

Taller 3
Inteligencia Artificial

Docente
Carlos Londoño
Ingeniero de Sistemas

Estudiante
Elkin Ramírez Jiménez
Stefany Lorena Sánchez

Corporación de estudios Tecnológicos del norte del valle

- 1) Consultar la historia de las redes neuronales artificiales, crear un mapa conceptual, que permita evidenciar los casos más importantes.

Redes Neuronales.

Una red neuronal es un sistema compuesto de muchos elementos procesadores simples operando por una estructura de la red.

Las redes neuronales no son más que un modelo artificial y simplificado del cerebro humano.

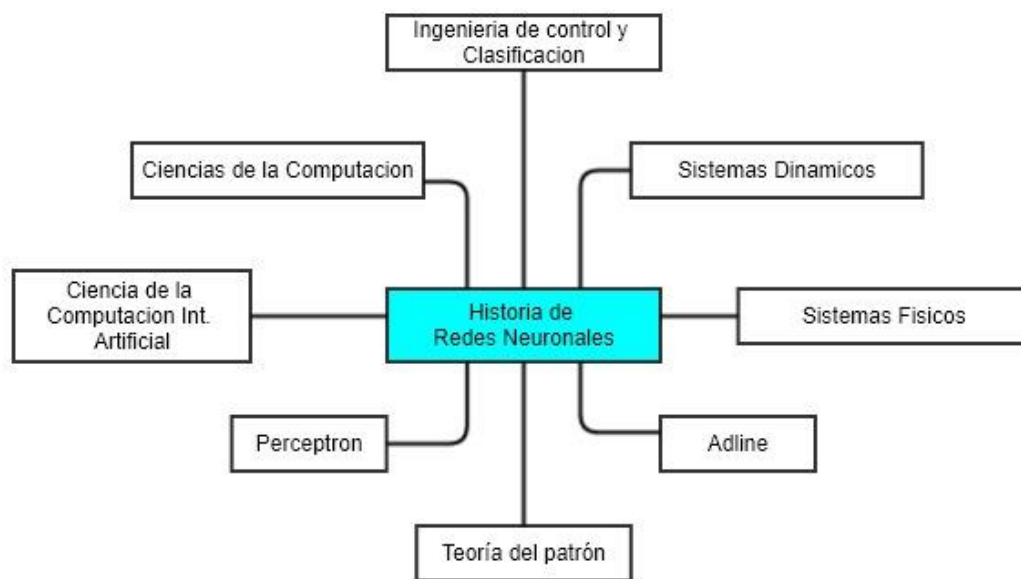


Ilustración 1 Redes Neuronales Historia

2)

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

Ventajas

- * La solución de problemas no lineales es uno de su fuerte
- * Las RNA son robustas, pueden fallar algunos elementos del procedimiento pero la red continuara trabajando.
- * Se pueden sintetizar algoritmos a través de un proceso de aprendizaje
- * Los procesos que mejor resuelve son los mismos que el ser humano resuelve mejor como lo son: asociación, evaluación y reconocimiento de patrones

* El auto organización, una RNA crea su propia representación de la información en su interior.

RNA puede manejar cambios no importantes en la información de entrada, como señales con ruido u otros cambios en la entrada

Desventajas

* La complejidad de aprendizaje para grandes tareas, cuantas más cosas se necesiten que aprenda una red, más complicado será enseñarle

* Tiempo de aprendizaje elevado. Esto depende de dos factores: primero 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 quiere enseñar.

* No se 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.

* eleva cantidad de datos para el entretenimiento, cuanto más flexible se requiera que sea la red neuronal, más información tendrá que enseñarle para que realice de forma adecuada la identificación.

* También hay problemas con las redes neuronales por falta de reglas definitorias que ayuden a realizar una red para un problema dado.

Nombra 10 aplicaciones de las redes neuronales.

La aplicación de las redes neuronales es muy amplia, aquí veremos una serie de campos donde podemos aplicar las redes neuronales.

El uso de las redes neuronales artificiales en la empresa

En el mundo de los negocios la aportación de las redes neuronales artificiales, ya sea a nivel estratégico, táctico u operativo, es extremadamente valiosa, precisamente por su importante contribución a la analítica predictiva que se pone de manifiesto:

- En la construcción de modelos explicativos, que ayudan a explorar conjuntos de datos en busca de variables o grupos de variables pertinentes.
- En su capacidad para aproximar cualquier función continua, haciendo posible que el analista no necesite tener ninguna hipótesis sobre el modelo subyacente.

- El valor clave de las redes neuronales artificiales es su capacidad para modelar funciones no lineales muy complejas, aunque también se adaptan bien a problemas altamente dimensionales, donde el número potencial de predictores es muy grande.
- Su principal debilidad reside en su tendencia al sobre-aprendizaje, ya que este tipo de redes aprenden a minimizar el error de predicción en los datos con los que son instruidas, que no es lo mismo que reducir al mínimo el error de predicción en una aplicación de negocio. Por eso mismo, al igual que con otras técnicas de modelado, los analistas deben probar los modelos producidos con redes neuronales artificiales en una muestra independiente y tomar las decisiones adecuadas en cuanto a la topología de red, funciones de transferencia, funciones de activación y el algoritmo de entrenamiento.

Generales

- Biología
- Empresas
- Medio ambiente
- Manufactura
- Finanzas

Específicas

- Reconocimiento de patrones
- Toma de decisiones
- Robótica
- Domótica
- Reconocimiento facial

- 4) ¿Qué son funciones de activación, cuales existen y para cuales redes neuronales se aplican?

Función de activación (activation function). 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, Θ_i) 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 señal *Net* (salida) generalmente se procesa por medio de una función de activación *F*, la cual producirá una señal que será la salida (*Out*) de la neurona. La función *F* también puede ser alguna otra función que simule mejor las características no lineales de transferencia de una neurona biológica. Si *F* reduce el rango de *Net* de manera que *Out* nunca salga de algún límite, independientemente de lo grande que sea *Net*, entonces *F* es una función *sigmoideal*, una función de este tipo es la función logística.

5) Que es un perceptron

a. Historia

Historia del Perceptrón

La evolución de la neurona artificial ha progresado a través de varias etapas. Las raíces de las cuales, están cimentadas dentro trabajo neurológica realiza principalmente por Santiago Ramón y Cajal y Sir Charles Scott Sherrington . Ramón y Cajal fue una figura prominente en la exploración de la estructura de tejido nervioso y demostró que, a pesar de su capacidad para comunicarse entre sí, las neuronas se separan físicamente de otras neuronas. Con una mayor comprensión de los elementos básicos del cerebro, se hicieron esfuerzos para describir cómo estas neuronas básicas podrían dar lugar a conductas manifiestas, a la que William James fue un destacado colaborador teórico.

Trabajando a partir de los comienzos de la neurociencia, Warren McCulloch y Walter Pitts en su artículo de 1943, "Un cálculo lógico de las ideas inmanentes en la actividad nerviosa", sostuvieron que las neuronas con una función de activación de umbral binario eran análogas a las oraciones lógica de primer orden. La McCulloch básica y Pitts neurona se veía algo como lo siguiente

Definición

El modelo biológico más simple de un perceptrón es una neurona y viceversa. Es decir, el modelo matemático más simple de una neurona es un perceptrón. La neurona es una célula especializada y caracterizada por poseer una cantidad indefinida de canales de entrada llamados dendritas y un canal de salida llamado axón. Las dendritas operan como sensores que recogen información de la región donde se hallan y la derivan hacia el cuerpo de la neurona que reacciona mediante una sinapsis que envía una respuesta hacia el cerebro, esto en el caso de los seres vivos.

Una neurona sola y aislada carece de razón de ser. Su labor especializada se torna valiosa en la medida en que se asocia a otras neuronas, formando una red. Normalmente, el axón de una neurona entrega su información como "señal de entrada" a una dendrita de otra neurona y así sucesivamente. El perceptrón que capta la señal en adelante se extiende formando una red de neuronas, sean éstas biológicas o de sustrato semiconductor (compuertas lógicas).

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

b. Fórmula matemática, explique sus términos

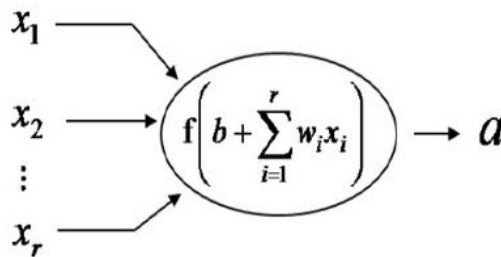


Ilustración 2 Fórmula Matemática de un Perceptron

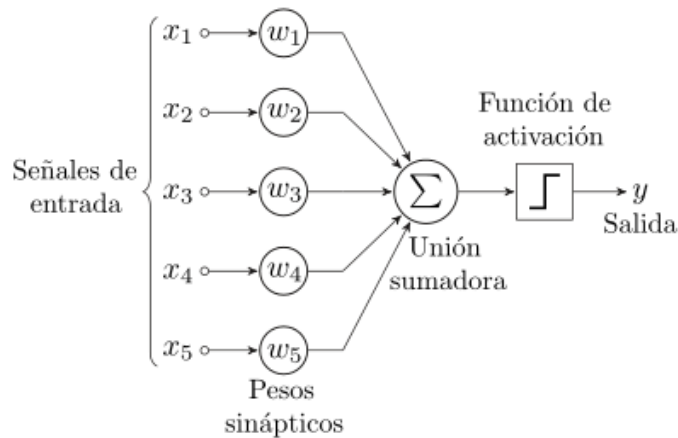
En el perceptrón, existen dos tipos de aprendizaje, el primero utiliza una tasa de aprendizaje mientras que el segundo no la utiliza. Esta tasa de aprendizaje amortigua el cambio de los valores de los pesos.¹

El algoritmo de aprendizaje es el mismo para todas las neuronas, todo lo que sigue se aplica a una sola neurona en el aislamiento. Se definen algunas variables primero:

- el $x(j)$ denota el elemento en la posición j en el vector de la entrada
- el $w(j)$ el elemento en la posición j en el vector de peso
- el y denota la salida de la neurona
- el δ denota la salida esperada
- el α es una constante tal que $0 < \alpha < 1$

Ilustración 3 Convenciones De Valor Perceptron

c. Dibuje la estructura



d. Ilustración 4 Perceptrón de 5 unidades

e. ¿para que se usa?

Un perceptrón al ser una neurona (Modelo matemático) lo que realiza es la asociación de más neuronas para poder llevar a cabo un proceso de respuesta y aprendizaje de las mismas.

Los perceptrones captan las señales y las extiende formando lo que conocemos como red neuronal.

f. ¿Cuál es su función de activación?

En redes computacionales, la Función de Activación de un nodo define la salida de un nodo dada una entrada o un conjunto de entradas. Se podría decir que un circuito estándar de computador se comporta como una red digital de funciones de activación al activarse como "ON" (1) u "OFF" (0), dependiendo de la entrada. Esto es similar al funcionamiento de un Perceptrón en una Red neuronal artificial.

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

En el perceptrón, existen dos tipos de aprendizaje, el primero utiliza una tasa de aprendizaje mientras que el segundo no la utiliza. Esta tasa de aprendizaje amortigua el cambio de los valores de los pesos.

El algoritmo de aprendizaje es el mismo para todas las neuronas, todo lo que sigue se aplica a una sola neurona en el aislamiento. Se definen algunas variables primero:

- el $x(j)$ denota el elemento en la posición j en el vector de la entrada
- el $w(j)$ el elemento en la posición j en el vector de peso
- el y denota la salida de la neurona
- el δ denota la salida esperada
- el α es una constante tal que $0 < \alpha < 1$

Los dos tipos de aprendizaje difieren en este paso. Para el primer tipo de aprendizaje, utilizando tasa de aprendizaje, utilizaremos la siguiente regla de actualización de los pesos:

$$w(j)' = w(j) + \alpha(\delta - y)x(j)$$

Para el segundo tipo de aprendizaje, sin utilizar tasa de aprendizaje, la regla de actualización de los pesos será la siguiente:

$$w(j)' = w(j) + (\delta - y)x(j)$$

h. Nombre 5 ejemplos donde se evidencie el uso del perceptron.

6) Adaline:

- Historia
- fórmula matemática, explique sus términos
- dibuje la estructura
- ¿para que se usa?
- ¿Cuál es su función de activación?
- ¿Cómo se entrena una red adaline indique el nombre del algoritmo y sus pasos?
- nombre 5 ejemplos donde se evidencie el uso de adaline.

En la década de 1950, con el avance de las computadoras, se hizo posible simular una red neuronal. Nathaniel Rochester, de los laboratorios de investigación de la IBM, dio el primer paso en la simulación de la red neuronal, pero desafortunadamente fracasó. 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 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.

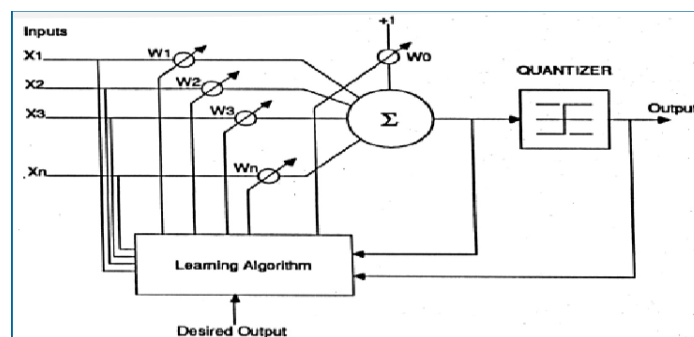


Ilustración 5Adline

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 (x_0, \dots, x_n) donde $x_0 = 1$, el correspondiente valor de pesos (w_0, \dots, w_n) y el valor deseado de salida d , el error cuadrático es el siguiente:

$$E = \left[d(t) - \sum_{i=1}^n W_i(t) X_i(t) \right]^2$$

El objetivo del método es minimizar dicho error mediante la modificación del vector de pesos (w_0, \dots, w_n) sumándole un Δw 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

$$E = d^2(t) - 2d(t) \sum_{i=1}^n W_i(t) X_i(t) + \left[\sum_{k=1}^n W_k(t) X_k(t) \right]^2$$

$$\frac{\partial E}{\partial W_i} = -2 \left[d(t) - \sum_{k=1}^n W_k(t) X_k(t) \right] X_i(t)$$

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

$$\Delta W_i(t+1) = \eta \left[d(t) - \sum_{k=1}^n W_k(t) X_k(t) \right] X_i(t)$$

El algoritmo de entrenamiento se presenta a continuación:

- Inicializar pesos (w_1, \dots, w_n) y threshold (w_0)
- Presentar vector de entrada (x_1, \dots, x_n) y la salida deseada $d(t)$
- Calcular la salida

$$y(t) = F_h \left(\sum_{i=0}^n w_i(t) * x_i(t) \right) \quad \text{donde } F_h(a) = 1 \text{ si } a > 0 \text{ y} \\ = -1 \text{ si } a \leq 0$$

$$w_i(t+1) = w_i(t) + \eta * \left[d(t) - \sum_{k=0}^n w_k(t) * x_k(t) \right] * x_i(t)$$

- Adaptar los pesos

Donde $0 < i < n$ y η es la tasa de aprendizaje

- 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 dependen de cada problema particular.

7) ¿Qué es y para que se usa las redes de retro propagación (backpropagation)

Al hablar de redes de retro propagación o redes de propagación hacia atrás hacemos referencia a un algoritmo de aprendizaje más que a una arquitectura determinada. La retro propagación 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 retro propagación 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 retro propagación 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) Algoritmo de BackPropagation

Descripción

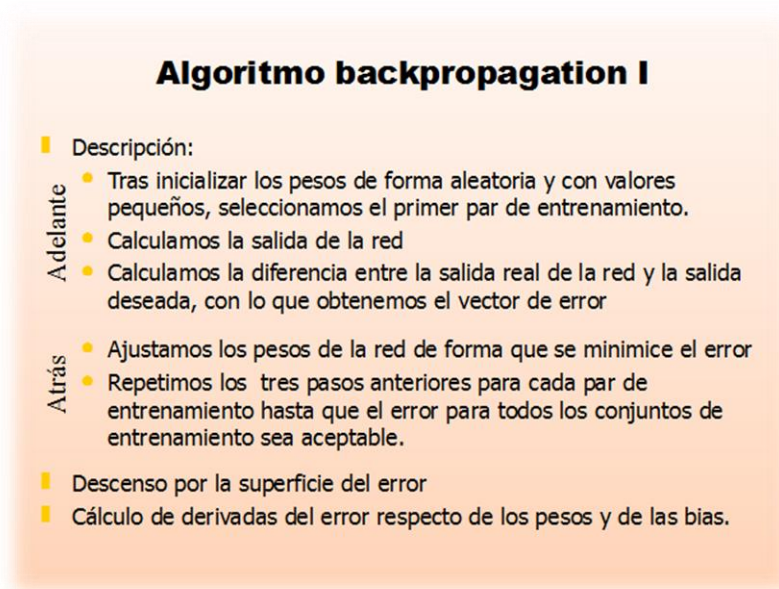


Ilustración 6 Algoritmo Backpropagation I

Algoritmo backpropagation I Descripción: Tras inicializar los pesos de forma aleatoria y con valores pequeños, seleccionamos el primer par de entrenamiento. Calculamos la salida de la red Calculamos la diferencia entre la salida real de la red y la salida deseada, con lo que obtenemos el vector de error Ajustamos los pesos de la red de forma que se minimice el error Repetimos los tres pasos anteriores para cada par de entrenamiento hasta que el error para todos los conjuntos de entrenamiento sea aceptable. Descenso por la superficie del error Cálculo de derivadas del error respecto de los pesos y de las bias. Adelante Atrás

Algoritmo backpropagation II

■ Detalles:

- SSE: $E = \sum_p E_p = \sum (y_{pk} - o_{pk})^2$
- $\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$

■ Pasos:

- Inicialización:
 - | Construcción de la red.
 - | Inicialización aleatoria de pesos y umbrales (-0.5, 0.5)
 - | Criterio de terminación (número máximo de iteraciones,...).
 - | Contador de iteraciones $n=0$.
- Fase hacia delante:
 - | Calcular la salida de la red para cada patrón de entrada.
 - | Calcular el error total cometido (SSE)
 - | Si la condición de terminación se satisface, parar
- Fase hacia atrás:

Ilustración 7 Algoritmo Backpropagation II

Algoritmo backpropagation II Detalles: SSE: $E = \sum E_p = \sum (y_{pk} - o_{pk})^2$ $\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$
 Pasos: Inicialización: Construcción de la red. Inicialización aleatoria de pesos y umbrales (-0.5, 0.5) Criterio de terminación (número máximo de iteraciones,...). Contador de iteraciones $n=0$. Fase hacia delante: Calcular la salida de la red para cada patrón de entrada. Calcular el error total cometido (SSE) Si la condición de terminación se satisface, parar Fase hacia atrás:

Algoritmo backpropagation III

- Fase hacia atrás:
 - | Incrementar el contador $n=n+1$.
 - | Para cada neurona de salida calcular: $\delta_k = (o_k - y_k) f'(net_k)$ donde $net_j = \sum_i w_{ij} x_i + b_j$
 - | Para cada unidad oculta, calcular $\delta_i = f'(net_i) \sum_k \delta_k w_{ik}$
 - | Actualizar pesos: $\Delta w_{ij}(n+1) = \eta \delta_j o_i + \alpha \Delta w_{ij}(n)$
 - | Volver a la fase hacia delante.
- Inconvenientes del algoritmo backpropagation:
 - Tiempo de entrenamiento no acotado.
 - Dependiente de las condiciones iniciales:
 - | Parálisis de la red.
 - | Mínimos locales.

Ilustración 8 Algoritmo Backpropagation III

Algoritmo backpropagation III Fase hacia atrás: Incrementar el contador $n=n+1$. Para cada neurona de salida calcular: $\delta_k = (o_k - y_k) f'(net_k)$ donde $net_j = \sum_i w_{ij} x_i + b_j$ Para cada unidad oculta, calcular $\delta_i = f'(net_i) \sum_k \delta_k w_{ik}$ Actualizar pesos: $\Delta w_{ij}(n+1) = \eta \delta_j o_i + \alpha \Delta w_{ij}(n)$ Volver a la fase hacia delante. Inconvenientes del algoritmo backpropagation: Tiempo de entrenamiento no acotado. Dependiente de las condiciones iniciales: Parálisis de la red. Mínimos locales.

Algoritmo Backpropagation IV

- Underfitting.
- Memorización o Sobreaprendizaje.
- Caracterización de la red. ¿Cuántas capas, cuántas neuronas en cada capa, ...?

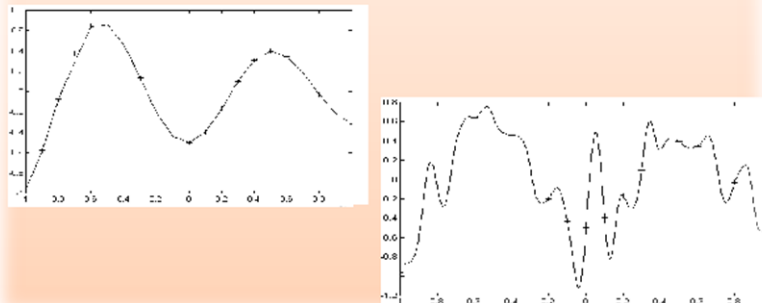


Ilustración 9 Algoritmo Backpropagation IV

Algoritmo Backpropagation IV Underfitting. Memorización o Sobre aprendizaje. Caracterización de la red. ¿Cuántas capas, cuántas neuronas en cada capa?

9) No realizado

¿Qué es el teorema de kolmogorov, explicar?

El teorema es un resultado de sistemas dinámicos sobre la persistencia de movimientos cuasiperiódicos. 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 teorema KAM 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