**INTELIGENCIA ARTIFICIAL**

**JHON JAMES CANO SÁNCHEZ**

**DOCENTE**

**CARLOS ALBERTO LONDOÑO**

**CORPORACIÓN DE ESTUDIOS TECNOLÓGICOS DEL NORTE DEL VALLE**

**SEPTIEMBRE 04 DE 2016.**

**TABLA DE CONTENIDO**

[REDES NEURONALES 1](#_Toc460773793)

[Consultar​ ​la​ ​historia​ ​de​ ​las​ ​redes​ ​neuronales​ ​artificiales,​ ​y​ ​haciendo​ ​uso​ ​de​ ​la herramienta​ ​text2mindmap,​ ​crear​ ​un​ ​mapa​ ​conceptual,​ ​que​ ​permita​ ​evidenciar​ ​los casos​ ​más​ ​importantes. 1](#_Toc460773794)

[Nombre​ ​6​ ​o​ ​más​ ​ventajas​ ​y​ ​desventajas​ ​que​ ​tiene​ ​el​ ​uso​ ​de​ ​las​ ​redes​ ​neuronales artificiales​ ​para​ la​ ​solución​ ​de​ ​problemas. 1](#_Toc460773795)

[Nombre​ ​10​ ​aplicaciones​ ​de​ ​las​ ​redes​ ​neuronales 3](#_Toc460773796)

[¿Que​ ​son​ ​funciones​ ​de​ ​activación,​ ​cuales​ ​existen​ ​y​ ​para​ ​cuáles​ ​redes​ ​neuronales​ ​se aplican? 4](#_Toc460773797)

[Tipos de Funciones de Activación 4](#_Toc460773798)

[¿Qué es un Perceptrón? 5](#_Toc460773799)

[Adaline 8](#_Toc460773800)

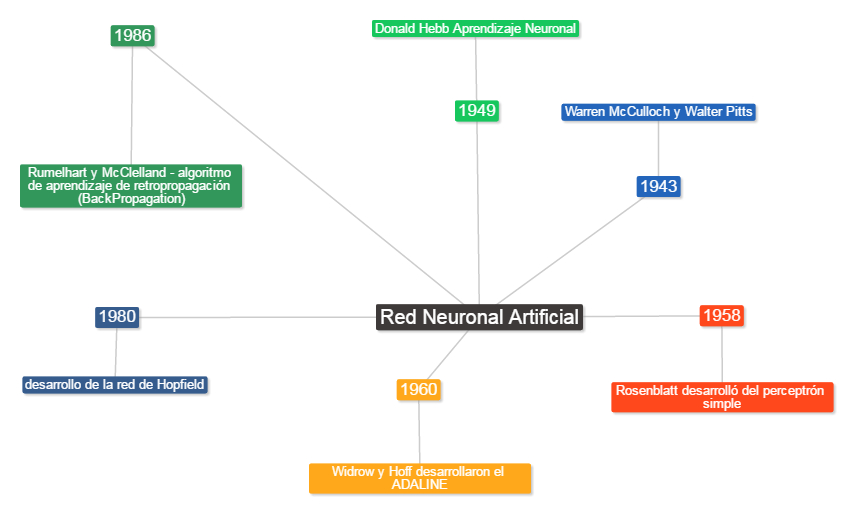
[¿Qué​ ​es​ ​y​ ​para​ ​que​ ​se​ ​usa​ ​las​ ​redes​ ​de​ ​retropropagación (BACKPROPAGATION)? 11](#_Toc460773801)

[¿Qué​ ​es​ ​el​ ​teorema​ ​de​ ​Kolmogorov,​ ​explicar? 14](#_Toc460773802)

[Bibliografía 16](#_Toc460773803)

# REDES NEURONALES

## Consultar​ ​la​ ​historia​ ​de​ ​las​ ​redes​ ​neuronales​ ​artificiales,​ ​y​ ​haciendo​ ​uso​ ​de​ ​la herramienta​ ​text2mindmap,​ ​crear​ ​un​ ​mapa​ ​conceptual,​ ​que​ ​permita​ ​evidenciar​ ​los casos​ ​más​ ​importantes.



## Nombre​ ​6​ ​o​ ​más​ ​ventajas​ ​y​ ​desventajas​ ​que​ ​tiene​ ​el​ ​uso​ ​de​ ​las​ ​redes​ ​neuronales artificiales​ ​para​ la​ ​solución​ ​de​ ​problemas.

Ventajas

Aprendizaje: Las RNA tienen la habilidad de aprender mediante una etapa que se llama etapa de aprendizaje. Esta consiste en proporcionar a la RNA datos como entrada a su vez que se le indica cuál es la salida (respuesta) esperada.

Auto organización: Una RNA crea su propia representación de la información en su interior, quitándole esta tarea al usuario.

Tolerancia a fallos: Debido a que una RNA almacena la información de forma redundante, ésta puede seguir respondiendo de manera aceptable aun si se daña parcialmente.

Flexibilidad: Una RNA puede manejar cambios no importantes en la información de entrada, como señales con ruido u otros cambios en la entrada (por ejemplo si la información de entrada es la imagen de un objeto, la respuesta correspondiente no sufre cambios si la imagen cambia un poco su brillo o el objeto cambia ligeramente).

Tiempo real: La estructura de una RNA es paralela, por lo cual si esto es implementado con computadoras o en dispositivos electrónicos especiales, se pueden obtener respuestas en tiempo real.

Desventajas

Complejidad de aprendizaje para grandes tareas, cuanto más cosas se necesiten que aprenda una red, más complicado será enseñarle.

Tiempo de aprendizaje elevado. Esto depende de dos factores: primero si se incrementa la cantidad de patrones a identificar o clasificar y segundo si se requiere mayor flexibilidad o capacidad de adaptación de la red neuronal para reconocer patrones que sean sumamente parecidos, se deberá invertir más tiempo en lograr que la red converja a valores de pesos que representen lo que se quiera enseñar.

No permite interpretar lo que se ha aprendido, la red por si sola proporciona una salida, un número, que no puede ser interpretado por ella misma, sino que se requiere de la intervención del programador y de la aplicación en si para encontrarle un significado a la salida proporcionada.

Elevada cantidad de datos para el entrenamiento, cuanto más flexible se requiera que sea la red neuronal, mas información tendrá que enseñarse para que realice de forma adecuada la identificación.

Falta de reglas definitorias que ayuden a realizar una red para un problema dado.

## Nombre​ ​10​ ​aplicaciones​ ​de​ ​las​ ​redes​ ​neuronales

* Gestión bibliotecaria
* Clasificación de la información
* Diseño de interface
* Filtrado de información
* Búsqueda incompleta
* Descubrimiento de información
* OCR reconocimiento óptico de caracteres
* Síntesis de voz desde texto
* Reconocimiento de patrones de movimiento
* Biometría

## ¿Que​ ​son​ ​funciones​ ​de​ ​activación,​ ​cuales​ ​existen​ ​y​ ​para​ ​cuáles​ ​redes​ ​neuronales​ ​se aplican?

La función de activación combina el potencial post-sináptico, que proporciona la función de propagación, con el estado actual de la neurona para conseguir el estado futuro de activación de la neurona. Sin embargo, es muy común que las redes neuronales no tomen su propio estado como un parámetro y que por tanto no se considere. Esta función es normalmente creciente monótona.

## Tipos de Funciones de Activación

Lineal

Algunas redes neuronales usan esta función de activación como el Adeline por su eficiencia y facilidad.

Escalón

Esta función es la más usada para redes neuronales binarias ya que no es lineal y es muy simple. Algunas redes que usan esta función son el Perceptrón y Hopfield. Para redes que trabajan en el rango [-1,1] se usa la función signo.

Hiperbólicas o tangenciales

Las redes con salidas continuas, como el Perceptrón multicapa con retropropagación, usan esta función ya que su algoritmo de aprendizaje necesita una función derivable.

## ¿Qué es un Perceptrón?

La contribución clave de Rosenblatt fue la introducción de una regla de aprendizaje para la formación de redes perceptrón para resolver problemas de reconocimiento de patrones. Demostró que su regla de aprendizaje siempre convergirá a los pesos correctos de la red, si existen pesos que solucionan el problema. El Perceptrón pudo incluso aprender cuando se inicializaba con valores aleatorios de sus pesos y vías.

El Perceptrón es limitado. Dichas limitaciones fueron publicadas en el libro Perceptrons por Marvin Minsky y Seymour Papert. Ellos demostraron que las redes perceptrón eran incapaces de implementar ciertas funciones elementales. No fue sino hasta la década de los 8O’s que estas limitaciones fueron superadas con las redes perceptrón mejoradas (multicapa) asociadas con reglas de aprendizaje.

Fórmula​ ​matemática,​ ​explique​ ​sus​ ​términos

• Ei = el estado de la neurona de la capa de entrada i,

• Pij = El peso actual asociado a la sinapsis que une la neurona i de la capa de entrada y la neurona j de la capa de salida.

• Ej = El estado de la neurona de la capa de salida j

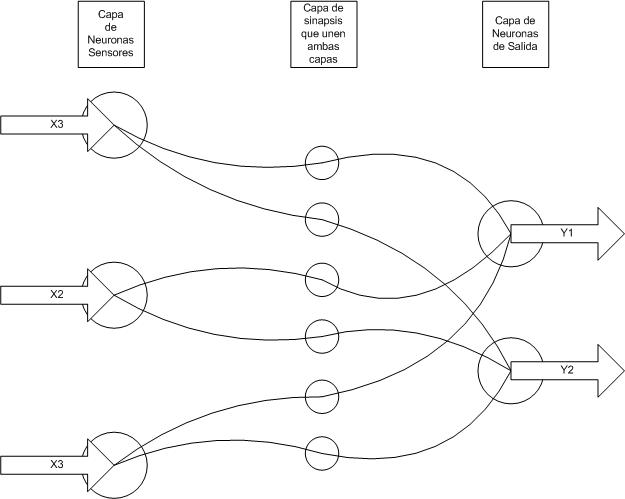
• Sj = El valor deseado pora esa neruona

• Pijnuevo = El peso Pij actualizado.

• TASA =Es una constante entre 0 y 1 que indica cuanto aprende la red

Pijnuevo = Pij + Tasa\*(( Ej - Sj) \*Ei)

Dibuje​ ​la​ ​estructura



¿Para​ ​qué​ ​se​ ​usa?

Es un clasificador, asigna a un vector de N valores un valor binario, usando una transformación no lineal. Así cada vector pertenece a una de las particiones que crea el perceptrón.

El perceptrón es una máquina de computación universal y tiene la expresividad equivalente a la lógica binaria ya que se puede crear un perceptrón que tenga el mismo comportamiento que una función boleana NAND y a partir de esta función se puede crear cualquier otra función boleana.

¿Cuál​ ​es​ ​su​ ​función​ ​de​ ​activación?

Función Escalón (Hardlim)

¿Cómo​ ​se​ ​entrena​ ​un​ ​perceptrón,​ ​indique​ ​el​ ​nombre​ ​del​ ​algoritmo​ ​y​ ​sus pasos?

Tomar un patrón al azar de la lista.

Se establece el patrón de entrada en los sensores, la capa de entrada.

Se establecen los valores deseados en las neuronas de la capa de salida.

Se actualizan las neuronas de la capa de Salida.

Solicitar que aprendan todas las sinapsis

Si las sinapsis han cambiado volver al paso 1

Paso 0: Inicialización

Inicializar los pesos sinápticos con números aleatorios del intervalo [-1,1]. Ir al paso 1 con k=1

Paso 1: (k-ésima iteración)

Calcular



Paso 2: Corrección de los pesos sinápticos Si z(k)≠y(k) modificar los pesos sinápticos según la expresión:



Paso 3: Parada Si no se han modificado los pesos en las últimas p iteraciones, es decir,



Parar. La red se ha estabilizado. En otro caso, ir al Paso 1 con k=k+1.

Nombre​ ​5​ ​ejemplos​ ​donde​ ​se​ ​evidencie​ ​el​ ​uso​ ​del​ ​perceptrón.

* formar el tiempo pasado de los verbos en ingles
* leer texto en inglés y manuscrito
* Pilotos automáticos de alto desempeño.
* Comandos por voz.
* Identificador de firmas

## Adaline

El Adaline (Adaptative Linear Element) es un tipo de red neuronal artificial desarrollada por el profesor Bernie Widrow y su alumno Ted Hoff en la Universidad de Stanford en 1959. El modelo está basado en la Neurona de McCulloch-Pitts. ADALINE, es parecida al perceptrón pero introduce en el modelo el ajuste de los parámetros en base el error cometido por la red.

Fórmula​ ​matemática,​ ​explique​ ​sus​ ​términos

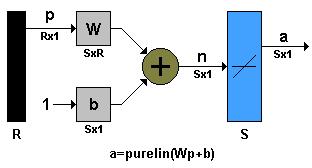
Monografias.comSiendo:

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

P: Número de informaciones que debe aprender la red.

Monografias.comError cometido en el aprendizaje de la información *k-ésima.* Se trata de encontrar pesos para las conexiones que minimicen esta función de error. Para ello, el ajuste de los pesos de las conexiones de la red se puede hacer de forma proporcional a la variación relativa del error que se obtiene al variar el peso correspondiente:

Monografias.com Mediante este procedimiento, se llegan a obtener un conjunto de pesos con los que se consigue minimizar el error medio.  
Dibuje​ ​la​ ​estructura



En donde:

p: Patrones de entrada

b: Umbrales de activación

a: Salida de la neurona

¿Para​ ​qué​ ​se​ ​usa?

Procesamiento de señales

Filtros capaces de eliminar ruidos en señales portadoras de información.

Filtros adaptativos para predecir el valor futuro de una señal a partir de su valor actual.

Eliminación de ecos en circuitos telefónicos.

¿Cuál​ ​es​ ​su​ ​función​ ​de​ ​activación?

¿Cómo​ ​se​ ​entrena​ ​una​ ​red​ ​Adaline​ ​indique​ ​el​ ​nombre​ ​del​ ​algoritmo​ ​y​ ​sus pasos?

Algoritmo LMS (LAST MEAN SQUARE)

1. Inicializar pesos (w1, ..., wn) y threshold (w0)

2. Presentar vector de entrada (x1,...,xn) y la salida deseada d(t)

3. Calcular la salida



Donde Fh(a) = 1 sí a>0 y

= -1 sí a<=0



4. Adaptar los pesos

Donde 0 < i < n y  es la tasa de aprendizaje

5. Repetir los pasos 2 a 4 hasta que las salidas reales y las deseadas sean iguales para todos los vectores del conjunto de entrenamiento

Siguiendo este método se garantiza que, para un conjunto de entrenamiento adecuado, después de un número finito de iteraciones el error se reduce a niveles aceptables. El número de iteraciones necesarias y el nivel de error deseado depende de cada problema particular.

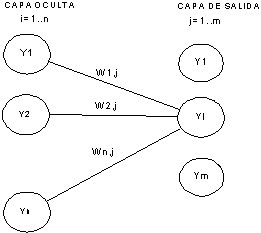
## ¿Qué​ ​es​ ​y​ ​para​ ​que​ ​se​ ​usa​ ​las​ ​redes​ ​de​ ​retropropagación (BACKPROPAGATION)?

Consiste en propagar el error hacia atrás, es decir, de la capa de salida hacia la capa de entrada, pasando por las capas ocultas intermedias y ajustando los pesos de las conexiones con el fin de reducir dicho error. Hay distintas versiones o reglas del algoritmo de retropropagación y distintos arquitecturas conexionistas a las que pueden ser aplicados.

Durante mucho tiempo no se dispuso de algoritmos para entrenar redes multicapa, y como las redes de una capa estaban muy limitadas en cuanto a lo que eran capaces de representar, el campo de las redes neuronales artificiales estaba estancado. La invención y perfeccionamiento del algoritmo de retropropagación dio un gran impulso al desarrollo de este campo. Tiene un buen fundamento matemático y a pesar de sus limitaciones ha expandido enormemente el rango de problemas donde se aplican las redes neuronales artificiales.

Explique​ ​el​ ​algoritmo​ ​BACKPROPAGATION,​ ​usando​ ​imágenes.

1. Pesos de las conexiones aleatorios.
2. Se presenta un patrón de entrada y calculamos la salida.
3. Dada una unidad j-ésima de la capa de salida y unidades i-ésimas de la capa oculta inmediatamente anterior, se calcula la entrada total ponderada y la salida o activación de la misma.



http://rna.50webs.com/images_tutorial/Image25.gif

http://rna.50webs.com/images_tutorial/Image26.gif

1. Una vez computadas las actividades de todas las unidades de salida se calcula una estimación del error, generalmente una función cuadrática de los errores individuales cometidos por cada unidad, siendo cada error individual la diferencia entre la salida deseada y la obtenida.

http://rna.50webs.com/images_tutorial/Image27.gif Siendo dj la salida deseada para la unidad j-ésima

1. Cómputo de la rapidez de variación del error al cambiar la actividad de cada unidad de salida (EA, error respecto a la actividad)

http://rna.50webs.com/images_tutorial/Image28.gif

Es justamente la diferencia entre la salida deseada y la salida real obtenida, es decir, la diferencia entre la actividad deseada y la actividad real

1. Cómputo de la rapidez de variación del error al cambiar la entrada total que recibe cada unidad de salida.

http://rna.50webs.com/images_tutorial/Image29.gif

Es igual a la tasa de variación del error al variar su activación multiplicado por la tasa de variación de la activación al cambiar su entrada

1. Cómputo de la rapidez de variación del error al ser modificado un peso de la conexión aferente a una unidad de salida.

http://rna.50webs.com/images_tutorial/Image30.gif

Es igual a la tasa de variación del error al variar su entrada, por la tasa de variación de la entrada al variar ese peso.

1. Conociendo EA para las unidades de cualquier capa se puede calcular d y EP con las expresiones ya conocidas.

http://rna.50webs.com/images_tutorial/Image34.gif

http://rna.50webs.com/images_tutorial/Image35.gif

1. Se dispone de la tasa de variación del error respecto al peso de una conexión (EP), se puede usar distintas reglas para modificar ese peso en aras a reducir dicho de error. Una de las primeras reglas que aprovechó este algoritmo es la *regla delta generalizada*, que calcula el incremento a aplicar a un peso como una proporción directa de la tasa de variación del error.   
   Siendo h el coeficiente de aprendizaje, típicamente con valores comprendidos entre 0.01 y 1.0.  
   http://rna.50webs.com/images_tutorial/Image36.gif  
   http://rna.50webs.com/images_tutorial/Image37.gif

## ¿Qué​ ​es​ ​el​ ​teorema​ ​de​ ​Kolmogorov,​ ​explicar?

Es un resultado de sistemas dinámicos sobre la persistencia de movimientos cuasi periódicos. Este teorema resuelve parcialmente el problema de los divisores pequeños (que origina problemas de convergencia en sistemas con múltiples frecuencias). El teorema explica cómo se modifica el aspecto de las trayectorias de un sistema integrable bajo pequeñas perturbaciones.

El movimiento en un sistema integrable está confinado a una [hiper]superficie toroidal. Diferentes condiciones iniciales del sistema originan diferentes toros en el espacio fásico. Que las trayectorias de un sistema integrable de dimensión *n* están confinadas a hiper superficies de tipo{\displaystyle \mathbb {T} ^{n}}  pueden deducirse del tratamiento de las variables acción-ángulo, al existir *n* variables "ángulo" periódicas.

El teorema establece que, si un sistema está sometido a una pequeña perturbación no lineal, algunos toros serán deformados y otros destruidos. Los que sobreviven son aquellos que tienen un cociente de frecuencias suficientemente irracional. Es decir, se destruyen aquellos cuyo cociente de frecuencias se acerca más a un número racional, dados por la relación

{\displaystyle \left\vert {\frac {\omega \_{2}}{\omega \_{1}}}-{\frac {m}{s}}\right\vert >{\frac {k(\epsilon )}{\sqrt {s}}}} 

Con {\displaystyle k(\epsilon \rightarrow 0)\rightarrow 0} . El último toro en destruirse es el más irracional de todos (el que guarda mayor semejanza con el número áureo). Informalmente el teorema establece que:

"Para perturbaciones suficientemente pequeñas, casi todos los toros invariantes se preservan [en el sistema perturbado] (excluyendo aquellos con vectores de frecuencia racionales)"

# Bibliografía

Chavez, R. (s.f.). *Ronald Chavez*. Obtenido de http://ronaldchavezblog.blogspot.com.co/2010/11/red-adeline.html

INC, H. (s.f.). *Hugo INC*. Obtenido de http://www.hugo-inc.com/RNA/Unidad%202/2.2.2.html

Neuronales, R. (s.f.). *Redes Neuronales*. Obtenido de http://www.redes-neuronales.com.es/tutorial-redes-neuronales/funciones-de-las-neuronas-artificiales.htm

Slideshare. (s.f.). *Slideshare*. Obtenido de http://es.slideshare.net/mentelibre/red-neuronal-backpropagation

Thales. (s.f.). *Thales*. Obtenido de http://thales.cica.es/rd/Recursos/rd98/TecInfo/07/capitulo4.html

UDLAP, C. (s.f.). *Catarina UDLAP*. Obtenido de http://catarina.udlap.mx/u\_dl\_a/tales/documentos/lem/oropeza\_c\_ca/capitulo4.pdf

Wikipedia. (s.f.). *Wikipedia*. Obtenido de https://es.wikipedia.org/wiki/Teorema\_de\_Kolmog%C3%B3rov-Arnold-Moser

Wikipedia. (s.f.). *Wikipedia*. Obtenido de https://es.wikipedia.org/wiki/Ley\_cero-uno\_de\_Kolmog%C3%B3rov