

# Aprendizaje no supervisado

Departamento de Informática  
Universidad Carlos III de Madrid  
Avda. de la Universidad, 30. 28911 Leganés (Madrid)

# Aprendizaje no supervisado

- Introducción
- Modelo básico
- Mapas autoorganizados de Kohonen
- Teoría de la resonancia adaptativa

# Introducción

- Características principales
  - No necesitan de un profesor externo
  - La red descubre en los datos de entrada y por si sola
    - Características
    - Regularidades
    - Correlaciones
    - Categorías
  - Sólo consigue resultados útiles si existe algún tipo de redundancia

# Introducción

- Características principales (II)
  - Suelen requerir tiempos de entrenamiento más cortos que las supervisadas
  - Modelos más cercanos a estructuras neurobiológicas
  - Arquitectura simple. Habitualmente son:
    - Una sola capa
    - Feed-forward
  - Tipos fundamentales
    - Kohonen (SOM)
    - Grossberg (ART)

# Introducción

- Problemas abordables con estas redes
  - Familiaridad:
    - Similitud entre una entrada y un valor tipo
  - Análisis de componentes principales
    - Componentes que caracterizan al conjunto de datos
  - Agrupamiento
    - Determinar la existencia de clases y clasificar los patrones
  - Prototipado
    - Obtener prototipo asociado a la clase del patrón de entrada

# Introducción

- Problemas abordables con estas redes (II)
  - Codificación
    - Reducción de dimensionalidad
  - Extracción y relación de características
    - Análisis de la topología
- Estas categorías no son disjuntas
  - Ej: codificación mediante componentes principales

# Modelo básico

- Regla de Hebb
  - Aportación fundamental a las RNA
  - Pretende contribuir a decidir cómo y en qué grado modificar las conexiones con aprendizaje no supervisado
  - Parte de la observación del comportamiento de células cerebrales

# Modelo básico

## ■ Regla de Hebb

“Cuando el axón de una célula A está lo suficientemente cerca para excitar a una célula B, y toma parte repetidamente en el proceso de disparo de dicha célula, se produce algún tipo de cambio metabólico en una de las células (o en las dos), que hace que la eficacia con la que A disparaba a B se vea incrementada”

- Esto viene a significar que los pesos que unen a dos neuronas aumentan si se activan a la vez y disminuyen si se activan en distinto momento



# Modelo básico

- Regla de Hebb
  - Abundan las propuestas de implementación para esta regla

$$\Delta w_{ij} = a_i \cdot a_j$$

$$\Delta w_{ij} = \Delta a_i \cdot \Delta a_j$$

$$\Delta w_{ij} = \mu (a_i - \bar{a}_i) (a_j - \bar{a}_j)$$

# Modelo básico

- Interacción lateral
  - Imitar el comportamiento del cerebro
  - Tratar de crear sistemas artificiales con propiedades observadas en los modelos biológicos
  - ¿Cómo conseguir que, en neuronas idénticas, el orden sea relevante?
  - ¿Cómo relacionar la topología con las propiedades de las entradas?
  - Modelo de Interacción Lateral:
    - Se conecta cada neurona con un conjunto próximo de ellas
    - Sólo interactúan si están conectadas

# Modelo básico

## ■ Interacción lateral (arquitectura)

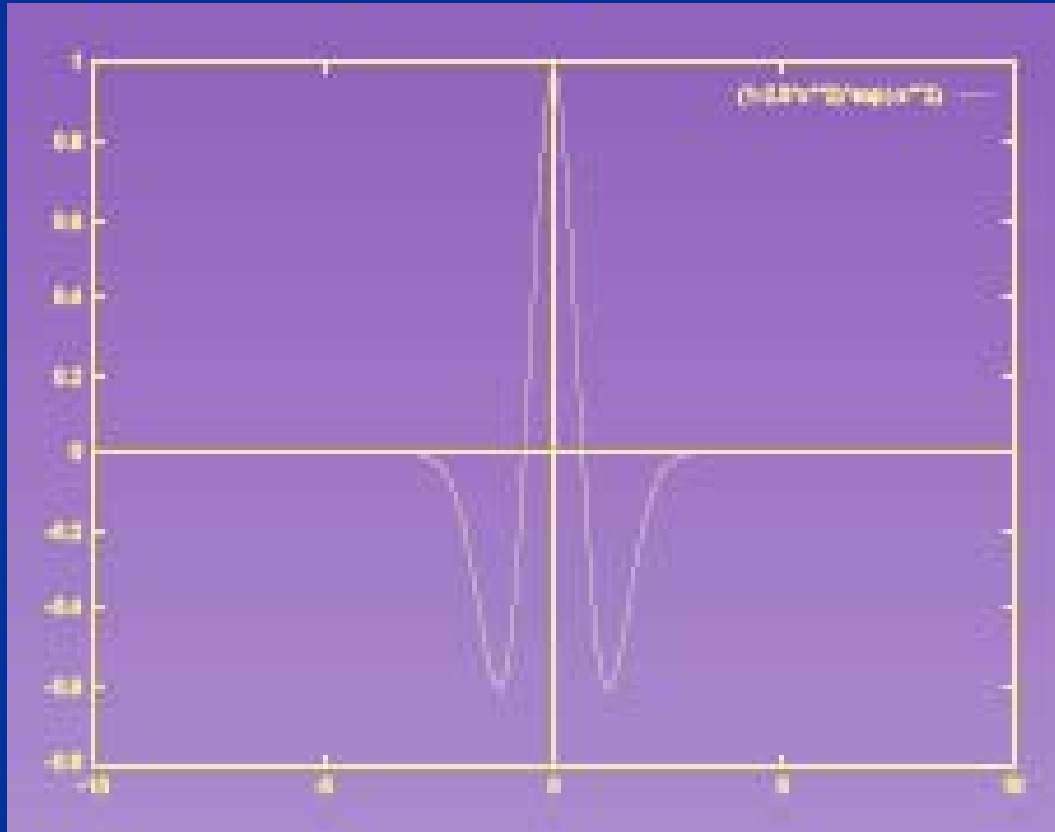


# Modelo básico

- Interacción lateral (arquitectura)
  - Cada célula conectada con las de su entorno cercano de forma excitatoria
  - Cada célula conectada con las de su entorno menos cercano de forma inhibitoria
  - No hay conectividad entre células alejadas
  - La intensidad de la conexión decrece con la distancia
    - Cuando una célula está activa influye en la actividad de las que la rodean
    - El orden de las células es significativo

# Modelo básico

- Interacción lateral (arquitectura)



# Modelo básico

- Interacción lateral (ecuaciones)
  - Su formulación en el campo discreto sería:

$$\mu_i(t) = \sigma\left[\theta_i(t) + \sum_{k=-1}^1 \gamma_k \mu_{i+k}(t-1)\right]$$

- Donde:
  - $\mu_i(t)$  es la salida de la unidad  $i$ .
  - $\sigma[x]$  es una función no lineal de tipo sigmoide.
  - $\gamma_k$  es un coeficiente extraído de muestrear en el tiempo la señal de la función anterior, de forma que para  $-1 < k < 1$ , tenga un valor positivo, y negativo en el resto.
  - $\theta_i(t)$  es el umbral de la célula  $i$ .

# Modelo básico

## ■ Características

- Aprendizaje no supervisado utilizado en modelos de RNA
- Objetivo: categorizar los datos de entrada
- Datos parecidos han de ser clasificados como pertenecientes a la misma categoría
- Hay una capa de clasificación compuesta por tantas neuronas como categorías pueda haber en los datos
- Cada categoría representada por un prototipo
- El prototipo recoge las características similares de los datos de su categoría
- En la capa de clasificación, cada célula se corresponde con un prototipo

# Modelo básico

## ■ Características (II)

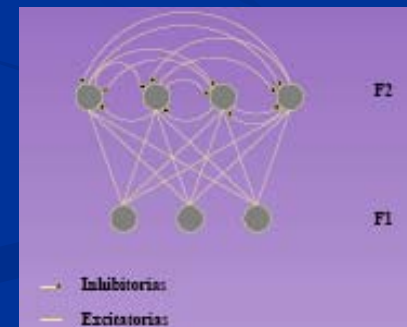
- El sistema debe relacionar cada célula (prototipo) con los datos de entrada que representa
- Debe agrupar los datos de entrada en categorías, por razones de similitud, y asignar a cada categoría un prototipo
- El prototipo sería utilizado para clasificar datos nuevos y desconocidos.



# Modelo básico

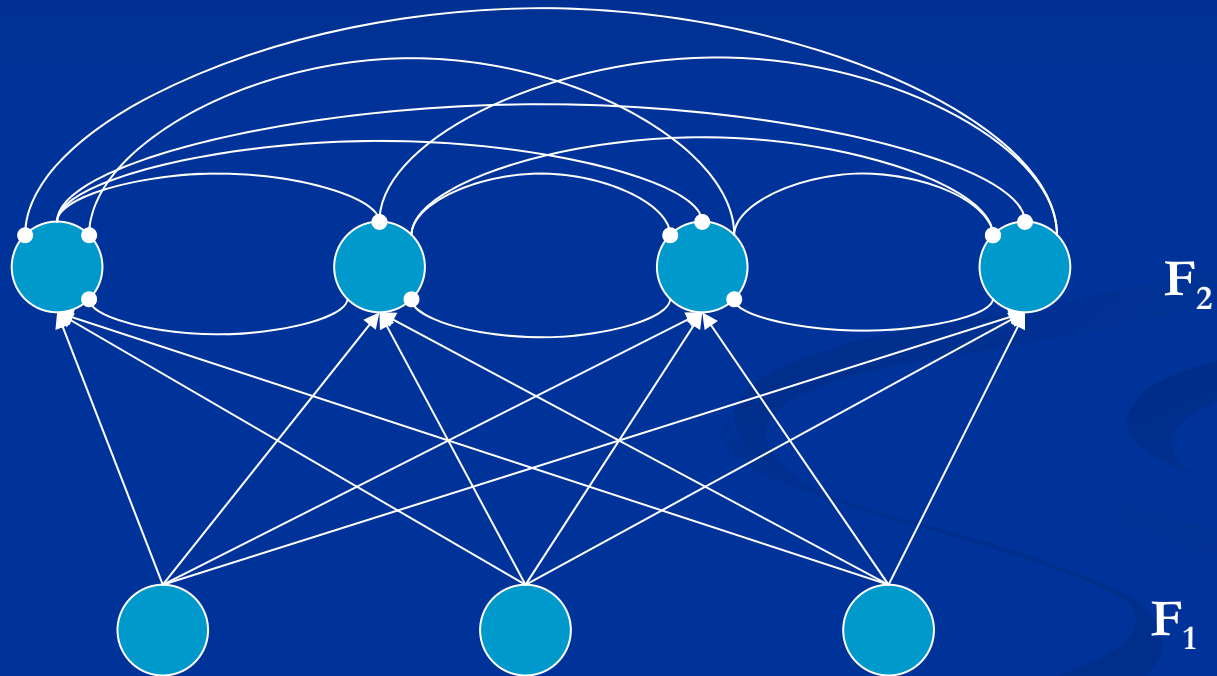
## ■ Arquitectura

- Dos capas: F1 de entrada y F2 de competición
- Cada célula de la capa F1 está conectada con todas las células de la capa F2 a través de conexiones ponderadas variables
- F2 tiene conexiones laterales inhibitorias entre todas las células de su capa, excepto con ella misma, en que la conexión es excitatoria:



# Modelo básico

## ■ Arquitectura



—● Conexión inhibitoria

→ Conexión excitatoria

# Modelo básico

## ■ Procedimiento

- 1. Se recibe el estímulo en F1.
- 2. Se propaga la señal hasta F2 y se calcula el valor de excitación para cada célula de F2.
- 3. Se inhiben las conexiones entre la capa F1 y la F2.
  - Se propaga la señal por la capa F2, calculándose los nuevos valores de excitación de las células.
  - Cuando sólo haya una célula (célula ganadora) con un valor de salida mayor que cero ir al paso 5.
- 4. Ir al paso 3.
- 5. Restablecer las conexiones entre las capas F1 y F2.
  - Calcular los nuevos valores para los pesos de las conexiones entre la capa F1 y la célula ganadora en el paso 3.

# Modelo básico

## ■ Procedimiento (II)

- La capa F2 se estabiliza cuando:

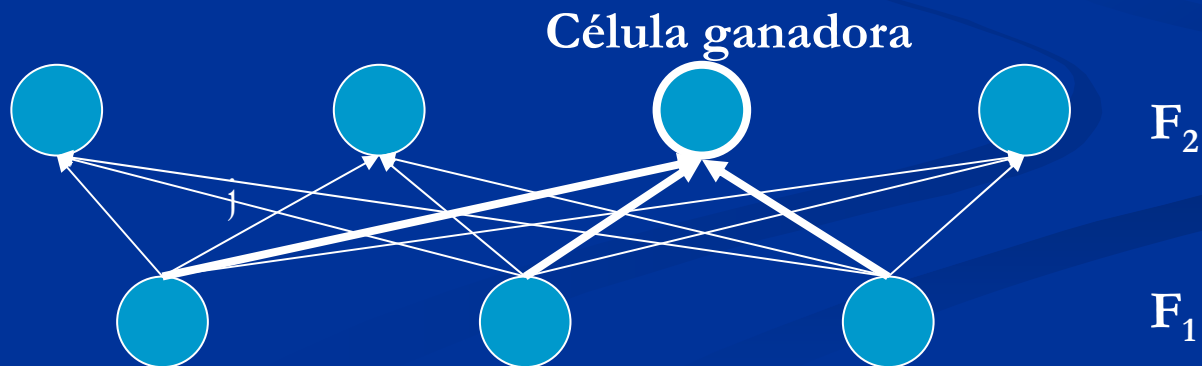
$$\forall i, i \neq j \quad \mu(F2_i) = 0 \quad \mu(F2_j) > 0$$

- Las células de la capa F2 compiten por la entrada
- En F2 los pesos son fijos:

$$\forall i, \sum_j^N W_{ij} = 1 \quad \forall i, j \quad W_{ij} = W_{ji}$$

# Modelo básico

- Aprendizaje
  - La célula ganadora representa al prototipo que es asignado al dato de entrada
  - Las conexiones entre la capa F1, la de la entrada, y la célula ganadora son reforzadas
  - Es por esto que se llama “el que gana se lo lleva todo” (winner takes all)



# Modelo básico

- Ecuaciones de aprendizaje

$$\frac{dW_{ij}}{dt} = (-W_{ij} + \theta_i) \times g(b_j)$$

Donde:

$$g(b_j) = \begin{cases} 1 & \text{si } j \text{ ganadora} \\ 0 & \text{en caso contrario} \end{cases}$$

$$\theta_i = \frac{I_i}{\sum_{j=1}^M I_j}$$

Siendo  $I_i$  el  $i$ -ésimo elemento del ejemplo de entrada

# Modelo básico

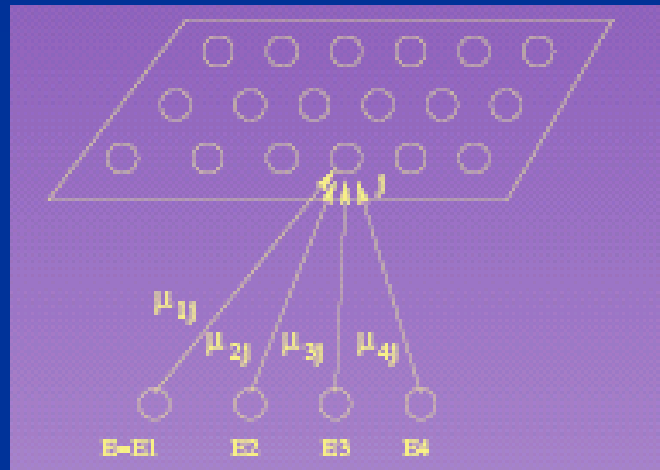
## ■ Problemas

- No es capaz de producir una codificación estable ante entradas arbitrarias
- Tiene una capacidad limitada de codificación
- Es necesario establecer a priori el número de categorías de la clasificación

# Mapas autoorganizados de Kohonen

- Arquitectura

- Permiten realizar clasificación no supervisada



- RNA de dos capas

- Entrada
- Competición



# Mapas autoorganizados de Kohonen

## ■ Ecuaciones

- Si hay  $n$  atributos la entrada será:  $e = \{e_1, \dots, e_n\}$
- Si hay  $m$  células en la capa de competición, entonces los pesos serán:

$$\begin{pmatrix} \mu_{11} & \mu_{12} & \dots & \mu_{1m} \\ \mu_{21} & \mu_{22} & \dots & \mu_{2m} \\ \dots & \dots & \dots & \dots \\ \mu_{n1} & \mu_{n2} & \dots & \mu_{nm} \end{pmatrix}$$

Donde  $\mu_{ij}$  es el peso de la conexión entre la célula  $i$ -ésima de la capa de entrada  $i$  la  $j$ -ésima de la capa de competición

# Mapas autoorganizados de Kohonen

## ■ Ecuaciones de distancia

- Si se hace referencia a la célula  $j$  de la capa de competición se tendrá el vector:

$$\mu_j = \{\mu_{1j}, \mu_{2j}, \dots, \mu_{ij}, \dots, \mu_{nj}\}$$

- Este vector tiene la misma dimensión que la entrada  $e$
- Se puede comparar:  $d_{ij} = d(e, \mu_j)$
- El modelo de Kohonen calcula una salida en la c.c. ( $\tau_j$ ) a través de la distancia  $\tau_j = d(e, \mu_j)$

# Mapas autoorganizados de Kohonen

- Ecuaciones de distancia
  - Se pueden elegir diversas distancias.
    - Ej si producto escalar

$$d(x,y) = \sum_{i=1}^n x_i y_i$$

- En este caso, la salida de la capa de competición será la habitual en los modelos supervisados (suma ponderada de las entradas por los pesos de las conexiones)

$$\tau_j = \sum_{i=1}^n e_i \mu_{ij}$$

# Mapas autoorganizados de Kohonen

- La función más utilizada es la distancia euclídea

$$d(\mathbf{x}, \mathbf{y}) = \left( \sum_{i=1}^n (x_i - y_i)^2 \right)^{1/2}$$

- Si se elige la distancia euclídea el modelo quedaría como sigue:

$$\tau_j = \left( \sum_{i=1}^n (e_i - \mu_{ij})^2 \right)^{1/2}$$

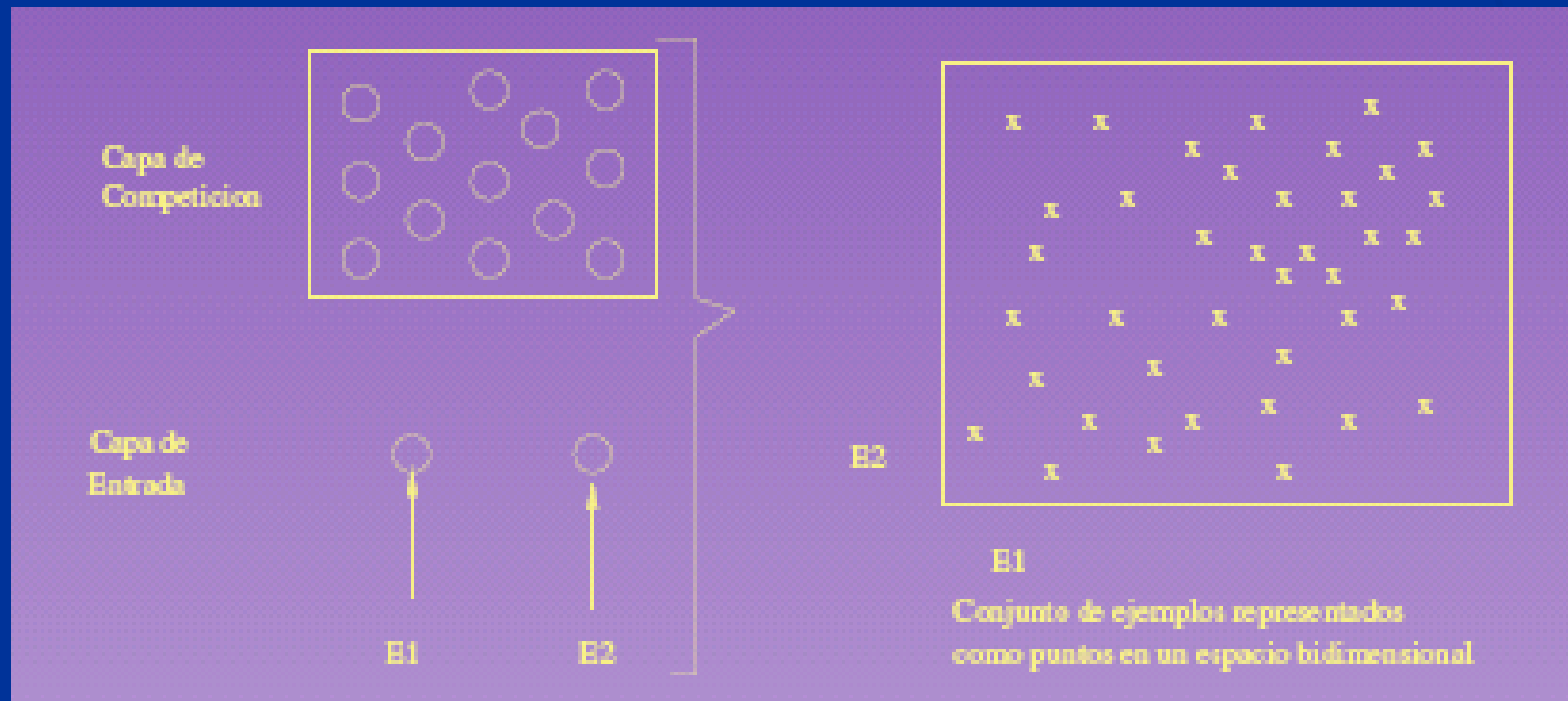
# Mapas autoorganizados de Kohonen

## ■ Funcionamiento

- Se recibe el vector de entrada
- Se propaga por las conexiones hasta llegar a la capa de competición
- Cada célula de la c.c. produce una salida al comparar la entrada con sus pesos
- Se selecciona aquella que produzca una salida más pequeña (célula ganadora)

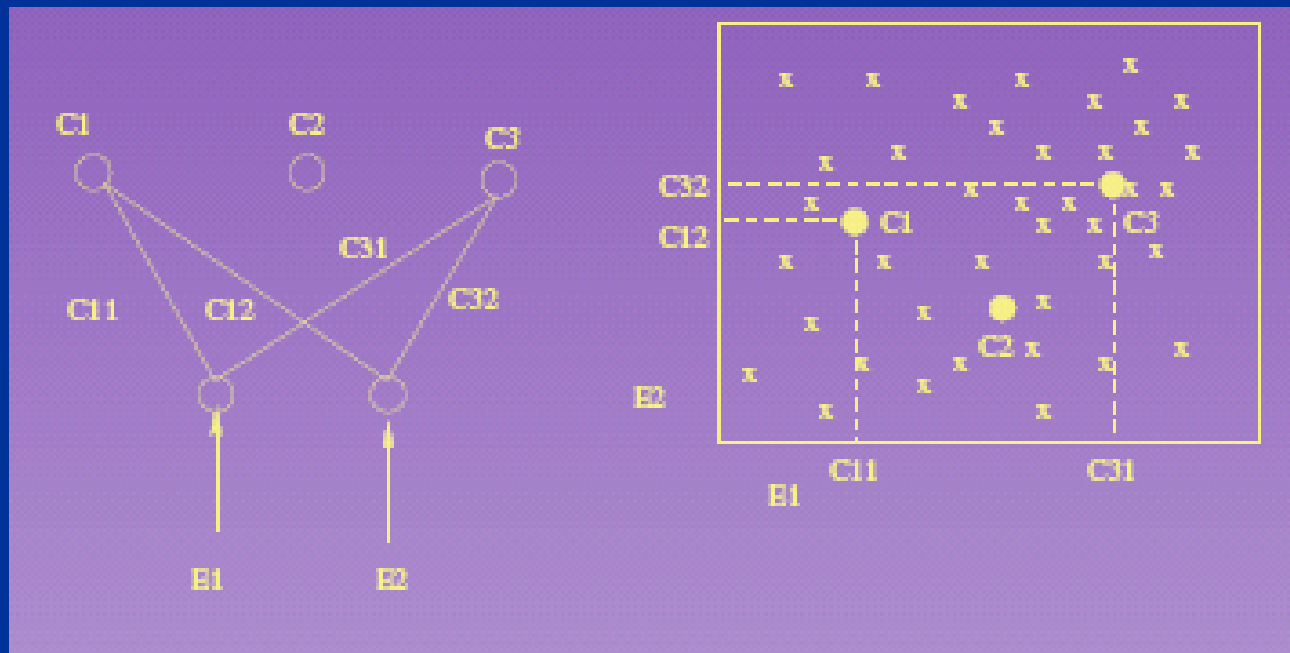
# Mapas autoorganizados de Kohonen

- Representación geométrica
  - Los ejemplos y los prototipos (células de la c.c.) se pueden representar como puntos en un espacio n dimensional



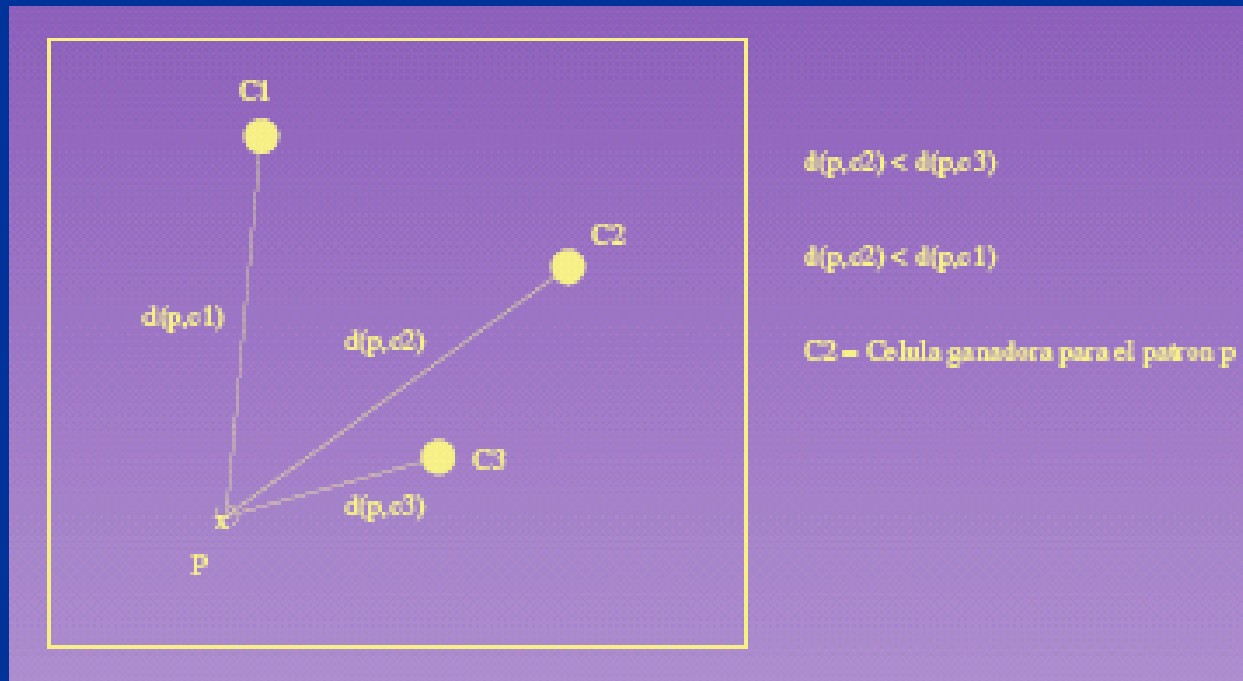
# Mapas autoorganizados de Kohonen

- Representación geométrica
  - Las coordenadas de los prototipos son los pesos de las células de la c.c.



# Mapas autoorganizados de Kohonen

- Representación geométrica ganadoras
  - Las coordenadas de los prototipos son los pesos de las células de la c.c.





# Mapas autoorganizados de Kohonen

## ■ Aprendizaje

- Aprendizaje Hebbiano, el incremento del valor de su peso es proporcional al valor de activación de las células que conecta ( $\tau_j \times e_i$ )

$$\frac{d\mu_{ij}}{dt} = \alpha(t) \tau_j(t) (e_i(t) - \mu_{ij}(t))$$

- El ganador se lo lleva todo

$$\tau_j = \begin{cases} 1 & \text{si } j \text{ ganadora} \\ 0 & \text{en caso contrario} \end{cases}$$

$$\frac{d\mu_{ij}}{dt} = \tau_j [\alpha(t) (e_i(t) - \mu_{ij}(t))]$$

# Mapas autoorganizados de Kohonen

## ■ Aprendizaje

- El valor de  $\alpha$  es decrementado una cantidad constante pequeña  $\beta$  tras cada ciclo completo de todos los patrones de aprendizaje:  $\alpha(t+1) = \alpha(t) - \beta$
- Mediante la asignación del parámetro  $\beta$  se pueden determinar el número total de ciclos de aprendizaje que sería:

$$\text{Iteraciones} = \frac{\alpha(t)}{\beta}$$

- El valor de  $\alpha$  es decrementado siguiendo un esquema logarítmico

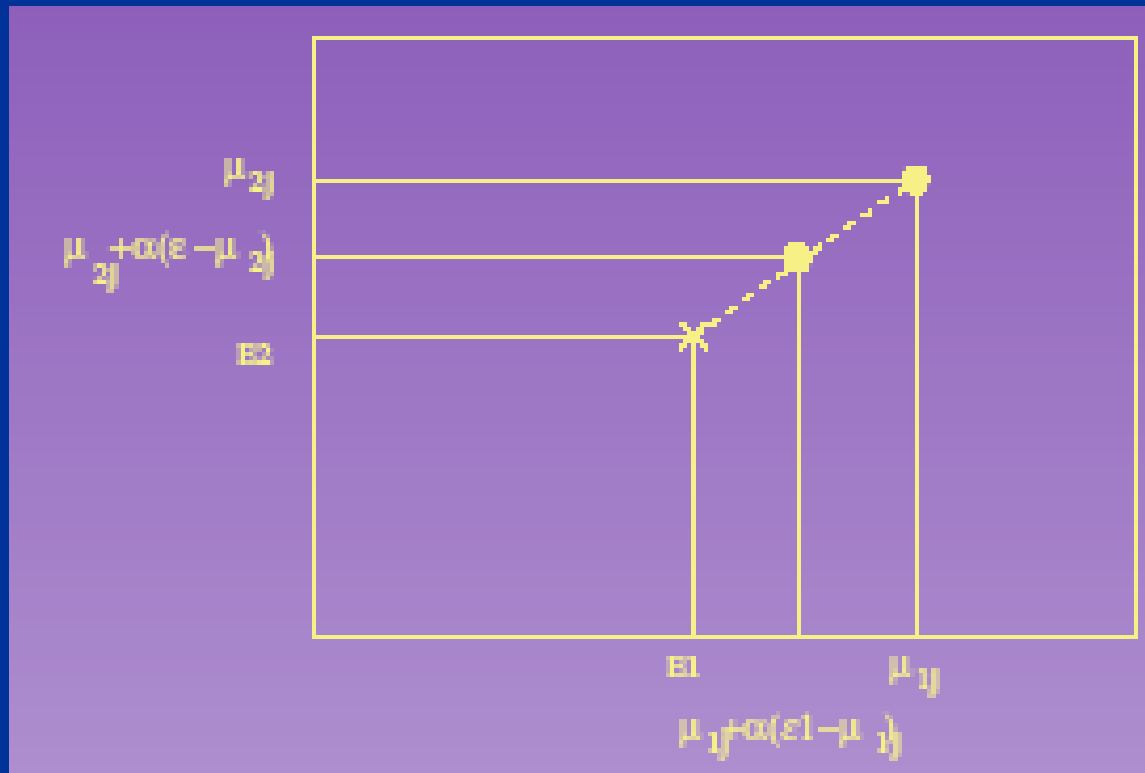
Siguiendo con la analogía  $\Delta_{ij} = \alpha(e_i - \mu_{ij})$

Donde  $\forall i=1, \dots, n$  donde  $C_j$  es la célula ganadora



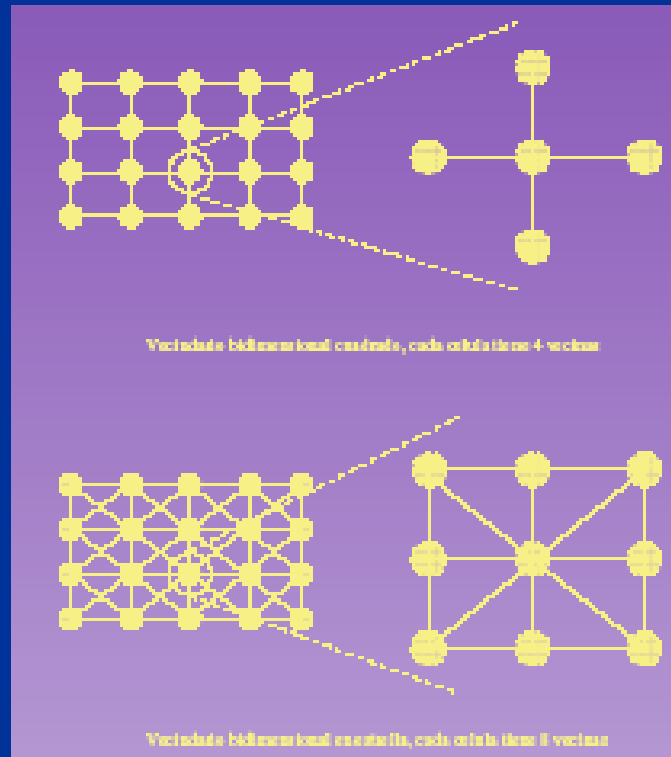
# Mapas autoorganizados de Kohonen

- Representación geométrica del aprendizaje
  - Se aproxima la célula al punto:



# Mapas autoorganizados de Kohonen

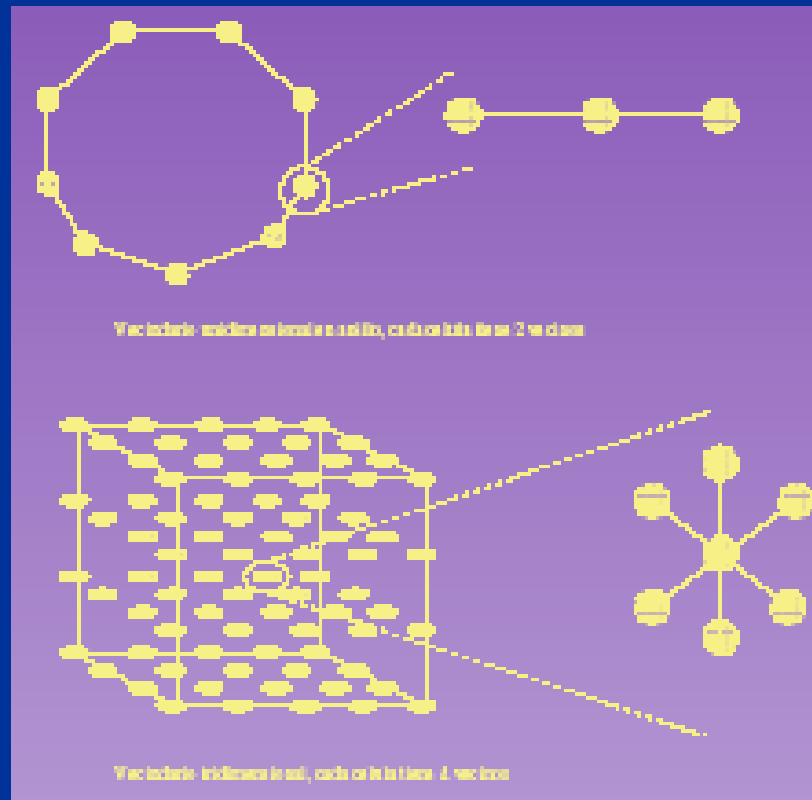
- Vecindario
  - Vecindarios bidimensionales



# Mapas autoorganizados de Kohonen

- Vecindario

- Vecindarios unidimensional y tridimensional



# Mapas autoorganizados de Kohonen

## ■ Ecuaciones de vecindario

- Se define una distancia de vecindario entre células  $d(c_i, c_j)$
- Se introduce el esquema en el aprendizaje

$$\frac{d\mu_{ij}}{dt} = \begin{cases} \frac{\alpha(t)}{d(c_i, c_j)} (e_i(t) - \mu_{ij}(t)) & \text{Si } c_i \text{ ganadora} \\ & \text{y } d(c_i, c_j) < \theta \\ 0 & \text{en caso contrario} \end{cases}$$

Donde  $\theta$  es el límite del vecindario

# Mapas autoorganizados de Kohonen

## ■ Método de Kohonen

1. Inicializar pesos.

Asignar a los pesos valores pequeños aleatorios.

2. Presentar una nueva entrada.

El conjunto de aprendizaje se presenta cíclicamente hasta llegar a la convergencia de la red. Actualizar  $\alpha$

3. Propagar el patrón de entrada hasta la capa de competición

Obtener los valores de salida de las células de dicha capa.

4. Seleccionar la célula C cuya salida sea mayor.

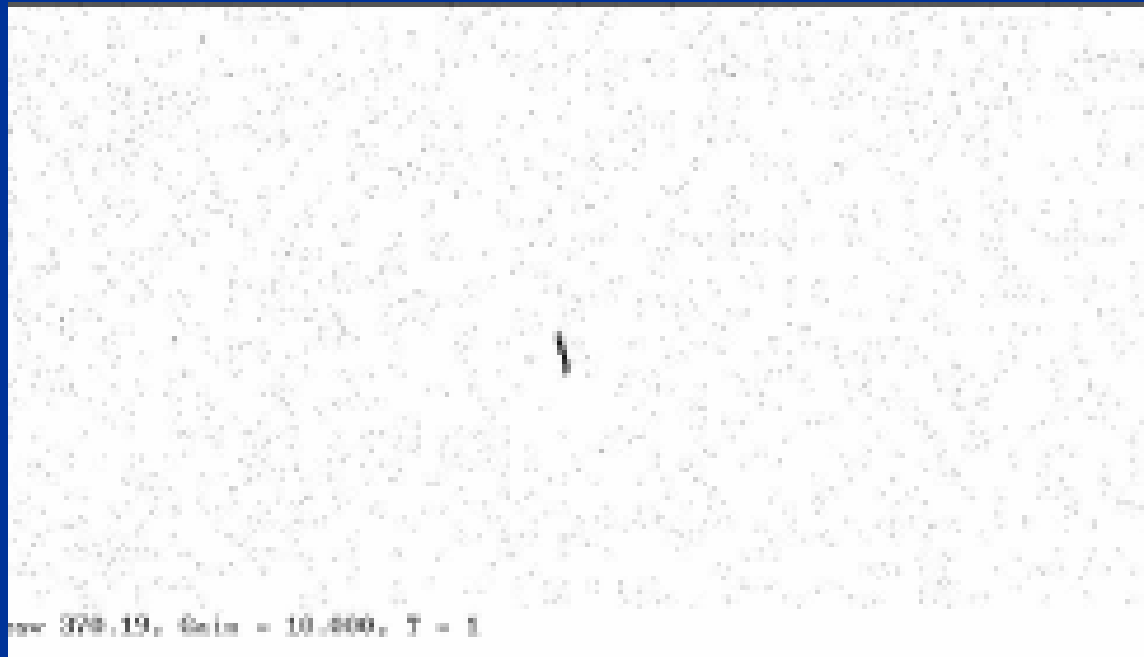
5. Actualizar las conexiones entre la capa de entrada y la célula C, así como las de su vecindad, según su grado de vecindad.

6. Si por encima de cierto umbral volver al paso 2, en caso contrario FIN.



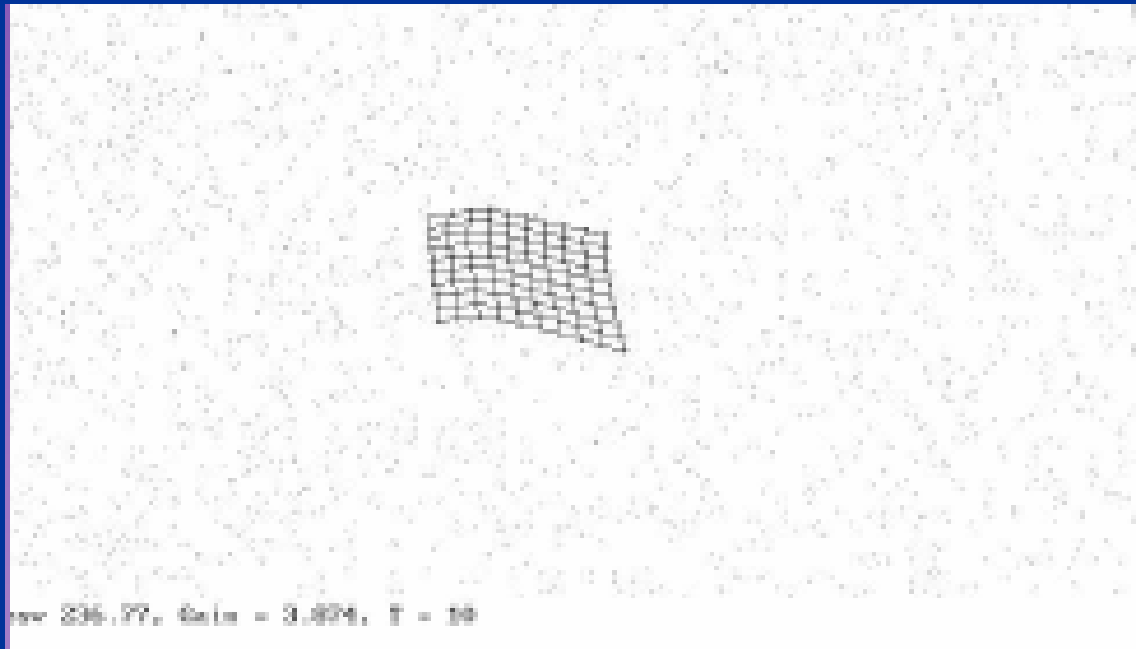
# Mapas autoorganizados de Kohonen

- Ejemplo: distribuciones uniformes
  - Distribución uniforme, iteración 1:



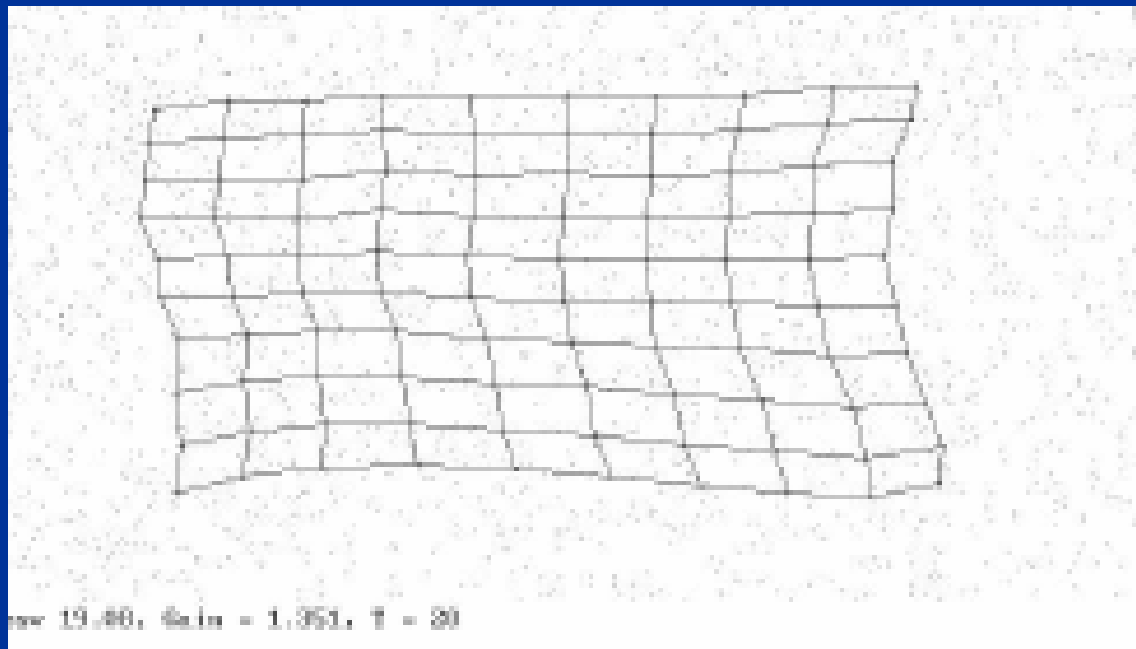
# Mapas autoorganizados de Kohonen

- Ejemplo: distribuciones uniformes
  - Distribución uniforme, iteración 10:



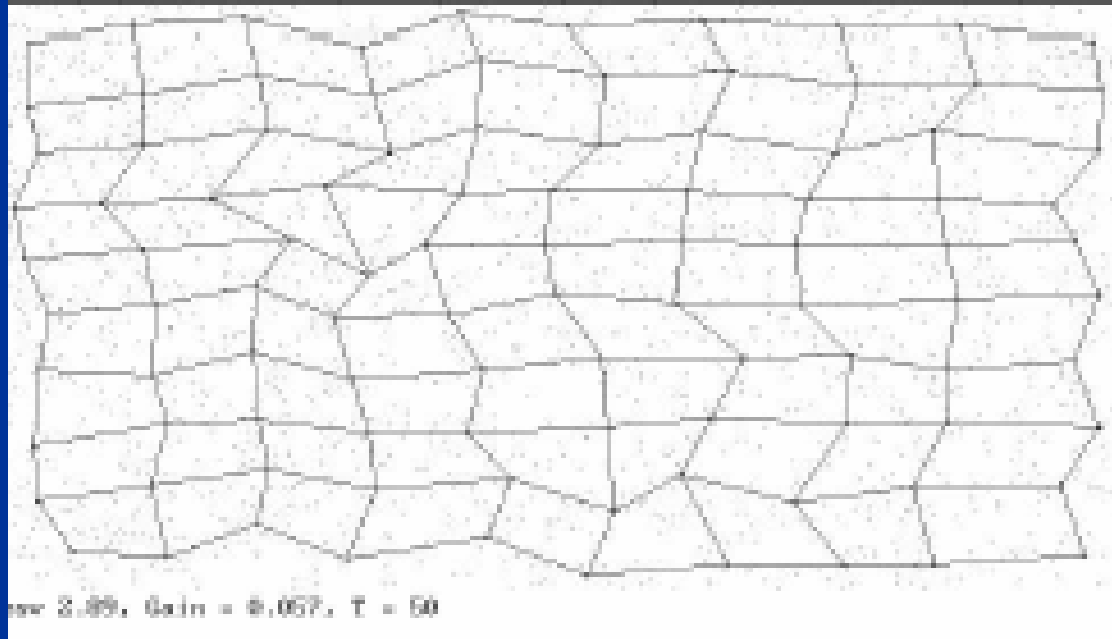
# Mapas autoorganizados de Kohonen

- Ejemplo: distribuciones uniformes
  - Distribución uniforme, iteración 20:



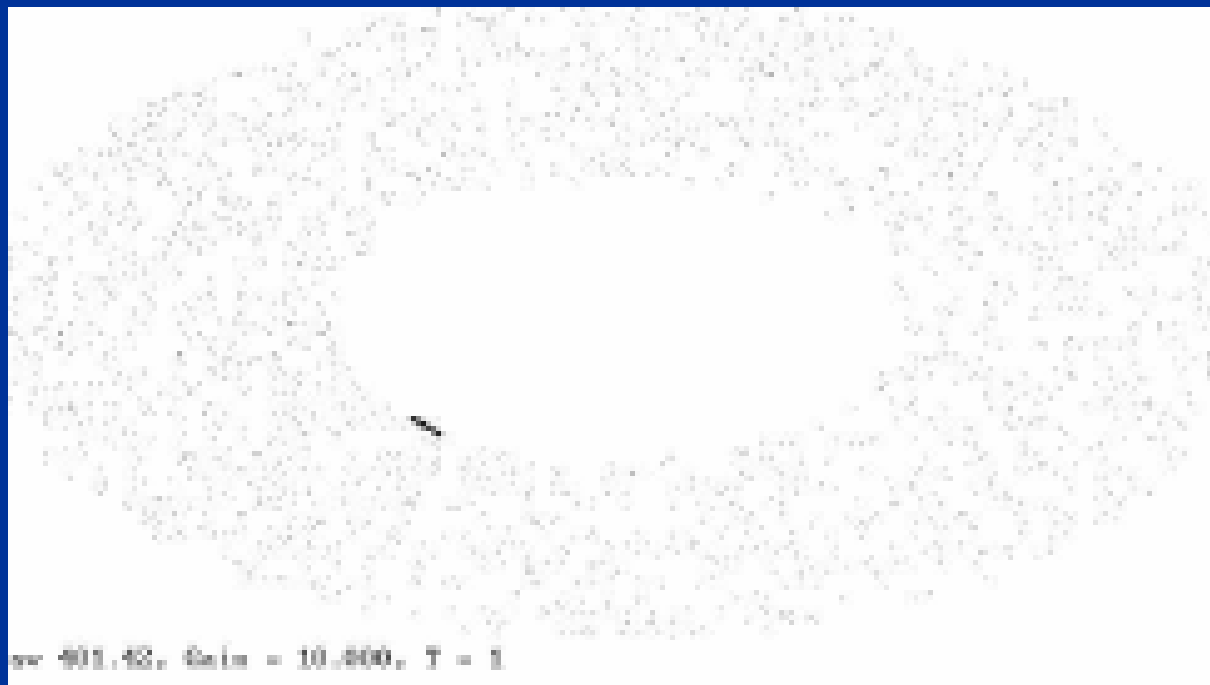
# Mapas autoorganizados de Kohonen

- Ejemplo: distribuciones uniformes
  - Distribución uniforme, iteración 50:



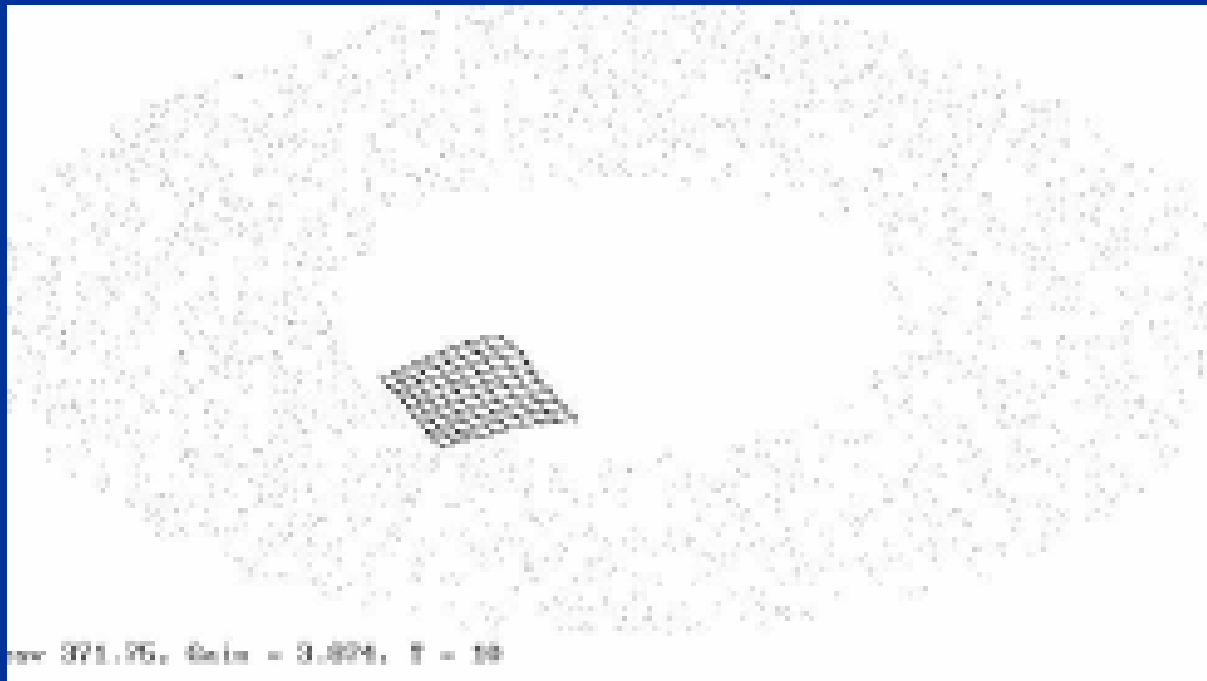
# Mapas autoorganizados de Kohonen

- Ejemplo: distribuciones toroidales
  - Distribución toroidal, iteración 1:



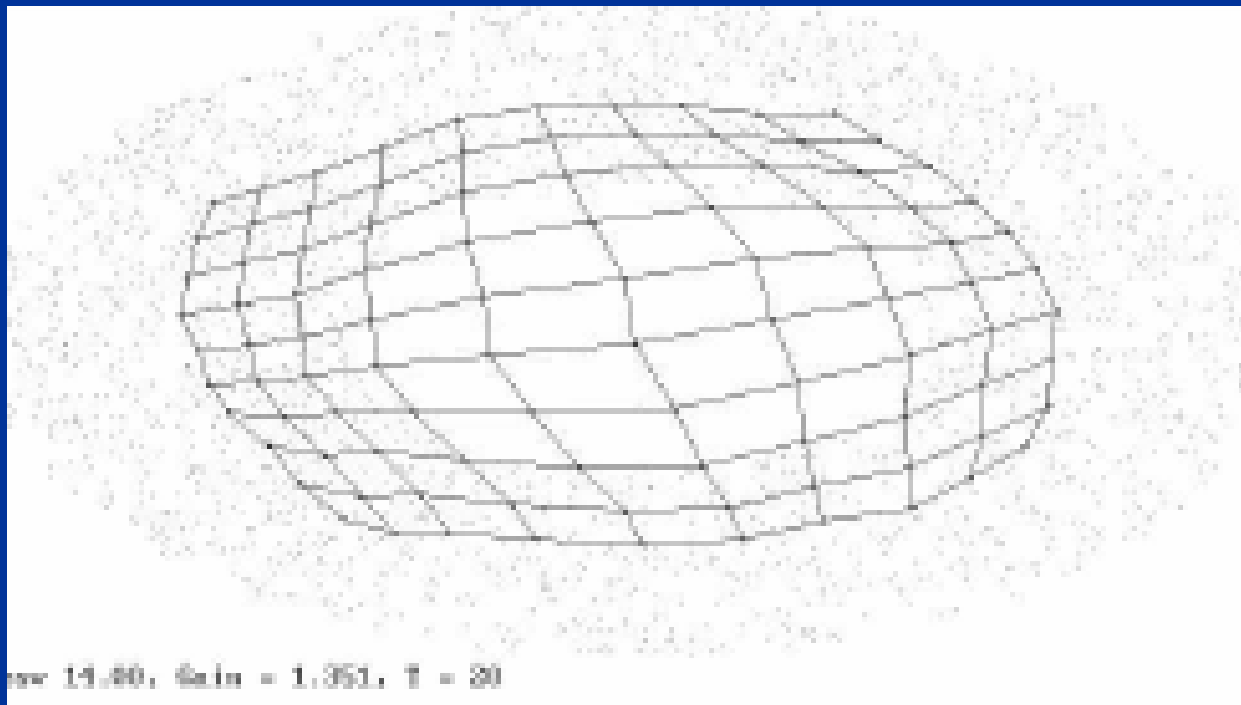
# Mapas autoorganizados de Kohonen

- Ejemplo: distribuciones toroidales
  - Distribución toroidal, iteración 10:



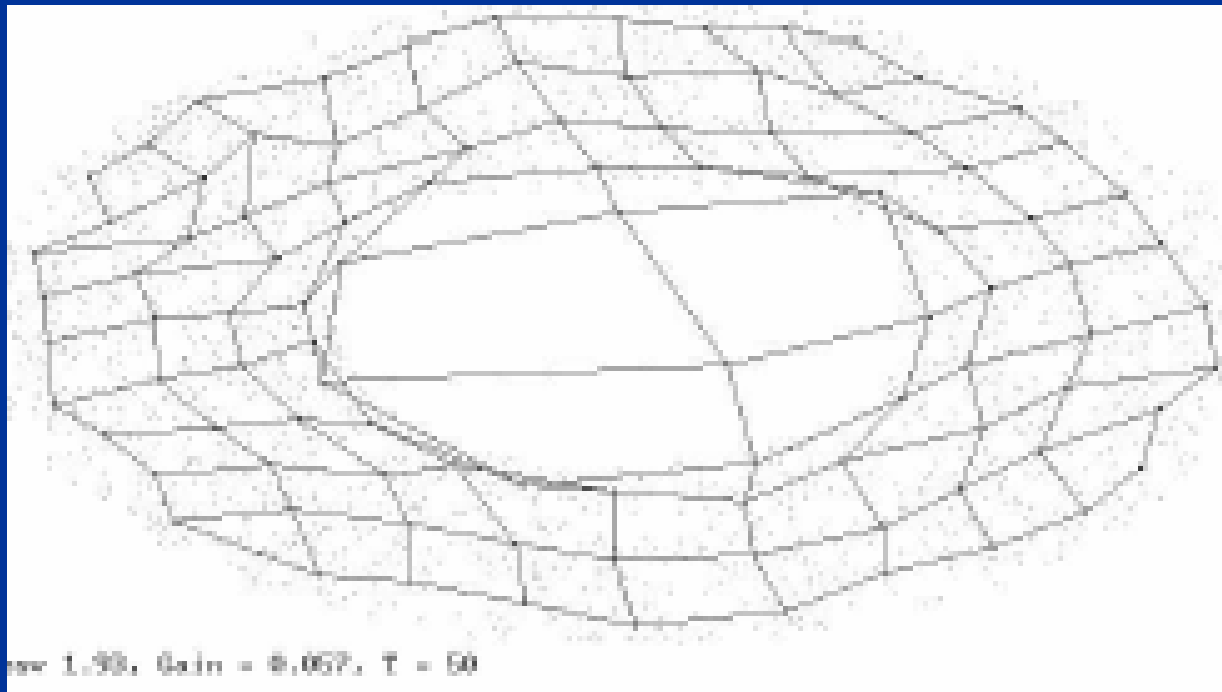
# Mapas autoorganizados de Kohonen

- Ejemplo: distribuciones toroidales
  - Distribución toroidal, iteración 20:



# Mapas autoorganizados de Kohonen

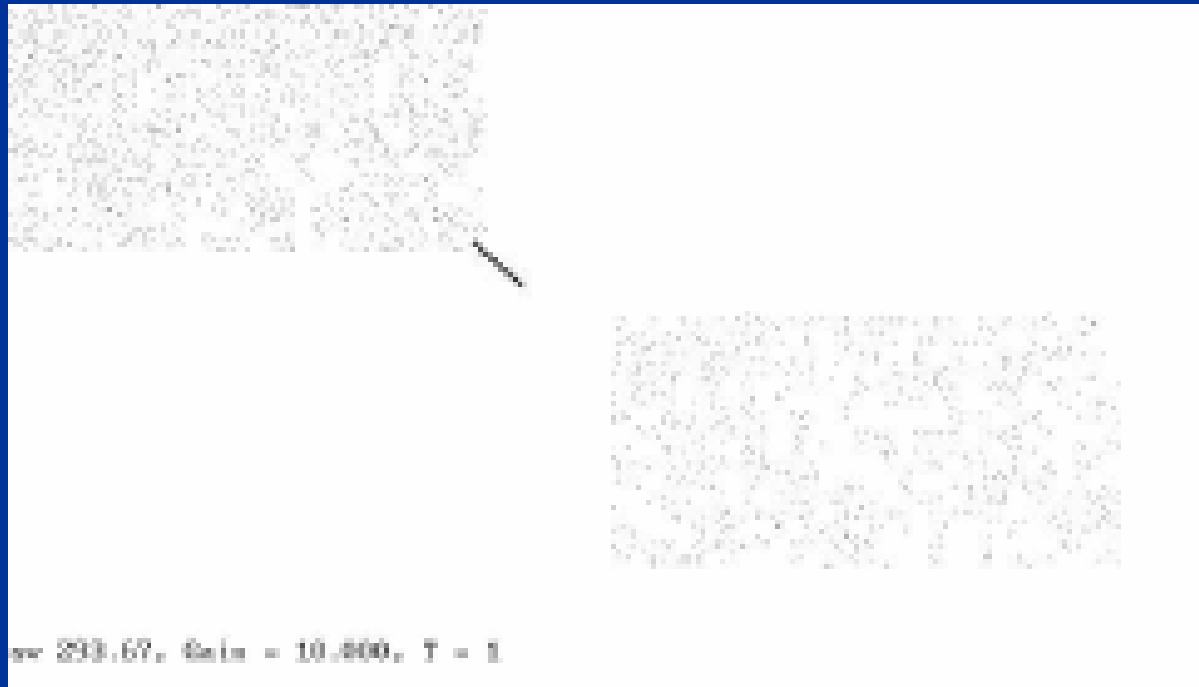
- Ejemplo: distribuciones toroidales
  - Distribución toroidal, iteración 50:





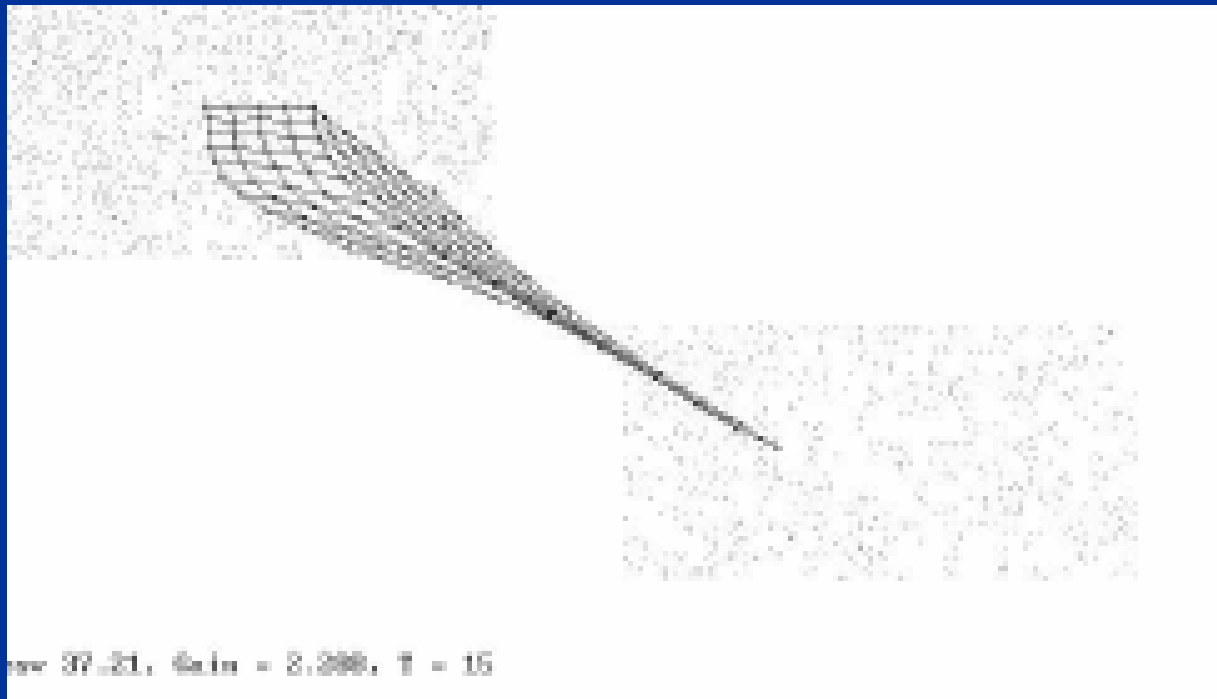
# Mapas autoorganizados de Kohonen

- Ejemplo: distribuciones biuniformes
  - Distribución biuniforme, iteración 1:



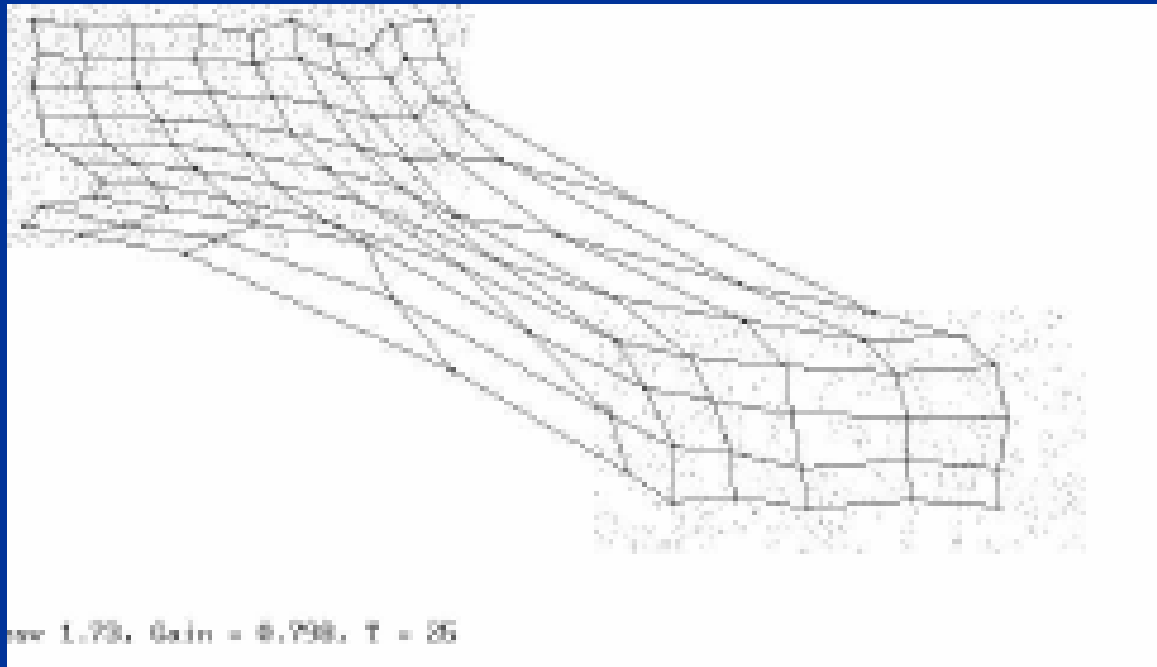
# Mapas autoorganizados de Kohonen

- Ejemplo: distribuciones biuniformes
  - Distribución biuniforme, iteración 15:



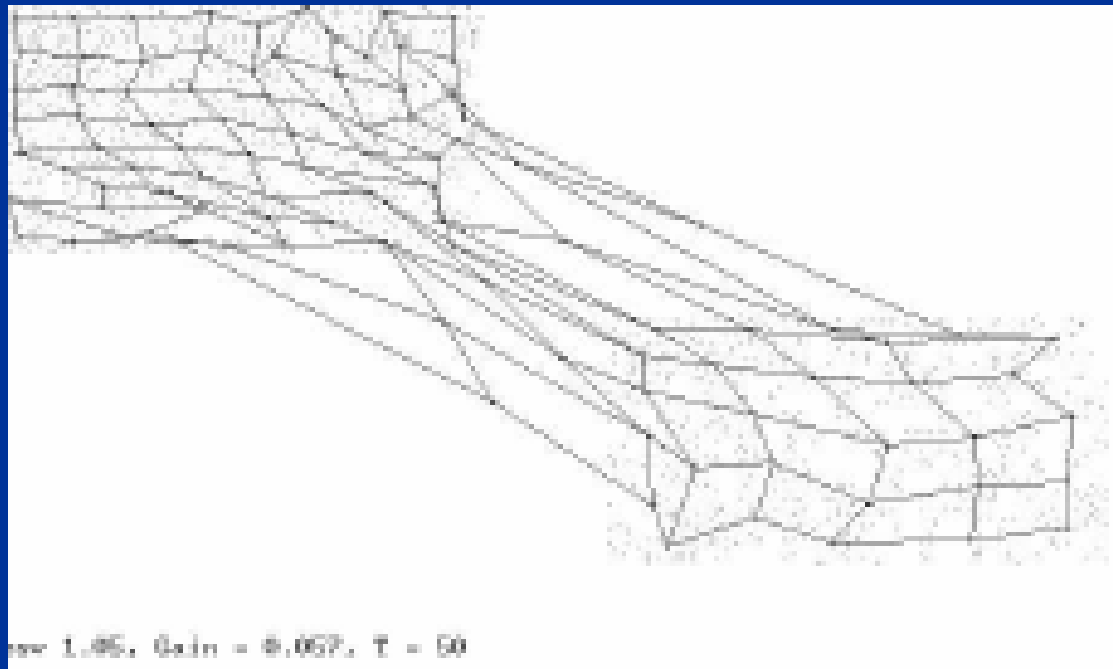
# Mapas autoorganizados de Kohonen

- Ejemplo: distribuciones biuniformes
  - Distribución biuniforme, iteración 25:



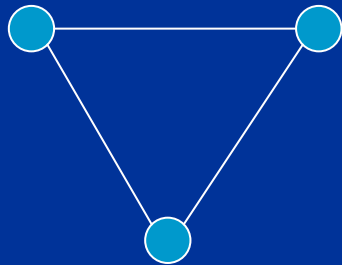
# Mapas autoorganizados de Kohonen

- Ejemplo: distribuciones biuniformes
  - Distribución biuniforme, iteración 50:

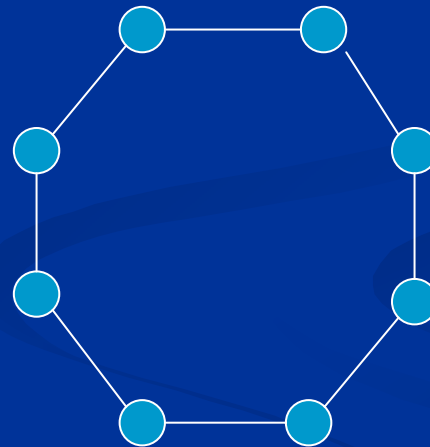


# Mapas autoorganizados de Kohonen

- Problema del viajante
  - Vecindario unidimensional para el problema del viajante:



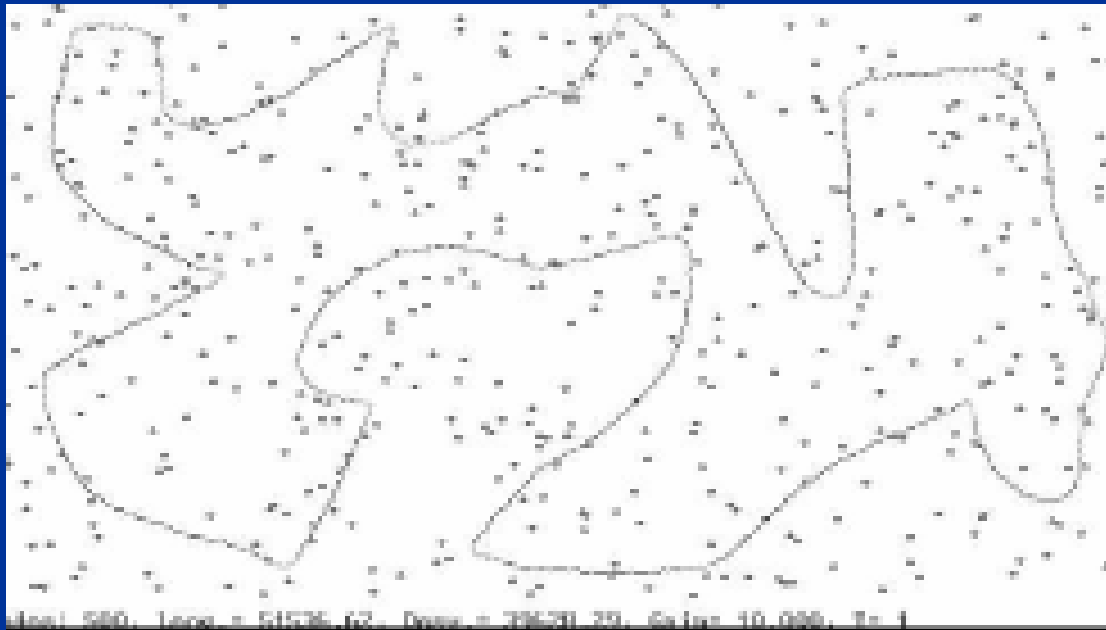
Vecindario para tres células



Vecindario para ocho células

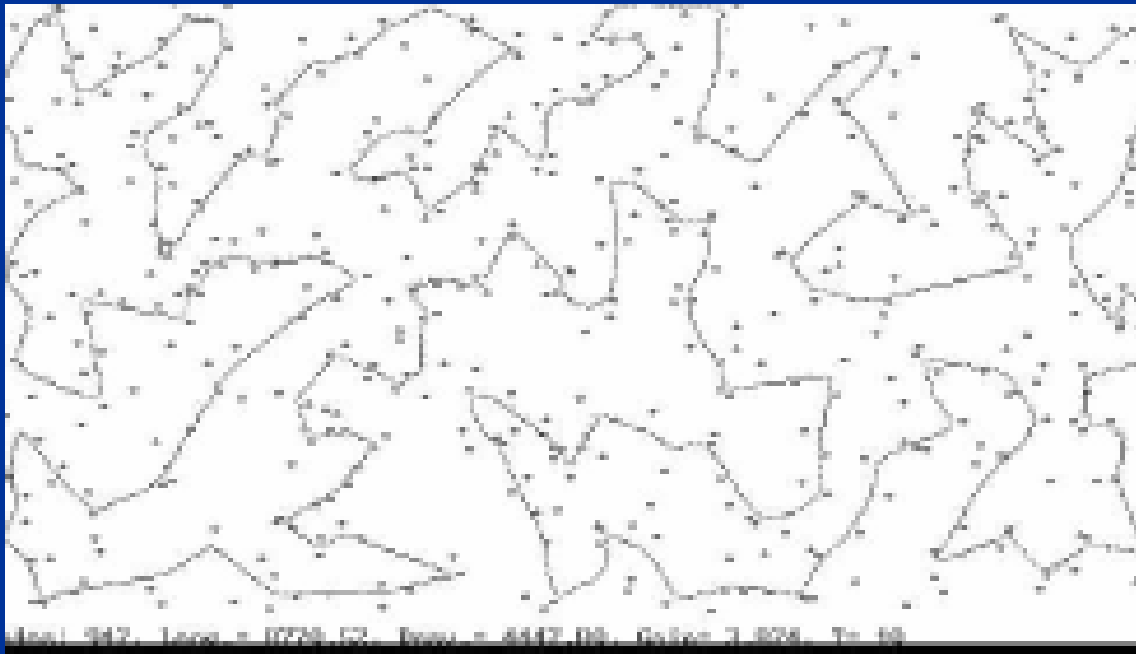
# Mapas autoorganizados de Kohonen

- Ejemplo: problema del viajante
  - Problema del viajante, iteración 1:



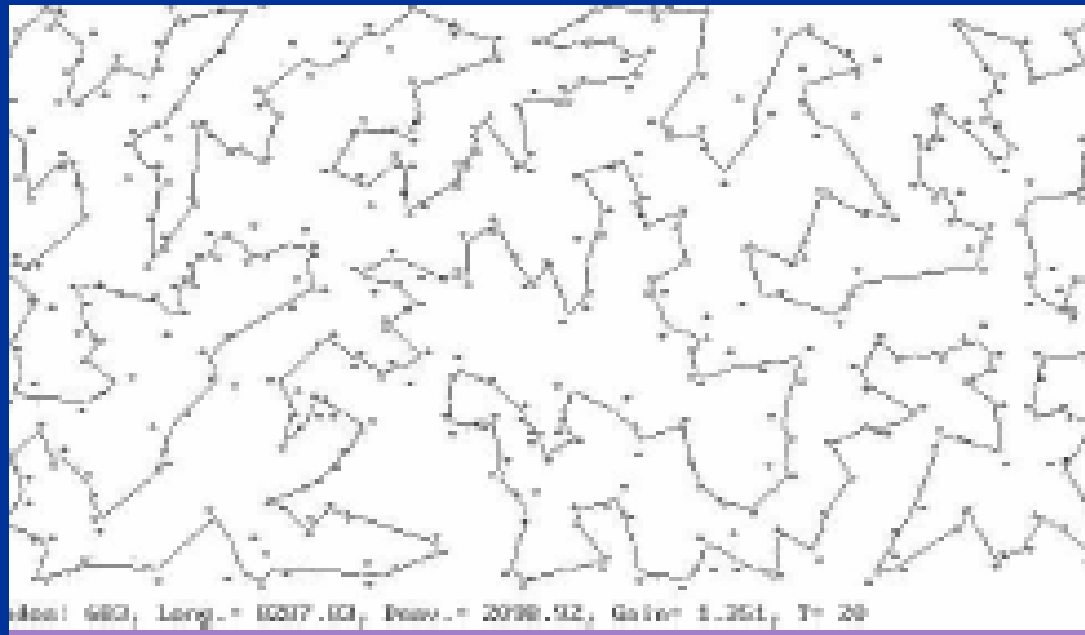
# Mapas autoorganizados de Kohonen

- Ejemplo: problema del viajante
  - Problema del viajante, iteración 10:



# Mapas autoorganizados de Kohonen

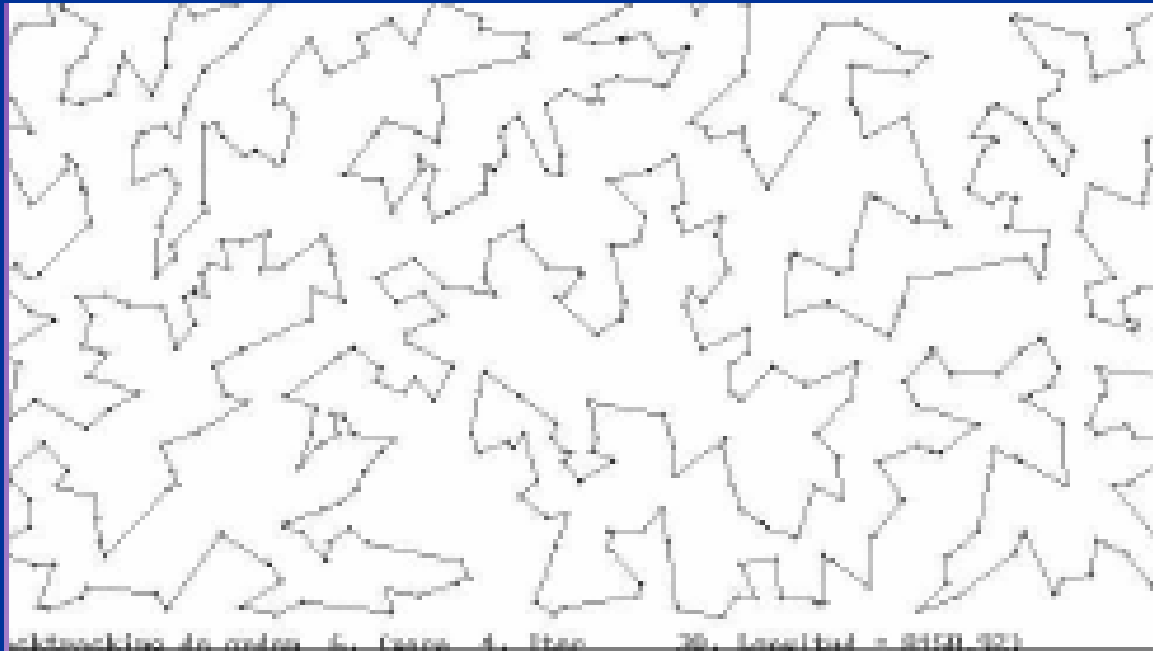
- Ejemplo: problema del viajante
  - Problema del viajante, iteración 20:





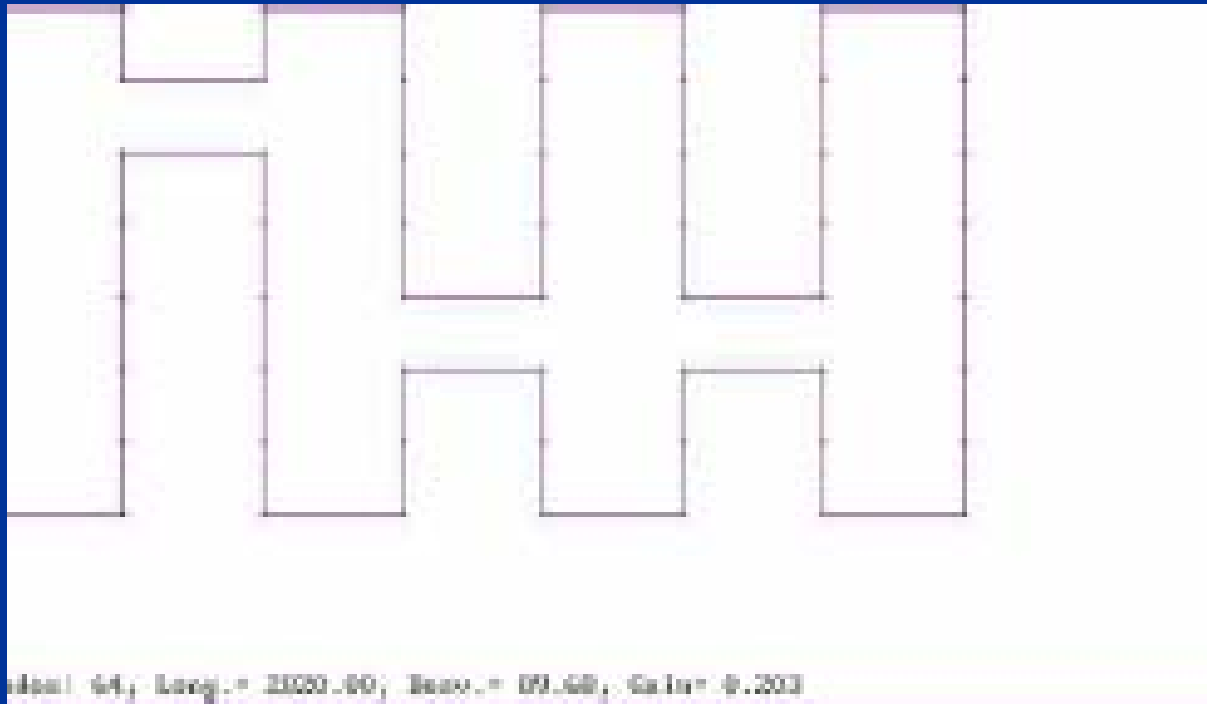
# Mapas autoorganizados de Kohonen

- Ejemplo: problema del viajante
  - Problema del viajante, iteración 500:



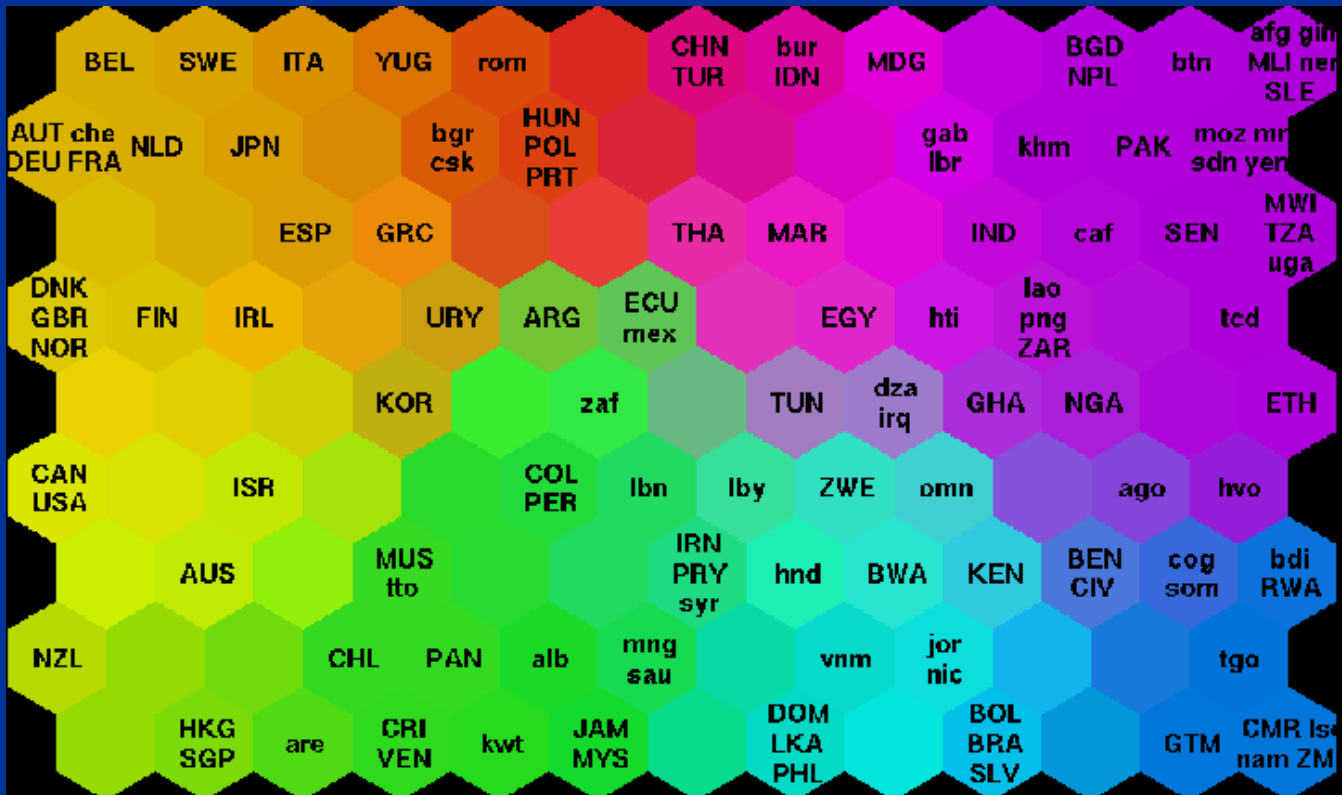
# Mapas autoorganizados de Kohonen

- Ejemplo: problema del viajante
  - Problema del viajante, 64 ciudades:



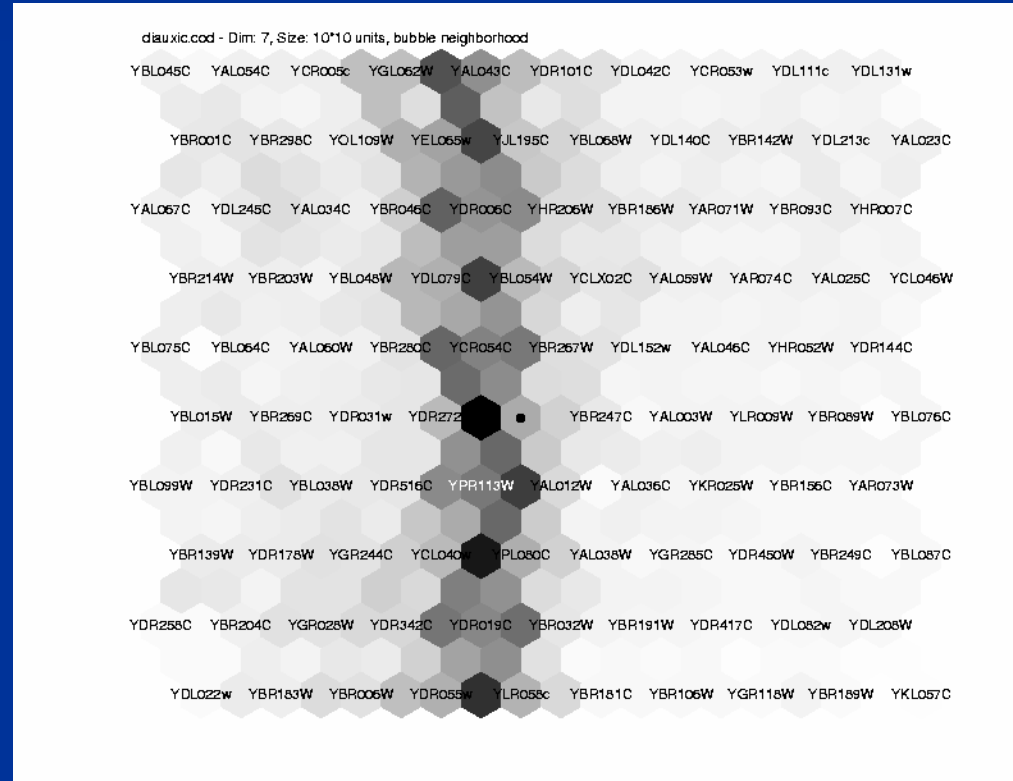
# Mapas autoorganizados de Kohonen

- Ejemplo: mapa de pobreza mundial
  - Generado a partir de 39 dimensiones



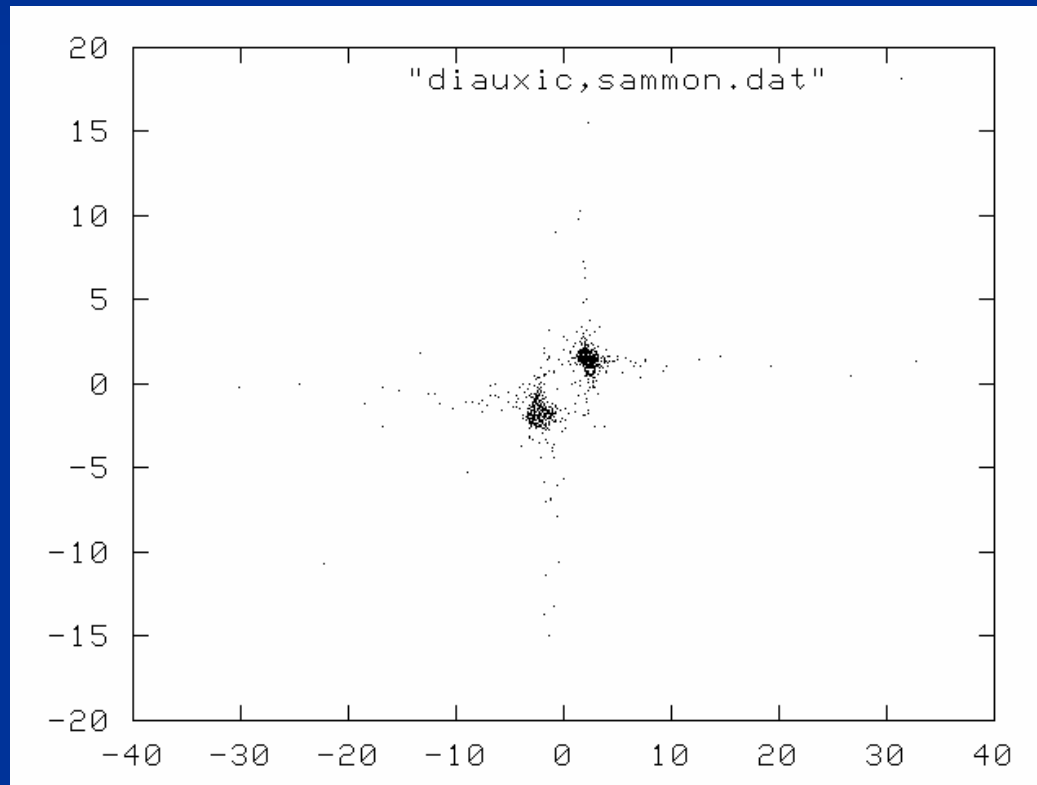
# Mapas autoorganizados de Kohonen

- Ejemplo: expresión de genes (7d)
- SOM



# Mapas autoorganizados de Kohonen

- Ejemplo: expresión de genes (7d)
  - Proyección de Sammon



# Teoría de la resonancia adaptativa

## ■ Introducción

- Familia de modelos propuestos por Grossberg
- Abordan el dilema estabilidad-plasticidad
  - Reacción plástica ante nuevos datos
  - Reacción estable ante datos irrelevantes
- Se busca que el sistema conmute entre los estados
  - Sistema de detección de la relevancia
    - Aprendizaje no supervisado

# Teoría de la resonancia adaptativa

## ■ Arquitectura ART1

### ■ Memoria a corto plazo STM

- Capacidad de recordar algo ocurrido en un intervalo reciente de tiempo
- Independiente de número de veces que haya ocurrido
- Identificado en RNA con el valor de activación

### ■ Memoria a largo plazo LTM

- Capacidad de recordar algo ocurrido en un intervalo distante de tiempo
- Requieren estímulo repetido en el tiempo
- Identificado en RNA con los pesos de las conexiones

# Teoría de la resonancia adaptativa

- Arquitectura ART1
  - El modelo consta de un dos subsistemas y un término de control
    - Subsistema atencional
      - Fundamentalmente una RNA con aprendizaje competitivo
    - Subsistema orientador
      - Distingue entre entradas relevantes e irrelevantes
    - Término de control
      - Controla módulos que van a actuar
      - Gobierna señales entre módulos



# Teoría de la resonancia adaptativa

## ■ Arquitectura ART1

### ■ Arquitectura básica: capas

#### ■ Capa F1 (entrada)

- Propaga la señal de entrada a F2
- Misma dimensión que la entrada

#### ■ Capa F2 (competitiva)

- Tantas neuronas como prototipos (inicialmente vacía)
- Conexiones excitatorias con ellas mismas
- Conexiones inhibitorias de fuerza fija e igual con el resto

# Teoría de la resonancia adaptativa

- Arquitectura ART1

- Arquitectura básica: interconexiones

- Bottom-up

- Las habituales

- Top-down

- Permiten el aprendizaje

- Facilitan la comparación de las predicciones con los datos de entrada

# Teoría de la resonancia adaptativa

## ■ Arquitectura ART1

### ■ Relación entre subsistemas

#### ■ Subsistema atencional

- Aprendizaje competitivo

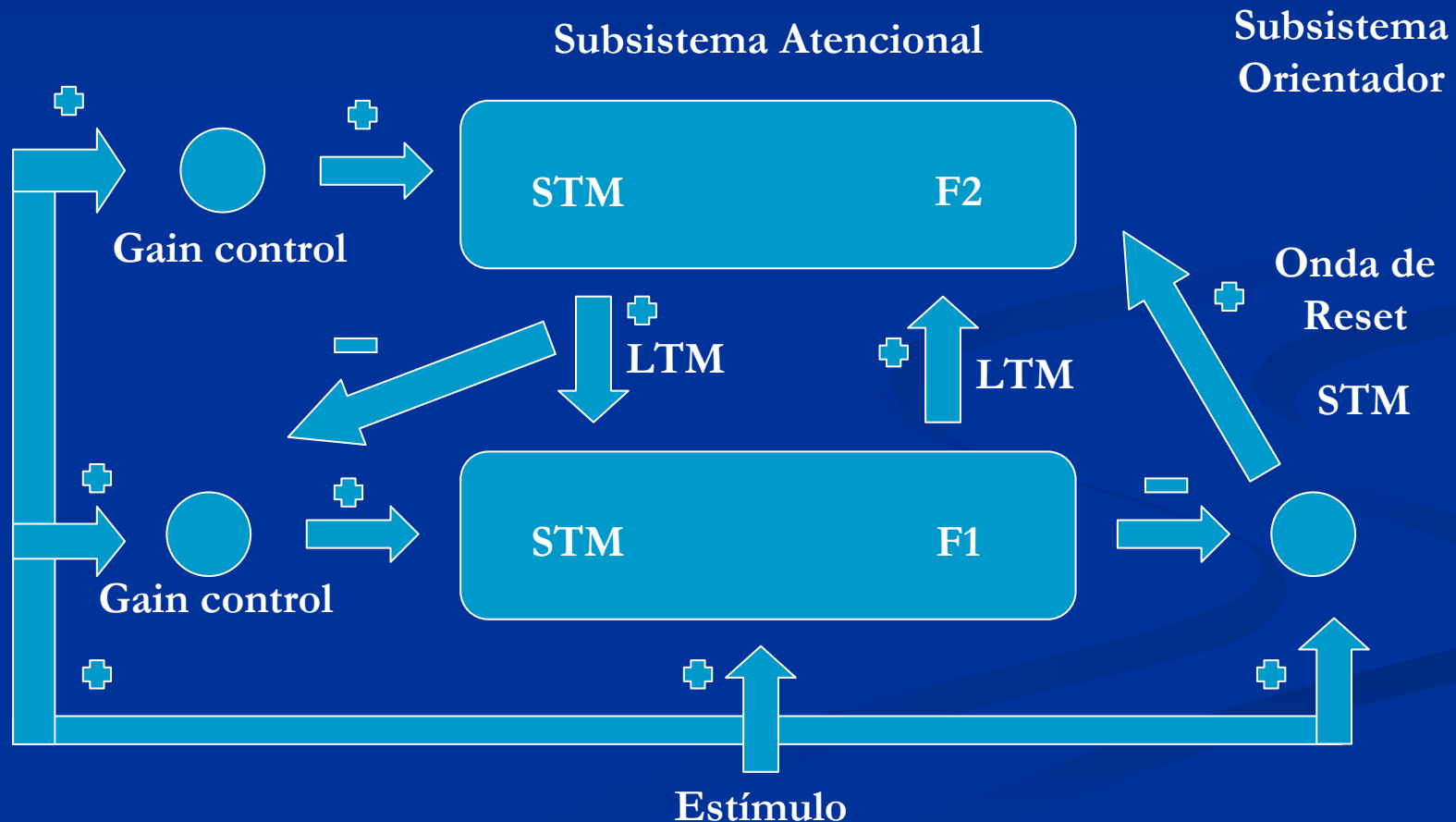
- Incapaz de mantener simultáneamente representaciones estables de categorías familiares y crear nuevas categorías

#### ■ El subsistema orientador

- Inhibe al atencional ante estímulos no familiares

# Teoría de la resonancia adaptativa

## ■ Arquitectura ART1



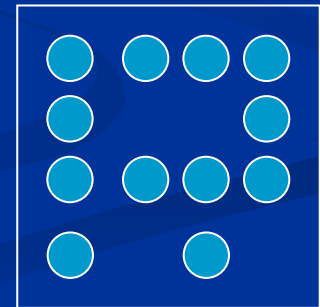
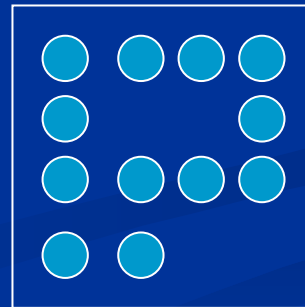
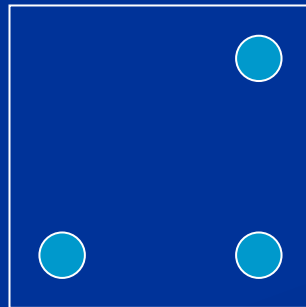
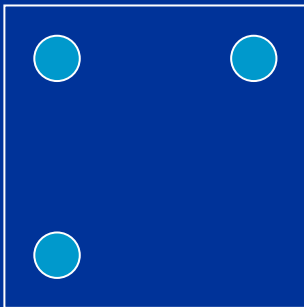
# Teoría de la resonancia adaptativa

## ■ Funcionamiento

- Conexiones bottom-up: almacenan los códigos de predicción
- Conexiones top-down: almacenan los valores de los prototipos
- El rendimiento del sistema es muy sensible al factor de vigilancia ( $\rho$ )

# Teoría de la resonancia adaptativa

- Características de los modelos ART
  - Unidades computacionales autoescalables
    - Determina el número de forma automática
    - Considera el contexto
    - Sensibilidad a definiciones de señal y ruido dependientes del aprendizaje



# Teoría de la resonancia adaptativa

- Acceso a memoria autoajustable
  - Más prototipos no implica más tiempo de categorización
    - Búsqueda paralela
    - Búsqueda ordenada
- Acceso directo a los códigos de reconocimiento
  - Alta velocidad de reconocimiento
  - Independiente de la complejidad del código aprendido
  - A medida que el aprendizaje es más autoconsistente y predictivamente exitoso, es más rápido

# Teoría de la resonancia adaptativa

- Uso del entorno como profesor
  - Si se penaliza el error de reconocimiento -> sistema más vigilante