

Universidad de las Ciencias Informáticas

Facultad 5



Título: Teoría de errores aplicada a los modelos de Redes Neuronales Artificiales utilizados en el proyecto “Desarrollo de Elementos Virtuales Inteligentes”.

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autor(es): Isleixys Arias Rodríguez.

Adiel Durán Rodríguez.

Tutor(es): MSc. Roberto Millet Luaces.

Ing. Liudmila Reyes Alvarez.

La Habana

Julio 2008

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Isleixys Arias Rodríguez

Adiel Durán Rodríguez

Firma del Autor

Firma del Autor

MSc. Roberto Millet Luaces

Ing. Liudmila Reyes Alvarez

Firma del Tutor

Firma del Tutor

Los errores son los umbrales del descubrimiento.

James Joyce

AGRADECIMIENTOS

A Millet, por ser un tutor magnífico, por darnos su apoyo y su confianza.

*A Adiel, por aguantar todas mis malcriadeces durante la tesis y obligarme a tener
responsabilidad.*

A Liudmila, porque en el último momento nos ayudó a mejorar.

*A mi papá y mamá María Esther, por apoyarme durante mi carrera, por estar a mi lado
cuando los necesitaba sin importar la distancia, por ser mi guía y mi apoyo.*

*A mi Tío Tomás, por sus regaños, sus gritos, por su ayuda incondicional, por su paciencia
y confianza en mi.*

A mi tía Lupe y mi prima nena por quererme tanto y ayudarme en todo momento.

*A mi mamita por decirme tantas veces que me mirara en su espejo y darme fuerzas para
continuar, por quererme tanto y estar orgullosa de mí.*

*A Yayi, Ani, Yani y Yosba, por ser los niños de mi alma, por quererlos tanto y ser la
inspiración de mi vida.*

A mis abuelos, aunque algunos ya no están, pero se que están orgullosos de mí.

*A Frank, por quererme y apoyarme en los momentos más difíciles, por darme ánimo y
amor.*

A mi familia de Cárdenas por su ayuda constante y estar conmigo en todo momento.

A mis amigas de la infancia que me han seguido de cerquita aunque ya no estén conmigo.

A mis amigas de 1er y 2do año, por su apoyo incondicional.

A mis amigas de ahora, gracias por soportarme y ayudarme en mis crisis.

A Annia, Andrea y Julián, por ser mi familia aquí en la escuela.

*A todos los profesores, amigos y familia, que han hecho posible que mi carrera llegue a su
fin.*

Isleixys.

AGRADECIMIENTOS

Al profesor Millet, por ser nuestro guía durante todo el desarrollo de la tesis.

A Isle, quien a pesar del ser malcriada fue muy buena compañera de tesis.

A Liudmila, por darnos animarnos siempre y ayudarnos.

A mis padres, por ser mis guías, mi apoyo durante toda mi carrera y confiar en mí.

A mi familia en general, por estar siempre preocupados y pendientes a mis estudios.

A mis amigos y amigas que han estado a mi lado cuando me ha hecho falta.

A mis antiguos profesores de la Cujae, quienes sentaron mis bases en la informática.

A los profesores de la UCI que de alguna forma me han brindado su ayuda. En general a todas las personas que me han apoyado durante mi carrera.

Adiel

DEDICATORIA

A mi abuela Pancha, por ser mi vida...
Isleixys

A mi abuela Tata, quien hubiera deseado estar conmigo en este momento...
Adiel.

RESUMEN

Actualmente en el proyecto “Desarrollo de Elementos Virtuales Inteligentes” (Uvi-Bot) se trabaja con los modelos de redes neuronales artificiales en el desarrollo de aplicaciones para realidad virtual. Este proyecto utiliza principalmente algunos de ellos cada uno de los cuales tiene sus funciones de transferencia para su funcionamiento aprendizaje, los usan en pequeñas aplicaciones o demos, y módulos de juegos desarrollados por el proyecto que está vinculado totalmente a la Inteligencia artificial. Se necesita investigar a fondo estos modelos de redes, su funcionamiento y posibles errores que se cometen en el interior de la red. A estos modelos no se le aplica la teoría de errores y los expertos no tienen conocimiento de cómo disminuir los errores más simples presentes en los mismos, lo que puede ser un problema a la hora de trabajar con los modelos de redes neuronales y con sus funciones de transferencias.

En este trabajo se realiza un análisis de cada modelo que utiliza el proyecto y se logra aplicar la teoría de errores a los modelos de redes neuronales artificiales, teniendo como resultado los diferentes errores cometidos en las redes neuronales artificiales y sus funciones de transferencias y cómo disminuir los errores más simples. El trabajo puede ser de utilidad para el futuro porque ayudaría a perfeccionar los modelos y a sus funciones, partiendo del conocimiento que le aporta la investigación realizada de cuáles son los errores y cómo evitar los más simples.

PALABRAS CLAVE

Red Neuronal Artificial, Neurona, Función de Transferencia, Teoría de Errores, Error.

TABLA DE CONTENIDOS

AGRADECIMIENTOS	I
AGRADECIMIENTOS	II
DEDICATORIA	III
RESUMEN	IV
INTRODUCCIÓN	1
CAPÍTULO I: "FUNDAMENTACIÓN TEÓRICA"	4
1.1 Introducción.....	4
1.2 Modelo biológico de redes neuronales.....	4
1.3 La neurona de lo biológico a lo artificial.....	5
1.4 Modelo de red neuronal artificial.....	6
1.4.1 Elementos de una Red Neuronal Artificial.....	8
1.4.2 Función de propagación.....	11
1.4.3 Función de activación o transferencia.....	11
1.4.4 Función de salida.....	14
1.4.5 Topología de las redes neuronales artificiales.....	14
1.4.6 Principales Modelos de Redes Neuronales Artificiales.....	15
1.4.7 Mecanismo de Aprendizaje de las Redes Neuronales Artificiales.....	17
1.4.7.1 Redes con Aprendizaje Supervisado.....	18
1.4.7.2 Redes con Aprendizaje No Supervisado	19
1.4.8 Entrada y Salida en las Redes Neuronales Artificiales.....	20
1.4.9 Ventajas de las Redes Neuronales Artificiales.....	21
1.4.10 Aplicaciones de las Redes Neuronales Artificiales.....	22
1.4.10.1 Ejemplos de Aplicaciones que utilizan las RNA.....	24
1.5 Teoría de Errores.....	25
1.5.1 La Incertidumbre y el Error.....	26
1.5.2 Clasificación de los errores.....	26
1.5.3 Error Absoluto y Error Relativo.....	28
1.5.4 Propagación de errores.....	29
1.5.6 Importancia de la teoría de errores en la informática.....	30
CAPÍTULO II: "DESCRIPCIÓN DE LOS MODELOS DE RNA"	32
2.1 Introducción.....	32
2.2 Selección de los modelos de redes neuronales artificiales.....	32
2.3 Modelos de Redes Neuronales Perceptrón.....	33
2.3.1 Estructura del modelo de red.....	34
2.3.2 Aprendizaje.....	35
2.4 Modelos de Redes Neuronales. Adeline.....	36

2.4.1 Estructura del modelo de red.....	37
2.4.2 Aprendizaje.....	38
2.5 Modelos de Redes Neuronales Backpropagation (Perceptrón Multicapa).....	41
2.5.1 Estructura del modelo de red.....	42
2.5.2 Aprendizaje.....	44
2.5.2.1 Entrenamiento de una Red de Unidades Sigmoidales.....	46
2.5.2.2 Características principales de la Retropropagación.....	47
2.6 Funciones de Transferencias Sigmoidal.....	47
2.6.1 Poder expresivo de Redes de Unidades Sigmoidales.....	48
2.7 Funciones de Transferencias Limitador fuerte (Hardlim).....	49
2.8 Funciones de Transferencias Lineal (purelin).....	50
CAPÍTULO III: “TEORÍA DE ERRORES APLICADA A LOS MODELOS”.....	51
3.1 Introducción.....	51
3.2 Representación Numérica en Sistemas Digitales.....	51
3.3 Errores en el modelo de RNA Perceptrón.....	52
3.4 Errores en el modelo RNA Adeline.....	54
3.5 Errores en el modelo RNA Backpropagation.....	56
CONCLUSIONES	59
RECOMENDACIONES	60
BIBLIOGRAFÍA.....	61
GLOSARIO	64

INTRODUCCIÓN

En los últimos años se han venido desarrollando diversas investigaciones en el área de la Inteligencia Artificial (IA), uno de los objetivos de la IA es simular el funcionamiento humano a través de las computadoras.

Para lograr este objetivo se han propuesto modelos alternativos, las Redes Neuronales Artificiales (RNA) cuenta con mayor popularidad y utilización para estos casos, por su capacidad de proporcionar un alto poder de autonomía. Las RNA son una forma de duplicar otra de las características propias de los humanos: la capacidad de memorizar y asociar hechos.

Nuestro país ha dado un gran paso de avance en la rama de la informática a pesar de sus limitaciones por el bloqueo impuesto por el gobierno de los Estados Unidos. Cuba desarrolla un revolucionario proceso de informatización de la sociedad y pretende insertarse en el mercado mundial con el desarrollo de software que apliquen la inteligencia artificial. Diferentes centros e instituciones nacionales han venido trabajando en el tema, la Universidad de las Ciencias Informáticas (UCI) nacida en el calor de la Batalla de Idea cuenta con la facultad número 5 que desarrolla un proyecto llamado “Desarrollo de Elementos Virtuales Inteligentes” que utilizan modelos de redes neuronales por su aplicación en la clasificación de patrones, aproximación de funciones, predicción, control, análisis de datos, compresión de datos, memorización asociativa y categorización de datos, estos modelos se usan en pequeñas aplicaciones, demos y módulos de juegos desarrollados. Se necesita analizar estos modelos de redes y sus funciones ya que no están validados por la teoría de errores, lo que afecta la calidad del proyecto.

Por tanto el **problema a resolver** queda formulado a modo de interrogante de la siguiente forma: ¿Cómo determinar los errores que se cometen en los modelos de Redes Neuronales Artificiales, con sus funciones de transferencias, utilizados en el proyecto “Desarrollo de Elementos Virtuales Inteligentes” de la facultad 5 de la UCI?

Para esta investigación se definió como **Objeto de Estudio** los modelos de Redes Neuronales Artificiales y sus funciones de transferencias utilizados en el proyecto “Desarrollo de Elementos Virtuales Inteligentes” de la facultad 5 de la UCI

El **campo de acción** que abarca esta investigación es la teoría de errores aplicada a los modelos de Redes Neuronales Artificiales y sus funciones de transferencias utilizados en el proyecto “Desarrollo de Elementos Virtuales Inteligentes” de la facultad 5 de la UCI.

El **objetivo general** de esta investigación es identificar los errores que se cometen en los modelos de Redes Neuronales Artificiales y sus funciones de transferencias, utilizados en el proyecto “Desarrollo de Elementos Virtuales Inteligentes” de la facultad 5 de la UCI.

Para dar cumplimiento al objetivo propuesto se plantean las siguientes tareas de investigación:

- Investigar qué es una red neuronal artificial.
- Analizar los modelos de redes neuronales artificiales utilizados en el proyecto “Desarrollo de Elementos Virtuales Inteligentes”.
- Estudiar las funciones de transferencia aplicadas por los modelos de Redes neuronales artificiales.
- Investigar cuáles son los modelos de redes neuronales más utilizados en el proyecto “Desarrollo de Elementos Virtuales Inteligentes” de la UCI.
- Estudiar la teoría de errores.
- Investigar cómo aplicar la teoría de errores a los modelos de redes neuronales y sus funciones de transferencia.
- Identificar los errores que se cometen en los modelos de Redes Neuronales Artificiales y sus funciones de transferencias.

Para el desarrollo de las tareas científicas y una adecuada investigación sobre el tema, fueron seleccionados los siguientes métodos científicos:

Métodos teóricos:

Analítico –Sintético: Se utiliza este método porque se hace un análisis de grandes volúmenes de documentos que hacen referencia a las redes neuronales y la teoría de errores, así como las informaciones y estudios relacionados con este tema. Con la exploración y comparación de la documentación se puede llegar a la esencia, los rasgos, la trayectoria y evolución de este tipo de redes.

Análisis Histórico-Léxico: Se hace uso de este método porque posibilita analizar la evolución de las redes neuronales y las características del periodo de tiempo en que se desarrollaron. Da la posibilidad de ver como surgen, como evolucionan y como se desarrollan este tipo de redes, lo cual es vital para facilitar la aplicación de la teoría de errores a estas redes.

Métodos empíricos:

Entrevista: Se realiza entrevista a los integrantes del proyecto para tener conocimiento de los tipos de modelos de redes neuronales que se utilizan y su funcionamiento.

El contenido de este trabajo se encuentra distribuido de la siguiente forma:

- Capítulo 1: “Fundamentación Teórica”: se hace un análisis bibliográfico donde se investigan las características de los modelos de redes neuronales, las funciones y la teoría de errores.
- Capítulo 2: “Modelos de redes neuronales”: se investiga con profundidad el proceso de aprendizaje de las redes neuronales utilizadas por el proyecto “Desarrollo de Elementos Virtuales Inteligentes”, se analiza además las funciones de transferencias utilizadas por estas redes y los errores cometidos en un sentido general.
- Capítulo 3:” Teoría de errores aplicada a los modelos”: se realiza un análisis de los modelos junto a sus funciones, destacando los posibles errores presentes en estos y algunas sugerencia de cómo disminuirlos.

CAPÍTULO I:” FUNDAMENTACIÓN TEÓRICA”.

1.1 Introducción.

El presente capítulo propone al lector un mejor acercamiento al estudio de los modelos de redes neuronales artificiales, sus características fundamentales, funcionamiento y aplicaciones; así como las pautas de lo que plantea la teoría de errores y las funciones utilizadas por las neuronas artificiales.

1.2 Modelo biológico de redes neuronales.

Hasta nuestra actualidad, en las ciencias y las tecnologías, las redes neuronales han sido objeto de estudio debido a su gran potencialidad para resolver problemas que hasta hace algunos años se necesitaba del ser humano. Estos sistemas, basados en el funcionamiento del cerebro humano y el estudio de la neurona como elemento principal de su funcionamiento, han ocupado un lugar muy popular en las ciencias del conocimiento y el trabajo de procesamiento de señales en la ingeniería informática, entre otros. Nace en el área de neurobiología la idea de imitar la forma de pensamiento humano una vez que se descubren las células nerviosas y la interesante forma de interconexión en forma de red para la transmisión y recepción de señales y estímulos.

El cerebro humano cuenta con millones de neuronas estrechamente relacionadas entre si que reciben señales de múltiples fuentes para ser procesadas y generar una salida de respuesta. Las neuronas son las células que forman la corteza cerebral de los seres vivos y están compuestas por elementos llamados cuerpo, axón y dendritas. Estas últimas forman una estructura de filamentos muy fina que rodean el cuerpo de la neurona. El axón es un tubo largo y delgado que se ramifica en su extremo en pequeños bulbos finales que casi tocan las dendritas de las células vecinas. La pequeña separación entre los bulbos finales y las dendritas se le denomina sinapsis. (A Mauricio Bernal Bohórquez, 2005)

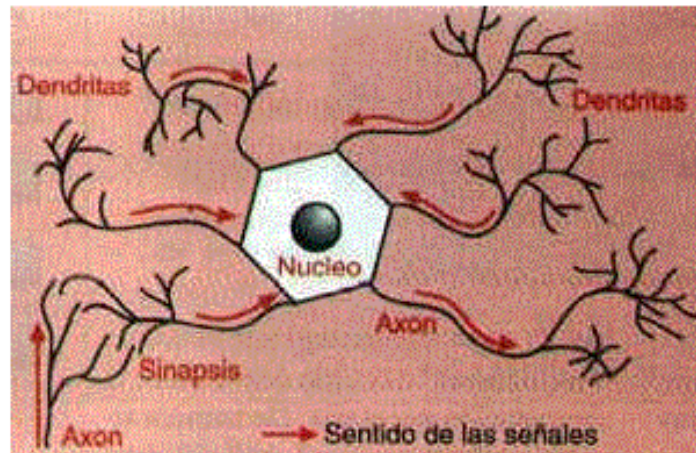


Ilustración 1 Modelo Biológico

De manera similar a como sucede en el resto de las células del cuerpo, las neuronas funcionan a través de impulsos eléctricos y reacciones químicas. El axón se encarga del contacto con las dendritas del resto de las neuronas vecinas a través de las sinapsis y de esta forma es que transmiten los impulsos eléctricos que utilizan las neuronas para el intercambio de información con las demás. Para que una neurona se active y envíe una señal al resto de sus conexiones, es necesario que suficientes neuronas a través de sus dendritas le transmitan en un corto periodo de tiempo impulsos eléctricos. La neurona envía el impulso por el axón y puede ser una señal inhibitoria o excitadora.

1.3 La neurona de lo biológico a lo artificial.

Las neuronas artificiales son modelos que tratan de simular el comportamiento de las neuronas biológicas pero de una forma muy simplificada. La neurona artificial o unidad de proceso como también suele llamarse, es un elemento bastante simple que forma parte de una entidad mayor, la red neuronal, la complejidad radica en los sistemas de interconexión en estas redes. Cada unidad de proceso consta de una serie de Entradas X_i , que equivalen en funcionalidad a las dendritas de donde reciben las señales, ponderadas por los pesos W_i , que representan como los impulsos entrantes son evaluados y se combinan con la función de red que nos dará el nivel de potencial de la neurona. La salida de la función de la red es evaluada en la función de activación que da lugar a la salida de la

unidad de proceso. (Ballesteros) En el siguiente esquema se muestra de forma grafica la estructura general de una neurona artificial.

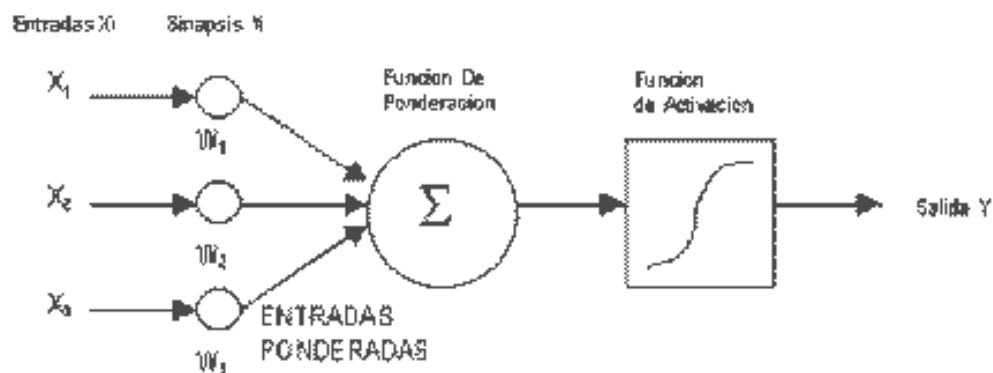


Ilustración 2 Neurona Artificial

En el modelo biológico las neuronas envían las señales a través de las dendritas, en este nuevo modelo se utilizan las entradas X_i con valores reales enteros o binarios. Con la unión de los valores de entrada y sus pesos correspondientes se realiza la misma función de las señales químicas inhibitorias y excitadoras que se dan en las sinapsis y que provocan un cambio en el comportamiento de la neurona. En la neurona biológica existe lo que se conoce como Potencial, que no es más que el total de señales que llegan a la neurona por sus dendritas; en el modelo artificial los valores de entrada y la función de ponderación o red son los que definen este Potencial. La función de ponderación suele ser una la suma ponderada de las entradas y los pesos sinápticos. La salida de función de ponderación llega a la función de activación que transforma este valor en otro en el dominio que trabajen las salidas de las neuronas. El valor de salida cumpliría la función de la tasa de disparo en las neuronas biológicas. (Ballesteros)

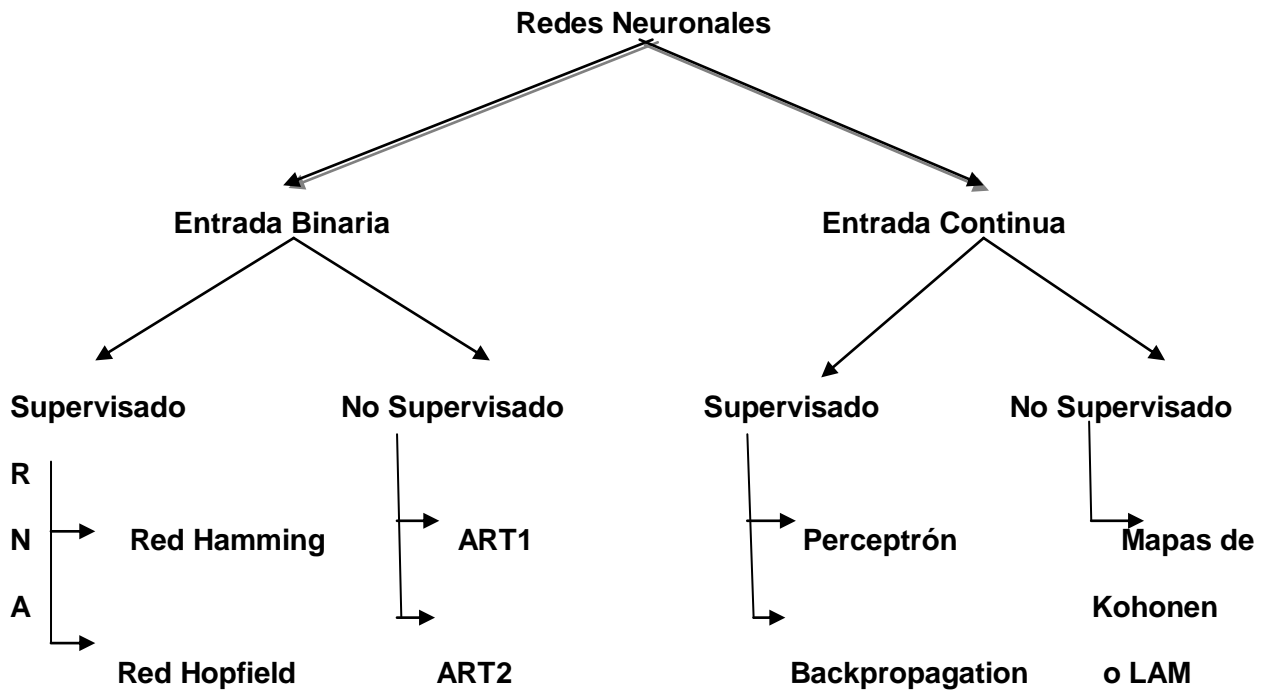
1.4 Modelo de red neuronal artificial.

Las redes neuronales artificiales surgen con la idea de simular el funcionamiento del cerebro humano mediante pequeños elementos de procesamiento fuertemente interconectados entre sí, con el objetivo

de reproducir o imitar el comportamiento del cerebro humano. Las redes neuronales realizan una simplificación, averiguando cuáles son los elementos relevantes del sistema, esto se hace porque la cantidad de información de que disponen puede ser abundante o redundante. Una elección adecuada de sus características, más una estructura conveniente, es el procedimiento convencional utilizado para construir redes capaces de realizar determinadas tareas. (Galdaméz, 2004)

Estas redes son definidas como sistemas de procesamiento de la información, las redes neuronales biológicas son la base de su estructura y funcionamiento. Están formadas por un gran número de elementos simples de procesamiento llamados nodos o neuronas que están organizados en capas. Cada neurona está conectada con otras neuronas mediante enlaces de comunicación, cada uno de los cuales tiene asociado un peso. (PALMER POL) El conocimiento para un determinado problema que tiene una red se encuentra y es determinado por sus pesos. La utilización de las redes neuronales artificiales se encuentra orientada fundamentalmente en dos direcciones de desarrollo; como modelos para el estudio del sistema nervioso y los fenómenos cognitivos, y como herramientas para la resolución de problemas prácticos como la clasificación de patrones y la predicción de funciones. (POL, 2002)

En general las redes neuronales artificiales se pueden clasificar de diversas maneras, según su topología, forma de aprendizaje, tipos de funciones de activación y valores de entradas. En el resto del capítulo se profundizará en estos aspectos.



1.4.1 Elementos de una Red Neuronal Artificial.

1. Unidad de proceso: La neurona artificial es un dispositivo simple de cálculo que a partir de un vector de entrada procedente del exterior o de otras neuronas, proporciona una única respuesta o salida. (González, 1995) Existen tres tipos de unidades en cualquier sistema: entradas, salidas y ocultas. Las unidades de entrada reciben señales desde el entorno, las de salida envían la señal fuera de la red, y las unidades ocultas son aquellas cuyas entradas y salidas se encuentran dentro del sistema. Se conoce como capa o nivel a un conjunto de neuronas cuyas entradas provienen de la misma fuente y cuyas salidas se dirigen al mismo destino. (Ochoa, 2003) Los elementos que conforman una neurona artificial son los siguientes:

- Conjunto de valores de entradas.

- Pesos sinápticos de la neurona W_{ij} : Representan la intensidad de interacción entre cada neurona presináptica j y la neurona postsináptica i . Dada una entrada si el peso es positivo tenderá a excitar a la neurona postsináptica, si el peso es negativo tenderá a inhibirla.
- Regla de propagación: Proporciona el valor del potencial postsináptico de la neurona i en función de sus pesos y entradas.
- Función de activación: Proporciona el estado de activación actual de la neurona i , en función de su estado anterior y de su potencial postsináptico actual.
- Función de salida: Proporciona la salida actual de la neurona i en función de su estado de activación.

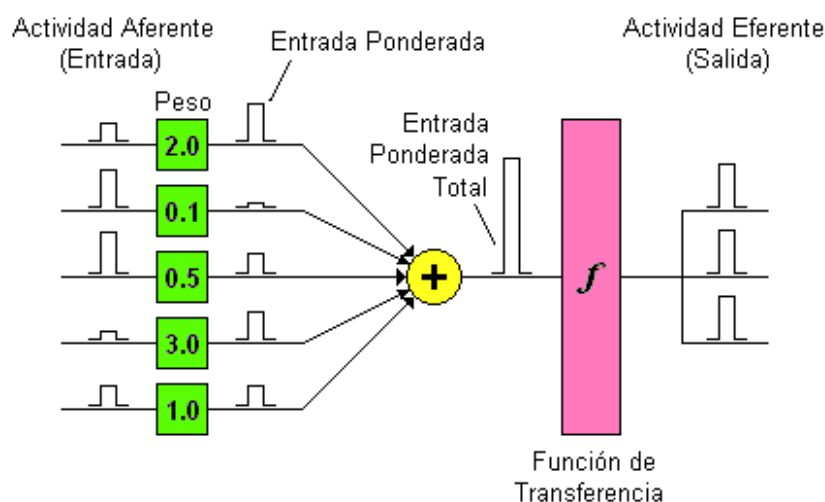


Ilustración 3 Esquema de unidad de proceso

2. Estado de Activación. Los estados del sistema en un tiempo t se representan por un vector $A(t)$. Los valores de activación pueden ser continuos o discretos, limitados o ilimitados. Si son discretos, suelen tomar un conjunto discreto de valores binarios, así un estado activo se indicaría con un 1 y un estado pasivo se representaría por un cero. (Galdaméz, 2004) En otros modelos se considera un conjunto de estados de activación, en cuyo valor entre $[0,1]$, o en el intervalo $[-1,1]$, siendo una función sigmoideal.

3. **Función de Salida o de Transferencia.** Se encuentra asociada a cada unidad y se encarga de transformar el estado actual de activación en una señal de salida. Existen cuatro funciones de transferencia típicas que determinan distintos tipos de neuronas:
 - Función Escalón
 - Función Lineal y Mixta
 - Sigmoidal
 - Función Gaussiana

4. **Conexiones entre neuronas.** Los pesos que tienen asociadas las conexiones que unen a las neuronas de una red neuronal tienen posibilidad de que la red adquiera conocimiento. Se considera que el efecto de cada señal es aditivo, de tal forma que la entrada neta que recibe una neurona es la suma del producto de cada señal individual por el valor de la sinapsis que conecta ambas neuronas y es lo que se conoce como red de propagación. Se utiliza una matriz W con todos los pesos, Si w_{ji} es positivo indica que la relación entre las neuronas es excitadora, es decir, siempre que la neurona i esté activada, la neurona j recibirá una señal que tenderá a activarla. Si w_{ji} es negativo, la sinapsis será inhibitoria. En este caso si i está activada, enviará una señal que desactivará a j . Finalmente si w_{ji} es 0 se supone que no hay conexión entre ambas. (Ochoa, 2003)

5. **Función o Regla de Activación.** Para que una neurona pase a un nuevo estado, es necesario aplicar una regla o función que combine las entradas con el estado actual de la neurona. Con la combinación de las diferentes entradas con los pesos correspondientes a sus conexiones la función genera un nuevo estado de activación a partir del estado que existía en la neurona. Esa función es conocida como la función de activación, y las salidas que se obtienen en una neurona para las diferentes formas de la función serán:
 - Función de Activación Escalón
 - Función de Activación Identidad
 - Función de Activación Lineal -Mixta
 - Función de Activación Sigmoidal

6. **Regla de Aprendizaje.** El aprendizaje puede ser comprendido como la modificación de comportamiento inducido por la interacción con el entorno y como resultado de experiencias conduce al establecimiento de nuevos modelos de respuesta a estímulos externos. (Galdaméz, 2004) En los pesos de las conexiones entre las neuronas en las redes neuronales es donde se

encuentra el conocimiento, no siendo así en el cerebro humano que se encuentra en la sinapsis. Todo proceso de aprendizaje implica cierto número de cambios en estas conexiones. En realidad, puede decirse que se aprende modificando los valores de los pesos de la red. (PALMER POL)

1.4.2 Función de propagación.

La función de propagación puede ser más compleja que simplemente una suma de productos. Las entradas y los pesos pueden ser combinados de diferentes maneras antes de pasarse el valor a la función de activación. Por ejemplo se puede usar como función de propagación, el mínimo, el máximo, la mayoría, producto, o diversos algoritmos de normalización. El algoritmo específico para la propagación de las entradas neuronales está determinado por la elección de la arquitectura. La función más habitual es la suma ponderada de todas las entradas. Podemos agrupar las entradas y pesos en dos vectores $(X_1; X_2; \dots; X_n)$ y $(W_{1j}; W_{2j}; \dots; W_{nj})$ y podemos calcular esta suma realizando el producto escalar sobre estos dos vectores. (Elementos básicos de una Red Neuronal Artificial(II), 2007)

Usualmente se suele clasificar en dos tipos:

Función Lineal de Base: Consiste en la sumatoria de las entradas ponderadas. Se trata de una función de tipo hiperplano, o sea, de primer orden. Es la función más comúnmente utilizada.

Función Radial de Base: El valor de red representa la distancia a un determinado patrón de referencia. Se trata de una función de tipo hiperesférico, de segundo orden y no lineal.

1.4.3 Función de activación o transferencia.

Una función de transferencia es un modelo matemático que entrega la respuesta de un sistema a una señal de entrada o excitación exterior. De manera general se pueden agrupar los siguientes tres tipos de funciones de activación:

Funciones lineales: Proporcionan una salida proporcional a la entrada.

Funciones de umbral: Teniendo en cuenta si la estimulación total que se recibe es superior o no a un valor umbral, entonces se proporciona una salida discreta (típicamente binario 0/1).

Funciones no lineales: Proporcionan una salida no proporcional a la entrada.

En la siguiente tabla se hace una relación de las principales funciones de transferencia empleadas en el entrenamiento de redes neuronales, no son todas las que existen.

Nombre	Relación Entrada-Salida	Función
Limitador Fuerte	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$	hardlim
Limitador Fuerte Simétrico	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$	hardlims
Lineal Positiva	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$	poslin
Lineal	$a = n$	purelin
Lineal Saturado	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$	satlin
Lineal Saturado Simétrico	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = +1 \quad n > 1$	satlins
Sigmoidal Logarítmico	$a = \frac{1}{1 + e^{-n}}$	logsig
Tangente Sigmoidal Hiperbólica	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$	tansig
Competitiva	$a = 1$ $a = 0$	compet

	<i>Neurona con n max</i>	
	<i>El resto de neuronas</i>	

Tabla 1 Funciones

La función de propagación, es transformada en la salida real de la neurona mediante un proceso algorítmico conocido como función de activación. En esta función el valor de la salida de combinación puede ser comparada con algún valor umbral para determinar la salida de la neurona. Si la suma es mayor que el valor umbral, neurona generará una señal. Si la suma es menor que el valor umbral, ninguna señal será generada. Normalmente el valor umbral, o valor de la función de transferencia, es normalmente no lineal. El uso de funciones lineales es limitado ya que el valor de la salida es proporcional al de la entrada, de hecho este fue uno de los problemas en los primeros modelos de redes neuronales como fue identificado por Minsky y Papert en Perceptrons. (Elementos básicos de una Red Neuronal Artificial(Parte II), 2007)

La función de activación pudiera ser algo tan simple como dependiente sobre si el resultado de la función de combinación es positivo o negativo (Elementos básicos de una Red Neuronal Artificial(Parte II), 2007) Al desarrollo de las arquitecturas multicapa se le atribuyen los avances más recientes en conexionismo. Estas utilizan funciones de activación no lineales como una función de umbral, una gaussiana o en la mayoría de los casos una función sigmoideal. El problema de los modelos no lineales radica en que son difíciles de describir en términos lógicos o matemáticos convencionales. Al principio, se pensaba que las neuronas usaban una función de umbral, es decir, que permanecían inactivas y se activaban sólo si la estimulación total superaba cierto valor límite. Después, se comprobó que las neuronas emitían impulsos de actividad eléctrica con una frecuencia variable, dependiendo de la intensidad de la estimulación recibida, y que tenían cierta actividad hasta en reposo, con estimulación nula. Estos descubrimientos llevaron al uso de funciones no lineales con esas características, como la función sigmoideal, con un perfil parecido al escalón de una función de umbral, pero continua.

Por otra parte, se puede añadir cierto nivel de ruido a las entradas antes de aplicar la función de activación, el cual normalmente es conocido como temperatura de la neurona. El modo de entrenamiento de la red en particular es el que determina la fuente y la cantidad de este ruido. Esto permite crear un modelo más parecido al cerebro ya que se añaden diferentes niveles de ruido que da como resultado una combinación más semejante. El uso del ruido por temperatura está aún en fase de

investigación y no es aplicado habitualmente en las aplicaciones. (Elementos básicos de una Red Neuronal Artificial(Parte II), 2007)

El valor de salida de la función de activación puede ser procesado de manera adicional mediante un escalamiento y limitación. El escalamiento simplemente multiplica el valor de la función de transferencia por un factor de escala y después se le suma un desplazamiento. Normalmente este tipo de escalamiento y limitación es usado principalmente en topologías usadas para verificar modelos neuronales biológicos. (Elementos básicos de una Red Neuronal Artificial(Parte II), 2007)

Las funciones de transferencia realizan dos tareas importantes: la primera, es su utilidad para limitar la salida de una neurona, y de esta forma lograr que los posibles resultados no crezcan a valores muy grandes; y la segunda, es que proporciona características de no linealidad, lo cual es muy importante en las redes neuronales artificiales para la ampliación de sus potencialidades y capacidad de respuesta a mayor número de problemas.

1.4.4 Función de salida.

Cada elemento de procesamiento tiene permitido una única salida que puede estar asociada con un número elevado de otras neuronas. Normalmente, la salida es directamente equivalente al valor resultante de la función de activación. Algunas topologías de redes neuronales, sin embargo, modifican el valor de la función de transferencia para incorporar un factor de competitividad entre neuronas que sean vecinas. Las neuronas tienen permitidas competir entre ellas, inhibiendo a otras neuronas a menos que tengan una gran fortaleza. (Elementos básicos de una Red Neuronal Artificial(Parte II), 2007)

1.4.5 Topología de las redes neuronales artificiales.

La arquitectura de las redes neuronales consiste en la organización y disposición de las neuronas formando capas más o menos alejadas de la entrada y salida de la red. En este sentido, los parámetros fundamentales de la red son: el número de capas, el número de neuronas por capa, el grado de conectividad y el tipo de conexiones entre neuronas. (González, 1995)

Redes Monocapa: Se establecen conexiones laterales, cruzadas o autorrecurrentes entre las neuronas que pertenecen a la única capa que constituye la red. Se utilizan en tareas relacionadas con lo que se conoce como autoasociación; por ejemplo, para generar informaciones de entrada que se presentan en la red de forma incompleta o distorsionada. (González, 1995)

Redes Multicapa: Son aquellas que disponen de conjuntos de neuronas agrupadas en varios niveles o capas. Una forma de distinguir la capa a la que pertenece la neurona, consiste en fijarse en el origen de las señales que recibe a la entrada y el destino de la señal de salida. Según el tipo de conexión, como se vio previamente, se distinguen las redes feedforward, y las redes feedforward/feedback. (González, 1995)

- Capa entrada: Recibe los valores de entrada de la red, puesto que en el vector entrada no se lleva a cabo ningún proceso, algunos estudiosos del tema no lo consideran una capa.
- Capas Ocultas: Se definen como capas intermedias que no tienen contacto directo con el medio exterior, la forma de conexión entre sus elementos es la que determina las diferentes topologías de la red.
- Capa de Salida: Recibe la Información de la(a) capa(as) oculta(as) y transmite la respuesta al medio externo.

1.4.6 Principales Modelos de Redes Neuronales Artificiales.

A continuación se describen algunos de los principales modelos de red, aunque no son todos los que existen:

- Perceptrón: Es la base de la mayor parte de la arquitectura de las RNA que se interconectan entre sí. La eficiencia sináptica se representa por factores de peso de interconexión w_{ij} , desde la neurona i , hasta la neurona j . Los pesos pueden ser positivos (excitación) o negativos (inhibición) y junto con las funciones $f(z)$ dictan la operación de la red neuronal. En un perceptrón, cada entrada es multiplicada por el peso W correspondiente, y los resultados son sumados, siendo evaluados contra el valor de umbral, si el resultado es mayor al mismo, el perceptrón se activa. Este modelo sólo puede resolver una función, si todos los posibles

resultados del problema pueden separarse de ésta forma (en dos secciones) es decir, que no se combinen entre sí. (2000)

- Adeline: Muy similar al Perceptrón, excepto en su función de transferencia que es de tipo lineal en lugar de un limitador fuerte como el caso del Perceptrón. Debido a que solo pueden resolver problemas linealmente separables estos dos modelos de red presentan la misma limitación. Sin embargo el algoritmo de Adeline es más potente en cuanto a la regla de aprendizaje, ya que minimiza el error medio cuadrático; característica que lo hace bastante práctico en las aplicaciones de procesamiento de señales digitales, por ejemplo las líneas telefónicas de gran distancia la utilizan para cancelar el ruido inherente a su recorrido. Este modelo consta de un único elemento de procesamiento, técnicamente no es una red neuronal, el cual se encarga de realizar la suma de los productos de los vectores de entrada y de pesos, y aplica una función de salida para obtener un único valor de salida.
- Backpropagation: El perceptrón solo es el ejemplo más elemental de una red neuronal, de hecho, no puede siquiera ser considerado una "red", puesto que no intervienen otros elementos. Si se combinan varios perceptrones en una "capa", y los estímulos de entrada después se suman tendremos ya una red neuronal. Una red neuronal muy eficaz para resolver fundamentalmente problemas de reconocimiento de patrones es la red neuronal de propagación hacia atrás. En este modelo red, se interconectan varias unidades de procesamiento en capas, las neuronas de cada capa no se interconectan entre sí. Sin embargo, cada neurona de una capa proporciona una entrada a cada una de las neuronas de la siguiente capa, esto es, cada neurona transmitirá su señal de salida a cada neurona de la capa siguiente. (2000)
- Hopfield: Las redes de Hopfield son redes de adaptación probabilística, recurrentes y completamente conectadas, aprenden a reconstruir los patrones de entrada que memorizaron durante el entrenamiento. Funciona como una memoria asociativa no lineal que puede almacenar internamente patrones presentados de forma incompleta o con ruido. De esta forma puede ser usada como una herramienta de optimización. El estado de cada neurona puede ser actualizado un número indefinido de veces, independientemente del resto de las neuronas de la

red pero en paralelo. Mientras que las redes en cascada (no recurrentes) dan soluciones estables, los modelos recurrentes dan soluciones inestables (dinámicas), lo que no siempre es aconsejable. (2000) El principal aporte de Hopfield consistió precisamente en conseguir que tales redes recurrentes fueran así mismo estables.

- Kohonen: Existen evidencias que demuestran que en el cerebro existen neuronas que se organizan en muchas zonas, de forma que las informaciones captadas del entorno a través de los órganos sensoriales se representan internamente en forma de capas bidimensionales. Aunque en gran medida esta organización neuronal está predeterminada genéticamente, es probable que de ella se origine mediante el aprendizaje. El cerebro parece poseer la capacidad inherente de formar mapas topológicos de las informaciones recibidas del exterior. Este modelo pertenece a la categoría de las redes competitivas o mapas de auto organización, es decir, aprendizaje no supervisado. El objetivo de este tipo de redes es clasificar los patrones de entrada en grupos de características similares, de manera que cada grupo activará siempre la(s) misma(s) salida(s). Las unidades de entrada reciben datos continuos normalizados, se normalizan así mismo los pesos de las conexiones con la capa de salida. Tras el aprendizaje de la red, cada patrón de entrada activará una única unidad de salida. Una vez entrenada, podemos usar a la red para clasificar patrones de entrada similares en el espacio n-dimensional. Una clase o grupo de patrones similares tiende a controlar una neurona específica, que representará el centro de una esfera n-dimensional. Esa neurona resultará la más activada frente a los patrones más parecidos a su vector de pesos. (2000)

1.4.7 Mecanismo de Aprendizaje de las Redes Neuronales Artificiales.

El aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada.

Durante el aprendizaje se producen cambios como son la destrucción, modificación y creación de conexiones entre las neuronas. La creación de una nueva conexión implica que el peso de la misma pasa a tener un valor nuevo valor, una conexión se destruye cuando su peso pasa a ser cero. Se

puede afirmar que el proceso de aprendizaje ha finalizado, o sea, que la red ha aprendido a reconocer los patrones deseados cuando los valores de los pesos permanecen estables: se cumple que $dw_{ij}/dt=0$.

Un criterio 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. Otro criterio suele considerar dos tipos de reglas de aprendizaje: las de aprendizaje supervisado y las correspondientes a un aprendizaje no supervisado, estas reglas dan pie a una de las clasificaciones que se realizan de las RNA: Redes neuronales con aprendizaje supervisado y 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 aprendizaje de la red. (González, 1995)

1.4.7.1 Redes con Aprendizaje Supervisado

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. (Galdaméz, 2004) Con el fin de conseguir que la salida se aproxime lo más posible a la deseada se comprueba la salida de la red y en el caso de que ésta no coincida con la deseada, se procederá a modificar los pesos de las conexiones,

Se consideran tres formas de llevar a cabo este tipo de aprendizaje:

- Aprendizaje por corrección de error: Consiste en ajustar los pesos en función de la diferencia entre los valores deseados y los obtenidos en la salida de la red; es decir, en función del error. (Galdaméz, 2004)
- Aprendizaje por refuerzo: Se basa en la idea de no indicar durante el entrenamiento exactamente la salida que se desea que proporcione la red ante una determinada entrada. La función del supervisor se reduce a indicar mediante una señal de refuerzo si la salida obtenida en la red se ajusta a la deseada (éxito=+1 o fracaso=-1), y en función de ello se ajustan los pesos basándose en un mecanismo de probabilidades. (Galdaméz, 2004)

- Aprendizaje estocástico: En este tipo de aprendizaje se realizan cambios aleatorios en los valores de los pesos de las conexiones de la red y se evalúa que efecto provocan en la salida a partir del objetivo que se desea y de distribuciones de probabilidad.

1.4.7.2 Redes con Aprendizaje No Supervisado

En estas redes el ajuste de los pesos de las conexiones entre neuronas no requiere una influencia externa. Se dice que estas redes son capaces de autoorganizarse puesto que no reciben ninguna información por parte del entorno que le indique si la salida que se genera es o no correcta y esto provoca que existan varias posibilidades en cuanto a la interpretación de la salida. Existen varias posibilidades en cuanto a la interpretación de lo antes expuesto, depende de su estructura y del algoritmo de aprendizaje empleado para su entrenamiento. Este tipo de red debe encontrar las características, regularidades, correlaciones o categorías que se puedan establecer entre los datos de entrada.

Para algunos casos con la salida se determina el grado de familiaridad o similitud entre la información que se le ha mostrado con anterioridad y la que se le está presentando en la entrada. En otro caso podría realizar una codificación de los datos de entrada, generando a la salida una versión codificada de la entrada, con menos bits, pero manteniendo la información relevante de los datos, o algunas redes con aprendizaje no supervisado lo que realizan es un mapeo de características, obteniéndose en las neuronas de salida una disposición geométrica que representa un mapa topográfico de las características de los datos de entrada, de tal forma que si se presentan a la red informaciones similares, siempre sean afectadas neuronas de salidas próximas entre sí, en la misma zona del mapa. (Galdaméz, 2004)

En general en este tipo de aprendizaje se suelen considerar dos tipos:

- Aprendizaje competitivo y cooperativo: Las neuronas compiten y cooperan unas con otras con el fin de llevar a cabo una tarea dada. Con este tipo de aprendizaje se pretende que cuando se presente a la red cierta información de entrada, solo una de las neuronas de salida se active (alcance su valor de respuesta máximo). Por tanto las neuronas compiten por activarse, quedando finalmente una, o una por grupo, como neurona vencedora (Galdaméz, 2004)

- Aprendizaje asociativo (Hebbiano): Consiste básicamente en el ajuste de los pesos de las conexiones de acuerdo con la correlación, así si las dos unidades son activas (positivas), se produce un reforzamiento de la conexión. Por el contrario cuando una es activa y la otra pasiva (negativa), se produce un debilitamiento de la conexión. (Galdaméz, 2004)

En estas redes asociativas se ha evitado el uso de redes complejas, con este fin se han definido dos tipos de estímulos:

Estímulo no condicionado: Corresponde a la entrada, refuerza el aprendizaje y ayuda a hacer la asociación con la salida deseada, se presenta de forma intermitente el estímulo para simular el proceso real de aprendizaje y memorización de la red; la mayoría de las veces el estímulo no condicionado se convierte en la salida deseada de la red.

Estímulo condicionado: Debe ser siempre presentado a la red y ésta debe asociarlo con la salida deseada; al final del proceso de aprendizaje la red debe ser capaz de entregar la respuesta correcta con la presentación de este único estímulo a su entrada, sin importar si el estímulo no condicionado ha sido presentado o no, pues la asociación ya ha sido realizada.

1.4.8 Entrada y Salida en las Redes Neuronales Artificiales.

Las RNA son sistemas que almacenan cierta información aprendida; esta información se registra de forma distribuida en los pesos asociados a las conexiones entre neuronas de entrada y salida. Existen dos formas primarias de realizar esa asociación de entrada/salida. (Galdaméz, 2004) Una primera sería la denominada heteroasociación, que se refiere al caso en el que la red aprende parejas de datos $[(A_1, B_1), (A_2, B_2) \dots (A_n, B_n)]$, de tal forma que cuando se presente cierta información de entrada A_i , deberá responder generándola correspondiente salida B_i . Por otra parte la segunda se conoce como autoasociación, donde la red aprende ciertas informaciones $A_1, A_2 \dots A_n$, de tal forma que cuando se le presenta una información de entrada realizará una autocorrelación, respondiendo con el dato almacenado más parecido al de la entrada.

Estos dos mecanismos de asociación dan lugar a dos tipos de redes neuronales: las redes heteroasociativas y las autoasociativas. (neuronales) Una red heteroasociativa podría considerarse aquella que computa cierta función, 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. Existen redes heteroasociativas con conexiones feedforward, feedforward/feedback y redes con conexiones laterales. También existen redes heteroasociativas multidimensionales y su aprendizaje puede ser supervisado o no supervisado. (Galdaméz, 2004)

Una red autoasociativa tiene como principal misión reconstruir una determinada información de entrada que se presenta incompleta o distorsionada, para esto le asocia el dato almacenado más parecido. Pueden implementarse con una única capa donde existen conexiones laterales o también autorrecurrentes, normalmente su aprendizaje es no supervisado.

Las redes neuronales artificiales también pueden clasificarse de acuerdo a de la forma en que se representan las informaciones de entrada y las respuestas o datos de salida. En muchas redes tanto los datos de entrada como de salida son de naturaleza analógica, cuando esto ocurre, las funciones de activación de las neuronas serán también continuas, del tipo lineal o sigmoideal. En otras redes sólo se admiten valores discretos o binarios en su entrada, generando también respuestas de tipo binario; en estos casos, las funciones de activación de las neuronas son de tipo escalón. Existe también un tipo de redes híbridas donde los de entrada pueden ser valores continuos, aunque las salidas de la red sean discretas.

1.4.9 Ventajas de las Redes Neuronales Artificiales.

El aprendizaje adaptativo es una de las características que más se destaca en el uso de las redes neuronales. Basadas en un entrenamiento o una experiencia inicial las redes tienen la capacidad de aprender a realizar diferentes tareas. Las RNA no necesitan un algoritmo específico para resolver problemas, durante su aprendizaje, los enlaces ponderados de las neuronas se ajustan de manera que se obtengan resultados específicos, en otras palabras, son capaces de generar su propia distribución de los pesos de los enlaces mediante este aprendizaje. También existen redes que aunque han completado su periodo inicial de entrenamiento continúan aprendiendo continuamente. La función del diseñador es únicamente la obtención de la arquitectura apropiada. Aunque para el diseñador el cómo la red aprenderá a discriminar no es significativo, si es necesario que desarrolle un buen algoritmo de

aprendizaje que proporcione la capacidad de discriminar de la red mediante un entrenamiento con patrones.

La información que reciben durante la etapa de aprendizaje es organizada por la propia red. Esta capacidad de autoorganizarse posibilita a las redes neuronales responder apropiadamente cuando se les presentan informaciones o situaciones con las que no habían interactuado con anterioridad.

Si se comparan con los sistemas computacionales tradicionales, los cuales pierden funcionalidad si ocurre un pequeño error de memoria, se notara que por el contrario en las redes neuronales, si se produce un fallo en un pequeño número de neuronas, no sufre una caída repentina aunque el comportamiento del sistema si se ve influenciado por esto. Las redes pueden aprender a reconocer patrones con ruido, distorsionados, o incompletos. Además pueden seguir realizando su función con cierto nivel de degradación aunque se destruya parte de la red. Al estar la información distribuida en las conexiones entre las neuronas las redes neuronales son más tolerantes a fallos. Es cierto también que puede existir redundancia en esta forma 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.

Hoy en día los computadores neuronales se diseñan y fabrican con hardware especial para obtener la capacidad de operar en tiempo real que puedan operar de forma paralela. La inserción de las RNA en las tecnologías actuales para aplicaciones específicas dentro de los sistemas existentes es de vital importancia. Debido a que una red puede ser rápidamente entrenada, comprobada, verificada y trasladada a una implementación hardware de bajo costo, lo antes planteado gana en facilidad. De esta manera, las redes neuronales se pueden utilizar para mejorar sistemas de forma incremental, y cada paso puede ser evaluado antes de acometer un desarrollo más amplio.

1.4.10 Aplicaciones de las Redes Neuronales Artificiales.

Las redes neuronales son una tecnología computacional emergente que puede utilizarse en un gran número y variedad de aplicaciones, tanto comerciales como militares. Hay muchos tipos diferentes de redes neuronales, cada uno de los cuales tiene una aplicación particular más apropiada. Separándolas según las distintas disciplinas algunos ejemplos de sus aplicaciones son: (Galdaméz, 2004)

Biología	<ul style="list-style-type: none"> • Aprender más acerca del cerebro y otros sistemas. • Obtención de modelos de la retina.
Empresa	<ul style="list-style-type: none"> • Reconocimiento de caracteres escritos. • Identificación de candidatos para posiciones específicas. • Optimización de plazas y horarios en líneas de vuelo. • Explotación de bases de datos. • Evaluación de probabilidad de formaciones geológicas y petrolíferas. • Síntesis de voz desde texto.
Medio Ambiente	<ul style="list-style-type: none"> • Analizar tendencias y patrones. • Previsión del tiempo.
Finanzas	<ul style="list-style-type: none"> • Previsión de la evolución de los precios. • Valoración del riesgo de los créditos. • Identificación de falsificaciones. • Interpretación de firmas.
Manufacturación	<ul style="list-style-type: none"> • Robots automatizados y sistemas de control (visión artificial y sensores de presión, temperatura, gas, etc.) • Control de producción en líneas de proceso. • Inspección de calidad. • Filtrado de señales.
Medicina	<ul style="list-style-type: none"> • Analizadores del habla para la ayuda de audición de sordos profundos. • Diagnóstico y tratamiento a partir de síntomas y/o de datos analíticos (encefalograma, etc.). • Monitorización en cirugía. • Predicción de reacciones adversas a los medicamentos. • Lectoras de Rayos X. • Entendimiento de causa de ataques epilépticos.
Militares	<ul style="list-style-type: none"> • Clasificación de las señales de radar. • Creación de armas inteligentes.

	<ul style="list-style-type: none">• Optimización del uso de recursos escasos.
--	---

Tabla 2 Aplicación de las RNA

1.4.10.1 Ejemplos de Aplicaciones que utilizan las RNA.

- **Attrasoft**

Se al desarrollo de software para el reconocimiento de imágenes a través de redes neuronales. Sus productos están diseñados para trabajar en un ambiente entre 1,000 y 100,000 neuronas externas de entrada y salida. Los programas de Attrasoft se utilizan para clasificar imágenes, buscar imágenes en internet, obtener predicciones sobre una serie de datos, reconocimiento de patrones, entre otras potencialidades. En general son completamente configurables: se puede modificar la cantidad de neuronas empleadas, y su entrenamiento, por lo que se puede adaptar a casi cualquier tipo de problema. Estas aplicaciones son muy utilizadas por entidades como el FBI y la Interpol en la búsqueda de sospechosos en las bases de datos de criminales registrados. (neural@electronica.com.mx, 2000)

- **PANDORA**

La línea de programas PANDORA de Prosoniq, esta diseñada para extraer de una señal de audio completa, los componentes musicales básicos mediante el Procesamiento Digital de Señales y el uso de Redes Neuronales Artificiales. Estos sistemas trabajan con archivos monoaurales y no modifican en nada a los demás instrumentos a parte de la voz; solo mantiene la reverberación de la misma, debido a que se crea artificialmente y se manipula como una mezcla de señales.

Estas aplicaciones brindan la posibilidad de incrementar o atenuar sonidos como la voz de los cantantes hasta que no sea escuchada (ambiente Karaoke) y además están diseñados para trabajar en tiempo real con señales de audio mono o estéreo. También se pueden utilizar para ciencias forenses, al extraer componentes básicos de una grabación en casos importantes. De esta forma se pueden eliminar el ruido existente en una grabación u otros componentes para dejar solamente las pruebas necesarias para el caso. (neural@electronica.com.mx, 2000)

- **Neural Stock**

Es un programa que combina las Redes Neuronales Artificiales, junto con tecnologías como Algoritmos Genéticos y Lógica Difusa, para obtener una capacidad de análisis acertada, recomendando operaciones de compra y venta en el mundo financiero y bursátil. Con este sistema se analiza el desarrollo del mercado financiero, día tras día, con lo que se logra entrenar a la red, y en base a todo su historial de datos, puede predecir el siguiente movimiento a efectuar, actuando como un asesor en la compra y venta lo cual permite maximizar la rentabilidad de las inversiones tanto como sea posible.

Se convierte en una herramienta muy poderosa al aprovechar las ventajas de internet al tener los datos de la bolsa en tiempo real o al final de cada día. En el mercado de los inversionistas, es necesario tener herramientas como ésta, ya que aunque da muy buenas sugerencias, no se "emociona" sino que permanece objetivo y frío, a diferencia de lo que pudiera suceder con una persona, al seguir alguna corazonada. (neural@electronica.com.mx, 2000)

1.5 Teoría de Errores.

Por mucha diligencia y cuidado al realizar cualquier medición, y por muy sensibles y precisos que sean los medios utilizados, es prácticamente imposible el evitar errores, considerando a éstos como la variación entre los valores hallados y el real o verdadero, el cual generalmente no es conocido. Tampoco el error, aunque fuera conocido, nos daría una medida cierta de su importancia, ya que esta dependerá no de la magnitud de dicho error, sino de la magnitud de la medida a valorar y de la necesidad de aproximación a su valor real.

El resultado de toda medición siempre tiene cierto grado de incertidumbre. Esto puede estar determinado por diferentes razones: las limitaciones de los instrumentos de medida, las condiciones en que se realiza la medición y/o las capacidades del experimentador. Es por ello que para tener una idea correcta de la magnitud con la que se está trabajando, es indispensable establecer los límites entre los cuales se encuentra el valor real de dicha magnitud.

Cuando se mide una cantidad, ya sea directa o indirectamente, la medida que se obtiene no es necesariamente el valor exacto de tal medida, ya que el resultado obtenido estará afectado por errores debidos a multitud de factores. Algo en apariencia tan sencilla como cronometrar el período de un

péndulo sufrirá errores debidos a la precisión del cronómetro, los reflejos del cronometrador, las corrientes de aire, el número de medidas efectuadas, entre otros errores que se propagan. En estos casos es necesario estimar el error cometido al efectuar una medida o serie de medidas. El conjunto de reglas matemáticas dedicado a su estudio se conoce como teoría de errores, y resulta imprescindible tanto para sacar todo el partido posible a un conjunto de datos experimentales como para evaluar la fiabilidad y confiabilidad de éstos. (Apuntes de Física. TEORÍA DE ERRORES)

No se puede conocer el valor exacto de una cantidad, puesto que siempre existen imprecisiones; tampoco se puede conocer el valor exacto de un error, puesto que dependen de procesos aleatorios y generalmente incontrolables. Sin embargo, es preciso dar un valor de la medida con su error, La Teoría de Errores deduce ciertas reglas para este análisis.

1.5.1 La Incertidumbre y el Error.

El término error se define como la diferencia entre el valor verdadero y el obtenido experimentalmente. Aunque existen importantes diferencias entre la incertidumbre y el error, estos dos conceptos están relacionados entre sí ya que la primera debe considerar todas las posibles fuentes de error del proceso de medida. Si se toma por ejemplo el caso de que un resultado tenga un error despreciable, debido a que, este resultado puede estar muy próximo al valor considerado verdadero, y por otra parte, la incertidumbre de este resultado puede ser muy elevada simplemente porque el analista está inseguro del resultado que ha obtenido debido al gran número de fuentes de error que puede tener el método analítico. Puede darse el caso que el error cometido en un caso 1 hipotético sea mucho mayor que el cometido en un caso 2, pero que la incertidumbre asociada al analizar ambos casos sea la misma porque se ha utilizado el mismo método analítico.

1.5.2 Clasificación de los errores.

Los errores no siguen una ley determinada y su origen está determinado por múltiples causas las cuales para su mejor entendimiento se pueden clasificar en dos grandes grupos, errores sistemáticos y errores accidentales.

EL primero de estos errores es aquel que es constante a lo largo de todo el proceso de medición, por lo cual afecta a todas las mediciones de un modo definido y es el mismo para todas ellas. Entre las causas más probables de este tipo de error están las siguientes:

- Errores instrumentales (de aparatos). Por ejemplo el error de calibrado es de este tipo.
- Error personal. Este es, en general, difícil de determinar y es debido a limitaciones de carácter personal. Un ejemplo de éste sería una persona con un problema de tipo visual o nervioso.
- Error de la elección del método. Corresponde a una elección inadecuada del método de medida de la magnitud. Este tipo de error puede ponerse de manifiesto cambiando el aparato de medida, el observador, o el método de medida.

El segundo tipo de errores son aquellos que se producen en las pequeñas variaciones que aparecen entre mediciones sucesivas realizadas por un mismo operador. Las variaciones no son reproducibles de una medición a otra, y no presentan más que por azar la misma magnitud en dos mediciones cualesquiera del grupo. Las causas fundamentales de estos errores son incontrolables para un observador. En su mayoría son de magnitud muy pequeña y para un gran número de mediciones se obtienen tantas desviaciones positivas como negativas. Aunque con los errores accidentales no se pueden hacer correcciones para obtener valores más concordantes con el real, si se emplean varios métodos estadísticos se puede llegar a algunas conclusiones relativas al valor más probable en un conjunto de mediciones.

Dentro de los errores más comunes se encuentran:

Error de redondeo. Se origina por el hecho de que una computadora sólo puede representar un número finito de términos. Para expresar una cantidad con un desarrollo decimal infinito, se tiene que prescindir de la mayoría de ellos; o sea, se presenta y se opera el número con menos cifras de las que realmente posee.

Error de truncamiento. Se originan al emplear al número finito de términos para calcular un valor que requiere un número infinito de términos.

Error medio cuadrático. Se define como la sumatoria de la diferencia entre las variables reales y estimadas al cuadrado, para evitar que en la función objetivo aparezcan diferencias negativas que inviertan el objetivo de la función.

El error aleatorio. Es aquel error que se produce independientemente de los sucesos únicos imposibles de controlar durante el proceso de medición. Se contrapone al concepto de error sistemático.

Error sistemático. Es el error que se produce de igual manera en todas las mediciones que se realizan de una magnitud. Puede estar originado en una particularidad del operador o del proceso de medición.

1.5.3 Error Absoluto y Error Relativo.

Para el trabajo práctico es necesario emplear medidas cuantitativas de los errores. Para ello se utilizan los términos de Error Absoluto y Error Relativo.

Si al realizar una medición obtenemos un valor de medida X y su valor verdadero es X_0 , entonces se conoce como error absoluto en dicha medida, a la diferencia: $\Delta X = X - X_0$; donde en general se supone que $|\Delta X| \ll |X_0|$. Este error nos da una medida de la desviación, en términos absolutos respecto al valor verdadero. En ocasiones nos interesa además resaltar la importancia relativa de esa desviación y para esto se usa el error relativo. Este otro error se define como el cociente entre el error absoluto y el valor verdadero: $\varepsilon = \Delta X / X_0$ en forma porcentual se expresará multiplicado por cien. Cuando indiquemos el valor de una medida, tendremos que indicar siempre el grado de incertidumbre de la misma, para lo que acompañaremos el resultado de la medida del error absoluto de la misma, expresando el resultado en la forma: $x \pm \Delta x$.

El error absoluto dado el significado de cota de imprecisión que tiene no debe tener más de dos cifras significativas, admitiéndose por convenio, que el error absoluto sólo puede darse con dos cifras significativas si la primera de ellas es un 1, o si siendo la primera un 2, la segunda no llega a 5. En todos los demás casos debe darse un valor con una sola cifra, aumentando la primera en una unidad si la segunda fuera 5 o mayor que 5. Los valores deben tener sólo las cifras necesarias para que su última cifra significativa sea del mismo orden decimal que la última del error absoluto, llamada cifra de acotamiento.

El error absoluto máximo de un número aproximado a cualquier número positivo mayor o igual que el error absoluto del número aproximado.

1.5.4 Propagación de errores.

Se origina debido a que se trabaja con valores que anteriormente tenían cotas de errores, evidentemente el valor final se verá afectado. Cuando se realizan operaciones con números aproximados, sus errores aparecen reflejados en el error del número resultante. Este efecto de transmisión del error de un número a otro, puede ser mayor o menor en dependencia de las operaciones y el orden en que se realicen, y se le conoce con el nombre de *propagación del error*.

Propagación del error en la adición

- El error absoluto de una suma de varios números aproximados es menor o igual que la suma de los errores absolutos de dichos números.
- El error absoluto máximo de una suma algebraica puede tomarse como la suma de los errores absolutos máximos de los sumandos.
- El error relativo máximo de una suma de números positivos puede tomarse como el mayor de los errores relativos máximos de los sumandos.

Propagación del error en la multiplicación

- El error relativo máximo de la multiplicación de números aproximados es igual a la sumatoria de los errores relativos máximos de los valores sumado a la multiplicación de dichos errores.
- El error absoluto máximo de de la multiplicación de números aproximados es igual al modulo de la multiplicación de dichos valores multiplicado por el error relativo máximo.

Propagación del error en la división

- El error relativo máximo del cociente de dos números aproximados puede tomarse como las suma de los errores relativos máximos de cada operando.

Propagación del error en la potenciación

- El error relativo máximo de un número aproximado elevado a un exponente positivo se puede tomar como la multiplicación del exponente con el error relativo máximo de del número.

1.5.6 Importancia de la teoría de errores en la informática.

En esfera informática el análisis de los errores es de gran utilidad para el desarrollo de esta ciencia, ya que la correcta aplicación optimiza los resultados de una investigación o perfecciona el trabajo en la utilización de los medios de informáticos. Pequeños errores en los códigos o software, o simplemente descuidos en la introducción o manipulación de los datos en el mundo informático, puede causar grandes y graves problemas y la pérdida de millonarias sumas o de vidas humanas, de aquí la vital importancia de minimizar tanto como sea posible los errores en las aplicaciones de informática. A continuación algunos ejemplos reales de los principales errores informáticos, conocidos como bug, que han provocado durante la historia grandes daños a nuestras sociedades.

Un "bug" en el software de vuelo de la sonda Mariner I provocó que, segundos después del lanzamiento de la nave, en julio de 1972, ésta se desviara de su curso preestablecido. Los responsables de la misión se vieron obligados a destruir el cohete cuando se encontraba sobrevolando el Atlántico. La investigación del accidente determinó que el problema estaba en una fórmula escrita a lápiz que luego fue inadecuadamente trasladada al lenguaje informático, lo que hizo que el cohete calculara mal la trayectoria que debía seguir. (Carletti, 2005)

El Therac-25 era un acelerador lineal empleado en los hospitales en la década de los 80 para tratar tumores. La máquina emitía radiación de alta energía sobre células cancerosas sin dañar el tejido circundante. Los operarios, con el tiempo y la práctica, conseguían gran velocidad tecleando la secuencia de comandos para iniciar un tratamiento. Pero debido a un fallo de programación, si durante este proceso efectuaban una corrección en menos de ocho segundos, la máquina podía emitir 100 veces más energía de la requerida. A consecuencia de este "bug" murieron al menos cinco pacientes y varias decenas sufrieron los efectos de verse expuestos a una elevada radiación. (Carletti, 2005)

Un problema con los microprocesadores provocó que éstos fallaran cuando tenían que dividir números con coma flotante. Por ejemplo, al dividir 4195835.0 entre 3145727.0 el resultado que

mostraba el microprocesador era 1.33374 en lugar de 1.33382, un error del 0.006%. Aunque el fallo afectaba a pocos usuarios, resultó todo un problema para Intel, que terminó por cambiar entre tres y cinco millones de chips, en una operación que le costó más de 475 millones de dólares. (Carletti, 2005)

No hay "bugs" tan conocidos como el del "efecto 2000", o el error que acabó con la Mars Climate estrellada contra el suelo de Marte, debido a un error en la conversión al Sistema Internacional de unidades —para pasar de millas a kilómetros— de los datos que se habían suministrado a su computadora. (Carletti, 2005)

CAPÍTULO II: “DESCRIPCIÓN DE LOS MODELOS DE RNA”.

2.1 Introducción

En este capítulo se realiza un estudio sobre los tres modelos de redes neuronales más comúnmente utilizados en el proyecto “Desarrollo de Elementos Virtuales Inteligentes” y las funciones de transferencias utilizadas por estos modelos. Se profundizará en sus características principales, sus estructuras y funcionamiento.

2.2 Selección de los modelos de redes neuronales artificiales.

Para la selección de los modelos a estudiar se realizó una entrevista a algunos de los miembros del proyecto “Desarrollo de Elementos Virtuales Inteligentes”, para conocer sobre las redes neuronales que más se utilizan en este proyecto y la justificación. De esta entrevista se obtuvieron las siguientes conclusiones:

El perceptrón multicapa es uno de los modelos de red neuronal más conocido hoy en día. Ha sido aplicado exitosamente en la solución de disímiles problemas, y este éxito se extiende a la industria de los juegos (Ej.: Battle-Cruiser: 3000AD, Black & White, Creatures, Dirt Track Racing, y Heavy Gear).

Ellos ofrecen algunas ventajas importantes sobre las técnicas tradicionales de inteligencia artificial. Primero, usando redes neuronales, los desarrolladores simplifican el código de máquinas finitas de estados o de sistemas basados en reglas, relegando las decisiones a uno o más procesos de entrenamiento. Segundo, las redes neuronales ofrecen el potencial para que los juegos de inteligencia artificial se adapten a la manera en que son jugados.

Los perceptrones multicapa resuelven problemas de cualquier complejidad, lo que hace posible su aplicación a disímiles escenarios dentro de los juegos. Su principal limitante para ser utilizadas en escenarios de realidad virtual, es su costo computacional, pero se pueden obtener topologías de perceptrones capaces de ejecutarse en tiempo real.

Las redes de Backpropagation son tipos de perceptrones multicapa, poseen n capas, y utilizan aprendizaje supervisado del tipo Backpropagation. Este tipo de aprendizaje es lo que las caracteriza como modelos. La ventaja del método de entrenamiento por Backpropagation, que es uno de los más populares entre los aprendizajes supervisados, es que garantiza la convergencia de la red durante el proceso de entrenamiento, y sobre todo, que se pueden realizar en tiempo real, lo que garantiza el aprendizaje de los agentes que utilicen el modelo cuando la aplicación se está ejecutando.

El modelo Adeline es un modelo muy sencillo, posee una única capa de neuronas, y solamente clasifica regiones que sean separables. La ventaja principal de este modelo es la velocidad en el procesamiento de datos y su capacidad de resolver problemas con alto nivel de inteligencia, pero sin llegar a ser altamente impredecibles.

Los modelos más utilizados en este proyecto son el perceptrón, la Adeline y la red Backpropagation.

2.3 Modelos de Redes Neuronales Perceptrón.

La red modelo perceptrón fue creada en el año 1957 por el psicólogo Frank Rosenblatt, con la intención de enseñar algunas propiedades fundamentales de los sistemas inteligentes en general. Rosenblatt opinaba que la herramienta de análisis más apropiada era la teoría de probabilidades, esto lo llevó a una teoría de separabilidad estadística que utilizaba para caracterizar las propiedades más visibles de estas redes de interconexión ligeramente aleatoria. El primer modelo fue desarrollado en un ambiente biológico imitando al funcionamiento del ojo humano, el fotoperceptrón como se le llamó, era un dispositivo que respondía a señales ópticas.

El Perceptrón era inicialmente un dispositivo de aprendizaje, en su configuración inicial no estaba en capacidad de distinguir patrones de entrada muy complejos, sin embargo mediante un proceso de aprendizaje era capaz de adquirir esta capacidad. Su entrenamiento implicaba un proceso de refuerzo mediante el cual la salida de las unidades de asociación se incrementaba o se decrementaba dependiendo de si las unidades de asociación contribuían o no a las respuestas correctas del Perceptrón para una entrada específica.

El Perceptrón, al constar de una sola capa de entrada y otra de salida con una única neurona, tiene una capacidad de representación bastante limitada, este modelo sólo es capaz de discriminar patrones

muy sencillos. Este es uno de los principales problemas que tiene esta red, su incapacidad para solucionar problemas que no sean linealmente separables. (1)A pesar de esta limitación, es aún hoy, un modelo de red de gran importancia, pues con base en su estructura se han desarrollado otros modelos de red neuronal como las redes Adeline y las multicapas.

2.3.1 Estructura del modelo de red.

La única neurona de salida del perceptrón realiza la suma ponderada de las entradas, resta el umbral y pasa el resultado a una función de transferencia de tipo escalón. La regla de decisión es responder +1 si el patrón presentado pertenece a una clase A, o -1 si el patrón pertenece a una clase B, la salida depende de la entrada neta, el valor de n es igual a la suma de las entradas ponderadas y W el valor de los pesos que cada una de las entradas de las neuronas en cada una de las capas.

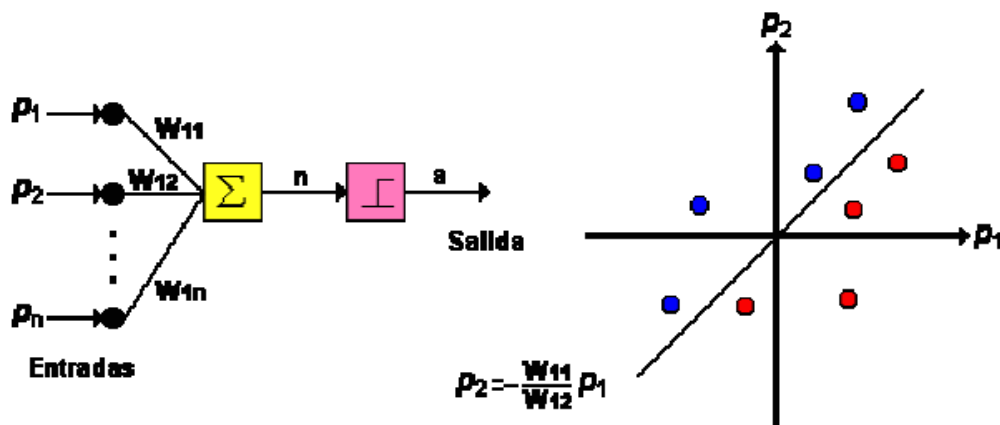


Ilustración 4 Estructura del modelo de red Perceptrón

La red Perceptrón utiliza principalmente dos funciones de transferencia, limitador fuerte (hardlim) con salida 1,0 y limitador fuerte Simétrico(hardlims) con salida 1, -1 (1). Su uso depende del valor de salida que se espera para la red, si la salida de la red es unipolar o bipolar, Una de las formas de de analizar el funcionamiento de una red de este tipo, es presentar en un mapa las regiones de decisión creadas en el espacio multidimensional de entradas de la red, y de esta forma se visualiza que patrones pertenecen a una clase u otra. Para esto se separa la región por un hiperplano cuya ecuación es

determinada por los pesos de las conexiones y el valor umbral de la función de activación de la neurona, los valores pueden ser fijos o utilizar diferentes algoritmo de entrenamiento.

2.3.2 Aprendizaje.

EL Perceptrón es un tipo de red que necesita conocer los valores de esperados para un conjunto de valores de entrada por lo que es un modelo de red supervisado. (1) Si comportamiento esta definido de la siguiente forma:

$$\{p_1, t_1\}, \{p_2, t_2\}, \dots, \{p_Q, t_Q\}$$

Siendo **P** los valores de entrada de la red y **T** los valores esperados. La salida de forma general esta dada por:

$a = F (\sum (W_i * P_i) + b)$ Donde **b** se define como la ganancia, término que permite aumentar el número de soluciones posibles, ya que permite desplazar la recta del origen de coordenadas. En general se el asigna el valor inicial de 1 y se ajusta durante el aprendizaje de la red.

Durante el entrenamiento de este tipo de red se expone un conjunto de patrones de entrada y los pesos de la red son ajustados de forma que al final del entrenamiento se obtengan las salidas deseadas para cada uno de esos patrones de entrada.

El algoritmo de entrenamiento del Perceptrón puede resumirse en los siguientes pasos:

- 1 Se inicializa la matriz de pesos y el valor de la ganancia, por lo general se asignan valores aleatorios a cada uno de los pesos **W_i** y al valor **b**.
- 2 Se presenta el primer patrón a la red, junto con la salida esperada en forma de pares entrada/salida.
- 3 Se calcula la salida de la red por medio de:

$$a = F (\sum (W_i * P_i) + b)$$

Donde **F** puede ser la función hardlim o hardlims.

- 4 Cuando la red no retorna la salida correcta, es necesario alterar el valor de los pesos, tratando de llevarlo hasta p y así aumentar las posibilidades de que la clasificación sea correcta, una

posibilidad es adicionar p a w haciendo que el vector w apunte en la dirección de p , y de esta forma después de repetidas presentaciones de p a la red, w se aproximará asintóticamente a p ; este es el procedimiento adoptado para la regla de aprendizaje del Perceptrón.

En todo este proceso se pueden definir 3 reglas fundamentales, las cuales cumplen la totalidad de las combinaciones de salidas y valores esperados con la utilización de las funciones de transferencia hardlim o hardlims, generalización que es posible introduciendo el error en las reglas de aprendizaje del Perceptrón:

$$\text{Si } e = 1, \text{ entonces } {}_1w^{\text{nuevo}} = {}_1w^{\text{viejo}} + p$$

$$\text{Si } e = -1, \text{ entonces } {}_1w^{\text{nuevo}} = {}_1w^{\text{anterior}} - p$$

$$\text{Si } e = 0, \text{ entonces } {}_1w^{\text{nuevo}} = {}_1w^{\text{anterior}}$$

Donde $e = t - a$

En una sola expresión se podría resumir así:

$${}_1w^{\text{nuevo}} = {}_1w^{\text{anterior}} + ep = {}_1w^{\text{anterior}} + (t - a)p$$

Y extendiendo la ley a las ganancias

$$b^{\text{nueva}} = b^{\text{anterior}} + e$$

2.4 Modelos de Redes Neuronales. Adeline.

El modelo de red neuronal Adeline es similar a la Perceptrón, la diferencia es que no utilizan la misma función de transferencia, la cual es una función de tipo lineal en lugar de un limitador fuerte. Esta red presenta la misma limitación del Perceptrón, en cuanto al tipo de problemas que pueden resolver, ambas redes pueden sólo resolver problemas linealmente separables. La red Adeline utiliza el algoritmo Least Mean Square (LMS), el cual es mucho más potente que la regla de aprendizaje del Perceptrón, ya que minimiza el error medio cuadrático. (1)

El elemento de procesamiento realiza la suma de los productos de los vectores de entrada y de pesos, y aplica una función de salida para obtener un único valor de salida, el cual debido a su función de transferencia lineal será +1 si la sumatoria es positiva o -1 si la salida de la sumatoria es negativa.

En este caso, la salida es la función unidad al igual que la función de activación; el uso de la función identidad como función de salida y como función de activación significa que la salida es igual a la activación, que es la misma entrada neta al elemento. En similitud con el Perceptrón, el límite de la característica de decisión para la red Adeline se presenta cuando $n = 0$.

2.4.1 Estructura del modelo de red.

Como la red Perceptrón, la Adeline es una red de aprendizaje supervisado que necesita conocer de antemano los valores asociados a cada entrada.

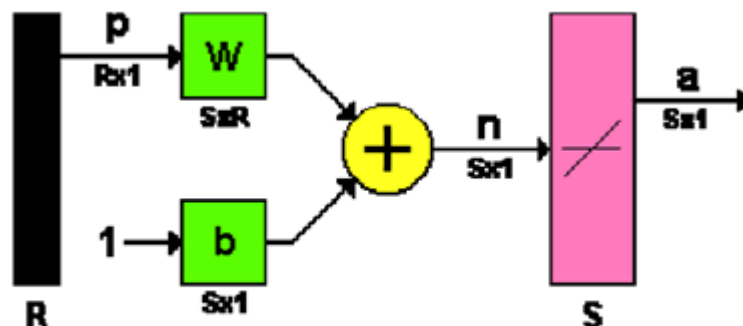


Ilustración 5 Estructura del modelo de red Adeline

Los pares de entrada/salida tienen la siguiente forma:

$$\{p_1, t_1\}, \{p_2, t_2\}, \dots, \{p_Q, t_Q\}$$

Donde p_Q es la entrada a la red y t_Q es su correspondiente salida deseada, cuando una entrada p es presentada a la red, la salida de la red es comparada con el valor de t que le es asociado.

La salida está dada por: $\mathbf{a} = \text{purelin}(\sum(\mathbf{W}_i * \mathbf{P}_i) + \mathbf{b})$

2.4.2 Aprendizaje.

El algoritmo LMS se deriva de la regla Widrow-Hoff delta, la que en términos generales para un proceso de actualización de los pesos de una red Adeline, se deduce de la siguiente manera:

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \alpha \frac{e(k)\mathbf{p}(k)}{|\mathbf{p}(k)|^2} \quad (1)$$

En el cual k representa la iteración actual del proceso de actualización, $\mathbf{W}(k+1)$ es el siguiente valor que tomará el vector de pesos y $\mathbf{W}(k)$ es el valor actual del vector de pesos. El error actual $e(k)$ es definido como la diferencia entre la respuesta deseada $\mathbf{t}(k)$ y la salida de la red $a(k) = \mathbf{W}^T(k)\mathbf{p}(k)$

antes de la actualización:

$$e(k) \equiv \mathbf{t}(k) - \mathbf{W}^T(k)\mathbf{p}(k) \quad (2)$$

La variación del error en cada iteración es representada por:

$$\Delta e(k) = \Delta(\mathbf{t}(k) - \mathbf{W}^T(k)\mathbf{p}(k)) = -\mathbf{p}^T(k) * \mathbf{W}(k) \quad (3)$$

En concordancia con la ecuación (1) la actualización de los pesos, teniendo en cuenta el error es:

$$\Delta \mathbf{W}(k) = \mathbf{W}(k+1) - \mathbf{W}(k) = \alpha \frac{e(k)\mathbf{p}(k)}{|\mathbf{p}(k)|^2} \quad (4)$$

Combinando las ecuaciones (3) y (4), se obtiene:

$$\Delta e(k) = -\alpha \frac{e(k)p^T(k)p(k)}{|p(k)|^2} = -\alpha e(k) \quad (5)$$

Extendiendo el algoritmo a la actualización de las ganancias, se tiene:

$$b(k+1) = b(k) + \alpha e(k)$$

De esta forma, el error es reducido por un factor α mientras los pesos van cambiando a medida que se presenta un valor de entrada. Cada vez que se presenta un nuevo patrón el ciclo de actualización inicia nuevamente; el siguiente error es reducido por un factor α , y el proceso continúa. Los valores iniciales del vector de pesos son usualmente escogidos como cero y se actualizan hasta que el algoritmo alcance convergencia. El algoritmo LMS corrige el error y si todos los patrones de entrada son de igual longitud, la actualización de pesos y ganancias tiende a minimizar el error medio cuadrático, esta es la principal propiedad de este algoritmo.

La función de error es una función matemática definida en el espacio de pesos multidimensional para un conjunto de patrones dados, es una superficie que tendrá muchos mínimos (globales y locales) y la regla de aprendizaje va a buscar el punto en el espacio de pesos donde se encuentra el mínimo global de esa superficie; aunque la superficie de error es desconocida, el método de gradiente descendiente consigue obtener información local de dicha superficie a través del gradiente, con esa información se decide qué dirección tomar para llegar hasta el mínimo global de dicha superficie.

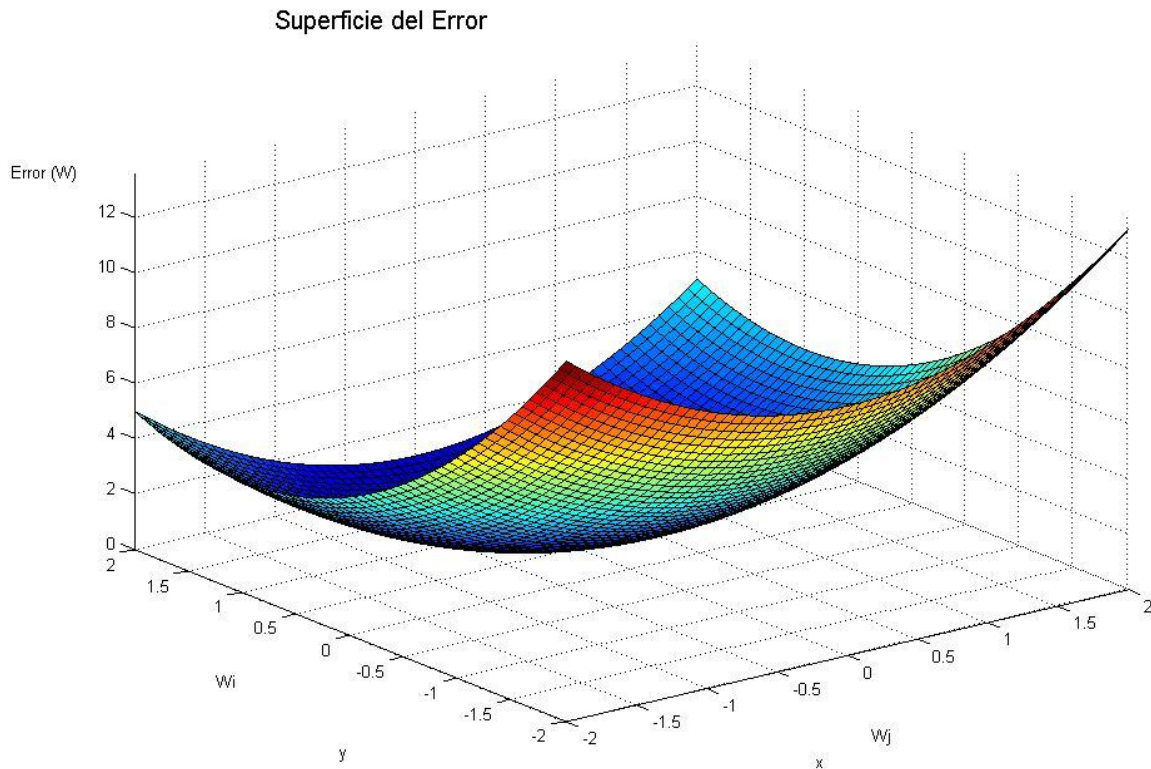


Ilustración 6 Superficie del error

Con este algoritmo calculando el gradiente en cada iteración (gradiente instantáneo) y no el gradiente sobre el error total después de haber presentado todos los patrones, la función para el error medio cuadrático es:

$$e^2(k) = (t(k) - a(k))^2$$

En esta ecuación $t(k)$ representa la salida esperada en la iteración k y la $a(k)$ representa la salida de la red; el error cuadrático esperado ha sido reemplazado por el error cuadrático en la iteración k . Esto brinda la ventaja de la simplificación del error medio cuadrático al poder ser calculado por medio del error en la iteración k , y así para calcular el error se necesita solo multiplicar el error por el número de entradas.

2.5 Modelos de Redes Neuronales Backpropagation (Perceptrón Multicapa).

Las redes multicapas nacieron con el objetivo de sobrepasar a las redes de una sola capa porque estas solo podían resolver problemas linealmente separables. Uno de los grandes avances logrados con la Backpropagation es que esta red aprovecha la naturaleza paralela de las redes neuronales para reducir el tiempo requerido por un procesador secuencial para determinar la correspondencia entre unos patrones dados. Además el tiempo de desarrollo de cualquier sistema que se esté tratando de analizar se puede reducir como consecuencia de que la red puede aprender el algoritmo correcto sin que alguien tenga que deducir por anticipado el algoritmo en cuestión. A diferencia de las redes lineales estas redes no necesitan conocer de antemano el valor de la entrada.

La popularidad del perceptrón multicapa se debe principalmente a que es capaz de actuar como un aproximador universal de funciones. Más concretamente, una red *backpropagation* conteniendo al menos una capa oculta con suficientes unidades no lineales puede aprender cualquier tipo de función o relación continua entre un grupo de variables de entrada y salida. Esta propiedad convierte a este tipo de red en una herramienta de propósito general, flexible y no lineal.

La Backpropagation es un tipo de red de aprendizaje supervisado, que emplea un ciclo propagación – adaptación de dos fases. Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, éste se propaga desde la primera capa a través de las capas superiores de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas.

Las salidas de error se propagan hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de la capa oculta que contribuyen directamente a la salida. Sin embargo las neuronas de la capa oculta sólo reciben una fracción de la señal total del error, basándose aproximadamente en la contribución relativa que haya aportado cada neurona a la salida original. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido una señal de error que describa su contribución relativa al error total. Basándose en la señal de error percibida, se actualizan los pesos de conexión de cada neurona, para hacer que la red converja hacia un estado que permita clasificar correctamente todos los patrones de entrenamiento. La importancia de este proceso consiste en que, a medida que se entrena la red, las neuronas de las capas intermedias se organizan a sí mismas de tal modo que las distintas neuronas aprenden a reconocer distintas características del espacio total de entrada.

2.5.1 Estructura del modelo de red.

Una red de n capas puede tener n tipos de redes Perceptrón en cascadas; la salida de la primera red es la entrada de la segunda y la salida de la segunda es la entrada de la tercera y así sucesivamente hasta llegar a la capa n . Cada capa puede tener diferentes números de neuronas y hasta distintas funciones de transferencias.

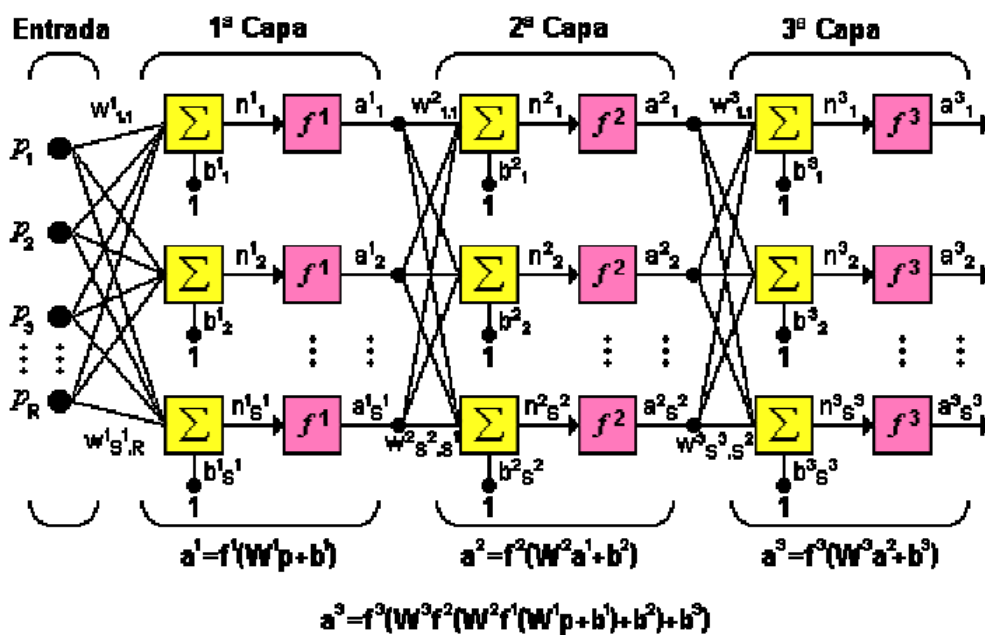


Ilustración 7 Modelo de red de 3 capas

Es importante recalcar que no existe una técnica para determinar el número de capas ocultas, ni el número de neuronas que debe contener cada una de ellas para un problema específico, esta elección es determinada por la experiencia del diseñador, el cual debe cumplir con las limitaciones de tipo computacional.

Se debe tener en cuenta que para aprovechar la capacidad de las RNA de aprender relaciones complejas o no lineales entre variables, es absolutamente imprescindible la utilización de funciones no lineales al menos en las neuronas de la capa oculta. Las RNA que no utilizan funciones no lineales, se limitan a solucionar tareas de aprendizaje que implican únicamente funciones lineales o problemas de clasificación que son linealmente separables. Por tanto, en general se utilizará la función

sigmoidal(logística o tangente hiperbólica) como función de activación en las neuronas de la capa oculta.

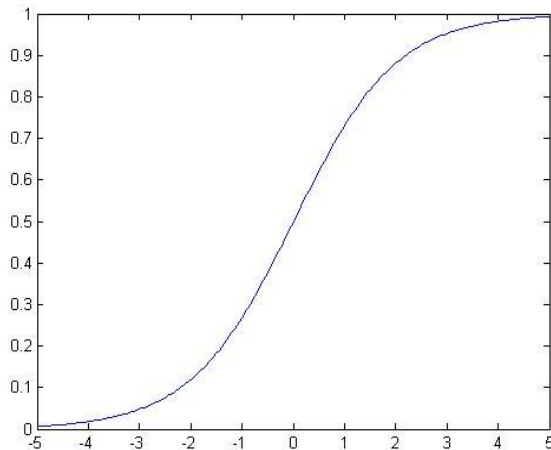


Ilustración 8 Función logarítmica

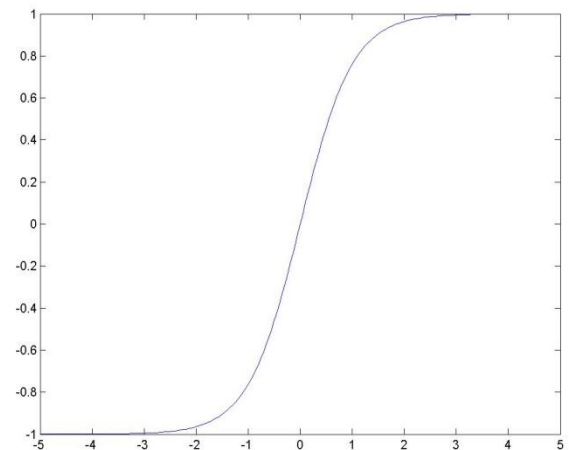


Ilustración 9 Función Hiperbólica

$$y = \frac{1}{1 + e^{-x}}$$

$$y = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Por su parte, la elección de la función de activación en las neuronas de la capa de salida dependerá del tipo de tarea impuesto. En tareas de clasificación, las neuronas normalmente toman la función de activación sigmoide. Así, cuando se presenta un patrón que pertenece a una categoría particular, los valores de salida tienden a dar como valor 1 para la neurona de salida que representa la categoría de pertenencia del patrón, y 0 ó -1 para las otras neuronas de salida. En cambio, en tareas de predicción o aproximación de una función, generalmente las neuronas toman la función de activación lineal.

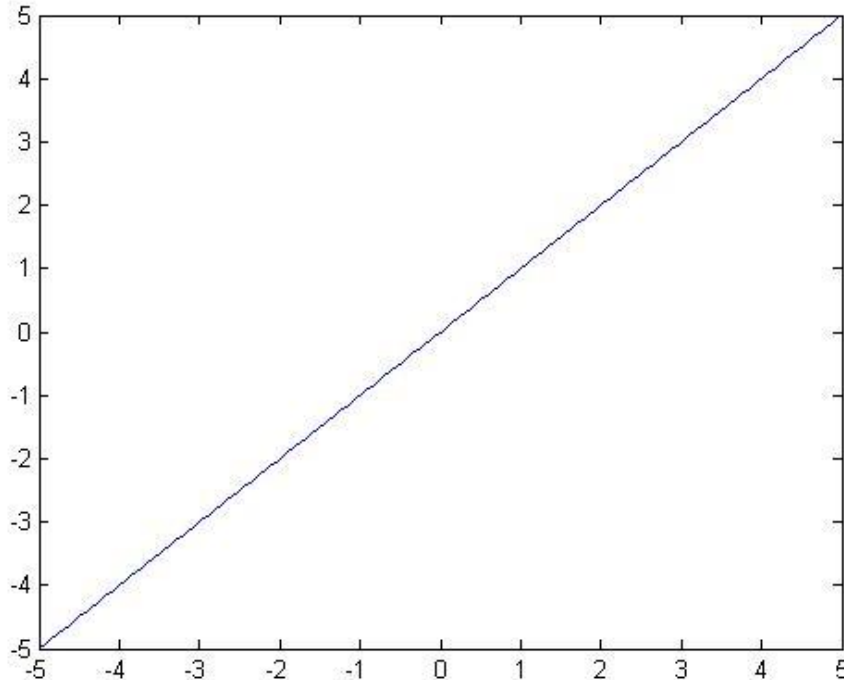


Ilustración 10 Función lineal
 $y = x$

2.5.2 Aprendizaje

Para el entrenamiento de una red es necesario antes de iniciar el proceso de aprendizaje tener bien definida la topología de la red. Se debe conocer el número de neuronas en la capa de entrada el cual depende del número de componentes de entradas. Además la cantidad de capas ocultas y las neuronas en cada una de ellas así como el número de neuronas en la capa de salida el cual depende del número de componentes de salida o patrones objetivos y las funciones de transferencia requeridas en cada capa. Con la base de toda la topología escogida se le asignan los valores iniciales a los parámetros necesarios en la red.

En la etapa de aprendizaje, el objetivo que se persigue es hacer mínima la discrepancia o error entre la salida obtenida por la red y la salida deseada por el usuario ante la presentación de un conjunto de

patrones denominado grupo de entrenamiento. La base matemática del algoritmo *backpropagation* para la modificación de los pesos es la técnica conocida como gradiente decreciente. Con este método el gradiente toma la dirección que determina el incremento más rápido en el error, mientras que la dirección opuesta, es decir la dirección negativa, determina el decremento más rápido en el error. Este algoritmo para redes multicapa es una generalización del algoritmo LMS, se deben ajustar los parámetros de la red para minimizar el error medio cuadrático.

Para una mejor comprensión a continuación se ilustrara el proceso de aprendizaje de forma gráfica. El conjunto de pesos que forma una red neuronal puede ser representado por un espacio compuesto por tantas dimensiones como pesos tenga. Para simplificar el problema se tomará por ejemplo una red formada por dos pesos. El paisaje se puede visualizar como un espacio de dos dimensiones y a cualquier combinación de valores de los dos pesos, le corresponderá un valor de error para el conjunto de entrenamiento; el cual se puede visualizar el espacio de la superficie que se crea.

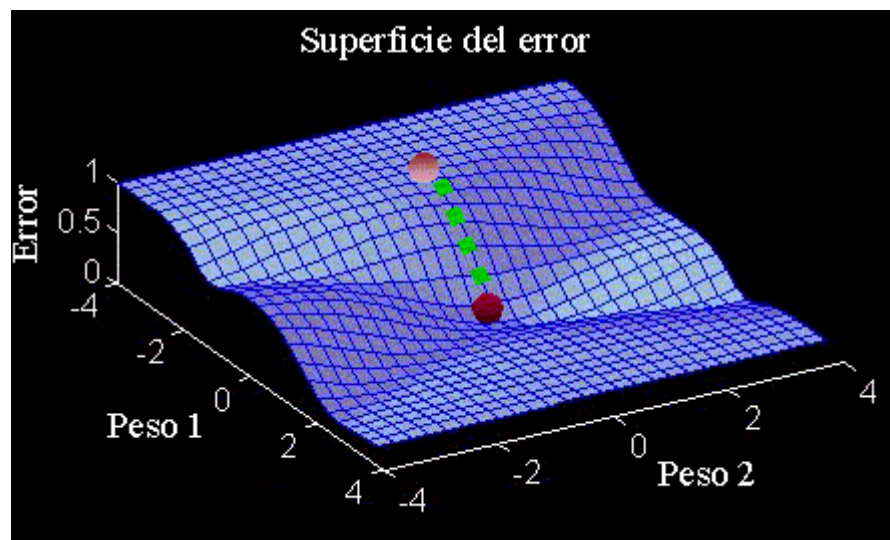


Ilustración 11 Error

Como se muestra en la figura anterior el proceso de entrenamiento comienza en un determinado punto representado por un círculo, definido por los pesos iniciales de la red. El algoritmo de aprendizaje se basa en obtener información local de la pendiente de la superficie, o sea del gradiente, y a partir de esta información modificar iterativamente los pesos de forma proporcional a dicha pendiente, a fin de asegurar el descenso por la superficie del error hasta alcanzar el mínimo más cercano desde el punto

de partida. Con un número mayor de pesos el espacio se convierte en un plano multidimensional inimaginable, aunque se seguirían aplicando los mismos principios.

2.5.2.1 Entrenamiento de una Red de Unidades Sigmoidales.

Método de Retropropagación sigue la idea de aplicar descenso por el gradiente en la red completa.

Gradiente:

$$\nabla E[\vec{w}] \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_n} \right]$$

Regla de Entrenamiento:

$$\Delta \vec{w} = -\eta \nabla E[\vec{w}]$$

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

Para cada ejemplo de entrenamiento haga lo siguiente:

1. Entre el ejemplo de entrenamiento a la red y calcula la salida que da la red
2. Para cada unidad de salida k

$$\delta_k \leftarrow o_k(1 - o_k)(t_k - o_k)$$

3. Para cada unidad escondida h

$$\delta_h \leftarrow o_h(1 - o_h) \sum_{k \in \text{outputs}} w_{h,k} \delta_k$$

4. Actualiza el valor de cada peso W_{ij} de la red

$$w_{i,j} \leftarrow w_{i,j} + \Delta w_{i,j}$$

donde

$$\Delta w_{i,j} = \eta \delta_j x_{i,j}$$

2.5.2.2 Características principales de la Retropropagación.

- Descenso por Gradiente sobre toda la red.
- Fácil de generalizar a redes de grafos dirigidos arbitrarios.
- Encuentra un mínimo local no necesariamente global. Funciona bien en la práctica (se puede correr múltiples veces para mejorar).
- Minimiza errores sobre los datos de entrenamiento.
- Lento de entrenar (puede tomar cientos de iteraciones).
- El uso de la red después de entrenarse es rápido.

2.6 Funciones de Transferencias Sigmoideal.

Entre las funciones más conocidas se destacan las funciones sigmoideas, ya que desde el punto de vista matemático, la utilidad de estas funciones es que estas y sus derivadas son continuas. Estas funciones trabajan bastante bien y son normalmente las elegidas. Se trata de una función continua no lineal con bastante aplicabilidad fisiológica. Esta función toma los valores de entrada, los cuales pueden oscilar entre más y menos infinito. Posee un rango comprendido entre 0 y 1 lo que permite que aplicado a las unidades de proceso de una RNA, independientemente de la entrada, la salida estará comprendida entre 0 y 1.

Función Sigmoideal: tasa de activación

En estas funciones la salida de una unidad vale 0.5 cuando la entrada es nula; esto significa que la unidad tiene cierta actividad, aún en ausencia de estimulación. Al aumentar la estimulación, la unidad aumenta su activación, y disminuye si la estimulación es inhibitoria, de forma parecida al modo en que

se comportan las neuronas reales. La función sigmoideal presenta las siguientes principales características:

- ❖ Acomodación de señales muy intensas sin producir saturación.
- ❖ Permite modelar funciones de salida continua.
- ❖ Admite señales débiles sin excesiva atenuación.
- ❖ Permite modelar funciones de probabilidad (regresión logística)
- ❖ Fácilmente derivable.

$$\frac{d\sigma(x)}{dx} = \sigma(x)(1 - \sigma(x))$$

La principal limitación de esta función es que no sirve para expresar polaridades, da siempre valores positivos. Existe una función alternativa, con cualidades parecidas, pero con un rango entre -1 y 1, llamada función tangente hiperbólica. Desde un punto de vista fisiológico, el signo negativo se puede interpretar como una disminución de la tasa de disparo de una neurona por debajo de la tasa de disparo en reposo.

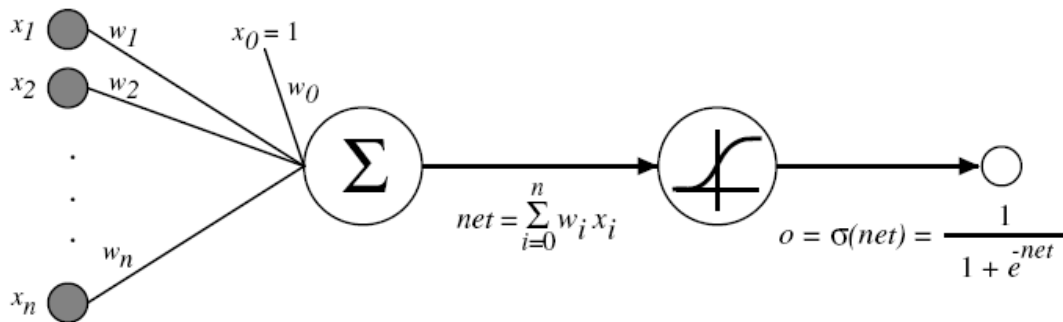


Ilustración 12 Unidad sigmoideal

2.6.1 Poder expresivo de Redes de Unidades Sigmoidales.

Funciones Booleanas:

- Pueden ser modeladas con redes con nivel oculto
- El número de unidades del nivel oculto puede ser exponencial

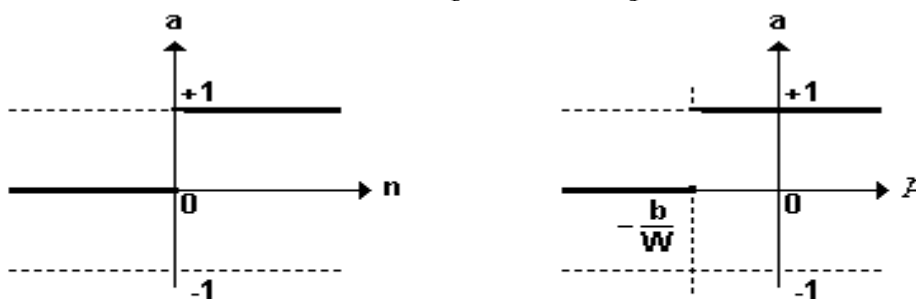
Funciones Continuas:

- Toda función continuas acotada puede ser aproximada con error fijo usando una red de unidades sigmoidales de un nivel oculto.
- Cualquier función puede ser aproximada con error fijo usando una red de unidades sigmoidales de dos niveles ocultos.

2.7 Funciones de Transferencias Limitador fuerte (Hardlim).

Esta función de transferencia acerca la salida de la red a cero, si el argumento de la función es menor que cero y la lleva a uno si este argumento es mayor que uno. Crea neuronas que clasifican las entradas en dos categorías diferentes, característica que le permite ser empleada en la red tipo Perceptrón.

$$a = \begin{cases} 1 & \text{si } n \geq 0 \\ 0 & \text{si } n < 0 \end{cases}$$



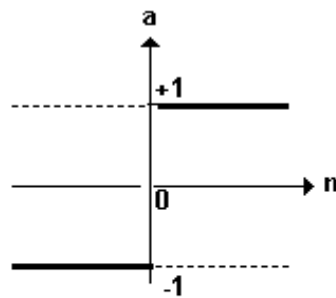
Función de Transferencia Limitador Fuerte

Entrada a una Neurona Limitador Fuerte

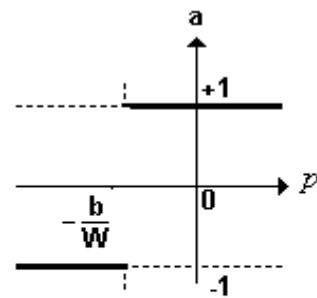
Ilustración 13 Limitador fuerte

Una modificación de esta función puede verse en la representación la función de transferencia **Hardlims** que restringe el espacio de salida a valores entre 1 y -1.

$$a = \begin{cases} 1 & \text{si } n \geq 0 \\ -1 & \text{si } n < 0 \end{cases}$$



Función de Transferencia Limitador Fuerte Simetrica



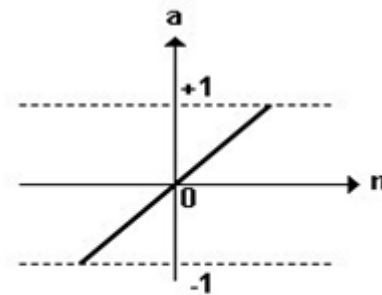
Entrada a una Neurona Limitador Fuerte Simetrica

Ilustración 14 Limitador fuerte simétrico

2.8 Funciones de Transferencias Lineal (purelin).

La salida de la función es igual a la entrada. Neuronas que emplean esta función de transferencia son utilizadas en la red tipo Adeline.

$$a = n$$



Función de Transferencia Lineal

Ilustración 15 Función lineal

CAPÍTULO III: “TEORÍA DE ERRORES APLICADA A LOS MODELOS”.

3.1 Introducción.

En este capítulo se realiza un estudio sobre los tres modelos de redes neuronales más comúnmente utilizados en el proyecto “Desarrollo de Elementos Virtuales Inteligentes” y las funciones de transferencias utilizadas por estos modelos. Se profundizara en sus características principales, sus estructuras y funcionamiento.

3.2 Representación Numérica en Sistemas Digitales.

Debido a la capacidad finita de la memoria y la forma de representar y operar los números en computadoras, sobresalen 3 cualidades fundamentales en el trabajo numérico en la máquina, los cuales hay que tener en cuenta para la resolución de problemas por métodos o modelos numéricos con las computadoras, por lo cual estos modelos de redes no están exentos de estas limitaciones:

1. Los conjuntos numéricos que manipula la computadora son finitos, por tanto son acotados superior e inferiormente. Para la representación de estos se utilizan dos notaciones conocidas como de punto fijo y de punto flotante. Con la primera se representan de forma exacta los números naturales y enteros; y para los números reales se utiliza la segunda notación. En ambas se toma una secuencia de n bits de memoria y se le asocia a cada uno un estado de cero o uno. De aquí se deriva que la cantidad de números representables para un valor de n es finita e igual a 2^n . Las propiedades del conjunto numérico representado dependerán de la notación que se utilice y del valor de n seleccionado.
2. Se produce un error de redondeo al representar y efectuar operaciones con números racionales. Con la notación de punto flotante no se puede representar todo el conjunto de los reales, sino solo un subconjunto finito de números racionales con una cantidad de cifras significativas a lo sumo. Puede suceder que algunos de los números reales que estén en el rango permisible, se representen de forma aproximada. En este caso el error de dicha aproximación dependerá de la precisión de la representación, o sea mientras mayor sea la precisión, menor será el error de redondeo. Aún en el caso de que varios números racionales

pertenezcan a un subconjunto determinado, es posible que el resultado de realizar operaciones entre ellos, no pertenezca a dicho conjunto, por esta razón también se introducen aproximaciones.

3. Las operaciones de adición y multiplicación en la máquina no son asociativas. Además, no se cumple la ley de distributividad de la multiplicación con respecto a la adición. En otras palabras el orden de las operaciones en la computadora influye en los resultados. Por ejemplo: Dado un subconjunto formado por números de dos cifras significativas $\{0,0 \dots\dots 9,9\}$

$$(5,0 * 0,4) * 3,1 = 2,0 * 3,1 = 6,2 \quad \text{y} \quad 5,0 (0,4 * 3,1) = 5,0 * 1,2 = 6,0$$

El error de redondeo cometido en este trabajo con dos cifras significativas puede verse como el error asociado a la operación.

3.3 Errores en el modelo de RNA Perceptrón.

Como se explicó en el capítulo anterior, en este modelo de red la única neurona de salida realiza la suma ponderada de sus entradas para luego dar una función de salida. Estas entradas están determinadas por los valores de la salida del resto de las neuronas multiplicada por los pesos que corresponden a cada una en su capa y sumado a la ganancia.

$$a = F (\sum (W_i * P_i) + b)$$

Los pesos (w) son valores reales que se escogen de forma aleatoria y se van ajustando mediante algoritmos durante el proceso de aprendizaje para llegar a la solución deseada. En este proceso se introducen errores que no son percibidos por el entrenador de la red, el hecho de que los pesos sean números reales cuales quiera y que sean escogidos al azar puede introducir un error aleatorio en el proceso. Además se debe tener en cuenta el tamaño del número que se le asigne a cada uno de los pesos, pues podrían introducirse errores de redondeo o truncamiento por la propia computadora y su limitación de capacidad de representación.

Al multiplicar los valores de entrada P por el peso correspondiente W , el valor que se obtiene es un número aproximado del valor real, o sea P puede tomar los valores 0, 1 ó -1, según la función que se

utilice, mientras W puede tomar cualquier valor por lo cual la precisión del valor de $Wi*Pi$ depende en gran medida de la precisión de W . Al sumar la influencia de la cada una de las neuronas en la salida $Wi*Pi$, se produce una propagación del error en esta suma, ya que cada uno de los sumando es un valor aproximado y este error se propaga hasta el resultado final. El valor de la ganancia b también es un valor aproximado que se debe tener en cuenta en la propagación del error en esta suma.

Aunque este error no se puede obviar ni desaparecer, si se puede disminuir en busca de un menor efecto en el resultado final. Para esto es recomendable que al efectuar una suma algebraica de números aproximados se mantengan inalterables los sumandos de mayor error absoluto máximo y se redondean los restantes, reteniendo una o dos cifras más que los anteriores. Después de sumar se redondea el resultado final, reduciéndolo en una cifra.

También es cierto que aplicar esta regla a algoritmos para una computadora no resulta muy sencillo, su complicación aumenta si se van a sumar varios números aproximados, como es el caso de una red de un gran número de neuronas en la capa de entrada. No obstante el error de redondeo que se produce y se propaga pudiera adquirir gran relevancia en una operación, sobre todo si se van a sumar números racionales grandes con otros muy pequeños. En este caso, teniendo en cuenta que la adición en las computadoras no es asociativa, se deben agrupar los números pequeños para sumarlos primero y luego adicionarlo a las mayores.

Algo muy similar ocurre en la actualización de los pesos y la ganancia en el algoritmo de aprendizaje de este modelo de red:

$${}_1w^{nuevo} = {}_1w^{anterior} + ep = {}_1w^{anterior} + (t - a)p$$

$$b^{nueva} = b^{anterior} + e$$

Como se puede apreciar en las ecuaciones anteriores ambos valores se actualizan a partir de valores aproximados, por lo cual estos últimos ya presentan un error y en el nuevo resultado también se verá influenciado por el error. Este proceso de actualización es iterativo y termina cuando se encuentren los valores que den la respuesta deseada o al menos aceptada para las entradas que se le administraron a la red y durante toda la actualización, por muy pequeño que sea, está presente este error de trabajar

con valores aproximados para obtener nuevos valores. El error se va propagando de una iteración a otra.

Otra de las posibles imprecisiones que presenta el Perceptrón, es cuando se hace uso la función hardlim, porque al tener posibles valores de salida 0 y en el aprendizaje estos valores de entrada a otra neurona son multiplicados por sus respectivos vectores de peso, provocando que tomen valor cero estos productos, lo cual ocasiona que estos valores de pesos no se actualicen y que el aprendizaje sea más lento, es por esto que se prefiere el uso de la hardlims.

Si se toma en cuenta que se trabaja con una red donde cada neurona se conecta con las demás del resto de las capas, se debe tener presente la propagación de errores que se genera en el modelo de red en general. Si cada una de las neuronas genera una salida que contiene una cota de error, este erro es propagado al resto de las neuronas conectadas a ellas, además de que la misma neurona en su valor de salida se ve influenciada por los valores de entrada que tiene de las neuronas de la capa anterior.

3.4 Errores en el modelo RNA Adeline.

En este modelo de red están presentes los errores que tratamos en el modelo perseptrón. La salida de igual manera esta influenciada por el valor de los pesos de cada una de los valores de entrada:

$$a = \text{purelin} (\sum (W_i * P_i) + b)$$

La situación descrita en el modelo anterior sobre el error en la suma de los valores de entrada y sus pesos se repite en este caso. La variación está en el algoritmo, en la forma en que se actualizan los pesos para entrenar la red:

$$W(k+1) = W(k) + \alpha \frac{e(k)p(k)}{|p(k)|^2}$$

En este nuevo método de actualización se mantiene el trabajo con valores aproximados de los pesos para obtener los nuevos valores de pesos de la red, lo cual como se explicó anteriormente introduce errores en el resultado final al terminar las iteraciones. Aunque a diferencia del modelo anterior los valores de peso generalmente se inicializan con 0 y se van actualizando hasta que converja el

algoritmo y se llegue a una respuesta. Como se puede apreciar aparece un nuevo término α , el cual no es un valor exacto o entero, sino una aproximación del valor real por lo cual también puede introducir errores de redondeo o truncamiento el cual es propagado cada vez que se actualice el valor del próximo peso y se entre en una nueva iteración.

La correcta elección de α controla la estabilidad y velocidad de la convergencia del proceso de entrenamiento, si se escoge un valor muy pequeño de α , el algoritmo pierde velocidad y tarda mucho en alcanzar convergencia, si por el contrario se toma un valor muy grande, el algoritmo pierde estabilidad y se torna oscilante alrededor del valor de convergencia. Para patrones de entrada independientes en el tiempo, la estabilidad es garantizada para valores de α que varíen entre: $0 < \alpha < 2$. Si se fija α en un valor mayor a 1 el error es innecesariamente sobre-correcto, por lo tanto un rango de valores prácticos para la tasa de aprendizaje es: $0.1 < \alpha < 1$.

En el cálculo del error, en cada iteración se puede introducir errores:

$$e(k) \equiv \mathbf{t}(k) - \mathbf{W}^T(k)\mathbf{p}(k)$$

Como puede notarse el error en la iteración k está influenciado por el valor del peso correspondiente el cual pudiera ser una aproximación de su valor real y por consiguiente afectar el resultado del error.

En la actualización de la ganancia y de la variación del error también se pueden introducir errores:

$$b(k+1) = b(k) + \alpha e(k)$$

$$\Delta e(k) = -\alpha \frac{e(k)\mathbf{p}^T(k)\mathbf{p}(k)}{|\mathbf{p}(k)|^2} = -\alpha e(k)$$

Como se muestra en las fórmulas anteriores, utilizadas en el algoritmo de aprendizaje de la Adeline, la actualización de la variación del error entre una iteración y otra se ve afectada por el valor de α . Esto trae como consecuencia que al trabajar con este valor aproximado la variación del error también se obtenga con una cota de error. En la actualización de la ganancia sucede algo muy similar, se trabaja con la multiplicación de dos números aproximados, el propio valor de α y el valor error en la iteración actual, el cual ya presenta una cota de error producida durante su cálculo.

3.5 Errores en el modelo RNA Backpropagation.

Al igual que en los modelos adeline y perceptrón, la entrada de una neurona de la red se ve influenciado por el error de aproximación de los pesos de cada una de las otras neuronas conectadas que le envían su salida. Esto provoca que la suma ponderada en cada neurona de los pesos multiplicados por los valores de entrada también sea un valor aproximado y con una cota de error, o sea, la salida de cada neurona de la red estará influenciada por esto.

Por otra parte en las funciones de transferencia que utiliza la red backpropagation también pueden introducirse errores:

$$y = \frac{1}{1 + e^{-x}}$$

$$y = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Como se puede apreciar el término e se encuentra elevado al término x que no es más que la suma ponderada del resto de las neuronas. Este último valor es un aproximado por lo cual contiene cota de error y este error se propaga a la potencia y por consiguiente afecta la salida de la función. Aunque no se puede eliminar del todo esta propagación del error, si es recomendable que siempre que sea posible se debe simplificar las expresiones que contengan potencia, con el objetivo de obtener expresiones con potencias de menor exponente.

Este modelo de red multicapa utiliza funciones de transferencia sigmoideas en las capas ocultas. Estas funciones tienen la particularidad de comprimir un rango de entrada infinito en un rango de salida finito y además su pendiente tiende a cero cuando los vectores de entrada son muy grandes. Esto ocasiona un problema de variación mínima sobre los pesos, haciendo que las redes entreguen respuestas erradas debido a que convergen aún cuando sus valores no son óptimos. En estas funciones también sucede que para valores muy grandes de entrada las salidas pudieran ser casi las mismas o iguales si se aplica un redondeo o truncamiento.

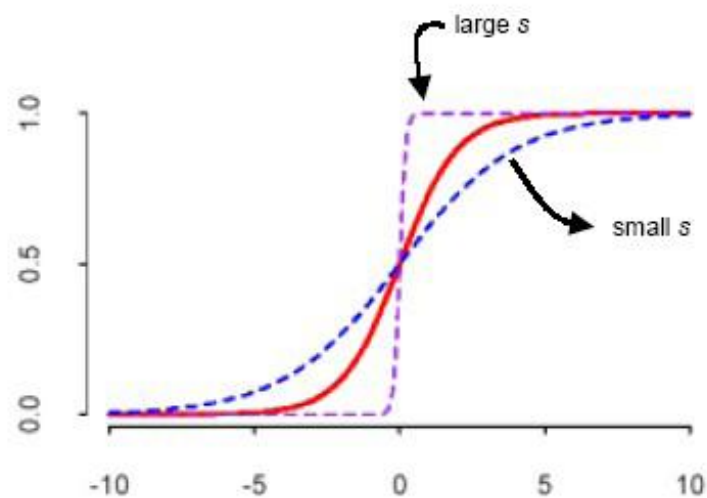


Ilustración 16 Función Sigmoide

Durante el entrenamiento de este modelo de red un peligro que puede surgir al utilizar el método de gradiente decreciente es que el aprendizaje converja en un punto bajo, sin ser el punto más bajo de la superficie del error. Tales puntos se denominan mínimos locales para distinguirlos del punto más bajo de esta superficie, denominado mínimo global. Sin embargo, el problema potencial de los mínimos locales ocurre en raras ocasiones en datos reales.

En las técnicas de gradiente descendiente es conveniente avanzar por la superficie del error con incrementos pequeños de los pasos, esto se debe a que tenemos una información local de la superficie y no se sabe lo lejos o lo cerca que se está del punto mínimo, con incrementos pequeños, aunque se tarde más en llegar, se evita que esto ocurra. El elegir un incremento adecuado influye en la velocidad de convergencia del algoritmo, esta velocidad se controla a través del valor de α , por lo general se debe escoger un número pequeño para asegurar que la red encuentre solución. Un valor pequeño de α provocaría un aumento en el número de iteraciones, si se toma un valor muy grande, los cambios en los pesos serían muy grandes, avanzando muy rápido por la superficie del error, con el riesgo de saltar el valor del mínimo del error y estar oscilando a su alrededor sin llegar a él. Se recomienda aumentar el valor de α a medida que disminuye el error de la red durante la fase de entrenamiento, para garantizar así una rápida convergencia sin perder de vista tomar valores pequeños para llegar al mínimo.

Es lógico suponer que si cada neurona tiene un valor de salida que contiene un error y este es enviado al resto de las neuronas de la siguiente capa exista una propagación de errores entre todas las neuronas de la red y el resultado final sea afectado debido a esto.

CONCLUSIONES

En el presente trabajo de diploma se logró dar cumplimiento a los objetivos planteados.

Se muestra cómo incorporar algunos elementos de la teoría de errores aplicados en la física, a la informática, principalmente en la inteligencia artificial.

Se tuvo en cuenta la clasificación de los errores en los modelos y sus funciones de transferencia.

Se determinaron por vez primera los posibles errores cometidos en los modelos de redes neuronales artificiales y sus funciones de transferencias, así como los métodos para disminuir algunos de estos errores.

No se puede afirmar cuál es el modelo más óptimo porque cada modelo responde a una tarea específica, pero los modelos de una sola capa presentan menos errores que los modelos de "n" capas, ya que los modelos de una sola capa solo tiene una entrada a diferencia de los modelos de "n" capas tienen varias entradas y los errores están más presentes.

Se analizó la propagación del error y mientras más variables presentan una función de transferencia mayor error de propagación puede existir.

RECOMENDACIONES

Continuar la búsqueda de otros errores en los modelos de RNA aplicándole el procedimiento de la incertidumbre.

Aplicar la teoría de errores a los modelos de RNA que no fueron analizados en el trabajo.

Vincular la teoría de errores en las diferentes ramas de la informática.

Incorporar los resultados obtenidos en esta investigación en algunos proyectos investigativos y/o productivos en la UCI.

BIBLIOGRAFÍA

1. Pereira, Universidad Tecnológica de. Redes neuronales Artificiales. [En línea] 2000. <http://www.ohm.utp.edu.co/neuronales>.
2. González, José Ramón Hiler. *REDES NEURONALES ARTIFICIALES. FUNDAMENTOS, MODELOS Y APLICACIONES*. s.l. : Ra-ma, 1995.
3. *MODELADO DE SERIES CLIMATOLÓGICAS MEDIANTE UNA RED*. Guzmán, Jesús Palazón González y Adela García. 1, Córdoba. : s.n., 2004, Vol. 11.
4. Bustamante, Miguel Vera y Juan Bustamante. *Modelo dinámico para la generación de pronóstico usando redes neurales artificiales (RNA)*. 2007.
5. Las Redes Neuronales Artificiales. [En línea] 2000. [http:// ohm.utp.edu.co/neuronales](http://ohm.utp.edu.co/neuronales).
6. *Apuntes de Física. TEORÍA DE ERRORES*. Sierra, Arturo Quirantes.
7. A., Munir A. Jalil B. y Martha Misas. *Evaluación de pronósticos del tipo de cambio utilizando*. 2006.
8. *Teoría de Errores y Guiones de Prácticas. Física Mecánica*. s.l. : DEPARTAMENTO DE FÍSICA - UNIVERSIDAD DE JAÉN.
9. *Introducción a la Teoría de Errores*. Universidad de Laguna : s.n.
10. S. Gil, E. Rodríguez. *Teoría de errores. Incertezas de medición*. s.l. : Física re-Creativa.
11. *TEORIA DE ERRORES*. Universidad Nacional de la Pampa : s.n.
12. ROJAS, PEDRO ENRIQUE ADAMES. *APUNTES DE CALCULO NUMÉRICO. SISTEMA DE NUMEROS PUNTO FLOTANTE*. UNIVERSIDAD CENTRO OCCIDENTAL LISANDRO ALVARADO : s.n., ENERO, 2001.
13. Carletti, Eduardo J. Los peores "bugs" (errores informáticos) de la Historia. [En línea] Noviembre de 2005. <http://axxon.com.ar>.
14. L., Carlos Hurtado. *Redes Neuronales (2)*. [PPT] Universidad de : s.n.
15. Ballesteros, Alfonso. Neural Networks Framework. *Neural Networks Framework*. [En línea] <http://www.redes-neuronales.netfirms.com>.
16. *Tutorial sobre Redes Neuronales Artificiales: El Perceptrón Multicapa*. Palmer, A., Montaña, J.J. y Jiménez, R. 2, 2001, Vol. 5.
17. Sotolongo, Guzmán, Maria Victoria. *Aplicaciones de las redes neuronales. El caso de la Bibliometría*. La Habana, Cuba : s.n., 2001.

18. Sandoval, Ing. Alejandro Cruz. *Estabilidad de entrada-estado (ISS) para identificación con redes neuronales dinámicas*. Mexico, DF : s.n., 2003.
19. carrión, José León. *Redes Neuronales Artificiales y la Teoría neuropsicológica de Luria*. Sevilla, España : s.n., 2002.
20. Herrera, Juan Carlos. REDES NEURONALES ARTIFICIALES. [En línea] 2007. <http://es.wordpress.com/>.
21. Sanz, Bonifacio Martín Y alfredo. *Redes Neuronales y Sistemas difusos*. Zaragoza, España. : s.n., 2001.
22. A Mauricio Bernal Bohórquez, Christian Fabián Díaz Moreno, Jamer Alexander Cuevas Sandoval, Omar Yesid Quijano Reyes. *IMPLEMENTACIÓN DE UN JUEGO EN LINEA UTILIZANDO REDES NEURONALES*. COLOMBIA : s.n., 2005.
23. Galdaméz, Ing. Cruz Antonio. *REDES NEURONALES*. Ciudadela Don Bosco, : s.n., 2004.
24. POL, DR. ALFONSO PALMER. *Redes Neuronales Artificiales aplicadas al Análisis de Datos*. PALMA DE MALLORCA : s.n., 2002.
25. PALMER POL, MONTAÑO MORENO. *¿Qué son las redes neuronales artificiales? Aplicaciones realizadas en el ámbito de las adicciones*. Universidad de las Islas Baleares : s.n.
26. Ochoa, Susana. Estructura de la Neurona Artificial. [En línea] 2003. <http://fluidos.eia.edu.co/hidraulica/articulosos/flujoentuberias/neuronal>.
27. *Elementos básicos de una Red Neuronal Artificial(Parte II)*. Paul, Neil, Eli. Ecuador : s.n., 2007.
28. neuronales, Características de las redes. Características de las redes neuronales. [En línea] <http://thales.cica.es/rd/Recursos/rd98/TecInfo/07/capitulo3.html>.
29. neural@electronica.com.mx. *Redes Neurales Artificiales*. [En línea] 15 de Octubre de 2000. <http://electronica.com.mx/neural/aplicaciones/>.
30. J.C. Moctezuma Eugenio, A. Sánchez Galvez, A. Ata Pérez. *IMPLEMENTACIÓN HARDWARE DE FUNCIONES DE TRANSFERENCIA*. Universidad Autónoma de Puebla, México : s.n.
31. Blanco, Manuel Alvarez. *Matemática numérica*. La Habana : Felix Varela, 2003.
32. Errores en las mediciones.

33. Luaces, MSc. Roberto Millet. *La teoría de errores en función de las técnicas de computación*. Cuba : s.n.

GLOSARIO

-A-

Aleatorio: Al azar, que no sigue un patrón, secuencia u orden determinado.

Algoritmo: Un conjunto de reglas bien definidas para la solución de un problema en un número finito de pasos.

Arco: Es la unión entre dos nodos y representa la dependencia entre dos variables.

-E-

Entrenamiento: Adiestramiento, técnica que se realiza para perfeccionar el ejercicio de una actividad, para mejorar un comportamiento

Estocástico: Un proceso estocástico es una sucesión de variables aleatorias indexadas por una variable continua o discreta. Cada una de las variables aleatorias del proceso tiene su propia función de distribución de probabilidad.

-H-

Heurística: Regla que permite orientar un algoritmo hacia la solución de un problema. Técnica de programación que permite a un sistema la creación gradual de un valor óptimo para una variable específica por medio del registro de los valores obtenidos en operaciones anteriores. Técnica empleada en los sistemas de inteligencia artificial.

-I-

IA: Inteligencia Artificial

Incertidumbre: Validación de los resultados de una medición de alta precisión.

Iteraciones: Número determinado de veces que se realiza un proceso.

-N-

Nodo: Un nodo es una variable aleatoria que puede tener varios estados, cada nodo será una estructura o registro que dispondrá de varios campos.

-R-

RNA: Red Neurona Artificial.

-T-

Topología: Es la organización y disposición de los elementos de una determinada estructura formando otras estructuras.