



Universidad Nacional Autónoma de México
Programa de Posgrado en Ciencias de la Administración

T e s i s

**Redes neuronales artificiales para la toma de
decisiones: el caso de tutores del Programa de Posgrado en
Ciencias de la Administración de la UNAM**

Que para obtener el grado de:

**Doctor en Ciencias de la
Administración**

Presenta: Graciela Bribiesca Correa

Tutor: Dra. Nadima Simón Domínguez

México, D.F., 6 noviembre de 2006



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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



Universidad Nacional Autónoma de México

Programa de Posgrado en Ciencias de la Administración

Facultad de Contaduría y Administración

Facultad de Química

Instituto de Investigaciones Sociales

Instituto de Investigaciones Jurídicas

T e s i s

**Redes neuronales artificiales para la toma de
decisiones: el caso de tutores del Programa de Posgrado en
Ciencias de la Administración de la UNAM**

Que para obtener el grado de:

**Doctor en Ciencias de la
Administración**

Presenta: Graciela Bribiesca Correa

Tutor: Dra. Nadima Simón Domínguez

México, D.F., 6 de noviembre de 2006

Agradecimientos:

Al Maestro Arturo Díaz Alonso, mi agradecimiento por su amistad y por el apoyo que siempre me ha brindado, usted es un ejemplo a seguir como Académico y Universitario.

A mi Directora y amiga Dra. Nadima Simón Domínguez, por su paciencia, entusiasmo, cariño y profesionalismo.

A mi compañera y amiga Dra. Lourdes Álvarez por su tiempo dedicado a este trabajo.

A mis padres a quienes quiero por siempre.

A mis hermanos Raúl, Jorge, Ernesto y Silvia por todo su cariño y confianza.

A mis hijos Héctor y Graciela por darme fuerzas para seguir adelante.

A Willy por su apoyo

Contenido

INTRODUCCIÓN

MARCO TEÓRICO Y CONCEPTUAL

CAPÍTULO 1. SISTEMAS DE INFORMACIÓN

- 1.1 TOMA DE DECISIONES
 - 1.1.1 *Toma de decisiones en la organización*
 - 1.1.2 *El modelo racional en la toma de decisiones*
- 1.2 IMPORTANCIA DE LOS SISTEMAS DE INFORMACIÓN EN LA TOMA DE DECISIONES
- 1.3 LOS SISTEMAS DE INFORMACIÓN EN LAS ORGANIZACIONES
- 1.4 PRINCIPALES TIPOS DE SISTEMAS EN LAS ORGANIZACIONES
- 1.5 CLASIFICACIÓN DE SISTEMAS DE INFORMACIÓN DE ACUERDO A SU APLICACIÓN
- 1.6 RELACIÓN DE LOS DIFERENTES TIPOS DE SISTEMAS DE INFORMACIÓN EN FUNCIÓN DEL TIPO DE DECISIÓN Y NIVEL DE ORGANIZACIÓN
 - 1.6.1 *Nivel operativo y decisiones estructuradas*
 - 1.6.2 *Nivel operativo y decisiones semiestructuradas*
 - 1.6.3 *Nivel de gestión y decisiones estructuradas*
 - 1.6.4 *Nivel de gestión y decisiones no estructuradas*
 - 1.6.5 *Nivel estratégico*
- 1.7 LOS SISTEMAS DE INFORMACIÓN CON TECNOLOGÍA DE INTELIGENCIA ARTIFICIAL.

CAPÍTULO 2. EL PROGRAMA DE POSGRADO EN CIENCIAS DE LA ADMINISTRACIÓN DE LA UNAM

- 2.1 ANTECEDENTES
- 2.2 SITUACIÓN ACTUAL DEL PROGRAMA DE POSGRADO EN CIENCIAS DE LA ADMINISTRACIÓN DE LA UNAM
 - 2.2.1 *Matrícula*
 - 2.2.2 *Planta académica*
 - 2.2.3 *Graduación*
 - 2.2.4 *Líneas de investigación.*
- 2.3 ANTECEDENTES DEL PROGRAMA TUTORAL DE POSGRADO.

CAPÍTULO 3. LA INTELIGENCIA ARTIFICIAL Y LAS REDES NEURONALES ARTIFICIALES

- 3.1 FUNDAMENTOS BIOLÓGICOS DE LAS REDES NEURONALES
 - 3.1.1 *El cerebro y la neurona biológica*
 - 3.1.2 *Estructura de una neurona*
 - 3.1.3 *La misión de las neuronas*
- 3.2 REDES NEURONALES BIOLÓGICAS
- 3.3 REDES NEURONALES ARTIFICIALES
 - 3.3.1 *¿Qué es una red neuronal artificial?*
- 3.4. CARACTERÍSTICAS DE LAS NEURONAS ARTIFICIALES
- 3.5 PRIMEROS MODELOS COMPUTACIONALES
- 3.6 SISTEMAS NEURONALES ARTIFICIALES
- 3.7 CLASIFICACIÓN DE LOS MODELOS DE RNA
 - 3.7.1 *El número de capas*
 - 3.7.2 *La topología de la red*
 - 3.7.3 *El tipo de respuesta*
 - 3.7.4 *La forma de los datos de entrada y salida*
- 3.8 COMPONENTES DE UNA RED NEURONAL ARTIFICIAL
- 3.9 CARACTERÍSTICAS DE LOS MODELOS DE REDES NEURONALES ARTIFICIALES MÁS IMPORTANTES Y ALGUNAS DE SUS APLICACIONES.

TABLA 1. CARACTERÍSTICAS DE LOS MODELOS DE REDES NEURONALES MÁS IMPORTANTES

- 3.10 REDES NEURONALES CON CONEXIONES HACIA DELANTE

TABLA 2. TÉRMINOS Y EXPRESIONES EQUIVALENTES EN AMBOS CAMPOS.

TABLA 3. EQUIVALENCIAS EXISTENTES ENTRE DIVERSAS TÉCNICAS ESTADÍSTICAS Y LAS RNA.

- 3.10.2 *Perceptron simple*
- 3.10.5 *Propagación de los patrones de entrada.*
- 3.10.6 *Cálculo de las activaciones para un Perceptron multicapa con dos neuronas de entrada, dos ocultas y una salida.*
- 3.10.7 *Diseño de la arquitectura del PERCEPTRON multicapa.*
- 3.11 ALGORITMO DE RETROPROPAGACIÓN (BACKPROPAGATION)
- 3.12 OBTENCIÓN DE LA REGLA DELTA GENERALIZADA.
- 3.13 DERIVADA DE LA FUNCIÓN DE ACTIVACIÓN.
- 3.14 CAPACIDAD DE GENERALIZACIÓN.
- 3.15 DIFERENTES APLICACIONES DE LAS REDES NEURONALES ARTIFICIALES

TABLA 4. PRODUCTOS COMERCIALES ESPECIALIZADOS DE REDES NEURONALES CON SU CORRESPONDIENTE APLICACIÓN

- 3.16 VENTAJAS DE LAS REDES NEURONALES

CAPÍTULO 4. METODOLOGÍA PARA DISEÑAR LA ARQUITECTURA DE UNA RED NEURONAL ARTIFICIAL QUE AYUDE AL PROCESO DE TOMA DE DECISIONES.

- PASO 1. PLANTEAMIENTO DEL PROBLEMA.
- PASO 2. REQUERIMIENTOS DE HARDWARE Y SOFTWARE PARA EJECUTAR LA RED NEURONAL.
- 2.1 DIVERSAS ARQUITECTURAS PARA EJECUTAR REDES NEURONALES ARTIFICIALES:

TABLA 5. HERRAMIENTAS DE DESARROLLO SOFTWARE COMERCIALES.

- PASO 3. REVISIÓN BIBLIOGRÁFICA
- PASO 4. ELECCIÓN DEL MODELO DE RNA
- 4.1 DEFINICIÓN DEL MODELADO DE LA RED NEURONAL ARTIFICIAL (MICROESTRUCTURA)
 - 4.1.1 *Unidades de procesamiento.*
 - 4.1.2 *Estado de la activación de cada neurona.*
 - 4.1.3 *Función de salida o transferencia*
 - 4.1.4 *Conexiones entre neuronas*
 - 4.1.5 *Función o regla de activación*
 - 4.1.5.1 *Función de activación escalón*
 - 4.1.5.2 *Función de activación lineal o identidad*
 - 4.1.5.3 *Función de activación lineal-mixta*
 - 4.1.5.4 *Función de activación sigmoideal*
 - 4.1.6 *Regla de aprendizaje*
- 4.2 DEFINICIÓN LA TOPOLOGÍA DE LA RED NEURONAL ARTIFICIAL (MESOESTRUCTURA).
 - 4.2.1 *Número de niveles o capas*
 - 4.2.2 *Grado de conectividad*
 - 4.2.3 *Tipo de conexiones entre neuronas*
 - 4.2.3.1 *Redes monocapa (1 capa).*
 - 4.2.3.2 *Redes multicapa*

TABLA 6. REDES NEURONALES MONOCAPA MÁS CONOCIDAS.

- 4.2.3.2 *Redes multicapa*

TABLA 7. REDES NEURONALES MULTICAPA MÁS CONOCIDAS.

- PASO 5. TIPO DE ASOCIACIÓN ENTRE LAS INFORMACIONES DE ENTRADA Y SALIDA.
- 5.1 REDES HETEROASOCIATIVAS
- 5.2 REDES AUTOASOCIATIVAS

TABLA 8. CLASIFICACIÓN DE LOS MODELOS DE REDES NEURONALES EN FUNCIÓN DEL TIPO DE ASOCIACIÓN.

- PASO 6. DATOS DISPONIBLES Y SELECCIÓN DE VARIABLES RELEVANTES.
- 6.1 INTRODUCCIÓN
- 6.2 REPRESENTACIÓN DE LA INFORMACIÓN DE ENTRADA Y SALIDA.
 - 6.2.1 *Valores reales y continuos*
 - 6.2.2 *Valores discretos*

TABLA 9. CLASIFICACIÓN DE LOS MODELOS DE REDES NEURONALES EN FUNCIÓN DEL TIPO DE REPRESENTACIÓN DE LAS INFORMACIONES DE ENTRADA Y SALIDA.

6.3 CARACTERÍSTICAS DE LOS MODELOS DE REDES NEURONALES MÁS CONOCIDOS

TABLA 10. RESUMEN DE LAS CARACTERÍSTICAS DE LOS MODELOS DE REDES MÁS CONOCIDOS.

PASO 7. MECANISMO DE APRENDIZAJE

7.1 REDES CON APRENDIZAJE SUPERVISADO

7.1.1 *Aprendizaje por corrección de error*

7.1.2 *Aprendizaje por refuerzo*

7.1.3 *Aprendizaje estocástico*

TABLA 11. MODELOS DE REDES MÁS CONOCIDOS CON APRENDIZAJE SUPERVISADO.

7.2 REDES CON APRENDIZAJE NO SUPERVISADO

b) *Familiaridad.*

c) *Análisis de las componentes principales.*

d) *Agrupamiento.*

e) *Prototipado.*

f) *Codificación.*

g) *Extracción y relación de las características.*

7.2.1 *Aprendizaje Hebbiano*

7.2.2 *Aprendizaje competitivo y cooperativo*

Tabla 12. Redes con aprendizaje no supervisado más conocidas

7.3 ELECCIÓN DE LOS CONJUNTOS DE APRENDIZAJE Y PRUEBA

PASO 8. PREPROCESAMIENTO

8.1 *Escalado por patrones (por filas)*

8.2 *Escalado global por entradas (por columnas)*

PASO 9. PROCESO DE ENTRENAMIENTO

9.1. *Inicialización de pesos.*

9.2. *Ritmo de aprendizaje*

9.3. *Neuronas ocultas*

9.3.1 *Recetas*

9.3.2 *Prueba y error*

9.3.3 *Métodos dinámicos*

9.3.4 *Parada del entrenamiento*

PASO 10. EVALUACIÓN DE LOS RESULTADOS

CAPÍTULO 5. CASO DE ESTUDIO: EL CASO DE CLASIFICACIÓN DE TUTORES EN EL PROGRAMA DE POSGRADO EN CIENCIAS DE LA ADMINISTRACIÓN DE LA UNAM.

INTRODUCCIÓN

1. *Análisis del problema.*

3. *Selección del modelo de red más adecuado al problema planteado.*

4. *Selección de valores de entrada (valores de entrenamiento).*

5. *Selección de valores de salida (valores objetivo).*

6. *La definición de la arquitectura de la red neuronal artificial para el caso de estudio propuesto.*

7. *Elección de funciones de activación para las capas ocultas y función de transferencia.*

8. *Elección de una regla de aprendizaje y fijación los parámetros iniciales.*

9. *Preparación del archivo con los ejemplos de entrenamiento; los patrones de entrada y los patrones de salida con los que se debe asociar.*

11. *Fijación de los parámetros de entrenamiento para cada corrida de la red.*

12. *Entrenamiento de la red*

13. *Resultados obtenidos al correr la red neuronal artificial propuesta.*

13.1 EJECUCIÓN DE LA RED PARA ENCONTRAR EL MEJOR DISEÑO.

Figura 51. Los resultados que arrojó la red con el mejor ajuste

13.2 LOS RESULTADOS DEL ENTRENAMIENTO

13.3 ECUACIÓN DE LA FUNCIÓN DISCRIMINANTE:

DESEADA/SALIDA

LIMITACIONES DEL ESTUDIO:

RECOMENDACIONES:

CONCLUSIONES:

BIBLIOGRAFÍA

ANEXO A

1. RELACIÓN DE TABLAS DE LA SITUACIÓN ACTUAL DEL PROGRAMA DE POSGRADO EN CIENCIAS DE LA ADMINISTRACIÓN DE LA UNAM.

ANEXO B

TABLA 9. LOS VALORES DE ENTRADA O ENTRENAMIENTO SE MUESTRAN EN LAS 16 VARIABLES COMO COLUMNAS.

TABLA 12. VALORES RESULTANTES DEL PREPROCESAMIENTO.

Índice de figuras

FIGURA 1. EJEMPLO DE UN SISTEMA DE INFORMACIÓN DE CONTROL DE CLIENTES.

FIGURA 2. MODELO RACIONAL PARA LA TOMA DE DECISIONES

FIGURA 3. EL PROCESO DECISIONAL

FIGURA 4. TIPOS DE SISTEMAS DE INFORMACIÓN

FIGURA 5. VISUALIZACIÓN DE LAS DECISIONES

FIGURA 6. DIFERENTES TIPOS DE SISTEMAS DE INFORMACIÓN SEGÚN EL TIPO DE DECISIÓN Y NIVEL DE ORGANIZACIÓN

FIGURA 7. IMITACIÓN DEL COMPORTAMIENTO INTELIGENTE Y SU RELACIÓN CON LAS FUNCIONES DE UN SISTEMA DE INFORMACIÓN.

FIGURA 8. ÁREAS DE ESTUDIO E INVESTIGACIÓN DE LA INTELIGENCIA ARTIFICIAL.

FIGURA 9. LA NEURONA BIOLÓGICA

FIGURA 10. SINAPSIS

FIGURA 11. NEURONAS BIOLÓGICAS DE UNA SANGUIJUELA.

FIGURA 12. MODELO MATEMÁTICO DE UNA NEURONA COMO UNIDAD DE PROCESO.

FIGURA 13. ESQUEMA DE UNA NEURONA DE MCCULLOCH-PITTS

FIGURA 14. ESQUEMA DE UNA RED DE TRES CAPAS TOTALMENTE INTERCONECTADAS

FIGURA 15. RED DE DOS CAPAS

FIGURA 16. RED DE UNA CAPA EQUIVALENTE

FIGURA 17. RED DE UNA SOLA CAPA INTERNA O MONOCAPA

FIGURA 18. RED MULTICAPA

FIGURA 19. CLASIFICACIÓN DE LAS REDES NEURONALES DE ACUERDO A SU TIPO DE APRENDIZAJE.

FIGURA 20. ESQUEMA DE LA ESTRUCTURA DE UNA RNA

FIGURA 21. ESTRUCTURA DE UNA RED MULTICAPA CON CONEXIONES HACIA DELANTE.

FIGURA 22. EJEMPLOS DE CONEXIONES CON PROPAGACIÓN HACIA ATRÁS

FIGURA 23. ARQUITECTURA DEL PERCEPTRON CON DOS ENTRADAS Y UNA SALIDA

FIGURA 24. SEPARACIÓN DE DOS CLASES MEDIANTE UN PERCEPTRON

FIGURA 25. ARQUITECTURA DEL PERCEPTRON MULTICAPA

FIGURA 26. FUNCIONES DE ACTIVACIÓN DE PERCEPTRON MULTICAPA

FIGURA 27. EJEMPLO DE ARQUITECTURA DEL PERCEPTRON MULTICAPA

FIGURA 28. EVOLUCIÓN DEL ERROR A LO LARGO DEL PROCESO DE APRENDIZAJE

FIGURA 29. GENERALIZACIÓN: (A) BUENAS PROPIEDADES; (B) ESCASAS PROPIEDADES

FIGURA 30. EVOLUCIÓN DE LOS ERRORES DE ENTRENAMIENTO Y VALIDACIÓN A LO LARGO DEL PROCESO DE APRENDIZAJE.

FIGURA 31. EJEMPLO DE UNA RED NEURONAL PARA ASIGNACIÓN DE CRÉDITOS.

FIGURA 33. FUNCIÓN DE TRANSFERENCIA ESCALÓN

FIGURA 34. FUNCIONES DE ACTIVACIÓN MIXTA

FIGURA 35. FUNCIONES DE ACTIVACIÓN CONTINUA (SIGMOIDAL)

FIGURA 36. FUNCIÓN DE TRANSFERENCIA GAUSSIANA

FIGURA 37. REPRESENTACIÓN DE LA FUNCIÓN DE TRANSFERENCIA DE UNA NEURONA.

FIGURA 38. REPRESENTACIÓN DE LA FUNCIÓN DE TRANSFERENCIA CON EL UMBRAL DE ACTIVACIÓN DE LA NEURONA.

FIGURA 39. FUNCIÓN DE ACTIVACIÓN ESCALÓN

FIGURA 40. FUNCIÓN DE ACTIVACIÓN LINEAL O IDENTIDAD

FIGURA 41. FUNCIÓN DE ACTIVACIÓN LINEAL-MIXTA

FIGURA 42. FUNCIÓN DE ACTIVACIÓN SIGMOIDAL

FIGURA 43. REPRESENTACIÓN DE UNA NEURONA CON VALOR DE SALIDA UNITARIA ASOCIADA A LA ENTRADA.

FIGURA 44. RED NEURONAL CON APRENDIZAJE SUPERVISADO

FIGURA 45. EVOLUCIÓN DEL ERROR DE APRENDIZAJE Y DEL ERROR DE PRUEBA EN FUNCIÓN DE LAS ITERACIONES, EN UN PROCESO DE APRENDIZAJE

FIGURA 47. SISTEMA NEURONAL ARTIFICIAL

GRÁFICA 48. PASOS PARA REALIZAR EL ANÁLISIS DEL PROBLEMA

FIGURA 49. ARQUITECTURA DE LA RED NEURONAL QUE CLASIFICA TUTORES EN EL PROGRAMA DE POSGRADO

FIGURA 50. FUNCIÓN DE ACTIVACIÓN DE PERCEPTRON MULTICAPA

Introducción

El aumento en la complejidad y en el tamaño de la organización moderna, han hecho que las funciones administrativas de planeación, organización y control sean más difíciles de ejecutar, aunque cada vez más indispensables para la estabilidad y el crecimiento de la empresa. Los gerentes necesitan más ayuda que la que pueden encontrar estudiando las experiencias de otros por lo que deben apoyarse en una ciencia básica y lograr un enfoque mucho más estructurado para tomar decisiones.

La toma de decisiones es una actividad fundamental en las organizaciones y requiere del buen manejo de la información; conseguir de manera eficiente la información adecuada, clasificarla para que pueda usarse según se necesite, ayuda a una organización a lograr sus objetivos. Según Effy Oz, [2001:32], “El éxito de una organización depende en gran medida de la calidad de las decisiones que tomen sus administradores, cuando para tomar decisiones implica manejar grandes cantidades de información y mucho procesamiento, los sistemas basados en computadora hacen que el trabajo sea efectivo y eficiente”.

Los responsables de la toma de decisiones han de asumir una racionalidad limitada debido a la complejidad de la mayor parte de los problemas organizativos, la incertidumbre de la tecnología y del entorno, las limitaciones del proceso cognitivo humano, las emociones y las preferencias individuales.

Los sistemas de información ayudan a enfrentar la racionalidad limitada y tomar mejores decisiones porque proporcionan los instrumentos para la planeación, organización, integración y control de sus operaciones en una forma más efectiva.

Los sistemas de información se han vuelto esenciales para que las organizaciones puedan enfrentar los cambios de las economías globales y las empresas comerciales. Estos proveen a las empresas de herramientas de comunicación para dirigir transacciones y administrar negocios a escala global. Son el fundamento para la toma de decisiones en cuanto a nuevos productos y servicios, además de ayudar a las empresas a administrar sus activos del conocimiento; también, hacen posible que las empresas adopten estructuras más sencillas, más descentralizadas y acuerdos más flexibles entre los empleados y la administración, Laudon [2004,31:39].

Las tecnologías de información y comunicación aplicadas a los sistemas de información han pasando de ser simples herramientas de tratamiento de datos para convertirse en la columna vertebral de cualquier organización, tanto a nivel interno como en lo referente a las relaciones con el exterior.

Con el nacimiento de la inteligencia artificial (IA) se plantea la posibilidad de diseñar máquinas “inteligentes”, sistemas expertos, algoritmos genéticos, computación molecular o redes neuronales artificiales (RNA) para apoyar a la toma de decisiones. Las redes neuronales artificiales son eficientes en tareas tales como el reconocimiento de patrones, en la resolución de problemas de clasificación y optimización, también pueden integrarse en un sistema para ayuda a la toma de decisiones, aunque no sean capaces de resolver todos los problemas ya que son modelos muy especializados.

El Programa del Doctorado en Ciencias de la Administración, es un programa en el cual participan tutores de la FCA y de diversas instituciones de la UNAM, en el país y en el extranjero. El control y los criterios para la adecuada clasificación y asignación de los tutores requieren de información concreta que permita tomar decisiones eficientes. En este sentido se requiere del manejo de nuevas tecnologías como la IA con la aplicación de redes neuronales artificiales que permita controlar a los tutores existentes e incorporar a nuevos de manera que se pueda monitorear la efectividad de los mismos. En este trabajo nos preguntamos:

¿Cuál es el modelo de red neuronal artificial más adecuado para clasificar tutores de manera eficiente en el Programa de Posgrado en Ciencias de la Administración de la UNAM?

Como hipótesis central consideramos que el modelo de red neuronal artificial más adecuado para lograr que la clasificación de tutores sea más eficiente en el Programa de Posgrado en Ciencias de la Administración de la UNAM, es el Perceptron multicapa, con “conexiones hacia adelante”, aprendizaje supervisado (algoritmo de aprendizaje con corrección de error), red heteroasociativa, con entrada analógica y salida binaria.

El **objetivo general** del trabajo es evaluar si el modelo del Perceptron multicapa, es el modelo de red neuronal artificial más adecuado para la clasificación de tutores en el Programa de Posgrado en Ciencias de la Administración de la UNAM; y en su caso, diseñar la arquitectura de la red y establecer una metodología para ello.

Los **objetivos específicos** son:

- Diagnosticar el Posgrado en relación a los sistemas existentes para la clasificación y evaluación de tutores.
- Identificar las ventajas de la aplicación de las redes neuronales artificiales en la toma de decisiones en instituciones de educación superior.
- Plantear la metodología en forma automatizada para diseñar la arquitectura y el modelo de una red neuronal que permita resolver un problema para la toma de decisiones, dicha metodología se incluye en un CD tipo multimedia, al final de este trabajo.

Para lograr estos objetivos se siguió la siguiente estrategia y metodología:

- a) Se realizó una revisión bibliográfica de 30 modelos de redes neuronales.
- b) Se llevó a cabo un análisis y una clasificación de los modelos más importantes y se establecieron sus características.
- c) Se realizó un diagnóstico del Posgrado.
- d) Con base en la problemática del Programa de Doctorado, se identificó cuál era el modelo de red neuronal más apropiado.
- e) Una vez identificado el modelo más apropiado como el Perceptron multicapa, se establecieron sus características.
- f) Se elaboró la metodología para diseñar la arquitectura de la red neuronal.
 - Se utilizó el paquete NeuroSolutions¹ de la empresa Neurodimension, Inc., para procesar la red neuronal diseñada.
 - A partir del diagnóstico del Posgrado, se realizaron entrevistas con los funcionarios del posgrado para conocer la problemática, se identificaron las variables más relevantes y se diseñó un instrumento de recolección de datos.
 - Se obtuvo información relativa a 128 tutores que han graduado alumnos de doctorado, a estos datos se les llamó *grupo de entrenamiento* o datos de entrada a la red neuronal.
 - Se obtuvo información relativa al desempeño de los 128 tutores, a estos datos se les llamó, *datos de salida* de la red.

¹ Copyright© 1994-2005. Este es el software utilizado para resolver el caso de estudio de este trabajo de investigación.

- Para realizar el pronóstico del desempeño de los tutores de nuevo ingreso o grupo de prueba, se obtuvo información relativa a 12 tutores de recién ingreso, a estos datos se les llamó *grupo de prueba* y se introdujeron a la red; una vez que ésta aprendió con los datos del grupo de entrenamiento.

Este trabajo de investigación consta de 5 capítulos, los primeros 3 componen el marco teórico: el **capítulo 1** contiene conceptos y aplicaciones de los sistemas de información así como la justificación de la inteligencia artificial y en especial la importancia de las redes neuronales artificiales para la toma de decisiones, cuando éstas son *no estructuradas*, el **capítulo 2** trata de los antecedentes y situación actual del Programa de Posgrado en Ciencias de la Administración de la UNAM y el **capítulo 3** contiene los fundamentos biológicos de las redes neuronales artificiales, los primeros modelos computacionales, así como sus características, su clasificación, estructura, los diferentes modelos y aplicaciones de las redes neuronales más importantes, tales como el Perceptron.

Por otro lado, el **capítulo 4** corresponde a la descripción de la metodología para diseñar la arquitectura de la red neuronal artificial, que ayudará al proceso de toma de decisiones. Aquí se describen los pasos que deben seguirse en primera instancia, para justificar el uso de una red neuronal y después, para definir los requerimientos de software y hardware para ejecutar la red neuronal propuesta. Se define la *topología* de la red de acuerdo al número de niveles o capas, grado de conectividad entre las neuronas y tipo de conexiones entre neuronas. De la misma manera se define la *microestructura* de acuerdo a tipos de unidades de proceso o neuronas, su estado de activación, su función de salida y de activación. Posteriormente, se analiza la forma de conexión con otras neuronas y la regla de aprendizaje de la red neuronal. Se explica cómo deben representarse las informaciones de entrada y salida y cómo preparar los datos, es decir el *preprocesamiento* como paso previo al proceso de entrenamiento y finalmente cómo se entrena la red y cómo se evalúan los resultados. En cada paso se enuncian los modelos que existen de acuerdo a las características mencionadas. Como aportación de este capítulo se incluye un CD tipo multimedia con los pasos descritos en la metodología.

En el **capítulo 5** se resuelve el problema planteado, siguiendo la metodología arriba descrita, planteando más a detalle cada paso; también, se define la arquitectura de la red neuronal, se elige la regla de aprendizaje, se fijan los parámetros iniciales, se prepara el archivo con los datos de entrenamiento, validación y prueba para finalmente ejecutar

la red neuronal propuesta y después de ejecutar y analizar el tamaño de los errores, se elige la mejor arquitectura de red, la cual se entrena y una vez que ésta “aprendió” nos proporciona los resultados de quienes serán los candidatos tutores que cumplirán con efectividad con los alumnos encomendados al Programa de Posgrado en Ciencias de la Administración de la UNAM.

Marco teórico y conceptual

Capítulo 1. Sistemas de información

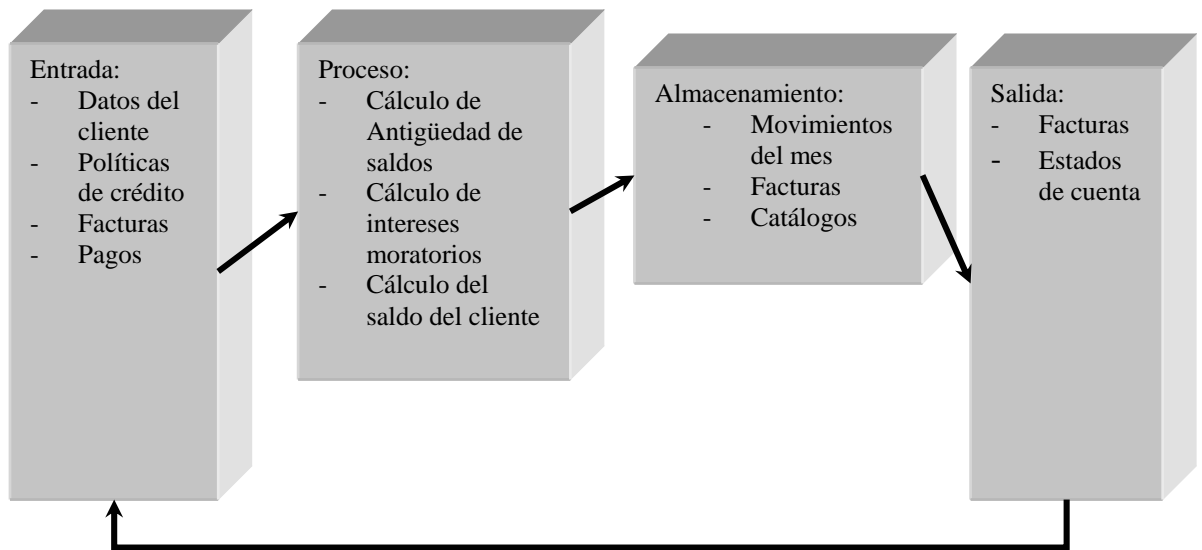
Un *sistema* es un grupo de elementos que se integran con el propósito común de lograr un objetivo, McLeod (2000:12) y un *sistema de información*, es un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control de una organización, Laudon (2004:8),

Para que el sistema pueda operar es necesario contar con equipo computacional. El recurso humano es el que interactúa con el sistema de información.

Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida, Cohen (1996:4) (véase la figura 1).

- **Entrada:** es el proceso mediante el cual el sistema de información toma los datos que requiere para procesar la información.
- **Almacenamiento:** a través de esta facultad la computadora puede recuperar la información guardada, y a este almacenamiento en estructuras de información se le conoce como archivos.
- **Procesamiento:** es la capacidad del sistema de información para efectuar operaciones o transformar los datos de entrada o los que se encuentran almacenados en reglas lógicas preestablecidas para la toma de decisiones.
- **Salida:** es la capacidad de un sistema de información para obtener información procesada, la cual puede convertirse en la entrada de otro sistema.

Figura 1. Ejemplo de un sistema de información de control de clientes.



Fuente: Laudon [2004:35]

Objetivos de los sistemas de información:

1. Automatizar procesos operativos.
2. Proporcionar información que sirva de apoyo al proceso de toma de decisiones
3. Lograr ventajas competitivas con su uso

1.1 Toma de decisiones

La toma de decisiones es un proceso en el que se debe escoger entre dos o más alternativas, [Huber, 1984]. Las decisiones cuyos procesos requieren apoyo de la información pueden clasificarse en dos tipos:

- *Decisiones repetitivas.*

Este tipo de decisiones se toman en niveles organizacionales intermedios y se caracterizan por ser, en cierta medida, predecibles, de tal suerte que se pueden desarrollar de antemano algunos modelos o programas, con el fin de preparar el momento de la toma de decisión. Además el efecto de estas operaciones con un horizonte de corto plazo se realiza sobre las operaciones cotidianas de la institución.
- *Decisiones no repetitivas.*

Este tipo de decisiones suele representarse en los niveles más altos de la organización y se caracterizan por un considerable grado de incertidumbre. Con frecuencia son llamados decisiones no estructuradas o no programadas, ya que el elemento relevante de las mismas es la imposibilidad de predecir el tipo y escenario de la decisión.

En este tipo de decisiones es necesario contar con herramientas que permitan obtener la información necesaria y oportuna, se utilizan entonces las herramientas de cómputo y el desarrollo de aplicaciones de usuario final.

Las decisiones repetitivas y no repetitivas pueden ser clasificadas de acuerdo al número de participantes durante el proceso en:

- **Decisiones independientes**
Son tomadas en forma aislada por una persona, es decir es el encargado de tomar la decisión ya que tiene la responsabilidad y autoridad para hacerlo.
- **Decisiones secuenciales**
Son generadas por un grupo de personas, por lo cual cada una de ellas solo participa en una parte de la decisión, y el resto es transferido a otras áreas o ejecutivos de la organización para completar el proceso.
- **Decisiones simultáneas.**
Sólo se toman en grupo ya que son producto de la interacción y negociación entre varias personas en forma simultánea.

En esta parte se resalta la importancia de la toma de decisiones con la ayuda de las tecnologías de información en una organización.

1.1.1 Toma de decisiones en la organización

El sustento para la toma de decisiones en una organización se encuentra en las funciones administrativas que son: planeación, organización, dirección y control.

- **Planeación**
¿Cuáles son los objetivos de la organización a largo plazo?

¿Qué estrategias son mejores para lograr este objetivo?

¿Cuáles deben ser los objetivos a corto plazo?

¿Qué tan altas deben ser las metas individuales?

- **Organización**

¿Cuánta concentración debe existir en una organización?

¿Cómo deben diseñarse los puestos?

¿Quién está mejor calificado para ocupar un puesto vacante?

¿Cuándo debe una organización instrumentar una estructura diferente?

- **Dirección**

¿Cómo manejo a un grupo de trabajadores que parecen tener una motivación baja?

¿Cuál es el estilo de liderazgo más eficaz para una situación dada?

¿Cómo afectará un cambio específico a la productividad del trabajador?

¿Cuándo es adecuado estimular el conflicto?

- **Control**

¿Qué actividades en la organización deberían ser controladas?

¿Cómo deben controlarse estas actividades?

¿Cuándo es significativa una organización en el desempeño?

¿Cuándo está la organización desempeñándose de manera efectiva?

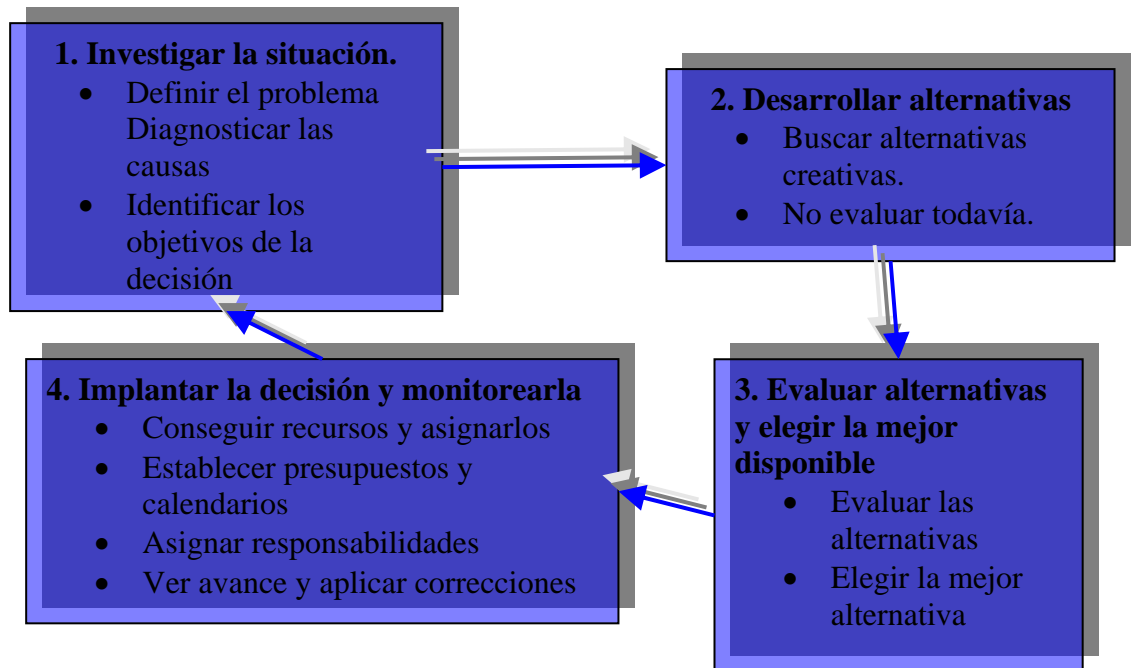
En la toma de decisiones se trata con problemas. Un problema surge cuando el estado real de las cosas no se ajusta al estado deseado. En muchos casos, un problema puede representar una oportunidad disfrazada.

1.1.2 El modelo racional en la toma de decisiones

Este modelo resulta muy útil para tomar decisiones no programadas; sirve para ir más allá de un razonamiento a priori, es decir de la suposición que existe una solución evidente que solo está en espera de que alguien la encuentre, [Franas, 1971].

Ninguno de los enfoques para tomar decisiones puede garantizar que se tome la decisión acertada. Sin embargo, recurrir a un enfoque racional inteligente y sistemático tiene más probabilidades de encontrar soluciones de gran calidad. El proceso básico de las decisiones racionales se presenta en cuatro etapas como se muestra en la figura 2.

Figura 2. Modelo racional para la toma de decisiones



Fuente: [Franas, 1971]

1.1.3 El proceso decisional: sus fases y etapas.

Los especialistas en toma de decisiones coinciden en que este proceso se desarrolla en seis fases, existiendo factores de comportamiento y ambiente que influyen en la decisión. El proceso de toma de decisiones va a requerir un tratamiento por separado dependiendo de quien decide:

- Si es una persona
- Si es un grupo pequeño y
- Una organización compleja

La decisión de una persona puede ser muy sencilla, pero la decisión de una organización podría considerar la aprobación de varios departamentos, por ejemplo de ingeniería, aspectos legales, estudios

de finanzas, políticas de la alta dirección y muchas veces la aprobación gubernamental, [Rodríguez, 1988].

En la figura 3 se representa el proceso que debería seguirse para tomar una decisión importante y compleja. A continuación se describen cada una de las fases del proceso decisional:

1.- *Se percibe y define un problema.* En este sentido el que decide será capaz de distinguir entre causas y efectos, sistemas y problemas, ya que de otra manera puede caer en manejos equivocados. En muchos casos una situación conflictiva puede originarse por una serie de factores que, si no se identifican con claridad hacen correr el riesgo de actuar solo sobre lo más visible pero menos importante.

El análisis del problema requiere máxima objetividad. El que decide debe tener en cuenta todos los antecedentes disponibles en la institución. Será capaz de precisar si se trata de un problema actual, que surgió en forma reciente, o si es fruto de desarrollos anteriores.

El análisis sistemático del problema permite despejar incógnitas y situarlo en su dimensión real.

2.- Después de analizar en forma exhaustiva el problema, ***se determina el resultado al que se desea llegar***, pensando tanto en lo que sería ideal como en lo posible de lograr en la situación concreta.

3.- *Después de ventilar en forma analítica la información válida y suficiente*, se idea el mayor número de soluciones para luego escoger las que se consideren viables. Posiblemente se requerirán nuevos datos para poder calcular de antemano los resultados de cada una de ellas, los cuales se compararán en las normas fijadas.

La tecnología moderna ha creado gran cantidad de medios capaces de reunir información, pero que ningún caso pueden reemplazar a una persona en la toma de decisiones. Los sistemas automatizados, investigación operativa y modelos matemáticos, se consideran solo como instrumentos para analizar los problemas.

El que decide ni debe transferir la responsabilidad de la toma de decisiones ni a las máquinas ni a los especialistas en dichas técnicas modernas, sin obtener de ellos todo el respaldo en datos y referencias para clasificación del problema y su análisis correspondiente.

En este sentido juegan un papel muy importante la experiencia, conocimientos y espíritu creador ejecutivo. Lo importante es evitar la adopción de soluciones rutinarias, que solo consiguen disimular y postergar el problema. Tales cualidades permiten también desechar ideas irracionales, que no satisfagan los objetivos y políticas de la empresa.

4.- *Se elige la alternativa que elimine el problema*, siendo a la vez la que mejor satisfaga esas condiciones y se acerque más al objetivo ideal.

El que decide debe tratar de establecer para cada alternativa pros y contras y, si es posible, introducir elementos cualitativos: costos, resultados esperados, tiempo requerido, entre otros.

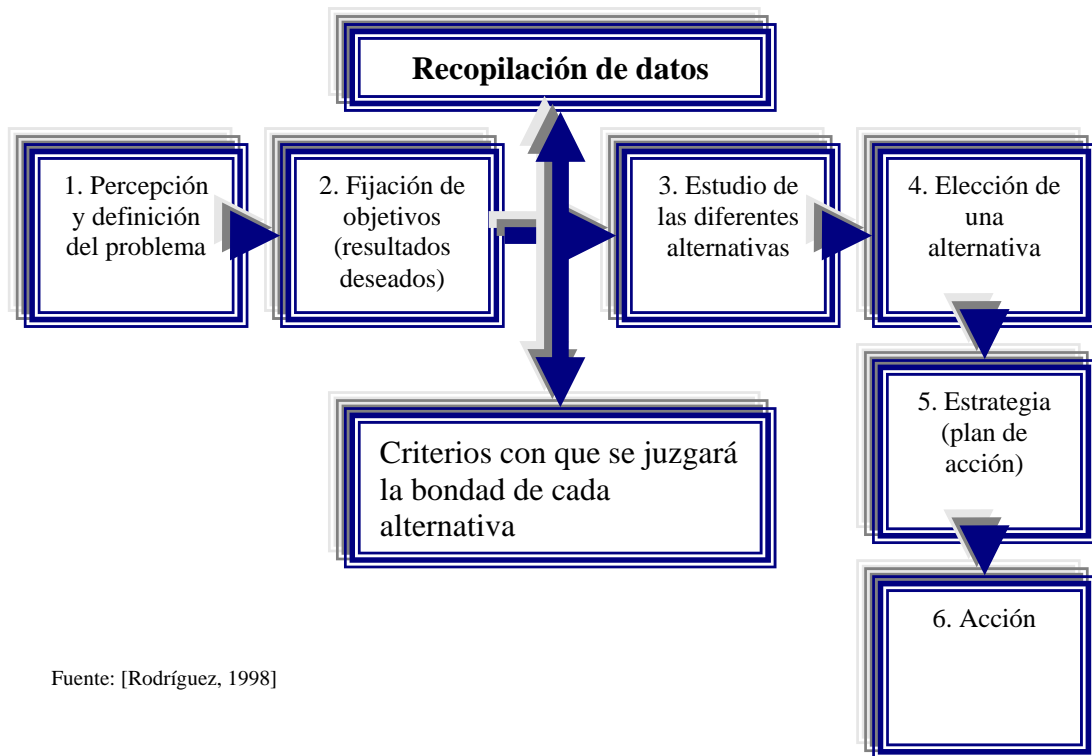
Un aspecto muy importante en esta fase es que la elección de la alternativa se acompañe de una segunda selección para el caso de que aquella no logre los resultados esperados o encuentre resistencia en su aplicación.

5.- *Se elabora el plan de acción de la mejor alternativa*; es decir, se fija una estrategia.

6.- *Se pone en práctica dicho plan*.

Esta última fase consiste en traducir la decisión en una acción, ya que una cosa es determinar la decisión y otra es que esta se aplique y solucione el problema.

Figura 3. El proceso decisional



1.2 Importancia de los sistemas de información en la toma de decisiones

La toma de decisiones es un proceso de pensamiento que ocupa toda la actividad que tiene como finalidad la solución de problemas.

La toma de decisiones es necesaria cuando tenemos un problema que resolver, o necesidades que satisfacer.

Los sistemas de información son de vital importancia en cualquier tipo de información ya que nos proporcionan las herramientas necesarias para que un tomador de decisiones pueda realizar su trabajo óptimamente.

Ya que dichos sistemas al proporcionar la información necesaria en el preciso momento y con la mayor eficiencia posible, ayudan a que la empresa crezca y se desarrolle.

1.3 Los sistemas de información en las organizaciones

Los sistemas de información son importantes en las empresas debido a que las organizaciones operan en un mundo de desaciertos e intervención gubernamental, de políticas impredecibles a nivel monetario, fiscal, impositivo y regulador; de ciclos de negocios y recesiones; de cambios abruptos en las políticas comerciales; de competencia doméstica e internacional; y de crecientes costos laborales. A decir verdad, éste es un ambiente implacable y competitivo en el que deben sobrevivir las organizaciones.

Para evitar el fracaso, sobrevivir, y lograr el éxito, las organizaciones deben explorar las dimensiones de la oportunidad de una gerencia informada, de la diferenciación de productos y servicios de una creciente productividad.

Claramente, la información es el arma principal que ayudará a la gerencia, a los productos, a los servicios y a la productividad a penetrar en el ambiente competitivo.

El encanto de la tecnología informática no hará avanzar estas dimensiones, pero sí lo hará la necesidad de contender y sobrevivir en un ambiente competitivo y violento, un ambiente que incluye una competencia internacional más fuerte. Debe quedar claro que las computadoras, la tecnología informática y la información de calidad no son los fines sino simplemente las armas competitivas que apoyan a las organizaciones para alcanzar las metas de los gerentes triunfadores, de productos y servicios excelentes y de una mayor productividad, y del éxito a final de cuentas.

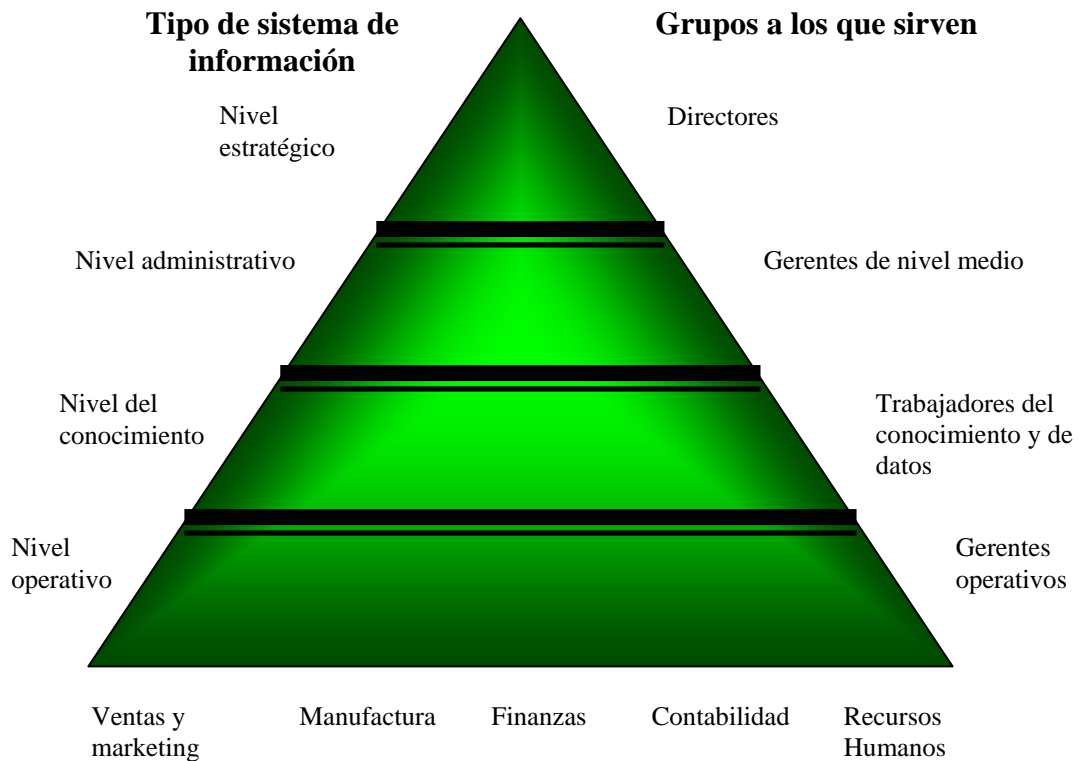
Cualquiera que sea la organización, las compañías que producen la información de la más alta calidad permanecerá como las más fuertes competidoras del ramo. Por otra parte, si una compañía no puede mejorar su información, quedará a la zaga de aquéllas que sí pueden.

1.4 Principales tipos de sistemas en las organizaciones

Dado que hay intereses, especialidades y niveles diferentes en una organización, existen diferentes tipos de sistemas. El autor Laudon

[2004:39] opina que, “un sistema por sí solo no proporciona toda la información que una empresa necesita”. La figura 4 ilustra una forma de describir los tipos de sistemas que se encuentran en una organización.

Figura 4. Tipos de sistemas de información



Fuente: Laudon & Laudon [2004:39]

En la Figura 4, la organización se divide en niveles estratégico, administrativo, del conocimiento y operativo y luego se divide en áreas funcionales como ventas y marketing, manufactura, finanzas, contabilidad y recursos humanos. Los sistemas se crean para dar servicio a estos diversos intereses de la organización.

Existen cuatro principales tipos de sistemas de información que dan servicio a los diferentes niveles de la organización, sistema a nivel operativo, sistema a nivel de conocimiento, sistema a nivel administrativo y sistema a nivel estratégico, Laudon [2004:39].

Los **sistemas a nivel operativo** apoyan a los gerentes operativos en el seguimiento de las actividades y transacciones elementales de la organización como ventas, ingresos, depósitos en efectivo, nómina, decisiones de crédito, etc. El objetivo principal de los sistemas a este nivel es responder las preguntas de rutina y seguir el flujo de las transacciones a través de la organización.

Los **sistemas a nivel de conocimiento** apoyan a los trabajadores del conocimiento y de datos de una organización. El propósito de estos sistemas es ayudar a las empresas comerciales a integrar el nuevo conocimiento en los negocios y ayudar a la organización a controlar el flujo del trabajo de oficina.

Los **sistemas a nivel administrativo** sirven a las actividades de supervisión, control, toma de decisiones y administrativas de los gerentes del nivel medio. La pregunta principal que plantean estos sistemas es: ¿van bien las cosas? Por lo general, este tipo de sistemas proporciona informes periódicos más que información instantánea de operaciones. Algunos sistemas a nivel administrativo apoyan la toma de decisiones no estructuradas según, Keen (1978). Los requisitos de información no siempre son claros. Estos sistemas a veces responden a preguntas “qué pasaría si”: Las respuestas a estas preguntas suelen requerir nuevos datos externos a la organización así como datos internos que no son fáciles de obtener de los sistemas a nivel operativo existentes.

Los **sistemas a nivel estratégico** ayudan a los directores a enfrentar y resolver aspectos estratégicos y tendencias a largo plazo, tanto en la empresa como en el entorno externo. Su función principal es compaginar los cambios del entorno externo con la capacidad organizacional existente. ¿Cuáles serán los niveles de empleo dentro de cinco años? ¿Cuáles son las tendencias a largo plazo de los costos de la industria y dónde encaja nuestra empresa? ¿Qué productos deberemos estar elaborando dentro de cinco años?

1.5 Clasificación de sistemas de información de acuerdo a su aplicación

La clasificación de los sistemas de información que hace el autor Laudon (2004) es de tres tipos, y ésta se refiere a la relación que los sistemas guardan de acuerdo a su aplicación, es decir, los sistemas

que sirven para automatizar procesos operativos dentro de una organización pertenecen al primer tipo y son los llamados frecuentemente *sistemas transaccionales*, ya que su función primordial consiste en procesar transacciones tales como pagos, cobros, pólizas, entradas, salidas, etc. El segundo tipo corresponde a los sistemas de información que apoyan el proceso de toma de decisiones son: *sistemas de soporte a la toma de decisiones*, *sistemas para la toma de decisión de grupo*, *sistemas expertos de soporte a la toma de decisiones* y *sistemas de información para ejecutivos*. El tercer tipo de sistemas, es el de los *sistemas estratégicos*, los cuales se desarrollan en las organizaciones con el fin de lograr ventajas competitivas, a través del uso de la tecnología de información.¹

1. Sistemas transaccionales.

Sus principales características son:

- o Automatizan tareas operativas de la organización. Con frecuencia son el primer tipo de sistemas de información que se implanta en las organizaciones. Se empieza apoyando las tareas a nivel operativo de la organización.
- o Son intensivos en entrada y salida de información; sus cálculos y procesos suelen ser simples y poco sofisticados.
- o Tienen la propiedad de ser recolectores de información, es decir, a través de estos sistemas se cargan las grandes bases de información para su explotación posterior.
- o Son fáciles de justificar ante la dirección general, ya que sus beneficios son visibles y palpables.

2. Sistemas de apoyo a la toma de decisiones.

Estos sistemas dan servicio al nivel administrativo de la organización, según, Laudon (2004:349), Los sistemas de apoyo a la toma de decisiones, ayudan a los gerentes a tomar decisiones que son exclusivas, rápidamente cambiantes y no especificadas fácilmente con anticipación; al mismo tiempo, combinan datos, modelos analíticos, herramientas sofisticadas y software para la toma de decisiones semiestructurada y no estructurada. Las principales características de éstos son:

¹ <http://www.monografias.com/trabajos7/confi.shtml>

- Suelen introducirse después de haber implantado los *sistemas transaccionales* más relevantes de la empresa, ya que estos últimos constituyen su plataforma de información.
- La información que generan sirve de apoyo a los mandos intermedios y a la alta administración en el proceso de toma de decisiones.
- Suelen ser intensivos en cálculos y escasos en entradas y salidas de información. Así, por ejemplo, un modelo de planeación financiera requiere poca información de entrada, genera poca información como resultado, pero puede realizar muchos cálculos durante su proceso.
- No suelen ahorrar mano de obra. Debido a ello, la justificación económica para el desarrollo de estos sistemas es difícil, ya que no se conocen los ingresos del proyecto de inversión.
- Suelen ser sistemas de información interactivos y amigables, con altos estándares de diseño gráfico y visual, ya que están dirigidos al usuario final.
- Apoyan la toma de decisiones que, por su misma naturaleza son repetitivos y de decisiones no estructuradas que no suelen repetirse. Por ejemplo, un *sistema de compra de materiales* que indique cuándo debe hacerse un pedido al proveedor o un *sistema de simulación de negocios* que apoye la decisión de introducir un nuevo producto al mercado.
- Estos sistemas pueden ser desarrollados directamente por el usuario final sin la participación operativa de los analistas y programadores del área de informática.

3. Sistemas estratégicos.

Sus principales características son:

- Su función primordial no es apoyar la automatización de procesos operativos ni proporcionar información para apoyar la toma de decisiones.
- Suelen desarrollarse, dentro de la organización, por lo tanto no pueden adaptarse fácilmente a paquetes disponibles en el mercado.
- Típicamente su forma de desarrollo es a base de incrementos y a través de su evolución dentro de la organización. Se inicia con un proceso o función en particular y a partir de ahí se van agregando nuevas funciones o procesos.

- o Su función es lograr ventajas que los competidores no posean, tales como ventajas en costos y servicios diferenciados con clientes y proveedores. En este contexto, los *sistemas estratégicos* son creadores de barreras de entrada al negocio. Por ejemplo, el uso de cajeros automáticos en los bancos es un *sistema estratégico*, ya que brinda ventaja sobre un banco que no posee tal servicio. Si un banco nuevo decide abrir sus puertas al público, tendrá que dar este servicio para tener un nivel similar al de sus competidores.
- o Apoyan el proceso de innovación de productos y proceso dentro de la empresa debido a que buscan ventajas respecto a los competidores y una forma de hacerlo es innovando o creando productos y procesos.

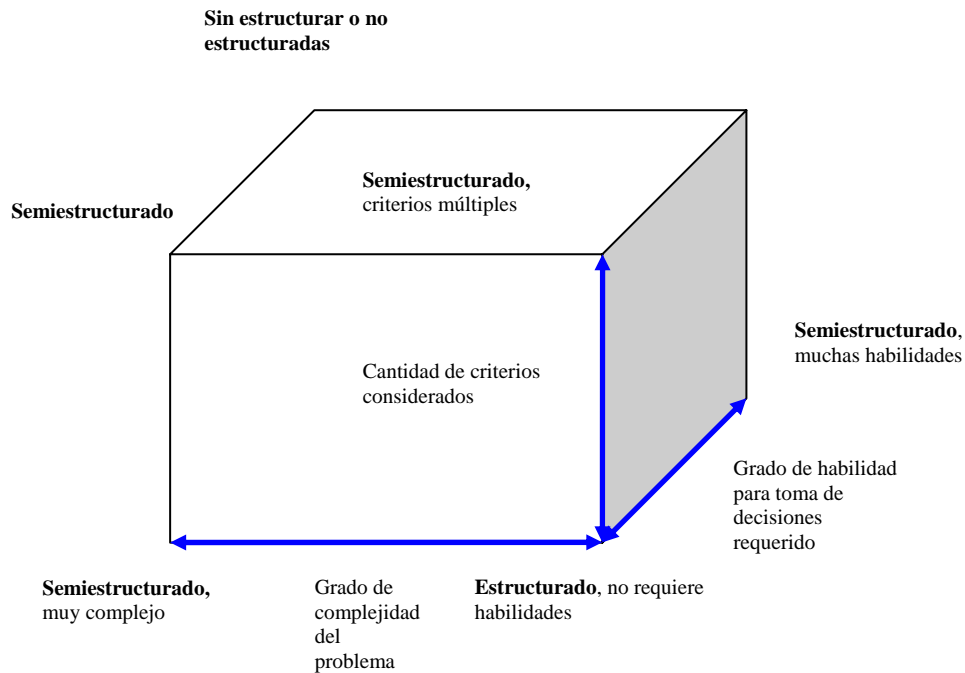
Un ejemplo de estos sistemas de información dentro de la empresa puede ser un sistema MRP (Manufacturing Resource Planning) enfocado a reducir sustancialmente el desperdicio en el proceso productivo, o bien, un centro de información que proporcione todo tipo de información; como situación de créditos, embarques, tiempos de entrega, etc. En este contexto los ejemplos anteriores constituyen un *sistema de información estratégico* si y sólo si, apoyan o dan forma a la estructura competitiva de la empresa.

1.6 Relación de los diferentes tipos de sistemas de información en función del tipo de decisión y nivel de organización

Las decisiones se clasifican en estructuradas, semiestructuradas y no estructuradas, Herbert A. Simon (1977:46) menciona que; “las decisiones estructuradas son repetitivas, rutinarias y existe ya un procedimiento definido para abordarlas; por el contrario, en las decisiones semiestructuradas y no estructuradas deben proporcionar juicios y aportar su propia evaluación”, según, Kendall (1997), “las decisiones estructuradas, son aquellas en donde todas o la mayoría de las variables son conocidas y pueden ser programadas totalmente, mientras que las decisiones no estructuradas son aquellas que son resistentes a la computarización y dependen principalmente de la intuición. Decisiones semiestructuradas son aquellas que son parcialmente programables, pero que todavía requieren discernimiento humano”.

En la Figura 5, el autor Kendall (1997:393), muestra la visualización de las decisiones estructuradas, como un cubo de tres dimensiones: grado de habilidad para toma de decisiones requerida, grado de complejidad del problema y cantidad de criterios de decisión requeridos.

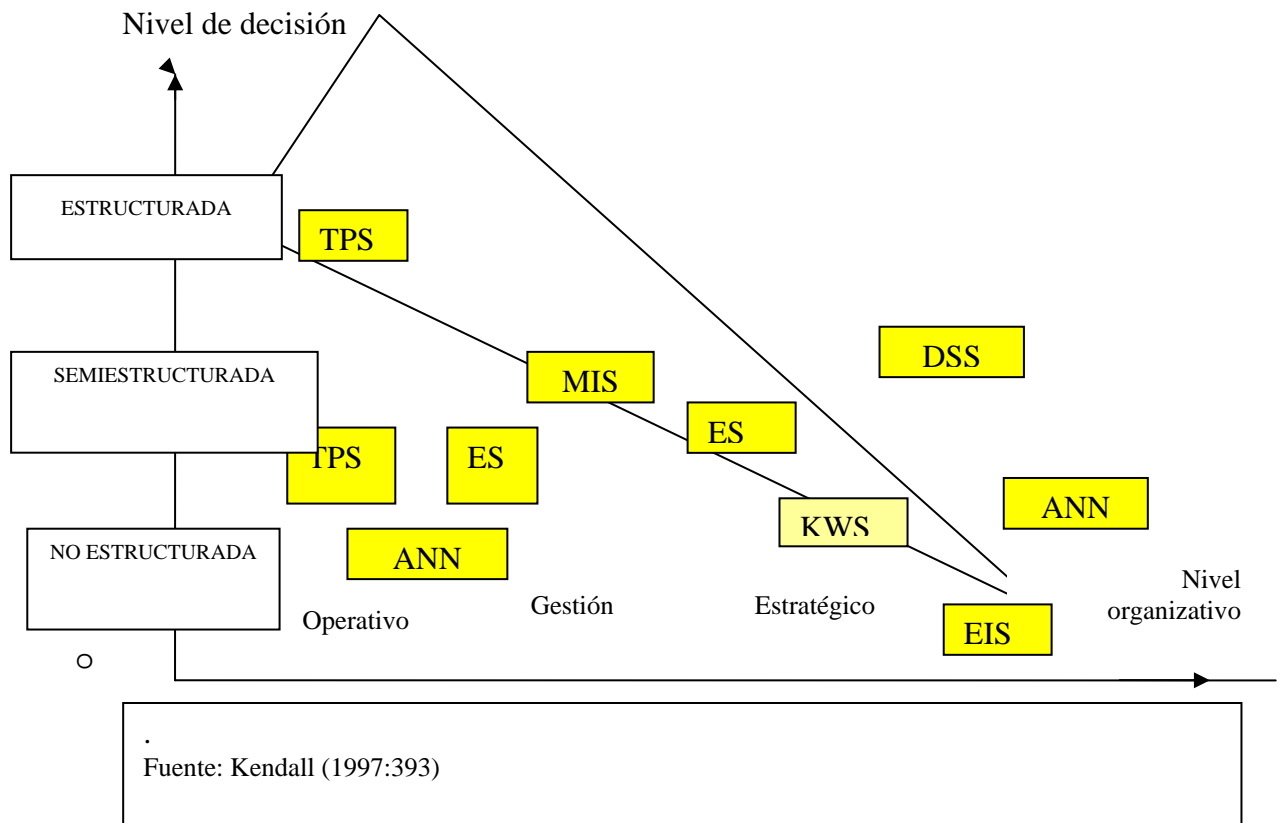
Figura 5. Visualización de las decisiones



Fuente: Kendall (1997:397)

El nivel al que se toman las decisiones afecta la elección del tipo de sistema más apropiado (Kendall 1997). Gorry y Cout-Morton (1971) han propuesto un esquema que relaciona el tipo de decisión (estructurada, semiestructurada y no estructurada) y el nivel organizacional (control operativo, de gestión y estratégico) con la herramienta a utilizar. Lógicamente en el nivel operativo dominan las decisiones estructuradas, en el nivel de gestión las semiestructuradas y en el estratégico las no estructuradas, formando una diagonal, como se muestra en la Figura 6.

Figura 6. Diferentes tipos de Sistemas de Información según el tipo de decisión y nivel de organización



- PS (Sistema de Proceso de Transacciones)
- MIS (Sistema de Gestión de la Información)
- EIS (Sistema de Información para Ejecutivos) Executive Information Systems (EIS)
- ANN (Artificial Neuronal Networks) o redes neuronales artificiales RNA
- ES (Sistema Experto)
- DSS (Sistema de Ayuda a la Toma de Decisiones)
- KWS (Sistemas de Trabajo del Conocimiento)

1.6.1 Nivel operativo y decisiones estructuradas

Como podemos apreciar en la Figura 6, en el nivel operativo y toma de decisiones estructuradas se encuentran los sistemas de información convencionales como los programas de contabilidad financiera y de costos, los de creación de nóminas, son los que en general realizan tareas operativas, éstos son los llamados Sistemas de

Proceso de Transacciones basados en programación convencional (Kendall, 1997).

1.6.2 Nivel operativo y decisiones semiestructuradas

El nivel operativo, pero con decisiones semiestructuradas, todavía está dominado por los programas convencionales, en lo que habría que incluir programas de control de tesorería, control de existencias y también las hojas de cálculo y sistemas gestores de bases de datos (Kendall, 1997).

Conforme las decisiones van siendo menos estructuradas, comienzan a aplicarse sistemas expertos y sistemas de ayuda a la toma de decisiones. Los sistemas de trabajo del conocimiento, sirven para mantener actualizado el conocimiento en la organización. Las organizaciones están empleando la tecnología de la IA para capturar el conocimiento individual y colectivo y codificar y extender su base de conocimiento.

Así, para la gestión financiera, cuando hay que tomar decisiones que manejan información incompleta o robusta, precisan incorporar el conocimiento de un especialista humano. Finalmente, en el nivel de toma de decisiones menos estructurada pueden incorporarse sistemas de redes neuronales artificiales, por ejemplo, la concesión de créditos o como en nuestro caso de estudio la decisión de asignar alumnos a tutores según su formación académica.

1.6.3 Nivel de gestión y decisiones estructuradas

Si observamos la Figura 6, sobre el nivel organizativo en el eje de las "x", las decisiones se vuelven más complejas. Si éstas son estructuradas, como en el análisis de presupuestos, contabilidad analítica, etc., dominan los programas convencionales. Este tipo de sistemas se denominan sistemas de gestión de información.

1.6.4 Nivel de gestión y decisiones no estructuradas

Conforme las decisiones se hacen menos estructuradas se hacen más necesarios los sistemas expertos. En algunos temas más específicos como la elaboración de presupuestos, la predicción de variables financieras, manejo de flujo de efectivo, podrían incorporarse modelos neuronales. (Figura 6).

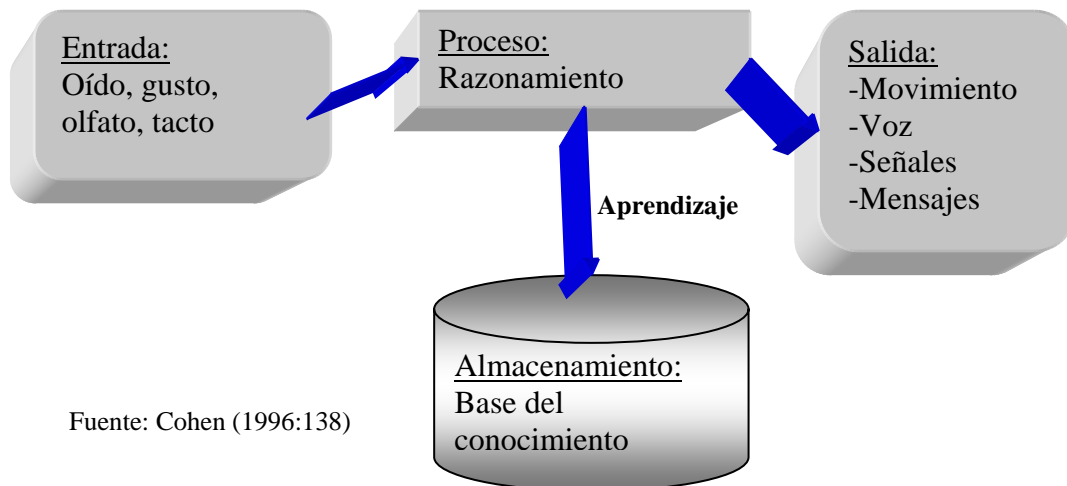
1.6.5 Nivel estratégico

En el nivel estratégico se encontrarían las hojas de cálculo, por su capacidad de simulación. Los llamados sistemas de información para ejecutivos dominan las decisiones menos estructuradas, con tareas como la planificación estratégica y la contabilidad directiva.

1.7 Los sistemas de información con tecnología de inteligencia artificial.

Las organizaciones están empleando la tecnología de la IA (véase figura 7), conforme las decisiones van siendo menos estructuradas, al mismo tiempo, las RNA pueden cubrir un hueco importante en las decisiones no estructuradas, debido a esa capacidad de encontrar relaciones complejas entre los patrones de entrada.

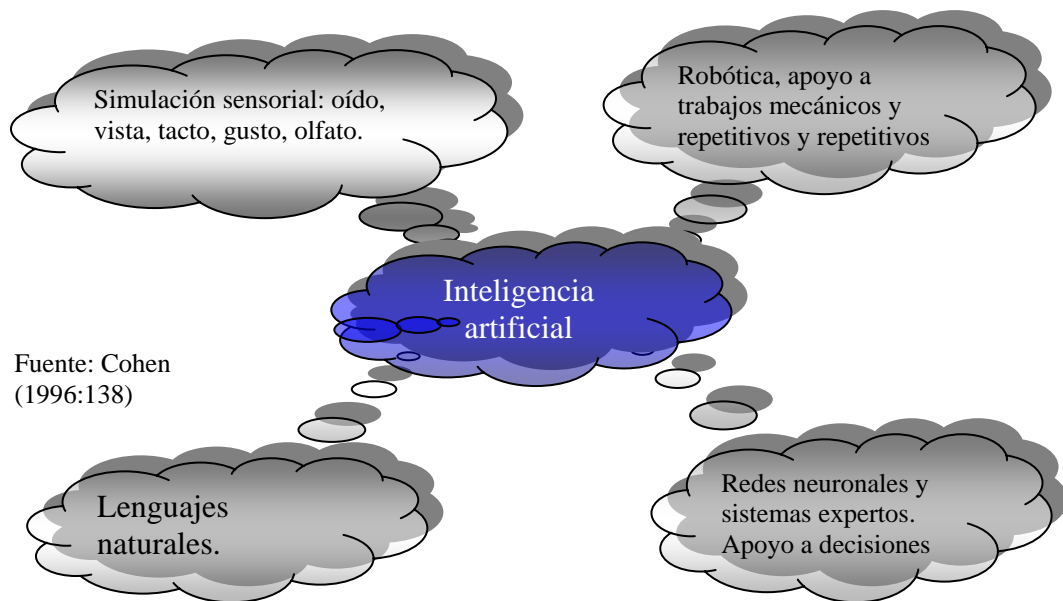
Figura 7. Imitación del comportamiento inteligente y su relación con las funciones de un sistema de información.



La mayoría de las decisiones gerenciales están basadas en pronósticos. Cada decisión se hace efectiva en algún punto en el futuro, por lo tanto debería estar basada en pronósticos de las condiciones futuras. La arquitectura de una red neuronal puede ser entrenada para predecir los valores futuros de las variables dependientes. Los requerimientos son el diseño del paradigma de la red y sus parámetros. El acercamiento de redes neuronales de retroalimentación de capas

múltiples consiste en una capa de entrada, una o varias capas ocultas y una capa de salida o resultado. Otro acercamiento es conocido como la red neuronal parcialmente recurrente, la cual puede aprender secuencias a medida que el tiempo transcurre y responde de manera diferente a los mismos patrones de estímulos de entrada a diferentes períodos de tiempo, dependiendo por supuesto de los distintos patrones de entrada. Ninguno de estos acercamientos es superior a cualquiera de los otros en cualquiera de los casos; sin embargo, una retroalimentación empapada que posea las características de una memoria dinámica, mejorará el funcionamiento de ambos acercamientos, (véase figura 8).

Figura 8. Áreas de estudio e investigación de la inteligencia artificial.



En el caso de estudio que nos ocupa plantearemos una metodología para obtener la arquitectura de red neuronal para mejorar la toma de decisiones no estructuradas, tales como la clasificación adecuada de tutores, por lo cual se muestra a continuación, la situación del Programa de Posgrado en Ciencias de la Administración de la UNAM y posteriormente en forma breve los conceptos de redes neuronales artificiales, sobre los que se basa este trabajo de investigación.

Capítulo 2. El Programa de Posgrado en Ciencias de la Administración de la UNAM

2.1 Antecedentes

Según, el REGLAMENTO GENERAL DE ESTUDIOS DE POSGRADO¹, “Son estudios de posgrado los que se realizan después de los estudios de licenciatura y tienen como finalidad la formación de académicos y profesionales del más alto nivel. Al término de los estudios de posgrado se otorgará el grado de maestro, grado de doctor o diploma de especialización. Estarán organizados en forma de programas de estructura flexible y procurarán la participación conjunta de las entidades académicas que cultivan disciplinas o ramas afines del conocimiento, conforme a las disposiciones contenidas en este Reglamento y a los lineamientos generales que, dentro del marco de sus atribuciones, formulen los Consejos Académicos de área, conociendo la opinión de los respectivos Consejos Técnicos, y apruebe el Consejo Universitario”.

En lo que se refiere a las entidades académicas² el Reglamento establece que: “Las escuelas, facultades, institutos y centros serán responsables de los programas de posgrado y, para efectos de este Reglamento, se denominarán entidades académicas.

Para continuar con el análisis de esta investigación, revisaremos la situación actual del Programa de Posgrado en Ciencias de la Administración, después el Sistema Tutorial Generacional y finalmente la relación entre ambos, para entender la importancia en la eficiencia Terminal.

2.2 Situación actual del Programa de Posgrado en Ciencias de la Administración de la UNAM

El Programa de Posgrado en Ciencias de la Administración de la UNAM, se muestra en base a su actual situación: en cuanto a la matrícula, la planta académica, el número de alumnos egresados de mayo de 2004 a mayo de 2005 y finalmente las líneas de investigación

¹ REGLAMENTO GENERAL DE ESTUDIOS DE POSGRADO UNAM (Aprobando por el Consejo Universitario en la sesión del 14 de diciembre de 1995) TÍTULO I, CAPÍTULO ÚNICO, Disposiciones Generales

² REGLAMENTO GENERAL DE ESTUDIOS DE POSGRADO UNAM, TÍTULO II, CAPÍTULO ÚNICO. Disposiciones Generales, Artículo 2

contempladas en el “*Plan de Programa de Posgrado en Ciencias de la Administración*”, Volumen I, 1999, Fondo Editorial FCA.

2.2.1 Matrícula

Actualmente el número de participantes registrados a los procesos de admisión al Programa de Posgrado en Ciencias de la Administración, es de 2,989, aceptados 1,431, distribuidos en doctorado 46, maestrías 1,066, especializaciones y sedes externas 319; estos datos se muestran distribuidos por campo de conocimiento en la tabla 3 del Anexo A, según datos información del Dr. Ricardo Varela [Varela, 2005].

La matrícula actual del Programa de Posgrado es de 1,339 alumnos distribuidos según se muestra en la Tabla 4, del Anexo A.

2.2.2 Planta académica

Para atender a este número de alumnos se requiere de una plantilla suficiente de maestros de tiempo completo y asignatura y de un sistema de tutores para contribuir así a una mejor formación de los alumnos mediante la atención personalizada de éstos para reforzarles los valores, actitudes, hábitos y habilidades y esto les permita un mejor desarrollo académico-profesional. Por tal razón, se establece el “Sistema Tutorial Generacional” en la UNAM en el semestre 2004-1 para cumplir con lo establecido en el Reglamento General de Estudios de Posgrado, en el sentido de que cada alumno de maestría deberá contar con un tutor desde que ingresa a la maestría hasta concluir sus estudios.

La planta de profesores actualmente, es de 472 y está integrada como se muestra en la tabla 5, del Anexo A:

2.2.3 Graduación

De mayo de 2003 a mayo de 2005 se han graduado 528 alumnos, según la Tabla 6 de Graduados por Campo de conocimiento del Anexo A.

La tabla 7 muestra el éxito alcanzado en eficiencia Terminal para el 2004, y esto se logró por la constante interacción del Cuerpo Tutorial, los alumnos y autoridades. Actualmente se cuenta con 232 tutores.

Tabla 7. Eficiencia Terminal

Campo de Conocimiento	Compromiso	Alcanzado	% Alcanzado
Especializaciones	75	107	143 %
Maestrías	115	265	230 %
Doctorado	10	14	140 %
Total	200	386	193 %

Fuente: Varela, Ricardo A. "Informe de Actividades 2003-2005", 2005

2.2.4 Líneas de investigación.

Las líneas de investigación³ que actualmente se desarrollan dentro de las áreas de conocimiento en el Programa de Posgrado en Ciencias de la Administración son 108 y se muestran en la Tabla 8 del Anexo A.

2.3 Antecedentes del Programa Tutorial de posgrado.

El *Programa Tutorial de Posgrado* en la UNAM⁴ establecido por el REGLAMENTO GENERAL DE ESTUDIOS DE POSGRADO, consiste en un conjunto de actividades que realizan tutores y alumnos relacionados con la integración, la retroalimentación del proceso educativo, la motivación del estudiante, el desarrollo de habilidades para el estudio y el trabajo, la orientación y el apoyo académico.

Este programa fue establecido en el Posgrado en el semestre 2004-1 para cumplir con el Reglamento General de Estudios de Posgrado.

El Programa del Doctorado, cuenta con 213 tutores, 98 de tiempo parcial y 114 de tiempo completo.

El Programa de Posgrado en Ciencias de la Administración se aprobó el 29 de enero de 1999, por el Consejo Académico de Área de Ciencias Sociales; conformado por un doctorado único en Ciencias de la Administración y seis maestrías: Maestría en Administración con cuatro vertientes (Organizaciones, Negocios Internacionales, Sistemas de Salud e Industrial), Maestría en Finanzas y Maestría en Auditoría. El 15 de marzo del 2001 se aprobó el Programa de Especializaciones en Ciencias de la Administración, conformado por las especializaciones en:

³ *Programa de Posgrado en Ciencias de la Administración*, Volumen I, 1999, Fondo Editorial FCA, (Págs. 26-32)

⁴ REGLAMENTO GENERAL DE ESTUDIOS DE POSGRADO, CAP III, Art. 25 al 28

Recursos Humanos, Alta Dirección, Mercadotecnia, Fiscal y Contraloría Pública⁵.

⁵ Varela, Ricardo A. "Informe de Actividades 2003-2005", 2005

Capítulo 3. La Inteligencia Artificial y las Redes Neuronales Artificiales

Dentro de la IA se pueden distinguir dos grandes áreas: la IA simbólica y la IA subsimbólica, la primera se ocupa de la construcción de sistemas “inteligentes”, éstos siguen un esquema de arriba hacia abajo (en inglés *top-down*) ya que es necesario disponer de una aproximación a la solución del problema y diseñarla completamente. Los sistemas expertos siguen este esquema, trabajan sobre una base de conocimiento de un experto sobre una materia, mediante mecanismos de inferencia parecidos a los que empleamos al razonar. En la IA subsimbólica no se realizan diseños a alto nivel capaces de resolver los problemas utilizando las técnicas de la disciplina, sino que se parte de sistemas genéricos que van adaptándose y construyéndose hasta formar por sí mismos un sistema capaz de resolver el problema. La perspectiva subsimbólica trata de estudiar los mecanismos de los sistemas nerviosos, del cerebro, así como su estructura, funcionamiento y características lógicas, con la intención de diseñar programas basados en dichas características que se adapten o generen sistemas capaces de resolver problemas. En este caso el diseño es de abajo hacia arriba (*bottom-up*), ya que los sistemas diseñados son simples e idénticos, recogen las características físicas de los sistemas que tratan de imitar, y se van generando cómputos cada vez más complejos, de forma automática, mediante mecanismos prefijados de aprendizaje. Es en este campo donde se encuadran las redes neuronales artificiales (Viñuela 2004:1-2).

Los antecedentes de las RNA parten precisamente de la necesidad de mejorar los problemas que presenta la arquitectura Von Neuman, la cual ha sido la base de los sistemas actuales de procesamiento digitales desarrollados a fines del siglo XIX; los problemas de esta arquitectura surgen al momento de abordar tareas del *mundo real*, donde se maneja información masiva, imprecisa y distorsionada [Martín y Sanz, 2002].

Por lo anterior los científicos han estudiado este tipo de modelos basados en la IA de donde se desprende el procesamiento paralelo distribuido (PDP) o también conocido como *redes neuronales artificiales* o también denominadas *sistemas neuronales artificiales (ANN Artificial Neural Networks)*¹. Dentro de la IA existen otros modelos de procesamiento y control como los sistemas expertos y algoritmos genéticos.

¹ Reciben también otras denominaciones como neurocomputación (Neural Computing) o sistemas conexionistas.

Estas nuevas tecnologías permiten incorporar un cierto tipo de “*inteligencia*” en un sistema de procesamiento y control para lograr una mejor toma de decisiones.

En pocas palabras, el objetivo de las RNA es llegar a diseñar máquinas con elementos de procesamiento paralelo, de modo que el comportamiento global de esa red “emule”, hasta cierto punto de la forma más simple posible, los sistemas neuronales biológicos, (Viñuela, 2004:2). El trabajo de investigadores como J.J. Hopfield y T. Kohonen, quienes introdujeron interesantes contribuciones al estudio de redes neuronales, en particular, Hopfield [Hopfield, 1982] quien usa la idea de una función de paso por umbral en las redes neuronales artificiales discretas. El factor clave de este modelo está en la capacidad de aprendizaje; posteriormente, Donald Hebb desarrolló un procedimiento matemático de aprendizaje en su libro *Organization of Behavior* (Hebb, 1949).

Los diversos modelos de las RNA se han inspirado en el sistema nervioso, emulando las características que se consideran esenciales para reproducir tareas como memoria asociativa, clasificación o extracción de rasgos, (Bonsón, 1999:33).

Dado que las RNA tratan de representar el conocimiento inspirándose en la estructura neuronal del cerebro, se puede considerar que una RNA es un sistema de procesamiento de información con ciertas características de comportamiento en común a las redes neuronales biológicas, por lo que se incorpora a este trabajo, de manera muy breve, los fundamentos biológicos de los sistemas neuronales.

3.1 Fundamentos biológicos de las Redes Neuronales

3.1.1 El cerebro y la neurona biológica

La teoría y modelado de redes neuronales artificiales está inspirada en la estructura y funcionamiento de los sistemas nerviosos, donde la neurona es el elemento fundamental.

Las redes neuronales son modelos que intentan reproducir el comportamiento del cerebro.

Rodolfo Llinás, uno de los padres de la neurociencia, en su libro “*El cerebro y el mito del yo*”, opina que el cerebro debe considerarse como una entidad viva que genera una actividad eléctrica definida y que esta actividad se puede describir como tormentas eléctricas “auto

controladas”, el cerebro vivo con sus tormentas eléctricas, no son más que descripciones que representan aspectos distintos de una misma cosa: el estado funcional de las neuronas. Por otro lado, el Dr. Lair Ribeiro en su libro “*Inteligencia Aplicada*”, menciona que el sistema nervioso tiene alrededor de cien mil millones de neuronas y que el elemento estructural y funcional más esencial, en el sistema de comunicación neuronal, es la *neurona*, siendo ésta la célula especializada en la obtención y transmisión de datos, para ello utiliza procesos electroquímicos y que el tamaño de una neurona puede oscilar entre 4 y 100 micras y su forma puede ser variada, que posee extensiones especializadas llamadas *dendritas*, que reciben la información, y *axones*, que la transmiten. Según Llinás, [2003:97], las neuronas facilitan y organizan la complejidad creciente de las transformaciones sensomotoras. Los autores Viñuela y Galván, [2004:3], describen la actividad de las neuronas mencionando que la mayoría utilizan productos de secreción como señales químicas (transmisores) para la transmisión de la información. Dicha información se envía, entre las distintas neuronas, a través de prolongaciones formando redes, en las cuales se elabora y almacena información. Además, una parte de las neuronas está en relación con receptores, a través de los cuales llegan comunicaciones procedentes del exterior o del interior del organismo hasta las redes neuronales.

3.1.2 Estructura de una neurona

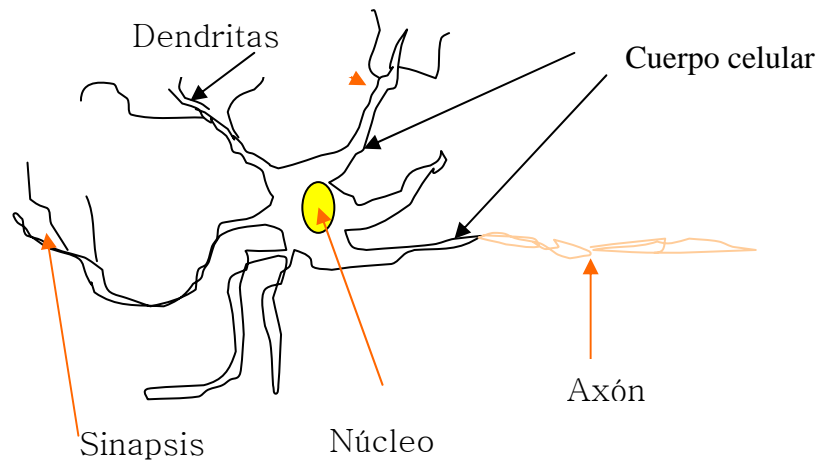
La neurona es una célula viva, por lo cual, contiene los mismos elementos que forman parte de todas las células biológicas. Una de las características que diferencian a las neuronas del resto de las células vivas, es su capacidad de comunicarse.

El Dr. Lair Ribeiro, [2003:22], describe la estructura de una neurona como sigue: (véase las Figuras 9 y 10)

1. Dendritas son las principales receptoras de la neurona.
2. Cuerpo celular
3. Núcleo es la unidad que contiene la información genética.
4. Axones son las principales conductoras de la neurona
5. Terminales presinápticos es una región en que las ramificaciones de los axones de una neurona (presináptica) transmiten señales a otra neurona (postsináptica). Las ramificaciones de un único axón pueden formar sinapsis con otras mil neuronas.

6. Capa de mielina es una sustancia grasa que ayuda a los axones a transmitir mensajes con mayor rapidez.

Figura 9. La neurona biológica



Fuente: [Ribeiro, 2003:22]

3.1.3 La misión de las neuronas

La misión de las neuronas comprende generalmente cinco funciones parciales, Viñuela y Galván, [2004:3]:

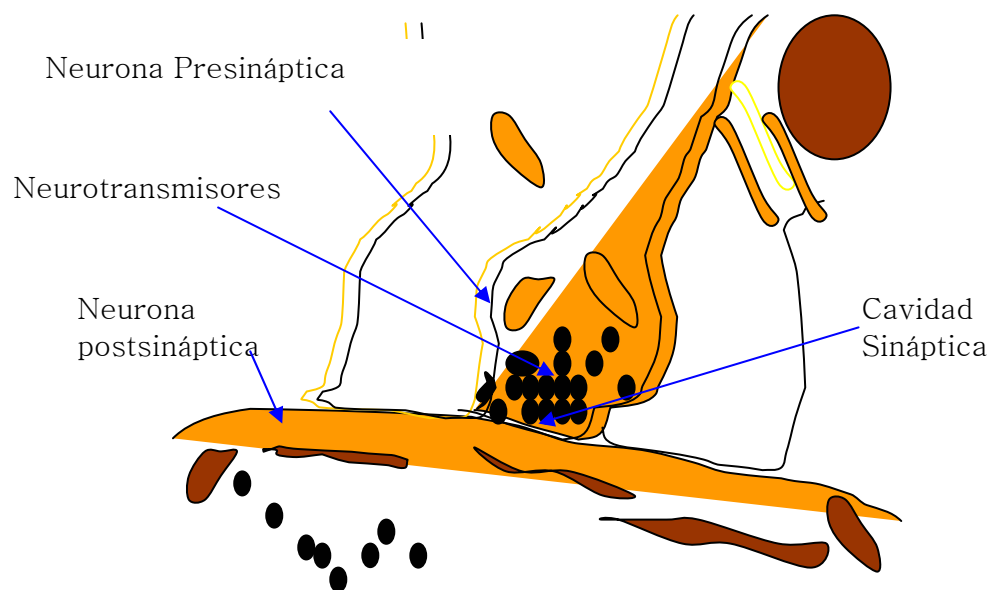
- Las neuronas recogen la información que llega a ellas en forma de impulsos procedentes de otras neuronas o receptores.
- La integran en un código de activación propio de la célula.
- La transmiten codificada en forma de frecuencia de impulsos a través de su axón.
- A través de sus ramificaciones el axón efectúa la distribución espacial de los mensajes.
- En sus terminales transmite los impulsos a las neuronas subsiguientes o a las células efectoras.

Una neurona recoge señales procedentes de otras neuronas a través de una pléyade de delicadas estructuras llamadas *dendritas*. La neurona emite impulsos de actividad eléctrica a lo largo de una fibra larga y delgada denominada *axón*, que se ramifica en millares de conexiones con otras neuronas.

Las señales nerviosas se pueden transmitir eléctrica o químicamente. La transmisión química prevalece fuera de la neurona, mientras que la eléctrica lo hace en el interior. La transmisión química se basa en el intercambio de neurotransmisores, mientras que la eléctrica hace uso de descargas que se producen en el cuerpo celular, y se propaga por el axón, a esta comunicación entre neuronas se le denomina *sinapsis*. [Ribeiro, 2003:23], (Véase figura 10).

La *sinapsis* son los elementos de unión entre axón y dendritas. El funcionamiento del cerebro será, en general como una enorme malla que propaga señales electro-químicas de unas neuronas a otras. La concentración iónica en la sinapsis es muy importante, según Viñuela (2004:4), dado que las neuronas no son elementos lineales: no se encargan simplemente de acumular la acumulación de las señales recibidas por sus dendritas, sino que funcionan a saturación; producen una señal de activación si la señal recibida supera un cierto umbral, permaneciendo inhibidas mientras tanto.

Figura 10. Sinapsis

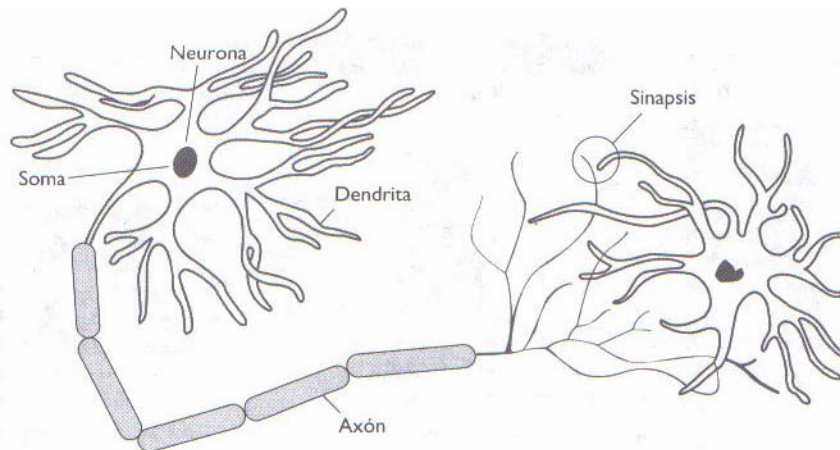


Fuente: [Ribeiro, 2003:30]

3.2 Redes neuronales biológicas

Las redes neuronales están diseñadas para imitar el proceso del pensamiento físico del cerebro biológico, la Figura 11 muestra dos neuronas del cerebro de una sanguijuela, (Laudon, 2004:333). Estos modelos sencillos han influido en el desarrollo de las RNA.

Figura 11. Neuronas biológicas de una sanguijuela.



Fuente: Laudon, [2004:333], Agencia de Proyectos de Investigación avanzada de la defensa (DARPA), 1988. No clasificado.

Algunas de las estructuras neuronales son determinadas en el nacimiento, otra parte es desarrollada a través del aprendizaje, proceso en que nuevas conexiones neuronales son realizadas y otras se pierden por completo. El desarrollo neurológico se hace crítico durante los primeros años de vida; por ejemplo, está demostrado que si a un cachorro de gato, se le impide usar uno de sus ojos durante un periodo corto de tiempo, él nunca desarrollara una visión normal en ese ojo.

Las estructuras neuronales continúan cambiando durante toda la vida, estos cambios consisten en el refuerzo o debilitamiento de las uniones sinápticas; por ejemplo, se cree que nuevas memorias son formadas por la modificación de esta intensidad entre sinapsis, así el proceso de recordar el rostro de un nuevo amigo, consiste en alterar varias sinapsis.

Como consecuencia de los primeros estudios sobre la base neuronal de los sistemas numéricos (relacionados con la memoria), se creía que el almacenamiento de la memoria asociativa, tanto implícita como explícita, requería de un circuito neuronal muy complejo. Entre quienes comenzaron a oponerse a este enfoque se hallaba Donald O. Hebb (1988:49) profesor de la universidad de Milner; Hebb sugirió que el aprendizaje asociativo podría ser producido por un mecanismo celular sencillo y propuso que las asociaciones podrían formarse por una actividad neuronal coincidente: "*Cuando un axón de una célula A excita otra célula B y participa en su activación, se produce algún proceso de desarrollo o cambio metabólico en una o en ambas células, de suerte que la eficacia de A, como célula excitadora de B, se intensifica*" (Hebb, 1988:49). Según la regla Hebbiana de aprendizaje, el

que coincida la actividad de las neuronas presinápticas (suministran el impulso de entrada) con la de las postsinápticas (reciben el impulso) es muy importante para que se refuerce la conexión entre ellas, este mecanismo es llamado pre-postasociativo.

Todas las neuronas conducen la información de forma similar (Ribeiro, 2003:31), ésta viaja a lo largo de axones en breves impulsos eléctricos, denominados potenciales de acción; los potenciales de acción que alcanzan una amplitud máxima de unos 100 MV (milésima de voltio) y duran 1 MS (milisegundo), son resultado del desplazamiento a través de la membrana celular de iones de sodio dotados de carga positiva, que pasan desde el fluido extracelular hasta el citoplasma intracelular; la concentración extracelular de sodio supera enormemente la concentración intracelular.

Muchas clases de neuronas del sistema nervioso están dotadas de tipos de actividad eléctrica intrínseca que les confiere propiedades funcionales características. Esta actividad eléctrica se manifiesta con variaciones diminutas de voltaje (del orden de milésimas de voltio) a través de la membrana que rodea a la célula (la membrana plasmática neuronal), (Llinás, 2003:97).

Cuando el desplazamiento de carga hace que la membrana se aproxime al umbral de generación de potenciales de acción, se produce un efecto excitador y cuando la membrana resulta estabilizada en la vecindad del valor de reposo se produce un efecto inhibitor. Cada sinapsis produce sólo un pequeño efecto, para determinar la intensidad (frecuencia de los potenciales de acción) de la respuesta cada neurona ha de integrar continuamente hasta unas 1000 señales sinápticas, que se suman en el soma o cuerpo de la célula.

Las sinapsis pueden ser a) *excitadoras* o b) *inhibitorias* según el neurotransmisor que libere, cada neurona recibe de 10.000 a 100.000 sinapsis y su axón realiza una cantidad similar de sinapsis. Las *sinapsis excitadoras* cuyos neurotransmisores provocan disminuciones de potencial en la membrana de la célula postsináptica, facilitando la generación de impulsos a mayor velocidad, mientras que las *sinapsis inhibitorias* sus neurotransmisores tienden a estabilizar el potencial de la membrana, dificultando la emisión de impulsos. En cada instante, algunas neuronas estarán activas y otras en reposo; la suma de los efectos excitadores o inhibidores determina si la célula será o no estimulada. Hilera, (2000:49)

De esta manera, la información se transmite de unas neuronas a otras y va siendo procesada a través de las conexiones sinápticas y las

propias neuronas. El *aprendizaje* de las redes neuronales se produce mediante la *variación de la efectividad de la sinapsis*, de esta manera cambia la influencia que unas neuronas ejercen sobre otras, de aquí se deduce que la arquitectura, el tipo y la efectividad de las conexiones en un momento dado, representan en cierto modo la memoria o estado de conocimiento de la red.

Según Hilera, (2000:63); “biológicamente, se puede aceptar que la información memorizada en el cerebro está mas relacionada con los valores sinápticos de las conexiones entre las neuronas que con ellas mismas; es decir, el conocimiento se encuentra en la sinapsis. En el caso de las redes neuronales artificiales, se puede considerar que el conocimiento se encuentra representado en los *pesos* de las conexiones entre neuronas”. El proceso de aprendizaje en una red neuronal artificial implica cierto número de cambios en las conexiones, por lo que puede decirse que *aprende* modificando los pesos de la red.

3.3 Redes neuronales artificiales

Las redes neuronales surgieron del movimiento conexionista, que nació junto con la IA simbólica o tradicional. Esto fue hacia los años 50 del siglo pasado, con algunos de los ordenadores de la época y las posibilidades que ofrecían. La IA simbólica se basa en que todo conocimiento se puede representar mediante combinaciones de símbolos, derivados de otras combinaciones que representan verdades incuestionables o axiomas. Así pues la IA tradicional asume que su conocimiento es independiente de la estructura simbólica que maneje, siempre y cuando la “máquina” realice algunas operaciones básicas entre ellos. En contraposición, los “conexionistas” intentan representar el conocimiento desde el estrato más básico de la inteligencia: el estrato físico, creen que el secreto para el aprendizaje y el conocimiento se halla directamente relacionado con la estructura del cerebro: concretamente con las neuronas y la interconexión entre ellas, de tal forma que, trabajan con grupos de neuronas artificiales, llamadas RNA, [Martín y Sanz, 2002].

La computadora posee la arquitectura de *Von Neumann*: un simple procesador que puede realizar operaciones simples aritméticas, lógicas y condicionales una a la vez, y una gran memoria.

El cerebro humano, en contraste, consta de un gran número de células especializadas llamadas neuronas, interconectadas entre sí (se estima que hay del orden de diez billones de neuronas en el cerebro

humano con un promedio de miles de conexiones por cada una). Las características del cerebro en contraste con una computadora son:

1. Es robusto y tolerante a fallas, diariamente mueren neuronas sin afectar su desempeño.
2. Es flexible, se ajusta a nuevos ambientes por aprendizaje, no hay que programarlo.
3. Puede manejar información difusa o inconsistente.
4. Es altamente paralelo
5. Es pequeño, compacto y consume poca energía.

3.3.1 ¿Qué es una red neuronal artificial?

Las RNA's son modelos computarizados inspirados en la estructura a bajo nivel del cerebro. Consisten en grandes cantidades de unidades de procesamiento sencillas llamadas neuronas, conectadas por enlaces de varias fuerzas. Las RNA también pueden ser construidas con "hardware" especial o simuladas en computadoras normales.

Kohonen, (1982:59) define una RNA como “una red de elementos simples (usualmente adaptativos) masivamente interconectados en paralelo y con organización jerárquica, los cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico”.

En síntesis, se puede considerar que una RNA es un sistema de procesamiento de información con ciertas características de comportamiento en común a las redes neuronales biológicas.

Las RNA's han sido aplicadas en un gran número de problemas reales de complejidad considerable. Su más importante ventaja es la de resolver problemas que son muy complejos para tecnologías convencionales, problemas que no tienen una solución determinística.

El autor Schalkoff (1997:2), en su libro “*Artificial Neural Networks*”; afirma que, “la estructura de una red está compuesta de un número de unidades interconectadas (neuronas artificiales). Cada unidad tiene una entrada/salida característica y se implementa con un cálculo local o función. La salida de cualquier unidad está determinada por su entrada/salida característica, su interconexión con otras unidades, y posibles entradas externas. La red desarrolla su funcionalidad a través de una o más formas de entrenamiento”.

Las RNA, son una tecnología computacional que puede utilizarse en un gran número y variedad de aplicaciones, tanto comerciales como

militares. Se pueden desarrollar redes neuronales en un periodo de tiempo razonable y pueden realizar tareas concretas mejor que otras tecnologías convencionales, incluyendo a los sistemas expertos. Las redes neuronales presentan una alta tolerancia a fallos del sistema y proporcionan un alto grado de paralelismo en el proceso de datos. Por esta razón es posible implementar redes neuronales de bajo costo.

La importancia de las RNA radica en que pueden resolver problemas como los de visión o aprendizaje, y su procesamiento en paralelo resulta necesario si se quiere lograr respuestas en tiempo real [Martín y Sanz, 2002]. Inspiradas en el sistema nervioso creado por la naturaleza, la tecnología de redes neuronales se utiliza para resolver un amplio abanico de complejos problemas científicos, de ingeniería y empresariales. Entre sus aplicaciones comerciales destacan la negociación de carteras de inversión, la minería de datos, el control de procesos, la supresión de ruidos, la compresión de datos y el reconocimiento de la voz, entre otros. Las redes neuronales resultan especialmente adecuadas para estos problemas ya que, al igual que su equivalente biológico, una red neuronal puede aprender y, por tanto, puede entrenarse para encontrar soluciones, reconocer patrones, clasificar datos y predecir acontecimientos futuros. A diferencia de los enfoques analíticos que se aplican generalmente en campos como la estadística y la teoría de control, las redes neuronales no precisan de un modelo explícito ni de premisas restrictivas de normalidad o linealidad. El comportamiento de una red neuronal se define por la forma en que se conectan sus elementos de computación individuales y por la fuerza de dichas conexiones, o ponderaciones. Las ponderaciones se ajustan automáticamente entrenando la red de acuerdo con una regla de aprendizaje especificada hasta que realiza la tarea deseada de forma correcta.

Originalmente la red neuronal no dispone de ningún tipo de conocimiento útil almacenado. Para que ejecute una tarea es preciso entrenar o enseñar a la red neuronal. El entrenamiento de una red neuronal se realiza mediante patrones-ejemplo, existiendo dos tipos de aprendizaje: *supervisado* y *no supervisado*. En el aprendizaje *supervisado*, se incorpora a la red parejas de patrones entrada-salida y la red neuronal aprende a asociarlos. Si el entrenamiento es *no supervisado* se suministran a la red únicamente los datos de entrada para que extraiga los rasgos característicos esenciales. Estas redes neuronales no supervisadas están relacionadas con modelos estadísticos como el análisis de conglomerados o las escalas multidimensionales.

Durante la fase de aprendizaje en la mayor parte de los modelos se produce una variación de los pesos sinápticos, es decir, de la intensidad de interacción entre las neuronas, lo que en terminología estadística equivale a calcular los coeficientes de las funciones de ajuste.

Los diferentes modelos neuronales se diferencian en el modelo de neurona, su organización, forma de las conexiones y el algoritmo de aprendizaje que emplea. Existen multitud de modelos y variantes, más de cincuenta, como son el modelo de Hopfield, Kohonen, el *counter-propagation*, la resonancia adaptativa o ART, el neocognitrón, etc. (Véase tabla 10).

La razón fundamental de emplear RNA reside en su gran capacidad de discernimiento en el reconocimiento de patrones, aun en presencia de "ruido" en los datos de entrenamiento de una red. Esto ha conducido al empleo de RNA en una gran cantidad de aplicaciones, dentro de los más variados ámbitos: medicina, finanzas, ciencias administrativas, ingeniería, etc.

En la actualidad las RNA's aprenden a reconocer patrones por medio de un entrenamiento basado en varios ejemplos diferentes. Ellas son eficientes en el reconocimiento de patrones en un conjunto de datos. También son efectivas en el aprendizaje de patrones a partir de datos incompletos, con ruido y hasta compuestos por ejemplos contradictorios. La habilidad de manipular estos datos imprecisos, hace que las redes sean muy eficaces en el procesamiento de información sin reglas claras o que no puedan ser formuladas fácilmente

Las neuronas se modelan mediante *unidades de proceso* (Figura 12). Cada unidad de proceso se compone de una red de conexiones de entrada, una función de red (de propagación), encargada de computar la entrada total combinada de todas las conexiones, un núcleo central de proceso, encargado de aplicar la función de activación, y la salida, por donde se transmite el valor de activación a otras unidades.

La función de red es típicamente la sumatoria ponderada, mientras que la función de activación suele ser alguna función de umbral o una función sigmoideal.

- ***Función de propagación o de red.*** Calcula el valor de base o entrada total a la unidad, generalmente como simple suma ponderada de todas las entradas recibidas, es decir, de las entradas multiplicadas por el peso o valor de las conexiones.

Equivale a la combinación de las señales excitatorias e inhibitorias de las neuronas biológicas.

- **Función de activación.** Es quizás la característica principal o definitoria de las neuronas, la que mejor define el comportamiento de la misma. Se usan diferentes tipos de funciones, desde simples funciones simples de umbral a funciones no lineales. Se encarga de calcular el nivel o estado de activación de la neurona en función de la entrada total.
- **Conexiones ponderadas.** Hacen el papel de las conexiones sinápticas, el peso de la conexión equivale a la fuerza o efectividad de la sinapsis. La existencia de conexiones determina si es posible que una unidad influya sobre otra, el valor de los pesos y el signo de los mismos definen el tipo (excitatorio/inhibitorio) y la intensidad de la influencia.
- **Salida.** Calcula la salida de la neurona en función de la activación de la misma, aunque normalmente no se aplica más que la función identidad, y se toma como salida el valor de activación. El valor de salida cumpliría la función de la tasa de disparo en las neuronas biológicas.

3.4. Características de las neuronas artificiales

La neurona artificial pretende mimetizar algunas de las características de las neuronas biológicas [Hilera, Martínez, 2000]. Cada neurona i -ésima está caracterizada en cualquier instante por un valor numérico denominado *valor o estado de activación para el cual existe una función de salida*, f_i , que transforma el estado actual de activación en una *señal de salida*, y_i . Dicha señal es enviada a través de los canales de comunicación unidireccionales a otras unidades de la red; en estos canales la señal se modifica de acuerdo con la sinapsis (el peso, W_i) asociada a cada uno de ellos según una determinada regla generando así la suma ponderada.

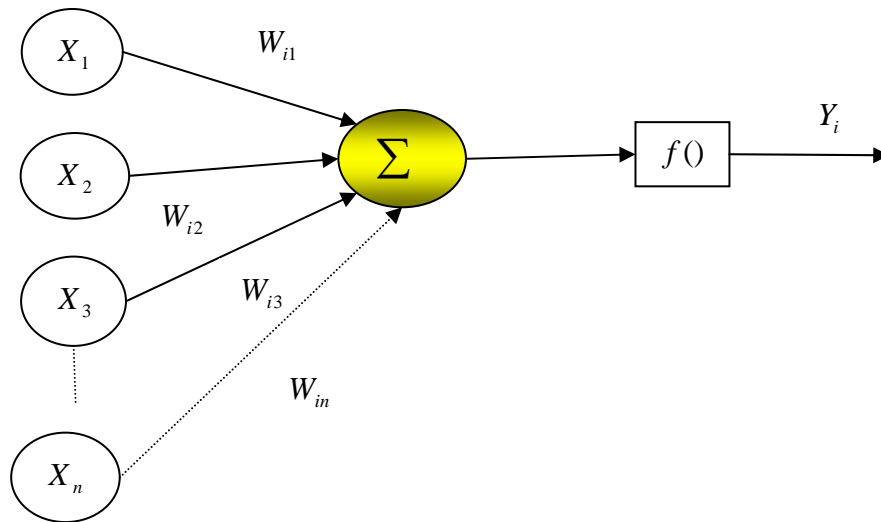
Las señales de entrada a una neurona artificial X_1, X_2, \dots, X_n son variables continuas en lugar de pulsos discretos, como se presentan en una neurona biológica. Cada señal de entrada pasa a través de una ganancia o peso, llamado peso sináptico o fortaleza de la conexión cuya función es análoga a la de la función sináptica de la neurona biológica. Los pesos pueden ser positivos (excitatorios), o negativos (inhibitorios), el nodo sumatorio acumula todas las señales de entradas multiplicadas por los pesos o ponderadas y las pasa a la salida a través de una función umbral o función de transferencia. La entrada neta a cada unidad puede escribirse de la siguiente manera:

$$neta_i = \sum_{i=1}^n W_i X_i = \vec{X} \vec{W}$$

Ecuación 1

Una idea clara de este proceso se muestra en la Figura 12 en donde puede observarse el recorrido de un conjunto de señales que entran a la neurona.

Figura 12. Modelo matemático de una neurona como unidad de proceso.



Fuente: [Martín y Sanz, 2001:19]

Las entradas X_n representan las señales que provienen de otras neuronas y que son capturadas por las dendritas.

- Los pesos W_i son la intensidad de la sinapsis que conectan dos neuronas; tanto X_n como W_i son valores reales.
- f es la función de transferencia.

Una vez que se ha calculado la activación del nodo, el valor de salida equivale a la ecuación 2.

$$x_i = f_i(neta_i)$$

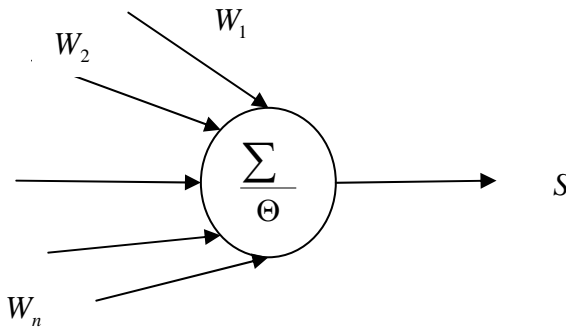
Ecuación 2

Donde f_i representa la función de activación para esa unidad, que corresponde a la función escogida para transformar la entrada $neta_i$ en el valor de salida x_i y que depende de las características específicas de cada red.

3.5 Primeros Modelos Computacionales

Isasi, Galván [2004:23], nos muestra el primer modelo, el cual fue propuesto por Warren McCulloch-Pitts y Walter Pitts en 1943 [McCulloch and Pitts, 1943]. El modelo tiene una estructura y funcionamiento simplificado de las células del cerebro con dos estados posibles: apagado (0) y encendido (1), (véase figura 13).

Figura 13. Esquema de una neurona de McCulloch-Pitts



$$S = \left\{ \begin{array}{ll} 1 & \text{Si } \sum X_i * W_i > \Theta \\ 0 & \text{en caso contrario} \end{array} \right.$$

Fuente: Isasi, Galván [2004:23]

La célula de McCulloch-Pitts recibe como entrada un conjunto de n valores binarios, $X = \{x_1, x_2, \dots, x_n\}$ procedente de las salidas de otras células, o de la entrada a la red; y produce una única salida también binaria, S . Cada célula se caracteriza por $n+1$ valores reales, de los cuales n son los pesos de las conexiones (w_i) correspondientes a las entradas x_i , y el otro es un valor de umbral θ , que puede ser distinto para cada célula. La célula opera en lapsos discretos. La forma de procesar la entrada es la siguiente: la neurona se activará y, por lo tanto, producirá un valor 1, si la suma de entradas multiplicada por los pesos supera al umbral θ .

$$S(t+1) = \begin{cases} 1 & \sum_i w_i x_i(t) > \theta \\ 0 & \text{en caso contrario} \end{cases}$$

A partir de este modelo, se define el primer modelo de red neuronal.

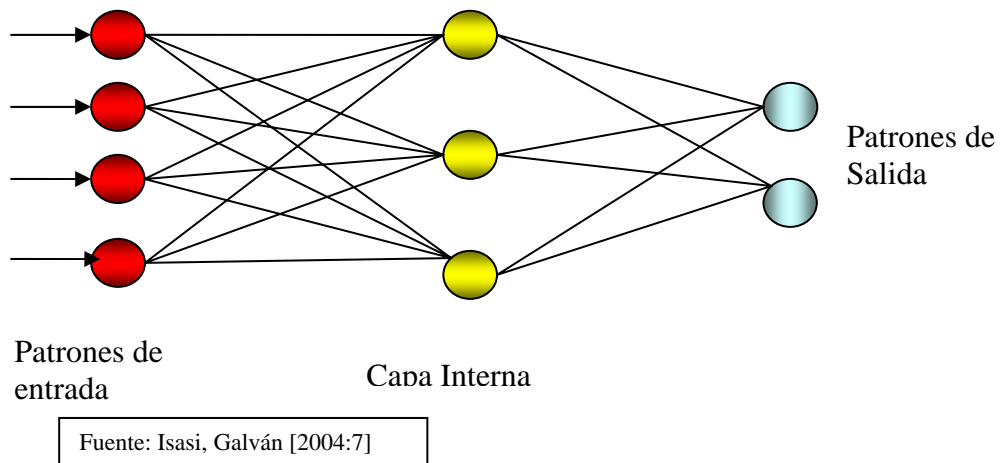
Una red neuronal es una colección de neuronas, según McCulloch y Pitts, todas las mismas escalas de tiempos, donde sus salidas están conectadas a las entradas de otras neuronas.

De este modo, una salida puede actuar sobre varias entradas, pero una entrada viene a lo sumo de una salida. La red tiene contacto con el exterior a través de líneas de entrada y salida. Las líneas de entrada de la red formarán parte de la entrada de alguna o de todas las neuronas de la red. Asimismo, las líneas de salida

Una RNA podría definirse, según [Isasi, Galván, 2004:7] como “un grafo cuyos nodos están constituidos por unidades de proceso idénticas, y que propagan información a través de los arcos. En este grafo se distinguen tres tipos de nodos: los de entrada, los de salida y los intermedios”.

La Figura 14 muestra el ejemplo de una unidad típica de proceso de una RNA. A la izquierda se ve una serie de entradas; cada una llega de la salida de otra neurona de la red. Una vez calculada de la salida de una neurona, ésta se propaga, vía conexiones de salida, a las células destino. Todas las conexiones de salida reciben el mismo valor de salida.

Figura 14. Esquema de una red de tres capas totalmente interconectadas



El funcionamiento de la red consiste en que para cada vector de entrada, éste es introducido en la red copiando cada valor de dicho vector en la neurona de la entrada correspondiente. Cada neurona de la red, una vez recibida la totalidad de sus entradas, las procesa genera una salida que es propagada a través de las conexiones entre neuronas, llegando como entrada a la neurona destino. Una vez que la entrada ha sido completamente propagada por toda la red, se producirá un vector de salida, cuyos componentes son cada uno de los valores de salida de las neuronas de salida.

El funcionamiento de una red de neuronas por capas como la de la figura 14 puede describirse mediante la ecuación 3:

$$\vec{S} = F(F(\vec{X} \cdot W_1) \cdot W_2). \text{ -----Ecuación 3}$$

Donde W_1 y W_2 son los pesos de la primera y segunda capa, respectivamente, F es la función de activación idéntica en todas las neuronas; \vec{X} es el vector de entrada a la red y \vec{S} es el vector de salida que la red produce. W_1 y W_2 son matrices de conexión entre las capas de la red, y por lo tanto se trata de multiplicaciones de matrices. La función de activación F desempeña un papel importante en un esquema de Red de Neuronas. Supóngase que se utiliza una función lineal como la siguiente: $F(x) = k \cdot x$. En este caso si sustituimos dicha función en la

ecuación 1 quedará como la ecuación 4 y la gráfica será una red de dos capas como se muestra en la figura 15:

$$\vec{S} = k \cdot (K \cdot \vec{X} \cdot W_1) \cdot W_2 \quad \text{--- Ecuación 4}$$

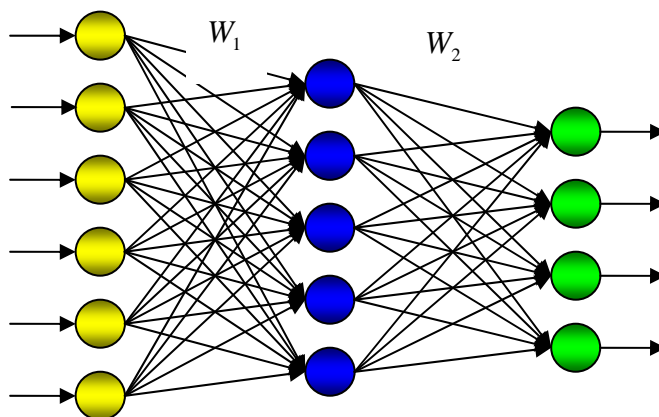
Lo cual es equivalente a una red con una sola capa de conexiones W_t donde:

$$W_t = k^2 \cdot W_1 \cdot W_2 \quad \text{----- Ecuación 5}$$

La ecuación 5 se muestra en la Figura 16.

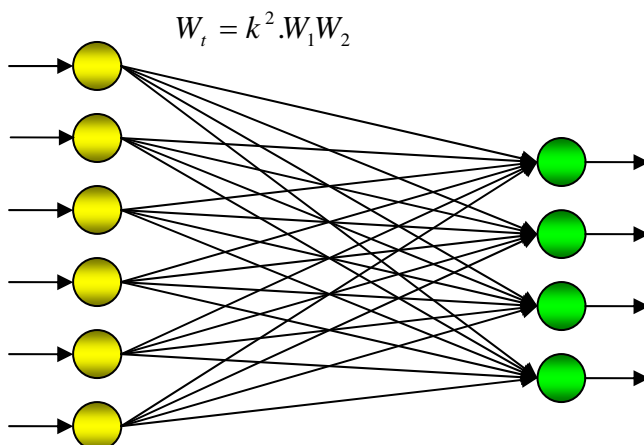
Dependiendo de la forma en que las neuronas se conectan entre sí se le denomina *patrón de conectividad o arquitectura de la red*. La estructura básica de interconexión entre neuronas, es la red multicapa, mostrada en la Figura 13. El primer lugar lo constituyen las entradas; estas unidades reciben los valores de unos patrones representados como vectores que sirven de entrada a la red. A continuación hay una serie de capas intermedias, llamadas ocultas, cuyas unidades responden a rasgos particulares que pueden aparecer en los patrones de entrada. Puede haber uno o varios niveles ocultos. El último nivel es el de salida. La salida de estas unidades sirve como salida de toda la red. Cada interconexión entre unidades de proceso actúa como una ruta de comunicación: a través de estas interconexiones viajan valores numéricos de una neurona a otra. Estos valores son evaluados por los pesos de las conexiones. Los pesos de las conexiones se ajustan durante la fase de aprendizaje para producir una red de neuronas artificiales final (Viñuela, 2004:9).

Figura 15. Red de dos capas



Fuente: Isasi Viñuela, 2004:9

Figura 16. Red de una capa equivalente



Fuente: Isasi Viñuela, 2004:9

3.6 Sistemas Neuronales Artificiales

Estos sistemas están compuestos por una gran cantidad de elementos de procesamiento (PE o nodos) que operan sobre la base de reconocimiento de patrones, y que pueden adquirir, almacenar y utilizar conocimiento experimental, obtenido a partir de ejemplos. Los PE se agrupan en capas, constituyendo una RNA.

3.7 Clasificación de los modelos de RNA

En general las redes neuronales se pueden clasificar de diversas maneras, según su *número de capas*, su *topología de red*, su *tipo de respuesta* o forma de aprendizaje (supervisado o no supervisado), la

forma de los datos de entrada y salida, Pérez (2003:28); un resumen de esta clasificación se observa en la figura 19 y en la tabla 1.

3.7.1 El número de capas

Las *redes monocapa* son las formadas por una sola capa (sin contar la capa de entrada). Las *redes multicapa* son las redes formadas por varias capas. (Figuras 17 y 18)

Figura 17. Red de una sola capa interna o monocapa

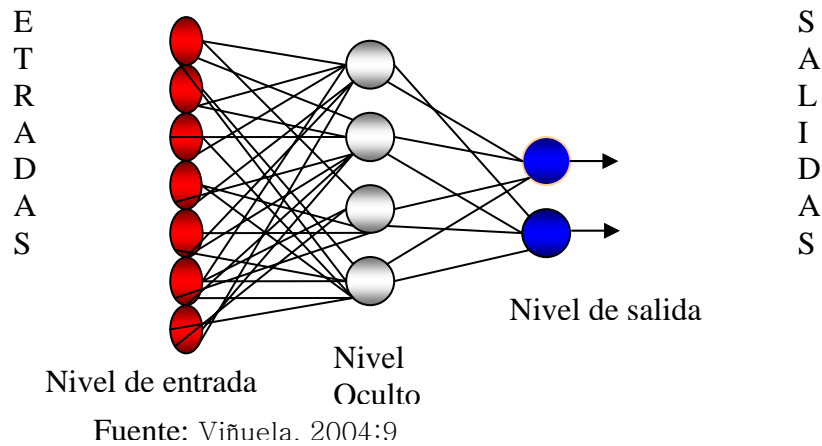
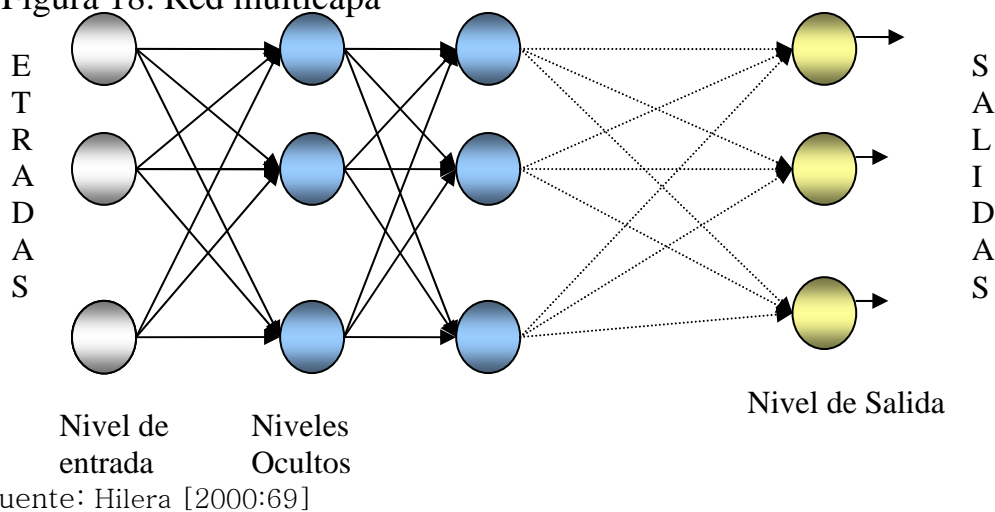


Figura 18. Red multicapa



3.7.2 La topología de la red

Una RNA está confeccionada y entrenada para llevar a cabo una labor específica. La *topología* de la red se refiere a los diferentes modelos de conexiones entre capas, en general se suelen distinguir dos básicos: las arquitecturas hacia adelante o *feedforward*, (la información se propaga hacia adelante) (véase figura 21), y las

realimentadas o *feedback* (las señales pueden fluir en ocasiones hacia atrás a través de lazos de realimentación, figura 22).

3.7.3 El tipo de respuesta

El aprendizaje es el proceso por el cual la red neuronal modifica sus pesos en respuesta a la información de entrada. Los cambios que se producen durante el proceso de aprendizaje se reducen a la destrucción, modificación y creación de conexiones entre neuronas. Una conexión se destruye cuando su peso se convierte en cero.

Durante el proceso de aprendizaje, los pesos de las conexiones de la red sufren modificaciones, por tanto se puede afirmar que este proceso ha terminado (la red ha aprendido) cuando los valores de los pesos permanecen estables ($dw_{ij}/dt = 0$).

Un aspecto importante respecto al aprendizaje en las redes neuronales es conocer cómo se modifican los valores de los pesos; es decir, cuáles son los criterios que se siguen para cambiar el valor asignado a las conexiones cuando se pretende que la red “*aprenda*” una nueva información.

Estos criterios determinan lo que se llama regla de aprendizaje de la red. De la forma general, se suelen considerar dos tipos de aprendizaje:

- Aprendizaje supervisado
- Aprendizaje no supervisado

La diferencia es que el aprendizaje supervisado tiene asignados valores en la capa de salida de la red neuronal; mientras que en el aprendizaje no supervisado no se le asignan valores a esta capa.

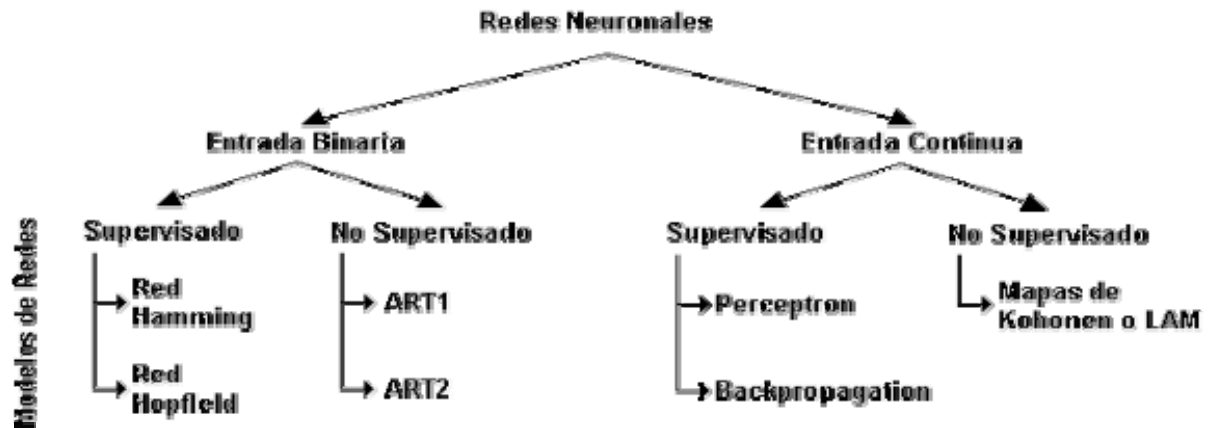
Las *redes autoasociativas* se entrenan para que asocien un patrón consigo mismo, de modo que, al presentar dicho patrón con un ruido superpuesto o de forma parcial, sean capaces de identificar el patrón, Pérez (2003:29).

Las *redes heteroasociativas* se entrenan para que ante la presentación de un patrón A respondan a otro diferente B.

La autoasociación implica aprendizaje no supervisado, mientras que la heteroasociación supone aprendizaje supervisado.

Las redes neuronales se pueden clasificar de acuerdo a su aprendizaje, véase la figura 19.

Figura 19. Clasificación de las Redes Neuronales de acuerdo a su tipo de aprendizaje.



Fuente: Hilera [2000:89]

3.7.4 La forma de los datos de entrada y salida

En algunas redes las entradas y las salidas son analógicas, es decir, tomas valores continuos. Otras redes solo aceptan valores secretos o binarios: $\{0,1\}$ ó $\{-1,1\}$ normalmente, y generan valores del mismo tipo. Existen otras redes que pueden aceptar entradas continuas y generar salidas discretas, Pérez (2003:29):

3.8 Componentes de una red neuronal artificial

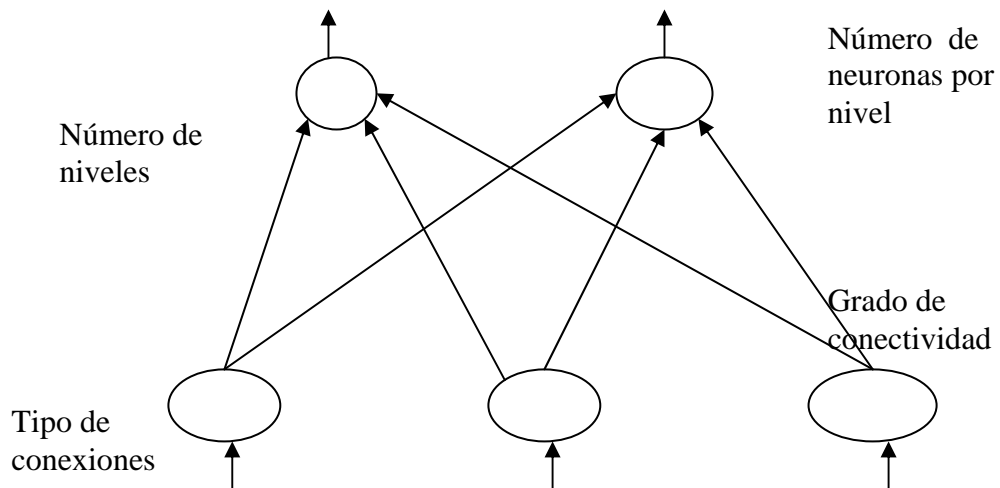
Según el autor Hilera (2000:66), los componentes de una red neuronal son:

- Unidades de procesamiento (la neurona)
- Estado de activación de cada neurona
- Patrón de conectividad entre las neuronas
- Regla de propagación
- Función de transferencia
- Regla de activación
- Regla de aprendizaje

Y por cada nodo (véase figura 20):

- Número de niveles o capas
- Número de neuronas por nivel
- Patrones de conexión
- Flujo de información

Figura 20. Esquema de la estructura de una RNA



Fuente: Hilera, Martínez [2000:66]

Niveles o capas de neuronas:

- La distribución de neuronas dentro de la red forman niveles o capas, dependiendo su ubicación dentro de la red, se distinguen tres tipos de capas (Figura 21):

Nivel de entrada: recibe las señales de la entrada de la red, algunos autores no consideran el vector de entrada como una capa pues allí no se lleva a cabo ningún proceso.

Niveles ocultos: estas capas son aquéllas que no tienen contacto con el medio exterior, sus elementos pueden tener diferentes conexiones y son éstas las que determinan las diferentes topologías de la red

Nivel de salida: recibe la información de la capa oculta y transmite la respuesta al medio externo.

La figura 20 muestra el esquema de una red multicapa. La conectividad entre los nodos de una red neuronal está relacionada con la forma en que las salidas de las neuronas están canalizadas para convertirse en entradas de otras neuronas. La señal de salida de un nodo puede ser una entrada de otro, o incluso ser una entrada de sí mismo (*autorrecurrente*). Cuando una salida de neuronas es entrada de neuronas del mismo nivel o de niveles precedentes, la red se describe

como de *propagación hacia delante* (figura 21). Cuando las salidas se conectan como entradas de neuronas a niveles previos, o del mismo nivel incluyéndose entre ellos mismos, la red es de *propagación hacia atrás* (figura 22). Las redes de propagación hacia atrás que tienen los lazos cerrados son sistemas recurrentes.

Figura 21. Estructura de una red multicapa con conexiones hacia delante.

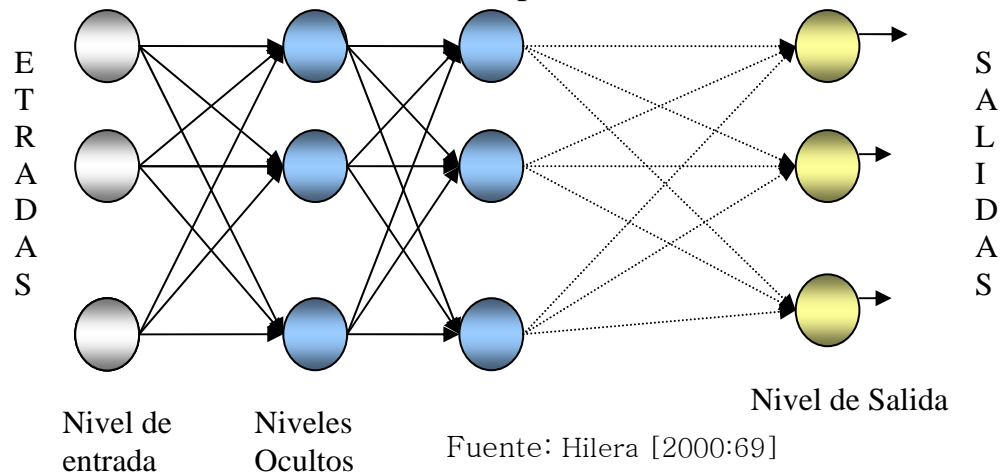
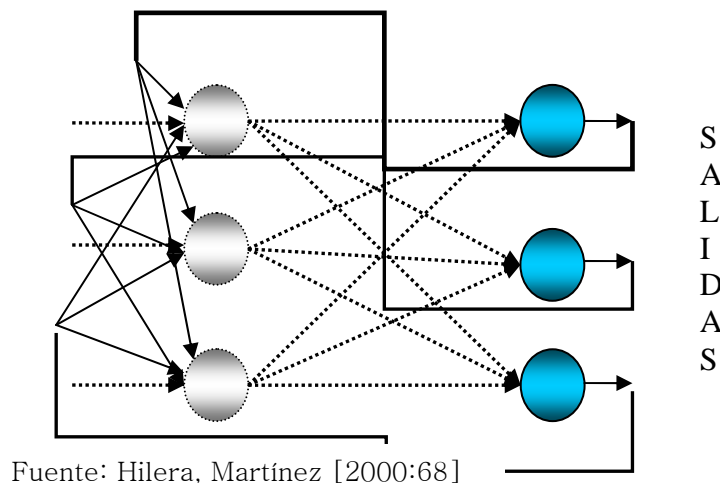


Figura 22. Ejemplos de conexiones con propagación hacia atrás



Nodo con propagación hacia atrás sobre si mismo



3.9 Características de los modelos de redes neuronales artificiales más importantes y algunas de sus aplicaciones.

Existen muchos tipos de redes neuronales, según (Hilera, 2000:30-33), éstos se muestran en la Tabla 1.

Tabla 1. Características de los modelos de redes neuronales más importantes

Nombre de la red	Año	Aplicaciones más importantes	Comentarios	Limitaciones	Inventada/ desarrollada por
Avalancha	1967	Reconocimiento de habla continua. Control brazos robot	Ninguna red sencilla puede hacer todo esto	No es fácil alterar la velocidad o interpolar el movimiento	Stephen Grossberg
Teoría resonancia adaptativa (ART)	1986	Reconocimiento de patrones (radar, sonar, etc.)	Sofisticada poco utilizada	Sensible a la translación, distorsión y escala.	Gail cartenter, Stephen Grossberg.
ADALINE/MADALINE	1960	Filtrado de señales, ecualizados adaptativo. Modems.	Rápida, fácil de implementar con circuitos analógicos o VLSI.	Sólo es posible clasificar espacios linealmente separados	Bernand Widrow.
Back propagation	1974-85	Síntesis de voz desde texto. Control de robots. Predicción. Reconocimiento de patrones	Red más popular. Numerosas aplicaciones con éxito. Facilidad de aprendizaje. Potente.	Necesita mucho tiempo para el aprendizaje y muchos ejemplos.	Paul Webrbos, david Parker, David Rumelhart.
Memoria asociativa bidireccional	1985	Memoria heteroasociativa de acceso por contenido	Aprendizaje y arquitectura simples.	Baja capacidad de almacenamiento los datos deben ser codificados.	Bart Kosko
Máquinas de Boltzmann y Cauchy	1985-86	Reconocimiento de patrones (imágenes sonar y radar). Optimización.	Redes simples. Capacidad de representación óptima de patrones.	La máquina de Boltzman necesita un tiempo muy largo de aprendizaje.	Jeffrey Hinton, Ferry Sejnowski, Harold Szu.
Brain-Estate-in-a-Box	1977	Extracción de conocimiento de bases de datos.	Posiblemente mejor realización que las redes de	Realización y potenciales aplicaciones no estudiadas	James Anderson

Nombre de la red	Año	Aplicaciones más importantes	Comentarios	Limitaciones	Inventada/ desarrollada por
			Hopfield	totalmente.	
Cerebellatron	1969	Control de movimiento de los brazos de un robot	Semejante a avalancha	Requiere complicadas entradas de control	David Marr, Jamnes Albus, Andrés Pellionez
Counter-propagation	1986	Compresión de imágenes	Combinación de perceptron TPM	Numerosas neuronas y conexiones	Robert Hecht Nielsen
Hopfield	1982	Reconstrucción de patrones y optimización	Puede implementarse en VLSL. Fácil de conceptuar.	Capacidad y estabilidad.	John Hopfield
Neocognitron	1978-84	Reconocimiento de caracteres manuscritos	Insensible a la translación, rotación y escala	Requiere muchos elementos de proceso, niveles y conexiones	K. Fukushima
Perceptron	1957	Reconocimiento de caracteres impresos	La red más antigua. Construida en HW.	No puede reconocer caracteres complejos.	Frank Rosenblatt
Self-Organizing-Map (SOM). Topology-Preserving-Map (TPM).	1980-84	Reconocimiento de patrones, codificación de datos, optimización.	Realiza mapas de características comunes de los datos aprendidos.	Requiere mucho entrenamiento.	Teuvo Kohonen

Fuente: Hilera, Martínez [2000:30-33], (Adaptada de [Hecht-Nielsen 88a])

La Tabla 1 muestra las redes neuronales más importantes que han existido desde 1957. La investigación realizada en este trabajo nos permite identificar la importancia del Perceptron y la red Back-Propagation. Para efectos de este protocolo, se considerarán estos dos modelos que incluyen el grupo de redes neuronales con conexiones hacia delante ya que por sus características son buenos *clasificadores de patrones* y utilizan *aprendizaje supervisado*, que es precisamente el tipo de modelo de red que requerimos en esta investigación, según Isasi y Viñuela [2004:27], “el modelo del Perceptron se concibió como un sistema capaz de realizar tareas de clasificación de forma automática”. La idea era disponer de un sistema que, a partir de un conjunto de ejemplos de clases diferentes, fuera capaz de determinar las ecuaciones de las superficies que hacían de frontera de dichas clases. A esto se le llama patrones o ejemplos de entrenamiento, indistintamente. Son dichos patrones de entrenamiento

los que aportaban la información necesaria para que el sistema construyera las superficies discriminantes, y además actuara como un discriminador para ejemplos nuevos desconocidos. El sistema al final del proceso, era capaz de determinar, para cualquier ejemplo nuevo, a qué clase pertenecía.

Tomando en cuenta lo anterior podemos afirmar que el modelo del Perceptron y la red Back-Propagation nos permitirán determinar a partir de un conjunto de ejemplos formados por las características académicas de los tutores como clases diferentes, las ecuaciones de las superficies que hagan frontera de dichas clases y con estos patrones de entrenamiento se aportará la información necesaria para que el sistema actúe como un discriminador para nuevos tutores candidatos, ofreciendo una óptima selección de los mismos.

Para entender con claridad el uso del modelo del Perceptron como un modelo capaz de realizar tareas de clasificación, se incluyen algunos de los conceptos teóricos del mismo.

3.10 Redes neuronales con conexiones hacia adelante

Este grupo incluye el Perceptron, las redes ADALINE y MADALINE y la red Back-Propagation. Los autores, Isasi, Viñuela [2004:27], describen de una manera práctica y muy clara los conceptos de estos modelos y es por eso que retomo los conceptos de su libro de “*Redes de Neuronas Artificiales*”.

3.10.1 EL PERCEPTRON

El modelo PERCEPTRON data de los años 50. Este modelo es un buen *clasificador de patrones* y utiliza *aprendizaje supervisado*.

Este fue el primer modelo de red neuronal artificial desarrollado por Rosenblatt en 1958 [Rosenblatt 58]. Despertó un enorme interés en los años 60, debido a su capacidad para aprender a reconocer patrones sencillos: un Perceptron, formado por varias neuronas lineales para recibir las entradas a la red y una neurona de salida, es capaz de decidir cuándo una entrada presentada a la red pertenecen a una de las dos clases que es capaz de reconocer.

Isasi y Galván [2004:27], afirman que, “este modelo se concibió como un sistema capaz de realizar tareas de clasificación de forma automática”. Es como un sistema que, a partir de un conjunto de ejemplos de clases diferentes, es capaz de determinar las ecuaciones de las superficies que hacen la frontera de dichas clases. La

información sobre la que se basa el sistema está constituida por los ejemplos existentes de las diferentes clases. A esto se les llama patrones o ejemplos de entrenamiento, éstos patrones de entrenamiento aportan la información necesaria para que el sistema construya las superficies discriminantes, y además actúa como un discriminador para ejemplos nuevos desconocidos. El sistema al final del proceso, es capaz de determinar, para cualquier ejemplo nuevo, a qué clase pertenece.

Es innegable que existe una estrecha relación entre los campos de la estadística y las redes neuronales artificiales, Pérez (2003:71).

Al relacionar ambos campos debemos considerar dos perspectivas. Por una parte, el campo de las RNA constituye una nueva herramienta para realizar cálculos de tipo estadístico. Por otra parte, numerosas de las técnicas y cálculos subyacentes al aprendizaje y operación de diversos modelos de RNA siguen métodos estadísticos bien conocidos, Pérez (2003:71).

La diferencia estriba en que mientras la estadística se dedica al análisis de datos, en las redes neuronales la inferencia significa aprender a generalizar a partir de datos con ruido. Las redes neuronales se suelen definir en términos de sus algoritmos o implementación, mientras que los métodos estadísticos se suelen definir en función de sus resultados.

Algunas RNA incluyen elementos probabilísticos, como la red de Hopfield, y algunas otras permiten implementar técnicas como el análisis cluster, el análisis discriminante y modelos de regresión.

Las RNA se vuelven muy útiles cuando se quieren resolver problemas para los cuales no se conocen algoritmos o expresiones que permitan alcanzar una solución o para los que se pueden definir algoritmos de solución, pero éstos consumen muchos recursos (tiempo de cálculo o datos). La aplicación de las RNA no precisa conocer ninguna expresión matemática que defina las relaciones entre los datos, ni ningún algoritmo de solución del problema, ya que es la misma red la que determina la forma de la solución del problema. Por ejemplo, si se dispone de una muestra de valores que siguen alguna distribución de probabilidad desconocida, se puede aplicar una RNA para intentar determinar los parámetros de dicha distribución, o para clasificar los valores de la muestra, Pérez (2003:73).

La autora, Pérez (2003:73) concluye; que desde el punto de vista estadístico, muchos de los problemas que se intentan resolver

con RNA pertenecen a la categoría de los problemas mal condicionados. En términos estadísticos, las RNA son estimadores no paramétricos que realizan estimaciones denominadas de modelo libre (no se impone ninguna forma funcional de partida).

Los términos y expresiones equivalentes sobre RNA y sobre estadística están contenidos en la tabla 2.

Tabla 2. Términos y expresiones equivalentes en ambos campos.

ESTADÍSTICA	RNA
Variables	Características o rasgos
Variables independientes, Predictores, regresores Variables explicativas	Entradas
Variables predichas	Salidas
Variables dependientes, respuestas o Valores observados	Valores de entrenamiento o valores objetivo
Residuos	Errores
Muestra	Conjunto de entrenamiento
Muestra de validación	Conjunto de prueba, conjunto de validación
Observación, caso	Patrón, par de entrenamiento, vector, ejemplo, caso.
Término de error	Ruido
Binario, dicotómico	binario (0/1), bivalente o bipolar (-1,1)
Modelo	Arquitectura
Estimación, ajuste del modelo, optimización	Entrenamiento, aprendizaje, adaptación o autoorganización
Criterio de estimación	Función de error, función de coste, función de Lyapunov
Estimaciones de parámetros	Pesos sinápticos
Interacciones	Neuronas de orden superior
Transformaciones	Enlaces funcionales
Regresión	Mapeado, aproximación de función
Interpolación, extrapolación o predicción	Generalización
Función de enlace inversa en	Función de activación, función de

ESTADÍSTICA	RNA
GLIM	transferencia
Regresión lineal, análisis discriminante, recocido simulado, búsqueda aleatoria	Métodos estadísticos
Función logística múltiple	Softmax
Términos de interacción o transformaciones	Enlaces funcionales
Efectos principales	conexiones directas de entrada y salida

Fuente: Pérez (2003:76).

Muchos modelos RNA son similares (en ocasiones iguales) a técnicas estadísticas, tales como modelos generalizados, regresión polinómica, regresión no paramétrica, análisis discriminante, regresión-proyección, componentes principales y análisis cluster; donde su énfasis está especialmente en la predicción de fenómenos complejos en vez de su explicación.

Lo que nos ocupa en esta investigación es la clasificación de tutores, la cual se relaciona con un análisis discriminante con RNA, donde su importancia radica en que a partir de ésta clasificación, se logre el pronóstico de buenos tutores que cumplan con el Sistema Tutorial Generacional.

En la tabla 3 se observan diversas técnicas tanto de RNA como de estadística.

Tabla 3. Equivalencias existentes entre diversas técnicas estadísticas y las RNA.

TÉCNICA RNA	TÉCNICA ESTADÍSTICA
ADALINE	Análisis discriminante lineal de 2 grupos
Aprendizaje competitivo o cuantificación vectorial adaptativa	Análisis cluster
aprendizaje Hebbiano	Análisis de componentes principales
Técnica Cottrell/Munro/Zipser	
Aprendizaje no supervisado. Codificación y autoorganización	Componentes principales, análisis cluster, reducción de datos o dimensiones (las variables dependientes e independientes son las mismas)
Aprendizaje supervisado o heteroasociación	Regresión y análisis discriminante (las variables dependientes e independientes son diferentes).

TÉCNICA RNA	TÉCNICA ESTADÍSTICA
ART1	Clasificación iterativa.
ART2	Algoritmo guía de Hadtigan
AVQ (red de Kohonen para cuantificación vectorial adaptativa)	Algoritmo iterativo para análisis cluster de k-medias.
Clasificación	Análisis discriminante.
Codificación topológicamente distribuida	Modelo aditivo (generalizado).
Contrapropagación	Regresograma basado en clusters k-medias
Entrenamiento mediante la minimización de la mediana de los cuadrados de los errores	LMS (Least Median Squares). Error cuadrático medio.
LMS (Least Median Squares).	OLS (Ordinary Least Squares). Error cuadrático ordinario.
LVQ	Una forma de análisis discriminante lineal utilizando un análisis cluster preliminar.
Perceptron simple	Modelo lineal generalizado (GLIM-Generalized Lineal Model).
Perceptron lineal con una capa oculta	Análisis de redundancia máxima, Componentes principales de variables instrumentales.
Perceptron lineal con una capa oculta	Proyección-regresión.
Predicción	Forecasting.
Propagación hacia adelante	Predicción.
Redes Feedforward sin capas ocultas	Modelos lineales generalizados
Redes Feedforward con una capa oculta	Proyección-regresión
Red neuronal probabilística	Análisis Kernel discriminante
Red neuronal de regresión general	Regresión Kernel.
Red de segundo orden	Regresión cuadrática, modelo de superficie de respuesta.
Red de orden superior	Regresión polinómica, modelo lineal con términos de interacción.
Adaptación de peso, regulación	estimación de la valoración, regresión Ridge

Fuente: Pérez (2003:78).

Las redes con propagación hacia adelante (o *redes feedforward*), forman parte de los modelos de regresión no lineal y discriminación, los cuales forman parte de la arquitectura de red propuesta en este trabajo de investigación para resolver el problema en cuestión.

3.10.2 Perceptron simple

El autor Pérez (2003:78) afirma que, “un Perceptron calcula su entrada neta como suma ponderada de las entradas, y le aplica una

función de activación de tipo escalón, que mapea cualquier entrada real sobre un rango generalmente limitado ($\{0,1\}$ ó $\{-1,1\}$, habitualmente). Los perceptrones se suelen entrenar con mínimos cuadrados (intentando minimizar el error entre la salida obtenida y la esperada). La regla de aprendizaje utilizada para actualizar los pesos de las conexiones realiza un descenso por el gradiente de la función de error”,

Un perceptron puede tener una o varias salidas, cada una con un sesgo y un conjunto de pesos asociados. En el caso de múltiples salidas, se suele aplicar la misma función de activación para todas ellas. Un perceptron con una función de activación del tipo:

- **Lineal**, es como un modelo de *regresión lineal*, posiblemente múltiple o multivariante.
- **Umbral**, es una *función discriminante lineal*. Si hay varias salidas, el Perceptron umbral, es una *función discriminante múltiple*.
- **Logística**, es un modelo de *regresión logística*. En vez de la función umbral, suele ser más útil una función logística múltiple para estimar las probabilidades condicionales de cada clase (en la jerga de RNA se denomina función de activación softmax).

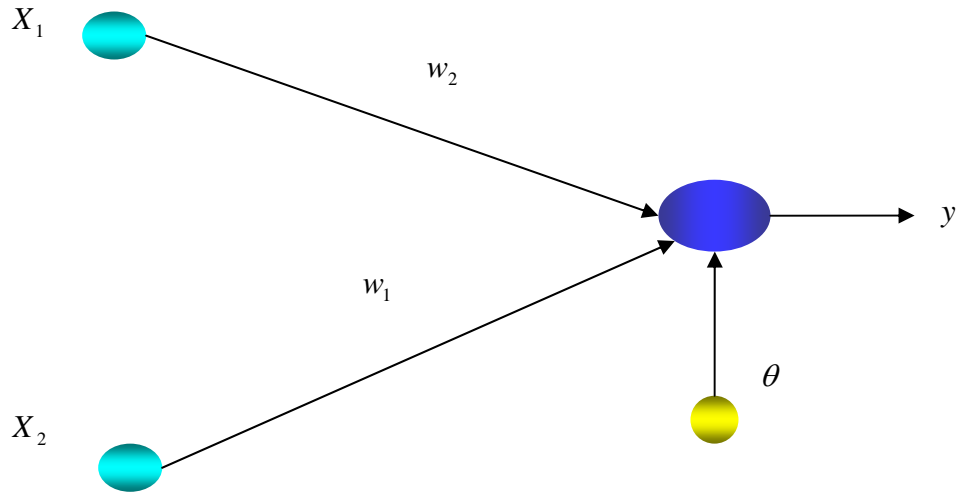
Para efectos del trabajo de investigación, se usará un perceptron con una función de activación del tipo umbral y dado que contamos con dos salidas estamos hablando de una *función discriminante múltiple*.

Los perceptrones se suelen entrenar mediante mínimos cuadrados. También se usa la máxima verosimilitud para proporciones binomiales cuando los valores esperados están entre 0 y 1.

3.10.3 Descripción del PERCEPTRON.

La arquitectura del Perceptron es sencilla, según (Isasi y Galván [2004:27]), es una estructura monocapa, con un conjunto de células de entrada, tantas como se requiera; y una o varias células de salida. Cada una de las células de entrada tiene conexiones con todas las células de salida, y son éstas las conexiones que determinan las superficies de discriminación del sistema. (Véase la figura 23).

Figura 23. Arquitectura del PERCEPTRON con dos entradas y una salida



Fuente: Isasi, Galván [2004:27]

En la Figura 23, las entradas son X_1 y X_2 , y la salida, y . Los pesos son w_1 y w_2 . Además, existe un parámetro adicional llamado umbral y denotado por θ . El umbral se utiliza como factor de comparación para producir la salida, y habrá tantos como células de salida existan en la red, uno por cada una.

La salida de la red se obtiene con el cálculo de la activación de la célula de salida mediante la suma ponderada por los pesos de todas las entradas, como se muestra en la ecuación 6:

$$y' = \sum_{i=1}^n w_i x_i \text{-----Ecuación 6}$$

La salida definitiva se produce al aplicarle una función de salida al nivel de activación de la célula. En un PERCEPTRON la función de salida es una función escalón que depende del umbral, la cual se muestra con la ecuación 7:

$$y = F(y', \theta) \text{-----Ecuación 7}$$

La función escalón está dada por: $F(s, \theta) = \begin{cases} 1 & \text{Si } s > \theta \\ -1 & \text{En caso contrario} \end{cases}$

Si sustituimos el valor de y' en la ecuación 7 y la función escalón, la salida se puede escribir en una sola ecuación 8; donde se obtiene la salida y .

$$y = F\left(\sum_{i=1}^n w_i x_i + \theta\right) \text{.....Ecuación 8}$$

Donde F ya no depende de ningún parámetro:

$$F(s) = \begin{cases} 1 & \text{si } s > 0 \\ -1 & \text{en caso contrario} \end{cases}$$

Esta ecuación equivale a introducir artificialmente en la salida un nuevo peso θ que no está conectado a ninguna entrada, sino a una ficticia con un valor constante de -1 .

La función de salida F es binaria y de gran utilidad en este modelo ya que al tratarse de un discriminante de clases, una salida binaria puede ser fácilmente traducible a una clasificación en dos categorías de la siguiente forma. (Véase Figura 24)

Si la red produce salida 1, la entrada pertenece a la categoría A.

Si la red produce salida -1 , la entrada pertenece a la categoría B.

En el caso de dos dimensiones, la ecuación 8 se transforma en:

$$w_1 x_1 + w_2 x_2 + \theta = 0 \text{ -----Ecuación 9}$$

Que es la ecuación de una recta de pendiente $-\frac{w_1}{w_2}$ y la ordenada pasa

por $-\frac{\theta}{w_1}$ (Gráficamente podría representarse como se muestra en la figura 24). En este caso, con dos células en la entrada, los patrones de entrenamiento pueden representarse como puntos en un espacio bidimensional. Si además dichos puntos pertenecen a una de dos categorías, A o B, la separación de dichas categorías podrá hacerse mediante la recta anterior.

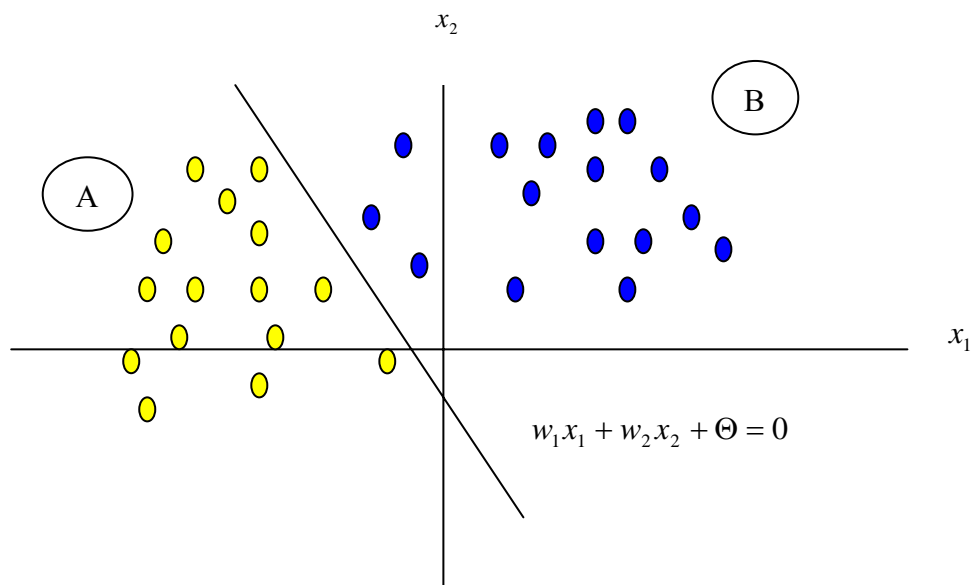
Es decir, la red define una recta, que en el caso de ser solución al problema discriminará entre las clases existentes en los datos de entrenamiento.

Si las clases están separadas, puede demostrarse que dicha recta discriminante existe; el problema es como determinar la ecuación de tal recta, a partir de los datos de entrenamiento. Esto es lo que hace precisamente el PERCEPTRON en su proceso de aprendizaje. El problema puede complicarse si en vez de dos percepciones hay muchas más. En este caso no habrá que determinar rectas, son hiperplanos. El proceso puede definirse en los siguientes términos:

Dado que:

Se dispone de un conjunto de ejemplos de entrenamiento, distribuidos en un espacio multidimensional, y de los que se conoce a qué categoría pertenecen. Se va a describir el caso en el que hay únicamente dos clases, A y B. Más adelante se comentarán las modificaciones del modelo para extenderlo a un caso genérico de n clases.

Figura 24. Separación de dos clases mediante un PERCEPTRON



Fuente: Fuente: Isasi, Galván [2004:27]

Hay que determinar la ecuación del hiperplano que deja a un lado los ejemplos de un tipo y a otros los del otro. La ecuación del hiperplano se reduce a partir de los ejemplos.

Más formalmente sería:

Dados conjuntos de puntos en

$\mathfrak{R}^n : A = (\vec{a}_1, \dots, \vec{a}_{a_n})$ y $B = (\vec{b}_1, \dots, \vec{b}_{b_n})$, obtener el hiperplano:

$w_1x_1 + \dots + w_nx_n + \theta = 0$, de tal forma que: $\forall \vec{a} \in A : w_1a_1 + \dots + w_na_n + \theta > 0$ y

$\forall \vec{b} \in B : w_1b_1 + \dots + w_nb_n + \theta < 0$

Este proceso constituye el aprendizaje del PERCEPTRON y se realiza mediante un proceso iterativo en el que paulatinamente se van modificando los valores de los pesos de las conexiones, hasta encontrar los valores que determinan las ecuaciones discriminantes. En función de la red neuronal esto sería equivalente a encontrar los valores de las conexiones que haga:

$$\forall \vec{a} \in A : y(\vec{a}) = 1$$

y

$$\forall \vec{b} \in B : y(\vec{b}) = -1$$

El proceso de aprendizaje consiste en lo siguiente: se introduce un patrón de los del conjunto de aprendizaje, perteneciente, por ejemplo, a la clase A. Se obtiene la salida que genera la red para dicho patrón. Si la salida producida es 1, la respuesta de la red para dicho patrón es correcta, y no se realizará ninguna acción; se procederá simplemente a introducir un nuevo patrón. Si por el contrario la salida producida es -1, entonces la respuesta de la red es incorrecta; la red categoriza el patrón como de la clase B. Esto es un error de clasificación, y es en este caso cuando se produce el aprendizaje. El aprendizaje consiste en modificar los valores de las conexiones. En este caso dado que la salida producida es inferior a la que se debería haber obtenido, los pesos son incrementados para que en la próxima presentación del mismo patrón pueda superar el umbral y producir la salida deseada, 1. Si el patrón que se introduce es de la clase B, y también se produce un error de clasificación, el proceso se invierte; los pesos se decrementan por la misma razón.

Si se llama X a cada uno de los patrones de entrenamiento y $d(X)$ a su clase asociada, tomando valores en $(1, -1)$, el proceso de aprendizaje se puede describir de la siguiente forma:

Empezar con valores aleatorios para los pesos y el umbral.

Seleccionar un vector de entrada X del conjunto de ejemplos de entrenamiento.

Si $y \neq d(X)$, la red da una respuesta incorrecta. Modificar w_i de acuerdo con:

$$\Delta w_i = d(X)x_i \quad \text{-----} \quad 3$$

Si no se ha cumplido el criterio de finalización, volver a 2.

En el paso tres se aprecia que si la salida de la red para un patrón es $y(X) = 1$, pero su clase es $d(X) = -1$, entonces el incremento es

negativo, $\Delta w_i = d(X) = -x_i$, mientras que si ocurre lo contrario, es positivo, como se describió anteriormente.

Puesto que el umbral es equivalente a un peso adicional, al que se denota por w_0 , cuya entrada es siempre 1 ($x_0 = 1$), la ecuación anterior se puede extender para el umbral de la siguiente manera:

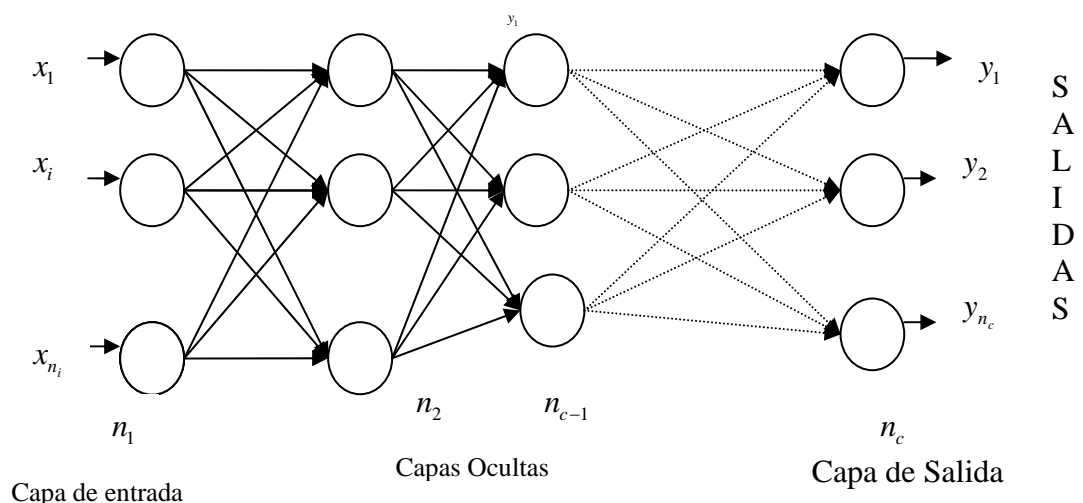
$$\Delta w_i = d(X)x_i, i = 0, \dots, n$$

3.10.4 Arquitectura del PERCEPTRON multicapa.

La arquitectura del PERCEPTRON multicapa se caracteriza por que tiene sus neuronas agrupadas en capas de diferentes niveles. Cada una de las capas esta formada por un conjunto de neuronas y se distinguen tres tipos de capas diferentes: la capa de entrada, las capas ocultas y la capa de salida, como se observa en la Figura 25.

Las neuronas de la capa de entrada no actúan como neuronas propiamente dichas, sino que se encargan únicamente de recibir las señales o patrones que proceden del exterior y propagan dichas señales a todas las neuronas de la siguiente capa. La última capa actúa como salida de la red, proporcionando al exterior la respuesta de la red para cada uno de los patrones de entrada. Las neuronas de las capas ocultas realizan un procesamiento no lineal de los patrones recibidos.

Figura 25. Arquitectura del PERCEPTRON multicapa



Fuente: Hilera [2000:69]

Como se observa en la figura 25, las conexiones del PERCEPTRON multicapa siempre están dirigidas hacia adelante, es decir, las neuronas de una capa se conectan con las neuronas de la siguiente capa, de ahí que reciban también el nombre de redes alimentadas hacia adelante o redes "feedforward". Las conexiones entre las neuronas lleva asociado un número real, llamado peso de la conexión. Todas las neuronas de la red llevan también asociado un umbral, que en el caso de PERCEPTRON multicapa suele tratarse como una conexión más a la neurona, cuya entrada es constante e igual a 1.

Generalmente, todas las neuronas de una capa están conectadas a todas las neuronas de la siguiente capa. De este modo, las neuronas de la capa de entrada están conectadas a todas las de la primera capa oculta; las neuronas de la primera capa oculta se conectan a las neuronas de la siguiente capa, etc. Se dice entonces que existe conectividad total o que la red está totalmente conectada.

Aunque en la mayor parte de los casos la arquitectura del PERCEPTRON multicapa está asociada al esquema de la figura 25, es posible también englobar dentro de este tipo de redes a arquitecturas con las siguientes características:

- Redes con conexiones de todas o ciertas neuronas de una determinada capa a neuronas de capas posteriores, aunque no inmediatamente posteriores.
- Redes en las que ciertas neuronas de ciertas capas no están conectadas a neuronas de la siguiente capa, es decir, el peso de la conexión es constante e igual a cero.

Cuando se aborda un problema con el PERCEPTRON multicapa, en la mayoría de los casos se parte de una arquitectura totalmente conectada, decir, todas las neuronas de una capa están conectadas a todas las neuronas de la siguiente capa. No es posible demostrar que si se solicitan arquitecturas en las que se eliminan conexiones o se añaden conexiones de una capa a capas no inmediatamente posteriores, se puedan obtener mejores resultados. Sin embargo, en ocasiones, y debido fundamentalmente a la naturaleza del problema, se pueden encontrar redes multicapa con estas características en sus conexiones.

3.10.5 Propagación de los patrones de entrada.

El PERCEPTRON multicapa define una relación entre las variables de salida de la red. Esta relación se obtiene propagando hacia adelante los valores de las variables de entrada. Para ello, cada neurona de la red procesa la información recibida por sus entradas y produce una respuesta o activación que se propaga, a través de las conexiones correspondientes, hacia las neuronas de la siguiente capa. A continuación se muestran las expresiones para calcular las activaciones de las neuronas de la red.

Sea un PERCEPTRON multicapa con C capas – $C - 2$ capas ocultas – y n_c neuronas en la capa c , para $c = 1, 2, \dots, C$. Sea $W^c = (w_{ij}^c)$ la matriz de pesos asociada a las conexiones de la capa c a la

capa $c+1$; y sea $U^c = (u_i^c)$ el vector de umbrales de las neuronas de la capa c para $c = 2, \dots, C$. se denota a_i^c a la activación de la neurona i de la capa c ; estas activaciones se calculan del siguiente modo:

- Activación de las neuronas de la capa de entrada (a_i^1). Las neuronas de la capa de entrada se encargan de transmitir hacia la red las señales recibidas del exterior. Por lo tanto:

$$a_i^1 = x_i \text{ para } i = 1, 2, \dots, n_1 \text{ -----(1)}$$

Donde $X = (x_1, x_2, \dots, x_{n_1})$ representa el vector o patrón de entrada a la red.

- Activación de las neuronas de la capa oculta (a_i^c). Las neuronas ocultas de la red procesan la información recibida aplicando la función de activación f a la suma de los productos de las activaciones que reciben por sus correspondientes pesos, es decir:

$$a_i^c = f \left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} a_j^{c-1} + u_i^c \right) \text{ Para } i = 1, 2, \dots, n_c \text{ y } c = 2, 3, \dots, C-1 \quad (2)$$

Donde a_j^{c-1} son las activaciones de las neuronas de la capa $c-1$.

- Activación de las neuronas de capa de salida (a_i^C). Al igual que en el caso anterior, la activación de estas neuronas viene dada por la función de activación f aplicada a la suma de los productos de las entradas que recibe por sus correspondientes pesos:

$$y_i = a_i^C = f \left(\sum_{j=1}^{n_{C-1}} w_{ji}^{C-1} a_j^{C-1} + u_i^C \right) \text{ Para } i = 1, 2, \dots, n_C \quad (3)$$

Donde $Y = (y_1, y_2, \dots, y_{n_C})$ es el vector de salida de la red.

La función f es llamada la *función de activación*. Para el PERCEPTRON multicapa, las funciones de activación más utilizadas son la *función sigmoideal* y la *función tangente hiperbólica*. Dichas funciones poseen como imagen un rango continuo de valores dentro de los intervalos $[0,1]$ y $[-1,1]$, respectivamente, y viene dadas por las siguientes expresiones:

- Función sigmoideal:

$$f_1(x) = \frac{1}{1 + e^{-x}} \quad (4)$$

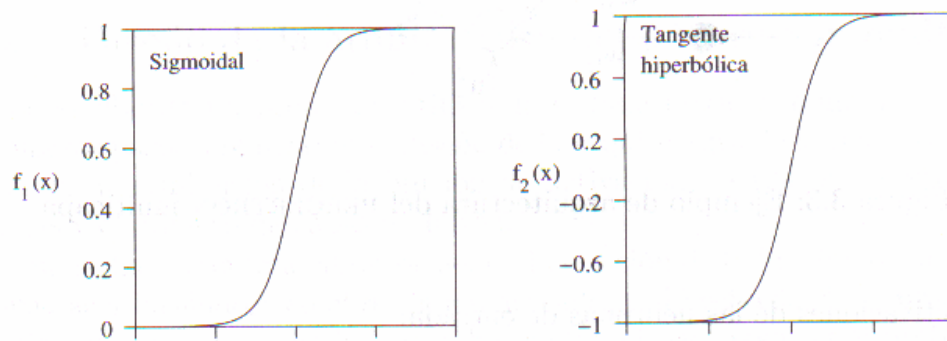
- Función tangente hiperbólica:

$$f_2(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (5)$$

Ambas son funciones crecientes con dos niveles de saturación: el máximo, que proporciona salida 1, y el mínimo, salida 0 para la

función sigmoideal y salida -1 para la tangente hiperbólica, como se observa en la Figura 26.

Figura 26. Funciones de activación de PERCEPTRON multicapa



Fuente: Isasi Galván [2004:49]

Generalmente la función de activación en el PERCEPTRON multicapa es común a todas las neuronas de red y es elegida por el diseñador, elección que se realiza únicamente basándose en los valores de activación que se desee que alcancen las neuronas. Ambas funciones están relacionadas mediante la expresión $f_2(x) = 2f_1(x) - 1$, por lo que la utilización de una u otra se elige únicamente en función del recorrido que interese.

En ocasiones y dependiendo de la naturaleza del problema, las neuronas de salida se distinguen del resto de las neuronas de la red, utilizando otro tipo de función de activación. En este caso, las más usadas son la función identidad y la función escalón.

De las ecuaciones (1), (2) y (3), se observa que el PERCEPTRON multicapa define, a través de sus conexiones y neuronas, una función continua no lineal del espacio R^{n_i} –espacio de los patrones de entrada– al espacio R^{n_c} –espacio de los patrones de salida–. Se puede escribir por tanto, que:

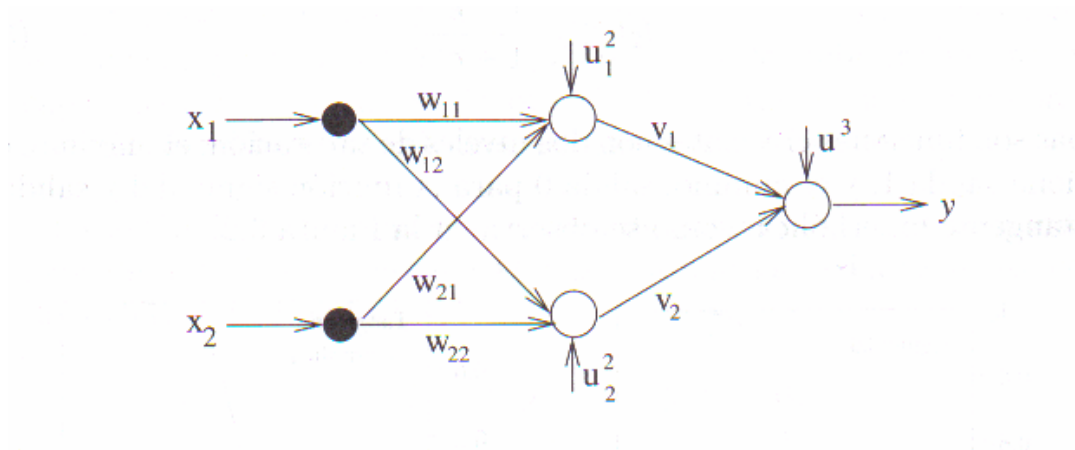
$$Y = F(X, W) \quad (6)$$

Donde Y es el vector formado por las salidas de la red, X es el vector de entrada a la red, W es el conjunto de todos los parámetros de la red –pesos y umbrales– y F es una función continua no lineal dada por las ecuaciones (1), (2) y (3).

3.10.6 Cálculo de las activaciones para un Perceptron multicapa con dos neuronas de entrada, dos ocultas y una salida.

Sea el perceptron multicapa que se muestra en la Figura 27, con una única capa oculta. Si x_1 y x_2 denotan las variables de entrada a la red, las activaciones de las neuronas de la red viene dadas por las siguientes expresiones:

Figura 27. Ejemplo de arquitectura del PERCEPTRON multicapa



Fuente: Isasi, Galván [2004:50]

Activaciones de las neuronas de entrada: $a_1^1 = x_1$ y $a_2^1 = x_2$

Activaciones de las neuronas ocultas:

$$a_1^2 = f(w_{11}a_1^1 + w_{21}a_2^1 + u_1^2) \text{ y } a_2^2 = f(w_{12}a_1^1 + w_{22}a_2^1 + u_2^2)$$

Activación de la neurona de salida:

$$y = a_1^3 = f(v_1a_1^2 + v_2a_2^2 + u^3)$$

1.1.1.1 Cuadro 1: Función XOR

1.1.1.2 Entrada	1.1.1.3 Salida
(0,0)	0
(0,1)	1
(1,0)	1
(1,1)	0

Cuadro 2: Activaciones del PERCEPTRON multicapa para la función XOR

Patrón de entrada	a_1^1	a_2^1	a_1^2	a_2^2	Salida de la red
(0,0)	0	0	0,129781	0,015875	0,128883
(0,1)	0	1	0,972618	0,204662	0,863310
(1,0)	1	0	0,964023	0,202889	0,858615
(1,1)	1	0	0,999843	0,802384	0,154993

Considérese los patrones que definen la función lógica XOR (cuadro 1).

Si los pesos y umbrales de la red que se muestran en la Figura 21, toman los valores:

$$w_{11} = 5,191129, w_{12} = 2,758669, w_{21} = 5,473012, w_{22} = 2,769596, v_1 = 5,839709,$$

$v_2 = -6,186834, u_1^2 = -1,90289, u_1^2 = -4,127002$ y $u^3 = -2,570539$, sustituyendo dichos parámetros en las expresiones anteriores y utilizando la función de activación sigmoideal (Ecuación 4), la salida de la red para los patrones de la función XOR toma los valores especificados en el cuadro 2.

1.1.2 3.10.7 Diseño de la arquitectura del PERCEPTRON multicapa.

Cuando se aborda un problema utilizado por el PERCEPTRON multicapa, uno de los primeros pasos a realizar es el diseño de la arquitectura de la red. Este diseño implica la determinación de la función de activación a emplear, el número de neuronas y el número de capas en la red. Como se ha comentado anteriormente, la elección de la función de activación se suele hacer basándose, en el recorrido deseado, y el hecho de elegir una u otra, generalmente no influye en la capacidad de la red para resolver el problema.

En lo que respecta al número de neuronas y capas, algunos de estos parámetros vienen dados por el problema y otros deben ser elegidos por el diseñador. Así, por ejemplo, tanto el número de neuronas en la capa de entrada, como el número de neuronas en la capa de salida, vienen dados por las variables que definen el problema. En algunas aplicaciones prácticas, no hay lugar a duda sobre el número de entradas y salidas. Sin embargo, existen problemas en los que el número de variables de entrada relevantes para el problema no se conocen con exactitud. En estos casos, se dispone de un gran número de variables, algunas de las cuales podrían no aportar información relevante a la red, y su utilización podrá complicar el aprendizaje, pues implicaría arquitecturas de gran tamaño y con alta conectividad. En estas situaciones es conveniente realizar

un análisis previo de las variables de entrada más relevantes al problema y descartar aquellas que no aportan información a la red.

El número de capas ocultas y el número de neuronas en estas capas deben ser elegidos por el diseñador. No existe un método o regla que determine el número óptimo de neuronas ocultas para resolver un problema dado. En la mayor parte de las aplicaciones prácticas, estos parámetros se determinan por prueba y error.

Partiendo de una arquitectura ya entrenada, se realizan cambios aumentando y disminuyendo el número de neuronas ocultas y el número de capas hasta conseguir una arquitectura adecuada para el problema a resolver, que pudiera no ser la óptima, pero que proporciona una solución.

Si bien el número de neuronas ocultas puede influir en el comportamiento de la red, es necesario indicar que en el caso del PERCEPTRON multicapa, generalmente, el número de neuronas ocultas no es parámetro significativo, pues dado un problema, puede existir una gran cantidad de arquitecturas capaces de resolver de manera adecuada dicho problema. Además, añadir o eliminar una neurona oculta no influye, de manera significativa, en la capacidad de la red.

En la actualidad existen líneas de investigación abiertas centradas en la determinación automática del número óptimo de neuronas ocultas de neuronas ocultas, así como de capas ocultas, para cada problema en particular. En el caso del PERCEPTRON multicapa, la mayor parte de estos trabajos se basan en la utilización de técnicas evolutivas, las cuales realizan una búsqueda en el espacio de las arquitecturas de redes guiada por la optimización del rendimiento de la red ([Millar et al., 1989], [Yao and Lin, 1997], [Gutiérrez et al., 2001])

3.11 Algoritmo de Retropropagación (*backpropagation*)

Viñuela, (Viñuela, 2004), describe la regla o algoritmo de aprendizaje como el mecanismo mediante el cual se van adaptando y modificando todos los parámetros de la red, en el caso de PERCEPTRON multicapa, se trata de un algoritmo de aprendizaje supervisado; es decir, la modificación de los parámetros se realiza para que la salida de la red sea lo más próxima posible a la salida proporcionada por el supervisor o salida deseada. Por tanto, para cada patrón de entrada a la red es necesario disponer de un patrón de salida deseada.

Puesto que el objetivo es que la salida de la red sea lo más próxima posible a la salida deseada, el aprendizaje de la red se formula con un problema de minimización del siguiente modo:

$$\text{Min}_w E \quad (7)$$

Siendo W el conjunto de parámetros de la red –pesos umbrales– y E una función error que evalúa la diferencia entre las salidas de la red y las salidas deseadas. En la mayor parte de los casos, la función error se define como:

$$E = \frac{1}{N} \sum_{n=1}^N e(n) \quad (8)$$

Donde N es el número de patrones o muestras y $e(n)$ es el error cometido por la red para el patrón n , dado por:

$$e(n) = \frac{1}{2} \sum_{i=1}^{n_c} (s_i(n) - y_i(n))^2 \quad (9)$$

Siendo $Y(n) = (y_1(n), \dots, y_{n_c}(n))$ y $S(n) = (s_1(n), \dots, s_{n_c}(n))$ los vectores de salida de la red y salidas deseadas para el patrón n , respectivamente.

De este modo si W^* es un mínimo de la función error E , en dicho punto el error es próximo a cero, lo cual indica que la salida de la red es próxima a la salida deseada, alcanzando así la meta de la regla de aprendizaje.

Por tanto, el aprendizaje del PERCEPTRON multicapa es equivalente a encontrar un mínimo de la función error. La presencia de funciones de activación no lineales hace que la respuesta de la red sea no lineal respecto a los parámetros ajustables, por lo que el problema de minimización es un problema no lineal, y, como consecuencia, tienen que utilizarse técnicas de optimización no lineales para su resolución. Dichas técnicas, están, generalmente, basadas en una adaptación de los parámetros siguiendo una cierta dirección de búsqueda. En el contexto de redes de neuronas, y en particular en el PERCEPTRON multicapa, la dirección de búsqueda más comúnmente usada es la dirección negativa del gradiente de la función E –método de descenso del gradiente–, pues conforme al cálculo de varias variables, ésta es la dirección en que la función decrece.

Aunque, estrictamente hablando, el aprendizaje de la red debe realizarse para minimizar el error total (Ecuación 8), el procedimiento

más utilizado esta basado en métodos de gradiente estocástico, los cuales consisten en una sucesiva minimización de los errores para cada patrón, $e(n)$, en lugar de minimizar el error total E . Por tanto, aplicando el método de descenso del gradiente estocástico, cada parámetro w de la red se modifica para cada patrón de entrada n de acuerdo con la siguiente ley de aprendizaje:

$$w(n) = w(n-1) - \alpha \frac{\partial e(n)}{\partial w} \quad (10)$$

Donde (n) es el error para el patrón n dado por la ecuación (9) y α es la razón o tasa de aprendizaje, parámetro que influye en la magnitud del desplazamiento en la superficie del error.

Debido a que las neuronas de la red están agrupadas en capas de distintos niveles, es posible aplicar el método del gradiente de forma eficiente, resultando el conocido algoritmo de RETROPROPAGACIÓN [Rumelhart, 1986] o regla delta generalizada. El término de retropropagación se utiliza debido a la forma de implementar el método del gradiente el PERCEPTRON multicapa, pues el error cometido en la salida de la red es propagado hacia atrás, transformándolo en un error para cada una de las neuronas ocultas de la red.

3.12 Obtención de la regla delta generalizada.

Viñuela (2004:54), desarrolla la regla delta generalizada para el aprendizaje del PERCEPTRON multicapa, en su libro “*Redes Neuronales Artificiales*”, para dicho desarrollo argumenta que es necesario distinguir dos casos: uno para los pesos de la capa oculta $C-1$ a la capa de salida y para los umbrales de las neuronas de salida, y otro para el resto de los pesos y umbrales de la red, pues las reglas de modificación de estos parámetros son diferentes.

Pesos de la capa oculta $C-1$ a la capa de salida y umbrales de la capa de salida.

Sea w_{ji}^{C-1} el peso de la conexión de la neurona j de la capa $C-1$ a la neurona i de la capa de salida. Utilizando el método de descenso del gradiente (Ecuación 10), dicho parámetro se modifica siguiendo la dirección negativa del gradiente del error:

$$w_{ji}^{C-1}(n) = w_{ji}^{C-1}(n-1) - \alpha \frac{\partial e(n)}{\partial w_{ji}^{C-1}} \quad (11)$$

Por tanto para la actualización de dicho parámetro es necesario evaluar la derivada del error $e(n)$ en dicho punto. De acuerdo con la expresión del error (Ecuación 9) y teniendo en cuenta, por un lado, que las salidas deseadas $s_i(n)$ para la red son constantes que no dependen del peso y, por otro lado, que el peso w_{ji}^{C-1} sólo afecta a la neurona de salida $i, y_i(n)$ (ver Ecuación 3), se obtiene que:

$$\frac{\partial e(n)}{\partial w_{ji}^{C-1}} = -(s_i(n) - y_i(n)) \frac{\partial y_i(n)}{\partial w_{ji}^{C-1}} \quad (12)$$

A este punto hay que calcular la derivada de la neurona de salida $y_i(n)$ respecto al peso w_{ji}^{C-1} . La salida de la red es la función de la activación f aplicada a la suma de todas las entradas por sus pesos, como se muestra en la ecuación (3). Aplicando la regla de la cadena para derivar la composición de dos funciones y teniendo en cuenta que, de todos los términos del sumatorio (Ecuación 3), el único en el que interviene el peso w_{ji}^{C-1} es $w_{ji}^{C-1} a_j^{C-1}$, y, por tanto el único cuya derivada es distinta de cero, se obtiene:

$$\frac{\partial y_i(n)}{\partial w_{ji}^{C-1}} = f' \left(\sum_{j=1}^{n_{C-1}} w_{ji}^{C-1} a_j^{C-1} + u_i^C \right) a_j^{C-1}(n) \quad (13)$$

Se define el término δ asociado a la neurona i de la capa de salida -capa C - y al patrón, $n, \delta_i^C(n)$ del siguiente modo:

$$n, \delta_i^C(n) = (s_i(n) - y_i(n)) f' \left(\sum_{j=1}^{n_{C-1}} w_{ji}^{C-1} a_j^{C-1} + u_i^C \right) \quad (14)$$

Reemplazando entonces en la ecuación (12) el valor de la derivada de la neurona de salida $y_i(n)$ dado por (13) y de acuerdo con el valor $\delta_i^C(n)$ se obtiene que:

$$\frac{\partial e(n)}{\partial w_{ji}^{C-1}} = \delta_i^C(n) a_j^{C-1}(n) \quad (15)$$

Finalmente reemplazando la derivada del error $e(n)$ respecto al peso w_{ji}^{C-1} obtenida en (15) en la Ecuación (11), se obtiene la ley para modificar dicho peso, la cual toma la siguiente expresión:

$$w_{ji}^{C-1}(n) = w_{ji}^{C-1}(n-1) + \alpha \delta_i^C(n) a_j^{C-1}(n) \quad (16)$$

Para $j = 1, 2, \dots, n_{c-1}$ $i = 1, 2, \dots, n_c$

En la Ecuación (16) obtenida anteriormente se observa que para modificar el peso de la conexión de la neurona j de la capa $C-1$ a la neurona i de la capa de salida, basta considerar la activación de la neurona de la que parte la conexión -neurona j de la capa $C-1$ - y el término δ de la neurona a la que llega la conexión -neurona salida i -, término que contiene el error cometido por la red para dicha neurona de salida (ver Ecuación 14).

La ley de aprendizaje obtenida para modificar los pesos de la última capa puede generalizarse para los umbrales de las neuronas de salida. Y en el PERCEPTRON multicapa el umbral de una neurona se trata como una conexión más a la neurona cuya entrada es constante e igual a 1. Siguiendo entonces, la ley anterior (Ecuación 16) se deduce que los umbrales de las neuronas de la capa de salida se modifican de acuerdo con la siguiente expresión:

$$u_i^C(n) = u_i^C(n-1) + \alpha \delta_i^C(n) \quad \text{Para } i = 1, 2, \dots, n_c \quad (17)$$

Pesos de la capa c a la capa $c+1$ y umbrales de las neuronas de la capa $c+1$ para $c = 1, 2, \dots, C-2$

Con el objetivo de que el desarrollo de la regla de aprendizaje para el resto de los pesos y umbrales de la red sea lo más claro posible, se elige un peso de la capa $C-2$ a la capa $C-1$. Sea w_{kj}^{C-2} el peso de la conexión de la neurona k de la capa $C-2$ a la neurona j de la capa $C-1$. Siguiendo el método del descenso del gradiente, la ley para actualizar dicho peso viene dada por:

$$w_{kj}^{C-2}(n) = w_{kj}^{C-2}(n-1) + \alpha \frac{\partial e(n)}{\partial w_{kj}^{C-2}} \quad (18)$$

En este caso y a diferencia del anterior -pesos hacia la capa de salida-, el peso w_{kj}^{C-2} influye en todas las salidas de la red, por que la derivada del error $e(n)$ (Ecuación 9) respecto de dicho peso viene dada por la suma de las derivadas para cada una de las salidas de la red, es decir:

$$\frac{\partial e(n)}{\partial w_{kj}^{C-2}} = - \sum_{i=1}^{n_N} (s_i(n) - y_i(n)) \frac{\partial y_i(n)}{\partial w_{kj}^{C-2}} \quad (19)$$

Para calcular a la derivada de la salida $y_i(n)$ respecto al peso w_{kj}^{C-2} es necesario tener en cuenta que este peso influye en la activación de la neurona j de la capa oculta $C-1$, a_j^{C-1} , respecto a dicho peso. Por tanto, y acuerdo con la ecuación (3), se tiene que:

$$\frac{\partial y_i(n)}{\partial w_{kj}^{C-2}} = f' \left(\sum_{j=1}^{n_{C-1}} w_{ji}^{C-1} a_j^{C-1} + u_i^C \right) w_{ji}^{C-1} \frac{\partial a_j^{C-1}}{\partial w_{kj}^{C-2}} \quad (20)$$

Sustituyendo este valor en la Ecuación (19) y de acuerdo con la definición de δ en el punto anterior (Ecuación 14), se obtiene:

$$\frac{\partial e(n)}{\partial w_{kj}^{C-2}} = \sum_{i=1}^{n_C} \delta_i^C(n) w_{ji}^{C-1} \frac{\partial a_j^{C-1}}{\partial w_{kj}^{C-2}} \quad (21)$$

Para obtener la ley de aprendizaje para el peso w_{kj}^{C-2} , sólo falta derivarla activación de la neurona j de la capa oculta $C-1$, a_j^{C-1} , respecto a dicho pesos. De nuevo aplicando la nueva regla de la cadena a la Ecuación (3.2), dicha derivada es:

$$\frac{\partial a_j^{C-1}}{\partial w_{kj}^{C-2}} = f' \left(\sum_{k=1}^{n_{C-2}} w_{kj}^{C-2} a_k^{C-2} + u_j^{C-1} \right) a_k^{C-2}(n) \quad (22)$$

Se define el valor δ para las neuronas de la capa $C-1$, $\delta_j^{C-1}(n)$, como:

$$\delta_j^{C-1}(n) = f' \left(\sum_{k=1}^{n_{C-2}} w_{kj}^{C-2} a_k^{C-2} + u_j^{C-1} \right) \sum_{i=1}^{n_C} \delta_i^C(n) w_{ji}^{C-1} \quad (23)$$

Sustituyendo (22) en la Ecuación (21) y de acuerdo con el valor $\delta_j^{C-1}(n)$ definido anteriormente, se obtiene:

$$\frac{\partial e(n)}{\partial w_{kj}^{C-2}} = \delta_j^{C-1}(n) a_k^{C-2}(n) \quad (24)$$

Y como consecuencia la ley de aprendizaje para modificar el peso w_{kj}^{C-2} viene dada por:

$$w_{kj}^{C-2}(n) = w_{kj}^{C-2}(n-1) + \alpha \delta_j^{C-1}(n) a_k^{C-2}(n) \quad (25)$$

Para $k = 1, 2, \dots, n_{C-2}$ y $j = 1, 2, \dots, n_{C-1}$

Al igual que la ley obtenida para modificar los pesos de la última capa (Ecuación 16), en este caso también se observa que para modificar el peso de la conexión de la neurona k de la capa $C-2$ a la neurona j de la capa de $C-1$, basta considerar la activación de la neurona de la que parte la conexión -neurona k de la capa $C-2$ - y el término δ de la neurona a la que llega la conexión -neurona j de la capa $C-1$ - (ver Ecuación 25).

La diferencia radical en la expresión del término δ . Para los pesos de la capa $C-2$ a $C-1$ dicho termino viene dado por la derivada de la función de activación y por la suma de los términos δ asociados a las neuronas de la siguiente capa -en este caso, las neuronas de salida- por los pesos correspondientes, como se indica en la Ecuación (23).

A este punto es posible generalizar fácilmente la ley dada por la Ecuación (25), para los pesos de la capa c a la capa $c+1$ ($c=1,2,\dots,C-2$). Para ello, basta tener en cuenta la activación de la que parte la conexión y el término δ asociados a las neuronas de la siguiente capa. De este modo:

$$w_{kj}^c(n) = w_{kj}^c(n-1) + \alpha \delta_j^{c+1}(n) a_k^c(n) \quad (26)$$

Para $k=1,2,\dots,n_c$, $j=1,2,\dots,n_{c+1}$ y $c=1,2,\dots,C-2$

Donde $a_k^c(n)$ es la activación de la neurona k de la capa c para el patrón n y $\delta_j^{c+1}(n)$ viene dado por la siguiente expresión:

$$\delta_j^{c+1}(n) = f' \left(\sum_{k=1}^{n_c} w_{kj}^c a_k^c + u_j^c \right) \sum_{i=1}^{n_{c+1}} \delta_i^{c+2}(n) w_{ji}^c \quad (27)$$

Es, posible, también, generalizar la ley de aprendizaje para el resto de los umbrales de la red; basta tratarlos como conexiones cuya entrada es constante e igual a 1. La ley para modificarlos viene dada por:

$$u_j^{c+1}(n) = u_j^{c+1}(n-1) + \alpha \delta_j^{c+1}(n) \quad (28)$$

Para $j=1,2,\dots,n_{c+1}$ y $c=1,2,\dots,C-2$

3.13 Derivada de la función de activación¹.

El cálculo de los valores δ (Ecuaciones 14 y 27) para cada neurona del PERCEPTRON multicapa requiere el cálculo de la derivada de la función de activación. El PERCEPTRON multicapa puede utilizar dos tipos de funciones de activación –función sigmoideal y función tangente hiperbólica, por lo que los valores δ dependen, en principio de la función de activación empleada. A continuación se van a obtener las derivadas para ambas funciones de activación, quedando así completado el cálculo de los valores δ .

- Derivado de la función sigmoideal.

Derivado de la expresión dada por la Ecuación (4), se obtiene que:

$$f_1'(x) = \frac{1}{(1+e^{-x})^2} (-e^{-x}) = \frac{1}{1+e^{-x}} \frac{e^{-x}}{1+e^{-x}} \quad (29)$$

Por tanto,

$$f_1'(x) = f_1(x)(1-f_1(x)) \quad (30)$$

Como consecuencia, cuando se utiliza la función sigmoideal los valores δ asociados a las funciones de salida (Ecuación 14) adoptan la siguiente forma:

$$\delta_i^c(n) = -(s_i(n) - y_i(n))y_i(n)(1-y_i(n)) \text{ Para } i=1,2,\dots,n_c \quad (31)$$

Y Los valores δ para el resto de las neuronas de la red (Ecuación 27), vienen dados por:

$$\delta_j^{c+1}(n) = a_j^c(n) \left(1 - a_j^c(n)\right) \sum_{i=1}^{n_{c+1}} \delta_i^{c+2}(n) w_{ji}^c \quad (32)$$

Para $j=1,2,\dots,n_{c+1}$ y $c=1,2,\dots,C-2$

- Derivada de la función tangente hiperbólica.

Teniendo en cuenta que $f_2(x) = 2f_1(x) - 1$, la derivada de la función $f_2(x)$ es:

¹ Isasi Viñuela, Galván León, *Redes de Neuronas Artificiales Un Enfoque Práctico*, PEARSON EDUCACIÓN, S.A., Madrid, 2004

$$f_2(x) = 2f_1(x)(1 - f_1(x)) \quad (33)$$

Cuando se utilice, por tanto, la función de activación tangente hiperbólica, los valores δ para las neuronas de la red adoptan las expresiones dadas por las Ecuaciones (31 y 32) multiplicadas por un factor de 2.

En la sección anterior se ha comentado que, generalmente, todas las neuronas del PERCEPTRON multicapa suelen utilizar la misma función de activación, salvo las neuronas de salida, que pueden tener como función de activación la función identidad, es decir, $f(x) = x$. En este caso, la derivada de la función de activación en la salida es 1 y, como consecuencia, los valores δ para las neuronas de salida adoptan la siguiente expresión:

$$\delta_i^c(n) = -(s_i(n) - y_i(n)) \text{ Para } i = 1, 2, \dots, n_c \quad (34)$$

Como se ha comentado en el apartado anterior, el objetivo del aprendizaje o entrenamiento del PERCEPTRON multicapa es ajustar los parámetros de la red de pesos y umbrales con el fin de las entradas presentadas produzcan salidas deseadas, es decir, con el fin de minimizar la función de error E (Ecuación 8).

En esta sección, se van a detallar los pasos que involucran el proceso completo de aprendizaje del PERCEPTRON multicapa.

Sea $\{(X(n), S(n)), n = 1, \dots, N\}$ el conjunto de muestras o patrones que representan el problema a resolver, donde $X(n) = (x_1(n), \dots, x_{n_1}(n))$ son los patrones de entrada a la red, $s(n) = (s_1(n), \dots, s_{n_c}(n))$ son las salidas deseadas para dichas entradas y N es el número de patrones disponibles. Generalmente, es frecuente encontrar los patrones de entrada y salida normalizados o escalados mediante una transformación lineal en los intervalos $[0, 1]$ o $[-1, 1]$ dependiendo de la función de activación empleada, sigmoidea o tangente hiperbólica, respectivamente. Es necesario señalar, sin embargo, que esta transformación de los patrones no es una condición necesaria para realizar el aprendizaje de la red, sino que los datos pueden presentarse a la red sin sufrir dicha normalización. Sólo será necesario tener en cuenta que, en el caso de que los patrones de salida se utilicen sin escalar, la función de activación de las neuronas de salida de la red debe ser la identidad, pues si se utilizan funciones de activación sigmoideas, las salidas de la red siempre estarán en los rangos de valores

$[0,1]$ o $[-1,1]$ y, por tanto nunca podrán aproximarse a la salida deseada.

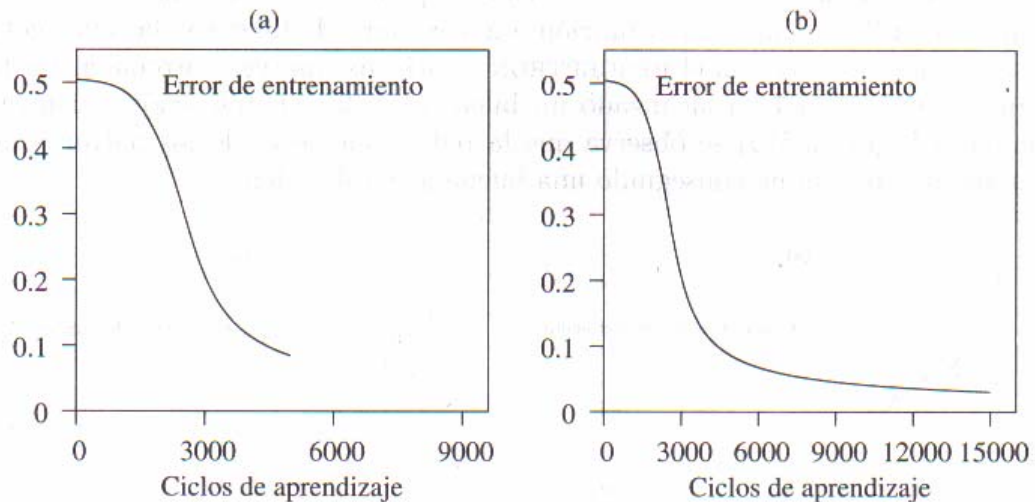
El proceso de aprendizaje del PERCEPTRON multicapa puede resumirse de la siguiente forma:

Partiendo de un punto aleatorio $W(0)$ del espacio R^{n_w} , donde n_w es el número de parámetros de la red –pesos más umbrales–, el proceso de aprendizaje desplaza el vector de parámetros $W(n-1)$ en el espacio R^{n_w} siguiendo la dirección negativa del gradiente del error en dicho punto, alcanzando así un nuevo punto en dicho espacio, $W(n)$, que estará mas próximo al mínimo de la función error que el anterior. El proceso continúa hasta que se encuentre un mínimo de la función error E , lo cual sucede cuando $\frac{\partial E}{\partial w} \approx 0$. En este momento y de acuerdo con los parámetros, éstos dejan de sufrir cambios significativos de una iteración a otra y el proceso de aprendizaje finaliza.

Por tanto y desde un punto de vista teórico, el proceso de aprendizaje del PERCEPTRON multicapa debe finalizar cuando $\frac{\partial E}{\partial w} \approx 0$, momento en que los parámetros de la red no cambian de una iteración a otra. Sin embargo desde el punto de vista práctico y a la hora de implementar el proceso de aprendizaje del PERCEPTRON multicapa, se suele fijar un número de ciclos de aprendizaje de modo que cuando se alcanza dicho número, se detiene el aprendizaje. En este punto se analiza si es necesario realizar más ciclos de aprendizaje o si son suficientes los establecidos a priori.

Este análisis se basa únicamente en observar si el error cometido por la red se mantiene prácticamente constante de una iteración a otra o si, por el contrario, el error sigue decreciendo. Así, por ejemplo, si se representa el error de entrenamiento cometido por la red en función del número de ciclos, en el caso de la situación presentada en la Figura 28(a), sería necesario realizar más ciclos de aprendizaje, mientras que en el caso de la Figura 28(b), es posible detener el proceso de aprendizaje de la red.

Figura 28. Evolución del error a lo largo del proceso de aprendizaje



Fuente: Isasi, Galván [2004:65]

Finalmente se debe señalar que el error de entrenamiento se estabiliza, precisamente, cuando las salidas de la red no cambian de una iteración a otra, lo cual viene provocado por que los parámetros de la red sufren cambios insignificantes de una iteración a otra. Como se comentó anteriormente, esto es debido a que se ha localizado un mínimo de la función error, es decir que $\frac{\partial E}{\partial w} \approx 0$.

El proceso de aprendizaje descrito anteriormente es el más comúnmente usado, aunque existe una, conocida con el nombre de proceso batch.

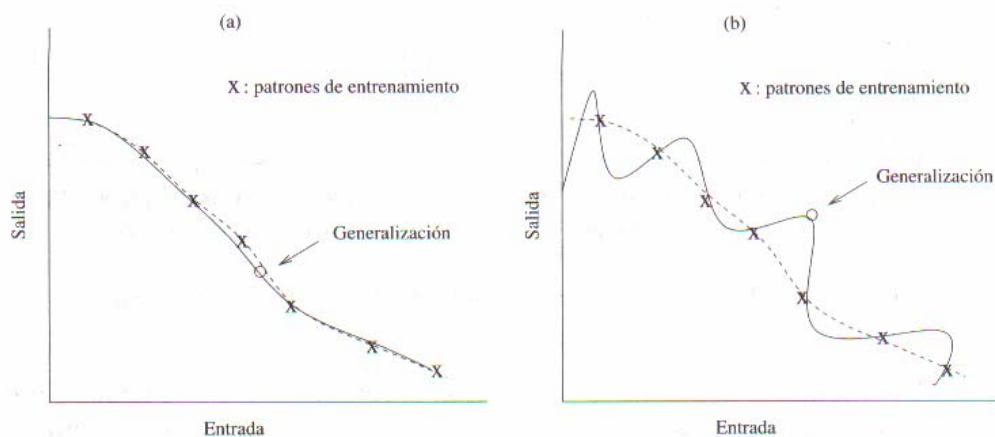
Básicamente este proceso se diferencia del anterior en que los parámetros de la red se modifican una vez que todos los patrones de entrenamiento han sido presentados a la red, y no para cada patrón de entrenamiento. Ambos mecanismos son equivalentes, aunque el más usado e implementado en los simuladores de redes de neuronas es el proceso de aprendizaje descrito anteriormente, también conocido con el nombre de proceso continuo.

3.14 Capacidad de generalización.

A la hora de evaluar el comportamiento de una red de neuronas, y en particular del PERCEPTRON multicapa, no sólo es importante saber si la red ha aprendido con éxito los patrones utilizados durante el aprendizaje, sino que es imprescindible, también conocer el comportamiento de la red ante patrones que no se han utilizado durante el entrenamiento. Es decir, de nada sirve disponer de una red que haya aprendido correctamente de los patrones de entrenamiento y que no responda adecuadamente ante patrones nuevos. Es necesario

que durante el proceso de aprendizaje la red extraiga las características de la muestra, para poder así correctamente a patrones diferentes. Esto se conoce como la capacidad de la red para generalizar las características presentes en el conjunto de muestras o capacidad de generalización de la red. La figura 29 muestra como la generalización podría ser llevada a cabo por una red. Los puntos etiquetados por cruces representan los patrones de entrenamiento, la línea punteada la función a aproximar y la línea sólida representa la salida que proporciona el PERCEPTRON multicapa una vez entrenada. En la Figura 29(a), la red ha alcanzado un buen nivel de generalización mientras que en la Figura 29 (b) se observa que la red ha memorizado los patrones de entrenamiento y no ha conseguido una buena generalización.

Figura 29. Generalización: (a) Buenas propiedades; (b) Escasas propiedades



Fuente: Isasi [2004:66]

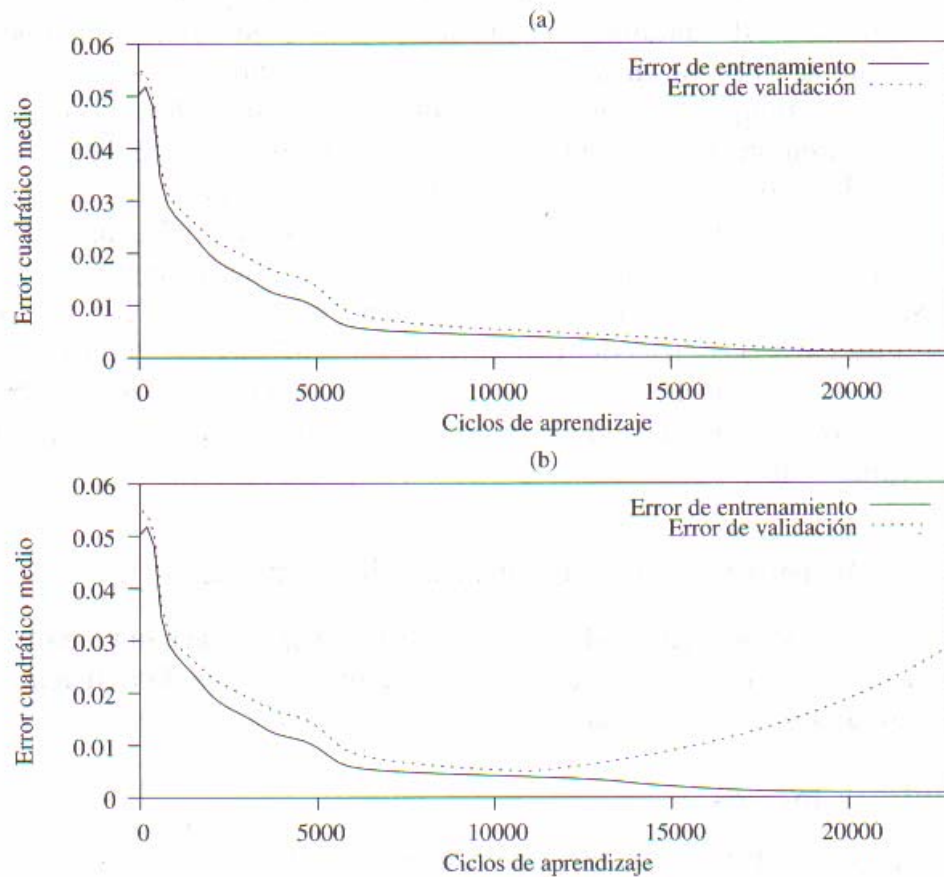
Por tanto cuando se realiza el proceso de aprendizaje de la red es muy importante, e incluso imprescindible, evaluar la capacidad de generalización. Para ello, es necesario disponer de dos conjuntos de muestras o patrones; uno para entrenar la red y modificar sus pesos y umbrales –*conjunto de entrenamiento*, y otro para medir la capacidad de la red para responder correctamente ante patrones que no han sido utilizados durante el entrenamiento – *conjunto de validación o prueba*-. Estos conjuntos se obtienen de las muestras disponibles sobre el problema y es conveniente que la separación sea aleatoria, con el fin tener conjuntos lo más representativos posibles, tanto de entrenamiento como de validación.

A veces, y dependiendo de las características de los conjuntos de entrenamiento riguroso podrían anular la capacidad de

generalización de la red. Por tanto, en ocasiones puede ser conveniente exigir un menor aprendizaje de la red sobre los patrones de entrenamiento, con el objetivo de obtener mejores propiedades de generalización. Con este propósito, necesario evaluar el error que comete la red sobre el conjunto de patrones de validación, no sólo cuando el proceso de aprendizaje a concluido, sino también durante el proceso de aprendizaje. Cada cierto número de ciclos, se debe presentar a la red el conjunto de patrones de validación y calcular el error cometido por la red sobre dicho conjunto.

Al igual que se analiza la evolución del error de entrenamiento a lo largo de los ciclos de aprendizaje, se debe analizar también la evolución del error de validación. Al observar la evolución de ambos errores, se pueden encontrar las siguientes situaciones: (Véase figuras 30(a) y 30 (b)).

Figura 30. Evolución de los errores de entrenamiento y validación a lo largo del proceso de aprendizaje.



Fuente: Isasi, Galván [2004:67]

- Ambos errores, de entrenamiento y validación, permanecen estables después de un cierto número de ciclos (Figura 30 (a)). En este caso, se puede decir que el aprendizaje ha acabado con éxito, pues la red ha sido capaz de extraer las características del problema, alcanzando un buen nivel de generalización.
- A partir de un cierto número de ciclos, el error de validación comienza a aumentar (Figura 30 (b)). En este caso, se puede decir, que el número de ciclos realizado (23,000) es adecuado para encontrar un mínimo del error de entrenamiento, pero a costa de perder propiedades de generalización de la red. Por tanto, no se puede afirmar que el aprendizaje haya acabado con éxito, sino que hubiera sido conveniente detener el proceso de aprendizaje en el momento en el que el error de validación comienza a crecer, para poder así disponer de una red con mejor capacidad de generalización. En estas situaciones se suele decir que se ha producido sobre aprendizaje en la red.

El sobre aprendizaje en un PERCEPTRON multicapa ocurre, por tanto, cuando la red ha aproximado correctamente los patrones de aprendizaje, pero no es capaz de responder adecuadamente ante los patrones de la validación, por ejemplo el caso que se observa en la figura 30 (b). Este hecho puede producirse como se ha comentado anteriormente, debido a un número elevado de ciclos de aprendizaje. Sin embargo, no es la única causa de sobre aprendizaje. En ocasiones, es producido por la utilización de demasiadas neuronas ocultas en la red. Un número excesivo de neuronas ocultas puede conducir a una escasa capacidad de generalización de la red. En estos casos, la red tiende a ajustar con mucha exactitud los patrones de entrenamiento. Particularmente, en problemas en los que las muestras poseen ruido, la utilización de muchas neuronas ocultas hacen que la red se ajuste al ruido de los patrones, impidiendo así la generalización.

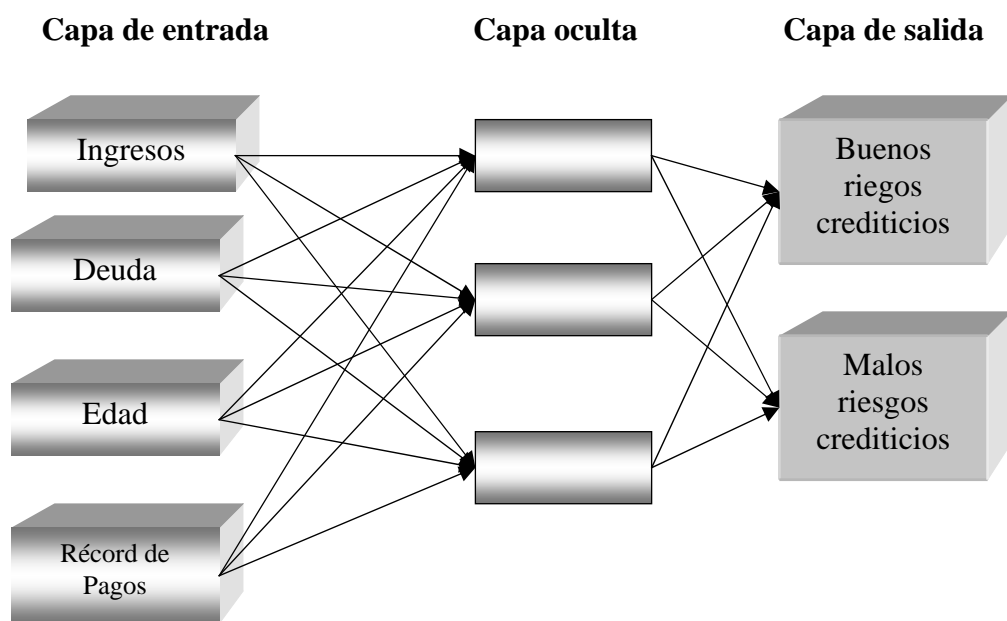
3.15 Diferentes aplicaciones de las redes neuronales artificiales

Los computadores digitales actuales superan al hombre en su capacidad de cálculo numérico y el manejo de símbolos relacionales. Sin embargo, el hombre puede solucionar problemas mucho más complejos de percepción (por ejemplo, reconocer a un amigo entre un tumulto desde un simple vistazo de su cara o al escuchar su voz, incluso por el modo de caminar; definir la habitabilidad de un aula a partir de sensaciones de temperatura, ruido, humedad, iluminación, etc.) a muy altas velocidades y sin necesidad de concebir un complejo modelo matemático o computacional. La respuesta está en la arquitectura del sistema neuronal biológico que es completamente diferente a la arquitectura del computador tradicional Von Neumann, como ya se mencionó. (Véase la tabla 1. Características y aplicaciones de los modelos de redes neuronales más importantes).

El autor, Laudon,[2004], hace referencia a que en la mayoría de las aplicaciones actuales, las redes neuronales tienen un mejor uso como auxiliares de los responsables de la toma de decisiones que como sustitutos de éstos. Ya que las RNA con su capacidad para deducir un significado de datos complicados o imprecisos, pueden ser utilizadas para detectar patrones o detectar una tendencia que es muy compleja de hallar con modelos determinísticos o por otras técnicas computacionales tradicionales. Una red entrenada puede ser vista como un experto en el manejo de la información que se le ha dado para analizar.

Las aplicaciones de las redes neuronales están surgiendo en la medicina, las ciencias y los negocios para resolver problemas de clasificación de patrones, análisis financieros y predictivos, control y optimización. La industria financiera está empezando a utilizar redes neuronales para descubrir patrones en vastas concentraciones de datos, lo cual podría ayudar a las empresas inversionistas a pronosticar el desempeño de capitales, clasificaciones de bonos corporativos o quiebras corporativas. Visa Internacional Inc. Está utilizando una red neuronal para detectar fraudes con tarjetas de crédito mediante el monitoreo del movimiento de todas las transacciones de Visa, en busca de cambios repentinos en los patrones de compra de los tarjetahabientes.

Figura 31. Ejemplo de una red neuronal para asignación de créditos.



Fuente: Herb Edelstein, Technology How-to: Mining data Warehouses”, *Information Week*, 8 de enero de 1996, derechos reservados © 1996, CMP Media, Inc., 600 Community Drive, Manhasset, Nueva York 11030. Reimpreso con permiso.

La figura² 31, muestra una red neuronal que utiliza reglas y que “*aprende*” de patrones de datos para construir una capa oculta de lógica. Luego la capa oculta procesa las entradas, clasificándolas con base en la experiencia del modelo.

En este momento se ha dado solución a problemas con altos niveles de incertidumbre que con métodos tradicionales jamás su habría obtenido. Soluciones tan novedosas e interesantes como la reconstrucción craneofacial para la identificación de hombres, música

² Laundon, Kenneth, y Laundon, Jane P, *Sistemas de información gerencial*, México, Pearson Educación, 2004

neurocomputacional, sistemas de detección de virus en computadores conocidos y desconocidos, identificación de usuarios en cajeros automáticos desde la imagen del iris de los ojos, reconocimiento de emisores en comunicaciones, diagnóstico de hepatitis, recuperación de telecomunicaciones ante fallas en el software, interpretación de palabras chinas, detección de minas submarinas, análisis de texturas, reconocimiento de objetos tridimensionales y reconocimiento de texto manuscrito.

El autor Gatley [1996] detalla una serie de aplicaciones que podrían tener las redes neuronales relacionadas a las ciencias administrativas y económicas.

Algunas de ellas son:

1. Fraude con tarjetas de crédito
2. Predicción de quiebras bancarios
3. Aplicaciones para las tarjetas de crédito
4. Precios de Arbitraje
5. Análisis del Mercado de Capitales
6. Análisis de la salud financiera de las empresas
7. Selección de prospectos de Venta
8. Proyecciones del Cash – flow
9. Predicción de cambios en la tendencia del mercado
10. Predicciones en los requerimientos de Personal

En Biología:

11. Aprender más acerca del cerebro y otros sistemas.
12. Obtención de modelos de la retina.

En Empresa:

13. Medio Ambiente
14. Analizar tendencias y patrones.
15. Previsión del tiempo.
16. Finanzas
17. Previsión de la evolución de los precios.
18. Valoración del riesgo de los créditos.
19. Identificación de falsificaciones.
20. Interpretación de firmas.

Manufacturación

21. Robots automatizados y sistemas de control (visión artificial y sensores de presión, temperatura, gas, etc.)
22. Control de producción en líneas de proceso.
23. Inspección de calidad.
24. Filtrado de señales.

En Medicina:

25. Analizadores del habla para la ayuda de audición de sordos profundos.
26. Diagnóstico y tratamiento a partir de síntomas y/o de datos analíticos (encefalograma, etc.).

En Monitorización en cirugía:

27. Predicción de reacciones adversas a los medicamentos.
28. Lectoras de Rayos X.
29. Entendimiento de causa de ataques epilépticos.

Militares:

30. Clasificación de las señales de radar.
31. Creación de armas inteligentes.
32. Optimización del uso de recursos escasos.

Cada uno de estos temas da pie para la realización de trabajos de investigación y desarrollo específicos en las RNA, lo que da una idea de la amplitud de este tema. Los autores, Hilera y Martínez [2000:33-34] nos muestran en la tabla 4, algunos ejemplos de aplicaciones de redes neuronales con su correspondiente producto comercial.

Tabla 4. Productos comerciales especializados de redes neuronales con su correspondiente aplicación

Producto	Vendedor	Aplicación
CAD/Chem Custom Formulation System	Al Ware	Diseño y formulación de productos químicos
Database Mining Workstation	HNC	Análisis y detección de relaciones entre elementos en BD (redit, scoring, valoración de propiedades, predicción de demanda, detección de fraude de tarjetas).
NeuroVision	Vision Harvest	Clasificación de objetos por forma, color y textura (inspección de productos industriales y agrícolas).

Producto	Vendedor	Aplicación
NT 5000 Neural Network Control System	California Scientific Software	Control de procesos de tiempo real.
NueX Smart Structures Tool	Charles River Analytics	Sistema híbrido RN/SE: Control de procesos/sistemas caóticos. Control adaptativo de procesos de refinado petroquímico.
Quality Plus	HNS	Inspección de calidad de productos basada en imagen (clasificación de manzanas y cítricos, reciclado de botellas, evaluación del estado de autopistas, etc.).
QuickStrokes	HNC	Reconocimiento de caracteres manuscritos.

Fuente: Hilera y Martínez [2000:33-34]

3.16 Ventajas de las Redes Neuronales

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

1. **Aprendizaje Adaptativo:** Es una de las características más atractivas de las redes neuronales, es la capacidad de aprender a realizar tareas basadas en un entrenamiento o una experiencia inicial.

En el proceso de aprendizaje, los enlaces ponderados de las neuronas se ajustan de manera que se obtengan unos resultados específicos. Una RNA no necesita un algoritmo para resolver un problema, ya que ella puede generar su propia distribución de los pesos de los enlaces mediante el aprendizaje. También existen redes que continúan aprendiendo a lo largo de su vida, después de completado e periodo inicial de entrenamiento.

La función del diseñador es únicamente la obtención de la arquitectura apropiada. No es problema del diseñador el cómo la red aprenderá a discriminar; sin embargo, si es necesario que desarrolle un buen algoritmo de aprendizaje que proporcione la capacidad de discriminar de la red mediante un entrenamiento con patrones.

2. **Autoorganización:** Las redes neuronales usan su capacidad de aprendizaje adaptativo para organizar la información que reciben durante el aprendizaje y/o la operación. Una RNA puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje. Esta autoorganización provoca la facultad de las redes neuronales de responder apropiadamente cuando se les presentan datos o situaciones a los que no habían sido expuestas anteriormente.
3. **Tolerancia a Fallos:** Comparados con los sistemas computacionales tradicionales, los cuales pierden su funcionalidad en cuanto sufren un pequeño error de memoria, en las redes neuronales, si se produce un fallo en un pequeño número de neuronas, aunque el comportamiento del sistema se ve influenciado, sin embargo no sufre una caída repentina. Hay dos aspectos distintos respecto a la tolerancia a fallos: primero, las redes pueden aprender a reconocer patrones con ruido, distorsionados, o incompletos. Segundo pueden seguir realizando su función (con cierta degradación) aunque se destruya parte de la red. La razón por la que las redes neuronales son tolerantes a fallos es que tienen su información distribuida en las conexiones entre neuronas, existiendo cierto grado de redundancia en ese tipo de almacenamiento, a diferencia de la mayoría de los ordenadores algorítmicos y sistemas de recuperación de datos que almacenan cada pieza de información en un estado único, localizado y direccionable.
4. **Operación en Tiempo Real:** Los computadores neuronales pueden ser realizados en paralelo, y se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad.
5. **Fácil inserción dentro de la tecnología existente.** Debido a que una red puede ser rápidamente entrenada, comprobada, verificada y trasladada a una implementación hardware de bajo costo, es fácil insertar RNA para aplicaciones específicas dentro de sistemas existentes (chips, por ejemplo). De esta manera, las redes neuronales se pueden utilizar para mejorar sistemas de forma incremental, y cada paso puede ser evaluado antes de realizar un desarrollo más amplio.

Capítulo 4. Metodología para diseñar la arquitectura de una red neuronal artificial que ayude al proceso de toma de decisiones.

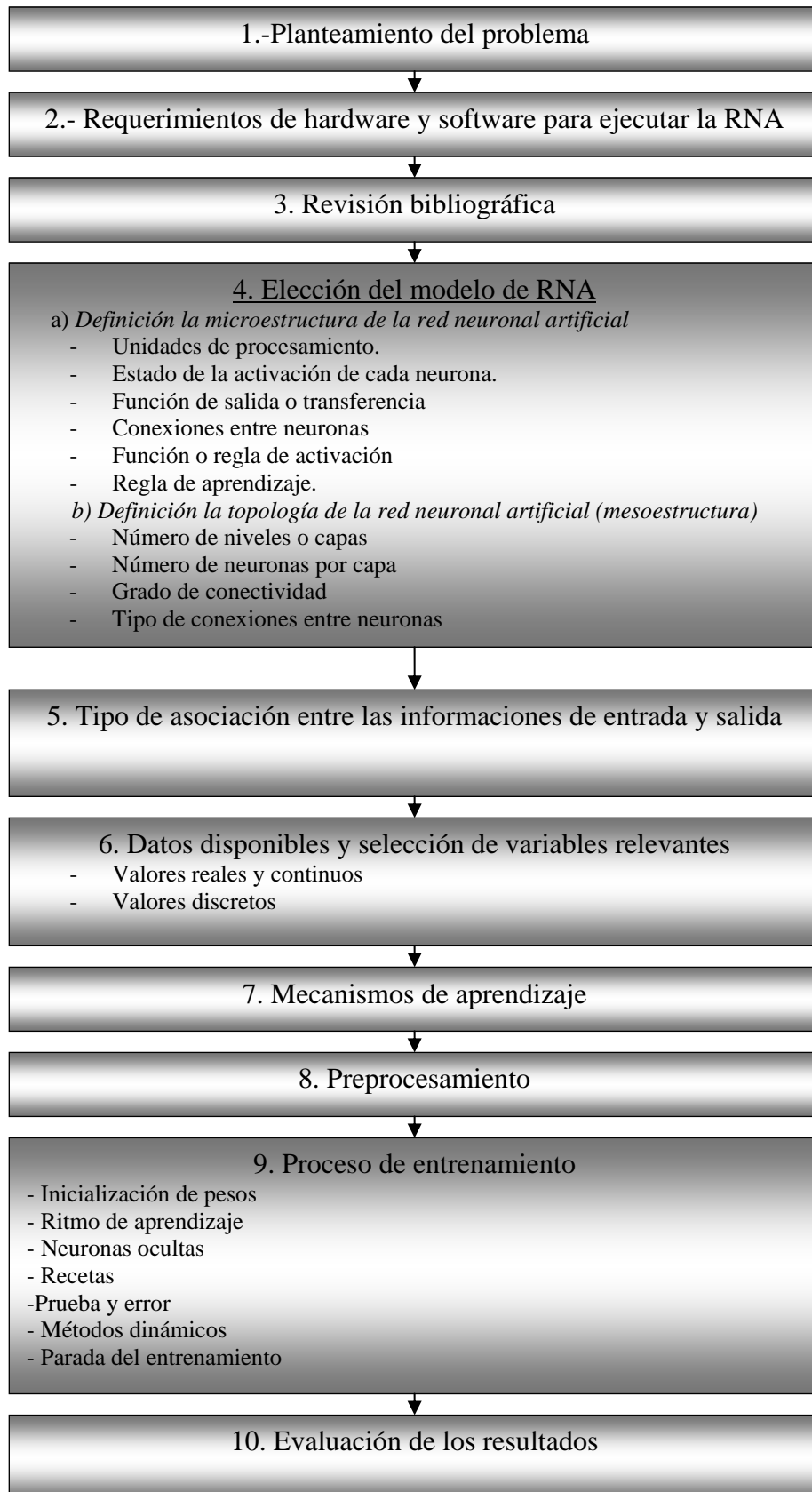
Como ya se ha mencionado, las redes neuronales artificiales son capaces de descubrir automáticamente la relación entrada-salida (o rasgos característicos) en función de datos empíricos, merced a su capacidad de aprendizaje a base de ejemplos, esto nos permite resolver problemas que ayuden a una mejor toma de decisiones.

Los objetivos que se persiguen al utilizar redes neuronales artificiales son mucho más modestos que la creación de un *cerebro artificial*. Las redes neuronales se utilizan en la resolución de problemas prácticos concretos, que normalmente no han sido resueltos mediante sistemas tradicionales. “Gracias a su capacidad de aprendizaje, robustez, no linealidad y tolerancia a la imprecisión e incerteza del entorno, desde hace unos años las redes neuronales, vienen alcanzando excelentes resultados en aplicaciones diversas”. Martín y Sanz. (2002:36).

La lista de problemas en los que se han aplicado con éxito las redes neuronales crece constantemente (Widrow 1994,93-105), esto se aprecia en cada nueva edición de congresos o revistas especializadas. Las preguntas que surgen inmediatamente son: ¿dónde radica el éxito de las redes neuronales artificiales?, ¿podré beneficiarme de estas nuevas herramientas?, ¿cuáles son las características que debe tener mi problema para que se justifique el empleo de una RNA?, ¿puedo desarrollar o modificar arquitecturas útiles para el problema en cuestión y, si es necesario entrenar la red (determinando los parámetros)?, ¿existen herramientas formales que puedan aplicarse para la solución y ejecución de una RNA que resuelva mi problema?, Schalkoff (1997:10).

Para resolver un problema mediante redes neuronales artificiales se debe aplicar una metodología que presente aspectos comunes con las técnicas convencionales de datos (véase la figura 32). Vamos a describir los pasos que deben seguirse para el diseño y la aplicación de una red neuronal, basados en lo que afirman los autores Martín y Sanz (2002:36). En el capítulo 5 de este trabajo se plantea una metodología más detallada de estos mismos pasos para resolver un ejemplo concreto de toma de decisiones: el caso de estudio de este trabajo de investigación, que tiene que ver con la clasificación de tutores del Programa de Posgrado en Ciencias de la Administración de la UNAM.

Figura 32. Descripción de los pasos metodológicos para diseñar la arquitectura de una red neuronal.



Paso 1. Planteamiento del problema.

El primer paso de esta metodología consiste en realizar una descripción detallada del problema para averiguar si algún aspecto de él podría ser resuelto mediante algún módulo neuronal. Menciona Giarratano (2001:46) “Una red neuronal es una buena opción cuando hay una cantidad considerable de datos empíricos y no contamos con un algoritmo que proporcione velocidad y precisión suficientes”. No debemos pensar que las redes neuronales son la mejor solución en todos los casos, en muchas ocasiones será más interesante aplicar un algoritmo convencional, Martín y Sanz. (2002:36) proponen algunas características que deben cumplir los problemas para ser resueltos con RNA's y las características que hacen desaconsejable el empleo de éstas.

a) Características que debe cumplir el problema para ser resuelto por una RNA:

- No se cuenta con un conjunto de reglas sistemáticas que describan completamente el problema.
- Se cuenta con muchos ejemplos o casos históricos (ésta es una condición indispensable para poder aplicar técnicas de RNA).
- Los datos procedentes del problema son imprecisos.
- El problema es de elevada dimensionalidad.
- Los métodos de RNA proporcionan una alternativa mucho más rápida y sencilla de desarrollar que otras técnicas convencionales.
- Si las condiciones de trabajo son cambiantes se puede hacer uso de la capacidad de la RNA para adaptarse a estos cambios re-entrenando el sistema con nuevos ejemplos.

b) Características que hacen desaconsejable el empleo de RNA.

- Existe algún algoritmo que resuelve el problema planteado.
- No se cuenta con un número adecuado de casos para entrenar la red neuronal.

- A veces no resulta fácil interpretar la operación de la red neuronal o predecir con total fidelidad el resultado que pueda proporcionar en todos los casos posibles.

Una vez que hemos analizado nuestro problema y determinamos que puede ser resuelto por una RNA, revisaremos los requerimientos de hardware y de software necesarios para ejecutar la red neuronal.

Paso 2. Requerimientos de hardware y software para ejecutar la red neuronal.

En este paso es donde debemos conocer las especificaciones de la arquitectura del sistema que requerimos para ejecutar la red neuronal en relación al problema planteado en el punto anterior, es decir, debemos responder a las preguntas de cómo vamos a resolver el problema y ¿qué clase de computadora está disponible en cuanto a: memoria, almacenamiento y procesadores para entrenar e implementar la red que resuelve el problema planteado? Schalkoff (1997:10). Entrenar la red para que obtenga las respuestas puede tomar horas o días, dependiendo del número de patrones que la red deba aprender y del hardware y el software que se va a utilizar; sin embargo, una vez que la red realiza el aprendizaje, ésta aprende con rapidez.

Si la simulación del software no es lo bastante rápida, se puede fabricar la RNA en chips para que responda en tiempo real; una vez que se ha entrenado a la red de trabajo y se han determinado los pesos, se puede construir un chip, AT&T y otras compañías están fabricando chips experimentales para redes neuronales que contienen cientos de neuronas, es muy probable que en los próximos años se fabriquen chips que contengan miles de ellas Giarratano (2001:45).

2.1 Diversas arquitecturas para ejecutar redes neuronales artificiales:

2.1.1. Los neurocomputadores:

Son un conjunto de procesadores conectados con cierta regularidad que operan concurrentemente, otra forma de ejecutar redes neuronales consiste en la implementación de éstas por medio de uno o varios circuitos integrados específicos, para así poder obtener una estructura que se comporte lo más similar posible a cómo lo haría una red neuronal, Hilera (2000:32).

2.1.2. La tecnología micro eléctrica:

Esta tecnología ha dado buenos resultados, sin embargo hay problemas que no se han podido resolver como la dificultad de obtener un alto grado de interconexión, o la entrada masiva de los datos, condicionada al número de pines, o, por último el poder conseguir sinapsis con pesos variables, lo cual resulta necesario si se desea que la red tenga un buen aprendizaje, Hilera (2000:32).

2.1.3. La tecnología electroóptica:

Esta tecnología apareció recientemente para ejecutar una red neuronal y su característica es que utiliza la luz como medio de transporte de la información permitiendo la transmisión masiva de ésta; sin embargo, dicha tecnología no ha sido todavía probada.

2.1.4. El ordenador convencional

Esta es la realización más simple y consiste en simular la red sobre un ordenador convencional y sobre un software específico. Es un procedimiento rápido, poco costoso e insustituible por el momento para realizar el entrenamiento y evaluación de las redes, pero cuya mayor desventaja radica en que se intenta simular redes con un alto grado de paralelismo sobre máquinas que ejecutan secuencialmente las operaciones, esto provoca que los valores intrínsecos de las redes no puedan obtenerse.

2.1.5 Transputers o arquitecturas sistólicas:

Estas arquitecturas son orientadas a la ejecución de procesos con un alto grado de paralelismo, tales como redes. El objetivo de tales redes es acelerar la simulación de la red neuronal, permitiendo, si es posible, una respuesta en tiempo real.

2.1.6 Los chips neuronales,

Esta es una aproximación radicalmente distinta en donde las neuronas y conexiones se emulan con dispositivos específicos, de forma que el circuito integrado refleja la arquitectura de la red. De esta forma se consigue gran velocidad, permitiendo en

muchas ocasiones el proceso en tiempo real, pero a costa de una pérdida notable de flexibilidad, Hilera (2000:34).

En lo que respecta al software, se ha llegado a un importante desarrollo con lenguajes simbólicos apoyados en arquitecturas convencionales sobre ordenadores. Actualmente, existen un número bastante elevado de productos software para el desarrollo de aplicaciones con redes neuronales, en la tabla 5 se muestran una serie de ejemplos de productos comerciales, estos productos están soportados en arquitecturas de hardware convencional.

Tabla 5. Herramientas de desarrollo software comerciales.

<i>Producto</i>	<i>Vendedor</i>	<i>Hardware</i>	<i>Sistema Operativo</i>
ANSim	SAIC	PC	DOS
AnSkit	SAIC	PC	DOS
Autonet	Peak Software	PC	DOS
Brain Maker	California Scientific Software	PC; PS/2; Mac	DOS
NeuroSolutions ¹	Neurodimension, Inc.	PC	DOS; Windows
Alyuda	Copyright © 2001-2005 Alyuda Research, Inc.	PC	Windows
NeuroIntelligence 2.2 (577)			
DynaMind	NeuroDynamix	PC	DOS
ExpertNet	ImageSoft	PC	DOS; Windows
Genesis	Neural Systems	PC	DOS
HNet	AND america	PC (ANDcoprocesor)	DOS; Windows
MacBrain	Neurix	Mac	
NetBuilder	NeuralWare	PC	DOS
NETS	COSMIC/NASA	PC; Mac; VAX; Sun; HP 9000	DOS; UNIX; VMS
NeuralCASE	NeuroSym	PC	DOS
Neuraldesk	Neural Computer Sciences	PC	Windows
Neurokit	Software Bytes	PC	DOS
NeuroShell	Ward Systems	PC	DOS
NeuralWorks Explorer	NeuralWare	PC; Mac	DOS; Mac
N-Net	AI Ware	PC; PS/2; Sun; SPARC; VAX	DOS; UNIX; VMS
SARA	IIC	PC; PS/2	DOS
ExploreNet	HNC	PC	DOS; Windows
Neural Networks Utility/2	IBM	PS/2	DOS; Windows; OS/2

¹ Copyright© 1994-2005. Este es el software utilizado para resolver el caso de estudio de este trabajo de investigación.

<i>Producto</i>	<i>Vendedor</i>	<i>Hardware</i>	<i>Sistema Operativo</i>
Neural Networks Utility/400	IBM	AS/400	OS/400
Neuralworks Pro II/Plus	NeuralWare	PC; PS/2; Mac; RS/6000; SPARC; DEC; HP; Silicon Graphics	DOS; OS/2; UNIX; VMS

Fuente: Hilera (2000:37).

Paso 3. Revisión bibliográfica

Recomiendan los autores Martín y Sanz (2002:212), que resulta bueno hacer un sondeo en busca de alguna aplicación parecida a la que se nos plantea observando la estrategia de otras personas en problemas similares; por lo que los autores mencionados recomiendan los recursos de Internet para obtener referencias recientes y programas de simulación, al final de este trabajo de investigación se incluyen la bibliografía y las páginas de Internet consultadas para la realización del mismo.

Paso 4. Elección del modelo de RNA

Una vez que hemos especificado con todo detalle las características de nuestro problema, debe elegirse un modelo de la RNA para comenzar hacer pruebas. Para lograr lo anterior debemos comenzar por definir la *microestructura y mesoestructura* o topología de la red neuronal, y ya una vez definidas, revisaremos los modelos con las arquitecturas más empleadas, para finalmente elegir el modelo más adecuado al problema a resolver.

4.1 Definición del modelado de la red neuronal artificial (microestructura)

En este paso de la metodología, se definirá la *microestructura*, o sea, el modelado de la red para lo cual se mostrarán los componentes necesarios para definirla: tipos de unidades de proceso o neuronas, su estado de activación, su función de salida y de activación. Posteriormente, se elegirá su conexión con otras neuronas y la regla de aprendizaje de la red neuronal.

4.1.1 Unidades de procesamiento.

Las unidades de proceso o neuronas N , se pueden ordenar arbitrariamente y designar la j -ésima unidad como U_j . Su trabajo

consiste en recibir las entradas de las neuronas vecinas y calcular un valor de salida, el cual es enviado a todas las neuronas restantes. [Hilera (2000:51)].

Se debe considerar que en cualquier sistema que se esté modelando debemos reconocer tres tipos de unidades: *entradas*, *salidas* y *ocultas*. Las unidades de *entrada* son las que reciben señales desde el entorno; estas son las entradas a la red. Las unidades de *salida* envían la señal fuera del sistema y son las salidas de la red. Las unidades *ocultas* son aquellas cuyas salidas cuyas entradas y salidas se encuentran dentro del sistema.

4.1.2 Estado de la activación de cada neurona.

Después de definir el conjunto de neuronas, [Hilera (2000:51)].

La representación requiere que los estados del sistema se encuentren en un tiempo t . Esto debe representarse por un vector de N números reales $A(t)$, que representan el *estado de activación* del conjunto de unidades de procesamiento. Cada elemento del vector representa la activación de una unidad en el tiempo t . La activación de una unidad U_i en el tiempo t se denomina como $a_i(t)$; es decir:

$$A(t) = (a_1(t), a_2(t), \dots, a_i(t), \dots, a_N(t))$$

Todas las neuronas que componen la red se hallan en cierto estado. Existen dos posibles estados (en una versión simple), reposo y excitado, a éstos se les denomina estados de activación, y a cada uno de los cuales se les asigna un valor. Los valores de activación pueden ser continuos o discretos. Si tomamos valores discretos, éstos pueden ser binarios, en donde un estado activo se indicaría con un 1, y se caracteriza por la emisión de un impulso, mientras que un estado pasivo se indicaría por un 0, y significaría que la neurona está en reposo. Si tomamos valores continuos, se les asigna un valor entre $[0,1]$ o en el intervalo $[-1,1]$, generalmente siguiendo una función sigmoideal.

4.1.3 Función de salida o transferencia

En esta etapa se deben definir las conexiones entre las neuronas que forman una red neuronal artificial. Cada unidad transmite señales a aquellas que están conectadas con su salida. Asociada con cada unidad U_i hay una función de salida $f_i(a_i(t))$, que

transforma el estado actual de activación $a_i(t)$ en una señal de salida $y_i(t)$;

$$y_i(t) = f_i(a_i(t))$$

El vector que contiene las salidas de todas las neuronas en un instante t es:

$$Y(t) = (f_1(a_1(t)), f_2(a_2(t)), \dots, f_i(a_i(t)), \dots, f_N(a_N(t)))$$

En algunos modelos, esta salida es igual al nivel de activación de la unidad, en cuyo caso la función f_i es la función identidad, $f_i(a_i(t)) = a_i(t)$. A menudo, f_i es del tipo sigmoideal, y suele *ser la misma para todas las unidades*.

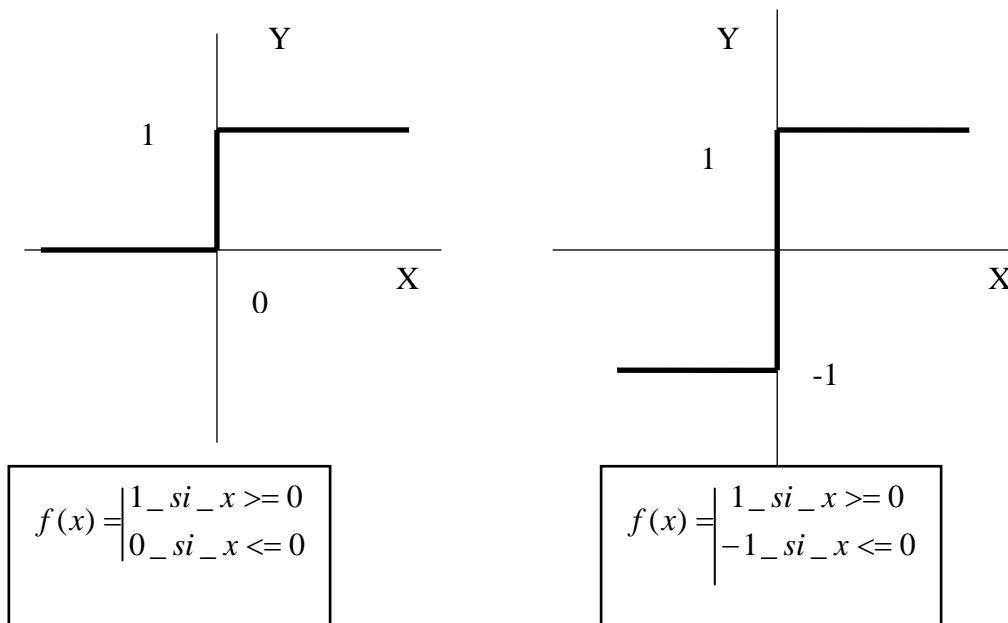
Como ya se mencionó en el marco teórico de este trabajo, existen cuatro funciones de transferencia que determinan distintos tipos de neuronas: función escalón, función lineal y mixta, sigmoideal y función gaussiana.

La función escalón o umbral únicamente se utiliza cuando las salidas de la red son binarias (dos posibles valores). La salida de una neurona se activa sólo cuando el estado de activación es mayor o igual que cierto valor umbral (la función puede estar desplazada sobre los ejes). La función lineal o identidad equivale a no aplicar función de salida. Se usa muy poco. Las funciones mixta y sigmoideal son las más apropiadas cuando queremos como salida información analógica.

*** Función escalón**

La forma más fácil de definir la activación de una neurona es considerar que ésta es binaria. La función de transferencia escalón se asocia a neuronas binarias en las cuales cuando la suma de las entradas es mayor o igual que el umbral de la neurona, la activación es 1; si es menor, la activación es 0 (o -1). Por otro lado, las redes formadas por este tipo de neuronas son fáciles de implementar en hardware, pero a menudo sus capacidades son limitadas (Véase figura 33).

Figura 33. Función de transferencia escalón



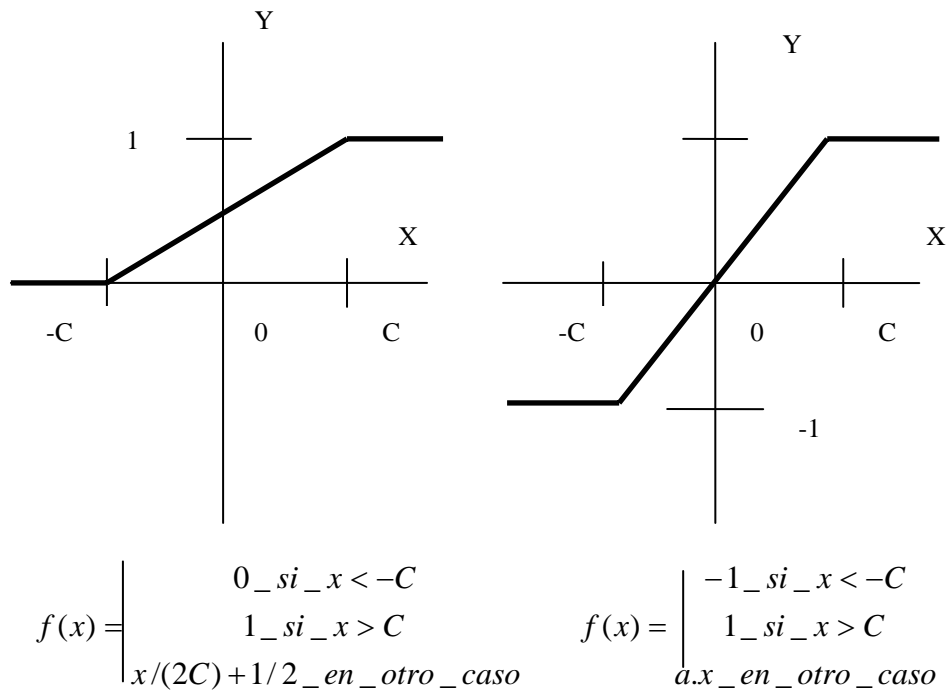
En ambos casos se ha tomado que el umbral es cero; en caso de que no lo fuera, el escalón quedaría desplazado.

Fuente: Hilera (2000:55)

* *Función lineal y mixta*

La función lineal o identidad responde a la expresión $f(x) = x$. En las neuronas con función mixta, si la suma de las señales de entrada es menor que un límite inferior, la activación se define como 0 (o -1). Si dicha suma es mayor o igual que el límite superior su activación es 1. Si la suma de entrada está comprendida entre ambos límites, superior e inferior, entonces la activación se define como una función lineal de la suma de las señales de entrada. Podemos representar las funciones de activación como se muestra en la figura 34, donde se toma el límite superior de la suma de todas las entradas de activación que afectan a la neurona durante el ciclo de operación (x) como c y el límite inferior como $-c$, y es la salida de activación de la neurona.

Figura 34. Funciones de activación mixta



Fuente: Hilera (2000:56)

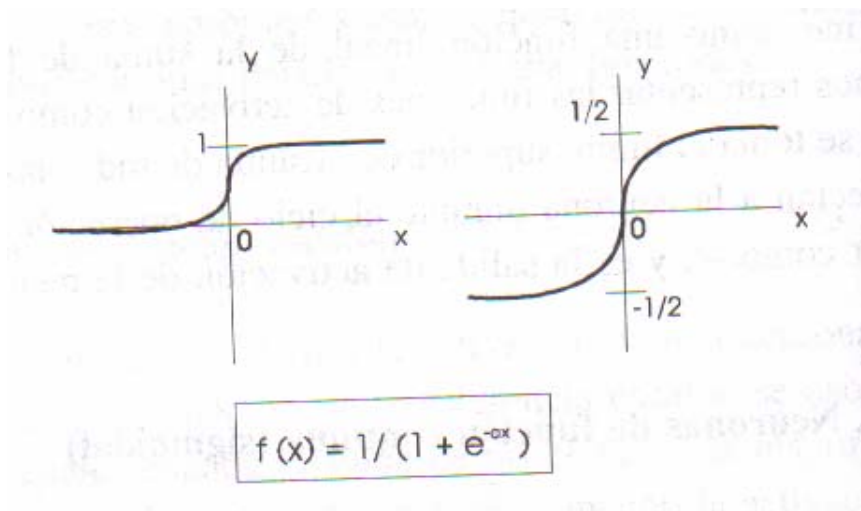
*** Función continua (sigmoideal)**

Cualquier función definida simplemente en un intervalo de posibles valores de entrada, con un incremento monótonico y que tenga ambos límites superiores e inferiores (por ejemplo, las funciones sigmoideal y arcotangente), podrá realizar la función de activación o de transferencia en forma satisfactoria.

Con la función sigmoideal, para la mayoría de los valores del estímulo de entrada (variable independiente), el valor dado por la función es cercano a uno de los valores asintóticos. Esto hace que en la mayoría de los casos, el valor de salida esté comprendido en la zona alta o baja del sigmoide. De hecho, cuando la pendiente es elevada, esta función tiende a la función escalón. Sin embargo, la importancia de la función sigmoideal (o cualquier otra función similar) es que su derivada es siempre positiva y cercana a cero para valores grandes positivos o negativos; además toma su valor máximo cuando x es 0. Esto hace que se puedan utilizar las reglas de aprendizaje definidas para las funciones escalón, con la ventaja, respecto a esta función, de que la derivada está definida en todo el intervalo (figura 35). La función escalón no podría definir la

derivada en el punto de transición, y esto no ayuda a los métodos de aprendizaje en los cuales se usan derivadas, Hilera (2000:56).

Figura 35. Funciones de activación continua (sigmoidal)

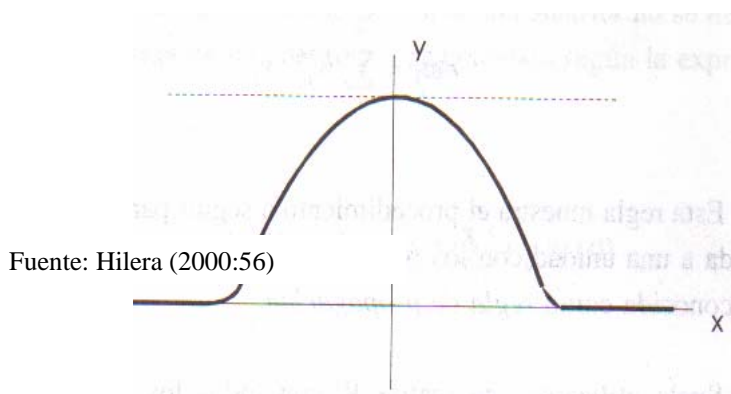


Fuente: Hilera (2000:56)

Función de transferencia gaussiana.

Los centros y anchura de estas funciones pueden ser adaptados, lo cual las hace más adaptativas que las funciones sigmoideas. Mapeos que suelen requerir dos niveles ocultos (neuronas en la red que se encuentran entre las de entrada y las de salida) utilizando neuronas con funciones de transferencia sigmoideas; algunas veces se pueden realizar con un solo nivel en redes de neuronas de función gaussiana. (Véase la figura 36).

Figura 36. Función de transferencia gaussiana



4.1.4 Conexiones entre neuronas

Las conexiones entre las neuronas tienen asociado un peso por medio del cual la red adquiere conocimiento. Consideremos y_i como el valor de salida de una neurona i en un instante dado. Una neurona recibe un conjunto de señales que le dan información del estado de activación de todas las neuronas con las que se encuentra conectada. Cada conexión (llamada sinapsis) entre la neurona i y la neurona j está ponderada por un peso w_{ji} . Normalmente, como simplificación, se considera que el efecto de cada señal es aditivo, de tal forma que la entrada neta que recibe una neurona (potencial postsináptico) net_j es la suma del producto de cada señal individual por el valor de las sinapsis que conecta ambas neuronas:

$$net_j = \sum_i^N w_{ji} \cdot y_i$$

Esta regla muestra los procedimientos a seguir para combinar los valores de entrada a una unidad con los pesos de las conexiones que llega a esa unidad y es conocida como *regla de propagación*.

Suele utilizarse una matriz W con todos los pesos w_{ji} que reflejan la influencia que sobre la neurona j tiene la neurona i . W es un conjunto de elementos positivos, negativos o nulos. Si w_{ji} es positivo, indica que la interacción entre las neuronas i y j es excitadora; es decir, siempre que la neurona i esté activada, la neurona j recibirá una señal de i que tenderá a activarla. Si w_{ij} es negativo, la sinapsis será inhibitoria. En este caso, si i está activada, enviará una señal a j que tenderá a desactivar a ésta. Finalmente, si $w_{ij} = 0$, se supone que no hay conexión entre ambas.

4.1.5 Función o regla de activación

En esta etapa es necesario establecer una regla que combine las entradas a una neurona con los pesos de las conexiones, también se requiere una regla que combine las entradas con el estado actual de la neurona para producir un nuevo estado de activación. Esta función F produce un nuevo estado de activación en una

neurona a partir del estado (a_i) que existía y la combinación de las entradas con los pesos de las conexiones (net_i).

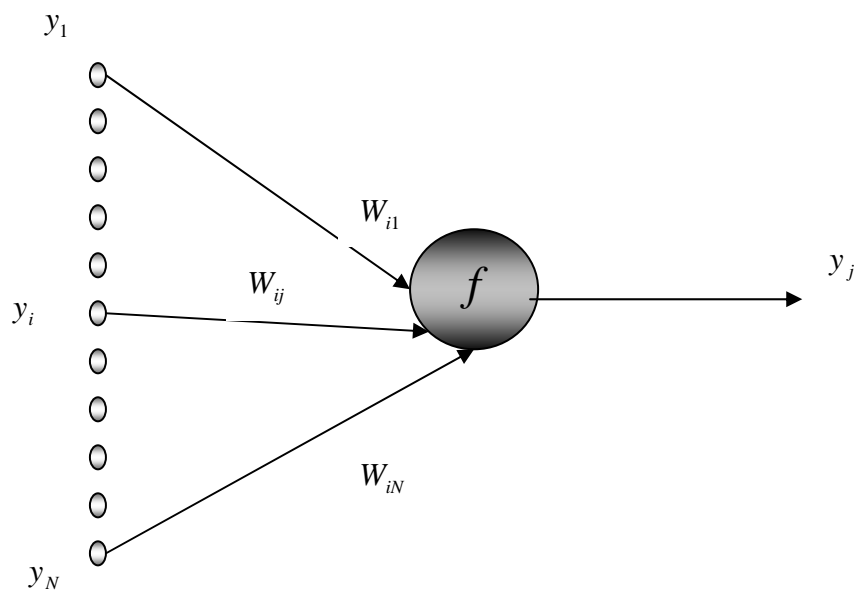
Dado el estado de activación $a_i(t)$ de la unidad U_i y la entrada total que llega a ella, Net_i el estado de activación siguiente, $a_i(t+1)$, se obtiene aplicando una función F , llamada *función de activación*.

$$a_i(t+1) = F(a_i(t), Net_i)$$

En la mayoría de los casos, F es la *función identidad*, por lo que el estado de activación de una neurona en $t+1$ coincidirá con el Net de la misma en t . (Véase la figura 37), en este caso, el parámetro que se le pasa a la función de salida, f , de la neurona será directamente el Net . El estado de activación anterior no se tiene en cuenta. Según esto, la salida de una neurona $i(y_i)$ quedará según la expresión:

$$y_i(t+1) = f(Net_i) = f\left(\sum_{j=1}^N w_{ij} y_j(t)\right)$$

Figura 37. Representación de la función de transferencia de una neurona.



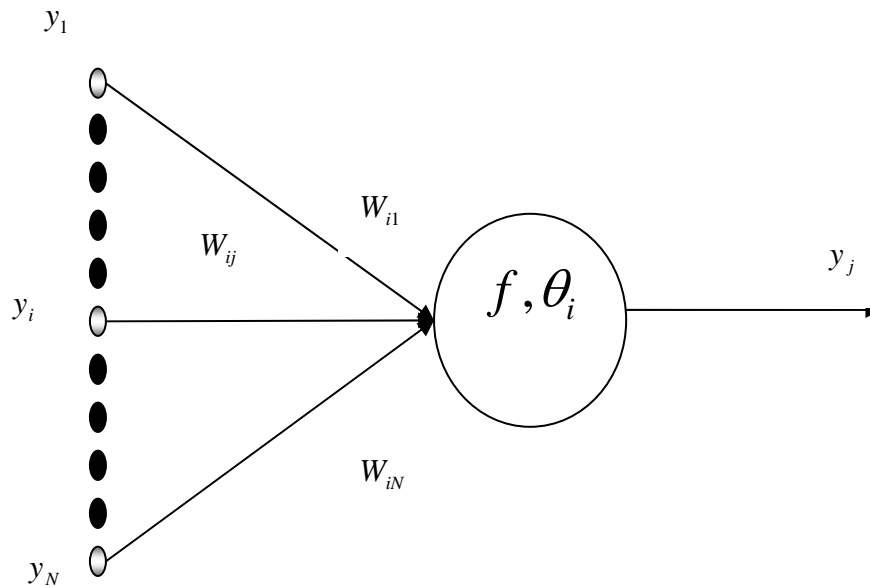
Fuente: Hilera (2000:56)

Por lo tanto, y en lo sucesivo, consideraremos únicamente la función f , que denominaremos indistintamente de *transferencia* o

de activación. Además, normalmente la función de activación no está centrada en el origen del eje que representa el valor de la entrada neta, sino que existe cierto desplazamiento debido a las características internas de la propia neurona y que no es igual a todas ellas, (figura 38). Este valor se denota como θ_i y representa el umbral de activación de la neurona i .

$$y_i(t+1) = f(Net_i - \theta_i) = f\left(\sum_{j=1}^N w_{ij} y_j(t) - \theta_i\right)$$

Figura 38. Representación de la función de transferencia con el umbral de activación de la neurona.



Fuente: Hilera (2000:57)

La salida que se obtiene de una neurona para las diferentes formas de la función f serán:

4.1.5.1 Función de activación escalón

Si el conjunto de los estados de activación es $E = \{0,1\}$, tenemos que:

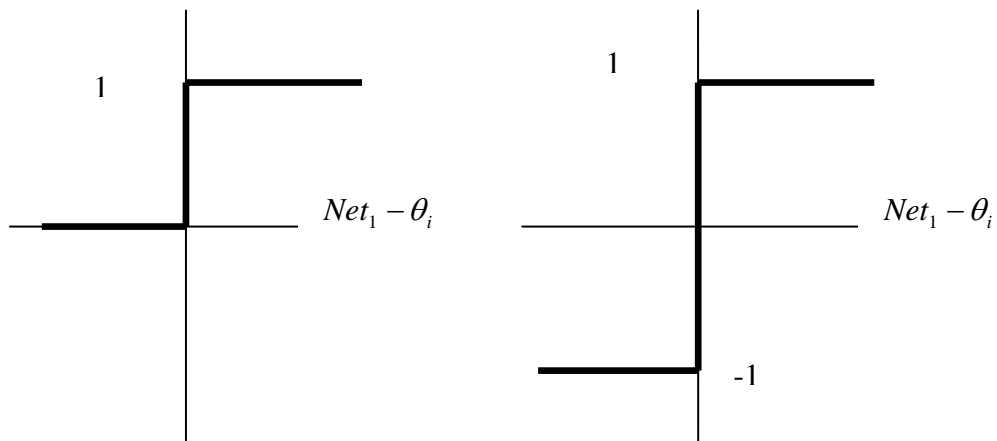
$$y_i(t+1) = \begin{cases} 1 & \text{si } [Net_i > \theta_i] \\ y(t) & \text{si } Net_i = \theta_i \\ 0 & \text{si } [Net_i < \theta_i] \end{cases}$$

Si el conjunto de los estados de activación es $E = \{-1,1\}$, tenemos que:

$$y_i(t+1) = \begin{cases} 1 & \text{si } [Net_i > \theta_i] \\ y(t) & \text{si } Net_i = \theta_i \\ -1 & \text{si } [Net_i < \theta_i] \end{cases}$$

Véase la figura 39.

Figura 39. Función de activación escalón



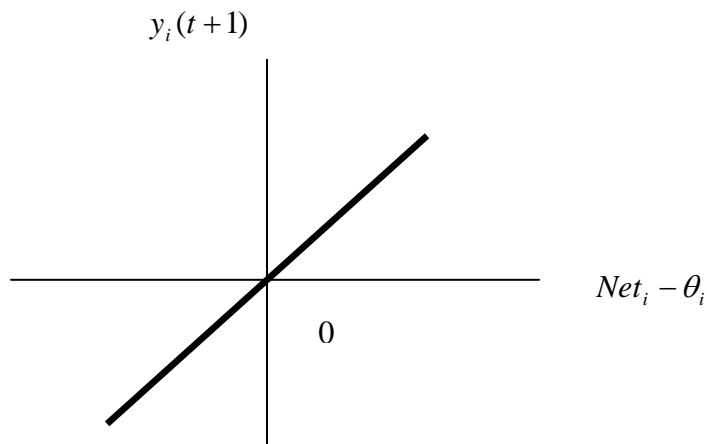
Fuente: Hilera (2000:61)

4.1.5.2 Función de activación lineal o identidad

El conjunto de estados E puede contener cualquier número real; el estado de activación coincide con la entrada total que ha llegado a la unidad, (véase figura 40):

$$y_i(t+1) = Net_i - \theta_i$$

Figura 40. Función de activación lineal o identidad



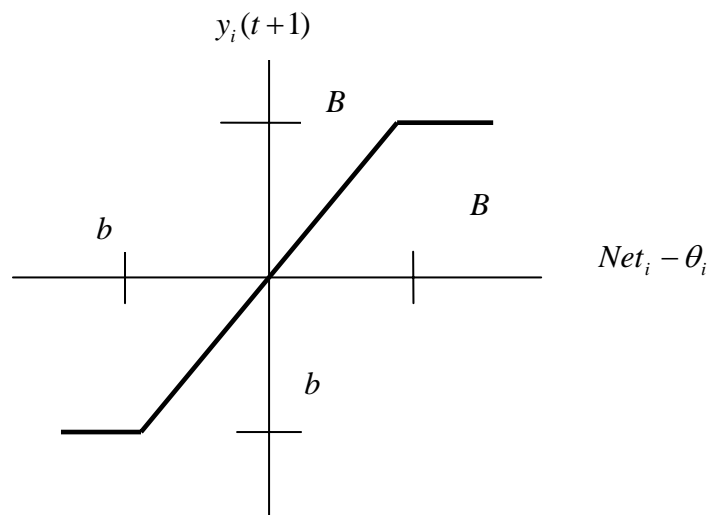
Fuente: Hilera (2000:61)

4.1.5.3 Función de activación lineal-mixta

$$y_i(t+1) = \begin{cases} b & Net \leq b + \theta_i \\ Net_i - \theta_i & b + \theta_i < Net_i < B + \theta_i \\ B & Net_i \geq B + \theta_i \end{cases}$$

Con esta función, el estado de activación de la unidad está obligado a permanecer dentro de un intervalo de valores reales prefijado, (véase la figura 41).

Figura 41. Función de activación lineal-mixta



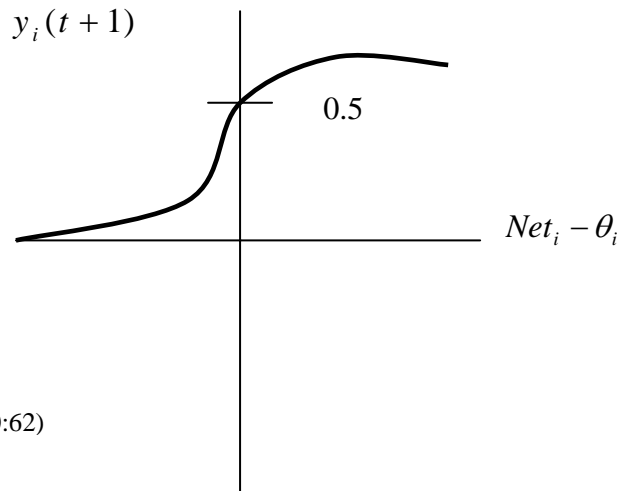
Fuente: Hilera (2000:61)

4.1.5.4 Función de activación sigmoideal

Es una función continua, por lo tanto el espacio de los estados de activación es un intervalo del eje real, (véase la figura 42).

$$y_i(t+1) = \frac{1}{(1 + e^{-(Net_i - \theta_i)})}$$

Figura 42. Función de activación sigmoideal



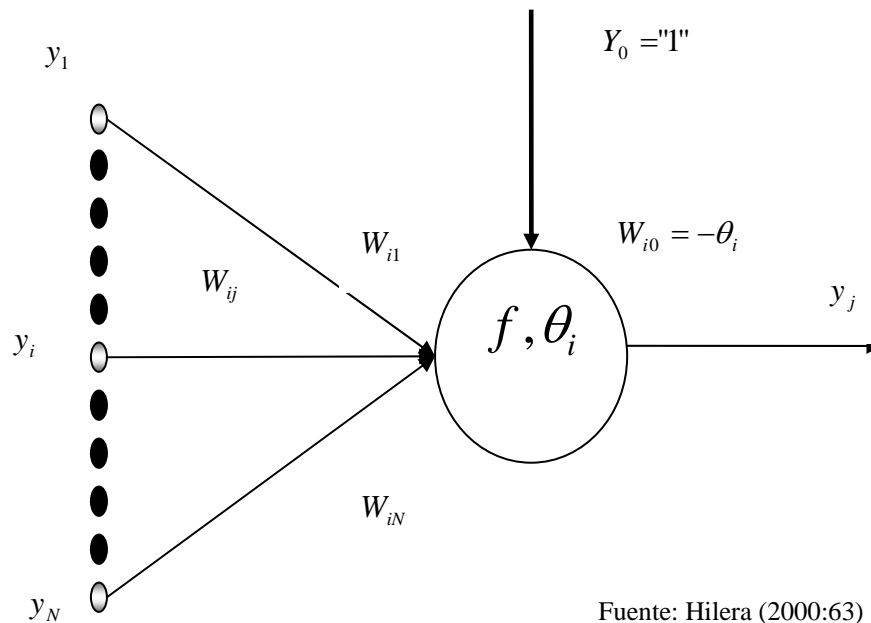
Fuente: Hilera (2000:62)

Para simplificar la expresión de la salida de una neurona i , es habitual considerar la existencia de una neurona ficticia, con valor de salida unitaria, asociada a la entrada de cada neurona i mediante una conexión con peso de valor $-\theta_i$. De esta forma la expresión de salida quedará, (véase la figura 43).

$$y_i(t+1) = f\left(\sum_{j=1}^N w_{ij} y_j(t) - \theta_i \cdot 1\right) = f\left(\sum_{j=0}^N w_{ij} y_j(t)\right) = f(Net_i)$$

.....

Figura 43. Representación de una neurona con valor de salida unitaria asociada a la entrada.



4.1.6 Regla de aprendizaje

Como se menciona en el marco teórico de este trabajo, se puede considerar que el conocimiento se encuentra representado en los *pesos* de las conexiones entre neuronas. Todo proceso de aprendizaje implica cierto número de cambios en estas conexiones. En realidad puede decirse que *aprende* modificando los valores de los pesos de la red.

Al igual que el funcionamiento de una red depende del número de neuronas de las que disponga y de cómo estén conectadas entre sí, cada modelo dispone de su o sus propias técnicas de aprendizaje.

4.2 Definición la topología de la red neuronal artificial (mesoestructura).

Una RNA está confeccionada y entrenada para llevar a cabo una labor específica. La *topología* de la red se refiere a los diferentes modelos de conexiones entre capas, en general se suelen distinguir dos básicos: las arquitecturas hacia adelante o *feedforward*, (la información se propaga hacia adelante) (véase figura 21), y las realimentadas o *feedback* (las señales pueden fluir

en ocasiones hacia atrás a través de lazos de realimentación, figura 22).

La topología o *mesoestructura* de una red neuronal está definida por:

- Número de niveles o capas
- Grado de conectividad entre las neuronas
- Tipo de conexiones entre neuronas
 - Redes monocapa
 - Redes multicapa

4.2.1 Número de niveles o capas

En esta etapa de la metodología, debemos tomar en cuenta la distribución de las neuronas dentro de la red ya que ésta forma niveles o capas con un determinado número de neuronas cada una. Dependiendo su situación dentro de la red, se pueden distinguir tres tipos de capas:

- *De entrada.* Es la capa que recibe directamente la información proveniente de las fuentes externas de la red.
- *Ocultas.* Son internas a la red y no tienen contacto directo con el entorno exterior. El número de niveles ocultos puede estar entre cero y un número elevado. Las neuronas de las capas ocultas pueden estar interconectadas de distintas maneras, lo que determina, junto con su número, las distintas tipologías de redes neuronales.
- *De salida.* Transfieren información de la red hacia el exterior.

La clasificación de las redes en términos topológicos, se suele distinguir entre las redes con una sola capa o nivel de neuronas y las redes con múltiples capas (2,3, etc.).

4.2.2 Grado de conectividad

Formas de conexión entre neuronas. Debemos tomar en cuenta que la conectividad de una red neuronal está relacionada con la forma en que las salidas de las neuronas están canalizadas para convertirse en entradas de otras neuronas. La señal de salida de un nodo puede ser la entrada de otro elemento de proceso, o incluso ser una entrada de sí mismo (conexión autorrecurrente).

Cuando ninguna salida de las neuronas es entrada de neuronas del mismo nivel o niveles precedentes, la red se describe como de

propagación hacia delante (Gráfica 42). Cuando las salidas pueden ser conectadas como entradas de neuronas de niveles previos o del mismo nivel, incluyéndose ellas mismas, la red es de propagación hacia atrás (tabla 6). Las redes de propagación hacia atrás que tiene lazos cerrados son sistemas recurrentes.

4.2.3 Tipo de conexiones entre neuronas

4.2.3.1 Redes monocapa (1 capa).

En las redes monocapa se establecen conexiones laterales entre las neuronas que pertenecen a la única capa que constituye la red.

Las redes monocapa se utilizan típicamente en tareas relacionadas con lo que se conoce como autoasociación; por ejemplo, para regenerar informaciones de entrada que se presentan a la red, incompletas o distorsionadas. (Véase tabla 6 y la figura 17).

Tabla 6. Redes neuronales monocapa más conocidas.

TIPOS DE CONEXIONES		MODELO DE RED
Conexiones laterales explícitas	Conexiones autorrecurrentes	BRAIN-STATE-IN-A BOX
		ADDITIVE GROSSBERG (AG)
		SHUNTING GROSSBERG (SG)
		OPTIMAL LINEAL ASOCIATIVE MEMORY
	Conexiones no autorrecurrentes	HOPFIELD
		BOLTZMANN MACHINE
		CAUCHY MACHINE
Crossbar		LEARNING MATRIX (LM)

Fuente (Hilera 2000: 71)

4.2.3.2 Redes multicapa

Las redes multicapa son aquellas que disponen de un conjunto de neuronas agrupadas en varios niveles o capas (2, 3, etc.). En estos casos, una forma para distinguir la capa a la que pertenece una neurona, consistirá en fijarse en el origen de las señales que recibe la entrada y el destino final de la salida, (véase la figura 18).

Conexiones hacia delante o feedforward. Todas las neuronas de una capa reciben señales de entrada de otra capa anterior, más cercana a las entradas de la red, y envían las señales de salida a una capa posterior, más cercana a la salida de la red.

Conexiones hacia atrás o feedback. En estas redes también existe la posibilidad de conectar las salidas de las neuronas de capas posteriores a las entradas de las capas anteriores.

Conexiones adelante/atrás o feedforward/feedback. Éstas disponen de conexiones hacia adelante como hacia atrás.

En la tabla 7, distinguimos las características topológicas de redes multicapa de 2, 3 y más capas, con conexiones *hacia delante*, *hacia atrás* y *adelante/atrás*, así como su modelo de red.

Tabla 7. Redes neuronales multicapa más conocidas.

Número de capas	Tipo de conexiones		Modelo de red
Dos capas	Conexiones hacia adelante feedforward		ADALINE/MADALINE
			PERCEPTRON
			LINEAR/ASSOC REWARD PENALTY
			LINEAR ASSOCIATIVE MEMORY
			OPTIMAR LINEAR ASSOC. MEM
			DRIVE-REINFORCEMENT (DR)
	Conexiones adelante/atrás Feedforward/feedback	Conexiones laterales implícitas y autorrecursivas	LEARNING VECTOR QUANTIZER
			TOPOLOGY PRESERVING MAP
		Sin conexiones laterales	BIDIRECTIONAL ASSOC. MEM.
			ADAPTIVE BAM.
Tres capas	Conexiones hacia adelante (feedforward)	TEMPORAL ASSOC. MEMORY	
		FUZZY ASSOCIATIVE MEMORY	
	Con conexiones laterales y autorrecursivas	COMPETITIVE ADAPTIVE BAM	
		ADAPTIVE RESONANCE THEORY (ART)	
Conexiones adelante/atrás y laterales	Sin conexiones laterales	ADAPTIVE HEURISTIC CRITIC (AHC)	
	Con conexiones laterales	BOLTZMANN/CAUCHY MACHINE	
		CUNTERPROPAGATION	
N capas	Conexiones hacia adelante	BOLZTMANN/CAUCHY MACHINE	
	Conexiones hacia adelante/hacia atrás	BACK-PROPAGATION (BPN)	
		COGNITRON/NEOCOGNITRON	

Fuente: Hilara (2000:74)

Paso 5. Tipo de asociación entre las informaciones de entrada y salida.

Desde un punto de vista computacional, una neurona artificial es un procesador elemental que recibe unos datos, y tras aplicar una función matemática, devuelve otros valores, que a su vez sirven de entradas a otras neuronas. Estas neuronas están interconectadas formando una determinada arquitectura de red neuronal, estructurada en capas de neuronas interconectadas que relacionan la entrada y la salida, Bonsón (1999:44).

Hilera (2000:90), afirma que; “las redes neuronales son sistemas que almacenan cierta información *aprendida*, esta información se registra de forma distribuida en los pesos asociados a las conexiones entre neuronas. Por lo tanto, puede imaginarse una red con cierto tipo de *memoria* que almacena unos datos en forma estable, datos que se grabarán en dicha memoria como consecuencia del aprendizaje de la red y que podrán ser leídos a la salida como respuesta a cierta información de entrada, comportándose entonces la red como lo que habitualmente se conoce por memoria asociativa; es decir, cuando se aplica un estímulo (dato de entrada) la red responde con una salida asociada a dicha información de entrada”

En este paso analizaremos las formas primarias de realizar esta asociación entre la entrada y salida de la información almacenada en la red, dichas formas son: *heteroasociativas* y *autoasociativas*. Una red *heteroasociativa*, es aquella que computa cierta información, que en la mayoría de los casos no podrá expresarse analíticamente, entre un conjunto de entradas y un conjunto de salidas, correspondiendo a cada posible entrada una determinada salida. Por otra parte, una red *autoasociativa* es una red cuya principal misión es reconstruir una determinada información de entrada que se presenta incompleta o distorsionada.

5.1 Redes heteroasociativas

Las redes *heteroasociativas* relacionan informaciones de entrada con diferentes informaciones de salida, cuentan con al menos de 2 capas, una para captar y retener la información de entrada y otra para mantener la salida con la información asociada. Esto es para que no se pierda la información inicial al obtenerse el dato asociado, lo cual no debe ocurrir, ya que el proceso de obtención de la salida se puede necesitar acceder varias veces a esta información, que, por tanto, deberá permanecer en la capa de entrada.

En cuanto a su conectividad, existen redes *heteroasociativas* con conexiones hacia delante o *feedforward* (Perceptron, Backpropagation, etc.), redes con conexiones hacia atrás *feedback* o *feedforward/feedback* (Art, Bam, etc.) y redes con conexiones laterales (Cabam). También existen redes *heteroasociativas* con las neuronas dispuestas en capas multidimensionales (Neocognitron, Topology preserving map). El aprendizaje de este tipo de redes puede ser con supervisión (Perceptron, Backpropagation, etc.) o sin supervisión (Art, Bam, etc.).

Las redes *heteroasociativas* pueden clasificarse según el objetivo pretendido con su utilización. Así, en algunos casos el objetivo es computar una función general de su entrada. En otros casos el objetivo es realizar una clasificación, relacionando (mapeando un gran número de informaciones de entrada con un pequeño número de informaciones de salida, que representan los conjuntos en los que se pueden clasificar los datos de entrada (este es precisamente el caso de estudio de nuestra investigación), En la tabla 8 puede apreciarse una clasificación de los modelos de redes en función de este tipo de asociación.

5.2 Redes autoasociativas

Una red *autoasociativa* asocia información de entrada con el ejemplar más parecido de los almacenados *conocidos* por la red. Este tipo de redes pueden implantarse con una sola capa de neuronas. Esta capa comenzará reteniendo la información inicial presentada a la entrada, y terminará representando la información autoasociada. Si se quiere mantener la información de entrada y salida, se deberán añadir capas adicionales; sin embargo, la funcionalidad de la red puede conseguirse con una sola capa.

En cuanto a la conectividad de este tipo de redes, existen conexiones laterales entre las neuronas (HOPFIELD, etc.) y, en algunos casos, conexiones *autorrecurrentes* (salida de una neurona como entrada de la misma), como ocurre con las redes BRAIN-ESTATE-IN-A-BOX).

Las redes *autoasociativas* suelen utilizarse en tareas de filtrado de información para la reconstrucción de datos, eliminando distorsiones o ruido. También se utilizan para explorar relaciones entre informaciones similares, para facilitar la búsqueda por contenido en bases de datos y para resolver problemas de optimización. En la tabla

8 se muestra una clasificación de redes neuronales en función del tipo de asociación.

Tabla 8. Clasificación de los modelos de redes neuronales en función del tipo de asociación.

REDES HETEROASOCIATIVAS	REDES AUTOASOCIATIVAS
PERCEPTRON	BRAIN-STATE-IN-A-BOX
ADALINE/MADALINE	HOPFIELD
BACKPROPAGATION	OPTIMAL LINEAR ASSOCIATIVE MEMORY
LINEAR REWARD PENALTY	ADDITIVE GROSSBERG
ASSOCIATIVE REWARD PENALTY	SHUNTING GROSSBERG
ADAPTIVE HEURISTIC CRITIC	
BOLZMANN MACHINE	
CAUCHY MACHINE	
LEARNING MATRIX	
TEMPORAL ASSOCIATIVE MEMORY	
LINEAR ASSOCIATIVE MEMORY	
OPTIMAL LINEAR ASSOCIATIVE MEMORY	
DRIVE-REINFORCEMENT	
FUZZY ASSOCIATIVE MEMORY	
DRIVE-REINFORCEMENT	
FUZZY ASSOCIATIVE MEMORY	
COUNTERPROPAGATION	
BIDIRECTIONAL ASSOCIATIVE MEMORY	
ADAPTIVE BIDIRECTIONAL ASSOCIATIVE MEMORY	
COGNITRON/NEOCOGNITRON	
TOPOLOGY PRESERVING MAP	
LEARNING VECTOR QUANTIZER	
ADAPTIVE RESONANCE THEORY	

Fuente: Hilera, Martínez (2000:92)

Paso 6. Datos disponibles y selección de variables relevantes.

6.1 Introducción

En este paso es importante saber la forma en que se van a disponer de los datos. Si contamos con ellos todos a la vez se puede hacer un aprendizaje por lotes (*batch*). En cambio, si los datos van llegando en tiempo real, será preciso emplear aprendizaje en línea (*on line*). También se requiere saber la cantidad de ejemplos que se van a emplear en el entrenamiento de la red.

Respecto a la selección de variables de entrada y salida, hay que tomar en cuenta lo siguiente, según el autor Martín (2002:214):

- Se deben elegir cuidadosamente las variables a emplear.
- Pocas variables o entradas restringen el espacio de búsqueda de los parámetros.
- Muchas variables independientes implican una alta dimensión del espacio de búsqueda, lo cual puede conducir a un error de generalización elevado, pero en este caso por exceso de varianza en la arquitectura.

6.2 Representación de la información de entrada y salida.

6.2.1 Valores reales y continuos

Las redes neuronales pueden también clasificarse en función de la forma en que se representan las informaciones de entrada y las respuestas o datos de salida véase la tabla 8. Así, en un gran número de redes, tanto los datos de entrada como de salida son de naturaleza analógica; es decir, son valores reales continuos, normalmente estarán normalizados y su valor absoluto será menor que la unidad. Cuando esto ocurre, las funciones de activación de las neuronas serán también continuas, del tipo lineal o sigmoidal.

6.2.2 Valores discretos

Otras redes, por el contrario, sólo admiten valores discretos o binarios $\{0,1\}$ en su entrada, generando también unas respuestas en la salida de tipo binario. En este caso, las funciones de activación de las neuronas serán del tipo escalón (véase la tabla 9). Existe también un tipo de redes híbridas, en la que las informaciones de entrada pueden ser valores continuos, aunque las salidas de la red son discretas.

Tabla 9. Clasificación de los modelos de redes neuronales en función del tipo de representación de las informaciones de entrada y salida.

REDES CONTINUAS Entrada analógica Salida analógica	REDES HÍBRIDAS Entrada analógica Salida binaria	REDES DISCRETAS Entrada binaria Salida binaria
BACKPROPAGATION	PERCEPTRON	DISCRETE HOPFIELD
BAIN-STATE-IN-A-BOX	ADALINE/MADALINE	LEARNING MATRIX
CONTINUOS	LINEAR/ASSOCIATIVE	TEMPORAL ASSOCIATIVE

REDES CONTINUAS Entrada analógica Salida analógica	REDES HÍBRIDAS Entrada analógica Salida binaria	REDES DISCRETAS Entrada binaria Salida binaria
HOPFIELD	REWARD PENALTY	MEMORY
LINEAL ASSOCIATIVE MEMORY	ADAPTIVE HEURISTIC CRITIC	BIDIRECTIONAL ASSOCIATIVE MEMORY
OPTIMAL LINEAR ASSOCIATIVA MEMORY		CONGNITRON/NEOCOGNITRON
DRIVE REINFORCEMENT		ADAPTIVE RESONANCE THEORY 1
COUNTERPROPAGATION		BOLTZMAN MACHINE
ADDITIVE GROSSBERG		CAUCHY MACHINE
ADAPTIVE BAM		
LEARNING VECTOR QUANTIZER		
TOPOLOGY PRESERVING MAP		
ADAPTIVE RESONANCE THEORY 2		

Fuente: Hilera, Martínez (2000:94-95)

6.3 Características de los modelos de redes neuronales más conocidos

En la tabla 10, se muestran por orden alfabético los modelos de red neuronal más conocidos, según la clasificación desarrollada por Hilera, Martínez (2000,96-100) y utilizada en esta metodología: topología, mecanismo de aprendizaje, tipo de asociación de las informaciones de entrada y salida y la forma de representación de esta información, señales de entrada y salida, y autor(es).

Tabla 10. Resumen de las características de los modelos de redes más conocidos.

MODELO DE RED	TOPOLOGÍA	APRENDIZAJE			ASOCIACIÓN AUTO/HETERO	INF. DE ENTRADA Y SALIDA	Autor (es)
		ON/OFF LINE	SUPERVISADO/NO SUPERVISADO	Regla			
ADALINE/MADALINE	2 CAPAS Feedforward	OFF	Supervisado	Corrección Error ("Least Mean Square")	Heteroasociativa	E: Analógica S: Binaria	Widrow Hoff 1960
ADAPTIVE BIDIRECTIONAL ASSOCIATIVE MEMORY, ABAM	2 CAPAS Feedforward/ Feedback	ON	No Supervisado	Hebbiano Diferencial	Heteroasociativa	Analógica	Kosko 1987
ADAPTIVE	3 CAPAS	ON	Supervisado	Refuerzo por	Heteroasociativa	E: Analógica	Barto 1983

MODELO DE RED	TOPOLOGÍA	APRENDIZAJE			ASOCIACIÓN AUTO/HETERO	INF. DE ENTRADA Y SALIDA	Autor (es)
		ON/OFF LINE	SUPERVISADO/NO SUPERVISADO	Regla			
HEURISTIC CRITIC. AHC	Feedforward			ajuste temporal		S: Binaria	
ADAPTIVE RESONANCE THEORY 1. ART1	2 CAPAS Feedforward/ feedback, conexiones laterales auto-recurrentes	ON	No Supervisado	Competitivo (Resonancia Adaptativa)	Heteroasociativa	Binarias	Carpenter Grossberg 1986
ADAPTIVE RESONANCE THEORY 2. ART2	2 CAPAS Feedforward/ feedback, conexiones laterales auto-recurrentes	ON	No Supervisado	Competitivo (Resonancia Adaptativa)	Heteroasociativa	Analógica	Carpenter Grossberg 1987
ADITIVE GROSSBERG. AG	1 CAPA conexiones laterales auto-recurrentes	ON	No Supervisado	Hebbiano o competitivo	Autoasociativa	Analógica	Grossberg 1968
ASSOCIATIVE REWARD PENALTY. ARP	2 CAPAS Feedforward	ON	Supervisado	Refuerzo Estocástico	Heteroasociativa	E: Analógica S: Binaria	Barto 1985
BACK-PROPAGATION	N CAPAS Feedforward	OFF	Supervisado	Corrección Error (Regla Delta Generalizada)	Heteroasociativa	Analógica	Rumelhart etc. 1986
RECURRENT BACKPROPAGATION	N CAPAS Feedforward/ Feedback	OFF	Supervisado	Corrección Error (Regla Delta Generalizada)	Heteroasociativa	Analógica	Pineda etc. 1987
BIDIRECTIONAL ASSOCIATIVE MEMORY BAM	2 CAPAS Feedforward/ Feedback	ON	No Supervisado	Hebbiano	Heteroasociativa	Binarias	Kosko 1988
BOLTZMAN MACHINE, BM	1 CAPA Conexiones laterales 3 CAPAS Feedforward	OFF	Supervisado	Estocástico ("Simulated Annealing") + Hebbiano o + Corrección Error	Heteroasociativa	Binarias	Hinton Ackley Sejnowsky 1984
BAIN-ESTATE-IN-A-BOX	1 CAPA Conexiones laterales auto-recurrentes	OFF	Supervisado	Corrección Error (Regla Delta Generalizada)	Autoasociativa	Analógica	Anderson 1977
CAUCHY MACHINE, CM	1 CAPA Conexiones laterales 3 CAPAS Feedforward	OFF	No Supervisado	Estocástico ("Fast Simulated Annealing")	Heteroasociativa	E: Analógica S: Binaria	Szu 1986
COGNITRON/NEOCOGNITRON	Jerarquía de niveles con capas bidimensionales Feedforward/ Feedback	OFF	No Supervisado	Competitivo	Heteroasociativa	Binarias	Fukushima 1975/1980
COMPETITIVE ABAM, CABAM	2 CAPAS Feedforward/ feedback, conexiones laterales auto-recurrentes	ON/OFF	No Supervisado	Hebbiano + competitivo	Heteroasociativa	Analógica	Kosko 1987

MODELO DE RED	TOPOLOGÍA	APRENDIZAJE			ASOCIACIÓN AUTO/HETERO	INF. DE ENTRADA Y SALIDA	Autor (es)
		ON/OFF LINE	SUPERVISADO/NO SUPERVISADO	Regla			
COUNTER-PROPAGATION	3 CAPAS Feedforward Conexiones laterales auto-recurrentes	OFF	Supervisado	Corrección Error + competitivo	Heteroasociativa	Analógica	Hecht-Niel-Sen87
DRIVE-REINFORCEMENT	2 CAPAS Feedforward	OFF	No Supervisado	Hebbiano (Drive Reinforcement)	Heteroasociativa	Analógica	Klopf 1986
FUZZY ASSOCIATIVE MEMORY, FAM	2 CAPAS Feedforward/ feedback,	OFF	No Supervisado	Hebbiano Borroso	Heteroasociativa	Analógica	Kosko 1987
CONTINUOS HOPFIELD	1 CAPA Conexiones laterales	OFF	No Supervisado	Hebbiano	Autoasociativa	Analógica	Hopfield 1984
DISCRETE HOPFIELD	1 CAPA Conexiones laterales	OFF	No Supervisado	Hebbiano	Autoasociativa	Binarias	Hopfield 1982
LEARNING MATRIX LM	1 CAPA Cossbar	OFF	No Supervisado	Hebbiano	Heteroasociativa	Binarias	Steinbuch 1961
LEARNING VECTOR QUANTIZER LVQ	2 CAPAS Feedforward/ feedback, conexiones laterales auto-recurrentes	OFF	No Supervisado	Competitivo	Heteroasociativa	Analógica	Kohonen 1981
LINEAR ASSOCIATIVE MEMORY, LAM	2 CAPAS Feedforward	OFF	No Supervisado	Hebbiano	Heteroasociativa	Analógica	Anderson 1968 Kohonen 77
LINEAR REWARD PENALTY, LRP	2 CAPAS Feedforward	ON	Supervisado	Refuerzo Estocástico	Heteroasociativa	E: Analógica S: Binaria	Barto 1985
OPTIMAL LINEAR ASSOCIATIVE MEMORY, OLAM	2 CAPAS Feedforward 1 CAPA Conexiones laterales auto-recurrentes	OFF	No Supervisado	Hebbiano (Optimal Least Mean Square correlation")	Autoasociativa ----- Heteroasociativa	Analógica	Wee 1968 Kohonen 73
PERCEPTRON	2 CAPAS Feedforward	OFF	Supervisado	Corrección error	Heteroasociativa	E: Analógica S: Binaria	Rosenblat 1958
SHUNTING GROSSBERG, SG	1 CAPA Conexiones laterales auto-recurrentes	ON	No Supervisado	Hebbiano o competitivo	Autoasociativa	Analógica	Grossberg 1973
SPARSE DISTRIBUTED MEMORY, SDM	3 CAPAS Feedforward/ Feedback,	OFF	No Supervisado	Hebbiano + Random Vector (LVQ) Preprocessing	Heteroasociativa	Binarias	Kanerva 1984
TEMPORAL ASSOCIAT. MEMORY, TAM	2 CAPAS Feedforward/ Feedback,	OFF	No Supervisado	Hebbiano	Heteroasociativa	Binarias	Amari 1972
TOPOLOGY PRESERVING MAP, TPM	2 CAPAS Feedforward/ feedback, conexiones laterales auto-recurrentes	OFF	No Supervisado	Competitivo	Heteroasociativa	Analógica	Kohonen 1982

Fuente: Hilera, Martínez (2000:100)

Paso 7. Mecanismo de aprendizaje

El aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. Los cambios que se producen durante el proceso de aprendizaje se reducen a la destrucción, modificación y creación de conexiones entre neuronas. En los sistemas biológicos existe una continua creación y destrucción de conexiones. En los modelos de redes neuronales artificiales, la creación de una nueva conexión implica que el peso de la misma pasa a tener un valor distinto de cero. De la misma forma, una conexión se destruye cuando su peso se convierte en cero.

Durante el proceso de aprendizaje, los pesos de las conexiones de la red sufren modificaciones, por lo tanto se puede afirmar que este proceso ha terminado (la red ha *aprendido*) cuando los valores de los pesos permanecen estables, es decir cuando $\frac{dw_{ij}}{dt} = 0$.

Un aspecto importante respecto al aprendizaje de las redes neuronales es conocer cómo se modifican los valores de los pesos; es decir, cuáles son los criterios que se siguen para cambiar el valor asignado a las conexiones cuando se pretende que la red *aprenda* una nueva información.

Estos criterios determinan lo que se conoce como la regla de aprendizaje de la red. De forma general, se suelen considerar dos tipos de reglas: las que responden a lo que habitualmente se conoce como aprendizaje supervisado y las correspondientes a un aprendizaje no supervisado.

Es por ello por lo que una de las clasificaciones que se realizan de las redes neuronales obedece al tipo de aprendizaje utilizado por dichas redes. Así se pueden distinguir:

- Redes neuronales con aprendizaje supervisado
- Redes neuronales con aprendizaje no supervisado

La diferencia fundamental entre ambos tipos estriba en la existencia o no de un agente externo (supervisor) que controle el proceso de aprendizaje de la red.

Otro criterio que se puede utilizar para diferenciar las reglas de aprendizaje se basa en considerar si la red puede *aprender* durante su funcionamiento habitual o si el aprendizaje supone la *desconexión* de la red; es decir su inhabilitación hasta que el proceso termine. En el

primer caso, se trataría de un aprendizaje ON LINE, mientras que en el segundo es lo que se conoce como OFF LINE.

Cuando el aprendizaje es OFF LINE, se distingue entre una fase de aprendizaje o entrenamiento y una fase de operación o funcionamiento, existiendo un conjunto de datos de entrenamiento y un conjunto de datos de *test* o prueba que serán utilizados en la correspondiente fase. En las redes con aprendizaje OFF LINE, los pesos de las conexiones permanecen fijos después que termina la etapa de entrenamiento de la red. Debido precisamente a su carácter estático, estos sistemas no presentan problemas de estabilidad en su funcionamiento.

En las redes con aprendizaje ON LINE no se distingue entre fase de entrenamiento y operación, de tal forma que los pesos varían dinámicamente siempre que se presente una información al sistema. En este tipo de redes, debido al carácter dinámico de las mismas, el estudio de estabilidad suele ser un aspecto fundamental de estudio.

7.1 Redes con aprendizaje supervisado

El aprendizaje supervisado se caracteriza porque el proceso de aprendizaje se realiza mediante un entrenamiento controlado por un agente externo (supervisor, maestro) que determina la respuesta que debería generar la red a partir de una entrada determinada. El supervisor comprueba la salida de la red y en el caso en que ésta no coincida con la deseada, se procederá a modificar los pesos de las conexiones, con el fin de conseguir que la salida obtenida se aproxime a la deseada, (véase figura 44).

Para llevar a cabo este aprendizaje se consideran tres formas que dan lugar a los siguientes *aprendizajes supervisados*: (véase tabla 11)

- **Aprendizaje por corrección de error**
- **Aprendizaje por refuerzo**
- **Aprendizaje estocástico**

1.1.1 7.1.1 Aprendizaje por corrección de error

Consiste en ajustar los pesos de las conexiones de la red en función de la diferencia entre los valores deseados y los obtenidos en la salida de la red; es decir, en función del error cometido en la salida.

Una regla de o algoritmo simple de aprendizaje por corrección de error podría ser:

$$\Delta w_{ji} = \alpha y_i (d_j - y_j)$$

Δw_{ji} , es la variación en el proceso de la conexión entre las neuronas i

y j , o sea ($\Delta w_{ji} = w_{ji}^{actual} - w_{ji}^{anterior}$).

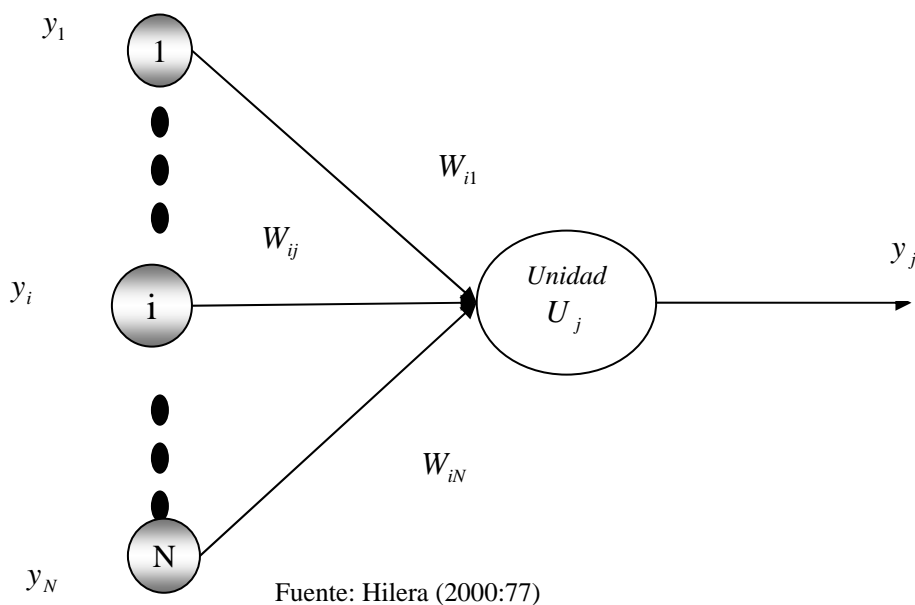
y_i , es el valor de salida de la neurona i .

d_j , es el valor de salida deseado para la neurona j .

y_j , es el valor de salida obtenido en la neurona j .

α , es el factor de aprendizaje ($0 < \alpha \leq 1$) que regula la velocidad del aprendizaje.

Figura 44. Red neuronal con aprendizaje supervisado



Un ejemplo de este tipo de algoritmo lo constituye la *regla de aprendizaje del Perceptron*, utilizada en el entrenamiento de la red del mismo nombre que desarrolló Roseblatt en 1958 (Roseblatt 58). Sin embargo, existen otros algoritmos más evolucionados que éste, que presenta algunas limitaciones, como el no considerar la magnitud del error global cometido durante el proceso completo de aprendizaje de la red, considerando únicamente los errores individuales (locales) correspondientes al aprendizaje de cada información por separado.

Un algoritmo muy conocido que mejora el del Perceptron y permite un aprendizaje más rápido y un campo de aplicación más amplio es

el propuesto por Widrow y Hoff en 1960 (Widrow 60), denominado regla delta o regla del mínimo error cuadrado (LMS Error: Least-Mean-Squared Error²), ésta se aplicó en las redes desarrolladas por los mismos autores, conocidas como ADALINE (Adaptative Linear Element), con una única neurona de salida, y MADALINE (Múltiple ADALINE), con varias neuronas de salida.

Widrow y Hoff definieron una función que permitía cuantificar el error global cometido en cualquier momento durante el proceso de entrenamiento de la red, lo cual es importante, ya que cuanta más información se tenga sobre el error cometido, más rápido se puede aprender.

Este error medio se expresa de la siguiente forma:

$$Error_{global} = \frac{1}{2P} \sum_{k=1}^P \sum_{j=1}^N (y_j^{(k)} - d_j^{(k)})^2$$

De donde:

N: Número de neuronas de salida (en el caso de ADALINE N = 1).

P: Número de informaciones que debe aprender de la red.

$\frac{1}{2P} \sum_{j=1}^N (y_j^k - d_j^k)^2$: Error cometido en el aprendizaje de la información k -ésima.

Por lo tanto de lo que se trata es de encontrar unos pesos para las conexiones de la red que minimicen esta función de error. Para ello, el ajuste de los pesos de las conexiones de la red se puede hacer de forma proporcional a la variación relativa del error que se obtiene al variar el peso correspondiente:

$$\Delta w_{ji} = k \frac{\partial Error_{global}}{\partial w_{ji}}$$

Mediante este procedimiento, se llegan a obtener un conjunto de pesos con los que se consigue minimizar el error medio.

Otro algoritmo de aprendizaje por corrección de error lo constituye la *regla delta generalizada* o algoritmo de retropropagación del error (*error backpropagation*), también conocido como regla LMS (*por sus siglas en inglés Least-Mean-Square error*) multicapa, se trata de una generalización de la regla delta para poder aplicarla a redes con conexiones hacia delante (*feedforward*) con capas o

² También es conocida como regla de Widrow-Hoff

niveles internos y ocultos de neuronas que no tienen relación con el exterior. Son redes con capa de entrada, capas ocultas y capa de salida. Este algoritmo de aprendizaje se explica más ampliamente en el capítulo 3 inciso 3.12 de este trabajo.

Las redes multicapa que utilizan este proceso de aprendizaje son Perceptron, ADALINE y MADALINE. Este proceso también puede aplicarse a otras pero su proceso de aprendizaje es mucho más lento.

1.1.2 7.1.2 Aprendizaje por refuerzo

El aprendizaje por refuerzo en la primera fase es sin supervisión y se refinan los valores de los pesos de las conexiones mediante un aprendizaje por corrección de error basado en la adaptación de la *regla delta* de Widrow-Hoff. Este es un aprendizaje supervisado, más lento que el anterior, se basa en no indicar durante el entrenamiento exactamente la salida que se desea que proporcione la red ante una determinada entrada.

1.1.3 7.1.3 Aprendizaje estocástico

Este tipo de aprendizaje consiste en hacer cambios en forma aleatoria a los valores de los pesos de las conexiones de la red y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidad, Hilera (2000:81).

En la tabla 11 podemos apreciar los modelos de redes más conocidos que manejan aprendizaje supervisado.

Tabla 11. Modelos de redes más conocidos con aprendizaje supervisado.

TIPO DE APRENDIZAJE SUPERVISADO		MODELO DE RED
Aprendizaje por corrección de error	OFF LINE	PERCEPTRON
		ADALINE/MADALINE
		BACKPROPAGATION
		BRAIN-STATE-IN-A-BOX
		COUNTERPROPAGATION
Aprendizaje por refuerzo	ON LINE	LINEAR REWARD PENALTY
		ASSOCIATIVA REWARD PENALTY
		ADAPTIVE HEURISTIC CRITIC
Aprendizaje	OFF LINE	BOLTZMANN MACHINE

estocástico		CAUCHY MACHINE
-------------	--	----------------

Fuente: Hilera, Martínez (2000:82)

7.2 Redes con aprendizaje no supervisado

Las redes neuronales artificiales con aprendizaje no supervisado son aquellas que no necesitan de un *profesor o supervisor* externo para realizar su aprendizaje, ya que son capaces de modificar sus pesos de las conexiones entre sus neuronas internamente, adaptándose al entorno de la mejor manera posible; es decir, la red descubre por sí sola características, regularidades, correlaciones o categorías en los datos de entrada, esto genera en la salida una versión codificada de la entrada, manteniendo la información relevante de la entrada. Viñuela [2004:123].

En el aprendizaje no supervisado, las redes no requieren influencia externa para ajustar los pesos de las conexiones entre sus neuronas, es decir la red no recibe ninguna información que le indique si la salida generada en respuesta a una determinada entrada es o no correcta; por lo cual estas redes son capaces de *autoorganizarse*. Hilera, Martínez (2000:82).

Es recomendable utilizar el aprendizaje no supervisado sólo cuando en los datos de entrada existe cierto tipo de redundancia.

Una red con aprendizaje no supervisado puede detectar patrones que van de acuerdo a su arquitectura. Qué tipo de proceso puede realizar la red y qué problemas puede resolver, depende de las posibilidades siguientes, Viñuela [2004:123].

a) Familiaridad.

Es el caso en el que mediante un valor continuo de salida se pondera el grado de similitud entre la entrada y un *valor tipo* o media de una serie de valores presentados con anterioridad.

b) Análisis de las componentes principales.

Se trata de detectar cuáles de las componentes del conjunto de entrada caracterizan en mayor grado al conjunto de datos, de forma que los demás pudieran eliminarse sin una significativa pérdida de información.

c) Agrupamiento.

Se divide el conjunto de datos en diferentes clases, y se decide a qué clase pertenece cada dato y qué caracteriza a cada una de las clases.

d) Prototipado.

Igual que en caso anterior, pero en vez de obtenerse como resultado a qué clase pertenece el dato de entrada, se obtiene un prototipo o ejemplar de la clase a que pertenece dicho dato de entrada.

e) Codificación.

Se obtiene a la salida de la red, una versión codificada del dato de entrada, es decir, un dato de menor dimensión que mantenga el máximo de información que le sea posible. Esto puede utilizarse para la compresión de datos antes de transmitirse por un canal de un ancho de banda limitado, por el cual la señal original no podría ser transmitida; siempre y cuando exista la posibilidad de construir una red capaz de restaurar el dato original.

f) Extracción y relación de las características.

Aquí se lleva a cabo un mapa topológico de los datos de entrada, de tal forma que los patrones de entrada parecidos, produzcan respuestas similares en células cercanas.

Los ejemplos anteriores se pueden combinar de varias maneras. Por ejemplo, se puede utilizar el método de componentes principales para reducir la dimensión de los datos, como paso previo al *clustering*.

En cuanto a los algoritmos de aprendizaje no supervisado, se consideran dos tipos:

- *Aprendizaje Hebbiano*
- *Aprendizaje competitivo y cooperativo*

En el primer caso, se pretende medir la familiaridad o extraer características de los datos de entrada, mientras que en el segundo suele orientarse hacia la clasificación de dichos datos.

7.2.1 Aprendizaje Hebbiano

El aprendizaje Hebbiano, Hebb, (Donald O. Hebb, 1949) consiste básicamente en el ajuste de los pesos de las conexiones de acuerdo con

la correlación (multiplicación en caso de valores binarios +1 y -1) de los valores de activación de las dos neuronas conectadas:

$$\Delta w_{ij} = y_i y_j$$

La expresión de Hebb, muestra que si las dos unidades son positivas, se produce un reforzamiento de la conexión. Pero si por el contrario, una es activa (positiva) y la otra pasiva (negativa), se produce un debilitamiento de la conexión. Como vemos se trata de una regla de aprendizaje no supervisado, pues la modificación de los pesos se realiza en función de los estados (salidas) de las neuronas obtenidos tras la presentación de cierto estímulo (información de entrada a la red), sin tener en cuenta si se desea obtener o no esos estados de activación.

Muchos autores han utilizado este tipo de aprendizaje, entre los más conocidos se encuentran Hopfield (1982:2554) con su red llamada RED HOPFIELD y Grossberg, (1968:368-372) con su red ADDITIVE GROSSBERG. En la tabla XX presenta el autor Hilera (2000:82), diferentes tipos de redes con aprendizaje no supervisado.

7.2.2 Aprendizaje competitivo y cooperativo

En este tipo de redes, se dice que las neuronas compiten (y cooperan) unas con otras con el fin de llevar a cabo una tarea dada. Con este aprendizaje, cuando se presenta a la red cierta información de entrada, sólo una de las neuronas de salida de la red, o una por cierto grupo de neuronas, se activa. Por lo que las neuronas compiten por activarse, quedando una, o una por grupo, como neurona *vencedora* (*winner-take-all-unit*), quedando anuladas, el resto, las cuales son forzadas por sus valores de respuesta mínimos.

Las clases o categorías en este tipo de aprendizaje deben ser creadas por la propia red, ya que se trata de un aprendizaje no supervisado, a través de las correlaciones entre los datos de entrada. Se forman categorías (clusters) de los datos similares. En la tabla 11, vemos la clasificación de las redes neuronales con *aprendizaje no supervisado* más importantes.

Tabla 12. Redes con aprendizaje no supervisado más conocidas

Tipo de aprendizaje no supervisado		Modelo de red
APRENDIZAJE HEBBIANO	OFF LINE	HOPFIELD
		LEARNING MATRIX
		TEMPORAL ASSOCIATIVE MEMORY
		LINEAR ASSOCIATIVE MEMORY (LAM)
		OPTIMAL LAM

Tipo de aprendizaje no supervisado		Modelo de red
		DRIVE-REINFORCEMENT
		FUZZY ASSOCIATIVE MEMORY
	ON LINE	ADDITIVE GROSSBERG
		SHUTING GROSSBERG
		BIDIRECTIONAL ASSOCIATIVE MEMORY (BAM)
		ADAPTIVE BAM
APRENDIZAJE COMPETITIVO/COOPERATIVO	OFF LINE	LEARNING VECTOR QUATIZER
		COGNITRON/NEOCOGNITRON
		TOPOLOGY PRESERVING MAP
	ON	ADAPTIVE RESONANCE THEORY

Fuente: Hilera, Martínez (2000:89)

7.3 Elección de los conjuntos de aprendizaje y prueba

La condición que se debe imponer a los ejemplos (patrones) que se presentan a la red neuronal artificial para que ese lleve a cabo el entrenamiento es que sean un número suficiente como para ser representativos del fenómeno que queremos modelar, Martín y Sanz, (2002:314).

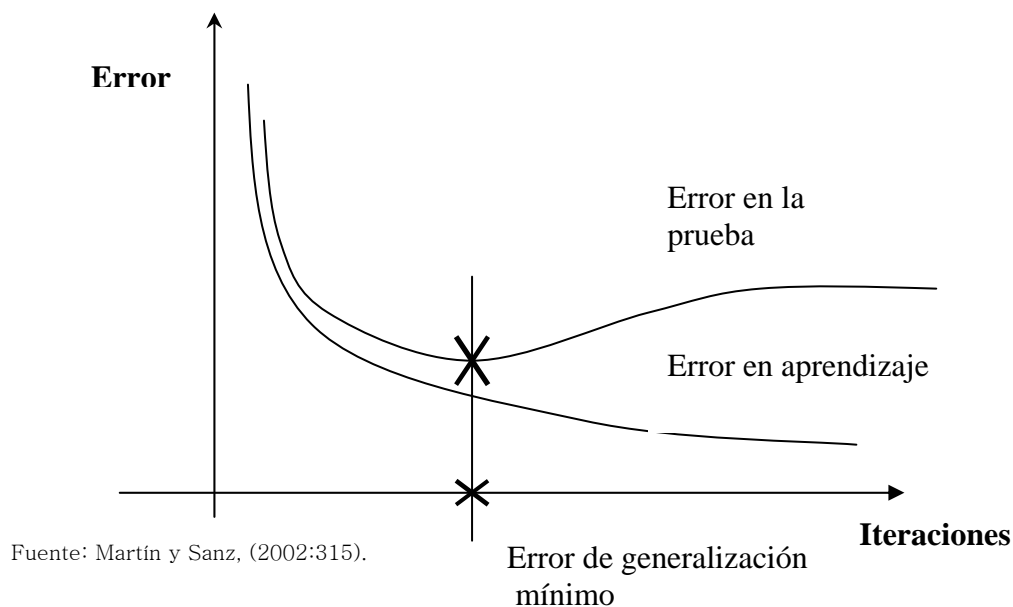
En el caso de que se use el modelo del Perceptron Multicapa se suele emplear una técnica llamada de *validación cruzada*, la cual consisten en dividir los ejemplos (patrones) en dos grupos: uno de aprendizaje y otro para prueba. De esta forma se procederá al entrenamiento de la red empleando solamente los ejemplos del conjunto de aprendizaje, mientras que se comprobará de vez en cuando el error que comete la red al ser aplicado sobre los casos del de prueba. Martín y Sanz, (2002:315).

En la fase inicial del entrenamiento, la red extrae los rasgos generales del conjunto de aprendizaje. Dado que el conjunto de prueba debe tener una información similar al conjunto de aprendizaje, a medida que transcurren las iteraciones se observa que los errores de los dos conjuntos van disminuyendo. El error del conjunto de aprendizaje siempre disminuye y rara vez empeorará. En cambio, si se observa que el error del conjunto de prueba alcanza un valor mínimo y después empeora (Véase la figura 45). A partir de ese momento, la red está memorizando los casos del conjunto de aprendizaje (incluso se está alejando de la tónica general, es decir, está perdiendo capacidad de generalizar (de extrapolar al conjunto de prueba las características del de aprendizaje). Esto se denomina **sobre-ajuste o sobreentrenamiento**, que debe evitarse puesto que se obtendrán malos resultados en la futura aplicación de la

red sobre casos nuevos. Detener el entrenamiento de la red cuando comienza a empeorar el error de la prueba constituye la técnica denominada *validación cruzada*. Martín y Sanz, (2002:315).

Los autores Martín y Sanz, nos recuerdan finalmente que los conjuntos de aprendizaje y de prueba, deben ser por separado representativos de lo que sucede en nuestro problema y que deben cubrir en lo posible los espacios de entrada y de salida, abarcando todas las posibilidades que puedan darse y que hay que aprender.

Figura 45. Evolución del error de aprendizaje y del error de prueba en función de las iteraciones, en un proceso de aprendizaje



Paso 8. Preprocesamiento

En este paso es aconsejable (sin ser imprescindible) darles un tratamiento previo a los datos de entrada y salida para adecuarlos a la red neuronal, a esto se le llama *preprocesamiento*, los datos que se proporcionen a la red deben poseer las siguientes cualidades, Martín y Sanz, (2002:316):

- Buena distribución (distribución estándar o uniforme).
- Rangos de valores parecidos para todas las entradas.

- Rangos acotados dentro del intervalo de trabajo de la función de activación empleada en las capas ocultas y de salida de la RNA.

Si se desea cambiar el tipo de **distribución de los datos**, se requiere aplicar una **transformación**; las transformaciones más frecuentes se basan en el empleo de logaritmos, raíces cuadradas y cúbicas.

Para modificar el **rango de valores** de las entradas se aplica un escalado. El objetivo es acotar sus valores entre $[0,+1]$ o $[-1,+1]$, límites de la función de activación, para evitar la saturación de ésta al realizar la suma ponderada de las entradas. Hay dos formas de obtener los factores de escalado para aplicar la transformación del rango:

8.1 Escalado por patrones (por filas)

En este escalado, se busca en los ejemplos de cada entrada, el valor máximo y mínimo y se les asigna 1 o 0 respectivamente, y escalamos en función de ellos los restantes. Con este esquema cada ejemplo tendrá forzosamente al menos un uno, un cero y valores intermedios.

8.2 Escalado global por entradas (por columnas)

Este escalado es relacionado con los valores de cierta entrada para todos los ejemplos (es parecido al anterior, pero fijándose en los valores de cierta entrada). En este caso tendremos ejemplos sin ceros ni unos, puede aplicarse dos transformaciones consecutivas:

- 1) Primero se realiza un *estandarizado*, es decir, se transforman las variables de forma que presenten media cero y varianza uno. Para lo cual se aplica la función de estandarización con cierta entrada.

$$x' = \frac{x - \bar{x}}{\sigma}$$

- 2) A continuación, pueden escalarse los nuevos valores al intervalo $[0,+1]$ o $[-1,+1]$ mediante una transformación lineal.

Gracias al escalado conseguimos que todas las entradas tengan la misma relevancia al igualar su variabilidad (gracias al estandarizado) y su rango de valores, sin importar el rango y distribución inicial.

También se manejan **otros tipos de preprocesamiento** más complejos, como la transformada de Fourier, análisis de componentes principales o técnicas de extracción de rangos (*feature extraction*). Algo que resulta sencillo y da muy buenos resultados es cambiar las entradas por otras que sean cocientes de ciertas variables (ratios).

Recomiendan los autores Martín y Sanz (2002:217), que lo comentado para las entradas puede aplicarse también a las salidas de la red, adecuando su distribución, viabilidad y rangos de valores.

Paso 9. Proceso de entrenamiento

Este es el paso determinante de todo el desarrollo de la aplicación. En este proceso existe una interacción entre los dos conjuntos de patrones (aprendizaje y prueba), la estructura de la RNA y el experimentador. Primero se debe ensayar con diferentes topologías (véase tabla 9) y con distintos parámetros de aprendizaje, midiendo el error de aprendizaje y el de generalización, hasta alcanzar los deseados. En el caso de no obtener los resultados aceptables, habrá que volver a revisar los pasos anteriores: puede ser que los conjuntos de aprendizaje y pruebas no sean representativos, que no se hayan elegido bien las variables, o que éstas no hayan sido preprocesadas o escaladas adecuadamente. Puede suceder también que no se haya elegido el modelo de red más apropiado o que su resolución con RNA's no sea viable.

Los parámetros de la red que pueden modificarse en los experimentos son los siguientes: pesos iniciales, ritmo de aprendizaje, número de neuronas ocultas y parada de entrenamiento.

9.1. Inicialización de pesos.

El método más utilizado es el de realizar una inicialización aleatoria en cierto intervalo. Existen otros enfoques de inicialización como la minimización de funciones a esta elección de pesos iniciales, con técnicas de regresión lineal y algoritmos genéticos (Masters, 93:120).

9.2. Ritmo de aprendizaje

El ritmo de aprendizaje controla el tamaño de los cambios en los pesos de las neuronas. Este ritmo toma valores muy dispares, puede ser desde 0.5 hasta 0.00001 y todavía menores, una de las razones de esta variabilidad es porque en el aprendizaje por lotes (*batch*), al acumular el cambio en los pesos propuesto por cada patrón, se obtiene un cambio mayor, cuantos más patrones se tengan en el conjunto de aprendizaje. Si elegimos un ritmo de aprendizaje pequeño implica que la RNA realizará cambios pequeños en sus pesos, lo cual es perjudicial por dos motivos: evita escapar de mínimos locales y disminuye la velocidad de

convergencia. Por otro lado, si elegimos un valor alto, puede ocasionar grandes variaciones en los pesos, y por lo tanto, ocasionar inestabilidad en la RNA o saturar sus neuronas. Se recomienda emplear el mayor ritmo de aprendizaje que no provoque excesivas oscilaciones en los errores mostrados en la RNA. (Martín, 2002:218), Schalkoff (1997,86)

9.3. Neuronas ocultas

No existe una técnica que indique el número de neuronas ocultas para resolver un problema dado. Pero sabemos que basta con una capa oculta para resolver cualquier problema de aproximación funcional como un Perceptron. Según estudios de (Funahashi, 89), para decidir el número de neuronas ocultas se pueden seguir las siguientes técnicas.

9.3.1 Recetas

9.3.2 Prueba y error

9.3.3 Métodos dinámicos

9.3.4 Parada del entrenamiento

Paso 10. Evaluación de los resultados

Finalizada la fase de entrenamiento y almacenados los pesos ideales, ya se puede aplicar la RNA sobre casos nuevos (no empleados en el entrenamiento) para medir su eficacia en forma completamente objetiva. Si se comprueba que se siguen obteniendo resultados dentro del margen de error deseado, se puede proceder a emplear la RNA dentro de su entorno de trabajo real.

Capítulo 5. Caso de estudio: el caso de clasificación de tutores en el Programa de Posgrado en Ciencias de la Administración de la UNAM.

Introducción

El problema a resolver en este caso de estudio es conocido en redes neuronales como de generalización; es decir, la red neuronal se entrena con entradas y se realiza una prueba con casos diferentes. Dentro de los problemas típicos de generalización se encuentran los de clasificación y predicción. En este trabajo nos interesa llevar a cabo la clasificación de tutores primeramente (el cual llamaremos *conjunto de entrenamiento*) y luego la predicción de tutores candidatos (*conjunto de prueba o validación*); una vez que se han clasificado y pronosticado los tutores, se creará una base de datos que alimentará un sistema de información, el cual nos permitirá asignarlos adecuadamente a los alumnos. En este trabajo de investigación nos interesa lograr solo la clasificación para lo cual se seguirá la metodología mostrada anteriormente en la figura 33.

La metodología que se seguirá para realizar este estudio será mediante el desarrollo de una RNA a través de los pasos que se muestran en la figura 46, los cuales se derivan de la metodología mostrada anteriormente en la figura 33, dicha metodología en forma automatizada, se incluye en un CD al final de este trabajo de investigación, con la finalidad de facilitar que otros investigadores puedan utilizar los modelos de redes neuronales.

Figura 46. Metodología para diseñar la arquitectura de la red neuronal

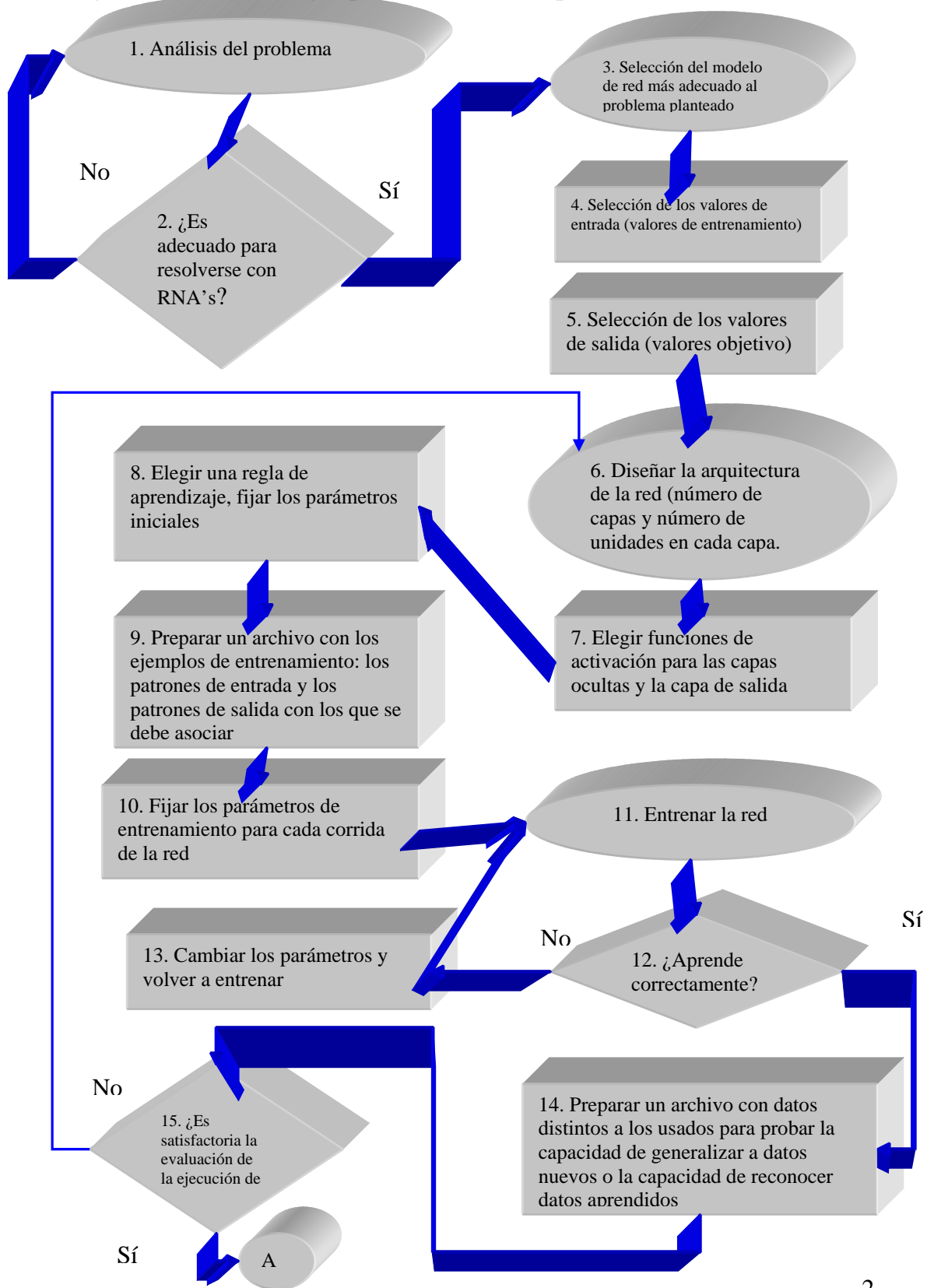
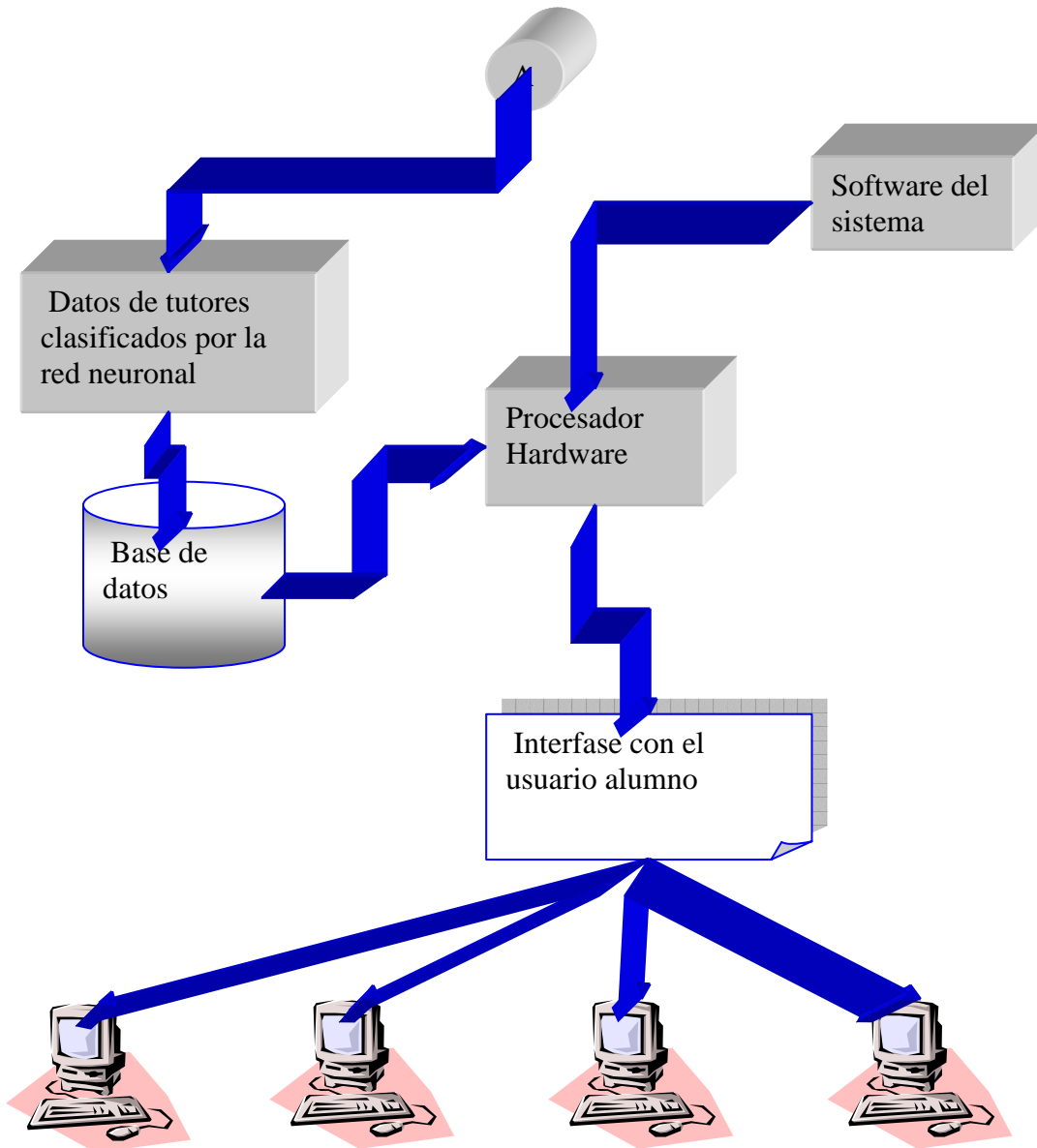


Figura 47. Sistema neuronal artificial



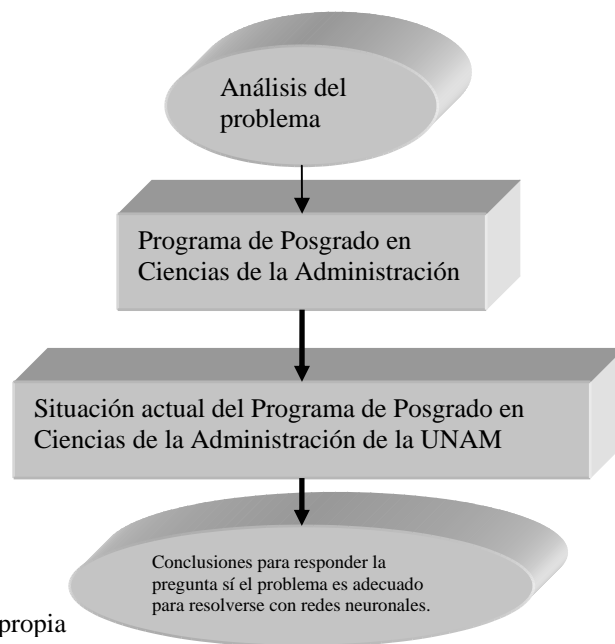
I. Planteamiento del problema

Se siguieron los pasos descritos en las metodologías propuestas en el capítulo y el inciso anteriores para realizar el diseño de la arquitectura de la red neuronal, (Véase las figuras 32 y 46).

1. Análisis del problema.

Para analizar el problema revisamos el capítulo 2 para conocer en qué consiste el Programa de Posgrado en Ciencias de la Administración de la UNAM, el Sistema Tutorial del Programa de Doctorado, y cuál es la relación entre ambos; esto nos permitirá saber si el problema es adecuado para resolverse con redes neuronales artificiales, (véase figura 48).

Gráfica 48. Pasos para realizar el análisis del problema



Fuente: Aportación propia

Los logros más importantes alcanzados en el Programa de Posgrado en Ciencias de la Administración en el periodo 2003-2005, según señala el Dr. Ricardo Varela, Coordinador del Programa de Posgrado en Ciencias de la Administración de la UNAM, en su Informe de Actividades del mismo periodo, han sido; “1. Ingreso del Doctorado en Ciencias de la Administración al Programa Integral de Fortalecimiento al Posgrado-CONACYT, 2. Aumento significativo en la eficiencia Terminal y 3.

Consolidación del Sistema Tutorial”. Resalta el esfuerzo realizado por alcanzar estos logros, y reconoce que día a día se busca fortalecer la excelencia académica en este Programa de Posgrado.

Si nos referimos a la eficiencia Terminal en el Programa, ésta fue a todas luces en este periodo, muy significativa, ya que el número de graduados de mayo de 2003 a mayo de 2005 fue de 528 alumnos en todos los programas del Posgrado (Tabla 6), representando un incremento de 193% con respecto a lo planeado (Tabla 8).

El Doctor Varela menciona que la eficiencia Terminal se debió a la constante interacción del Cuerpo Tutorial con los alumnos y las autoridades y que la meta propuesta será de 300 graduados al finalizar el año 2005 en los diferentes niveles del Programa, [Varela, 2005].

A pesar del aumento en la eficiencia Terminal, el Programa de Posgrado de Ciencias de la Administración, requiere para su mejor funcionamiento de un sistema automatizado que contribuya a tomar decisiones para la clasificación de tutores al momento en que los estudiantes se inscriben a dicho Programa. Actualmente, esta clasificación se lleva a cabo en forma manual y resulta compleja, por el gran número de estudiantes inscritos (1,339) en los diferentes programas del posgrado¹ (Véase Tabla 4) y por la planta académica que consta de 472 profesores², quienes tienen en conjunto 108 líneas de investigación³ (véase Tablas 5 y 7). Además, de la gran cantidad de información involucrada que tiene que ver con los datos de los profesores-tutores y alumnos. Un elemento más de complejidad en la información, es debido a que el Programa comprende varias Entidades Académicas participantes, además de la Facultad de Contaduría y Administración, está la Facultad de Química, el Instituto de Investigaciones Sociales y el Instituto de Investigaciones Jurídicas. Esto complica la asignación y clasificación de tutores por el gran número de variables contenidas en todo este contexto.

Tomando en cuenta la situación anterior, con este trabajo de investigación se pretende ayudar a una mejor toma de decisiones, auxiliados de nuevas tecnologías de información como la inteligencia artificial y en especial las redes neuronales artificiales, para lograr una adecuada y automatizada clasificación tutor-alumno, con el planteamiento

¹ Ídem
² Ídem

³ *Programa de Posgrado en Ciencias de la Administración*, Volumen I, 1999, Fondo Editorial FCA, (Págs. 26-32)

del diseño de un sistema neuronal artificial; dado que las variables involucradas son numerosas. La habilidad de manipular e interpretar gran cantidad de datos, así como aprender, generalizar, abstraer e intuir, hace que las redes sean muy eficaces en el procesamiento de información. El diseño adecuado de la arquitectura de un modelo de red neuronal artificial, permitirá no sólo realizar una adecuada clasificación tutor-alumno sino hacer un seguimiento supervisado de las actividades a los estudiantes y tutores con eficiencia y oportunidad para evitar en gran medida la deserción escolar y aumentar en lo posible, la eficiencia Terminal.

La importancia de las RNA radica en que pueden resolver problemas como los de visión o aprendizaje, y su procesamiento en paralelo resulta necesario si se quiere lograr respuestas en tiempo real [Martín y Sanz, 2002]. Las RNA, resultan especialmente adecuadas para estos problemas ya que, al igual que su equivalente biológico, una red neuronal puede “aprender” y, por tanto, puede entrenarse para encontrar soluciones, reconocer patrones, clasificar datos y predecir acontecimientos futuros.

Para el caso de estudio que nos ocupa en este trabajo de investigación, se detectaron los siguientes problemas, por no existir una adecuada clasificación tutor-alumno:

- 1) Los alumnos no identifican los aspectos que directa e indirectamente contribuyen a su desempeño académico por la falta de un tutor adecuado.
- 2) Aumento en la deserción escolar por falta de orientación de un tutor adecuado.
- 3) Disminución de la eficiencia Terminal por falta de orientación de un tutor adecuado.

La utilidad de este trabajo consiste en, pronosticar a través de una red neuronal a los profesores tutores que reúnan las características adecuadas para lograr la titulación de los alumnos dentro del tiempo señalado por el Programa Tutorial del Doctorado⁴.

Tomando en cuenta lo mencionado y apelando al fortalecimiento de esa búsqueda de excelencia académica en el Programa de posgrado que menciona el Dr. Ricardo Varela en su Informe, estoy convencida de la

⁴ REGALAMENTO GENERAL DE ESTUDIOS DE POSGRADO, CAP. III, Art. 25 al 28

importancia que tiene la toma de decisiones para fortalecer dicho Programa.

En resumen, este análisis nos permite ver que el problema de clasificación de tutores puede resolverse con **una red neuronal artificial**, (véase la figura 31 y el inciso A de la metodología). El siguiente paso, es elegir el tipo de red más adecuado que siga los lineamientos del problema aquí planteado, para lo cual se determina el hardware y software requerido.

Siguiendo la metodología planteada en el capítulo 4, una vez identificado el problema, seguimos al paso II.

II. Requerimientos de hardware y software para ejecutar la RNA

Como ya se mencionó en metodología descrita en la Figura 30, ésta es la etapa donde debemos conocer las especificaciones de la arquitectura del sistema que requerimos para ejecutar la red neuronal en relación al problema planteado, es decir, debemos responder a las preguntas de cómo vamos a resolver el problema y ¿qué clase de computadora está disponible en cuanto a; memoria, almacenamiento y procesadores para entrenar e implementar la red que resuelve el problema planteado?

Dentro de las arquitecturas descritas en el capítulo anterior para ejecutar la red neuronal que requerimos para la solución del problema, elegiremos un ordenador convencional con un software comercial, por tratarse de una muestra de datos.

En lo que respecta al software; dado que se ha llegado a un importante desarrollo con lenguajes simbólicos apoyados en arquitecturas convencionales sobre computadoras, se eligió el software de la empresa Neurodimension, Inc. y el software de Ayuda NeuroIntelligence⁵, descritos en la tabla 5.

III. Revisión bibliográfica

⁵ Copyright © 2001-2005 Ayuda Research, Inc.

En cuanto a la revisión bibliográfica, se realizó un sondeo en busca de alguna aplicación parecida a la que se nos plantea, observando la estrategia de otras personas en problemas similares. Al final de este trabajo de investigación se incluyen la bibliografía y las páginas de Internet consultadas para la realización del mismo.

Una vez revisada la bibliografía se eligió un modelo de red neuronal que satisficiera los requerimientos del problema aquí planteado.

IV: Elección del modelo de RNA

3. Selección del modelo de red más adecuado al problema planteado.

Para elegir el modelo de red más adecuado a nuestro problema aplicaremos los criterios explicados en el marco teórico de este trabajo en el párrafo 2.7, y estos son:

- a) El número de capas
- b) La topología de la red
- c) El tipo de respuesta
- d) La forma de los datos de entrada y salida

3.1 El número de capas.

Para conocer el número de capas debemos primero considerar que; el problema a resolver en este caso de estudio es conocido en redes neuronales como de generalización, es decir, la red neuronal se entrena con datos como *entradas* para realizar el entrenamiento de la red y para llevar a cabo la *prueba*, ésta se realiza con otros datos diferentes. Dentro de los problemas típicos de generalización se encuentran los de clasificación y predicción. En este trabajo nos interesa llevar a cabo la clasificación de tutores; con lo cual, a partir del conocimiento de la existencia de un conjunto de clases, se determina la regla, o el conjunto de reglas, que permita asignar, a cada nueva observación, a la clase que pertenece. Por lo que el número de capas será de tres; es decir, una capa de entrada, una oculta y otra de salida, por lo tanto estamos hablando de una *red multicapa*.

3.2 La topología de la red.

Sabemos que una parte de nuestro problema es clasificar tutores candidatos; para lo cual debemos alimentar la red con los datos de

profesores y obtener una salida de acuerdo a las características del desempeño de los tutores; es decir, que las conexiones de las capas en la red se alimentarán siempre “*hacia delante*” o en *forma unidireccional*, (*o feedforward*).

3.3 El tipo de respuesta.

El *tipo de respuesta* se refiere al tipo de aprendizaje de la red propuesta y ésta por su funcionamiento en relación a los tutores debe tener una topología con “*conexiones hacia delante*”, con un *aprendizaje supervisado*, ya que se dispone de la información sobre la clase para cada ejemplo; es decir es una red con un tipo de respuesta *heteroasociativa*, ya que la red se entrena para que ante la presencia de un patrón A (conjunto de profesores candidatos) responda a otro diferente B (sí ayudarán o no al egreso de los alumnos). Una vez que la red ha aprendido se introducirán los datos de profesores candidatos al Programa y se podrán clasificar de acuerdo a “Tutor que **Sí** apoya el egreso” y “Tutor que **No** apoya el egreso”; es decir, los tutores serán evaluados por los alumnos egresados del Programa y por algunas otras características propias de los tutores del Posgrado en Ciencias de la Administración de la UNAM, y estos resultados corresponderán a la salida de la red.

3.4 La forma de los datos de entrada y salida.

El caso de la red propuesta estará determinada por valores de entrada discretos y continuos, es decir los capturados por la evaluación de profesores tutores, a los cuales se les asignarán valores discretos en las entradas categóricas, en las numéricas se estandarizarán y los valores de salida corresponderán también a valores discretos los cuales estarán dados por el desempeño de los tutores; es decir, “Tutor que **Sí** apoya el egreso”, corresponderá a la salida “1” y “Tutor que **No** apoya el egreso”, a la salida “0”.

Tomando en cuenta las características anteriores, podemos afirmar que en cuanto al número de capas el modelo de red debe ser con *una capa de entrada*, *una capa interna* y una *capa de salida*; en cuanto a la topología, ésta será con conexiones “*hacia delante*”, es decir, es una red con un tipo de respuesta *heteroasociativa* y *aprendizaje dirigido*; en cuanto a la forma de los datos de la capa de entrada y salida, éstos deberán ser *discretos*. Por todas las características anteriores, llegamos a la conclusión que el modelo de red más adecuado para la resolución del problema planteado, es

el *perceptron multicapa*. El siguiente paso es la selección de valores de entrada y salida para diseñar la arquitectura de la red (véase figura 33).

4. Selección de valores de entrada (valores de entrenamiento).

Los valores de entrada, se tomarán como variables independientes⁶, que en la jerga de redes neuronales son los *valores de entrenamiento*. La variable dependiente es categórica, se refiere a si los tutores graduaron alumnos o no con valores de 1 y 0 respectivamente. Los datos que se usaron como *conjunto de entrenamiento* fueron tomados 249 profesores tutores pertenecientes al Programa de Doctorado en Ciencias de la Administración de la UNAM, dentro de los cuales se consideró a 48 como tutores candidatos; los primeros 162 se consideran como datos de *entrenamiento*, 39 como *conjunto de validación* y los 48 restantes como datos para *prueba*, éstos serán los prospectos para que una vez que la red aprenda, éstos califican como futuros tutores para el programa. Los datos fueron proporcionados por la coordinación de la Facultad de Contaduría y Administración, esta cédula se muestra en la Tabla 9 del anexo C.

Partición de los datos:

Las variables se agrupan en una base de datos de 48 columnas y 482 renglones, de las cuales se aceptaron 16 columnas y 249 renglones (casos) para el entrenamiento con 12 columnas con variables categóricas:

1. Principal o Secundario
2. Semestre de inscripción al Programa
3. Datos entrenamiento Inscritos hasta el 2003-2 graduados al 2006-2
4. ¿Coincide el área del doctorado del tutor con la del alumno?
5. ¿Pertenece al S.N.I. el tutor?
6. ¿Tuvo beca el alumno?
7. ¿Es académico el alumno?
8. Numero de alumnos del DCA como tutor principal en los tres primeros años del Programa
9. Numero de alumnos del DCA como tutor secundario en los tres primeros años del Programa
10. Tipo de contratación del tutor en la UNAM
11. Evaluación del tutor por parte del alumno
12. ¿El tutor apoya la graduación?

Las 4 columnas restantes como variables numéricas:

⁶ Término estadístico

1. Año ingreso al Programa
2. Al 2006-2 el numero de semestres en el Programa
3. Edad de alumno al ingreso del Programa
4. Edad del tutor

Una columna con los datos de salida:

- ¿El tutor apoya la graduación?

Se consideraron:

- Los registros para entrenamiento fueron 162 (65.06%)
- Los registros para validación fueron 39 (15.66%)
- Los registros para prueba fueron 48 (19.28%)

El rango:

- El rango de las columnas con variables categóricas de entrada fue de: [-1..1]
- El rango en la columna de salida fue de: [0..1]
- Las variables numéricas se estandarizaron.

Después de la captura de los datos de la capa de entrada, procederemos a encontrar los valores objetivo, es decir, los valores de la capa de salida.

5. Selección de valores de salida (valores objetivo).

Los valores de salida, se tomarán como *variables dependientes*⁷, que en la jerga de redes neuronales son los *valores objetivo*, los cuales serán también valores discretos, y éstos serán obtenidos a través del comportamiento de algunos de los tutores inscritos actualmente en el Programa Tutorial del Doctorado y que corresponden a la salida como: “¿El tutor apoya el egreso?”.

Se considera un buen tutor el que ha ayudado a la graduación como Tutor Principal o Tutor Secundario, motivando y guiando a los alumnos para que concluyan su trabajo de investigación.

Para obtener una salida binaria debemos cambiar las columnas de salida a simbólicas, es decir, trasladar esta información textual en columnas de 0's o 1's. De la siguiente manera:

“El tutor apoya el egreso”:

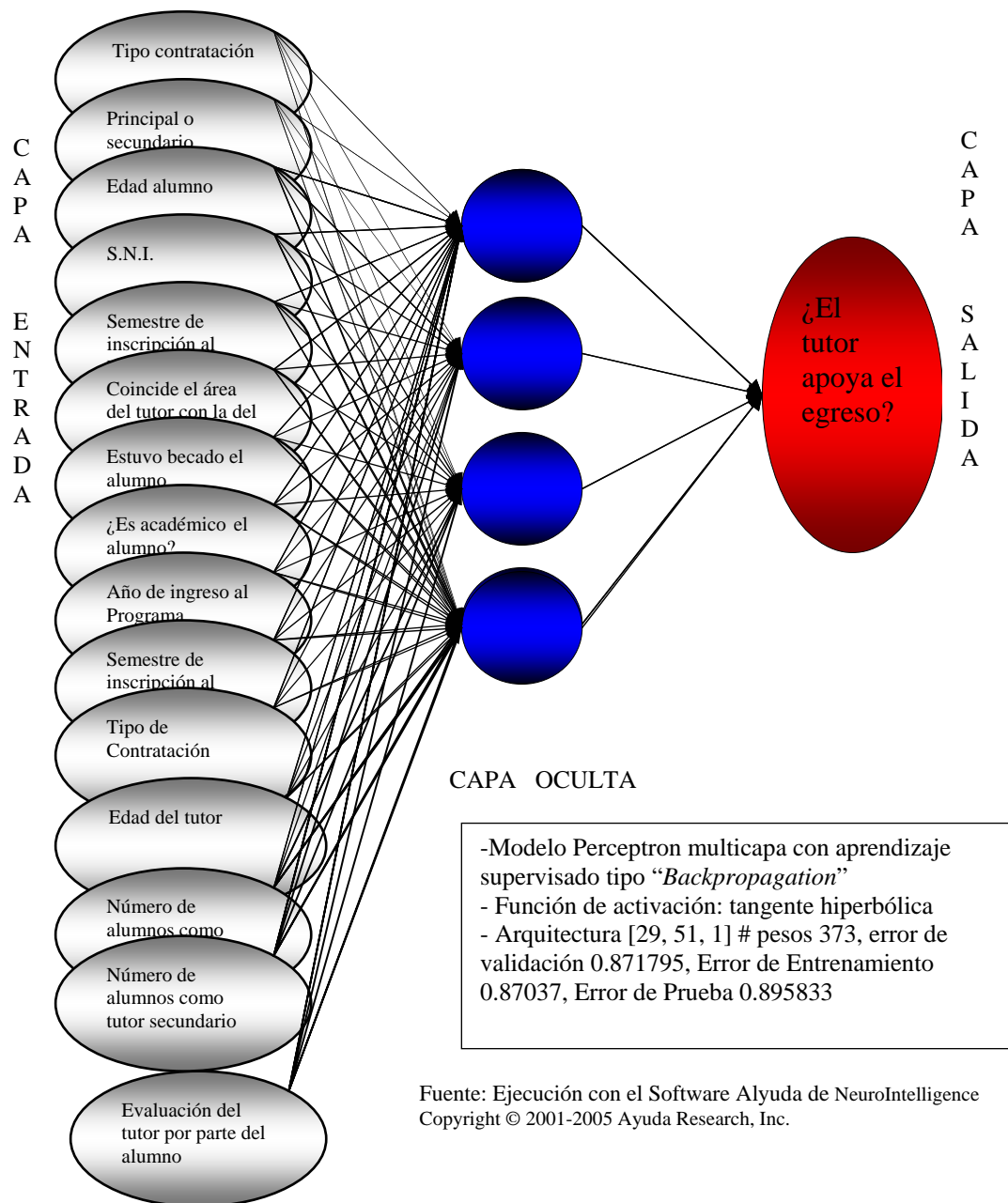
⁷ Término estadístico

- Sí = 1
- No = 0

6. La definición de la arquitectura de la red neuronal artificial para el caso de estudio propuesto.

Para definir la arquitectura nos basamos en los párrafos anteriores donde ya describimos el modelo de la red que usaremos en esta investigación y éste es un perceptron multicapa, con 2 capas, con “*conexiones hacia adelante*”, con un tipo de *aprendizaje supervisado* (con un algoritmo de corrección de error), con un tipo de asociación entre las informaciones de entrada y salida *heteroasociativa*, por la representación de la información de entrada y salida, ésta se clasifica como una *red híbrida*, es decir la entrada es analógica y la salida binaria. Por todo esto, y por la descripción de la información de entrada y salida, podemos al fin definir la arquitectura de la red como se muestra en la figura 49.

Figura 49. Arquitectura de la red neuronal que clasifica tutores en el programa de Posgrado



7. Elección de funciones de activación para las capas ocultas y función de transferencia.

Todas las neuronas que componen la red se hallan en cierto estado, podemos decir que existen dos posibles estados, *reposo* y *excitado*, a éstos se les denomina *estados de activación*, y a cada uno de los cuales se les asigna un valor. Los valores de activación asignados pueden ser discretos o continuos (Hilera, 2000:55).

En el modelo considerado, el conjunto de estados de activación es de sólo dos estados, en cuyo caso se asignará un valor entre $[0,1]$ o $[-1,1]$, generalmente siguiendo una *función tangente hiperbólica*. La importancia de usar esta función es que su derivada siempre es positiva y cercana a cero para los valores grandes positivos o negativos, esto ayuda a que exista un mejor aprendizaje.

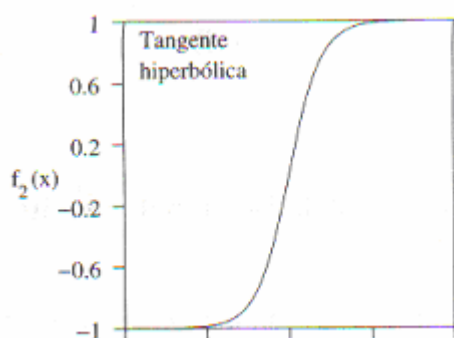
En nuestro modelo el PERCEPTRON multicapa, consideramos la función de activación, la *tangente hiperbólica*. Dicha función posee como imagen un rango continuo de valores dentro de los intervalos $[0,1]$ y $[-1,1]$, respectivamente, y viene dada por la siguiente expresión:

Función tangente hiperbólica:

$$f_2(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

Esta es una función creciente con dos niveles de saturación: el máximo, que proporciona salida 1, y el mínimo, con salida -1, como se observa en la Figura 50.

Figura 50. Función de activación de PERCEPTRON multicapa



Fuente: Viñuela [2004:49]

8. Elección de una regla de aprendizaje y fijación los parámetros iniciales.

Por la arquitectura de la red, previamente definida y las características mencionadas la regla de aprendizaje será del tipo *aprendizaje supervisado* (con un algoritmo de corrección de error) tipo *Backpropagation*.

El objetivo del aprendizaje o entrenamiento del PERCEPTRON multicapa es ajustar los parámetros de la red de pesos y umbrales— con el fin de que las entradas presentadas produzcan salidas deseadas, es decir, con el fin de minimizar la función de error E .

Los pasos que se siguieron para el proceso completo de aprendizaje del PERCEPTRON multicapa son los descritos en el libro de RNA de los autores Pedro Isasi Viñuela e Inés M. Galván León [2004:64], y son los siguientes:

Paso 1: Se inicializarán los pesos y umbrales de la red. Generalmente esta inicialización es aleatoria y con valores alrededor de cero.

Paso 2: Se toma un patrón n del conjunto de entrenamiento, $(X(n), S(n))$, y se propagará hacia la salida de la red el vector de entrada $X(n)$ utilizando las ecuaciones (1, 2 y 3 descritas en el marco teórico punto 3.2.1), obteniéndose así las respuestas de la red para dicho vector de entrada, $Y(n)$.

Paso 3: Se evalúa el error cuadrático cometido por la red para el patrón n .

Paso 4: Se aplica la regla delta generalizada para modificar los pesos y los umbrales de la red. Para ello se siguen los siguientes pasos:

- Se calcularán los valores δ para todas las neuronas de la capa de salida.
- Se calcularán los valores δ para el resto de las neuronas de la red, empezando desde la última capa oculta y *retropropagando* dichos valores hacia la capa de entrada.

- Se modificarán pesos y umbrales de la red, para el resto de los parámetros de la red.

Paso 5: Se repiten los pasos 2, 3 y 4 para todos los patrones de entrenamiento, completando así una iteración o ciclo de aprendizaje.

Paso 6: Se evalúa el error total E cometido por la red. Dicho error también recibe el nombre de error de entrenamiento, pues se calculará utilizando los patrones de entrenamiento.

Paso 7: Se repiten los pasos 2, 3, 4, 5 y 6 hasta alcanzar un mínimo del error del entrenamiento, para lo cual se realizarán m ciclos de aprendizaje.

Por tanto y desde un punto de vista teórico, el proceso de aprendizaje del PERCEPTRON multicapa debe finalizar cuando $\frac{\partial E}{\partial w} \approx 0$, momento en que los parámetros de la red no cambian de una iteración a otra. Sin embargo desde el punto de vista práctico y a la hora de implementar el proceso de aprendizaje del PERCEPTRON multicapa, se suele fijar un número de ciclos de aprendizaje de modo que cuando se alcanza dicho número, se detiene el aprendizaje. En este punto se analiza si es necesario realizar más ciclos de aprendizaje o si son suficientes los establecidos a priori.

Este análisis se basa únicamente en observar si el error cometido por la red se mantiene prácticamente constante de una iteración a otra o si, por el contrario, el error sigue decreciendo. Así, por ejemplo, si se representa el error de entrenamiento cometido por la red en función del número de ciclos, será necesario realizar más ciclos de aprendizaje.

Con los datos anteriores se realizará el procesamiento y formato de los datos. En esta etapa se detectará cuáles datos requieren algún tipo de pre-procesamiento para cómputo para lo cual se diseñará un sistema de cómputo (prototipo) que relacionará las tablas de dichos datos y algunas interfaces para su manejo y comprensión. Con lo anterior ya se crea una arquitectura de red para posteriormente entrenarla con algunos ejemplos que cumplan con los objetivos planteados. La creación del modelo de entrenamiento se realiza primero con una simple red, es decir con una sola capa oculta, esto

nos permitirá validar un modelo prototipo de acuerdo al comportamiento de dicho modelo sabremos cuántas capas más se requerirán.

Una vez terminado y probado este prototipo vamos a variar los parámetros de entrenamiento para ver si llega a los valores indicados y así sabremos que está funcionando correctamente (véase la figura 49).

Por otro lado se debe considerar otro grupo de parámetros relevantes que llamaremos “**la clasificación de tutores**”, este grupo contendrá el número de parámetros relevantes para la correcta clasificación del grupo de profesores tutores, ahora ya contamos con los valores de entrada de nuestra red y los valores de salida. El número de parámetros considerados en nuestra capa de entrada nos indicará el número de unidades en la capa de salida o el número de pesos deseados. El resto de los elementos: números de capas ocultas y número de neuronas en cada una de éstas, así como las funciones de transferencia y el nivel de conectividad entre neuronas, se definen de acuerdo a las situaciones que se presenten durante la ejecución de la red neuronal. Cuando concluya el entrenamiento o esté a punto de concluir se analizará el comportamiento de los valores de prueba, y nuevamente se analizará pero ahora con valores que sean ajenos a los valores de prueba, con la idea de ver como se comporta la red en la vida real. Finalmente se prueban los valores obtenidos con otros modelos de redes neuronales para determinar si el modelo es óptimo asignando los tutores idóneos a los alumnos.

9. Preparación del archivo con los ejemplos de entrenamiento; los patrones de entrada y los patrones de salida con los que se debe asociar.

Se deben señalar los renglones, referidos a los casos “profesores” que servirán como entrenamiento, validación cruzada, prueba o producción. Se requiere sólo de entrenamiento para correr el proceso de entrenamiento, Sin embargo, la validación cruzada es muy útil para prevenir un sobre-entrenamiento, así que se señalaron los renglones que corresponden a los maestros candidatos a tutores para prueba y de esta forma la red pueda pronosticar cuáles tutores sí ayudarán al egreso de los alumnos en el programa de Posgrado en Ciencias de la Administración de la UNAM, para esta prueba de utilizarán los datos de 249 profesores tutores en total; los cuales serán divididos en tres

grupos: 162 para el entrenamiento, 39 para validación y 48 para prueba.

10. Preprocesamiento de los datos de entrada y salida

En esta etapa, se preprocesan los datos para ponerlos en una forma útil a la red neuronal.

El archivo con los datos de salida se muestra en la tabla 10 (véase anexo B).

Las columnas antes del preprocesamiento eran 16 y después 28, el rango de escala de las columnas de entrada fue de $[-1..1]$, y de la columna de salida fue de $[0..1]$.

Los parámetros de escala de las columnas numéricas fueron:

- Año ingreso al Programa: 0.4
- Al 2006-2 el número de semestres en el Programa: 0.2
- Edad de alumno al ingreso del Programa: 0.043478
- Edad del tutor: 0.042553.

Las columnas categóricas de acuerdo a sus parámetros:

- Principal o Secundario: dos-estados
- Semestre de inscripción al Programa: 1-de-11
- Datos entrenamiento Inscritos hasta el 2003-2 graduados al 2006-2: dos-estados
- ¿Coincide el área del doctorado del tutor con la del alumno?: dos-estados.
- ¿Pertenece al S.N.I. el tutor?: dos-estados
- ¿Tuvo beca el alumno?: dos-estados
- ¿Es académico el alumno?: dos-estados
- Numero de alumnos del DCA como tutor principal en los tres primeros años del Programa: dos-estados
- Numero de alumnos del DCA como tutor secundario en los tres primeros años del Programa: dos-estados
- Tipo de contratación del tutor en la UNAM: dos-estados
- Evaluación del tutor por parte del alumno 1= Malo, 2= Regular, 3= Bueno, 4= Muy Bueno, 5= Excelente: 1-de-5
- ¿El tutor apoya la graduación?: dos-estados

11. Fijación de los parámetros de entrenamiento para cada corrida de la red.

El modelo de red tipo perceptron multicapa (MLP Multilayer Perceptron), utilizado en esta investigación, y como ya se mencionó con aprendizaje dirigido tipo *Backpropagation*, con una capa oculta y 13 nodos.

Esta red se usará para modelar los datos introducidos. Cuando se ejecuta el proceso de entrenamiento, los archivos se crean automáticamente y se cargan dentro de la red activa en el software de NeuroSolutions⁸. Cada proceso de entrenamiento despliega los resultados correspondientes al mismo, en esta etapa la red será entrenada una vez.

12. Entrenamiento de la red

Los resultados fueron obtenidos siguiendo la metodología anterior y corriendo una red neuronal del software llamado NeuroSolutions for Excel 5⁹. Este software fue diseñado para permitir desarrollar una completa solución al problema ya que además tiene la flexibilidad de personalizar las aplicaciones utilizando Visual Basic.

El objetivo del aprendizaje o entrenamiento del PERCEPTRON multicapa es ajustar los parámetros de la red de pesos y umbrales— con el fin de que las entradas presentadas produzcan salidas deseadas, es decir, con el fin de minimizar la función de error E .

Los pasos que se seguirán para el proceso completo de aprendizaje del PERCEPTRON multicapa son los descritos en el libro de RNA de los autores Pedro Isasi Viñuela e Inés M. Galván León [2004:64], y ya fueron descritos en el punto 3.3.7 del marco teórico de este trabajo. Por tanto y desde un punto de vista teórico, el proceso de aprendizaje del PERCEPTRON multicapa debe finalizar cuando $\frac{\partial E}{\partial w} \approx 0$, momento en que los parámetros de la red no cambian de una iteración a otra. Sin embargo desde el punto de vista práctico y a la hora de implementar el proceso de aprendizaje del PERCEPTRON multicapa, se suele fijar un número de ciclos de aprendizaje de modo que cuando se alcanza dicho número, se

⁸ NeuroDimension, Inc. Copyright 1994-2005, versión Number 5.01

⁹ <http://www.nd.com/neurosolutions/products>

detiene el aprendizaje. En este punto se analiza si es necesario realizar más ciclos de aprendizaje o si son suficientes los establecidos a priori.

Este análisis se basa únicamente en observar si el error cometido por la red se mantiene prácticamente constante de una iteración a otra o si, por el contrario, el error sigue decreciendo. Así, por ejemplo, si se representa el error de entrenamiento cometido por la red en función del número de ciclos, será necesario realizar más ciclos de aprendizaje.

Esta red se usó para modelar los datos introducidos. Cuando se ejecutó el proceso de entrenamiento, los archivos se crearon automáticamente y se cargaron dentro de la red activa en el software de NeuroSolutions¹⁰. Cada proceso de entrenamiento desplegó los resultados correspondientes al mismo, en esta etapa la red se entrenó una vez más, y los resultados, se muestran a continuación:

13. Resultados obtenidos al correr la red neuronal artificial propuesta.

Los resultados que se obtuvieron al correr la red neuronal descrita en la figura 47, con el software de NeuroSolutions y Ayuda de NeuroIntelligence. El tipo de reportes de entrenamiento que se obtuvieron fueron para la *clasificación* de tutores, el número de entradas a la red, fue de 118, el número de capas ocultas fue de una y el número de salidas fue de 1. La función de transferencia que se usó fue la *Tangente Hiperbólica* (vea Figura 49). El tipo de aprendizaje fue *dirigido* tipo *Backpropagation*.

13.1 Ejecución de la red para encontrar el mejor diseño.

Después de correr 12 redes se encontró el mejor diseño de la arquitectura de la red; es decir, con el mejor ajuste, éstos datos se muestra en la tabla 12, donde se observan los ajustes de las redes con diferentes arquitecturas. El criterio para saber cual fue la mejor arquitectura es que los errores de entrenamiento, validación y prueba se mantienen constantes, y el ajuste es el valor más alto; por lo que concluimos que la mejor arquitectura fue [29,51,1]; es decir, es una red con 29 datos en la capa de entrada, 51 procesadores de

¹⁰ NeuroDimension, Inc. Copyright 1994-2005, versión Number 5.01

activación en la capa oculta y una salida, la función de activación de la capa oculta fue: *tangente hiperbólica*, la salida fue ¿El tutor apoya la graduación?, la función de error fue: *suma de cuadrados del error*, la función de activación: *logística*, la clasificación del modelo tuvo límites de confianza de: *nivel de aceptación 0.5 y rechazo 0.5*.

En proceso de clasificación utiliza funciones discriminantes lineales que son adecuadas cuando la variable dependiente es categórica, al igual que una recta de regresión trata de explicar la máxima cantidad de variación de su variable dependiente, estas combinaciones lineales internas explican las variaciones o diferencias en la variable categórica dependiente. La primera función discriminante se construye para explicar la mayor cantidad de variación (diferencia) de los grupos discriminantes. La segunda función discriminante, que es ortogonal e independiente de la primera explica el mayor porcentaje de la varianza restante (residual) tras haber eliminado la varianza de la primera función. Los niveles de significación que se obtuvieron, del 0.5 revelan que ninguna de las variables contempladas en los resultados se excluya del modelo.

En la figura 51 se muestran las iteraciones contra la CCR%¹¹ (Por sus siglas en inglés “Correct Classification Rate”) de 5 redes, incluida la del mejor ajuste.

Después de examinar el aprendizaje en las curvas y los valores de la Tabla 11, se puede apreciar que el Perceptron Multicapa realizó un buen trabajo de aprendizaje en la clasificación de tutores. Para verificar esta conclusión, necesitamos correr el conjunto de datos de prueba a través del modelo de entrenamiento de la red neuronal. Automáticamente todos los procesos de entrenamiento salvan los pesos cuando el error es mínimo en el entrenamiento, se salvan los “mejores pesos”, los cuales son usados en la siguiente corrida para probar la validez de la arquitectura del modelo de red neuronal.

¹¹ CCR es usada en tareas de clasificación como una característica cualitativa. Esta tasa se calcula dividiendo el número de registros correctamente reconocidos entre el número de registros. CCR es una medida en unidades relativas o en porcentajes.

Figura 51. Los resultados que arrojó la red con el mejor ajuste

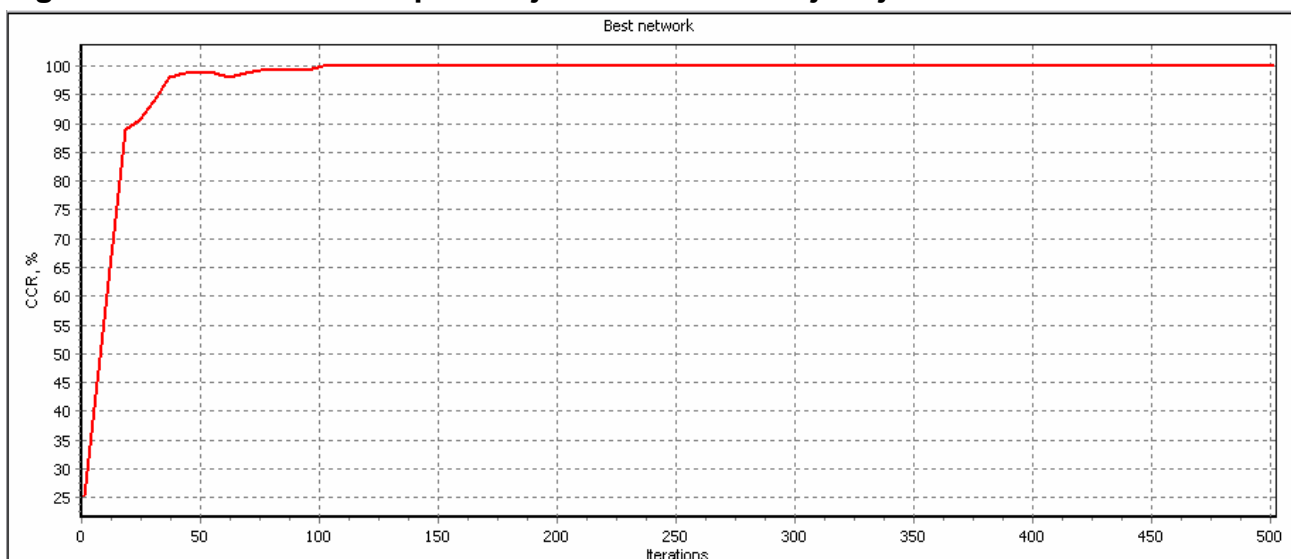


Tabla 11. Valores de la red con el mejor ajuste

<i>Parámetro</i>	<i>Valor</i>
ID	8
Arquitectura	[29-51-1]
# de pesos	373
Error de Entrenamiento	0.87037
Error de Validación	0.871795
Error de Prueba	0.895833

Copyright © 2001-2005 Ayuda Research, Inc.

13.2 Los resultados del entrenamiento

Los resultados del entrenamiento después de 500 iteraciones se muestran en la tabla 11, donde pueden apreciarse los pesos de la función discriminante, la variable con más peso es la que corresponde a la “Evaluación del Tutor por parte del alumno”, la cual representa casi un 30%; Le siguen en importancia las variables relativas al año y semestre de ingreso al Programa con 15.8% y 22.5%. Con menos peso, le siguen las variables relativas a la actividad docente de los alumnos.

Tabla 11. Los pesos de la función discriminante

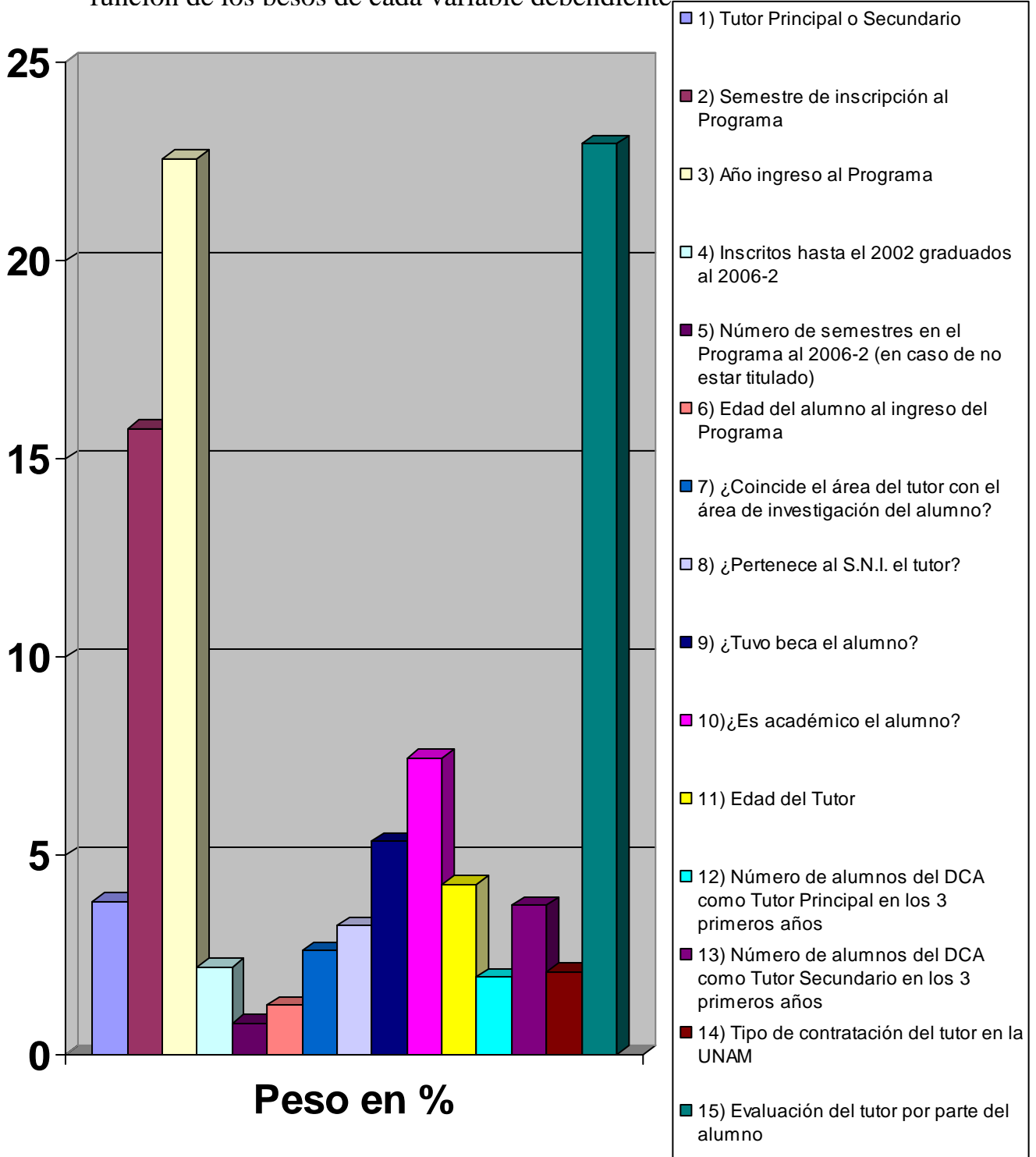
<i>Nombre de la variable de entrada</i>	<i>Peso en %</i>
1) Tutor Principal o Secundario	3.829943
2) Semestre de inscripción al Programa	15.764538
3) Año ingreso al Programa	22.564033

4) Inscritos hasta el 2002 graduados al 2006-2	2.189886
5) Número de semestres en el Programa al 2006-2 (en caso de no estar titulado)	0.773628
6) Edad del alumno al ingreso del Programa	1.228764
7) ¿Coincide el área del tutor con el área de investigación del alumno?	2.599909
8) ¿Pertenece al S.N.I. el tutor?	3.230108
9) ¿Tuvo beca el alumno?	5.356163
10) ¿Es académico el alumno?	7.471147
11) Edad del Tutor	4.277006
12) Número de alumnos del DCA como <i>Tutor Principal</i> en los 3 primeros años	1.936007
13) Número de alumnos del DCA como <i>Tutor Secundario en los 3 primeros años</i>	3.74285
14) Tipo de contratación del tutor en la UNAM	2.084323
15) del tutor por parte del alumno	22.951695

Fuente: Ejecución con el Software Alyuda de NeuroIntelligence
 Copyright © Evaluación 2001-2005 Ayuda Research, Inc.

En la figura 52 se muestran los resultados de los pesos del entrenamiento de la red neuronal de las variables de entrada utilizadas en esta investigación.

Figura 52. Resultados del entrenamiento de la red en función de los pesos de cada variable dependiente



Fuente: Ejecución con el Software Alyuda de NeuroIntelligence
 Copyright © 2001-2005 Ayuda Research, Inc.

13.3 Ecuación de la función discriminante:

$$V_{dep} = 3.829943X_1 + 15.764538X_2 + 22.564033X_3 + 2.189886X_4 + 0.773628X_5 + 1.228764X_6 + 2.59909X_7 + 3.230108X_8 + 5.356163X_9 + 7.471147X_{10} + 4.2777006X_{11} + 1.936007X_{12} + 3.74285X_{13} + 2.084323X_{14} + 22.951695X_{15}$$

Los 249 profesores tutores considerados para la ejecución de la red, se dividieron en tres grupos: un grupo de 162 fue considerado como datos de *entrenamiento*, 39 como *validación* y los 48 restantes como grupo para *prueba*, estos últimos son los tutores que se clasificaron por la red neuronal y los resultados que nos arrojó son los que se muestran en la siguiente matriz de clasificación:

Los 48 profesores considerados como datos de prueba quedaron clasificados en la tabla 12.

Tabla 12. Matriz de clasificación de los datos de prueba.

<i>Deseada/salida</i>	<i>Sí</i>	<i>No</i>
<i>Sí</i>	37	3
<i>No</i>	5	3

Fuentes: Ejecución con el Software Alyuda de NeuroIntelligence Copyright © 2001-2005 Ayuda Research, Inc. y ejecución del modelo de red con el Software de NeuroDimension, Inc. Copyright 1994-2005, Versión 5.01, con datos de profesores del Programa de Doctorado en Ciencias de la Administración proporcionados por la coordinación (junio, 2006).

En la tabla 12, puede observarse que 40 tutores fueron bien clasificados, lo que representa el 83.3% del total de los datos de prueba.

Limitaciones del estudio:

- La aplicación de redes neuronales artificiales se hizo solo para el Programa de Doctorado en Ciencias de la Administración de la UNAM, dadas las restricciones de tiempo y de información; sin embargo, esta aplicación puede generalizarse para el caso de otros programas del Posgrado.

Recomendaciones:

- Se sugiere que se realicen evaluaciones periódicas del desempeño de los tutores mediante encuestas a los alumnos.
- Se sugiere que se actualicen los datos de los tutores periódicamente mediante la creación del Currículum único que puede actualizarse en tiempo real, ya que esto permitiría retroalimentar la red neuronal para generalizar con datos nuevos y reconocer datos aprendidos previamente.

Conclusiones:

El objetivo de este trabajo de investigación se cumplió al probar que el modelo del Perceptron Multicapa, es el modelo de red neuronal artificial más adecuado para la clasificación de tutores en el Programa de Posgrado en Ciencias de la Administración de la UNAM.

El diagnóstico que se realizó al posgrado nos permitió identificar los problemas existentes en el manejo de la información para el control de los tutores asignados al Programa del Doctorado.

Se encontraron muchas ventajas al aplicar redes neuronales para la toma de decisiones en instituciones de educación superior.

Se logró plantear una metodología automatizada para diseñar la arquitectura y el modelo de una red neuronal que permitirá una mejor comprensión de este tema para resolver problemas que ayuden a la toma de decisiones.

Se respondió a la pregunta de investigación de acuerdo a la hipótesis planteada, considerando que el modelo de red neuronal más adecuado para clasificar tutores de manera eficiente en el Programa de Posgrado en Ciencias de la Administración de la UNAM, es el Perceptrón Multicapa, con “conexiones hacia adelante”, aprendizaje supervisado (algoritmo de aprendizaje con corrección de error), red heteroasociativa, con entrada analógica y salida binaria.

La red neuronal trabaja en forma paralela y distribuida y su importancia radica en que puede resolver problemas de aprendizaje y su procesamiento en paralelo resulta necesario si que quieren lograr respuestas en tiempo real. Una red neuronal “aprende a base de ejemplos”, por esa razón puede entrenarse para encontrar soluciones, en nuestro caso de investigación nos permitió clasificar datos y predecir acontecimientos futuros. Por esta razón concluimos en este trabajo la importancia de éstas para que la toma de decisiones sea eficiente y eficaz.

La red neuronal artificial, que se plantea en este trabajo de investigación, puede hacer no sólo que “aprenda” sino también que generalice, el Perceptron multicapa, resulta ser el modelo de red neuronal artificial más adecuado para lograr que la clasificación de tutores sea más eficiente y dar así seguimiento al Programa de Posgrado en Ciencias de la Administración de la UNAM.

Para el caso de estudio presentado en esta investigación, se llevaron a cabo las etapas presentadas en la metodología, las cuales resultaron suficientes para encontrar el diseño de la arquitectura de la red neuronal que clasificará a los tutores candidatos al Programa. La teoría correspondiente a la metodología presentada en este trabajo se anexa en el trabajo en un CD tipo multimedia para su mejor comprensión a las personas interesadas. (Bribiesca G, 2006).

Los programas NeuroSolutions y Alyuda NeuroIntelligence de simulación de redes neuronales proporcionan una visión interesante del problema, y son útiles en la etapa de desarrollo, aunque casi siempre es necesario utilizar un software específico.

Si la aplicación requiere el manejo de bases de datos robustas y en tiempo real, puede resultar necesario el empleo de hardware neuronal específico.

El modelo planteado puede resolver la clasificación de tutores con más variables de entrada y con datos en tiempo real. Sin embargo, no se descartan nuevas investigaciones con otros modelos de redes con “aprendizaje no dirigido” y probar los resultados.

Finalmente, podemos concluir que las RNA deberán emplearse siempre y cuando se justifique su uso de acuerdo al problema que se va a resolver y una vez justificado se debe encontrar el modelo adecuado; una red neuronal no tiene porque ser siempre la solución idónea. En este sentido debe tenerse en cuenta una perspectiva global en la que incluyan aspectos económicos, velocidad de desarrollo, robustez y fiabilidad.

Se puede apreciar que el Perceptron Multicapa realizó un buen trabajo de aprendizaje en la clasificación de tutores. Esto se concluyó por los valores mínimos de error en la validación y el entrenamiento de la red, y cuando los parámetros de la red no cambiaron de una iteración a otra, además se revisó cuidadosamente el rango de error mínimo y máximo para determinar cuántas veces más se debía entrenar la red. Después se analizó el desempeño de este entrenamiento usando el conjunto de datos y el entrenamiento previo de la red.

Por lo anterior podemos afirmar que las redes neuronales permiten la clasificación adecuada de tutores, a través del Perceptron Multicapa.

Bibliografía

Bonsón, Enrique, “*Tecnologías Inteligentes para la Gestión Empresarial*”, Editorial Alfaomega, Ra-ma, 1999.

Cohen, “*Sistemas de Información para la Toma de Decisiones*”, 2ª Edición, Editorial McGraw-Hill, ISBN: 9701008820, 1996.

Effy Oz, “*Administración de sistemas de información*”, Thomson Learning, México, 2001, ISBN 970-686-043-6.

Franas J. Bridges, Kenneth W. Olm y Allison Barn Hill “*Management Decisions and Organizational Policy*”, Boston: Allyn & Bacon, 1971.

Giarratano Joseph “*Sistemas expertos, principios y programación*”, Tercera edición, Internacional Thomson Editores, México, 2001, ISBN 970-686-059-2.

Grossberg, “*Some nonlinear networks capable of learning a partial pattern of arbitrary complexity*”. Proceedings of the National Academy of Sciences, 59, Págs. 368-372, 1968.

Hebb, D. *Organization of Behavior*. John Wiley & Sons, New York. 1949. (Reimpreso parcialmente en el texto “*Neurocomputing*” (J. Anderson y E. Rosenfeld ed.), MIT Press, 1988)

Hilera, Martínez. “*Redes Neuronales Artificiales. Fundamentos, Modelos y Aplicaciones*”, ISBN 970-15-0571-9, Editorial Alfaomega Ra-ma, 2000.

Hopfield, J *Neural networks and physical systems with emergent collective computational abilities*. In proceedings of *National Academy of Science*, volume 81, 1982, Pages 3088-3092.

Huber C. P. *Toma de decisiones en la gerencia*. Editorial Trillas, 1984.

Isasi Viñuela, Galván León, *Redes de Neuronas Artificiales Un Enfoque Práctico*, Editorial Pearson Education, S.A., Madrid, 2004.

Kendall, Kenneth & Kendall Julie, " *Análisis y Diseño de Sistemas*", ISBN 968-880-694-3, México, Editorial Prentice Hall Hispanoamericana, S.A. 1997.

Kohonen. " *Self-organized formation of topologically correct feature maps*". Biological Cybernetics, 1982. Reimpreso en el texto " *Neurocomputing*" (J. Anderson y E. Rosenfeld ed.), MIT Press. 1988.

Laundon, Kenneth C y Laundon, Jane P. *Sistemas de información gerencial*, Editorial Pearson Education, México, 2004.

Levin, Richard I., " *Estadística para Administradores*", 1988, ISBN 968-880-1526, Editorial Prentice-Hall Hispanoamericana, S.A. Págs. 371-376.

Llinás, Rodolfo, R. " *El cerebro y el mito del yo: El papel de las neuronas en el pensamiento y el comportamiento humanos*", Bogotá Colombia, 2003, ISBN 958-04-6798-6.

Martín y Sanz. " *Redes Neuronales y Sistemas Difusos*". Editorial Alfaomega, 2002, ISBN 970-15-0733-9.

McLeod, Raymond Jr. *Sistemas de información gerencial*, Editorial Editorial Pearson Education, S. A., 2000, México, 2000.

Moravec, H. Mind Children. *The Future of Robot and Human Intelligence*. Harvard University Press, 1988.

Masters, T. " *Practical Neural Networks Recipes in C++* " Academic Press, 1993.

O'Brien, " *Sistemas de Información Gerencial*", 4ª Edición, Editorial McGraw-Hill, ISBN: 9584101773. (25-APR-01).

Pérez, Delgado María Luisa, Martín Quintín " *Aplicaciones de las Redes Neuronales Artificiales a la Estadística*", Cuadernos de Estadística, Editorial LA MURALLA S.A., HESPÉRIDES, S. L. Impreso en Madrid, 2003.

PROGRAMA DE POSGRADO EN CIENCIAS DE LA ADMINISTRACIÓN, “*Fundamentos del Programa*”, Volumen I, 1999, Fondo Editorial FCA. Aprobado por el Consejo Académico del Área de Las Ciencias Sociales, el 29 de enero de 1999. Entidades Académicas Participantes: Facultad de Contaduría y Administración, Facultad de Química, Instituto de Investigaciones Sociales, Instituto de Investigaciones Jurídicas.

REGLAMENTO GENERAL DE ESTUDIOS DE POSGRADO Aprobado por el Consejo Universitario en su sesión del 14 de diciembre de 1995, Publicado en la Gaceta UNAM el 11 de enero de 1996.

Ribeiro Lair, “*Inteligencia Aplicada*”. Editorial Planeta, 2º edición, España, 2003, ISBN 84-08-4437-0.

Rodríguez Estrada Mauro, “*Manejo de problemas y toma de decisiones*” 2ª Edición. Editorial: Manual Moderno, 1988.

Rosenblatt F. “*The Perceptron: A theory of statistical separability in cognitive systems*”, 1958, Technical Report VG-1196-G-1, Cornell Aeronautical Laboratory.

Russell Stuart, Norving Peter, “*Inteligencia Artificial Un Enfoque Moderno*”, ISBN: 0-13-790395-2, Editorial Prentice Hall, 2004.

Schalkoff, Robert J. “*Artificial Neural Networks*”, (Computer Science), ISBN 0-07-115554-6 Printed in Singapore, Co-publisher by the MIT Press and the McGraw-Hill Companies, Inc., 1997.

Simón, Herbert A. “*The New Science of Management Decision*”, ed. Rev. (Englewood Cliffs, N); Prentice Hall, 1977, 46

Software de NeuroSolutions. NeuroDimension, Inc. Copyright 1994-2005, Version Number 5.01, Authors: Curt Lefebvre, Jose Principe, David Samson, Dan Wooten, Gary Geniesse, Michele Lucas, Craig Fancourt, Jason Gestenberger, Neil Euliano, Gary Lynn, Mark Allen, Dorothee Marossero.

Varela, Ricardo A. “Informe de Actividades 2003-2005”, UNAM, 2005.

Widrow, B., Rumelhart, D. E., Lehr, M. E. *Neural networks: Applications in industry, business and science. Communications of the ACM*, 37, 3, 93-105, 1994.

Páginas consultadas

- [html://ciberconta.unizar.es/INICIO.HTML](http://ciberconta.unizar.es/INICIO.HTML)
- <http://www.monografias.com/trabajos7/sisinf/sisinf.shtml>
- <http://www.redcientifica.com/doc/doc199903310003.html>, Xavier Padern, ISSN: 1597-0223
- [Http://ciberconta.unizar.es/LECCION/REDES/180.HTM](http://ciberconta.unizar.es/LECCION/REDES/180.HTM)
- <http://home.ubalt.edu/ntsbarsh/stat-data/Forecasts.htm#rneural>
- <http://www.nd.com/neurosolutions/products>

ANEXO A

Relación de tablas de la situación actual del Programa de Posgrado en Ciencias de la Administración de la UNAM.

Tabla 3. Número de aspirantes registrados a los procesos de admisión del 2003-2 al 2005-2.

<i>Campo de conocimiento</i>	<i>Total Aspirantes</i>	<i>Total Aceptados</i>	<i>% Aceptados</i>
Total	2,989	1,431	48
Doctorado	122	46	38
Subtotal Maestrías	2,112	1,066	49
Administración (Organizaciones)	632	253	40
Administración (Negocios Internacionales)	308	184	60
Administración (Sistemas de Salud)	106	73	69
Administración (Industrial)	252	81	32
Finanzas	648	258	40
Auditoría	96	56	58
Sedes externas	170	161	98
Subtotal Especializaciones	655	319	49
Fiscal	481	200	42
Mercadotecnia	58	41	71
Alta Dirección	46	43	93
Recursos Humanos	70	35	50

Fuente: [Varela, 2005]

Tabla 4. Matrícula del Programa de Posgrado en Ciencias de la Administración 2005.

Campo de Conocimiento	Matrícula actual
Total	1,339
Doctorado	66
Subtotal Maestrías	959
Administración (Organizaciones)	334

Administración (Negocios Internacionales)	152
Administración (Sistemas de Salud)	72
Administración (Industrial)	66
Finanzas	263
Auditoría	55
Sedes externas	17
Subtotal Especializaciones	314
Fiscal	243
Mercadotecnia	24
Alta Dirección	23
Recursos Humanos	24

Fuente: [Varela, 2005]

Tabla 5. Planta académica

	Doctorado	Maestrías	Especializaciones
	2005	2005	2005
Aspirante	1		
Externo a la UNAM	33		
Jubilado	2	7	1
No Vigente	5		
Prof. Asig. Def. Nivel A	6	17	1
Prof. Asig. Def. Nivel B	0	6	2
Prof. Asig. Int. Nivel A	53	131	51
Prof. Asig. Int. Nivel A	0	1	
Prof. Medio Tiempo	3	1	
Prof. Tiempo Completo	116	29	6
Total	219	192	61

Fuente: [Varela, 2005]

Tabla 6. Graduados por Campo de conocimiento

Campo de conocimiento	Graduados
Especializaciones	
1) Fiscal	88
2) Fiscal (Contribuciones)	9
3) Mercadotecnia	10
4) Dirección de Recursos Humanos	12
5) Alta Dirección	4
6) Informática	1
7) Administración (Sistemas de Información)	1
8) Contraloría Pública	1
9) Administración de Personal	1

Campo de conocimiento	Graduados
10) Finanzas	2
Subtotal	129
Maestrías	
1) Administración	3
2) Administración de las Organizaciones	178
3) Administración (Negocios Internacionales)	32
4) Administración de la Atención Médica y de Hospitales	5
5) Administración (Sistemas de Salud)	24
6) Administración Industrial	12
7) Contaduría	4
8) Finanzas	48
9) Auditoría	77
Subtotal	383
Doctorado	
1) Administración (Organizaciones)	6
2) Ciencias de la Administración	10
Subtotal	16
Total	528

Fuente: [Varela, 2005]

Tabla 7. Eficiencia Terminal

Campo de Conocimiento	Compromiso	Alcanzado	% Alcanzado
Especializaciones	75	107	143 %
Maestrías	115	265	230 %
Doctorado	10	14	140 %
Total	200	386	193 %

Fuente: [Varela, 2005]

Tabla 8. Líneas de investigación por área de conocimiento.

Área de Conocimiento	Líneas de Investigación
Teoría de la Administración	<ol style="list-style-type: none"> 1) Epistemología de la administración. 2) Sistemas de control para México. 3) Planeación estratégica. 4) Nuevas tendencias administrativas en las empresas. 5) Procesos administrativos en las empresas. 6) Modelos gerenciales. 7) Elaboración, diseño y control de proyectos sociales y privados. 8) Estrategias asociativas de las micro, pequeñas y

Área de Conocimiento	Líneas de Investigación
	<p>medianas empresas.</p> <p>9) Enfoque sistémico en la administración.</p> <p>10) Responsabilidad social.</p> <p>11) Análisis de problemas y toma de decisiones.</p>
Teoría de la Organización	<p>12) Modelos organizacionales.</p> <p>13) Organizaciones que aprenden: análisis sistémico del comportamiento organizacional.</p> <p>14) Comportamiento humano, enfocado a procesos de mejora continua.</p> <p>15) Reingeniería de las organizaciones y del sector público.</p> <p>16) Diagnóstico de empresas y aplicación del enfoque sistémico en las áreas de administración.</p> <p>17) Investigación interdisciplinaria sobre problemas de desastres.</p> <p>18) Sistemas de soporte informático.</p> <p>19) Desarrollo de la metodología sobre el análisis sistémico.</p> <p>20) Análisis de sistemas para la procuración de justicia.</p> <p>21) Análisis de estructuras organizacionales.</p> <p>22) Flexibilidad y organización.</p> <p>23) Control de gestión.</p> <p>24) Procesos de privatización.</p> <p>25) Taxonomía de organizaciones.</p> <p>26) Instrumentos de medición para las organizaciones.</p> <p>27) Participación de las pequeñas y medianas empresas en la integración de cadenas productivas.</p> <p>28) Promoción empresarial.</p> <p>29) Estilos de administración.</p> <p>30) Promoción de negocios.</p> <p>31) Desarrollo sustentable.</p> <p>32) Modernización industrial.</p>
Finanzas	<p>33) Teoría financiera.</p> <p>34) Las finanzas y los negocios internacionales.</p> <p>35) El mercado bursátil promotor del desarrollo de actividades productivas.</p> <p>36) El sistema bancario mexicano, desafío, oportunidades y los esquemas de conformidad requeridos.</p> <p>37) Finanzas agropecuarias.</p> <p>38) Finanzas corporativas y medio ambiente macroeconómico: (Proyectos de inversión y aspectos financieros de planeación estratégica).</p> <p>39) Proyectos de inversión.</p> <p>40) Administración de riesgos. En evaluación de proyectos de inversión ligados a la exportación.</p> <p>41) Alianzas estratégicas de la gran empresa en México.</p> <p>42) Modelos de valuación de empresas.</p> <p>43) Mercado de valores.</p> <p>44) Política financiera.</p> <p>45) Mercado de futuros.</p>

Área de Conocimiento	Líneas de Investigación
	46) Fusiones y adquisiciones.
Mercadotecnia	47) Uso de las técnicas de investigación de mercados en México. 48) Economía informal y canales de distribución. 49) Comportamiento del consumidor en la selección de centros comerciales. 50) Uso de la mercadotecnia social en la solución de problemas del sector salud. 51) Investigación de mercados. 52) Comportamiento del consumidor. 53) Gestión de la mercadotecnia. 54) Análisis de la competencia. 55) Nuevos productos.
Recursos Humanos	56) Relaciones entre estilos de personalidad y efectividad organizacional. 57) Trabajo de estudiantes. 58) Seguimiento de estudiantes. 59) Capacitación y educación. 60) Sentimiento del consumidor. 61) Desarrollo organizacional. 62) Cultura organizacional. 63) Clima organizacional. 64) Dinámica organizacional. 65) El comportamiento humano en las organizaciones, (liderazgo, motivación, comunicación, etc.)
Operaciones	66) Optimización de los recursos. 67) Calidad en producción y servicios. 68) Modelos para optimización de recursos. 69) Estudios técnico-económicos y desarrollo de modelos matemáticos para la toma de decisiones.
Auditoría	70) Auditoría administrativa y organizacional. 71) Evaluación institucional. 72) Muestreo estadístico en Auditoría. 73) Evaluación del riesgo en Auditoría.
Contabilidad	74) Contabilidad gubernamental. 75) Homologación y discrepancia entre los principios de contabilidad generalmente aceptados entre los países firmantes del Tratado de Libre Comercio Norteamericano. 76) Homologación de la profesión contable en los países firmantes del tratado de Libre Comercio Norteamericano.
Negocios Internacionales	77) Acuerdos comerciales internacionales. 78) Modalidades culturales en los Negocios Internacionales. 79) Entorno económico 80) Marcas, patentes y productos. 81) Análisis de la competencia internacional. 82) Operaciones multinacionales.

Área de Conocimiento	Líneas de Investigación
Administración de la Innovación Tecnológica	83) Competitividad tecnológica en las empresas. 84) Gestión tecnológica/Innovación tecnológica. 85) Política científica y tecnológica. 86) Innovación tecnológica para la pequeña y microempresa. 87) Competitividad en pequeñas y medianas empresas regionales. 88) Los significados sociales de las nuevas tecnologías. 89) Antropología industrial. 90) Competitividad empresarial. 91) Integración de cadenas productivas.
Matemáticas	92) Métodos de estadística. 93) Sistemas dinámicos. 94) Matemáticas aplicadas a la administración y finanzas. 95) Teoría general de sistemas y aplicaciones. 96) Teoría de control óptimo de aplicaciones. 97) Investigación de operaciones. 98) Aplicación de matemáticas y computadoras a problemas de la educación (orientada a estudiantes y profesores de la educación); Educación Continua. 99) Simulación.
Economía	100) Energía/Economía. 101) Economía industrial, sector manufacturero. 102) Problemas regionales en el espacio rural. Teoría de la región. 103) Nuevas metodologías para las historias de la vida. 104) Teoría económica internacional: parte real, parte financiera. 105) Desarrollo y planeación a largo plazo. 106) Macroeconomía aplicada. 107) Microeconomía aplicada. 108) Economía de la empresa.

Tabla 8. Fuente: *Programa de Posgrado en Ciencias de la Administración*, Volumen I, 1999, Fondo Editorial FCA, (Págs. 26-32)

ANEXO B

Tabla 9. Los valores de entrada o entrenamiento se muestran en las 16 variables como columnas.

Clave Alumno	Clave Tutor	Principal o Secundario	Semestre de inscripción al Programa	Año ingreso al Programa	Datos entrenamiento Inscritos hasta el 2003-2 graduados al 2006-2	Al 2006-2 el número de semestres en el Programa	Edad de alumno al ingreso del Programa	¿Coincide el área del doctorado del tutor con la del alumno?	¿Pertenece al S.N.I. el tutor?	¿Tuvo beca el alumno?	¿Es académico el alumno?	Edad del tutor	Numero de alumnos del DCA como tutor principal en los tres primeros años del Programa	Numero de alumnos del DCA como tutor secundario en los tres primeros años del Programa	Tipo de contratación del tutor en la UNAM	Evaluación del tutor por parte del alumno 1= Malo, 2= Regular, 3= Bueno, 4= Muy Bueno, 5= Excelente	¿El tutor apoya la graduación?
--------------	-------------	------------------------	-------------------------------------	-------------------------	---	---	--	--	--------------------------------	-----------------------	--------------------------	----------------	---	--	---	---	--------------------------------

Fuente: Datos proporcionados por la coordinación de la FCA

