TALLER N° 2 REDES NEURONALES ARTIFICIALES (RNA)

MAURICIO GRANADA QUINTERO

CARLOS ALBERTO LONDOÑO LOAIZA

INTELIGENCIA ARTIFICIAL

CORPORACION DE ESTUDIOS TECNOLOGICOS DEL NORTE DEL VALLE

COTECNOVA

CARTAGO VALLE 2016

Contenido

[1. Mapa Conceptual 4](#_Toc460880972)

[2. Ventajas y desventajas de las redes neuronales 5](#_Toc460880973)

[Ventajas: 5](#_Toc460880974)

[1. Aprendizaje 5](#_Toc460880975)

[2. Auto organización 5](#_Toc460880976)

[4. Flexibilidad: 5](#_Toc460880977)

[5. Tiempo real 5](#_Toc460880978)

[ Desventajas 5](#_Toc460880979)

[3. Aplicación de las redes neuronales 6](#_Toc460880980)

[ Biología 6](#_Toc460880981)

[ Empresa 6](#_Toc460880982)

[ Medio Ambiente 6](#_Toc460880983)

[ Finanzas 6](#_Toc460880984)

[ Manufacturación 6](#_Toc460880985)

[ Medicina 6](#_Toc460880986)

[ Militares 7](#_Toc460880987)

[5. Perceptron 7](#_Toc460880988)

[a. Historia 7](#_Toc460880989)

[b. Formula 8](#_Toc460880990)

[c. Estructura del Perceptron 8](#_Toc460880991)

[d. ¿Para qué se usa? 8](#_Toc460880992)

[e. ¿Cuál es la función de activación? 8](#_Toc460880993)

[f. Como se entrena un Perceptron 9](#_Toc460880994)

[ALGORITMO DE ENTRENAMIENTO DEL PERCEPTRON 9](#_Toc460880995)

[6. ADALINE 9](#_Toc460880996)

[a. Historia 9](#_Toc460880997)

[b. Formula 10](#_Toc460880998)

[c. Estructura 10](#_Toc460880999)

[d. Para que se usa 10](#_Toc460881000)

[e. Función de Activación 11](#_Toc460881001)

[f. Entrenamiento de la red Adaline 11](#_Toc460881002)

[Algoritmo: 11](#_Toc460881003)

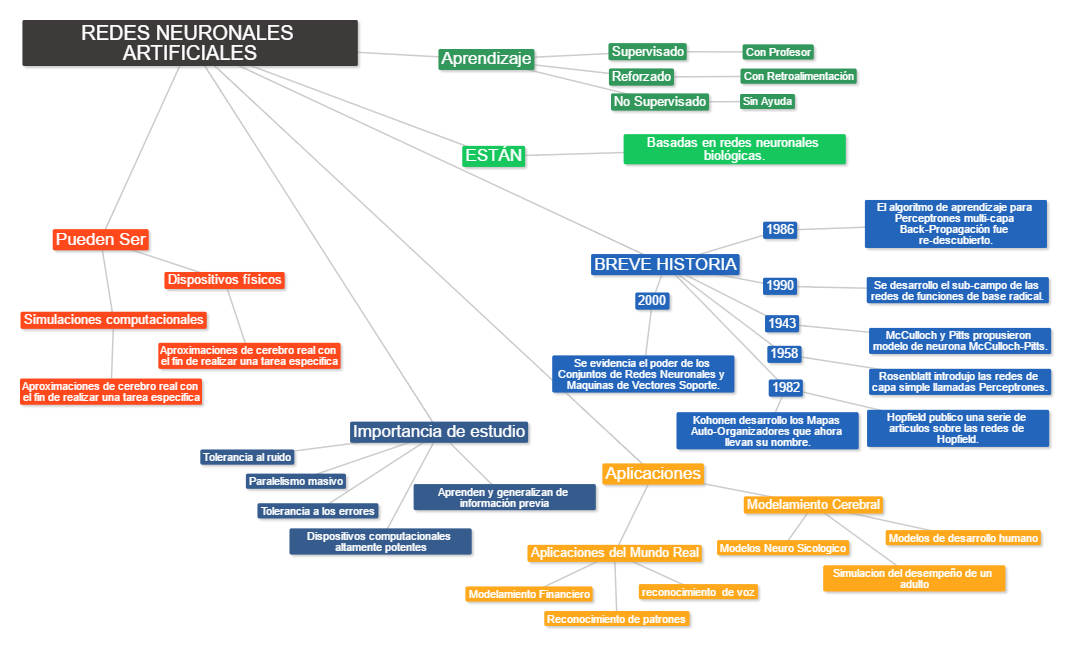
[7. Redes de retropropagación (backpropagation) 11](#_Toc460881004)

[8. Explicación del algoritmo backpropagation, usando imágenes: 12](#_Toc460881005)

[10. Teorema de kolmogorov: 17](#_Toc460881006)

[BIBLIOGRAFIA 18](#_Toc460881007)

# Mapa Conceptual



# Ventajas y desventajas de las redes neuronales

## Ventajas:

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.
6. **Reconocimiento:**

Son excelentes como clasificadores/reconocedores de patrones – y pueden ser usadas donde las técnicas tradicionales no funcionan.

1. **Plausibles:** Lo que significa que pueden proveer pistas de cómo trabaja el cerebro según progresen.

* Desventajas**:**

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. La mayoría de ellas, de todas maneras, padecen nuestra falta de hardware.
5. • 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.
6. Otros problemas con las redes neuronales son la falta de reglas definitorias que ayuden a realizar una red para un problema dado.
7. Aplicación de las redes neuronales**:**

* Biología:

- Aprender más acerca del cerebro y otros sistemas.

- Obtención de modelos de la retina.

* Empresa:

- Evaluación de probabilidad de formaciones geológicas y petrolíferas.

- Identificación de candidatos para posiciones específicas.

- Explotación de bases de datos.

- Optimización de plazas y horarios en líneas de vuelo.

* Medio Ambiente:

- Analizar tendencias y patrones.

- Previsión del tiempo.

* Finanzas:

- Previsión de la evolución de los precios.

- Valoración del riesgo de los créditos.

- Identificación de falsificaciones.

- Interpretación de firmas.

* Manufacturación:

- Robots automatizados y sistemas de control (visión artificial y sensores de presión, temperatura, gas, etc.).

- Control de producción en líneas de procesos.

- Inspección de la calidad.

* Medicina:

- Analizadores del habla para ayudar en la audición de sordos profundos. - Diagnóstico y tratamiento a partir de síntomas y/o de datos analíticos (electrocardiograma, encefalogramas, análisis sanguíneo, etc.).

- Monitorización en cirugías.

- Predicción de reacciones adversas en los medicamentos.

- Entendimiento de la causa de los ataques cardíacos.

* Militares:

- Clasificación de las señales de radar.

- Creación de armas inteligentes. - Optimización del uso de recursos escasos.

- Reconocimiento y seguimiento en el tiro al blanco.

1. **FUNCION DE ACTIVACIÓN:** calcula la activación de la unidad en función de la entrada total y la activación previa, aunque en la mayor parte de los casos es simplemente una función no decreciente de la entrada total.

Los tipos que existen de esta función son: la función escalón, la función lineal, la función sigmoidal binario y bipolar, la función tangente hiperbólica y la función gaussiana

Se aplican para las redes neuronales

1. Perceptron**:**

## Historia

En 1943, Warren McCulloch y Walter Pitts introdujeron una de las primeras neuronas artificiales. La característica principal de su modelo de neurona es que una 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.

El Perceptron es limitado. Dichas limitaciones fueron publicadas en el libro Perceptron 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 Perceptron mejoradas (multicapa) asociadas con reglas de aprendizaje.

## Formula

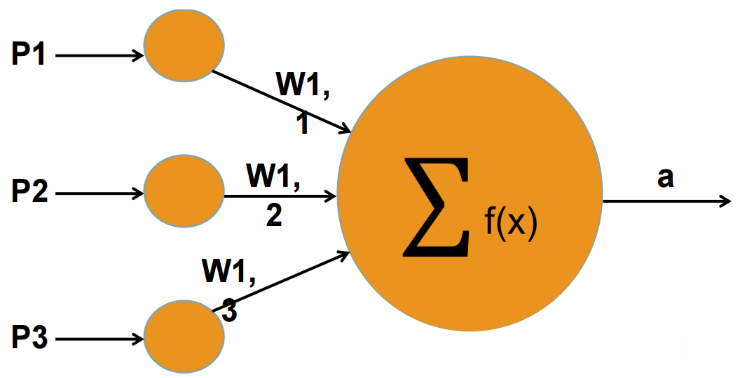
El Perceptron usa una matriz para representar las redes neuronales y es un discriminador terciario que traza su entrada x (un vector binario) a un único valor de salida f(x) (un solo valor binario) a través de dicha matriz.



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.

## Estructura del Perceptron



## ¿Para qué se usa?

El perceptrón simple sólo sirve para clasificar problemas linealmente separables, cosa que ya se podía hacer mediante métodos estadísticos, y de una forma mucho más eficiente.

## ¿Cuál es la función de activación?

Función activación de las neuronas de la capa de salida es de tipo escalón dando de esta manera solo salidas binarias (0 y 1).

## Como se entrena un Perceptron

El entrenamiento de un perceptrón es por medio de la regla de aprendizaje delta:

Para cada peso W se realiza un ajuste dW según la regla:

dW = LR ( T - Y ) X

Donde LR es la razón de aprendizaje, T el valor deseado, Y el valor obtenido, y X la entrada aplicada al perceptrón.

### ALGORITMO DE ENTRENAMIENTO DEL PERCEPTRON

Iniciar: w0 aleatoriamente; t ĸ 0

Repetir Mientras quede vectores mal clasificados

Seleccionar aleatoriamente un vector x

si x ȫ P (salida esperada 1) y wt • x < 0 (salida calculada 0) entonces

[Hay un error]

wt+1 ĸ wt + x ; t ĸ t+1

fin si

si x ȫ N (salida esperada 0) y wt • x  0 (salida calculada 1) entonces

[Hay un error]

wt+1 ĸ wt – x ; t ĸ t+1

fin si

fin repite.

# ADALINE

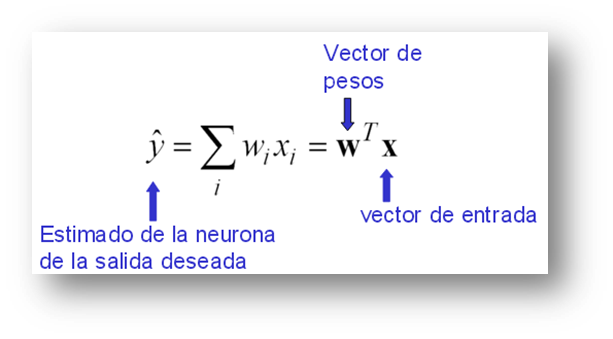
## Historia

La Red Adaline fue desarrollada en el 1960 por Bernard Widrow y su estudiante Marcian Hoff de la universidad de Stanford.

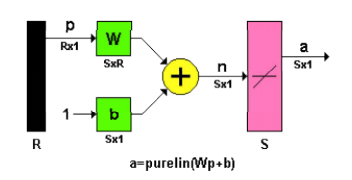
ADALINE proviene de Adaptive Lineal Element (Elemento Lineal Adaptativo), pero antes de que se le diera este nombre esta red sufrió un cambio ya que primeramente se llamaba Adaptive Lineal Neuron (Neurona Linear Adaptiva), dicho cambio se dio por que la Red Adaline es un dispositivo que consta de un único elemento de procesamiento, como tal no es técnicamente considerada una red neuronal.

Adaline fue desarrollada para el reconocimiento de patrones binarios, por ejemplo predecir el siguiente bit en una línea telefónica.

## Formula



## Estructura



## Para que se usa

La principal aplicación de las redes de tipo Adaline se encuentra en el campo de procesamiento de señales. Específicamente en el diseño de filtros, estas son capaces de eliminar ruido en señales portadoras de información.

Otra aplicación que tiene esta red es la de los filtros adaptivos.

Un Adaline tiene la suficiente capacidad 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).

Predecir el valor futuro de una señal a partir de su valor actual.

El error predicción va a variar, es decir la predicción va a ser mayor o menor dependiendo de la señal que se va a predecir. Si dicha señal corresponde a una serie de temporal el Adaline, transcurrido un tiempo, estará apta para dar predicciones correctas.

## Función de Activación



## Entrenamiento de la red Adaline

El entrenamiento de la red consiste en adaptar los pesos a medida que se vayan presentando los patrones de entrenamiento y salidas deseadas para cada uno de ellos.

Para cada combinación E/S se realiza un proceso automático de pequeños ajustes en los valores de los pesos hasta que se obtienen las salidas correctas.

### Algoritmo:

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.

# Redes de retropropagación (backpropagation)

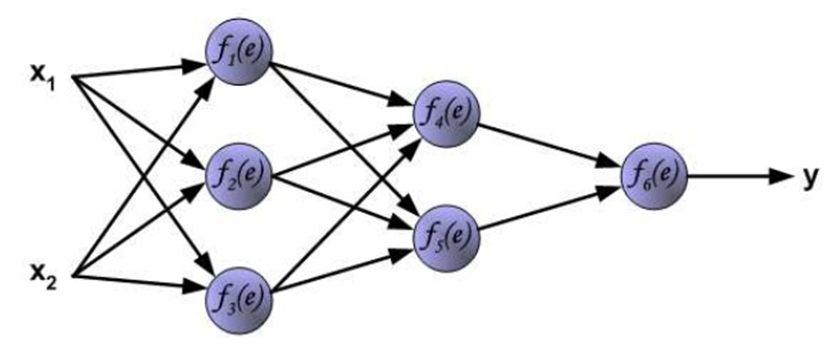
REDES DE RETROPROPAGACION: Fue creada en 1986 por Rumelhart, Hinton y Williams. El trabajo de la red consistía en aprender la asociación que existe entre los patrones de entrada y las clases correspondientes. Este método está basado en la generalización de la regla delta.

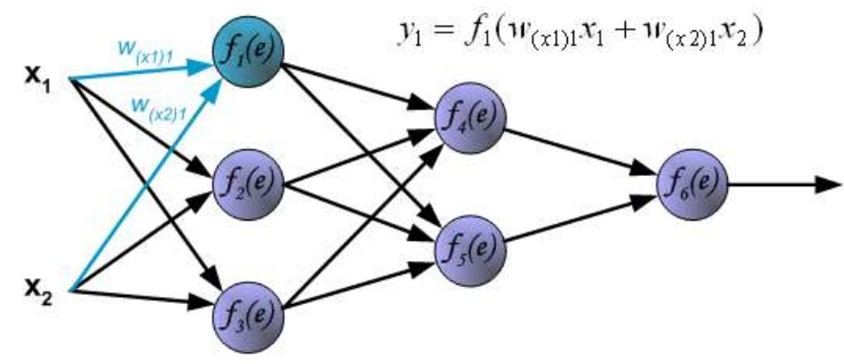
El algoritmo backpropagation (propagación del error hacia atrás) es una regla de aprendizaje que se puede aplicar en modelos de redes con más de dos capas de neuronas. Una característica importante de este algoritmo es la representación interna del conocimiento que es capaz de organizar en la capa intermedia de las células para conseguir cualquier correspondencia entre la entrada y la salida de la red.

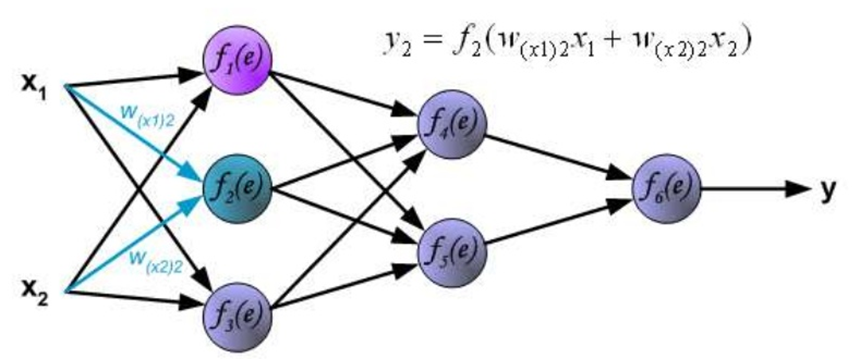
La importancia de esta red consiste en su capacidad de autoadaptar los pesos de las neuronas de las capas intermedias para aprender la relación que existe entre un conjunto de patrones dados como ejemplo y sus salidas correspondientes, para poder aplicar esa misma relación, después del entrenamiento, a nuevos vectores de entrada con ruido o incompletas, dando una salida activa si la nueva entrada es parecida a las presentadas durante el aprendizaje. Esta característica, exigida en los sistemas de aprendizaje, es la capacidad de generalización, entendida como la facilidad de dar salidas satisfactorias a entradas que el sistema no ha visto nunca en su fase de entrenamiento.

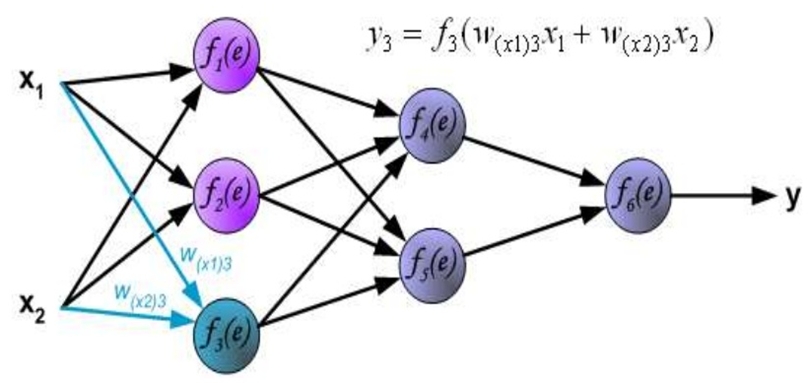
# Explicación del algoritmo backpropagation, usando imágenes:

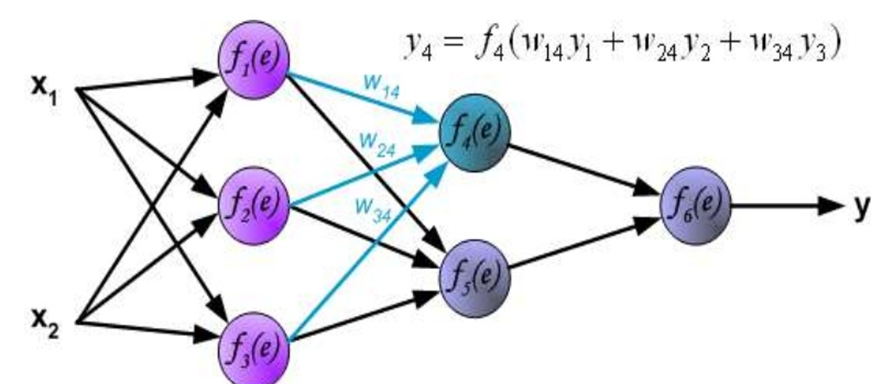
El proceso de enseñanza del algoritmo de aprendizaje para redes neuronales de retro propagación de múltiples capas. Para ilustrar este proceso, la red neural que utilizaremos consiste de tres capas con dos entradas y una salida como se muestra en la siguiente arquitectura:

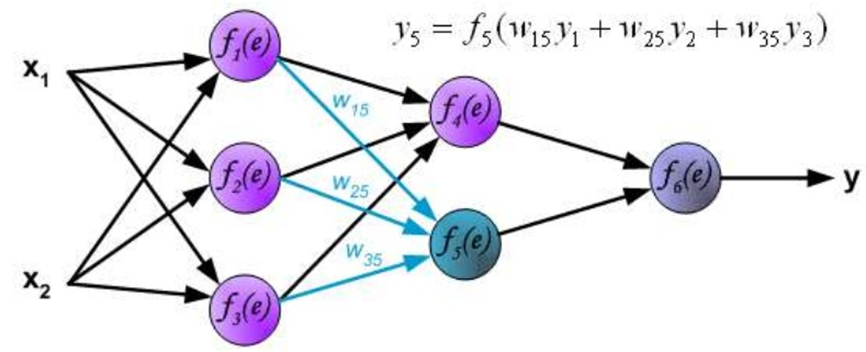
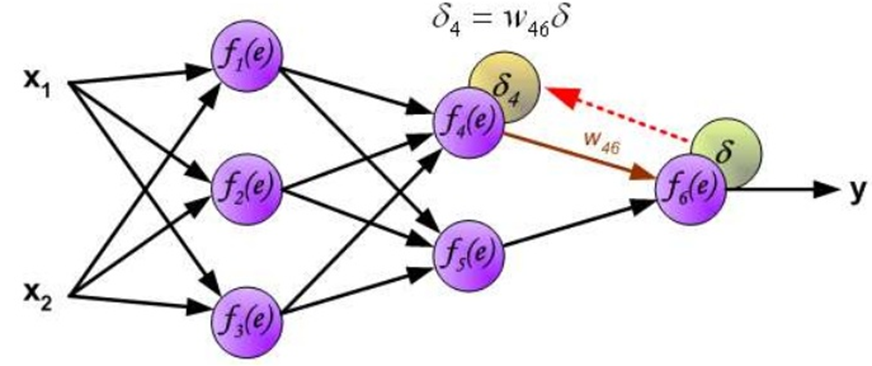
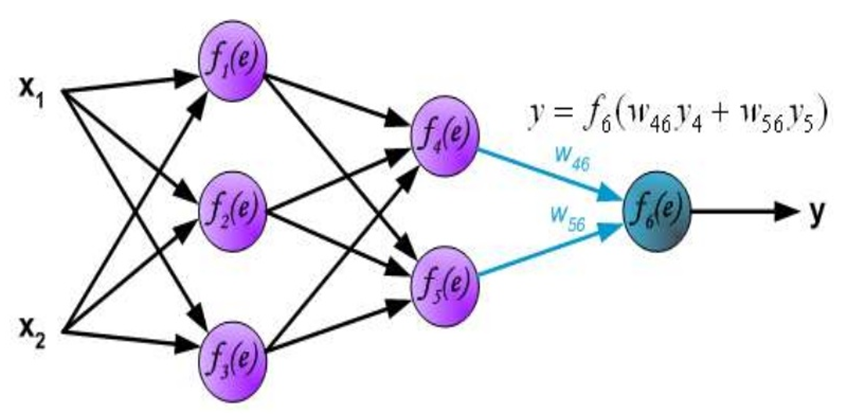


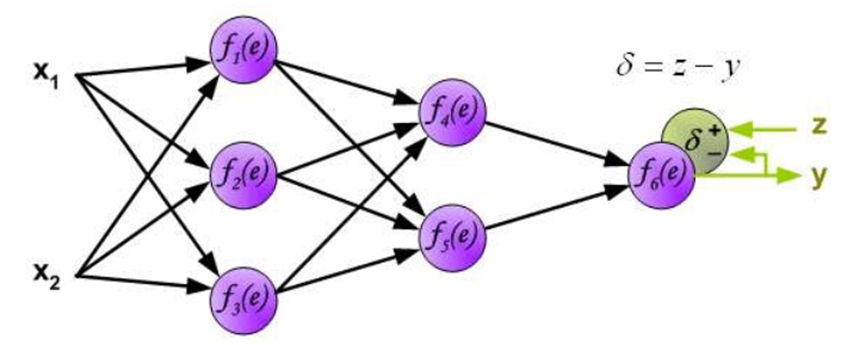


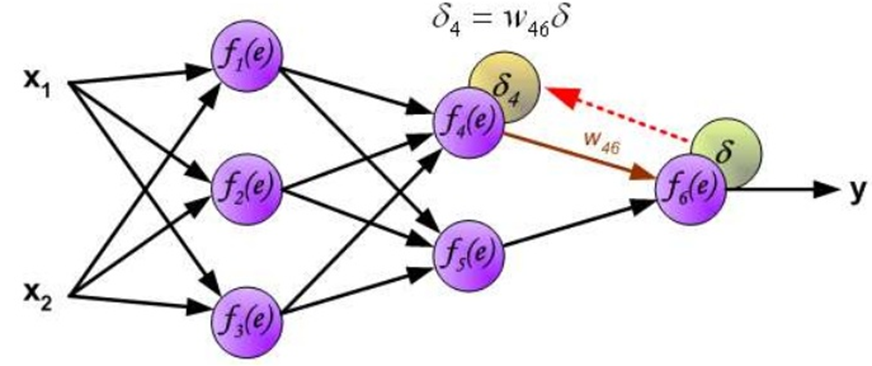


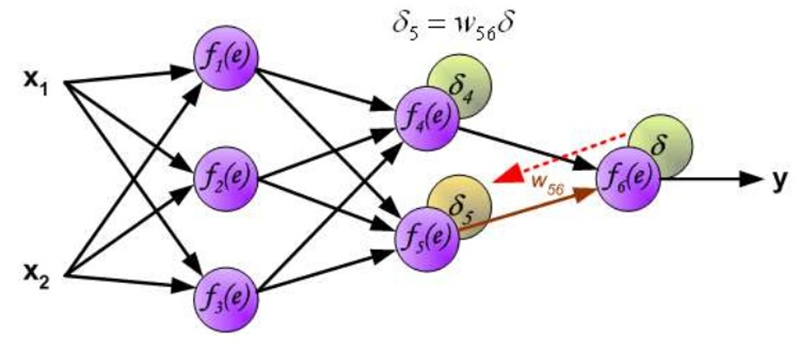


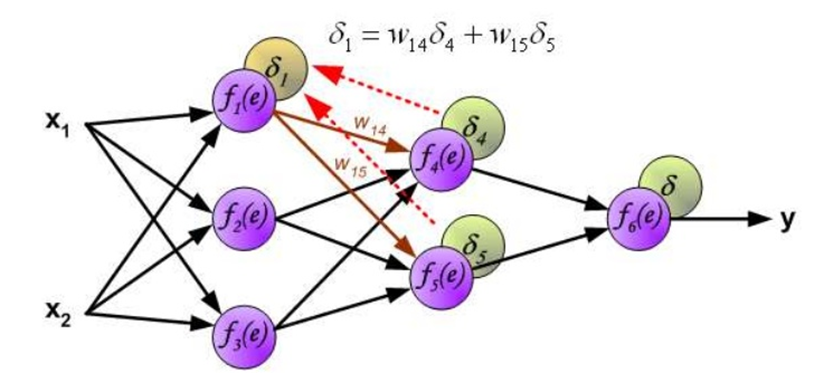


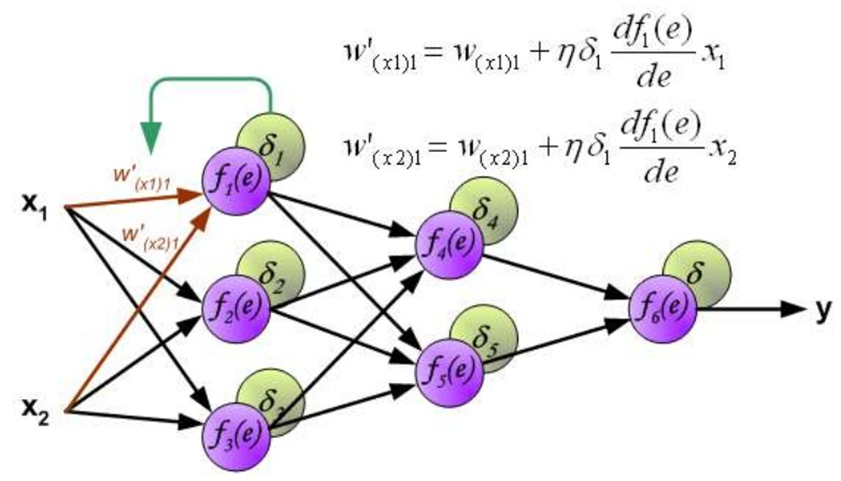


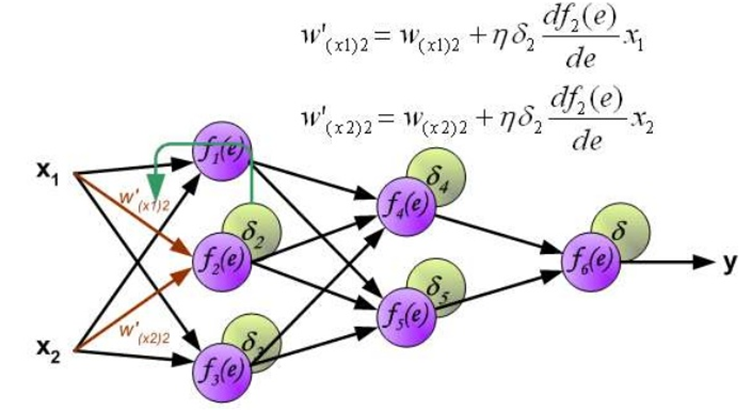


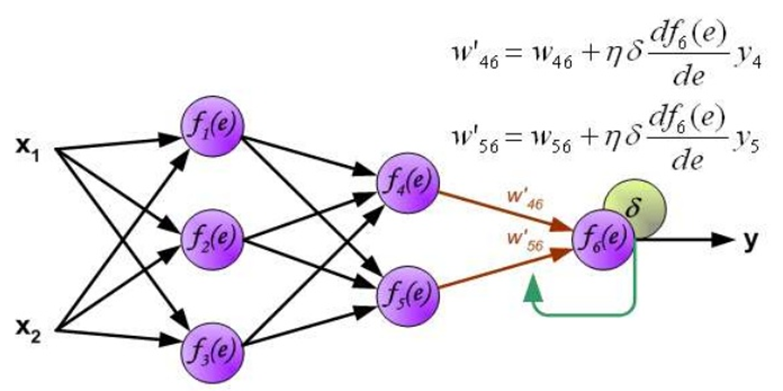












# Teorema de kolmogorov:

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 producen otros efectos secundarios indeseables.

# BIBLIOGRAFIA

http://juanitajuanabernal.blogspot.com.co/

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

<https://www.frro.utn.edu.ar/repositorio/catedras/quimica/5_anio/orientadora1/monograias/matich-redesneuronales.pdf>

http://dianainteligenciaartificial.blogspot.com.co/

ftp://decsai.ugr.es