TALLER 3

PRESENTADO POR:

ASDRUBA ANDRES VELEZ VELEZ

LUIS EDIN IBARGUEN MOSQUERA

PRESENTADO A:

CARLOS ALBERTO LONDOÑO LOAIZA

ING. DE SISTEMAS

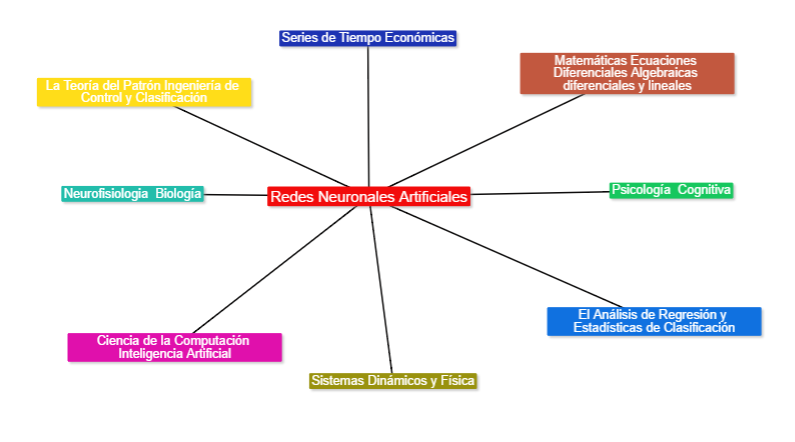
CORPORACION DE ESTUDIOS TECNOLOGICOS DEL NORTE DEL VALLE

COTECNOVA

CARTAGO VALLE DEL CAUCA

2018-02-28

**1.**



2. Ventajas y desventajas que tiene el uso de las redes neuronales artificiales para la solución de problemas?

Ventajas:

* Las redes neuronales pueden sintetizar algoritmos a través de un proceso de aprendizaje.
* Para utilizar la tecnología neuronal no es necesario conocer los detalles matemáticos, solo se requiere estar familiarizado con los datos de trabajo
* La solución de problemas no lineales es uno de los fuertes de las redes neuronales.
* Las redes son robustas, pueden fallar algunos elementos de procesamiento pero la red continua trabajando, esto es contrario a lo que sucede en programación tradicional

Las desventajas son:

Que las redes neuronales se deben entrenar para cada problema. Además, es necesario realizar múltiples pruebas para determinar la arquitectura adecuada, el entrenamiento es largo y consume varias horas de CPU.

3. 10 aplicaciones de las redes neuronales.

Las características de las redes neuronales hacen que sus posibles aplicaciones sean muy amplias

* Finanzas, ventas, manufacturación.
* predicción de índices.
* Detención de fraudes.
* Identificación de falsificaciones.
* Interpretación de firmas
* Predicción de la rentabilidad de acciones
* Campañas de venta
* Reconocimiento de gráficos
* Inspección de calidad
* Reconocimiento de escritura manual cursiva

4. R// proporciona el estado de activación de la neurona en función del estado anterior y del valor postsináptico

Una neurona biológica puede estar activa (excitada) o inactiva (no excitada); es decir, que tiene un “estado de activación”. Las neuronas artificiales también tienen diferentes estados de activación; algunas de ellas solamente dos, al igual que las biológicas, pero otras pueden tomar cualquier valor dentro de un conjunto determinado.

La función activación calcula el estado de actividad de una neurona; transformando la entrada global (menos el umbral, Θi) en un valor (estado) de activación, cuyo rango normalmente va de (0 a 1) o de (–1 a 1). Esto es así, porque una neurona puede estar totalmente inactiva (0 o –1) o activa (1).

La función activación, es una función de la entrada global (gini) menos el umbral (Θi). Las funciones de activación más comúnmente utilizadas se detallan a continuación:

1) Función lineal:

Los valores de salida obtenidos por medio de esta función de activación serán: a·(gini - Θi), cuando el argumento de (gini - Θi) esté comprendido dentro del rango (-1/a, 1/a). Por encima o por debajo de esta zona se fija la salida en 1 o –1, respectivamente. Cuando a = 1 (siendo que la misma afecta la pendiente de la gráfica), la salida es igual a la entrada.

2) Función sigmoidea:

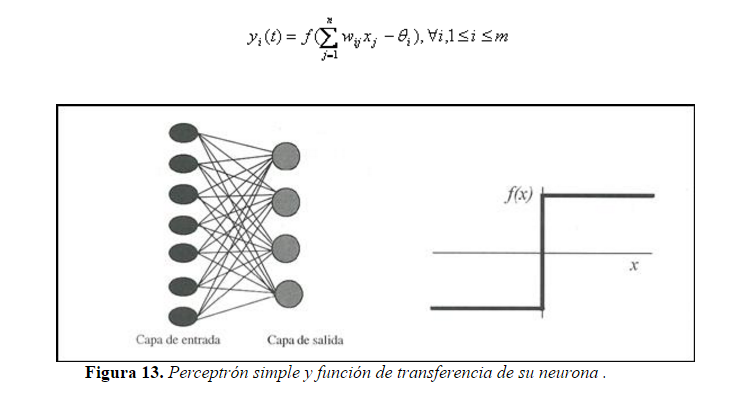
Los valores de salida que proporciona esta función están comprendidos dentro de un rango que va de 0 a 1. Al modificar el valor de g se ve afectada la pendiente de la función de activación.

3) Función tangente hiperbólica:

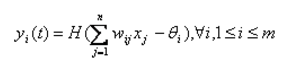
Los valores de salida de la función tangente hiperbólica están comprendidos dentro de un rango que va de -1 a 1. Al modificar el valor de g se ve afectada la pendiente de la función de activación

5 R//.Perceptron

Este modelo neuronal fue introducido por Rosenblatt a finales de los años cincuenta. La estructura del perceptrón se inspira en las primeras etapas de procesamiento de los sistemas sensoriales de los animales (por ejemplo, el de visión), en los cuales la información va atravesando sucesivas capas de neuronas, que realizan un procesamiento progresivamente de más alto nivel.

El perceptrón simple es un modelo neuronal unidireccional, compuesto por dos capas de neuronas, una de entrada y otra de salida (Figura 19). La operación de una red de este tipo, con n neuronas de entrada y m neuronas de salida, se puede expresar de la siguiente forma:

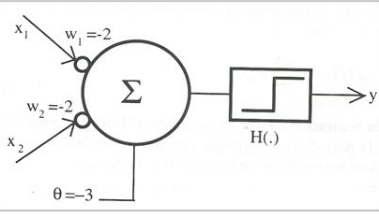
Las neuronas de entrada no realizan ningún cómputo, únicamente envían la información (en principio consideraremos señales discretas {0, 1}) a las neuronas de salida. La función de activación de las neuronas de la capa de salida es de tipo escalón (Ver Figura 5). Así, la operación de un perceptrón simple puede escribirse



con H(.) la función Heavside o escalón..

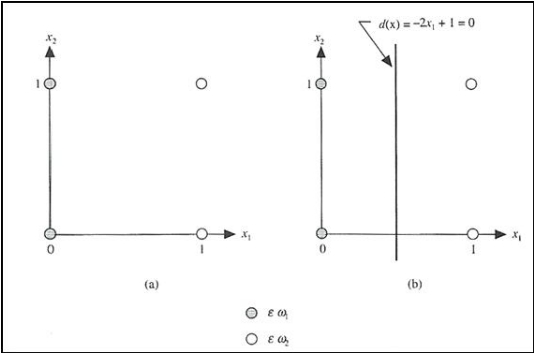
Por tanto, concluimos añadiendo que el perceptrón simple está formado por *dispositivos de umbral* y, por tanto, son útiles para la representación de funciones booleanas.

*Ejemplo:* Sea el perceptrón que se muestra en la figura 14, tal que su salida es Screenshot_3.png



**Figura 14.** *Perceptrón que implementa una puerta*

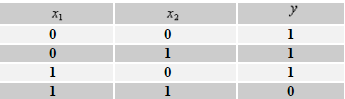
*lógica NAND de dos entradas*



Las entradas http://grupo.us.es/gtocoma/pid/pid10/RedesNeuronales_archivos/image150.gif y http://grupo.us.es/gtocoma/pid/pid10/RedesNeuronales_archivos/image152.gifconstituyen un plano, de tal forma que si lo representamos y marcamos los posibles valores de las entradas obtenemos el resultado mostrado en la ilustración 15.

**Figura 15.***A) Patrones que pertenecen a dos clases. B) Límite de decisión determinado por entrenamiento.*

Dando valores lógicos a http://grupo.us.es/gtocoma/pid/pid10/RedesNeuronales_archivos/image150.gif y http://grupo.us.es/gtocoma/pid/pid10/RedesNeuronales_archivos/image152.gif podemos construir la siguiente tabla de verdad:



Podemos comprobar que se trata de la función lógica NAND.

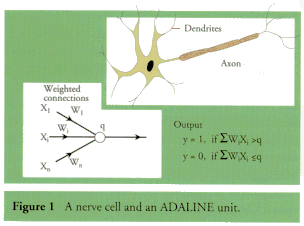
NOTA: Puede demostrarse que un nodo de tipo umbral solamente puede implementar funciones linealmente separables, como la NAND. La XOR (OR-Exclusiva) no es linealmente separable, por lo que no puede ser implementada por un nodo sencillo como el anterior.

**6- Adaline**

En la década de 1950, con el avance de las computadoras, se hizo posible simular una red neuronal. Nathanial Rochester, de los laboratorios de investigación de la IBM, dio el primer paso en la simulación de la red neuronal, pero desafortunadamente fracaso. Posteriormente se hicieron otras simulaciones ya con éxito.

En 1959, Bernard Widrow y Marcian Hoff de la universidad de Stanford desarrollaron un modelo llamado “ADALINE” y “MADALINE”. ADALINE proviene de ADAptive LINear Element y MADALINE de Multiple ADAptive LINear Element (Many Adalines).

ADALINE fue desarrollado para el reconocimiento de patrones binarios, por ejemplo, predecir el siguiente bit en una línea telefónica. En la figura 1 podemos observar la representación de una ADALINE y su analogía con una Neurona.



ADALINE funciona tomando la suma de los pesos de las entradas y produce una salida con 0 o 1 dependiendo si pasa o no un umbral, esto haciendo analogía al funcionamiento de una neurona que se dispara si la actividad total procedente de las conexiones con las otras neuronas sobrepasa un nivel.

Varias ADALINE pueden ser organizadas en capas de tal manera que se obtengan grandes arquitecturas formando una red MADALINE la cual produce funciones más complicadas.

MADALINE fue la primera red neuronal aplicada a un problema real, se usó como un filtro para eliminar el eco en las líneas telefónicas.

En los años siguientes el neuro-biólogo Frank Rosenblatt comenzó a trabajar con el perceptrón. Se encontró que una simple capa de perceptrones era suficiente para clasificar un conjunto de valores continuos en una de dos clases.

Desafortunadamente el perceptrón estaba limitado y fue probado por Marvin Minsky y Seymour Papert durante los años de desilusión en el libro “Perceptrons”.

Durante los próximos 20 años se dejó atrás el estudio de las redes neuronales y fue hasta 1982 con John Hopfield que se retomó el interés en la materia. Hopfield presentó un documento a la Academia Nacional de Ciencias en el cual a través de análisis matemático mostraba como las redes neuronales funcionan y para que podrían servir.

# ADALINE

## ADALINE que por sus siglas en inglés significa ADAptive LINear Element es un dispositivo que consta de un solo elemento de procesamiento, por lo que técnicamente no es una red. Sin embargo, es un elemento muy importante, ya que de él se derivan redes más complejas. En la siguiente figura se muestran los elementos de una ADALINE.

## 

#### ADAptive LINear Element

Una ADALINE consiste de un ALC (Adaptive Linear Combiner) y un cuantizador (función bipolar de salida). Se alimenta con un vector de entrada (valores observados) y con una entrada constante igual a 1 denominada sesgo (bias). Posteriormente se efectúa una suma ponderada de los valores de entrada con sus pesos asociados; si el resultado es positivo, la salida del ADALINE es 1, en caso contrario es 0 (o -1). En consecuencia, ADALINE sólo resuelve adecuadamente problemas binarios linealmente separables.

Algoritmo de Entrenamiento

El entrenamiento del ADALINE está basado en la regla LMS (Least Mean Square) que busca minimizar el error cuadrático medio por medio de la regla delta.

Dado un vector de entrada (x0, ..., xn) donde x0 =1, el correspondiente valor de pesos (w0, ..., wn) y el valor deseado de salida d, el error cuadrático es el siguiente:



El objetivo del método es minimizar dicho error mediante la modificación del vector de pesos (w0, ..., wn) sumándole un **w** de tal forma que nos acerquemos al error mínimo en la dirección del gradiente negativo, es decir, lo más rápidamente posible.

El procedimiento de derivación se presenta a continuación:



Por lo que el error local será reducido más rápidamente si se ajustan los pesos de acuerdo a la regla delta:



El algoritmo de entrenamiento se presenta a continuación:

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.

**Adaline/ADALINEYMADALINE2.doc**

**7- ¿Qué es y para que se usa las redes de retropropagación (backpropagation)**

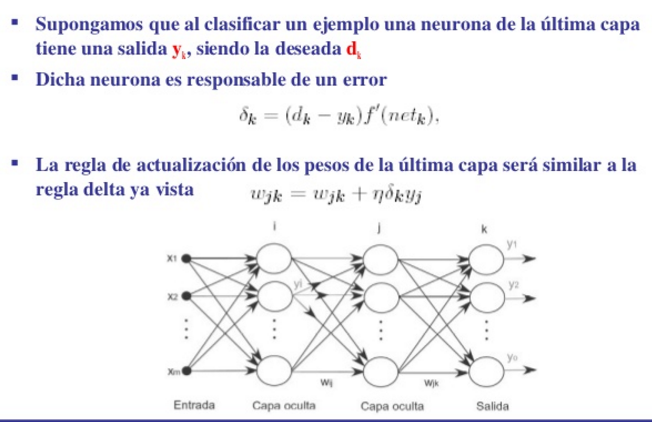
Al hablar de redes de retropropagación o redes de propagación hacia atrás hacemos referencia a un algoritmo de aprendizaje más que a una arquitectura determinada. La retropropagación consiste en propagar el error hacia atrás, es decir, de la capa de salida hacia la capa de entrada, pasando por las capas 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 distintas 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.

Al parecer el algoritmo fue ideado a principios de los 70 por Werbos, y redescubierto a principios de los 80 por Parker y Rumelhart independientemente, sin embargo, no se hizo popular hasta 1986, cuando Rumerlhart, Hinton y Williams presentaron una descripción clara y concisa del mismo. Y es que en un primer momento no se valoró como se merecía. El hecho de que permaneciera en el olvido tanto tiempo también debe ser una consecuencia de la condición interdisciplinar del campo, repartido entre las matemáticas y ciencias de la computación, las neurociencias y la psicología.

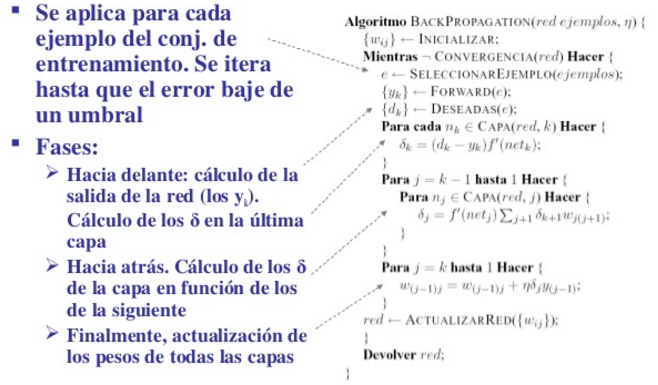
Desde la fecha clave de 1986 han surgido nuevas versiones que han tratado de aumentar la velocidad de convergencia del algoritmo y han tratado de superar algunos de sus inconvenientes, como la tendencia a alcanzar mínimos locales y no globales, punto que será discutido más tarde.

**8. explique algoritmo backpropagation**

****

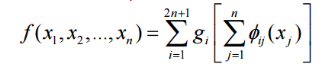
[**https://www.slideshare.net/alexjhampierrojasherrera/aprendizaje-redes-neuronales**](https://www.slideshare.net/alexjhampierrojasherrera/aprendizaje-redes-neuronales)

**backpropagation-algotitmo**

****

**10. teorema de kolmogorov**

Cualquier función continua f(x1,x2,……xn) definida en [0 1]n , n>=2, se puede representar mediante la expresión



Donde las funciones gi son funciones continuas y reales de una sola variable, elegidas adecuadamente, y las funciones oij son continuas y monótonas crecientes independientes de f. este resultado establece que cualquier resultado vectorial continua de Rn en-Rm definida sobre un conjunto compacto se puede expresar en términos de sumas y composiciones de un numero dinito de funciones de una sola variable. Sin embargo, las rede3s neuronales constituyen una clase de funciones mas reducida que la establecida por el teorema de kolmogorov ya que limitan funciones oij a funciones digmoidales.

**Bibliografías**

<https://www.slideshare.net/alexjhampierrojasherrera/aprendizaje-redes-neuronales>

<http://repositoriodigital.ipn.mx/bitstream/123456789/8628/1/Archivo%20que%20incluye%20portada,%20%C3%ADndice%20y%20texto.pdf>

<https://netixzen.blogspot.com.co/2015/03/perceptron-simple-explicacion-y.html>

<https://docs.google.com/presentation/d/1lBuF0J0UUkIKC5OrOVUZe2soIufsQeB6PLFw0dhOsfQ/edit#slide=id.g34251c74dc_0_134>