**TALLER N° 3**

**KELLY FERNANDA VÁSQUEZ ZAPATA**

**JHONATTAN LEANDRO BEDOYA MEJIA**

**TRABAJO DE:**

**INTELIGENCIA ARTIFICIAL**

**PRESENTADO A:**

**CARLOS ALBERTO LONDOÑO LOAIZA**

**CARTAGO VALLE**

**CORPORACIÓN DE ESTUDIOS TECNOLÓGICOS DEL**

**NORTE DEL VALLE**

**TECNOLOGIA EN SISTEMAS DE INFORMACIÓN**

**SEMESTRE V**

**2018**

**CONTENIDO**

[HISTORIA DE REDES NEURONALES ARTIFICIALES 4](#_Toc507624523)

[Mapa conceptual 4](#_Toc507624524)

[VENTAJAS de las rna 5](#_Toc507624525)

[DESVENTAJAS DE LAS RNA 6](#_Toc507624526)

[aplicaciones de redes neuronales 7](#_Toc507624527)

[FUNCIÓN DE ACTIVACIÓN 8](#_Toc507624528)

[PERCEPTRÓN 10](#_Toc507624529)

[Historia 10](#_Toc507624530)

[Fórmula matematica 10](#_Toc507624531)

[Estructura de la red 11](#_Toc507624532)

[¿Pára qué se usa? 11](#_Toc507624533)

[Función de activación 12](#_Toc507624534)

[Entrenamiento de una red Perceptrón y algoritmo 12](#_Toc507624535)

[Evidencias 12](#_Toc507624536)

[ADALINE 14](#_Toc507624537)

[Historia 14](#_Toc507624538)

[Fórmula matemática 14](#_Toc507624539)

[Estructura de la red 15](#_Toc507624540)

[¿Pára qué se usa? 16](#_Toc507624541)

[Función de activación 16](#_Toc507624542)

[Entrenamiento de una red Adaline y algoritmo 17](#_Toc507624543)

[Evidencia 18](#_Toc507624544)

[REDES DE RETROPROPAGACIÓN (BACKPROPAGATION) 18](#_Toc507624545)

[¿Para qué se usa? 19](#_Toc507624546)

[algoritmo backpropagation 19](#_Toc507624547)

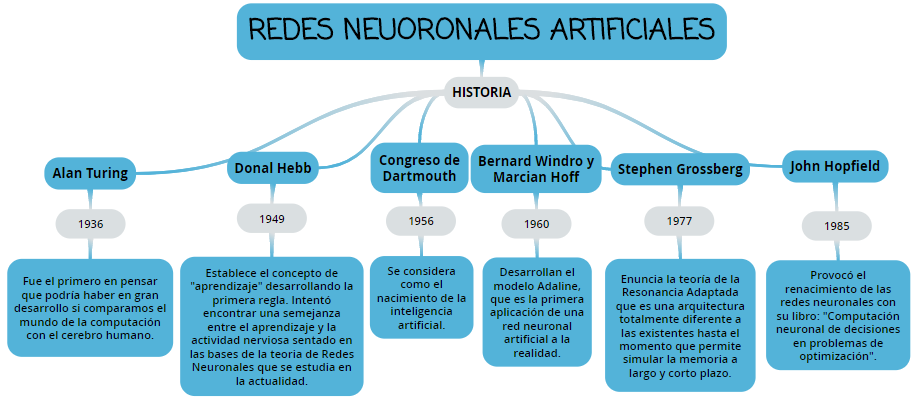
[TEOREMA DE KOLMOGOROV 21](#_Toc507624548)

[Bibliografía 22](#_Toc507624549)

# HISTORIA DE REDES NEURONALES ARTIFICIALES

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). (htt)

## Mapa conceptual



# VENTAJAS de las rna

1. **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.
2. **Auto organización**: Una RNA crea su propia representación de la información en su interior, descargando al usuario de esto.
3. **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.
4. **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)
5. **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.

La capacidad de aprendizaje adaptativo, es una de las características más atractivas de redes neuronales. Aprenden a llevar a cabo ciertas tareas mediante un entrenamiento con ejemplos ilustrativos. Como las redes neuronales pueden aprender a diferenciar patrones mediante ejemplos y entrenamientos, no es necesario elaborar modelos ni necesidad de especificar funciones de distribución de probabilidad.

1. Solucione de problemas no lineales son unos de los fuertes.

# DESVENTAJAS DE LAS RNA

1. Complejidad de aprendizaje para grandes tareas, cuanto más cosas se necesiten que aprenda una red, más complicado será enseñarle.
2. 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.
3. 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.
4. Elevada cantidad de datos para el entrenamiento, cuanto más flexible se requiera que sea la red neuronal, más información tendrá que enseñarle para que realice de forma adecuada la identificación.
5. Padecen nuestra falta de hardware. La capacidad de las redes neuronales radica en su habilidad de procesar información en paralelo (esto es, procesar múltiples pedazos de datos simultáneamente).
6. Las redes neuronales son muy malas para cálculos precisos, procesamiento serie, y no son capaces de reconocer nada que tenga inherentemente algún tipo de patrón.

# aplicaciones de redes neuronales

**1.** **Biología**:

* Aprender más acerca del cerebro.
* Obtención del modelo de la retina.

**2.** **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.

**3.** **Medio ambiente**:

* Analizar tendencias y patrones.
* Previsión del tiempo.

**4. Finanzas**:

* Previsión de la evolución de los precios.
* Valoración del riesgo de los créditos.
* Interpretación de firmas.
* Identificación de falsificaciones.

**5. Manufacturación**:

* Robots y sistemas de control (visión artificial y sensores de presión, temperatura, etc).
* Control de producción en líneas de proceso.

**6. 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.

**7. Militares**:

* Clasificación de las señales de radar.
* Creación de armas inteligentes.
* Reconocimiento y seguimiento en el tiro al blanco.
* De la clasificación anterior podemos deducir que la mayoría de ellas consisten en realizar un reconocimiento de patrones: búsqueda, clasificación, reconstrucción.

**8. Telecomunicaciones:**

* Comprensión de datos e imágenes.

**9. Transporte:**

* Diagnostico de frenos en camioneta. Seguimiento de flotas.

**10. Bancos:**

* Lectura de cheques y documentos.

# FUNCIÓN DE ACTIVACIÓN

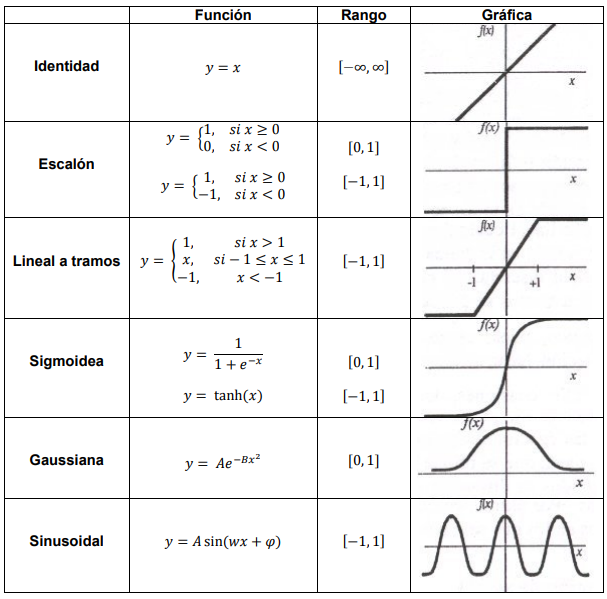
La función de activación determina el estado de activación actual de la neurona en base al potencial resultante 𝑛𝑒𝑡𝑖 y al estado de activación anterior de la neurona 𝑎𝑖(𝑡 −1). El estado de activación de la neurona para un determinado instante de tiempo t puede ser expresado de la siguiente manera:

𝑎𝑖(𝑡) = 𝑓(𝑎𝑖 (𝑡 − 1), 𝑛𝑒𝑡𝑖 (𝑡))

Sin embargo, en la mayoría de los modelos se suele ignorar el estado anterior de la neurona, definiéndose el estado de activación en función del potencial resultante ℎ𝑖:

𝑎𝑖(𝑡) = 𝑓( 𝑛𝑒𝑡𝑖(𝑡))

Algunas de las funciones de activación más utilizadas en los distintos modelos de redes neuronales son: (Barrera, n.d.)



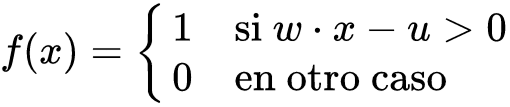
# PERCEPTRÓN

## Historia

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). (htt, p. 2)

## Fórmula matematica

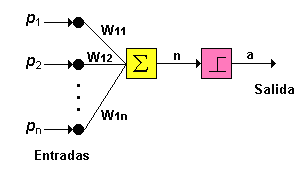
El perceptrón usa una matriz para representar las redes neuronales y es un discriminador terciario que traza su entrada **X** (un vector binario) a un único valor de salida **f(x)** (un solo valor binario) a través de dicha matriz.



Donde **W** es un vector de pesos reales y **W ⋅ X** es el producto escalar (que computa una suma ponderada). **U** es el ‘umbral’, el cual representa el grado de inhibición de la neurona, es un término constante que no depende del valor que tome la entrada.

El valor de **f(x)** (0 o 1) se usa para clasificar **X** como un caso positivo o un caso negativo, en el caso de un problema de clasificación binario. El umbral puede pensarse de como compensar la función de activación, o dando un nivel bajo de actividad a la neurona del rendimiento. La suma ponderada de las entradas debe producir un valor mayor que **U** para cambiar la neurona de estado 0 a 1. (Sinfallas, 2017)

## Estructura de la 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 la clase **A**, o -1 si el patrón pertenece a la clase **B**, la salida depende de la entrada neta (*n* = suma de las entradas *pi* ponderadas). (Quiero apuntes, n.d.)

## ¿Pára qué se usa?

De lo cual 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”.

## Función de activación

* **Función escalón:** Se asocia a neuronas binarias.
* **Función lineal y mixta:** Se asocia a neuronas con función mixta.
* **Función tangente hiperbólica:** Entrenamientos supervisados.
* **Función sigmoidal:** Similar a la función escalón, solo usa valores positivos.
* **Función de gauss:** Mapeos ocultos.

## Entrenamiento de una red Perceptrón y algoritmo

C:\Users\Kelly Vasquez\AppData\Local\Microsoft\Windows\INetCache\Content.Word\perc.pngEl Perceptrón es un tipo de red de aprendizaje supervisado, es decir necesita conocer los valores esperados para cada una de las entradas presentadas; su comportamiento está definido por pares de esta forma:

Cuando p es aplicado a la red, la salida de la red es comparada con el valor esperado t, y la salida de la red está determinada por:



Los valores de los pesos determinan el funcionamiento de la red, estos valores se pueden fijar o adoptar utilizando diferentes algoritmos de entrenamiento de la red.

## Evidencias

**Automotriz**

* Sistemas automáticos de navegación.
* Comando por voz.

**Telecomunicaciones**

* Compresión de datos e imágenes.
* Servicios automáticos de información.
* Traducción de lenguaje hablado en tiempo real.

**Electrónica**

* Predicción de secuencias de códigos.
* Control de procesos.
* Análisis de fallas de circuitos.
* Visión de máquina.
* Síntesis de voz.
* Modelado no lineal.

**Voz**

* Reconocimiento de voz.
* Compresión de voz.
* Sintetizadores de texto a voz.

**Seguridad**

* Reconocimiento de rostros.
* Identificación.
* Acceso de personas.

# ADALINE

## Historia

1960 – Bernard Windrow y 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.

## Fórmula matemática

Monografias.com Siendo:

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

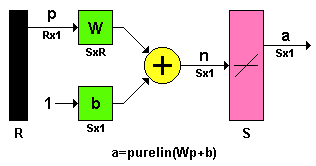
**P:** Número de informaciones que debe aprender la red.

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:

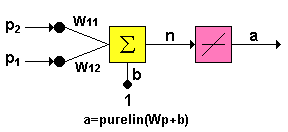
Monografias.comMediante este procedimiento, se llegan a obtener un conjunto de pesos con los que se consigue minimizar el error medio.

## Estructura de la red

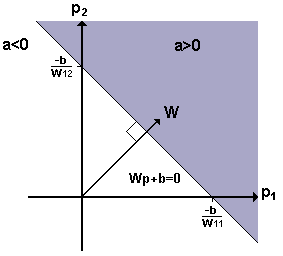
La estructura general de la red tipo Adaline:



Para una red Adaline de una sola neurona con dos entradas:



Específica la línea que separa en dos regiones el espacio de entrada:



La salida de la neurona es mayor que cero en el área gris, en el área blanca la salida es menor que cero. Como se mencionó anteriormente, la red Adaline puede clasificar correctamente patrones linealmente separables en dos categorías. (Quiero apuntes, n.d.)

## ¿Pára qué se usa?

Permite un aprendizaje rápido es propuesto por [Widrow 60], denominado regla *delta* o regla del mínimo error cuadrado (LMS Error: Least-Mean-Squared Error), que se aplicó en las redes desarrolladas por los mismos, conocidas como ADALINE y MADALINE).

[Widrow 60] definieron una función que permitiría cuantificar el error global cometido en cualquier momento durante el proceso de entrenamiento de la red, lo cual es importante, ya que cuanta más información se tenga sobre el error cometido, más rápido se puede aprender.

## Función de activación

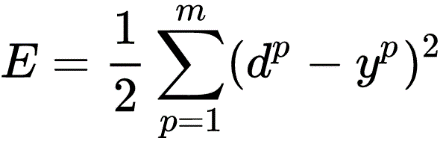
Aprendizaje estocástico. Este tipo de aprendizaje consiste básicamente en realizar cambios aleatorios en los valores de los pesos de las conexiones de la red y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidad.

En el aprendizaje estocástico se suele hacer una analogía en términos termodinámicos, asociando la red neuronal con un sólido físico que tiene cierto estado energético. En el de caso de la red, la energía de la misma representaría el grado de estabilidad de la red, de tal forma que el estado de mínima energía correspondería a una situación en la que los pesos de las conexiones consiguen que su funcionamiento sea el que más se ajusta al objetivo deseado.

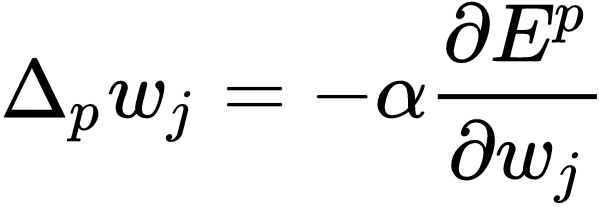
Según lo anterior, el aprendizaje consistiría en realizar un cambio aleatorio de los valores de los pesos y determinar la energía de la red. Si la energía es menor después del cambio; es decir, si el comportamiento de la red se acerca al deseado, se acepta el cambio; de lo contrario, se aceptaría el cambio en función de una determinada y preestablecida distribución de probabilidades.

## Entrenamiento de una red Adaline y algoritmo

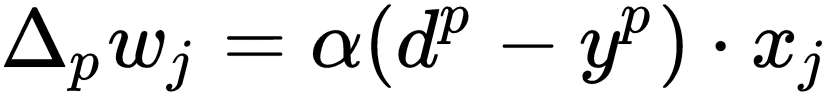
A diferencia del perceptrón, a la hora de modificar los pesos durante el entrenamiento, el Adaline tiene en cuenta el grado de corrección de la salida estimada respecto a la deseada.​ Esto se consigue mediante la aplicación de la regla Delta, y que se define, para un patrón de entrada *xp* con una salida estimada *y p* y una salida deseada *d p*, como |*d p* – *y p*|.

Dado que el objetivo del Adaline es poder estimar de la manera más exacta la salida (conseguir una salida exacta es prácticamente imposible en la mayoría de los casos), se busca minimizar la desviación de la red para todos los patrones de entrada, eligiendo una medida del error global. Normalmente se utiliza el error cuadrático medio.

La manera de reducir este error global es ir modificando los valores de los pesos al procesar cada entrada, de forma iterativa, mediante la regla del descenso del gradiente. Suponiendo que tenemos una constante de aprendizaje *α*:



Si operamos con la derivada, queda:



Que será la expresión que utilizaremos por cada entrada para modificar los pesos.

## Evidencia

* **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.

# 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. (Magomar, n.d.)

## ¿Para qué se usa?

El funcionamiento de la red backpropagartion (BPN) consiste en el aprendizaje de un conjunto predefinido de pares de entradas-salidas dados como ejemplo: primero se aplica un patrón de entrada como estímulo para la primera capa de las neuronas de la red, se va propagando a través de todas las capas superiores hasta generar una salida, se compara el resultado en las neuronas de salida con la salida que se desea obtener y se calcula un valor de error para cada neurona de salida. A continuación, estos errores se transmiten hacia atrás, partiendo de la capa de salida hacia todas las neuronas de la capa intermedia que contribuyan directamente a la salida, recibiendo de error aproximado a la neurona intermedia a la salida original. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido un error que describa su aportación relativa al error total. Basándose en el valor del error recibido, se reajustan los pesos de conexión de cada neurona, de manera que en la siguiente vez que se presente el mismo patrón, la más salida esté cercana a la deseada.

# algoritmo backpropagation

**Paso 1**: Inicializar los pesos de la red con valores pequeños aleatorios.

**Paso 2**: Presentar un patrón de entrada y especificar la salida deseada que debe generar la red.

**Paso 3**: Calcular la salida actual de la red. Para ello presentamos las entradas a la red y vamos calculando la salida que presenta cada capa hasta llegar a la capa de salida, ésta será la salida de la red. Los pasos son los siguientes:

Monografias.comSe calculan las entradas netas para las neuronas ocultas procedentes de las neuronas de entrada. Para una neurona *j* oculta:

En donde el índice *h* se refiere a magnitudes de la capa oculta; el subíndice *p*, al *pésimo* vector de entrenamiento, y *j* a la *jésima* neurona oculta. El término Monografias.com puede ser opcional, pues actúa como una entrada más.

Monografias.comSe calculan las salidas de las neuronas ocultas: Monografias.com Se realizan los mismos cálculos para obtener las salidas de las neuronas de salida:

Monografias.com **Paso 4:** Calcular los términos de error para todas las neuronas.

Monografias.comSi la neurona *k* es una *neurona de la capa de salida*, el valor de la *delta* es:

La función *f* debe ser derivable.

En general disponemos de dos formas de función de salida:

La función lineal: Monografias.com

La función sigmoidal: Monografias.com

La selección de la función depende de la forma que se decida representar la salida: si se desea que las neuronas de salida sean binarias, se utiliza la función sigmoidal, en otros casos, la lineal.

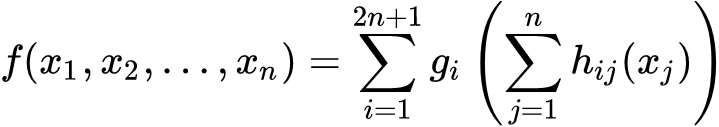
Monografias.comPara una función lineal, tenemos: Monografias.com mientras que la derivada de una función sigmoidal es:

Por lo que los términos de error para las neuronas de salida quedan:

Monografias.com Monografias.com para la salida lineal.

# TEOREMA DE KOLMOGOROV

Es un notable teorema de Andréi Kolmogórov que establece que una función continua de diversas variables se puede presentar como una superposición de funciones de una única variable. Más concretamente establece que una función continua de *n* variables definida sobre el conjunto [0,1]*n* siempre puede expresarse mediante sumas y 2(*n*+1)(*n*+1) funciones de una sola variable en la forma:



Donde las 2*n*+1 funciones *gi* son continuas, de una sola variable y dependen de *f*. Y (htt1)las 2(*n*+1)*n* funciones *hij* son universales, es decir, no dependen de *f*. (2014)

# Bibliografía

(s.f.). Obtenido de http://www.itnuevolaredo.edu.mx/takeyas/apuntes/Inteligencia%20Artificial/Apuntes/tareas\_alumnos/RNA/Redes%20Neuronales2.pdf

(s.f.). Obtenido de http://www.monografias.com/trabajos12/redneuro/redneuro2.shtml#ixzz58NEvKN3m

(21 de abril de 2014). Obtenido de https://es.wikipedia.org/wiki/Teorema\_de\_superposici%C3%B3n\_de\_Kolmog%C3%B3rov

Barrera, J. A.-T. (s.f.). *Cucei*. Obtenido de http://www.cucei.udg.mx/sites/default/files/pdf/toral\_barrera\_jamie\_areli.pdf

*Magomar*. (s.f.). Obtenido de http://magomar.webs.upv.es/rna/tutorial/RNA\_backprop.html

*Quiero apuntes*. (s.f.). Obtenido de http://www.quieroapuntes.com/redes-neuronales-artificiales\_1.html

*Sinfallas*. (14 de noviembre de 2017). Obtenido de https://sinfallas.wordpress.com/2017/11/14/el-perceptron-y-perceptron-multicapa-que-es-y-con-que-se-come/