INTELIGENCIA ARTIFICIAL

HERIBERTO DAVID YEPES

CRISTHIAN URREGO SALAZAR

CORPORACION DE ESTUDIOS

TECNOLÓGICOS DEL NORTE DEL VALLE

INGENIERIA DE SISTEMAS

CARTAGO

2018

INTELIGENCIA ARTIFICIAL

HERIBERTO DAVID YEPES

CRISTHIAN URREGO SALAZAR

TALLER # 3

PROFESOR

CARLOS ALBERTO LONDOÑO

INGENIERO EN SISTEMAS

CORPORACION DE ESTUDIOS

TECNOLÓGICOS DEL NORTE DEL VALLE

INGENIERIA DE SISTEMAS

CARTAGO

2018

**TALLER # 3**

**1.**

**Historia de las redes neuronales**

Conseguir diseñar y construir máquinas capaces de realizar procesos con cierta inteligencia ha sido uno de los principales objetivos de los científicos a lo largo de la historia. De los intentos realizados en este sentido se han llegado a definir las líneas fundamentales para la obtención de máquinas inteligentes: En un principio los esfuerzos estuvieron dirigidos a la obtención de autómatas, en el sentido de máquinas que realizaran, con más o menos éxito, alguna función típica de los seres humanos. Hoy en día se continúa estudiando en ésta misma línea, con resultados sorprendentes, existen maneras de realizar procesos similares a los inteligentes y que podemos encuadrar dentro de la llamada Inteligencia Artificial (IA).

Las primeras explicaciones teóricas sobre el cerebro y el pensamiento ya fueron dadas ya por Platón (427-347 a.C.) y Aristóteles (348-422 a.C.). Las mismas ideas también las mantuvo Descartes (1569-1650) y los filósofos empiristas del siglo XVIII.

**1936 - Alan Turing**. Fue el primero en estudiar el cerebro como una forma de ver el mundo de la computación. Sin embargo, los primeros teóricos que concibieron los fundamentos de la computación neuronal fueron Warren McCulloch, un neurofisiólogo, y Walter Pitts, un matemático, quienes, en 1943, lanzaron una teoría acerca de la forma de trabajar de las neuronas (Un Cálculo Lógico de la Inminente Idea de la Actividad Nerviosa - Boletín de Matemática Biofísica 5: 115-133). Ellos modelaron una red neuronal simple mediante circuitos eléctricos.

**1949 - Donald Hebb.** Escribió un importante libro: La organización del comportamiento, en el que se establece una conexión entre psicología y fisiología. Fue el primero en explicar los procesos del aprendizaje (que es el elemento básico de la inteligencia humana) desde un punto de vista psicológico, desarrollando una regla de como el aprendizaje ocurría

**1950 - Karl Lashley.** En sus series de ensayos, encontró que la información no era almacenada en forma centralizada en el cerebro sino que era distribuida encima de él.

**1956 - Congreso de Dartmouth**. Este Congreso frecuentemente se menciona para indicar el nacimiento de la inteligencia artificial.

**1957 - Frank Rosenblatt.** Comenzó el desarrollo del Perceptrón. Esta es la red neuronal más antigua; utilizándose hoy en día para aplicación como reconocedor de patrones. Este modelo era capaz de generalizar, es decir, después de haber aprendido una serie de patrones podía reconocer otros similares, aunque no se le hubiesen presentado anteriormente. Sin embargo, tenía una serie de limitaciones, por ejemplo, su incapacidad para resolver el problema de la función OR-exclusiva y, en general, era incapaz de clasificar clases no separables linealmente. En 1959, escribió el libro Principios de Neurodinámica, en el que confirmó que, bajo ciertas condiciones, el aprendizaje del Perceptrón convergía hacia un estado finito (Teorema de Convergencia del Perceptrón).

**1960 - Bernard Widrow/Marcial Hoff**. Desarrollaron el modelo Adaline (ADAptative LINear Elements). Esta fue la primera red neuronal aplicada a un problema real (filtros adaptativos para eliminar ecos en las líneas telefónicas) que se ha utilizado comercialmente durante varias décadas.

**1961 - Karl Steinbeck**: Die Lernmatrix. Red neuronal para simples realizaciones técnicas (memoria asociativa).

**1967 - Stephen Grossberg.** A partir de sus conocimientos fisiológicos, ha escrito numerosos libros y desarrollado modelo de redes neuronales. Realizó una red: Avalancha, que consistía en elementos discretos con actividad que varía en el tiempo que satisface ecuaciones diferenciales continuas, para resolver actividades como reconocimiento continuo de habla y aprendizaje de los brazos de un robot.

**1969 - Marvin Minsky/Seymour Papert.** En este año surgieron críticas que frenaron, hasta 1982, el crecimiento que estaban experimentando las investigaciones sobre redes neuronales. Minsky y Papera, del Instituto Tecnológico de Massachussets (MIT), publicaron un libro Perceptrons. Probaron (matemáticamente) que el Perceptrón no era capaz de resolver problemas relativamente fáciles, tales como el aprendizaje de una función no-lineal. Esto demostró que el Perceptrón era muy débil, dado que las funciones no-lineales son extensamente empleadas en computación y en los problemas del mundo real.

**1974 - Paul Werbos.** Desarrolló la idea básica del algoritmo de aprendizaje de propagación hacia atrás (backpropagation); cuyo significado quedó definitivamente aclarado en 1985.

**1977 - Stephen Grossberg**. Teoría de Resonancia Adaptada (TRA). La Teoría de Resonancia Adaptada es una arquitectura de red que se diferencia de todas las demás previamente inventadas. La misma simula otras habilidades del cerebro: memoria a largo y corto plazo.

**1977 - Teuvo Kohonen**. Ingeniero electrónico de la Universidad de Helsinki, desarrolló un modelo similar al de Anderson, pero independientemente.

**1980 - Kunihiko Fukushima**. Desarrolló un modelo neuronal para el reconocimiento de patrones visuales.

**1985 - John Hopfield.** Provocó el renacimiento de las redes neuronales con su libro: "Computación neuronal de decisiones en problemas de optimización."

**1986 - David Rumelhart/G. Hinton.** Redescubrieron el algoritmo de aprendizaje de propagación hacia atrás (backpropagation). A partir de 1986, el panorama fue alentador con respecto a las investigaciones y el desarrollo de las redes neuronales. En la actualidad, son numerosos los trabajos que se realizan y publican cada año, las aplicaciones nuevas que surgen (sobretodo en el área de control) y las empresas que lanzan al mercado productos nuevos, tanto hardware como software (sobre todo para simulación).

**Mapa conceptual**

/Users/heriberto/Downloads/Diagrama en blanco.pdf

**2.**

|  |  |
| --- | --- |
| **Ventajas** | **Desventajas** |
| 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, descargando al usuario de esto. 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 (ej. 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 cuál si esto es implementado con computadoras o en dispositivos electrónicos especiales, se pueden obtener respuestas en tiempo real. | Complejidad de aprendizaje para grandes tareas, cuanto más cosas se necesiten que aprenda una red, mas 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 mas 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 mas flexible se requiera que sea la red neuronal, mas información tendrá que enseñarle para que realice de forma adecuada la identificación. |

**3.**

**Biología**

Aprender más acerca del cerebro

Obtención del modelo de la retina

**Empresa**

Evaluación de probabilidad de formaciones geológicas y petrolíferas

Explotación de bases de datos

Optimización de plazas y horarios en líneas de vuelo

Reconocimiento de caracteres escritos

**Medio ambiente**

Analizar tendencias y patrones

Previsión del tiempo

**Finanzas**

Previsión de la evolución de los precios

Valoración del riesgo de los créditos

Interpretación de firmas

Identificación de falsificaciones

**Manufacturación**

Robots y sistemas de control (visión artificial y sensores de presión, temperatura, etc)

Control de producción en líneas de proceso

**Medicina**

Analizadores del habla para la ayuda de audición de sordos

Monitorización en cirugía

Predicción de reacciones adversas a los medicamentos

Lectores de rayos X

Diagnóstico y tratamiento a partir de síntomas y/o datos analíticos

**Militares**

Clasificación de las señales de radar

Creación de armas intelegentes

Reconocimiento y seguimiento en el tiro al blanco

**4.**

La función de activación combina el potencial post sináptico, que nos 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.

**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 Perceptron multicapa con retro propagación, usan esta función ya que su algoritmo de aprendizaje necesita una función derivable.

|  |  |  |  |
| --- | --- | --- | --- |
| Funciones de activación | | | |
| Neurona | lineal | Escalón | Hiperbólicas o tangenciales |
| Adeline | x |  |  |
| Perceptrón |  | x | x |
| Hopfield |  | x |  |

**5.**

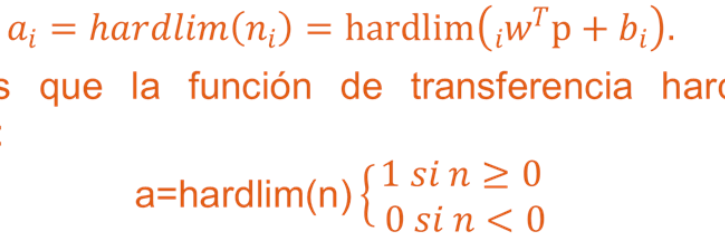
**a.**

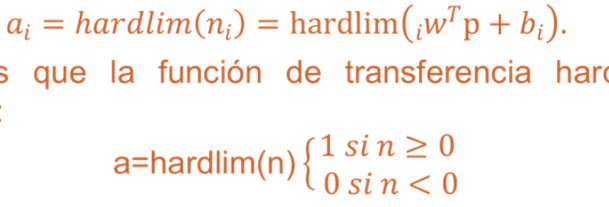
En 1943, Warren McCulloch y Walter Pitts introdujeron una de las primeras neuronas artificiales. La característica principal de su modelo de neurona es que un suma ponderada de las señales de entrada se compara con un umbral para determinar la neurona de salida. Cuando la suma es mayor o igual al umbral, la salida es 1. Cuando la suma es menor que el umbral, la salida es 0.

A finales de 1950 Frank Rosenblatt y otros investigadores desarrollaron una clase de redes neuronales llamadas perceptrones. Las neuronas de estas redes eran similares a las de McCulloch y Pitts.

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 bias.

**b.**





R cuantas entradas.

S cuantas neuronas en las capas internas y externas.

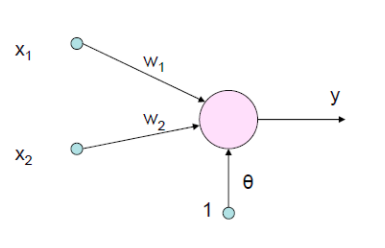
M cuantas capas

F que función de activación usar

W pesos sinápticos

B polarización que se debe usar

**c.**

****

**d.**

Se desarrolla un algoritmo capaz de generar un criterio para seleccionar un sub-grupo a partir de un grupo de componentes más grande. La limitación de este algoritmo es que si dibujamos en un plot estos elementos, se deben poder separar con un hiperplano únicamente los elementos “deseados” discriminándolos (separándolos) de los “no deseados”. El perceptrón puede utilizarse con otros perceptrones u otro tipo de neurona artificial, para formar redes neuronales más complicadas.

El perceptrón usa una matriz para representar las redes neuronales y es un discriminador terciario que traza su entrada X (un vector binario) a un único valor de salida f(x) (un solo valor binario) a través de dicha matriz.

**e.**

**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.

**f**

Para comprender como se realiza el entrenamiento de una red de neuronas, lo primero que debemos de hacer es comprender el entrenamiento de una sola neurona, para luego acoplar varios entrenamientos individuales en uno colectivo para toda la red.

Para el entrenamiento de un perceptrón lo primero que debemos hacer es inicializar aleatoriamente su vector de pesos asociado, y después ir actualizando este para conseguir mejores resultados.

|  |
| --- |
|  |
|  | http://grupo.us.es/gtocoma/pid/pid10/RedesNeuronales_archivos/image178.gif |

Para cada iteración del algoritmo se actualizará el vector de pesos:

|  |
| --- |
|  |
|  | http://grupo.us.es/gtocoma/pid/pid10/RedesNeuronales_archivos/image180.gif |

De manera que:

Siendo  un parámetro al que denominaremos tasa de entrenamiento con un valor positivo y que hará la función de hacer que la convergencia sea más o menos rápida. Es recomendable que la tasa de entrenamiento sea pequeña ( entre 0.1 y 0.2) para que los resultados sean correctos.

Esta forma de recalcular los pesos se basa en el gradiente descendente de la función de error.

**Función de error**

La función de error mide de una manera numérica la diferencia existente entre la salida ofrecida por la red y la salida correcta.

|  |
| --- |
|  |
|  | http://grupo.us.es/gtocoma/pid/pid10/RedesNeuronales_archivos/image182.gif |

Se calcula de la siguiente manera:

Donde:

-        D es el conjunto de ejemplos de entrenamiento.

-        **T**d es la salida correcta.

-        **O**d es la salida calculada por la red.

**g.**

Telecomunicaciones

Electrónica

Compresión de datos e imágenes

Servicios automáticos de información

Traducción de lenguaje hablado en tiempo real

Predicción de secuencias de códigos

Control de procesos

Análisis de fallas de circuitos

Visión de máquina

Síntesis de voz

Modelado no lineal

**6.**

**a.**

A finales de los 50’s bernard widrow empezaba a trabajar en redes neuronales, en el tiempo en el que frank rosenblatt desarrollo la regla de aprendizaje del perceptron.

En 1960 Widrow y su asesorado Marcian Hoff, presentaron la red ADALINE (adaptive linear neuron), y una regla de aprendizaje la cual denominaron algoritmo LMS (least mean square).

La red ADALINE es muy similar al perceptron, excepto que su función de transferencia es linear, en vez de escalón. Tanto el ADALINE como el perceptron sufren de la misma limitación: solo pueden resolver problemas linealmente separables

**b.**

Las m entradas representan un vector *x* de entrada que pertenece al espacio *Rm*.

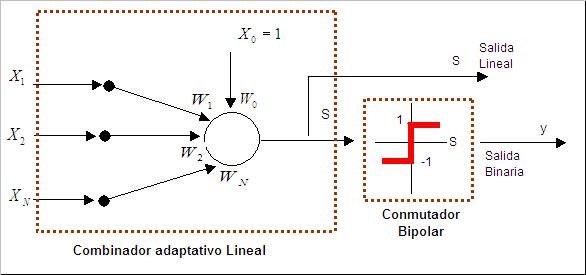
Por cada neurona, existe un vector *w* de pesos sinápticos que indican la fuerza de conexión entre los valores de entrada y la neurona. En la práctica representan la ponderación de cada entrada sobre la neurona.

Una constante *0*.

La salida *y* de la neurona se representa por la función de activación, que se define como:

Captura%20de%20pantalla%202018-02-28%20a%20la(s)%2011.34.42%20a.%20m..png

**c.**



**d.**

Las redes Adaline se basan en el objetivo de evaluar de la forma más correcta la salida, para de esta manera poder minimizar el error medio cuadrático, por tanto son muy similares al Perceptrón, lo único en lo que cambian es la transferencia en la salida.

Adaline tiene una considerable diferencia con respecto a Perceptron, puesto que en la modificación de los pesos que se dan en la preparación, la Red Adaline tiene muy presente el valor de corrección de la salida estimada con relación a la esperada.

**e.**

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

**f.**

1.- Inicializar los pesos en forma aleatoria

2.- Introducir un patrón de entrada

3.- Calcular la salida (y), compararla con la deseada (d) y obtener la diferencia (dp - yp)

4.- Multiplicar el resultado del paso anterior por la entrada correspondiente a cada uno de los pesos y ponderarla por la tasa de aprendizaje.

5.- Actualizar los pesos, sumando al valor antiguo la cantidad obtenida en el paso anterior

6.- Si no se ha cumplido el criterio de parada, regresar al paso 2, si se ha acabado todos los patrones repetir el algoritmo.

**g.**

Asociación de patrones: se puede aplicar a este tipo de problemas siempre que los patrones sean linealmente separables.

En el campo del procesamiento de señales:

Filtros de ruido: Limpiar ruido de señales transmisoras de información.

Filtros adaptativos: Un adaline es capaz de predecir el valor de una señal en el instante t+1 si se conoce el valor de la misma en los p instantes anteriores (p es >0 y su valor depende del problema). El error de la predicción será mayor o menor según qué señal queramos predecir. Si la señal se corresponde a una serie temporal el Adaline, pasado un tiempo, será capaz de dar predicciones exactas.

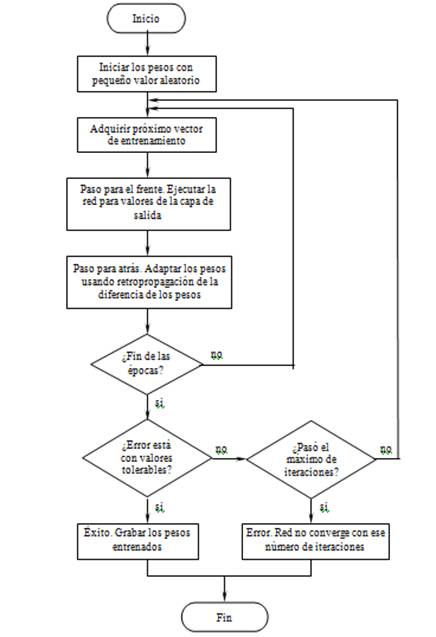
Se pueden combinar múltiples Adalines formando lo que se denomina el Madaline.

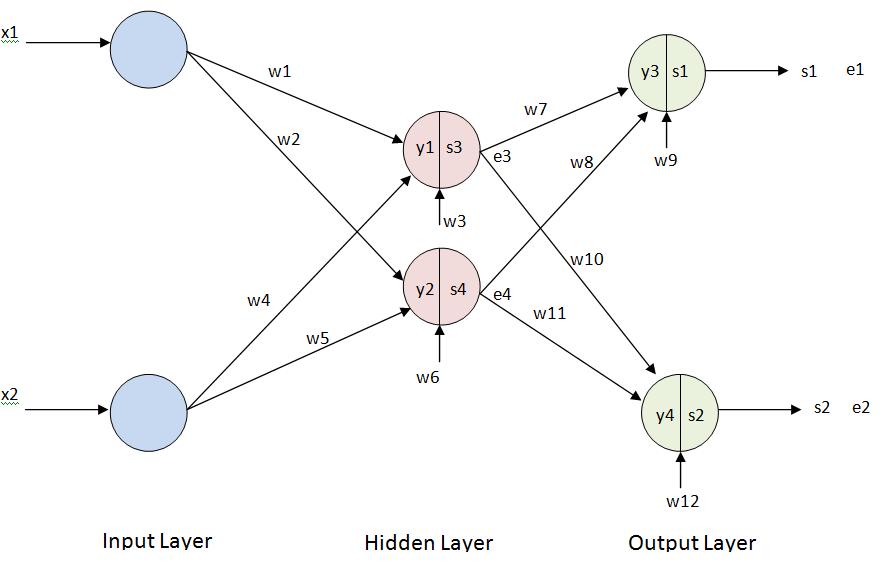
**7.**

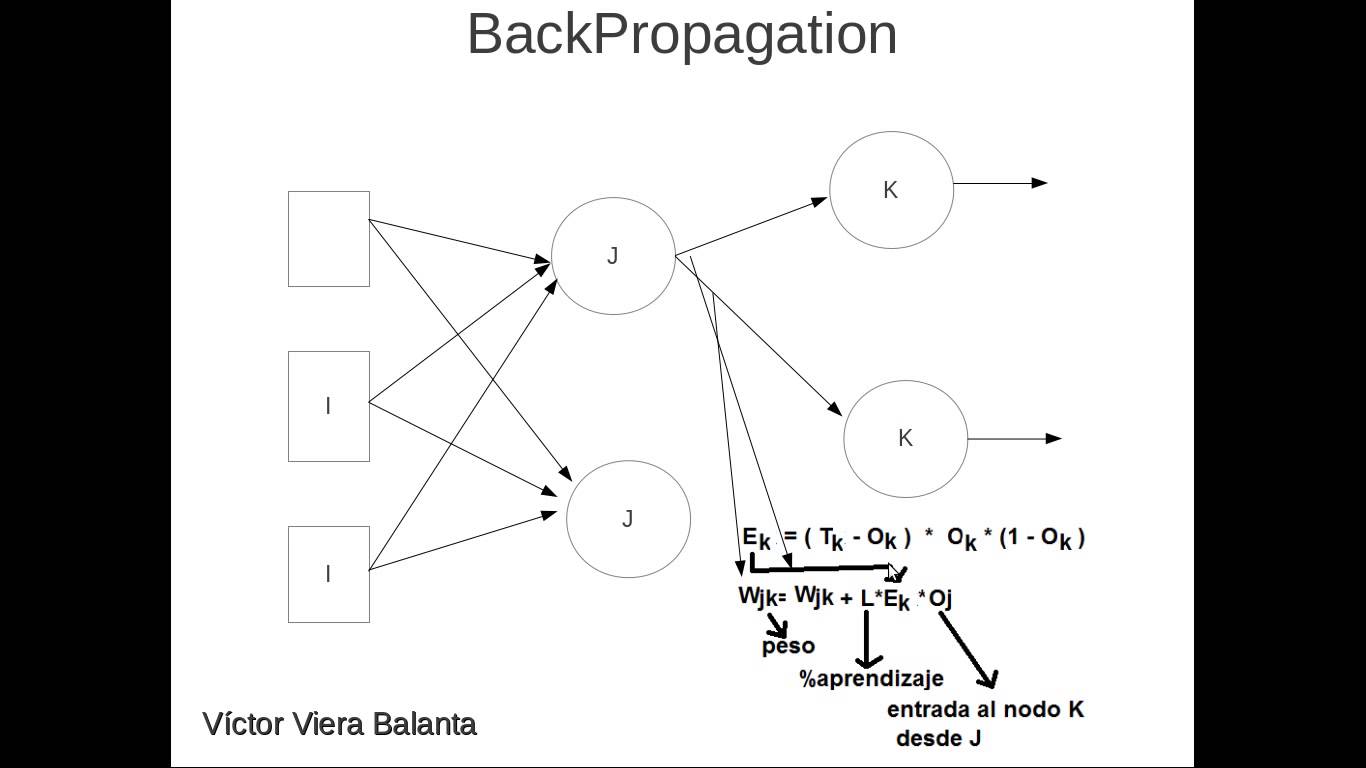
Al hablar de redes de retropropagación o redes de propagación hacia atrás hacemos referencia a un algoritmo de aprendizaje más que a una arquitectura determinada. La retropropagación consiste en propagar el error hacia atrás, es decir, de la capa de salida hacia la capa de entrada, pasando por las capas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

**8.**







**10.**

Teorema de Kolmogorov: “Dada cualquier función continua f(0,1)n --> Rm, existe una RNA de 3 capas, de propagación hacia adelante, con n elementos de proceso en la capa de entrada, m en la de salida y (2n+1) en la capa oculta; que implementa dicha función de forma exacta”

Sin embargo, el Teorema de Kolmogorov es un teorema de existencia. No hay ninguna técnica que nos diga cómo obtener la arquitectura de red para un determinado problema, ni mucho menos cuál es la arquitectura de red óptima para el problema.

A pesar de todos los éxitos del MLP y del algoritmo de BP, todavía quedan por resolver algunos problemas importantes:

El tiempo de computación necesario para el entrenamiento puede llegar a ser muy alto. Como hemos visto, es un problema asumible, pues una vez entrenada, la red puede congelar sus pesos y continuar funcionando.

Parálisis de red: Cuando los pesos alcanzan valores muy altos, la función sigmoide hace que la salida de la capa oculta sea muy cercana a 0 ó 1. De las ecuaciones del algoritmo de BP se deduce que el incremento de pesos en esos casos es prácticamente nulo, lo cual produce una parálisis en el entrenamiento. Una posible solución es añadir una pequeña cantidad de ruido a la salida de la neurona.

Mínimos locales: la función de error de una red compleja está llena de valles y picos, y la propia naturaleza del algoritmo puede producir la caída en uno de los valles, que no es necesariamente el mínimo. Una posible solución es incrementar el número de neuronas ocultas, pero como se ha visto, estos produce otros efectos secundarios indeseables.

BIBLIOGRAFIA

http://www.itnuevolaredo.edu.mx/takeyas/apuntes/Inteligencia%20Artificial/Apuntes/tareas\_alumnos/RNA/Redes%20Neuronales2.pdf

http://redes-neuronales.wikidot.com/definicion-ventajas-desventajas

https://gruporedesneuronales.wordpress.com/definiciones-basicas/

http://www.redes-neuronales.com.es/tutorial-redes-neuronales/funciones-de-las-neuronas-artificiales.htm

http://grupo.us.es/gtocoma/pid/pid10/RedesNeuronales.htm#entrenamiento

https://es.slideshare.net/mentelibre/red-neuronal-adaline

http://dianainteligenciaartificial.blogspot.com.co/2015/07/red-adaline.html

http://magomar.webs.upv.es/rna/tutorial/RNA\_backprop.html