Redes neuronales artificiales

Alberto Reyes Ballesteros

INEEL
INSTITUTO NACIONAL
DE ELECTRICIDAD Y
ENERGÍAS LIMPIAS

INEEL
CECSE
CENTRO ESPECIALIZADO
DE CAPACITACIÓN PARA
EL SECTOR ENERGÍA

Introducción

Tema de investigación en la actualidad.

Las RNA's propuestas por las ciencias de la computación y las neurociencias son resultado del estudio de las funciones y estructuras del cerebro.

Son modelos computacionales basados en estos antecedentes biológicos para resolver problemas complejos como:

- Reconocimiento de patrones
- Procesamiento rápido de la información
- Aprendizaje y adaptación

Tareas de Aprendizaje

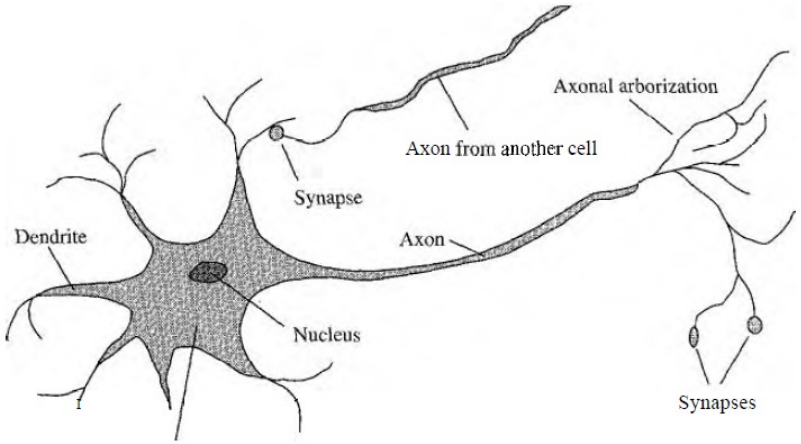
- Estimación o Regresión: Las clases son continuas.
- La meta es inducir un modelo para poder predecir el valor de la clase dados los valores de los atributos.
- Se usan, por ejemplo, árboles de regresión, regresión lineal, redes neuronales, LWR, etc.

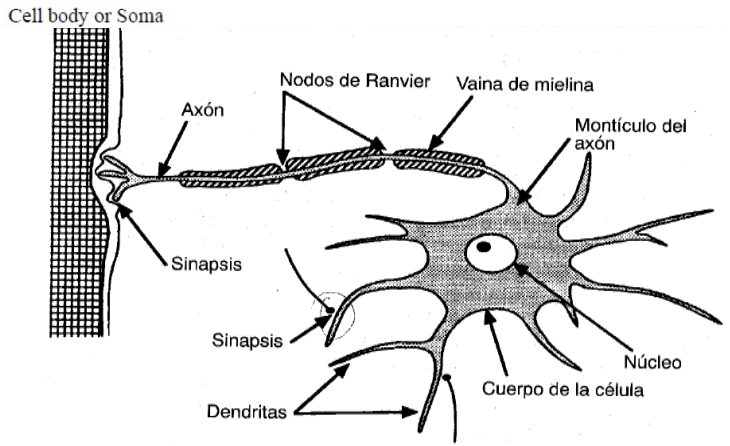
Redes neuronales de tipo biológico

Una neurona consta de un cuerpo de la célula o soma que contiene el núcleo de la célula. Del núcleo derivan varias fibras llamadas dendritas y una fibra larga llamada axón.

- Dendrita: llevan señales de los nervios al cuerpo de la célula.
- Axón: lleva señales de salida de la neurona a las dendritas de otras neuronas.
- Sinapsis: uniones entre neuronas.

Existen 10^{11} neuronas, cada una con 10^{3}-10^{4} conexiones a otras neuronas con pulsos eléctricos de milisegundos.





Introducción

Características:

- ✓Habilidad para aproximar funciones no-lineales arbitrarias.
  - Proveen modelos no-lineales requeridos para el diseño de controladores no-lineales y adaptables.
- ✓Estructura paralela:
  - Tolerancia a fallas y rapidez de operación.
- ✓Entrenadas con datos históricos del sistema en estudio
  - Habilidad para generalizar y ser adaptadas en línea.
- ✓Aplicables a procesos multivariables.

Redes NeuronalesArtificiales

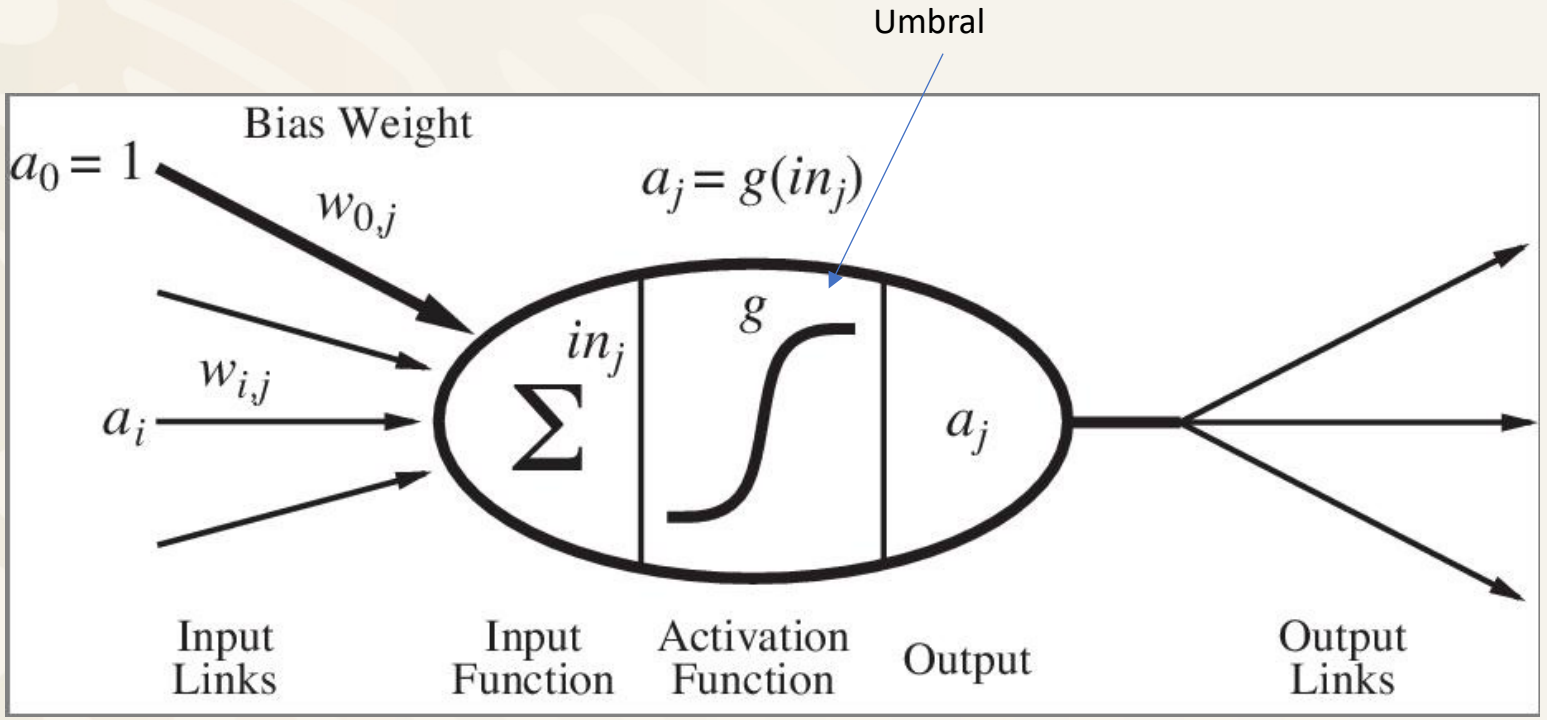
Estructura inspirada en un modelo simplificado de las neuronas biológicas.

- Se forma de un conjunto de elementos sencillos (neuronas) que tiene varias entradas y una salida.

$$\text{Salida} = f\left(\sum W_i E_i\right)$$

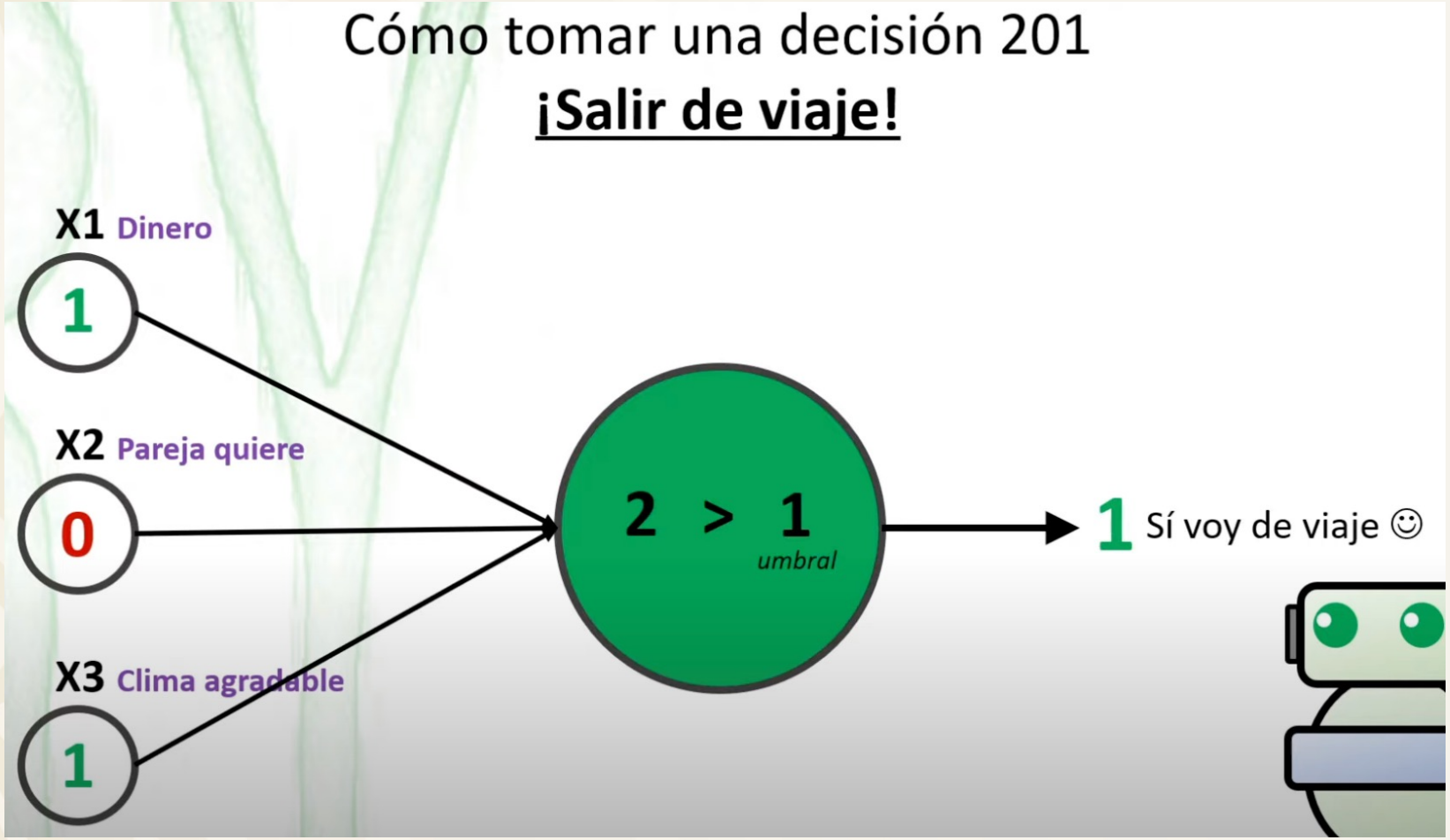
- Estos elementos se interconectan entre sí para formar redes (red neuronal).
- Las RN se entrenan para aprender relaciones de entrada-salida mediante la presentación de ejemplos, modificando los pesos.

Neurona básica o perceptrón

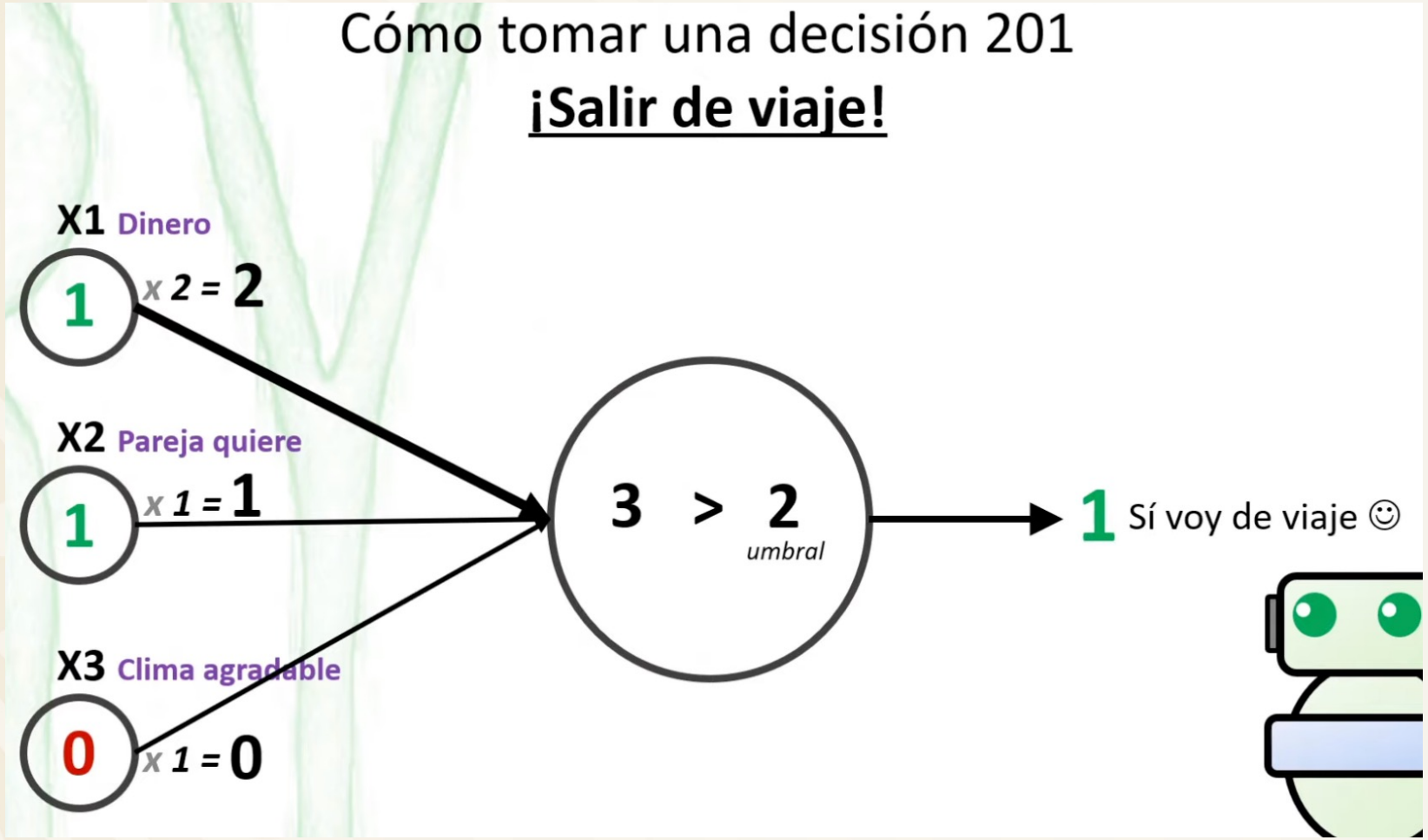


$$in_j = \sum_{i=0}^{n} w_{i,j} a_i \qquad a_j = g(in_j) = g\left(\sum_{i=0}^{n} w_{i,j} a_i \right)$$

Ejemplo

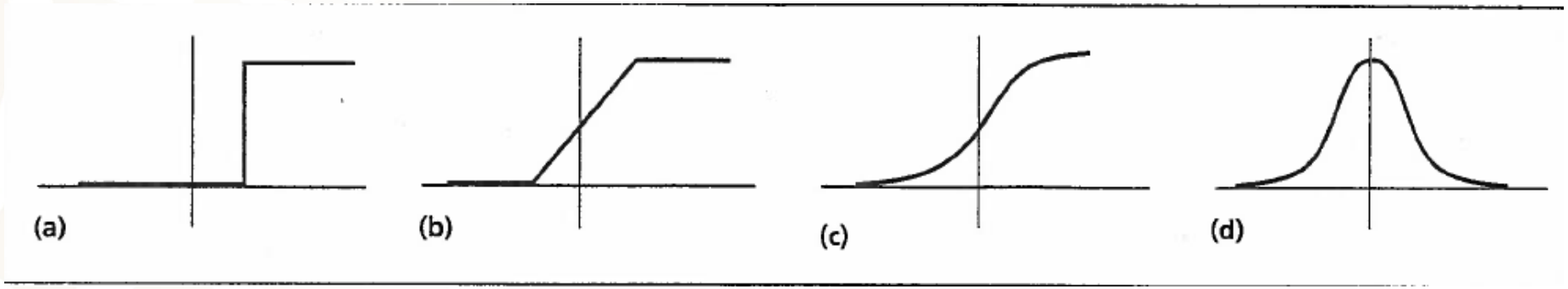

Cómo tomar una decisión 201
¡Salir de viaje!
X1 Dinero
1
X2 Pareja quiere
0
X3 Clima agradable
1
2 > 1
umbral
1 Sí voy de viaje ☺

Ejemplo


Cómo tomar una decisión 201
¡Salir de viaje!
X1 Dinero
1
x 2 = 2
X2 Pareja quiere
1
x 1 = 1
X3 Clima agradable
0
x 1 = 0
3 > 2
umbral
1 Sí voy de viaje ☺

Funciones de activación básicas

Existen diferentes funciones de activación g



Función umbral (o escalón)

Función rampa

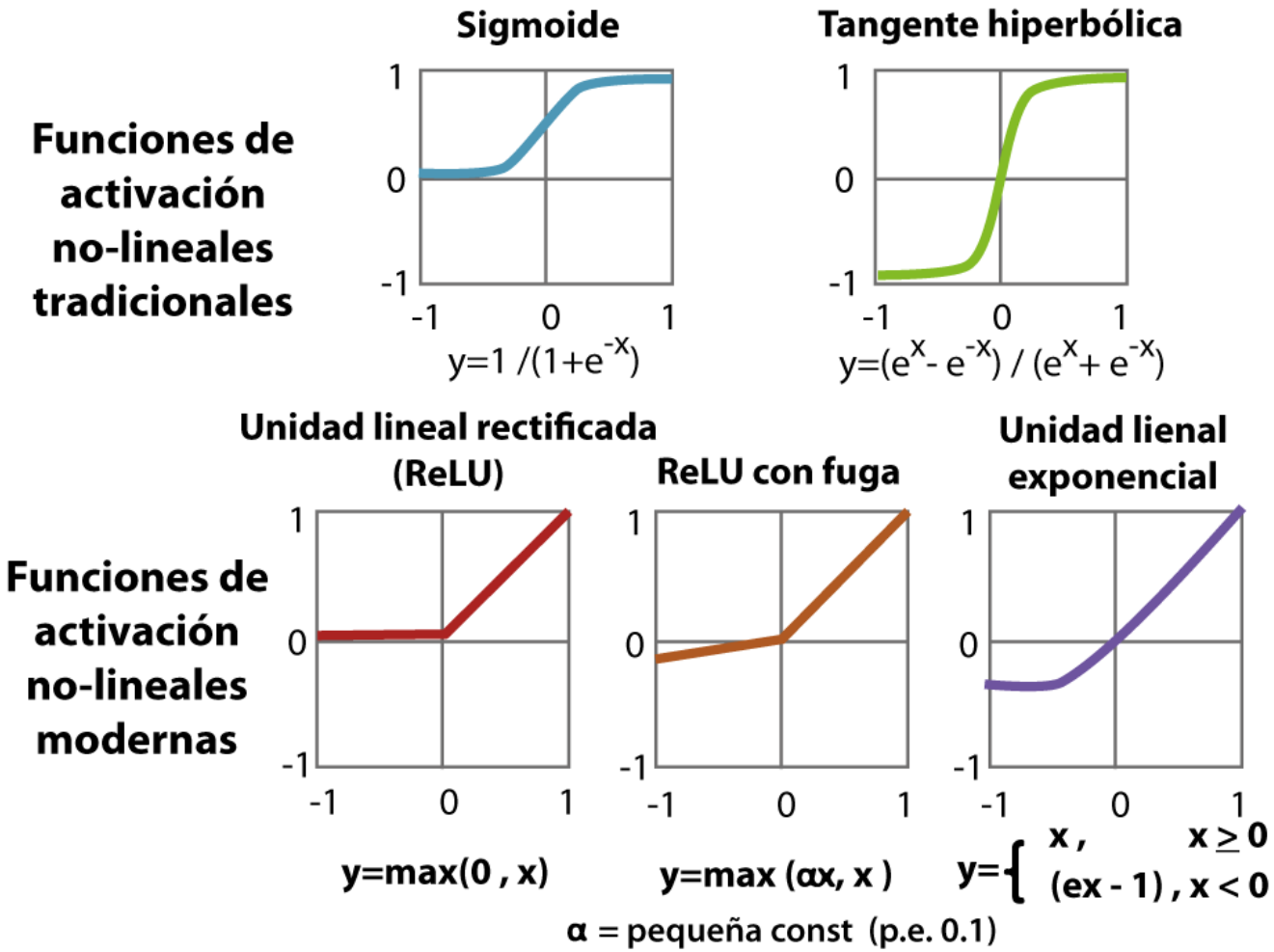
$$f(u_i) = \frac{1}{1+e^{-u_i/\sigma}}$$

Función sigmoidal (o logística)

$$f(u_i) = c * e^{u_i^2/\sigma^2}$$

Función gaussiana

Funciones de activación tradicionales vs las más usadas actualmente



Fases de aplicación

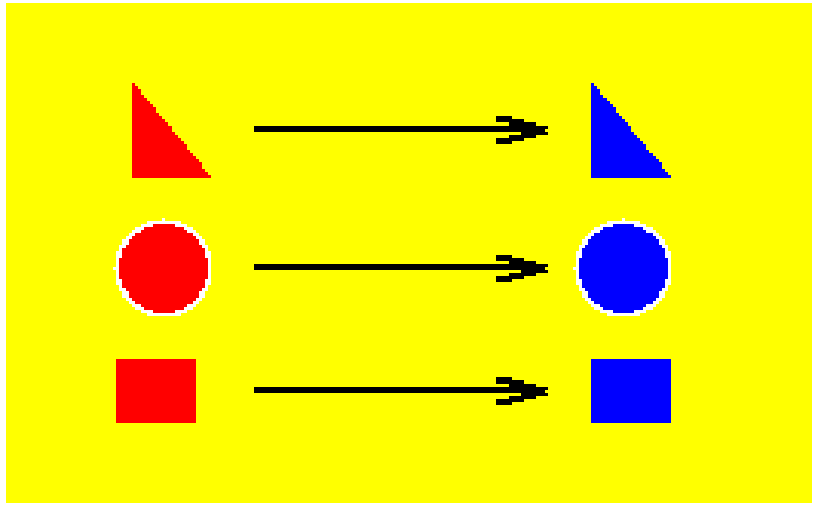
Fase de aprendizaje (training):

- aprenden por la actualización o cambio de los pesos sinápticos que caracterizan a las conexiones.
- Se usa un conjunto de datos o patrones de entrenamiento.

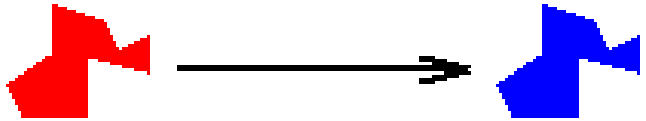
Fase de prueba (testing):

- Una vez calculados los pesos de la red, se comparan la(s) salida(s) de la red con la salida deseada.

Fases de aplicación



Training



Testing

Red neuronal multicapa

Redes neuronales con una o más capas ocultas.

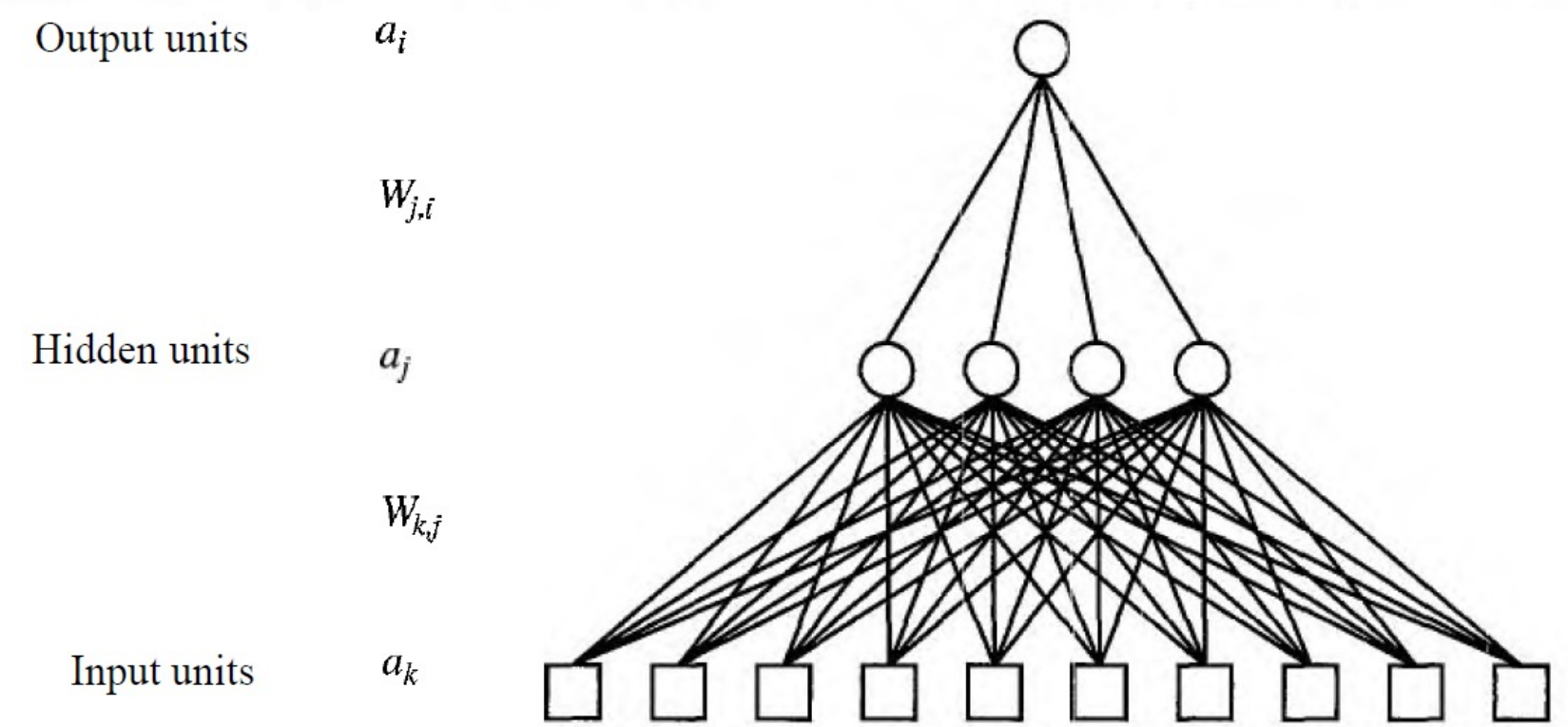
MLP - Multilayer Perceptrons (Perceptrón Multicapa)

Normalmente cada capa oculta de una red usa el mismo tipo de función de activación.

La función de activación de la salida es sigmoidal o lineal.

Son llamados aproximadores universales.

Red neuronal multicapa



RNA multicapa con una capa oculta y 10 entradas.

Red neuronal multicapa

- La capa de entrada tendrá tantas neuronas como número de variables de entrada y la capa de salida tantas neuronas como variables de salida.
- Las capas ocultas son capas de abstracción que consideran cuanto se refina un modelo

Red neuronal multicapa

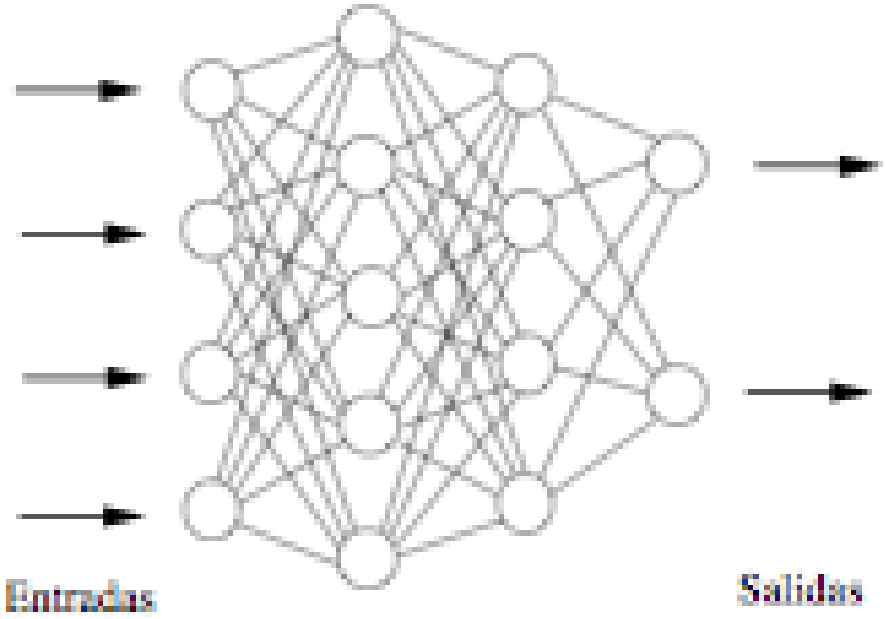
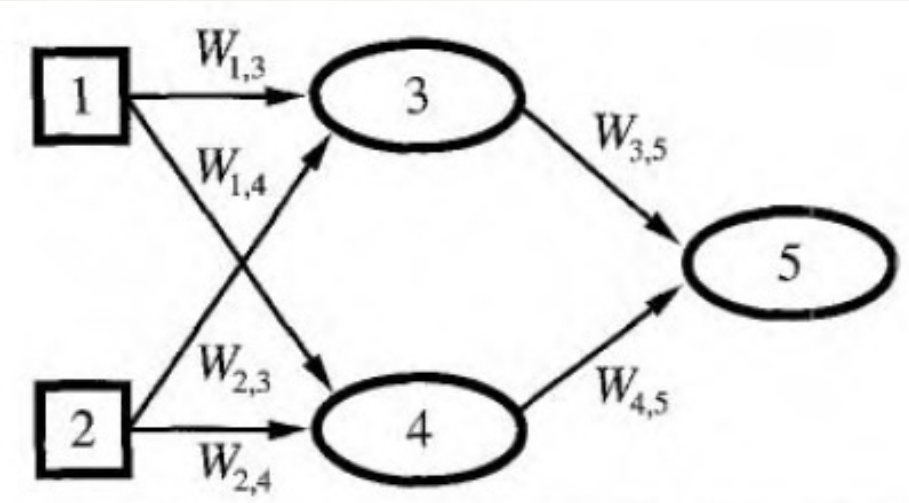


Figura: Red Neuronal prototípica.

Conforme se agregan más neuronas a la red las funciones que se podrían aprender son más complejas

Ejemplo de cálculo de la salida de la red



Red neuronal simple con dos entradas, una capa oculta de dos unidades y una salida

$$
\begin{aligned}
a_5 &= g(W_{3,5}a_3 + W_{4,5}a_4) \\
&= g(W_{3,5}g(W_{1,3}a_1 + W_{2,3}a_2) + W_{4,5}g(W_{1,4}a_1 + W_{2,4}a_2))
\end{aligned}
$$

a_3 a_4

g es la función de activación

BP-Backpropagation

- Se compara la salida “deseada” con la salida “actual” y se genera un error.

$$e(w) = \frac{1}{2}\sum_{k=1}^{M}[d_k - y_k(w)]^2$$

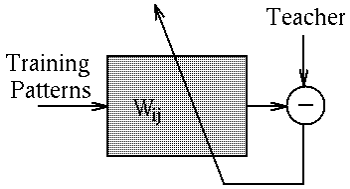
Error medio cuadrático

Salida deseada

Salida real (pesada)

- Se utiliza dicho error para modificar los “pesos” en las neuronas de salida.

$$w_{ij}(k+1) = w_{ij}(k) + \Delta_{ij}(k)$$



- Se propaga dicho error hacia atrás, modificando los pesos en las demás neuronas en la red.

Backpropagation: Seudocódigo

1. Inicialización aleatoria de pesos
2. Aplicar patrón de entrada
3. Propagación de la entrada a través de todas las capas
4. La RNA genera salidas y se calcula el error para cada neurona de salida
5. Los errores se transmiten hacia atrás, partiendo de la capa de salida hacia las neuronas de la capa intermedia
6. Este proceso se repite capa por capa.
7. Se reajustan los pesos de conexión da cada neurona en base al error recibido.

Algoritmo Backpropagation

function BACK-PROP-LEARNING(*examples*, *network*) **returns** a neural network
inputs: *examples*, a set of examples, each with input vector x and output vector y
network, a multilayer network with **L** layers, weights $W_{j,i}$, activation function g

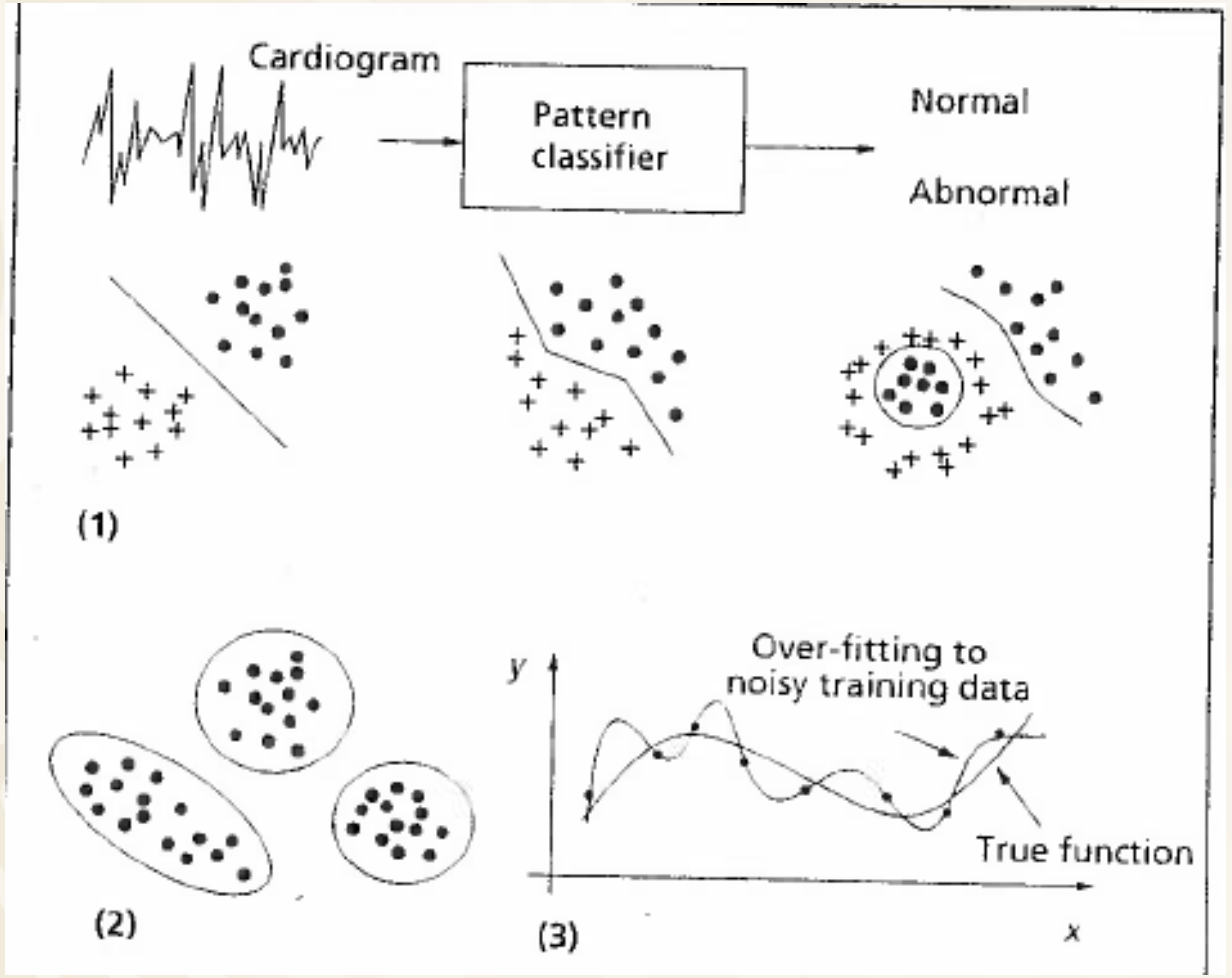
repeat
for each e **in** *examples* **do**
for each node j in the input layer **do** $a_j \leftarrow x_j[e]$
for $\ell = 2$ **to** **L** **do**
$in_i \leftarrow \sum_j W_{j,i}\, a_j$
$a_i \leftarrow g(in_i)$
for each node i in the output layer **do**
$\Delta_i \leftarrow g'(in_i) \times (y_i[e] - a_i)$
for $\ell = L - 1$ **to** 1 **do**
for each node j in layer ℓ **do**
$\Delta_j \leftarrow g'(in_j) \sum_i W_{j,i}\, \Delta_i$
for each node i in layer $\ell + 1$ **do**
$W_{j,i} \leftarrow W_{j,i} + \alpha \times a_j \times \Delta_i$
until some stopping criterion is satisfied
return NEURAL-NET-HYPOTHESIS(*network*)

examples

	x1	x2	xj	…	y1	…	yi
1							
2							
…							
e							

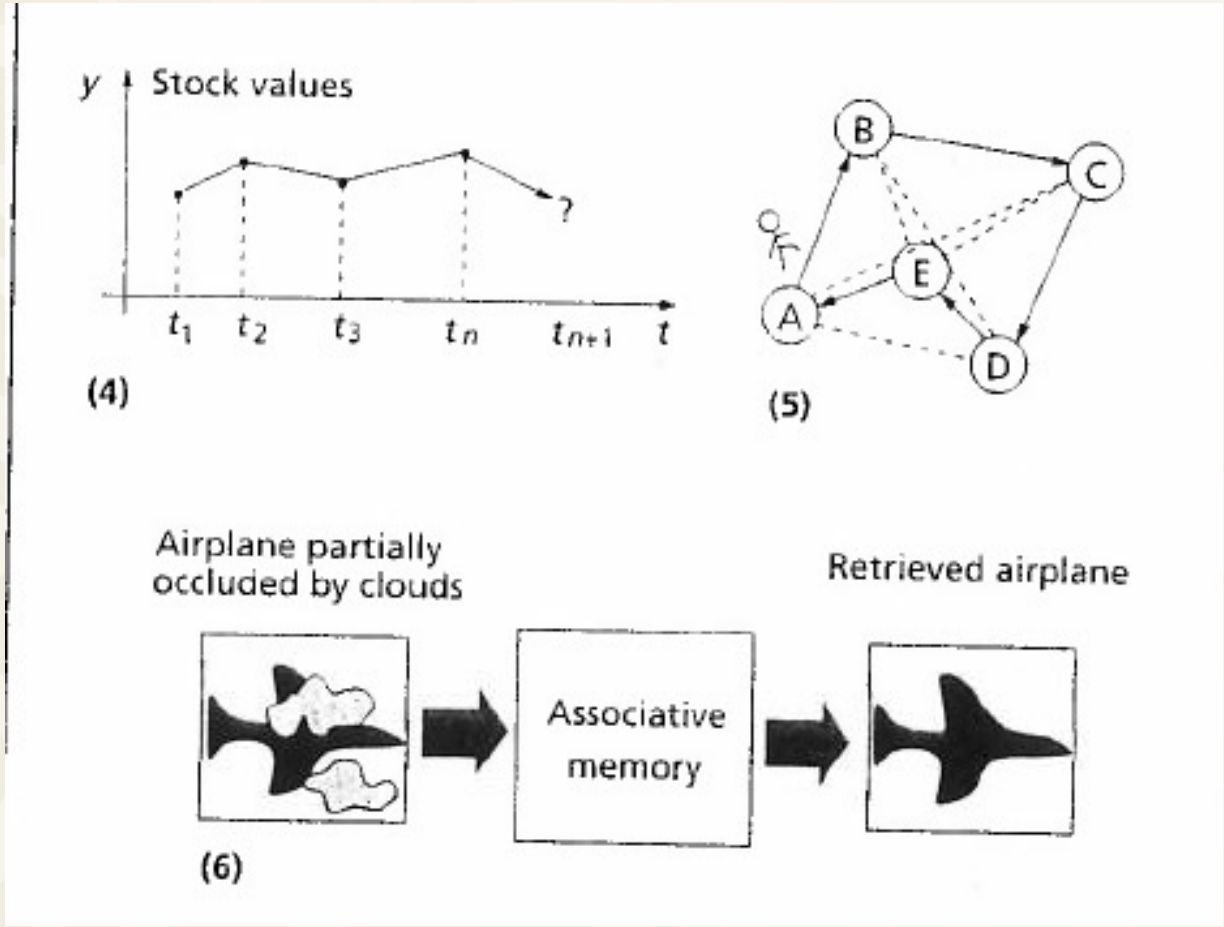
The back-propagation algorithm for learning in multilayer networks.

Tipos de aplicaciones



(1) Clasificación de patrones
(2) Categorización en grupos (clusters)
(3) Aproximación de funciones

Tipos de aplicaciones



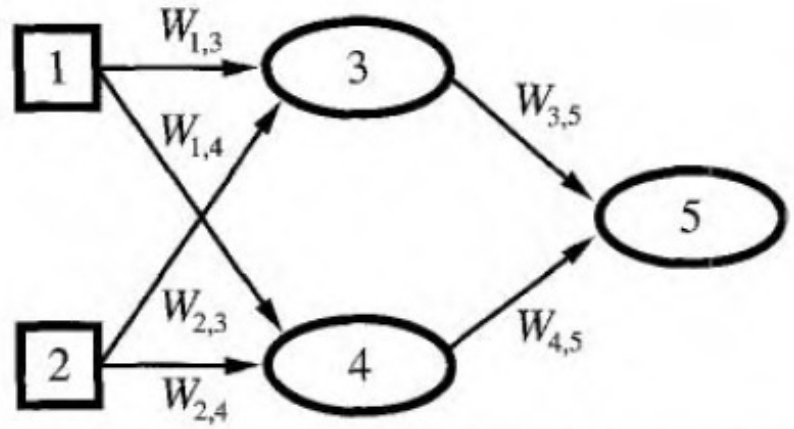
(4) Pronóstico

(5) Optimización

(6) Recuperación de contenidos

Ejercicio

Utilice excel para calcular la salida del nodo 5 de la red de la figura usando como entradas el primer ejemplo del conjunto de datos mostrado y calcule el error medio cuadrático. La función de activación es la sigmoide ($g = 1 / (e^{-U})$). Explique sus resultados y suba el ejemplo a la plataforma.



W(1,3)	0.5
W(1,4)	0.8
W(2,3)	1
W(2,4)	0.9
W(3,5)	1.5
W(4,5)	2

X1	x2	y
1	2	3
2	3	5
3	4	7
4	5	9

INEEL
INSTITUTO NACIONAL DE ELECTRICIDAD Y ENERGÍAS LIMPIAS

INEEL CECSE
CENTRO ESPECIALIZADO DE CAPACITACIÓN PARA EL SECTOR ENERGÍA

GRACIAS

Alberto Reyes Ballesteros
areyes@ineel.mx