APRENDIZAJE COMPETITIVO SIMPLE

Las unidades de salida compiten para activarse (winner-takes-all)

Objetivo: *agrupar (cluster) o categorizar los datos de entrada ↔ encontrar clases a partir de las correlaciones de los datos* no rotulados *de entrada.*

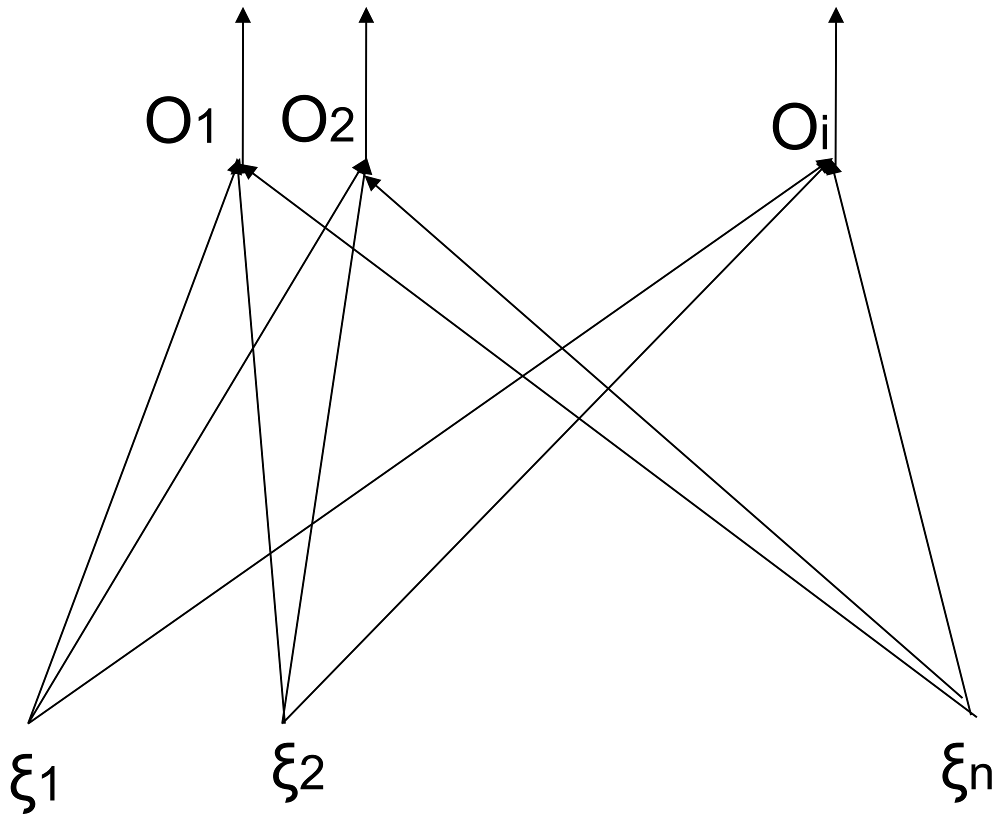
Utilidad principal: *Cuantización vectorial → compresión: cada entrada es reemplazada por el índice de la unidad de salida que activa.*

Diversas aplicaciones: aproximación de funciones
procesamiento de imágenes
análisis estadístico
optimización combinatoria

Concepto relacionado: *feature mapping* (mapeo de características)

Diferencia:
Aprendizaje competitivo simple: sin relación geométrica entre las neuronas.
Feature mapping: unidades espacialmente organizadas (→ métrica)

Evolución histórica común:
- *Rosenblatt*
- *von der Malsburg:* precedente de los SOM
- *Fukushima*: antecedente de las redes convolucionales
- *Grossberg*
- *Kohonen*: SOM
- *Rumelhart* y *Zipser*



Entradas y salidas binarias en principio (0/1)

Llamamos *unidad ganadora* a la neurona i* que maximice la entrada neta

$$h_i = \sum_j w_{ij}\xi_j = \mathbf{w}_i \cdot \boldsymbol{\xi}$$

→ se cumple $\quad \mathbf{w}_{i^*} . \boldsymbol{\xi} \geq \mathbf{w}_i . \boldsymbol{\xi} \quad$ para todo i

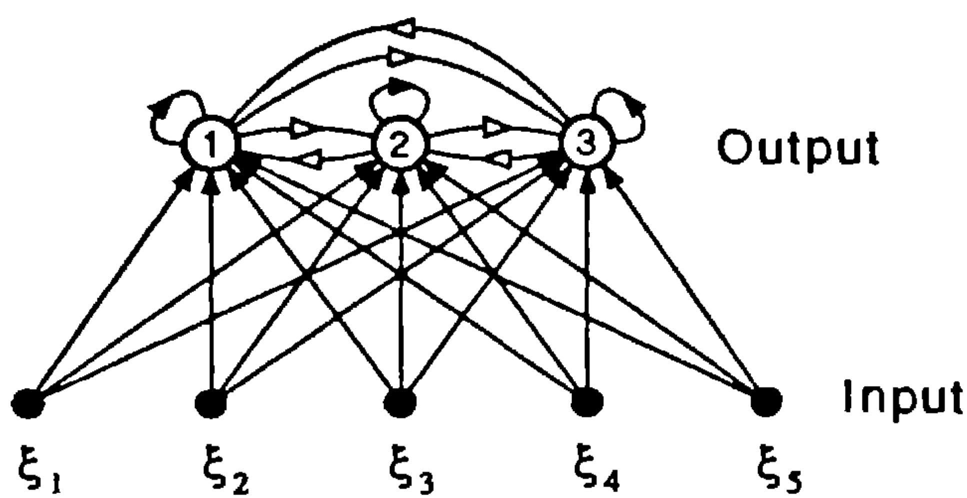
O, equivalente si los pesos están normalizados (e.g. $\|w_i\| = 1$)

$$\left| \mathbf{w}_{i^*} - \boldsymbol{\xi} \right| \leq \left| \mathbf{w}_i - \boldsymbol{\xi} \right| \qquad \text{para todo i}$$

Implementación del ganador:

- *En una simulación computacional*: eligiendo el máximo h_i.

- *En una red real*: - conexiones laterales inhibitorias
  - conexiones autoexcitatorias
  - Elección correcta de las conexiones laterales y de la función de activación



Actualización de los pesos

- Regla ingenua: $\Delta w_{i^*j} = \eta \xi_j^\mu$ (unidad ganadora)

→ *los pesos crecerán sin cota. Unica neurona ganadora.*

- Más eficiente: $$\Delta w_{i^*j} = \eta' \left(\frac{\xi_j^\mu}{\sum_j \xi_j^\mu} - w_{i^*j} \right)$$

o bien $$\Delta w_{i^*j} = \eta(\xi_j^\mu - w_{i^*j})$$

para entradas ya normalizadas

Regla completa $$\boxed{\Delta w_{ij} = \eta O_i(\xi_j^\mu - w_{ij})}$$

(recordar que O_i ***= 0*** salvo O_{i^*} ***= 1***)

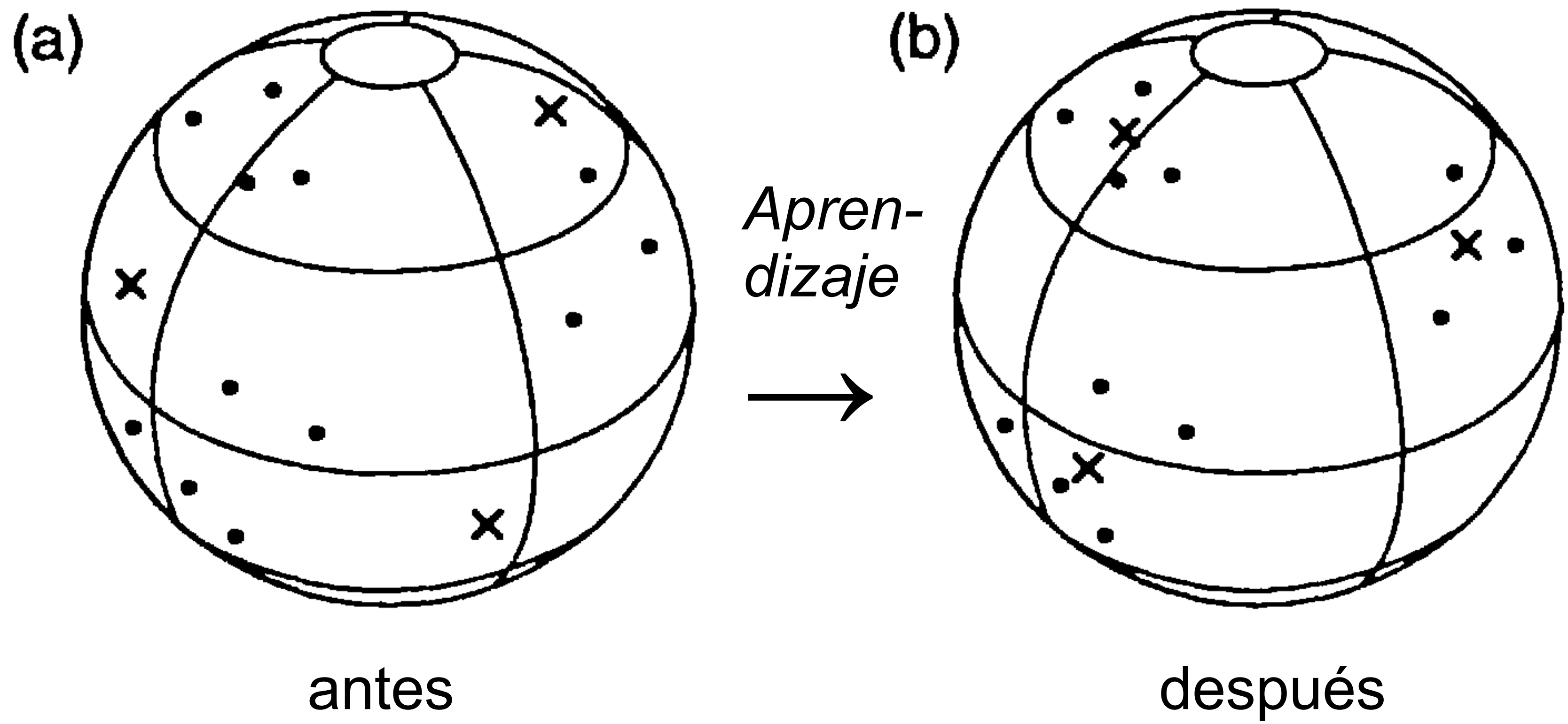
Formación de Clusters

1- Valores inciales pequeños y al azar para los w_i

2- Elegir un patrón de entrada ξ y presentarlo a la red (puede ser a partir de una distribución $P(\xi)$).

3- Hallar la i* ganadora y actualizar los w_{i^*j}
($\rightarrow$ acerca el w_{i^*} a la entrada ξ actual e incrementa la probabilidad de que esa unidad ganadora lo sea en el futuro ante el mismo ξ o parecido)

4- ir a 2-



Puntos: patrones de entrada
Cruces: neuronas

<u>Importante</u>: para que se alcance el equilibrio, la muestra debe ser representativa (= todas las posibles entradas ser elegidas con igual probabilidad)

Funciones de costo

La solución óptima de un problema de clustering no está, en general, claramente definida → distintos criterios.

Función de costo standard:

$$E\{w_{ij}\} = \frac{1}{2}\sum_{ij\mu} M_i^\mu (\xi_j^\mu - w_{ij})^2 = \frac{1}{2}\sum_\mu |\boldsymbol{\xi}^\mu - \mathbf{w}_{i\bullet}|^2.$$

asociada a la regla de aprendizaje presentada.
La *matriz de membresía* M:

$$M_i^\mu = \begin{cases} 1 & \text{if } i = i^*(\mu)\,; \\ 0 & \text{otherwise.} \end{cases}$$

especifica si un patrón *μ* activa una neurona i → M cambiará durante el entrenamiento (i* depende de la entrada μ y de los w_{ij}).
Gradiente descendente:

$$\langle \Delta w_{ij} \rangle = -\eta \frac{\partial E}{\partial w_{ij}} = \eta \sum_\mu M_i^\mu (\xi_j^\mu - w_{ij})$$

Con *η* adecuado → *convergencia en promedio a un mínimo local.*

El problema de las unidades muertas

Neuronas con w lejos de cualquier entrada → nunca ganarán.

<u>Soluciones posibles</u>:

- Inicializar los pesos con muestras de la distribución de entrada

- Actualizar también los pesos de las unidades perdedoras, pero con un η menor.

- Si hay una *métrica*, actualizar en forma decreciente según cercanía al ganador (esencia de los *mapas autoorganizados*).

- Otras estrategias (referencias en el Hertz).

Desventajas

con respecto a representaciones distribuidas:

- *Una neurona de salida por cada clase* → N unidades ↔ N clases en lugar de 2^N

- *No robustas ante falla o degradación*: la pérdida de una neurona de salida conlleva la de toda la clase que representa

- *No pueden representar conocimiento jerárquico* (categorías dentro de categorías)