceleración para entrenar y correr la red arriba de 150 veces más rápido que en una XT, de acuerdo a lo publicado por la compañía.



4.2.5 Neuralworks Professional II²¹

Este paquete es producido por Neuralware, una compañía con una exitosa historia en desarrollar redes neurales para la resolución de problemas reales.

Neuralworks funciona correctamente en computadoras especificadas por la compañía, pero se necesitará más si se desea obtener cualquier valor de salida real. Los requerimientos mínimos son los siguientes:

- IBM PC o compatible
- memoria principal de 512K bytes
- monitor CGA blanco y negro
- disco duro de 3M bytes (no hay opción de disco flotante)
- disco flotante de 5¼, doble lado, doble densidad
- MS-DOS versión 3 o mayor

Es muy cómodo el uso de Neuralworks Professional con un ratón. Muchos tipos de PC pueden correr este software, incluyendo la Epson Equity II+, IBM PC AT, Zenith Z248 (8 y 12MHz) y Zenith Z150. No es recomendable la utilización de un equipo menor que AT por consideraciones de velocidad.

Los precios para el software de Neuralware dependen de la computadora en que se esté usando. Generalmente, mientras más poderosa es la computadora, más es el costo de la versión. Además de estar disponible para las series de IBM PC, Neuralworks trabaja con otras computadoras como Ncube, Sun y Macintosh. En la versión de IBM PC se puede empezar con Neuralworks Explorer, esta versión tiene pocas neuronas totales en la red y menos opciones que Neural Probes; sin embargo, es una opción barata si se desea sólo sentirse confortable con la herramienta y la tecnología. Explorer es altamente compatible con Neuralworks Professional II. Si el usuario desea invertir más en su compra, se puede agregar un generador de código. También se puede comprar un paquete dinámico de red para definición del usuario.

Neuralworks es una herramienta compleja con muchos niveles de menús y submenús. Neuralware se documenta en una forma gradual, incluyendo todos los detalles que le hacen una buena opción.

Neuralware anunció en enero de 1,991 la versión Neuralworks Professional II/Plus. Esta nueva versión incorpora algunos aspectos útiles, entre los más importantes están los siguientes:

- soporte para IBM RS 6000, coprocesador i860 y para chips de arquitectura Digital Neural Network
- una facilidad de explicación que usa técnicas matemáticas para determinar cuáles entradas tienen el mayor efecto en el resultado de la red
- siete nuevos tipos de red, incluyendo 8,000 variaciones del tipo con retroalimentación
- cuarenta nuevos comandos para dar al usuario más control sobre la arquitectura de la red
- un sistema de menú con íconos que realiza la interfase gráfica del usuario mucho más intuitiva, sobre todo para los usuarios sin experiencia

21

Raeth, Peter G. **Applying Technology with Neuralworks Professional II.** IEEE Computer, vol.24, No.1, enero 1991, págs. 111-112

4.3 Máquinas neurales

Como se mencionó anteriormente, una máquina neural es el hardware que modela el paralelismo de las neuronas. Algunos productos han sido desarrollados, sin embargo, especificaciones sobre éstos no se han publicado. A continuación se dará una lista de algunos productos disponibles en el mercado hasta enero de 1,991; de nuevo, debe tomarse en cuenta que la tecnología ha progresado y que actualmente muchos otros productos están a la venta.

- ANZA
coprocesador de red neural para AT compatibles; incluye software e interfase de programación
- ANZA PLUS
coprocesador de red neural para AT compatibles
- ANZA PLUS/VME
coprocesador de red neural para estaciones de trabajo Sun
- MD/210 Fuzzy Set Comparator
Implementación en electrónica de neuronas de Hopfield. Micro Devices
- N1000
Herramientas de desarrollo de red neural para aplicaciones de procesamiento de señal e imagen, incluyendo una computadora 80386
- CHIPS Reconocimiento Inteligente de Patrones
Almacenan una matriz de 1,000 por 64 elementos de peso y la multiplica con un vector de entrada
- SIGMA Neurocomputer Workstations
Sistemas basados en 80386 con procesador Delta, ANSkIt, Delta_C, Delta Macro y ANSpec
- DENDROS-1
Chip de red neural que produce el producto punto de entradas y pesos de conexión de 22 sinapsis
- DENDROS-1 Evaluation Board
Usa los chips Dendros-1 para crear una red neural basada en electrónica
- TRW MarkV Neural Processor
Sistema de procesamiento paralelo basado en MC68020. Corre en VAX/VMS, incluye herramientas
- Ward Systems's Interface
Interfase utilitaria para importar datos de sistemas de base de datos
- Ward Systems's Accelerator Board
Tablero de aceleración
- Intel i80170 Neural Network Support
Entrena redes para ejecutar en un chip de red neural
- California Scientific Software's Accelerator Board
Se utiliza para entrenar y correr una red neural arriba de 150 veces más rápido que en una XT, de acuerdo a lo relatado por la compañía+

5. APLICACIÓN

5.1 Introducción

En los capítulos anteriores se expuso al lector la teoría general sobre redes neurales, también se presentaron algunos productos de hardware y software disponibles en el mercado. En este capítulo se hace referencia a la utilización de un software de red neural para obtener la predicción de un índice de precios.

Existe una secuencia lógica para trabajar con redes neurales, la cual se compone de los siguientes pasos:

1. Definir el caso de estudio
2. Identificar la información
3. Reunir la información
4. Preparar la información
5. Crear los archivos de entrenamiento y pruebas
6. Construcción de la red
7. Entrenar la red neural
8. Probar la red neural
9. Ejecutar la red neural

En el contenido de este capítulo se exponen con claridad cada uno de los pasos antes indicados. En la sección 5.2 se trata la definición del caso de estudio. Los pasos 3 y 4 se desarrollan en la sección 5.3. En la sección 5.4 se tratan los aspectos relacionados con la preparación de los datos. El paso 5, el cual se refiere a la creación de los archivos de entrenamiento y pruebas, se desarrolla en la sección 5.5. En la sección 5.6 se realiza la construcción de la red. El entrenamiento de la red y sus resultados se presentan en la sección 5.7. Los resultados de las pruebas a la red se muestran en la sección 5.8. En la sección 5.9 se presentan las predicciones calculadas por la red, es decir, el paso 9.

Habiendo finalizado los pasos referentes a la red neural, se presenta un breve análisis de los resultados obtenidos en la predicción, éste en la sección 5.10.

5.2 Definición del caso de estudio

Se denomina índice de precios a una media ponderada de los precios de una serie de bienes y servicios. Para obtener un índice, se pondera cada uno de los precios según la importancia económica de los bienes incluidos. En el caso del índice de precios al consumidor (IPC), se asigna a cada bien un peso fijo proporcional a su importancia relativa en los presupuestos de gasto de los consumidores.²²

El IPC mide el costo de una canasta de mercado de bienes y servicios de consumo, entre los cuales se encuentran los precios de los alimentos, la ropa, la vivienda, los combustibles, el transporte, la asistencia médica, matrículas de estudio y otras mercancías que se compran diariamente.

Este índice es el más utilizado de la inflación y mide los cambios en el ingreso real del consumidor, así como el cambio relativo en los precios de determinados artículos o servicios adquiridos

²² Samuelson, Paul A. y William D. Nordhaus. **Economía**. Décimo Cuarta Edición. Editorial McGraw-Hill. España 1993, pág. 713.

en dos diferentes períodos de tiempo. El valor que tiene la mercancía en el primer período se denomina precio base y el valor que ésta tiene en un período posterior se denomina precio actual.

Para obtener el cambio de los precios, se mide el costo de una cesta de bienes y servicios de consumo, mejor conocida como canasta básica. Algunos de los bienes y servicios incluidos son: alimentos, ropa, vivienda, combustibles, transporte, asistencia médica y matrículas de estudio.

En Guatemala, el Instituto Nacional de Estadística (INE), específicamente la Sección de Precios al Consumidor, realiza los estudios concernientes al IPC. Para obtener este índice, el INE recoge mensualmente los precios de los productos que componen la canasta básica, durante todo el año. Los productos se clasifican en agrupaciones, grupos, subgrupos y artículos: actualmente se tienen 9 agrupaciones, 37 grupos, 112 subgrupos y 212 artículos en total.

¿Cómo se ponderan los diferentes precios cuando se elabora un índice de precios? Sería incorrecto simplemente sumarlos o ponderarlos por su masa o volumen. Los índices de precios se elaboran ponderando cada precio según la importancia económica del artículo en cuestión. En el caso del IPC, se asigna a cada artículo un peso fijo denominado ponderación, el cual es proporcional a su importancia relativa en los presupuestos de gasto de los consumidores. En Guatemala las ponderaciones más recientes de cada artículo se basan en la encuesta de ingresos y gastos en bienes de consumo de los meses de marzo 1983 hasta abril 1983.

El índice de precios al consumidor tiene una importancia relevante en lo que respecta a las ventas, pues mediante su predicción se puede pronosticar el monto de la demanda. Este dato es requerido por productores y consumidores, pues cada uno de ellos desea obtener mayores utilidades.

5.2.1 Hardware y software utilizado

Para obtener la predicción del IPC mediante la utilización de una red neural, se seleccionó el software Brainmaker, cuyas características se presentaron en la sección 4.2.4.

La aplicación se desarrolló en un ambiente de red, en el cual las características del hardware utilizado se presentan a continuación:

- estación de trabajo
 - computadora 486 de 66 Mhz
 - tarjeta de gráficas SuperVGA, 1 Mbyte memoria de video
 - monitor Super VGA
 - unidad de disco flotante 5¼, alta densidad
 - unidad de disco flotante 3½, alta densidad
 - tarjeta de red NE-2000
- servidor
 - server 486 de 100 Mhz
 - disco de 2 Gbytes
 - memoria RAM de 32 Mbytes
 - protocolo Novell 3.12
 - tarjeta FDDI y coaxial

5.3 Identificación de la información

Los datos se recogen en períodos mensuales y el tipo de la moneda que se utiliza es el Quetzal. A continuación se exponen brevemente los significados de los datos no calculados:

- la ponderación (A) se refiere a la importancia relativa que tiene el producto en el presupuesto del consumidor
- el precio base (B) es el valor en el cual se cotizó el producto en los meses de marzo '83 y abril '83
- el precio actual (D) es el valor en el cual se cotiza el producto en el mes que se realiza el análisis, es decir, el mes actual

En la tabla 4 se presentan los datos requeridos para realizar el cálculo del IPC.

Ponderación	Precio Base	Precio actual	Constante	Participación ponderada	I.P.C.
A	B	D	$C=A/B$	$E=D*B$	$F=E/A*100$

Cálculo del Índice de precios al Consumidor
Tabla 4

Para adoptar una convención y para claridad del lector, los datos se presentan en un orden cronológico ascendente, colocando al inicio los datos más antiguos y al final los datos más recientes.

5.4 Reuniendo los datos

Para los fines de esta aplicación se cuenta con la información proporcionada por el INE, la cual incluye datos mensuales para el cálculo del IPC y que corresponden a noviembre de 1,993 hasta octubre de 1,995.

Se ha seleccionado el azúcar para desarrollar la aplicación, debido a que representa un gran porcentaje de las exportaciones que nuestro país realiza y tiene una alta ponderación en la canasta básica del consumidor guatemalteco.

La tabla 5 muestra los datos proporcionados por el INE, siguiendo en forma fiel la representación que el Instituto utiliza. A continuación se define el significado de cada una de las columnas que componen la tabla:

- Año
año al cual corresponde la información proporcionada
- Mes
mes del año al cual corresponde la información proporcionada
- Ponderación
importancia relativa que tiene el producto en el presupuesto del consumidor guatemalteco
- Precio base
valor en el cual se cotizó el producto en el período marzo 1983 hasta abril 1983
- Constante
cociente de la ponderación y el precio base
- Precio actual
valor en el cual se cotiza el producto en el mes indicado

- Participación ponderada
producto del precio actual y la constante
- I.P.C.
índice de precios al consumidor

Año	Mes	Ponderación	Precio Base	Constante	Precio actual	Participación ponderada	I.P.C.
1992	noviembre	0.98073	0.17	5.769	1.01	5.82669	594.1176
	diciembre	0.98073	0.17	5.769	1.05	6.05745	617.6471
1993	enero	0.98073	0.17	5.769	1.1	6.3459	647.0588
	febrero	0.98073	0.17	5.769	1.1	6.3459	647.0588
	marzo	0.98073	0.17	5.769	1.1	6.3459	647.0588
	abril	0.98073	0.17	5.769	1.11	6.40359	652.9412
	mayo	0.98073	0.17	5.769	1.1	6.3459	647.0588
	junio	0.98073	0.17	5.769	1.11	6.40359	652.9412
	julio	0.98073	0.17	5.769	1.14	6.57666	670.5882
	agosto	0.98073	0.17	5.769	1.13	6.51897	664.7059
	septiembre	0.98073	0.17	5.769	1.13	6.51897	664.7059
	octubre	0.98073	0.17	5.769	1.15	6.63435	676.4706
	noviembre	0.98073	0.17	5.769	1.16	6.69204	682.3529
	diciembre	0.98073	0.17	5.769	1.19	6.86511	700
1994	enero	0.98073	0.17	5.769	1.25	7.21125	735.2941
	febrero	0.98073	0.17	5.769	1.24	7.15356	729.4118
	marzo	0.98073	0.17	5.769	1.26	7.26894	741.1765
	abril	0.98073	0.17	5.769	1.26	7.26894	741.1765
	mayo	0.98073	0.17	5.769	1.28	7.38432	752.9412
	junio	0.98073	0.17	5.769	1.31	7.55739	770.5882
	julio	0.98073	0.17	5.769	1.33	7.67277	782.3529
	agosto	0.98073	0.17	5.769	1.32	7.61508	776.4706
	septiembre	0.98073	0.17	5.769	1.32	7.61508	776.4706
	octubre	0.98073	0.17	5.769	1.32	7.61508	776.4706
	noviembre	0.98073	0.17	5.769	1.31	7.55739	770.5882
	diciembre	0.98073	0.17	5.769	1.32	7.61508	776.4706
1995	enero	0.98073	0.17	5.769	1.41	8.13429	829.4118
	febrero	0.98073	0.17	5.769	1.42	8.19198	835.2941
	marzo	0.98073	0.17	5.769	1.42	8.19198	835.2941
	abril	0.98073	0.17	5.769	1.47	8.48043	864.7059
	mayo	0.98073	0.17	5.769	1.45	8.36505	852.9412
	junio	0.98073	0.17	5.769	1.43	8.24967	841.1765
	julio	0.98073	0.17	5.769	1.43	8.24967	841.1765
	agosto	0.98073	0.17	5.769	1.43	8.24967	841.1765
	septiembre	0.98073	0.17	5.769	1.43	8.24967	841.1765
	octubre	0.98073	0.17	5.769	1.44	8.30736	847.0588

Ponderaciones del índice de precios al consumidor

PRODUCTO: Azúcar

Tabla 5

5.4 Preparación de los datos

Los datos presentados en la tabla 5, deben ser alimentados a la red neural. En su actual formato, la red no puede reconocer la diferencia entre los datos de entrada y los datos de salida. En las siguientes secciones se exponen los pasos que permiten preparar adecuadamente la información.

5.4.1 Definición de la matriz

La información recopilada se arregla en forma matricial, donde cada fila representa un ejemplo con preguntas y respuestas, mientras que cada columna es un componente de la pregunta, o bien, la respuesta.

En esta aplicación, cada fila corresponde a los datos requeridos para calcular el IPC durante un mes determinado. En lo referente a las columnas, algunas de ellas corresponden a entradas de la red y solamente una de ellas es el patrón de salida.

Luego de haber identificado la información requerida por la red neural, es necesario definir los datos de entrada y los patrones. En Brainmaker, a esta acción se le denomina etiquetar las columnas. Existen 4 tipos de etiquetas:

- not used (no usada): significa que la columna no será usada para crear la red neural
- Annotate (anotación): despliega la columna pero no la usa para realizar ningún cálculo
- input (entrada): utiliza la columna para realizar los cálculos
- Pattern (patrón): es la salida que la red debe calcular

En esta aplicación, se ha etiquetado cada columna como se muestra en la tabla 6.

Annote	NotUsed	NotUsed	NotUsed	Input	Input	Pattern
No.	Ponderación	Precio Base	Constante	Precio actual	Participación ponderada	I.P.C.
1	0.98073	0.17	5.769	1.01	5.82669	594.11
2	0.98073	0.17	5.769	1.05	6.05745	617.64
3	0.98073	0.17	5.769	1.1	6.3459	647.05
4	0.98073	0.17	5.769	1.1	6.3459	647.05
5	0.98073	0.17	5.769	1.1	6.3459	647.05
6	0.98073	0.17	5.769	1.11	6.40359	652.94
7	0.98073	0.17	5.769	1.1	6.3459	647.05
8	0.98073	0.17	5.769	1.11	6.40359	652.94
9	0.98073	0.17	5.769	1.14	6.57666	670.58
10	0.98073	0.17	5.769	1.13	6.51897	664.70
11	0.98073	0.17	5.769	1.13	6.51897	664.70
12	0.98073	0.17	5.769	1.15	6.63435	676.47
13	0.98073	0.17	5.769	1.16	6.69204	682.35
14	0.98073	0.17	5.769	1.19	6.86511	700
15	0.98073	0.17	5.769	1.25	7.21125	735.29
16	0.98073	0.17	5.769	1.24	7.15356	729.41
17	0.98073	0.17	5.769	1.26	7.26894	741.17
18	0.98073	0.17	5.769	1.26	7.26894	741.17

Annote	NotUsed	NotUsed	NotUsed	Input	Input	Pattern
No.	Ponderación	Precio Base	Constante	Precio actual	Participación ponderada	I.P.C.
19	0.98073	0.17	5.769	1.28	7.38432	752.94
20	0.98073	0.17	5.769	1.31	7.55739	770.58
21	0.98073	0.17	5.769	1.33	7.67277	782.35
22	0.98073	0.17	5.769	1.32	7.61508	776.47
23	0.98073	0.17	5.769	1.32	7.61508	776.47
24	0.98073	0.17	5.769	1.32	7.61508	776.47
25	0.98073	0.17	5.769	1.31	7.55739	770.58
26	0.98073	0.17	5.769	1.32	7.61508	776.47
27	0.98073	0.17	5.769	1.41	8.13429	829.41
28	0.98073	0.17	5.769	1.42	8.19198	835.29
29	0.98073	0.17	5.769	1.42	8.19198	835.29
30	0.98073	0.17	5.769	1.47	8.48043	864.70
31	0.98073	0.17	5.769	1.45	8.36505	852.94
32	0.98073	0.17	5.769	1.43	8.24967	841.17
33	0.98073	0.17	5.769	1.43	8.24967	841.17
34	0.98073	0.17	5.769	1.43	8.24967	841.17
35	0.98073	0.17	5.769	1.43	8.24967	841.17
36	0.98073	0.17	5.769	1.44	8.30736	847.05

Datos etiquetados
Tabla 6

La columna correspondiente al número correlativo se marca como annote porque se requiere que durante las etapas de entrenamiento y pruebas se despliegue el número de ejemplo.

Las columnas precio actual y participación ponderada se etiquetan como entradas porque es a través de ellas que se realiza el cálculo del IPC.

Las columnas ponderación, precio base y constante se marcan como not used, ésto debido a que luego de seleccionar las entradas, éstas no son requeridas para el cálculo del IPC. En las siguientes tablas, estas columnas no serán mostradas por razones de espacio, sin embargo, es elección del usuario presentarlas o no a la red neural.

5.4.2 Datos representados como cambios

Los indicadores deben ser presentados a la red como cambios entre un período y otro, en esta aplicación se presentan como los cambios de un mes a otro. Utilizar estos cambios, en lugar de los valores reales, es importante por dos razones:

- Una red neural entrenada con valores reales no puede reconocer un valor futuro fuera del rango de los datos de entrenamiento. Por ejemplo, si se entrena una red para predecir el IPC dentro de un rango de 594 a 847, no será capaz de predecir un valor para el IPC que sea igual o menor a 593, o bien, mayor o igual a 848.
- Se provee un rango menor de datos, lo cual siempre es mejor.

Los cambios se calculan tanto para las entradas como para el patrón de salida. El cambio para una determinada columna se realiza restando el valor de un período N al valor del período N+1. Por ejemplo: el cambio en el precio actual correspondiente a los meses de noviembre 1,992 a diciembre 1,992 se calcula restando 1.01 de 1.05, el valor es 0.04. El valor 0.04 es colocado en la fila correspondiente al período 1, o sea, noviembre 1,992, pues representa el incremento que se dio a partir de noviembre en el precio actual. Debido a que todos los cambios entre el período N y el período N+1 se registran en el período N, el valor de cambio para octubre de 1,995 no puede ser calculado.

Las columnas precio actual y participación ponderada, anteriormente etiquetadas como input, deben ahora ser etiquetadas not used; las columnas diferencia precio actual y diferencia participación ponderada son etiquetadas como input. Este cambio se debe a que ya no se utilizan los valores reales para realizar la predicción, en adelante se usarán los valores de cambio. De igual forma, la columna correspondiente al IPC, la cual mostró los valores reales, mostrará ahora los valores de cambio. Nuevamente, las columnas etiquetadas como not used, no serán mostradas en las tablas, sin embargo queda a discreción del usuario presentarlas a la red neural. Ver tabla 7.

Annote	Not used	Input	Not used	Input	Pattern
No.	Precio actual	Diferencia Precio actual	Participación ponderada	Diferencia participación ponderada	I.P.C.
1	1.01	0.04	5.82669	0.23076	23.52941
2	1.05	0.05	6.05745	0.28845	29.41176
3	1.1	0	6.3459	0	0
4	1.1	0	6.3459	0	0
5	1.1	0.01	6.3459	0.05769	5.882353
6	1.11	-0.01	6.40359	-0.05769	-5.88235
7	1.1	0.01	6.3459	0.05769	5.882353
8	1.11	0.03	6.40359	0.17307	17.64706
9	1.14	-0.01	6.57666	-0.05769	-5.88235
10	1.13	0	6.51897	0	0
11	1.13	0.02	6.51897	0.11538	11.76471
12	1.15	0.01	6.63435	0.05769	5.882353
13	1.16	0.03	6.69204	0.17307	17.64706
14	1.19	0.06	6.86511	0.34614	35.29412
15	1.25	-0.01	7.21125	-0.05769	-5.88235
16	1.24	0.02	7.15356	0.11538	11.76471
17	1.26	0	7.26894	0	0
18	1.26	0.02	7.26894	0.11538	11.76471
19	1.28	0.03	7.38432	0.17307	17.64706
20	1.31	0.02	7.55739	0.11538	11.76471
21	1.33	-0.01	7.67277	-0.05769	-5.88235
22	1.32	0	7.61508	0	0
23	1.32	0	7.61508	0	0
24	1.32	-0.01	7.61508	-0.05769	-5.88235
25	1.31	0.01	7.55739	0.05769	5.882353
26	1.32	0.09	7.61508	0.51921	52.94118
27	1.41	0.01	8.13429	0.05769	5.882353
28	1.42	0	8.19198	0	0
29	1.42	0.05	8.19198	0.28845	29.41176

Annote	Not used	Input	Not used	Input	Pattern
No.	Precio actual	Diferencia Precio actual	Participación ponderada	Diferencia participación ponderada	I.P.C.
30	1.47	-0.02	8.48043	-0.11538	-11.7647
31	1.45	-0.02	8.36505	-0.11538	-11.7647
32	1.43	0	8.24967	0	0
33	1.43	0	8.24967	0	0
34	1.43	0	8.24967	0	0
35	1.43	0.01	8.24967	0.05769	5.882353
36	1.44	*****	8.30736	*****	*****

Valores de cambio
Tabla 7

5.4.3 Tendencia

Las redes neurales se utilizan en múltiples aplicaciones en las cuales se sabe que existe una tendencia. Una red neural basada en este tipo de información, como lo es la predicción del IPC, tiene un objetivo: usar datos históricos para predecir los cambios en el valor de dicho índice.

Para que una red neural sea capaz de identificar una tendencia, es necesario que en cada ejemplo se presenten, además de las entradas anteriormente identificadas, los datos históricos; es decir, en cada fila de la matriz presentada deben considerarse los datos históricos que influyen en el patrón de salida. Con el propósito de reflejar las tendencias en datos históricos se crea una columna adicional para cada indicador.

Se ha considerado que en el cálculo del IPC existe una tendencia del precio actual y de la participación ponderada. En esta aplicación se está prediciendo el valor del IPC del próximo mes, por lo tanto, deben ser creadas dos nuevas columnas: una que corresponda al precio actual del próximo mes y que se denomina precio futuro; la segunda columna corresponde a la participación ponderada del próximo mes y se denomina participación ponderada futuro.

Debido a que no existen datos posteriores al mes de octubre 1,995, el valor de las columnas precio futuro y participación ponderada futuro para ese período no puede ser calculada. Estas dos columnas deben ser consideradas ahora como entradas a la red y por lo tanto son etiquetadas input. A continuación se presenta la tabla con las dos nuevas columnas, incluyendo aquellas etiquetadas not used, para que el lector identifique claramente la relación de la información.

Annote	Not used	Input	Input	Not used	Input	Input	Pattern
No.	Precio actual	Diferencia precio actual	Precio futuro	Participación ponderada	Diferencia participación ponderada	Participación ponderada futuro	I.P.C.
1	1.01	0.04	1.05	5.82669	0.23076	6.05745	23.52941
2	1.05	0.05	1.1	6.05745	0.28845	6.3459	29.41176
3	1.1	0	1.1	6.3459	0	6.3459	0
4	1.1	0	1.1	6.3459	0	6.3459	0
5	1.1	0.01	1.11	6.3459	0.05769	6.40359	5.882353
6	1.11	-0.01	1.1	6.40359	-0.05769	6.3459	-5.88235

Annote	Not used	Input	Input	Not used	Input	Input	Pattern
No.	Precio actual	Diferencia precio actual	Precio futuro	Participación ponderada	Diferencia participación ponderada	Participación ponderada futuro	I.P.C.
7	1.1	0.01	1.11	6.3459	0.05769	6.40359	5.882353
8	1.11	0.03	1.14	6.40359	0.17307	6.57666	17.64706
9	1.14	-0.01	1.13	6.57666	-0.05769	6.51897	-5.88235
10	1.13	0	1.13	6.51897	0	6.51897	0
11	1.13	0.02	1.15	6.51897	0.11538	6.63435	11.76471
12	1.15	0.01	1.16	6.63435	0.05769	6.69204	5.882353
13	1.16	0.03	1.19	6.69204	0.17307	6.86511	17.64706
14	1.19	0.06	1.25	6.86511	0.34614	7.21125	35.29412
15	1.25	-0.01	1.24	7.21125	-0.05769	7.15356	-5.88235
16	1.24	0.02	1.26	7.15356	0.11538	7.26894	11.76471
17	1.26	0	1.26	7.26894	0	7.26894	0
18	1.26	0.02	1.28	7.26894	0.11538	7.38432	11.76471
19	1.28	0.03	1.31	7.38432	0.17307	7.55739	17.64706
20	1.31	0.02	1.33	7.55739	0.11538	7.67277	11.76471
21	1.33	-0.01	1.32	7.67277	-0.05769	7.61508	-5.88235
22	1.32	0	1.32	7.61508	0	7.61508	0
23	1.32	0	1.32	7.61508	0	7.61508	0
24	1.32	-0.01	1.31	7.61508	-0.05769	7.55739	-5.88235
25	1.31	0.01	1.32	7.55739	0.05769	7.61508	5.882353
26	1.32	0.09	1.41	7.61508	0.51921	8.13429	52.94118
27	1.41	0.01	1.42	8.13429	0.05769	8.19198	5.882353
28	1.42	0	1.42	8.19198	0	8.19198	0
29	1.42	0.05	1.47	8.19198	0.28845	8.48043	29.41176
30	1.47	-0.02	1.45	8.48043	-0.11538	8.36505	-11.7647
31	1.45	-0.02	1.43	8.36505	-0.11538	8.24967	-11.7647
32	1.43	0	1.43	8.24967	0	8.24967	0
33	1.43	0	1.43	8.24967	0	8.24967	0
34	1.43	0	1.43	8.24967	0	8.24967	0
35	1.43	0.01	1.44	8.24967	0.05769	8.30736	5.882353
36	1.44	*****	*****	8.30736	*****	*****	*****

Indicadores de la tendencia
Tabla 8

5.4.4 Datos preparados

Ahora se cuenta con toda la información necesaria para alimentar a la red neural, únicamente resta eliminar las filas cuyos datos no pudieron ser calculados. En esta aplicación, la fila que se elimina corresponde al mes de octubre 1,995, período 36. La matriz de datos, incluyendo sólo las columnas etiquetadas annote e input se presenta en la tabla 9.

Annote	Input	Input	Input	Input	Pattern
No.	Diferencia precio actual	Precio futuro	Diferencia participación ponderada	Participación ponderada futuro	I.P.C.
1	0.04	1.05	0.23076	6.05745	23.52941
2	0.05	1.1	0.28845	6.3459	29.41176
3	0	1.1	0	6.3459	0
4	0	1.1	0	6.3459	0
5	0.01	1.11	0.05769	6.40359	5.882353
6	-0.01	1.1	-0.05769	6.3459	-5.88235
7	0.01	1.11	0.05769	6.40359	5.882353
8	0.03	1.14	0.17307	6.57666	17.64706
9	-0.01	1.13	-0.05769	6.51897	-5.88235
10	0	1.13	0	6.51897	0
11	0.02	1.15	0.11538	6.63435	11.76471
12	0.01	1.16	0.05769	6.69204	5.882353
13	0.03	1.19	0.17307	6.86511	17.64706
14	0.06	1.25	0.34614	7.21125	35.29412
15	-0.01	1.24	-0.05769	7.15356	-5.88235
16	0.02	1.26	0.11538	7.26894	11.76471
17	0	1.26	0	7.26894	0
18	0.02	1.28	0.11538	7.38432	11.76471
19	0.03	1.31	0.17307	7.55739	17.64706
20	0.02	1.33	0.11538	7.67277	11.76471
21	-0.01	1.32	-0.05769	7.61508	-5.88235
22	0	1.32	0	7.61508	0
23	0	1.32	0	7.61508	0
24	-0.01	1.31	-0.05769	7.55739	-5.88235
25	0.01	1.32	0.05769	7.61508	5.882353
26	0.09	1.41	0.51921	8.13429	52.94118
27	0.01	1.42	0.05769	8.19198	5.882353
28	0	1.42	0	8.19198	0
29	0.05	1.47	0.28845	8.48043	29.41176
30	-0.02	1.45	-0.11538	8.36505	-11.7647
31	-0.02	1.43	-0.11538	8.24967	-11.7647
32	0	1.43	0	8.24967	0
33	0	1.43	0	8.24967	0
34	0	1.43	0	8.24967	0
35	0.01	1.44	0.05769	8.30736	5.882353

Archivo de datos

Tabla 9

5.5 Creación archivos de entrenamiento y pruebas

En la tabla 9 se muestra toda la información disponible para realizar la predicción del IPC. Para poder obtener esta predicción calculada por la red, es necesario entrenarla, realizar luego pruebas y finalmente ejecutarla.

Hasta ahora se cuenta con 35 ejemplos para presentar a la red, de ellos, se utilizarán algunos para entrenamiento y el resto para realizar las pruebas. No puede especificarse con precisión la cantidad de ejemplos que deben ser utilizados para el entrenamiento, pues éste difiere de una a otra aplicación. El software Brainmaker sugiere que se utilice el 90% de los ejemplos en la fase de entrenamiento y el 10 % restante para probar la red.

Los datos de la tabla 9 se presentan a Brainmaker como el archivo de datos, con el cual se generan los archivos de entrenamiento y pruebas. En cada uno de estos archivos se elige aleatoriamente determinado número de ejemplos, siguiendo la misma estructura en que presenta el archivo de datos. La cantidad de ejemplos que Brainmaker coloca en cada archivo depende del porcentaje indicado por el usuario. En el caso de esta aplicación se indicó que el 83% corresponde al archivo de entrenamiento y el 17% al archivo de pruebas; es decir, 29 ejemplos se utilizarán para entrenamiento y 6 ejemplos para pruebas de la red.

En la tabla 10 muestra el archivo de entrenamiento y la tabla 11 el archivo de pruebas.

Annote	Input	Input	Input	Input	Pattern
No.	Diferencia precio actual	Precio futuro	Diferencia participación ponderada	Participación ponderada futuro	I.P.C.
1	0.04	1.05	0.23076	6.05745	23.52941
2	0.05	1.1	0.28845	6.3459	29.41176
4	0	1.1	0	6.3459	0
5	0.01	1.11	0.05769	6.40359	5.882353
6	-0.01	1.1	-0.05769	6.3459	-5.88235
7	0.01	1.11	0.05769	6.40359	5.882353
8	0.03	1.14	0.17307	6.57666	17.64706
10	0	1.13	0	6.51897	0
11	0.02	1.15	0.11538	6.63435	11.76471
12	0.01	1.16	0.05769	6.69204	5.882353
13	0.03	1.19	0.17307	6.86511	17.64706
14	0.06	1.25	0.34614	7.21125	35.29412
16	0.02	1.26	0.11538	7.26894	11.76471
17	0	1.26	0	7.26894	0
18	0.02	1.28	0.11538	7.38432	11.76471
19	0.03	1.31	0.17307	7.55739	17.64706
20	0.02	1.33	0.11538	7.67277	11.76471
22	0	1.32	0	7.61508	0
23	0	1.32	0	7.61508	0
24	-0.01	1.31	-0.05769	7.55739	-5.88235
25	0.01	1.32	0.05769	7.61508	5.882353
26	0.09	1.41	0.51921	8.13429	52.94118
28	0	1.42	0	8.19198	0
29	0.05	1.47	0.28845	8.48043	29.41176
30	-0.02	1.45	-0.11538	8.36505	-11.7647
31	-0.02	1.43	-0.11538	8.24967	-11.7647
32	0	1.43	0	8.24967	0
34	0	1.43	0	8.24967	0

Annote	Input	Diferencia	Precio futuro	Diferencia participación	Participación ponderada futuro	Pattern
No.	Diferencia	Precio futuro	Diferencia participación	Participación ponderada futuro	I.P.C.	
35	0.01	1.44	0.05769	8.30736	5.882353	

Datos de entrenamiento

Tabla 10

Annote	Input	Diferencia	Precio futuro	Diferencia participación	Participación ponderada futuro	Pattern
No.	Diferencia	Precio futuro	Diferencia participación	Participación ponderada futuro	I.P.C.	
3	0	1.1	0	6.3459	0	
9	-0.01	1.13	-0.05769	6.51897	-5.88235	
15	-0.01	1.24	-0.05769	7.15356	-5.88235	
21	-0.01	1.32	-0.05769	7.61508	-5.88235	
27	0.01	1.42	0.05769	8.19198	5.882353	
33	0	1.43	0	8.24967	0	

Datos para pruebas

Tabla 11

5.6 Construcción

Haciendo referencia a la tabla 9, hasta ahora se han definido 4 entradas a la red y 1 salida. Es necesario definir otros elementos, como lo son el tipo de red, cantidad de neuronas escondidas y la función de transferencia. Estos y otros aspectos son tratados en las tres secciones siguientes.

5.6.1 Tipo de red

El software Brainmaker utiliza una red de propagación hacia atrás. Esta red aprende por ejemplo y repetición, es entrenada con un conjunto de ejemplos. Cada uno de estos ejemplos es presentado a la red una y otra vez, hasta que encuentra los pesos adecuados de sus conexiones.

El factor para determinar cuándo debe parar la red es conocido como la tolerancia del error, el cual puede ser especificado en términos del porcentaje del rango de salida. En esta aplicación se obtienen los valores máximos y mínimos de los datos presentados en la tabla 9, columna correspondiente al IPC. El valor mínimo de salida es -11.7647 y el valor máximo es 52.94118; el rango se calcula como la diferencia del valor absoluto de ambas cifras:

$$52.94118 - 11.7647 = 41.17648$$

Si la tolerancia al error se fija en 0.1, como lo sugiere Brainmaker, el error de tolerancia es equivalente a multiplicar el rango por la tolerancia:

$$0.1 * 41.17648 = 4.117648$$

5.6.2 Función de transferencia

Brainmaker permite elegir cualquiera de las siguientes funciones de transferencia: lineal, umbral, step, Sigmoid o Gauss.

Debido a que no se cuenta con un criterio específico para elegir la función de transferencia y tomando en cuenta que este trabajo será consultado por lectores interesados en el desempeño de las redes neurales, se ha decidido realizar el estudio con cada una de las opciones existentes. Para ilustrar al usuario el desempeño de la red con cada una de estas funciones, se especifican los resultados obtenidos durante las etapas de entrenamiento, pruebas y ejecución.

5.6.3 Neuronas escondidas

La cantidad de neuronas escondidas que deba tener una red no puede ser definido con precisión, pues depende de la aplicación que se desarrolla. En términos generales, a menor cantidad de ejemplos, menor cantidad de neuronas escondidas.

Como se expuso en la sección 2.8, se sugiere que el número inicial de estratos escondidos se calcule en alguna de las tres formas diferentes. De los valores obtenidos aplicando estas formas se seleccionó el rango para el número de las neuronas escondidas, el cual es [1,2]. Al igual que con la elección de la función de transferencia, al usuario se mostrará el resultado de las etapas de entrenamiento, pruebas y ejecución, con 1 y 2 neuronas escondidas.

En las tres secciones siguientes se presentan los cálculos para cada una de las formas, con el propósito de aclarar al lector cómo se obtuvo el rango antes indicado.

El número de conexiones de la red depende del número de neuronas escondidas. La forma de realizar las conexiones varía de una a otra red, de tal forma que el número total no puede ser calculado a través de una fórmula para todas las redes. A continuación se muestra la cantidad de conexiones que Brainmaker realiza según el número de neuronas escondidas.

ESTRATO	NEURONAS	CONEXIONES
Entrada	4	0
1	10	50
Salida	1	11

Número de conexiones con 1 neurona escondida
Tabla 12

ESTRATO	NEURONAS	CONEXIONES
Entrada	2	0
1	10	30
2	10	110
Salida	1	11

Número de conexiones con 2 neuronas escondidas
Tabla 13

5.6.3.1. Menos de la mitad del número de entradas y salidas

Haciendo referencia a la información de la tabla 9, la cantidad de entradas que tiene la red es 4 y la cantidad de salidas es 1. Por lo tanto, el número de neuronas escondidas debe cumplir con:

$$\#_neuronas_escondidas < (4 + 1)/2$$

Es decir, el número de las neuronas escondidas debe ser menor que 2.5 neuronas. Debido a sólo se puede implementar un número entero de neuronas, se puede deducir que el rango sugerido para las neuronas escondidas es [1,2].

5.6.3.2. Cinco al diez por ciento del número de los casos de entrenamiento

En la tabla 10 se presentan 29 ejemplos para entrenamiento. El número sugerido de neuronas escondidas tiene como valor mínimo $29 * 5\% = 1.45$ y como valor máximo $29 * 10\% = 2.9$. Al igual que se mencionó en la sección anterior, el número de neuronas debe ser entero. El rango sugerido para las neuronas escondidas es [1,3].

5.6.3.3. Número de ejemplos dividido el número de entradas y multiplicado por la tolerancia

El número de ejemplos es 29, el número de entradas es 4 y la tolerancia de entrenamiento que sugiere el software Brainmaker es 0.1. Por lo tanto, el número sugerido de neuronas escondidas es:

$$(29/4) * 0.1 = 0.725$$

Debido a que el número de neuronas debe ser entero, el número calculado se aproxima a 1.

5.7 Entrenando la red neural

Como se indicó en las secciones 5.6.2 y 5.6.3, se presentan los resultados del entrenamiento para una red neural construida con 1 neurona y luego para una red construida con 2 neuronas. Para cada una de estas redes, además, se presentan los resultados obtenidos con cada una de las diferentes funciones de transferencia. En total se mostrará el resultado de 10 redes neurales diferentes.

Para el entrenamiento se presentan los datos de la tabla 10, son 29 ejemplos en total. La tolerancia de entrenamiento es 0.1 y el error de tolerancia es 4.117648.

La mejor red obtenida durante el entrenamiento no siempre es aquella que todos los ejemplos cumplen con la tolerancia de entrenamiento. En algunas ocasiones es mejor utilizar una red que cumple con un porcentaje de los ejemplos de entrenamiento.

En esta aplicación se presentaron 5 redes en las cuales el 100% de los ejemplos no cumplió con la tolerancia de entrenamiento. El criterio utilizado para detener el entrenamiento de estas redes fue el siguiente: si durante 5 minutos el IPC calculado por la red no sufre de variación, se detiene el entrenamiento de la red. En la tabla 14 se encontrará la cantidad de ejemplos que en cada una de las redes cumplieron con la tolerancia.

Como se mencionó anteriormente, se presentan 10 resultados de entrenamiento; cada uno de estos resultados está compuesto por un conjunto de datos, los cuales se indican a continuación:

- Función de transferencia
nombre de la función de transferencia utilizada por las neuronas de la red
- Neuronas escondidas
número total de neuronas escondidas utilizadas en la red
- Total de ejemplos
número total de ejemplos presentados a la red durante el entrenamiento, tomando en cuenta la tolerancia al error
- Ejemplos buenos
número total de ejemplos, de los 29 presentados, para los cuales la salida de la red cumple con la tolerancia
- IPC
valor del IPC calculado por la red neural
- Tiempo
total de tiempo utilizado en el entrenamiento, bajo las condiciones especificadas en la sección 5.2.1

Función de transferencia	Neuronas escondidas	Total de ejemplos	Ejemplos buenos	I.P.C.	Tiempo (hh:mm:ss)
Lineal	1	334,741	23	-529.38	00:47:00
	2	86,327	23	-529.38	00:12:42
umbral	1	5,539	29	8.0041	00:00:49
	2	343,040	29	11.416	00:54:32
step	1	754,964	5	-11.758	02:01:20
	2	391,006	2	-11.758	01:01:00
Sigmoid	1	2,523	29	3.5019	00:00:21
	2	2,030	29	2.3645	00:00:17
Gauss	1	242,900	4	-11.758	00:41:05
	2	12,934	29	5.8873	00:01:55

Resultados del entrenamiento
Tabla 14

5.8 Pruebas de la red neural

Las pruebas realizadas a la red son muy importantes, porque independientemente de los resultados durante el entrenamiento, ésta es la única forma para determinar qué tan bien realiza la red su ejecución.

Si una red se entrena bien, es decir, si todos los ejemplos cumplieron con la tolerancia de entrenamiento, pero no tiene buenos resultados durante las pruebas, es posible que haya memorizado las respuestas. La habilidad de una red neural para reconocer información que no ha visto antes se llama generalización. El objetivo de la etapa de pruebas es determinar si la red aprendió a generalizar en lugar de memorizar.

La memorización puede ocurrir por diversas razones, pero usualmente ocurre cuando no se muestran suficientes ejemplos para lograr el aprendizaje.

Durante la etapa de pruebas se presenta a la red un conjunto de ejemplos que son nuevos. El formato de los datos debe ser igual al formato presentado durante el entrenamiento, pero los ejemplos que se presentan deben ser diferentes a aquellos correspondientes al entrenamiento. Para cada uno de

estos ejemplos la red calcula una salida, si esta salida cumple con la tolerancia de prueba, el ejemplo es considerado bueno, de lo contrario se considera malo. Luego de presentar todos los ejemplos a la red, el usuario conoce la cantidad de ejemplos buenos y malos; es una decisión personal considerar si la red se ha desempeñado bien o mal durante las pruebas.

En esta aplicación se prueba la ejecución de las 10 redes neurales entrenadas con antelación. A cada una de estas redes se le mostraron los datos de la tabla 11, totalizando 6 ejemplos presentados a cada red. La tolerancia de pruebas utilizada es 0.100, pues es la sugerida por Brainmaker.

Para resumir las pruebas realizadas a cada una de las redes neurales, se muestra al lector una tabla con el desempeño que cada red tuvo con todos los ejemplos presentados, mostrando el IPC calculado en cada uno de ellos. En total se presentan 10 tablas y cada una de ellas incluye la siguiente información:

- Ejemplo No.
número con el cual se identifica al ejemplo, tomado de la tabla 11
- Resultado de la red
valor de cambio calculado por la red neural, referente al IPC
- Ejemplo bueno/malo
indica si el ejemplo fue tomado como bueno o malo, considerando la tolerancia de prueba

Ejemplo No.	Resultado de la red	Ejemplo bueno/malo
3	-529.38	bueno
9	-529.38	bueno
15	-529.38	bueno
21	-529.38	bueno
27	-529.38	bueno
33	-529.38	bueno

Pruebas red lineal con 1 neurona
Tabla 15

Ejemplo No.	Resultado de la red	Ejemplo bueno/malo
3	-529.38	bueno
9	-529.38	bueno
15	-529.38	bueno
21	-529.38	bueno
27	-529.38	bueno
33	-529.38	bueno

Pruebas red lineal con 2 neuronas
Tabla 16

Ejemplo No.	Resultado de la red	Ejemplo bueno/malo
3	-3.3540	bueno
9	-8.0773	bueno
15	-8.4564	bueno
21	-9.5780	bueno
27	8.0041	bueno
33	-0.1313	bueno

Pruebas red umbral con 1 neurona
Tabla 17

Ejemplo No.	Resultado de la red	Ejemplo bueno/malo
3	-6.1659	bueno
9	-6.1659	bueno
15	-6.1659	bueno
21	-6.1659	bueno
27	11.416	bueno
33	-6.1659	bueno

Pruebas red umbral con 2 neuronas
Tabla 18

Ejemplo No.	Resultado de la red	Ejemplo bueno/malo
3	-11.578	bueno
9	-11.578	bueno
15	-11.578	bueno
21	-11.578	bueno
27	-11.578	bueno
33	-11.578	bueno

Pruebas red step con 1 neurona
Tabla 19

Ejemplo No.	Resultado de la red	Ejemplo bueno/malo
3	-11.578	bueno
9	-11.578	bueno
15	-11.578	bueno
21	-11.578	bueno
27	-11.578	bueno
33	-11.578	bueno

Pruebas red step con 2 neuronas
Tabla 20

Ejemplo No.	Resultado de la red	Ejemplo bueno/malo
3	0.3583	bueno
9	-4.8073	bueno
15	-4.9969	bueno
21	-4.7599	bueno
27	3.5809	bueno
33	-0.8422	bueno

Pruebas red Sigmoid con 1 neurona
Tabla 21

Ejemplo No.	Resultado de la red	Ejemplo bueno/malo
3	1.2587	bueno
9	-3.0538	bueno
15	-4.0016	bueno
21	-4.1596	bueno
27	2.2381	bueno
33	-1.6321	bueno

Pruebas red Sigmoid con 2 neuronas
Tabla 22

Ejemplo No.	Resultado de la red	Ejemplo bueno/malo
3	-11.758	bueno
9	-11.758	bueno
15	-11.758	bueno
21	-11.758	bueno
27	-11.758	bueno
33	-11.758	bueno

Pruebas red Gauss con 1 neurona
Tabla 23

Ejemplo No.	Resultado de la red	Ejemplo bueno/malo
3	2.1275	bueno
9	-2.9590	bueno
15	-5.6761	bueno
21	-5.3918	bueno
27	8.9361	bueno
33	-0.4631	bueno

Pruebas red Gauss con 2 neuronas
Tabla 24

Es importante resaltar que los resultados en las tablas 15, 16, 19, 20 y 23 demuestran que las correspondientes redes no aprendieron durante el entrenamiento, ellas memorizaron los resultados. La tabla 14 indica que estas redes no cumplieron con tener el 100% de los ejemplos considerados buenos.

Fácilmente se puede concluir que las redes Lineal con 1 y 2 neuronas escondidas, step con 1 y 2 neuronas escondidas y Gauss con 1 neurona escondida, memorizaron la salida, pues para todos los ejemplos presentados el valor de la columna 'Resultado obtenido por la red' en las tablas 15, 16, 19, 20 y 23 es el mismo.

Luego de la fase de entrenamiento el usuario de la red puede determinar que ésta requiere algunas modificaciones en su construcción. Si éste es el caso, luego de realizar los cambios a la red, necesariamente debe realizar las etapas de entrenamiento y pruebas; en caso contrario, la red está lista para obtener el cálculo de cualquier dato.

En esta aplicación no se realizan cambios en ninguna red, pues desde el inicio se realizó el entrenamiento y las pruebas para todas las construcciones de redes neurales deseadas. Las redes que memorizaron los resultados no serán utilizadas en la etapa de ejecución de la red, ya que se sabe anticipadamente que el resultado será el mismo para cualquier entrada.

5.9 Ejecución de la red

Ejecutar la red significa obtener de ella los resultados para los cuales fue entrenada. La red, ahora entrenada y probada, necesita entradas y no requiere que se presenten las salidas. Haciendo referencia a la tabla 9, solamente las primeras 5 columnas deben ser presentadas a la red; es decir, no se presenta a la red la columna etiquetada Pattern.

Como se expuso en la sección anterior, solamente 5 redes neurales serán consideradas en la etapa de ejecución: umbral de 1 y 2 neuronas escondidas, Sigmoid de 1 y 2 neuronas escondidas y Gauss de 2 neuronas escondidas.

Para la predicción del IPC deben tomarse en cuenta valores no presentados anteriormente a la red, es decir, los valores para la predicción deben diferir a aquellos correspondientes a las entradas que se presentaron durante el entrenamiento. Se tiene esta restricción, pues existe la probabilidad que el resultado obtenido corresponda al patrón presentado en el entrenamiento. No existe ningún inconveniente si se utilizan los datos de las pruebas para ejecutar la red, ya que la red no realizó ningún ajuste en los pesos de las conexiones cuando estos ejemplos fueron presentados.

Ahora se tiene la tarea de definir los valores utilizados para predecir el IPC: debido a que únicamente se cuenta con la información mostrada en la tabla 9, se utilizarán los ejemplos de las pruebas para la ejecución, los cuales fueron presentados en la tabla 5.5.2.

Considerando que no se requiere presentar el patrón para la ejecución de la red, se puede tomar la estructura mostrada en la tabla 11, etiquetando la columna correspondiente al IPC como not used (antes etiquetada Pattern). El formato de los datos presentados para la ejecución de la red se muestra en la tabla 25.

Annote	Input	Input	Input	Input	Not used
No.	Diferencia precio actual	Precio futuro	Diferencia participación ponderada	Participación ponderada futuro	I.P.C.
3	0	1.1	0	6.3459	0
9	-0.01	1.13	-0.05769	6.51897	-5.88235

Annote	Input	Input	Input	Input	Not used
No.	Diferencia precio actual	Precio futuro	Diferencia participación ponderada	Participación ponderada futuro	I.P.C.
15	-0.01	1.24	-0.05769	7.15356	-5.88235
21	-0.01	1.32	-0.05769	7.61508	-5.88235
27	0.01	1.42	0.05769	8.19198	5.882353
33	0	1.43	0	8.24967	0

Datos para la ejecución
Tabla 25

Luego de presentar estos datos a la red, se obtuvo el pronóstico del IPC. Para que el lector pueda seguir con claridad los resultados obtenidos por cada una de las 5 redes ejecutadas, se presentan los resultados en las 5 tablas siguientes. En cada tabla se incluye la siguiente información:

- Ejemplo No.
número con el cual se identifica al ejemplo, tomado de la tabla 11, la cual muestra los datos de prueba
- Resultado de la red
valor calculado por la red neural
- IPC del período
valor real del IPC en el período, según la tabla 6, información proporcionada por el INE
- IPC pronosticado
valor del IPC calculado por la red neural durante la prueba

Ejemplo No.	Resultado de la red	IPC del período	IPC pronosticado
3	-3.3540	647.05	643.696
9	-8.0773	670.58	662.5027
15	-8.4564	735.29	726.8336
21	-9.5780	782.35	772.772
27	8.0041	829.41	821.4059
33	-0.1313	841.17	841.0387

Pruebas red umbral con 1 neurona
Tabla 26

Ejemplo No.	Resultado de la red	IPC del período	IPC pronosticado
3	-6.1659	647.05	640.8841
9	-6.1659	670.58	664.4141
15	-6.1659	735.29	729.1241
21	-6.1659	782.35	776.1841
27	11.416	829.41	840.826
33	-6.1659	841.17	835.0041

Pruebas red umbral con 2 neuronas
Tabla 27

Ejemplo No.	Resultado de la red	IPC del período	IPC pronosticado
3	0.3583	647.05	647.4083
9	-4.8073	670.58	665.7727
15	-4.9969	735.29	730.2931
21	-4.7599	782.35	777.5901
27	3.5809	829.41	832.9909
33	-0.8422	841.17	840.3278

Pruebas red Sigmoid con 1 neurona
Tabla 28

Ejemplo No.	Resultado de la red	IPC del período	IPC pronosticado
3	1.2587	647.05	648.3087
9	-3.0538	670.58	667.5262
15	-4.0016	735.29	731.2884
21	-4.1596	782.35	778.1904
27	2.2381	829.41	831.6481
33	-1.6321	841.17	839.5379

Pruebas red Sigmoid con 2 neuronas
Tabla 29

Ejemplo No.	Resultado de la red	IPC del período	IPC pronosticado
3	2.1275	647.05	649.1775
9	-2.9590	670.58	667.621
15	-5.6761	735.29	729.6139
21	-5.3918	782.35	776.9582
27	8.9361	829.41	838.3461
33	-0.4631	841.17	840.7069

Pruebas red Gauss con 2 neuronas
Tabla 30

5.10 Análisis de los resultados

En esta sección se realiza el análisis de los resultados obtenidos por las redes ejecutadas en la sección anterior. El objetivo de analizar estos resultados es determinar la red con el mejor desempeño para la aplicación, es decir, cuál de las 5 redes ha pronosticado de mejor forma el IPC.

Con el propósito de realizar este análisis de la forma más clara posible, se muestra al lector una gráfica de barras, con el IPC real y el IPC calculado por la red; ambos valores fueron mostrados en las tablas 26 a 30.

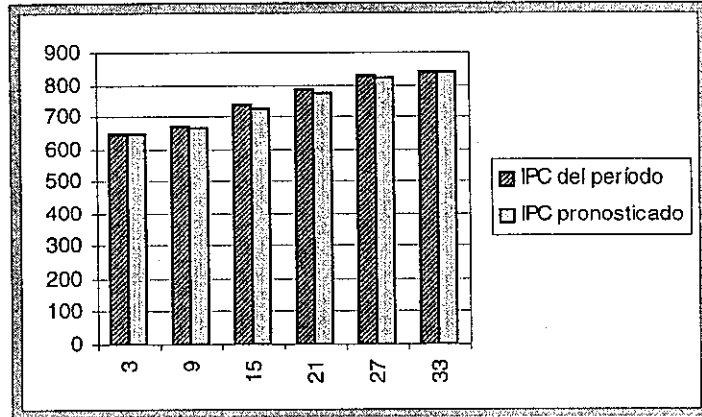


FIGURA 23
Red umbral con 1 neurona escondida

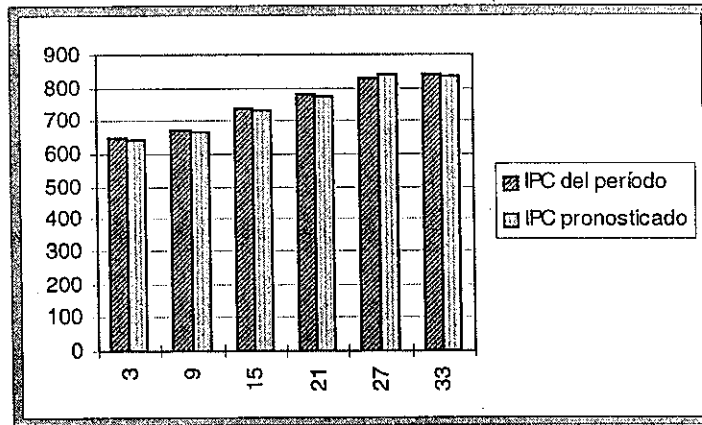


FIGURA 24
Red umbral con 2 neuronas escondidas

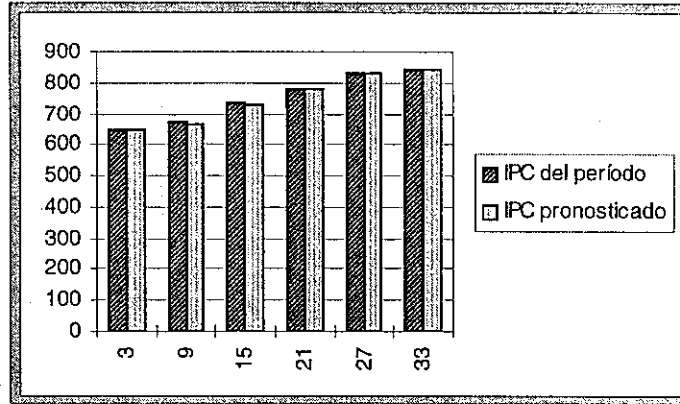


FIGURA 25
Red Sigmoid con 1 neurona

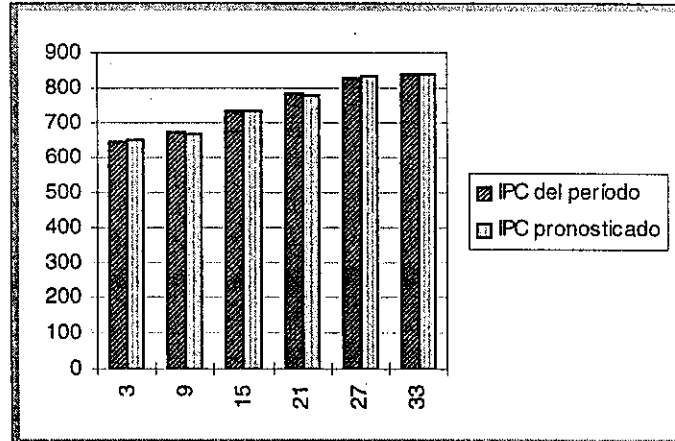


FIGURA 26
Red Sigmoid con 2 neuronas escondidas

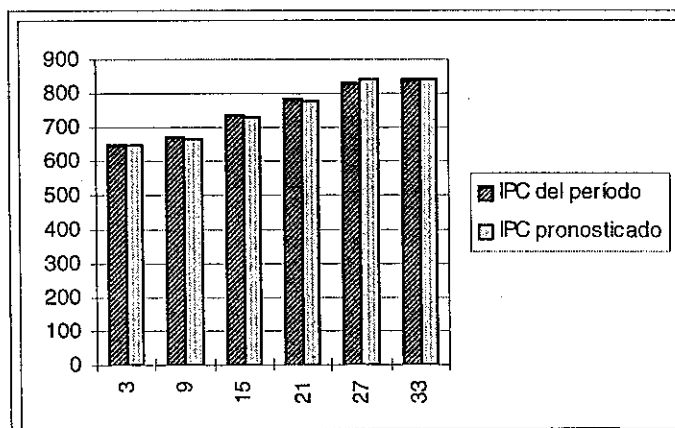


FIGURA 27
Red Gauss con 2 neuronas escondidas

Para elegir la red que mejor desempeño presenta en una aplicación el usuario debe elegir aquella cuyos pronósticos se acerquen más a la realidad. En esta aplicación, específicamente, los valores pronosticados por cada una de las 5 redes no difiere significativamente; por lo tanto, el rango de los valores pronosticados no se considera un factor determinante para seleccionar la red neural que se aplique mejor en la predicción del IPC.

Otro factor que puede ser determinante para elegir la red es el tiempo de entrenamiento, en el caso de esta aplicación se puede recurrir a los datos de la tabla 14, en la cual se muestran los resultados del entrenamiento. De esta tabla se han extraído los datos de las 5 redes cuyos resultados se analizan, dando origen a la tabla 31.

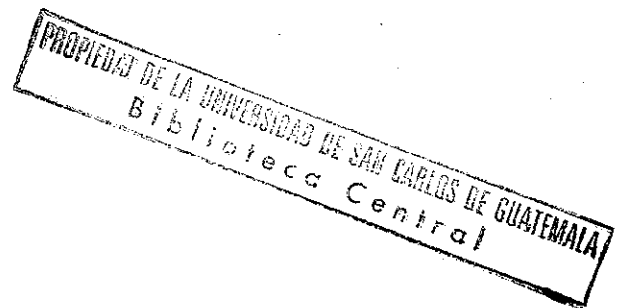
Función de transferencia	Neuronas escondidas	Tiempo (hh:mm:ss)
umbral	1	00:00:49
	2	00:54:32
Sigmoid	1	00:00:21
	2	00:00:17
Gauss	2	00:01:55

Resultados del entrenamiento de las 5 redes analizadas
Tabla 31

Observando estos resultados se puede concluir que, con excepción de la red umbral con 2 neuronas escondidas, el resto de ellas tienen tiempos de entrenamiento similares. La red Sigmoid con 2 neuronas escondidas presentó el menor tiempo de entrenamiento; por lo tanto, se puede seleccionar esta red para obtener el pronóstico del IPC.

Conclusiones

- En Guatemala existe un amplio campo para la aplicación de las redes neurales artificiales, un ejemplo de ello es la aplicación que se ha incluido en esta tesis. Sin embargo, el acceso a este tipo de información está restringido únicamente a aquellas personas con la posibilidad de comunicación a fuentes del extranjero.
- En el pensum de los estudiantes que se preparan para ser profesionales en informática, no se incluye un estudio profundo sobre las redes neurales artificiales, por lo cual ellos no identifican esta poderosa herramienta como una solución en sus aplicaciones.
- Las redes neurales artificiales no son adecuadas para realizar cálculos con un alto nivel de procesamiento numérico y aunque la salida de la red se obtiene rápidamente, es posible que definir la mejor red tome más tiempo de lo que se ha contemplado. Este tipo de red puede obtener mejores resultados que los métodos tradicionales, en tareas que requieren conjuntos de datos incompletos, información confusa o contradictoria.
- Las redes neurales artificiales pueden ser utilizadas en aplicaciones muy impresionantes, pero no pueden replicar todos los aspectos del cerebro humano porque éste es demasiado complejo y aún no se puede definir con precisión su funcionamiento.
- Una de las mejores formas de asegurarse que la red neural está dando respuestas correctas, es incorporar chequeos de validación en el conjunto de entrenamiento. Por ejemplo, se puede incluir un segundo patrón que sea opuesto al patrón actual: si el patrón actual es un número, sólo será necesario multiplicarlo por -1 y declarar esta nueva columna como patrón. Cuando se corre la red ambos patrones deben ser aproximadamente opuestos, si no lo son, se puede sospechar que la red está adivinando la respuesta.
- No existe una fórmula para determinar con precisión el número de neuronas escondidas y el número de ejemplos requeridos por la red neural artificial, pero en términos generales, mientras más ejemplos y menos neuronas escondidas se utilicen, el desempeño de la red es mejor. Si el número de neuronas escondidas se aproxima al número de ejemplos, o bien, el número de ejemplos es insuficiente, se incrementa la posibilidad de la memorización durante el entrenamiento.



Recomendaciones

- El uso de las redes neurales provee grandes ventajas y se recomienda su utilización a todas aquellas personas, de cualquier campo que manejen información confusa o incompleta. También serán beneficiados aquellos que busquen respuestas rápidas cuya obtención no necesite fórmulas o reglas. Algunas de las áreas de aplicación para las redes neurales son: reconocimiento de patrones, predicción de tendencias, modelación, reducción del ruido, análisis de imágenes, clasificación y evaluación.
- En el pensum de las carreras de informática debe incluirse el tema de las redes neurales artificiales, específicamente en un curso concerniente a inteligencia artificial. Es necesario motivar a los estudiantes a que busquen información en fuentes del extranjero, para actualizar y reforzar los conocimientos adquiridos durante su formación académica.
- Una red neural compuesta por varios estratos, debe utilizar la misma función de transferencia por todas las neuronas de un mismo estrato; sin embargo, la función de transferencia sí puede variar de un estrato a otro.
- Las redes neurales que utilizan el algoritmo de propagación hacia atrás son ampliamente usadas, pero deben considerarse dos aspectos importantes. El primero de ellos es que este tipo de red sólo percibe un ejemplo a la vez, no tienen memoria para recordar el ejemplo anterior. El segundo es que puede tomar un largo número de iteraciones para converger a la solución deseada.
- Para obtener una red neural que cumpla con los fines de la aplicación, es necesario que el número de ejemplos presentados en el entrenamiento y las pruebas pertenezcan a un rango de dos a diez veces el número de neuronas en la red y que sea mayor a tres.
- Para definir la mejor red neural de una aplicación específica es recomendable iniciar el entrenamiento con pocas neuronas escondidas y a medida que se obtienen los resultados se puede incrementar el número de las neuronas, si se considera necesario.
- Se recomienda realizar las pruebas a la red neural luego del entrenamiento, pues sólo de esta forma el usuario puede decidir con precisión si la red logró aprender correctamente o no. Para considerar exitoso el aprendizaje de la red, es esencial que el sistema ejecute correctamente la clasificación de ejemplos de prueba, los cuales difieren de los ejemplos de entrenamiento.

Bibliografía

Ahiya, Vyay. **Design and Analysis of Computer Communication Networks**. s.d.e.

Arbid, Michael A. **Cerebros, Máquinas y Matemáticas**. España: Alianza Editorial, s.a.

Arbid, Michael A. **Cibernética**. s.d.e.

Cordón Engel, Ricardo Alfredo. **Un Modelo Matemático del Cerebro**. Tesis: Matemática, Universidad del Valle de Guatemala, 1979

Lawrence, Jeannette. **Introduction to Neural Networks. Design, Theory and Applications**. California Scientific Software. Nevada City, C.A. Estados Unidos de América, s.e. 1994

Maselli I., Raúl A. **Inteligencia Artificial**. Tesis: Computación, Universidad Francisco Marroquín, 1983

MC Cielland, James L. **Explorations in Parallel Distributed Processing**. Estados Unidos de América, 1988

Obermeier, Klaus K. **Neural Networks. In Depth**. Revista BYTE. Volumen 14, número 8. Estados Unidos de América, agosto 1989. 214-245 pp.

Rumelhart, David E. **Parallel Distributed Processing**. Volumen 1. Foundations. Estados Unidos de América, 1987

Rumelhart, David E. **Parallel Distributed Processing**. Volumen 2. Psychological and Biological Models. Estados Unidos de América, 1987

Samuelson, Paul A. y William D. Nordhaus. **Economía**. 14 edición. España: Editorial Mc_Graw Hill, 1994. 713-714 pp.

Stevenson, William J. **Estadística para Administración y Economía**. México: Editorial Harla, 1981. 481-484 pp.

Wolfram, Stephen. "Programación en Ciencias y en Matemáticas". **Revista Investigación y Ciencia**, 1987