COMPUTACION INTELIGENTE
(aprendizaje)

Jose Aguilar
Cemisid, Facultad de Ingeniería
Universidad de los Andes
Mérida, Venezuela
aguilar@ing.ula.ve

Redes Neuronales Artificiales

RNA:

Habilidad de aprender del medio ambiente y mejorar su desempeño por medio del aprendizaje.

Aprendizaje:

Proceso por el cual los parámetros (pesos sinápticos) de una RNA se adaptan por estimulación del medio ambiente.

Algoritmo de Aprendizaje:

Conjunto bien definido de reglas para actualizar los pesos sinápticos.

Paradigma de Aprendizaje:

Modelo del medio ambiente en el cual la RNA opera.

APRENDIZAJE

- ***REDES NEURALES:***
  - ***CAPACIDAD PARA APRENDER***
  - ***OPTIMIZA SU RENDIMIENTO ASI***

APRENDIZAJE: *MODIFICAR PESOS DE LAS CONEXIONES DE LAS NEURONAS (CREAR, DESTRUIR, MODIFICAR)*

$$Wij = Wij + \Delta Wij$$

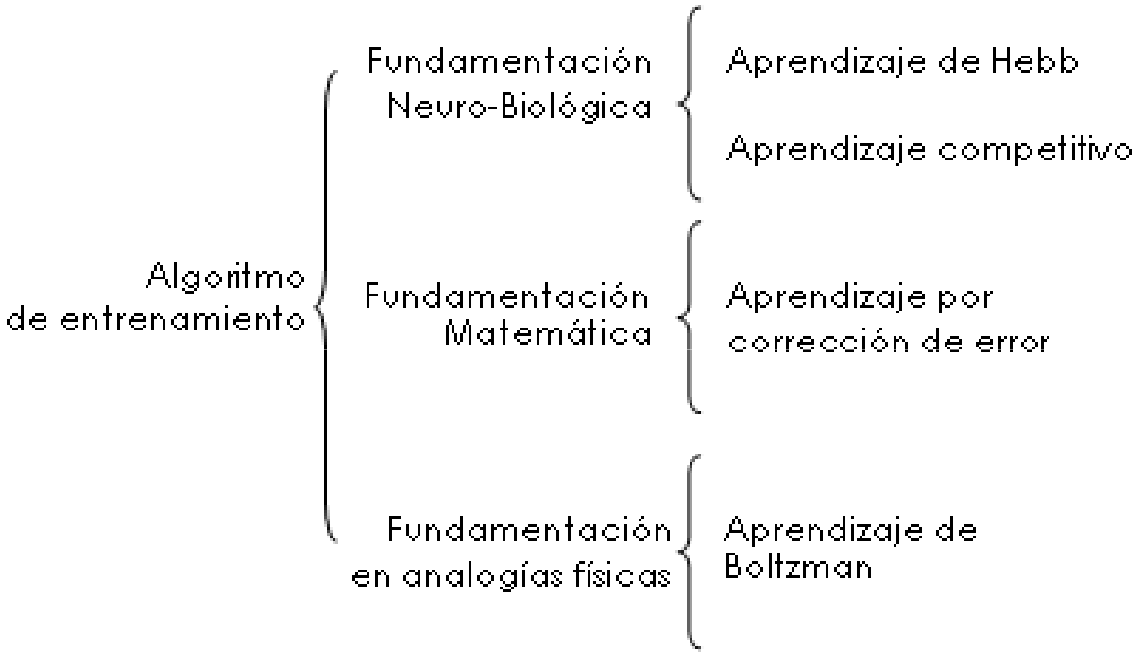
ENTRENAMIENTO O APRENDIZAJE

El entrenamiento o aprendizaje de una RNA se basa en un proceso que permite que la red aprenda a comportarse según unos objetivos específicos realizando para ellos modificaciones en sus parámetros de pesos sinápticos y sesgo a través de reglas de aprendizaje. En particular, las reglas de aprendizaje son procedimientos que se siguen para cambiar el valor asignado los pesos.

El aprendizaje es esencial para la mayoría de las arquitecturas de RNA, por lo que la elección de un algoritmo de aprendizaje es algo de gran importancia en el desarrollo de una red. El aprendizaje implica que una unidad de procesamiento es capaz de cambiar su comportamiento, es decir su proceso de entrada-salida, como resultado de los cambios en el medio.

Para alcanzar el aprendizaje de una red neuronal artificial existen diferentes algoritmos de entrenamiento, que se pueden clasificar de acuerdo a su fundamentación conceptual, esto es:

ENTRENAMIENTO O APRENDIZAJE



Clasificación de los algoritmos de entrenamiento basados en su fundamentación conceptual

Modelos Neuronales

Clasificación por tipo de aprendizaje y arquitectura

Híbridos

Supervisados

- Realimentados
- Unidireccionales

No supervisados

- Realimentados
- Unidireccionales

Reforzados

Características de una RNA

- ✓ *Mecanismos de Aprendizaje:* Es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. Los cambios que se producen durante el proceso de aprendizaje son la destrucción, modificación y creación de conexiones entre las neuronas.

DEFINE COMO SE RELACIONA CON SU ENTORNO

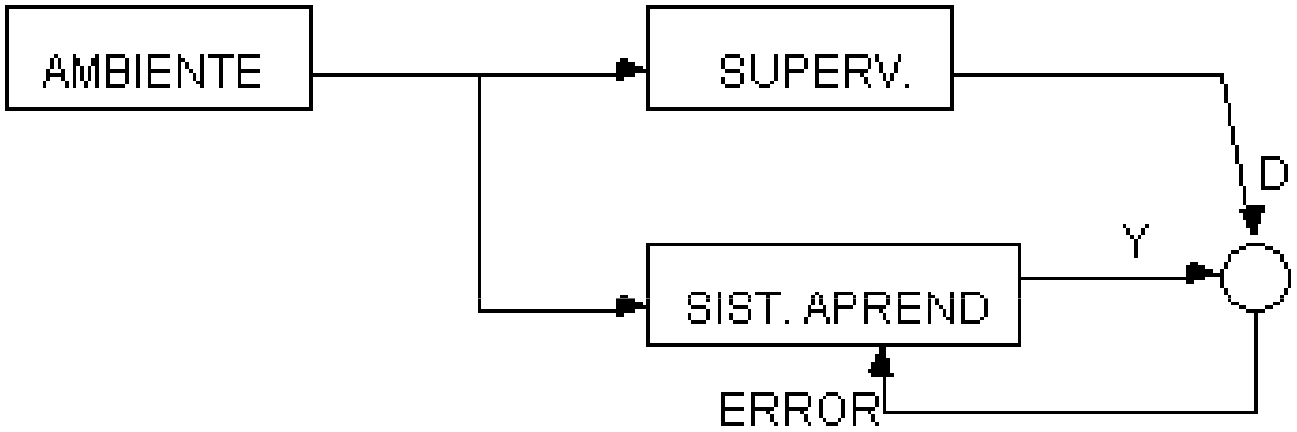
Tipos de aprendizaje:

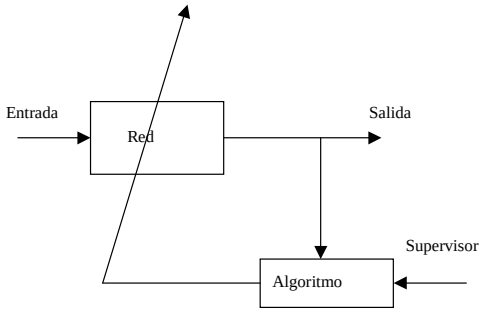
- ***SUPERVISADO***
- ***NO SUPERVISADO***
- ***REFORZADO***

Características de una RNA

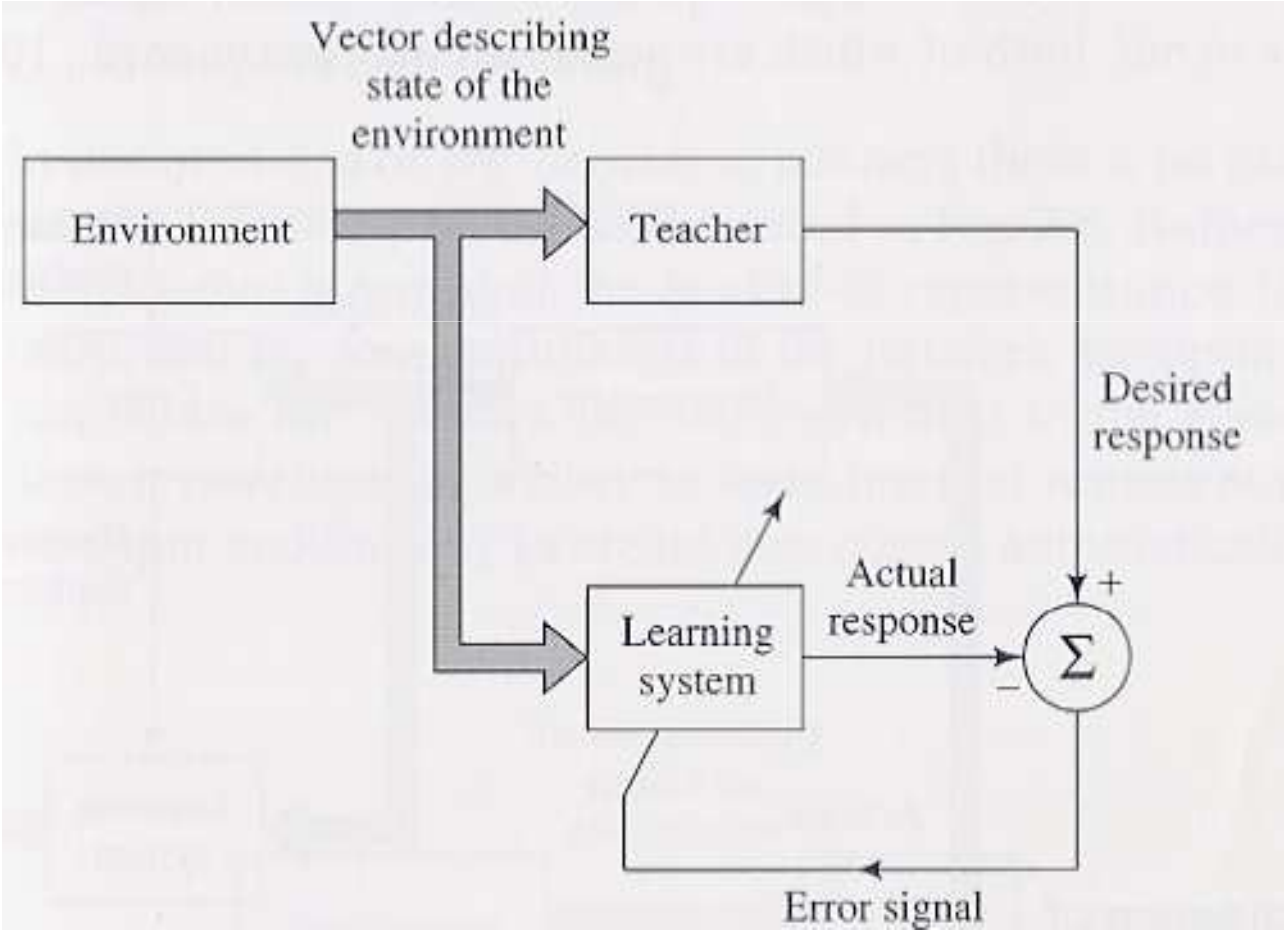
- ✓ *Redes con Aprendizaje Supervisado:* Se realiza mediante un entrenamiento controlado por un agente externo que determina la respuesta que debería generar la red a partir de una entrada determinada.
  - Aprendizaje por corrección de error: Consiste en ajustar los pesos en función de la diferencia entre los valores deseados y los obtenidos en la salida de la red.
  - Aprendizaje por refuerzo: La función del supervisor es indicar mediante una señal de refuerzo si la salida obtenida en la red se ajusta a la deseada (éxito=+1 o fracaso=-1)
  - Aprendizaje estocástico: Este tipo de aprendizaje consiste básicamente en realizar cambios aleatorios en los valores de los pesos de las conexiones de la red.

SUPERVISADO





Paradigmas de Aprendizaje Supervizado



SUPERVISADO

A. CONTROLADO POR SUPERVISOR

B. SALIDA RED DEBE CONCORDAR CON LA DESEADA

C. SE DAN DATOS DE ENTRADA Y SALIDA

NO SUPERVISADO (AUTOORGANIZADO)

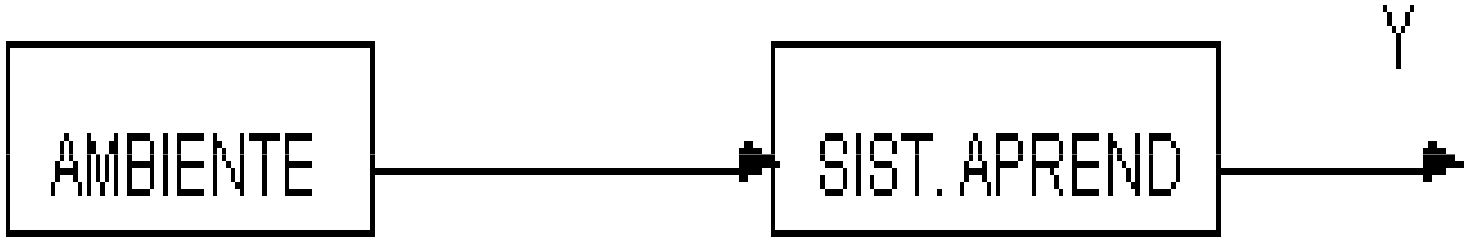
A. NO RECIBE INFORMACIÓN DE SU ENTORNO

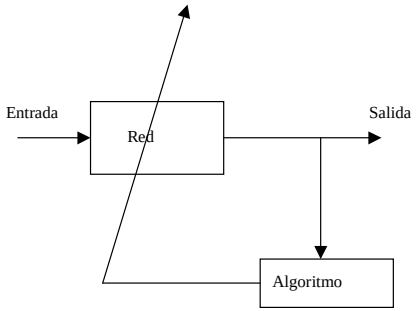
B. CON LOS DATOS SE BUSCAN CORRELACIONES PARA VER QUE TAN BIEN ESTÁ RINDIENDO LA RED

C. RENDIMIENTO SE MIDE SEGÚN COMO UNO INTÉRPRETE RESULTADOS

D. MAPAS TOPOGRÁFICOS

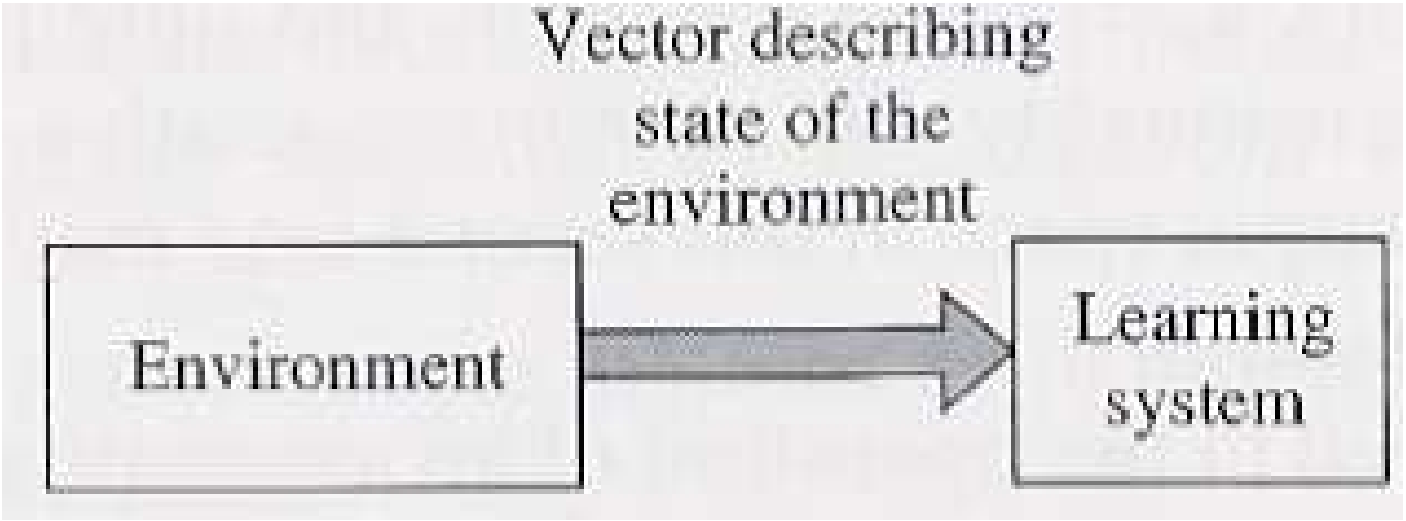
NO SUPERVISADO (AUTOORGANIZADO)





Paradigmas de Aprendizaje
No Supervizado

- Auto-organizado
- Medida de calidad de representación, independiente del contexto
- Los parámetros son optimizados con respecto a esa medida
- Clasificación

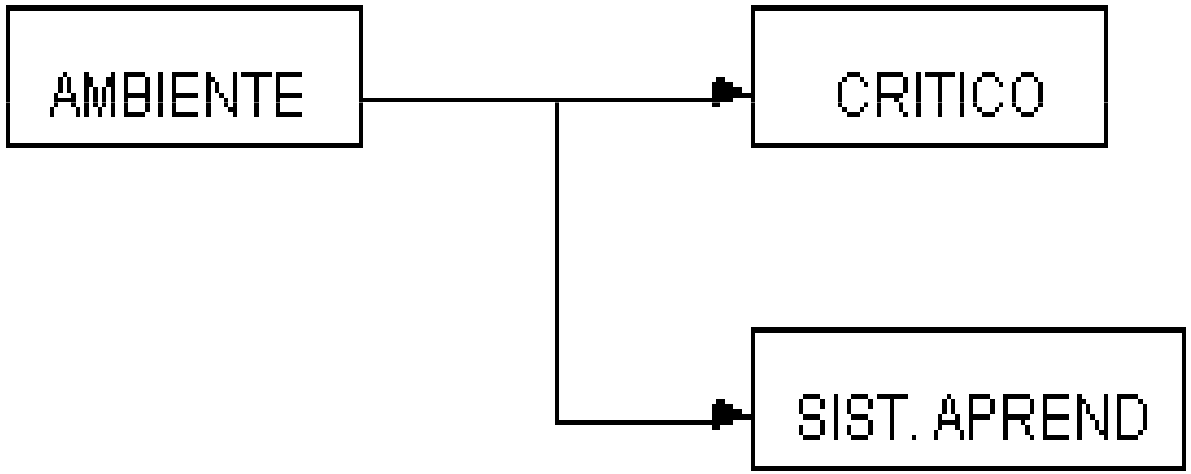


REFORZADO

A. SUPERVISOR INDICA SI SALIDA SE AJUSTA A LO DESEADO O NO

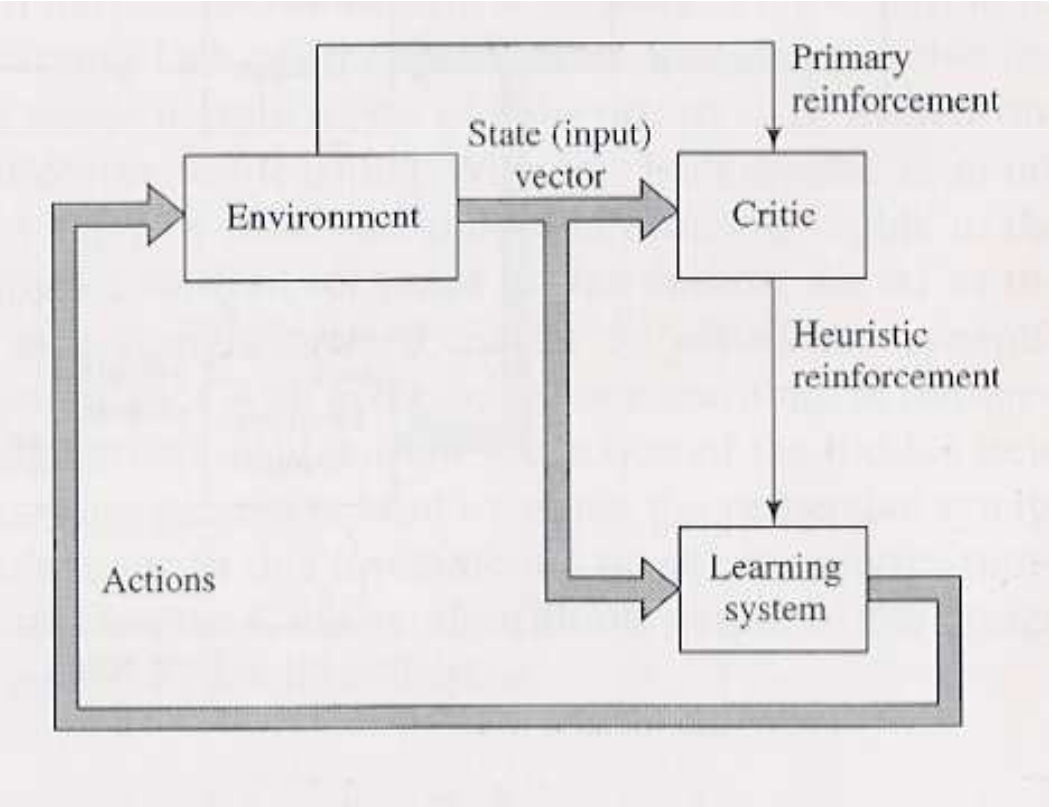
B. SUPERVISOR HACE PAPEL DE CRÍTICO MÁS QUE DE MAESTRO

REFORZADO



Paradigmas de Aprendizaje Por Refuerzos

- Minimizar un índice escalar de aptitud
- Refuerzo retardado
- Asignación de Crédito y Culpa a acciones
- Aprende a realizar tareas basado solamente en el resultado de sus experiencias



Estrategias de Aprendizaje

CORRECCIÓN DE ERROR

SUPERVISADO

CONOCIDO TAMBIEN COMO ***DESCENSO DE GRADIENTE***

$$E_k(t) = D_k(t) - Y_k(t)$$

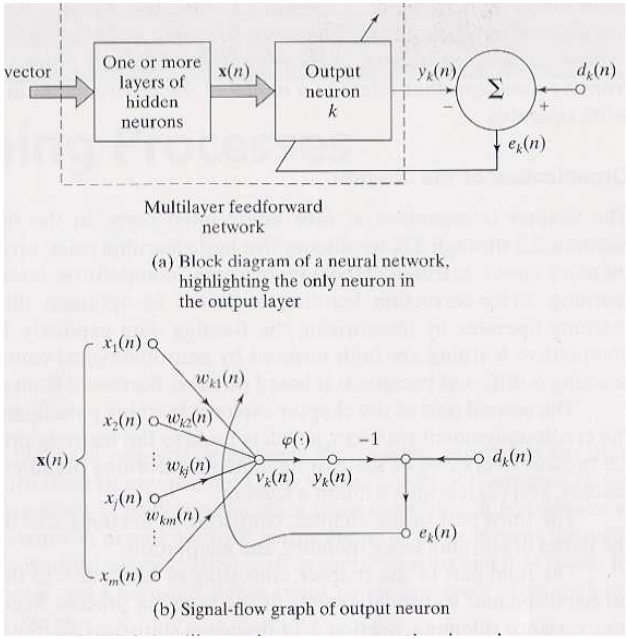
D_k: *respuesta deseada*

Y_k: *respuesta neurona k* $\qquad Y_k=F(X_k)$

X_k: *entrada neurona k*

$$\Delta Wij(t) = \alpha\, Ei(t)\, Xj(t)$$

α: *taza de aprendizaje*



REGLA DE WIDROW-HOLF

$$\text{ERROR GLOBAL} = \frac{1}{2P} \sum_{K=1}^{P} \sum_{J=1}^{N} \left(Y_J^K - d_J^K \right)$$

N: Número de Neuronas

P: Número de Información a Aprender

$$\Delta Wji = \alpha \partial \text{ ERROR}_{GLOBAL} / \partial Wji$$

CORRECCIÓN DE ERROR

" Regla Delta":

$$E_{global} = E/2 \; \Sigma p \Sigma n (E_k^2(t))$$

p: No ejemplos *n: No neuronas*

$$\Delta Wij(t) = \alpha \, dE_{global}/dWij$$

CORRECCIÓN DE ERROR

RETROPROPAGACIÓN DEL ERROR

1. *CALCULAR EDO. DE LA RED (Yi)*
2. *CALCULAR ERROR (1)*
3. *MINIMIZAR ERROR (2)*
4. *AJUSTAR PESOS*

$$Wij = Wij + \Delta Wij$$

HEBBIANO

A. MÁS VIEJO (no supervisado)

B. DOS O MAS NEURONAS ACTIVADAS SIMULTANEAMENTE

=> REFORZAR LA CONEXIÓN ENTRE ELLAS

$$\Delta W_{ij} = \alpha Y_i Y_j$$

Algoritmos de Aprendizaje Hebbian

1. Si dos neuronas vecinas se activan simultáneamente, el peso de su conexión se incrementa.
2. Si dos neuronas vecinas se activan asíncronamente, la conexión se debilita o elimina.

La conexión se llama sinapsis hebbiana

Algoritmos de Aprendizaje Hebbian

Mecanismos en aprendizaje hebbiano

1. Dependiente del Tiempo. Aprendizaje depende de los tiempos de activación.
2. Local. Activaciones Espacio-Temporales.
3. Interactivo. Depende de la interacción entre neuronas vecinas.
4. Conjuncional (correlacional). La co-ocurrencia de señales produce una modificación de pesos.

Algoritmos de Aprendizaje Hebbian

W_{kj} : peso sináptico

x_j :señal presináptica

y_k: señal postsináptica

$$\Delta w_{kj}(n) = F(y_k(n), x_j(n))$$

$\Delta w_{kj}(n) = \eta\, y_k(n)\, x_j(n)$ — hipótesis de Hebb

Algoritmos de Aprendizaje Basado en Memoria

Almacenar ejemplos entrada – salida

$$\{(x_i, d_i)\} \quad i=1, .., N$$

Ejemplo: Clasificación binaria

Clases: C_1 y C_2

x_{test} (nuevo) es clasificado examinando su vecindad

Algoritmos de Aprendizaje Basado en Memoria

Componentes:

- Criterio para definir vecindad
- Regla de aprendizaje

Ejemplo: Vecino más Cercano (Nearest Neighbor)

x^* e $\{x_1, x_2, .., x_N\}$ es el vmc de x_{test} si

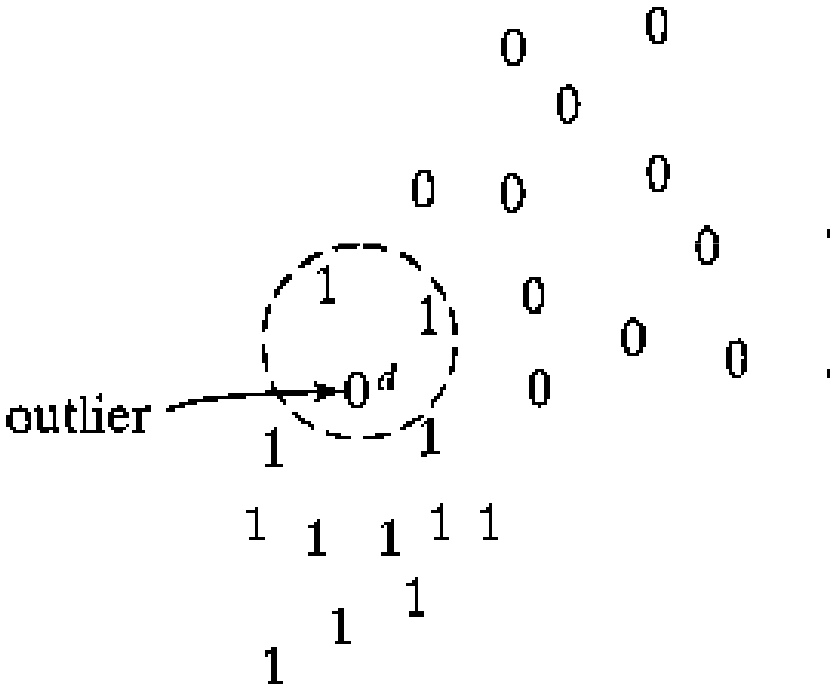
$Min_i\ d(x_i, x_{test}) = d(x^*, x_{test})$

Donde $d(x_i, x_{test})$ es la distancia Euclideana

Algoritmos de Aprendizaje Basado en Memoria

k Vecinos más cercanos:

- Identificar los k vecinos mas cercanos a x_{test}
- Asignar a x_{test} la clase más común en sus k vecinos más cercanos

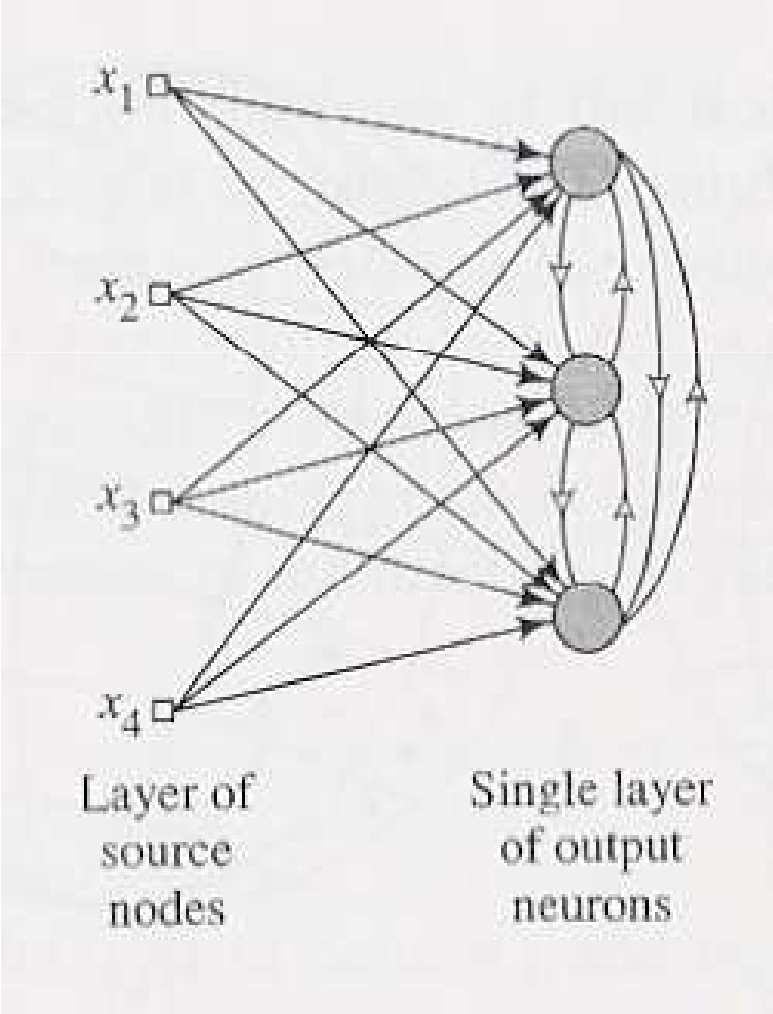


Algoritmos de Aprendizaje Competitivo

- Las neuronas de salida compiten para activarse.

- Solo una neurona de salida estará activa a la vez.

- Usadas en clasificación.



COMPETITIVO

A. *NEURONAS COMPITEN (COOPERAN PARA HACER UNA TAREA)*

B. *APRENDIZAJE*

=> NEURONAS QUE COMPITEN SE INHIBEN ,

NEURONAS QUE COOPERAN SE EXCITAN

=> Wij REFORZADO SI NEURONA j NO ES EXCITADA POR i Y VICEVERSA

$$\Delta Wij = \begin{cases} \alpha (Xi - Wij) \text{ SI NEURONA j GANA} \\ 0 \text{ SI NEURONA j PIERDE} \end{cases}$$

Algoritmos de Aprendizaje Competitivo

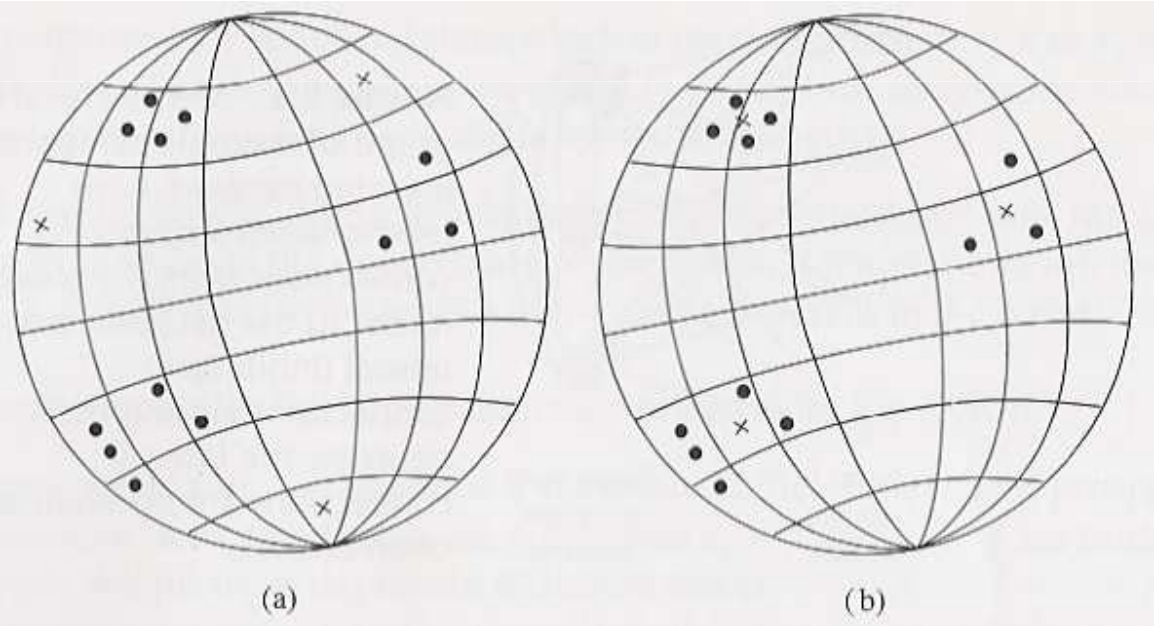
$y_k = 1$ si $v_k > v_j \quad \forall j \neq k$

$y_k = 0$ otro caso

$\Sigma_i w_{kj} = 1 \ \forall k$

$\Delta w_{kj} = \eta (x_j - w_{kj})$

$\Delta w_{kj} = 0$



REFORZADO

Señal de refuerzo

$$W_{ji}(t)=W_{ji}(t-1)+\alpha(r-\sigma_j)e_{ji})$$

donde

$$e_{ji}=\partial ln(g_i)/\partial W_{ji} \qquad g_i=Pr(y_j=b_{kj}/W_{ji}, A_k)$$

EVOLUTIVO

A. USA ALGORITMOS EVOLUTIVOS

B. CAMBIOS POSIBLES:

PARÁMETROS DE LA RED DE NEURONAS

ARQUITECTURA DE LA RED

RED DE RETROPROPAGACIÓN

A. 1986 RUMELHART, HINTON, WILLIAMS

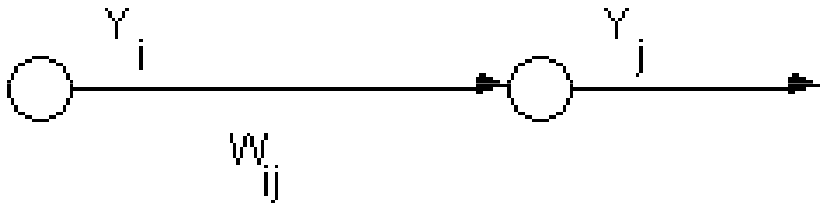
B. ASOCIA PATRONES DE ENTRADA A LA RN

C. APRENDIZAJE SE PUEDE APLICAR PARA MAS DE 2 CAPAS

- SE PRESENTE ENTRADA A LA 1era CAPA
- SE PROPAGA A TRAVES DE LAS CAPAS HASTA LA SALIDA
- SE COMPARA RESULTADO OBTENIDO EN LA NEURONAS DE SALIDA CON LA DESEADA
- SE CALCULA ERROR Y SE TRANSMITE HACIA ATRÁS

RED DE RETROPROPAGACIÓN

D. RN SE ADAPTA MODIFICANDO LOS PESOS DE LAS CAPAS INTERMEDIAS



$\Delta Wij(t+1)=\alpha\ \phi_{pj} Y_{pi}$; p: patrón a aprender

ϕ_{pj}= **D_{pj}-Y_{pj}**

RED DE RETROPROPAGACIÓN

PROBLEMA: NO SE CONOCE SALIDA NEURONAS OCULTAS

=> $\phi_{pj} = (D_{pj} - Y_{pj}) * F'(net_j)$ **NEURONAS SALIDA**

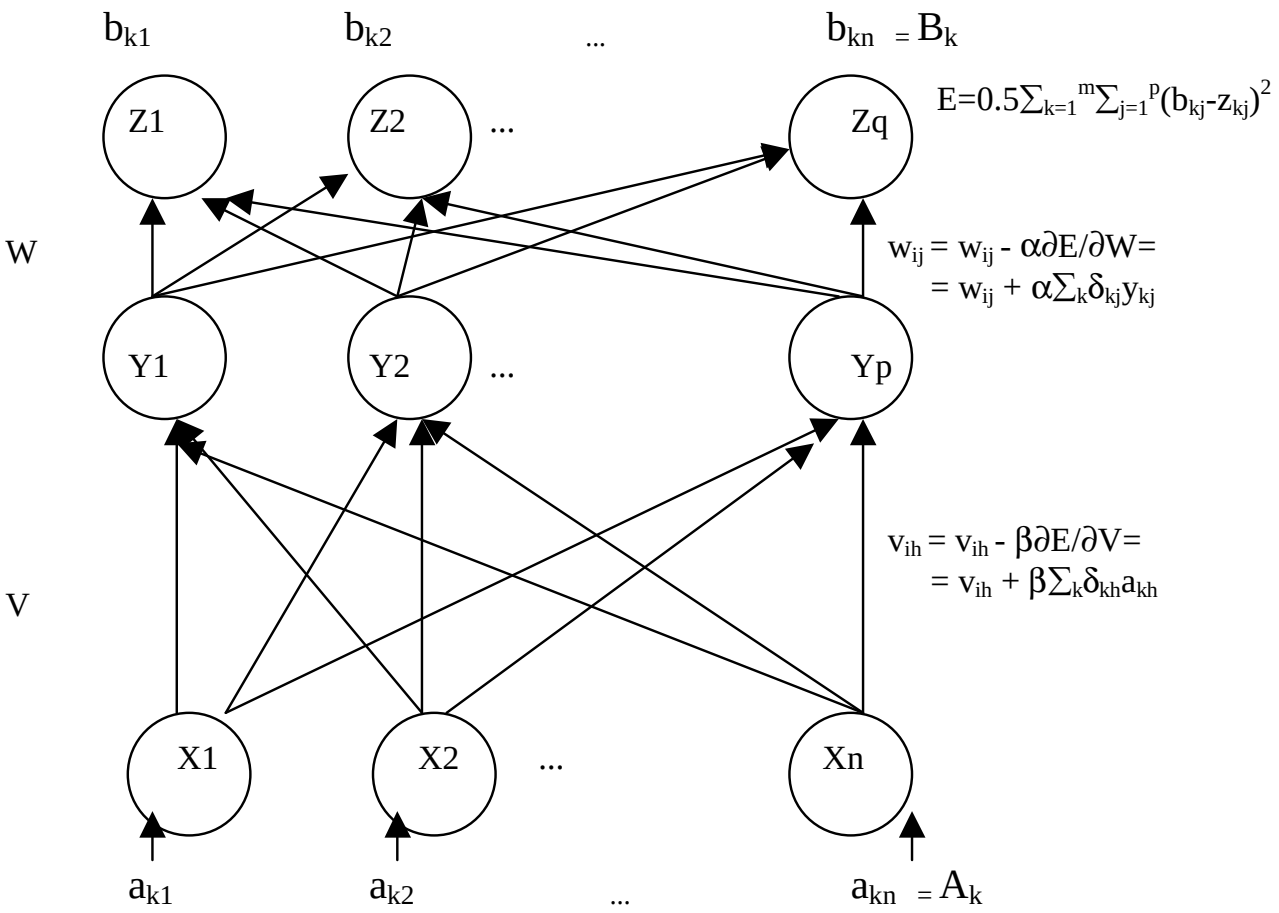
=> $\phi_{pj} = (\Sigma_k \phi_{pk} W_{kj}) * F'(net_j)$ **NEURONAS OCULTAS**

ADEMÁS,

$$\Delta Wij(t+1) = \alpha \phi_{pi} Y_{pj}$$

con momento: $\Delta Wij(t+1) = \alpha \phi_{pi} Y_{pj} + \beta(Wij(t) - Wij(t-1))$

RED DE RETROPROPAGACIÓN



APRENDIZAJE ESTOCASTICO

- **Caso clásico: Boltzman**

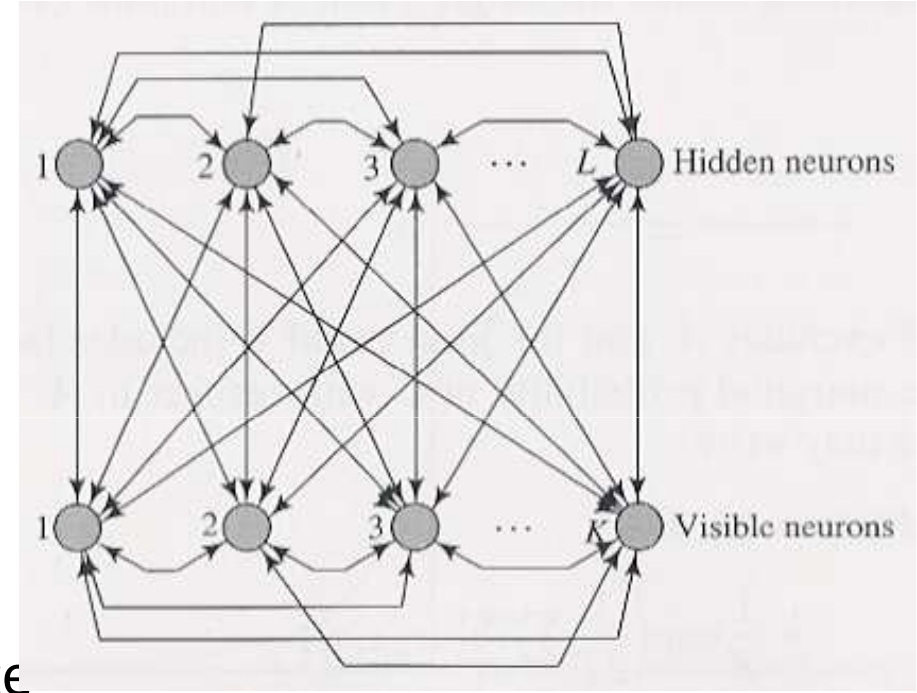
- **Función de energía:**

$$E=1/2\sum i \sum j w_{ji} s_j s_i$$

- **Proceso de aprendizaje**

$$W(s_j\text{-}s'_j)= 1/ (1+\exp(-\Delta E_j/T))$$

Algoritmos de Aprendizaje Boltzmann



Estructura recurrente

Estados: 1 y -1

$E = - \frac{1}{2} \sum_j \sum_k w_{kj} x_k x_j$ para $j \neq k$ — Energy

Seleccionar neurona k aleatoriamente

Cambiar su estado a temperatura T con probabilidad

$$P(x_k \rightarrow -x_k) = 1 / (1 + \exp(-\Delta E_j / T))$$

$\Delta w_{kj} = \eta (\rho_{kj}^{+} - \rho_{kj}^{-})$ para $j \neq k$

Algoritmos de Aprendizaje Boltzmann

- Neuronas:
  - Fijas (entradas)
  - Libres
- Utilizadas para modelar (aprender) una distribución de probabilidad de los datos de entrada.
- Completan entradas incompletas.

ENTRENAMIENTO O APRENDIZAJE

- Al finalizar la **fase de entrenamiento** de una red neuronal artificial los valores de sus parámetros se han ajustado, por lo que se espera que la **red haya aprendido lo suficiente** para resolver otro problema similar satisfactoriamente,

- Una manera de **medir el éxito o fracaso de red neuronal artificial** es **observar el error total** por barrida que se va obteniendo durante el aprendizaje. Este error se obtiene al comparar los resultados obtenidos por la red y la respuesta ideal esperada.

- En teoría, **mientras más barridas ejecute la red mejor entrenada se encontrará la red neuronal**. Sin embargo, ciertas topologías no ofrecen este comportamiento ya que su porcentaje de error puede disminuir en la i-ésima barrida y aumentar en la barrida i +1.

Tareas de Aprendizaje

- **Aproximación**
- **Asociación**
  - **Autoasociativa**
  - **Heteroasociativa**
- **Clasificación**
- **Predicción**
- **Control** planta:u(t),y(t) modelo:r(t),d(t) lim|d(t)-y(t)|=0
- **Filtraje**

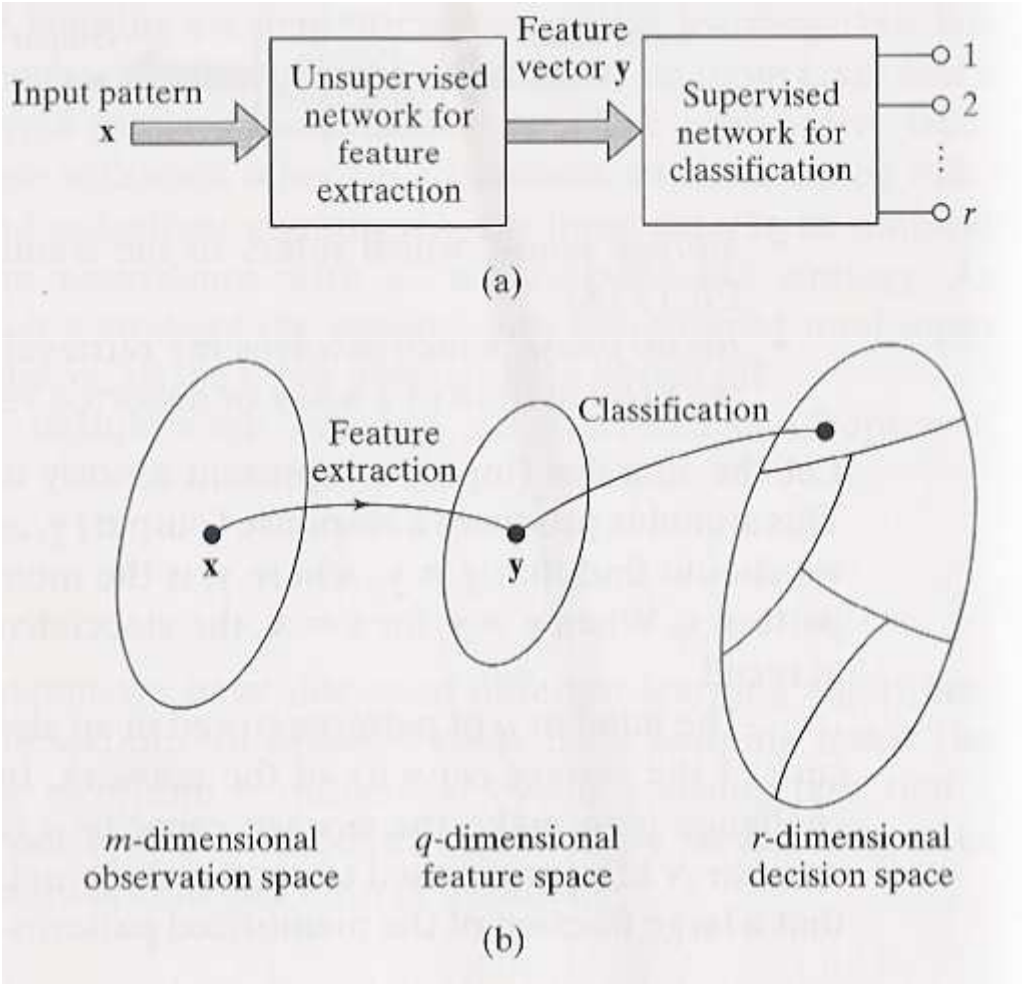
Tareas de Aprendizaje
Asociación de Patrones

- Memoria Asociativa
- Almacenar Patrones
- Patrones con ruido
- Recordar patrones

$x_k \rightarrow y_k$, k=1, 2, ... , q

- Autoasociativo: $x_k = y_k$
- Heteroasociativo: $x_k \neq y_k$
- Meta de Diseño:
  Recordar + patrones
  con - neuronas

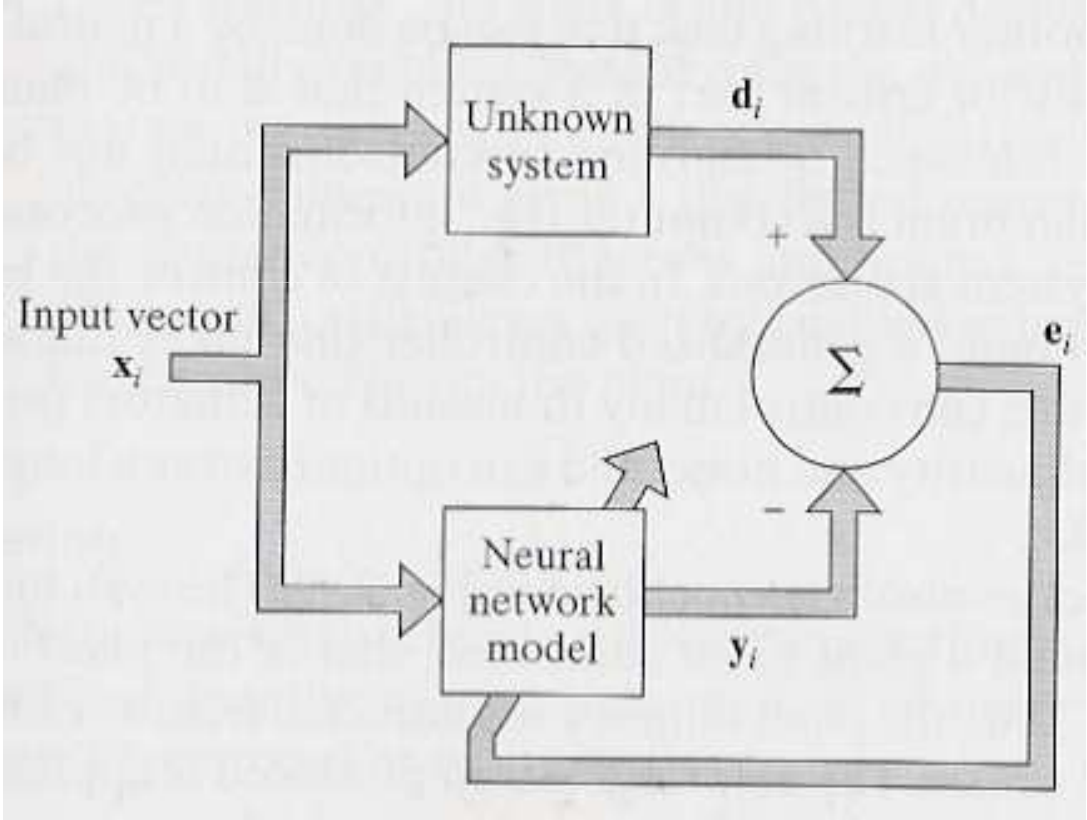
Tareas de Aprendizaje
Reconocimiento de Patrones



Clasificación

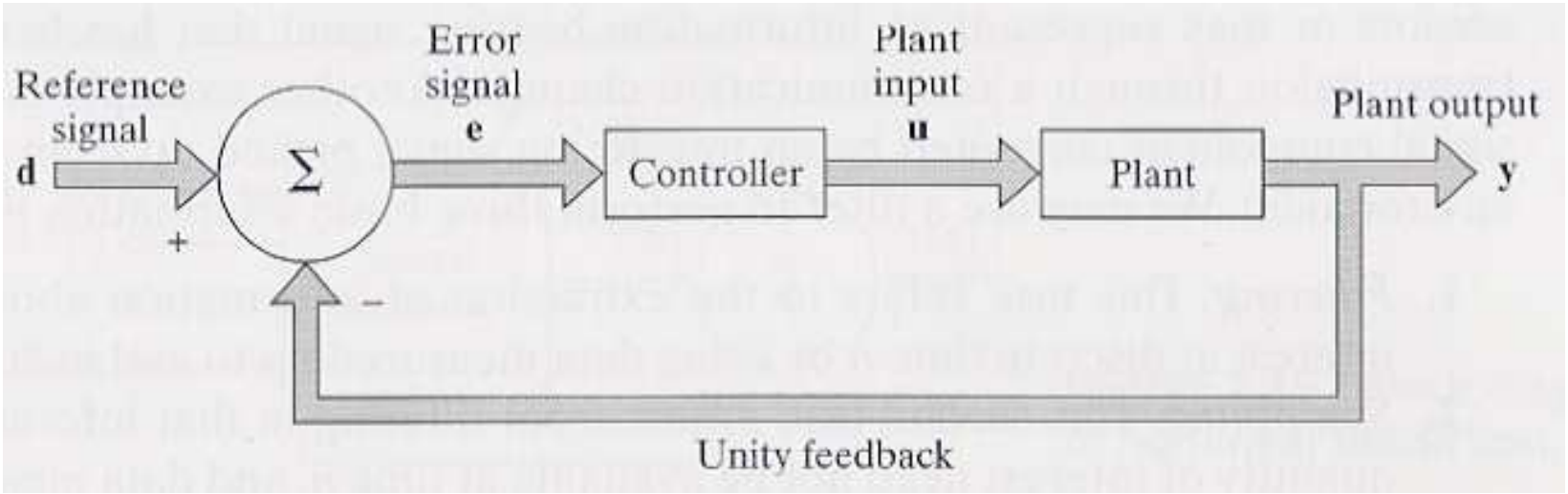
Tareas de Aprendizaje
Aproximación Funcional

- d = f(x)
- T = {(xi, di)},
  i=1, ..., N
- F aproximación de
  f
- Meta:
  $|F(x) - f(x)| < \varepsilon, \forall x$



Tareas de Aprendizaje
Control

- d = f(x)
- T = {(xi, di)}, i=1, ..., N
- F aproximación de f
- Meta: $|F(x) - f(x)| < \varepsilon, \forall x$

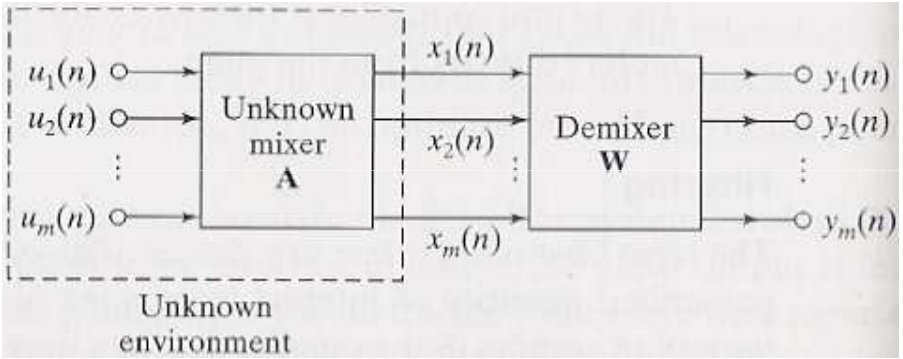
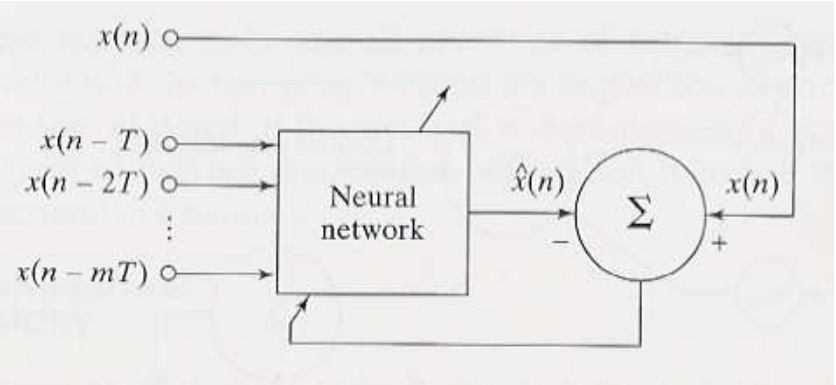


Tareas de Aprendizaje
Filtrado

- Filtrado. Extracción de información de una variable en el tiempo discreto n, usando mediciones hasta n.
- Smoothing. Filtrado con retardo.
- Predicción. Derivar información acerca del futuro de una variable, usando información pasada.

Tareas de Aprendizaje
Filtrado

- Problema del Cocktail.



- Predicción no Lineal.