



Inteligencia Artificial

Unidad 3

Redes Neuronales

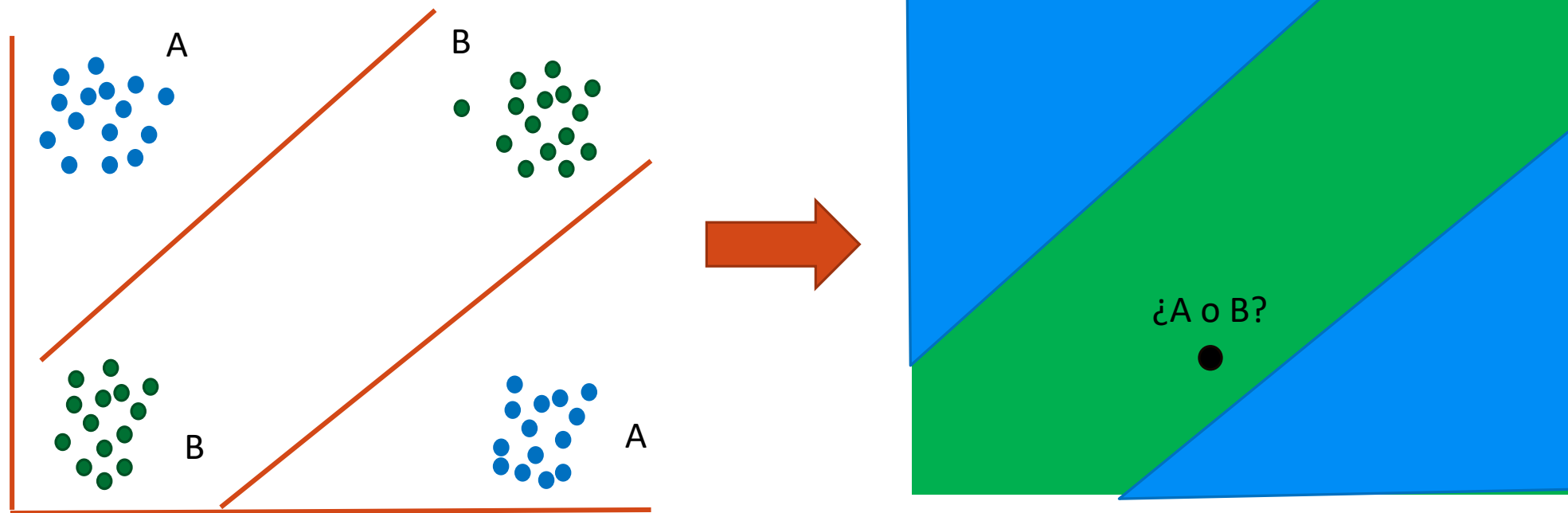
MAPAS AUTO-ORGANIZADOS

- Hugo David Calderón
- Heider Sanchez Enriquez
- Willy Ugarte



Aprendizaje Supervisado

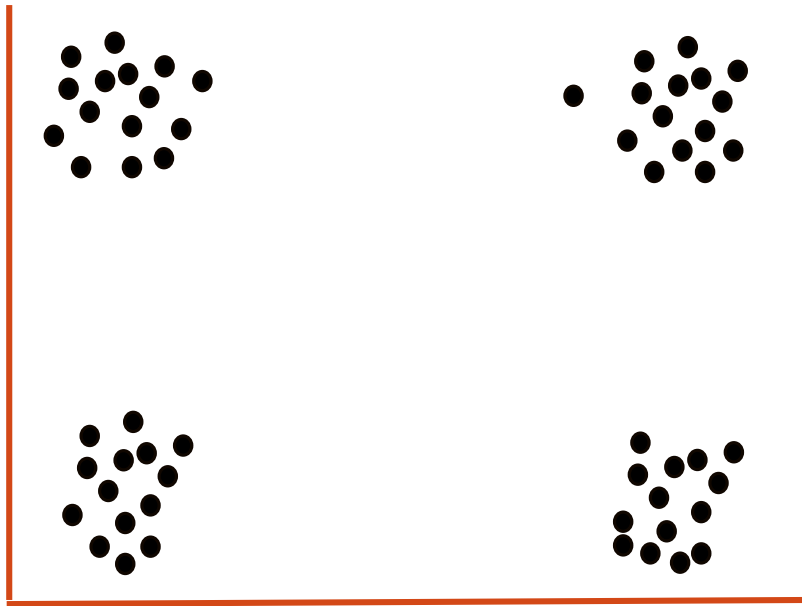
Hasta ahora, las redes neuronales estudiadas son usadas para el aprendizaje supervisado, es decir, la base de conocimiento está formada por datos etiquetados (conjunto de entrenamiento), y el objetivo es entrenar un modelo para predecir nuevos datos no etiquetados.





Aprendizaje No Supervisado

¿Qué pasaría si no tenemos la etiqueta en los datos? ¿Cómo saber en cuantas clases se agrupan los datos? Es aquí en donde tenemos que hacer uso de técnicas no supervisadas.

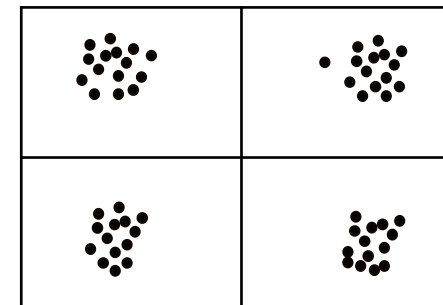
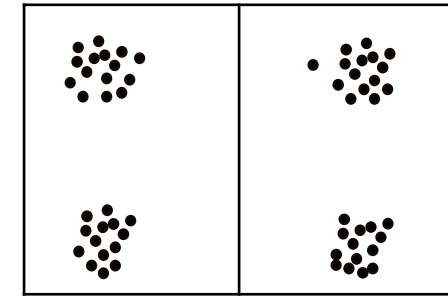
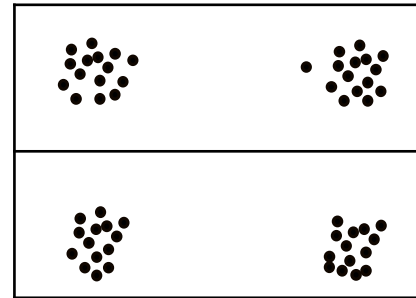
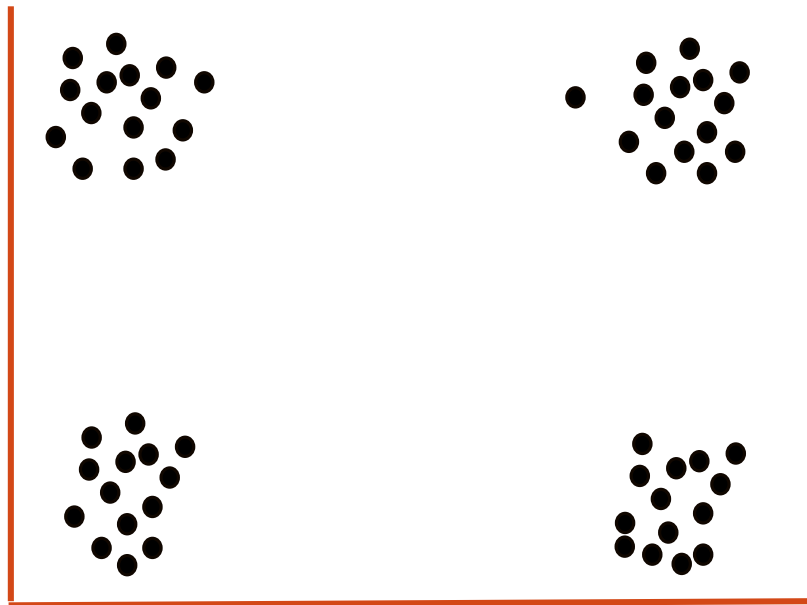


En el aprendizaje no supervisado el modelo es ajustado automáticamente a las datos.



Aprendizaje No Supervisado

En el siguiente conjunto de puntos nosotros podemos establecer visualmente una malla de separación de puntos basándonos en la cohesión interna. Cada celda agrupa puntos cercanos entre si.

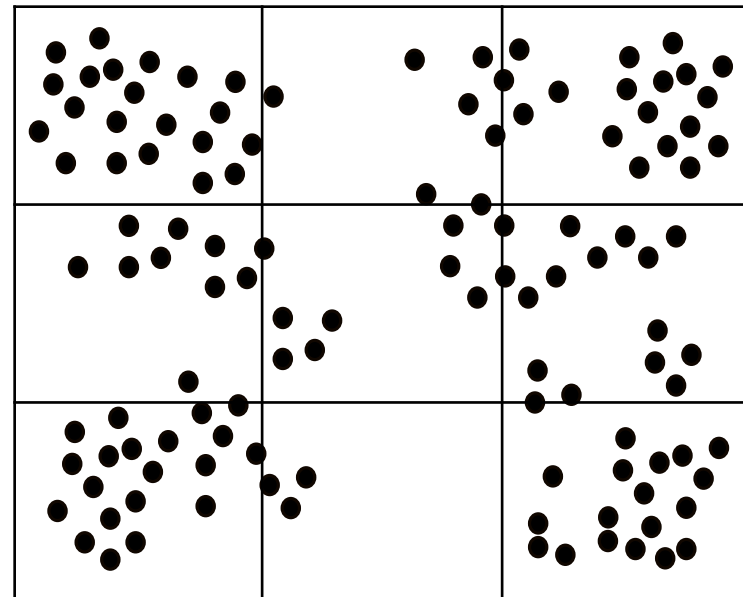
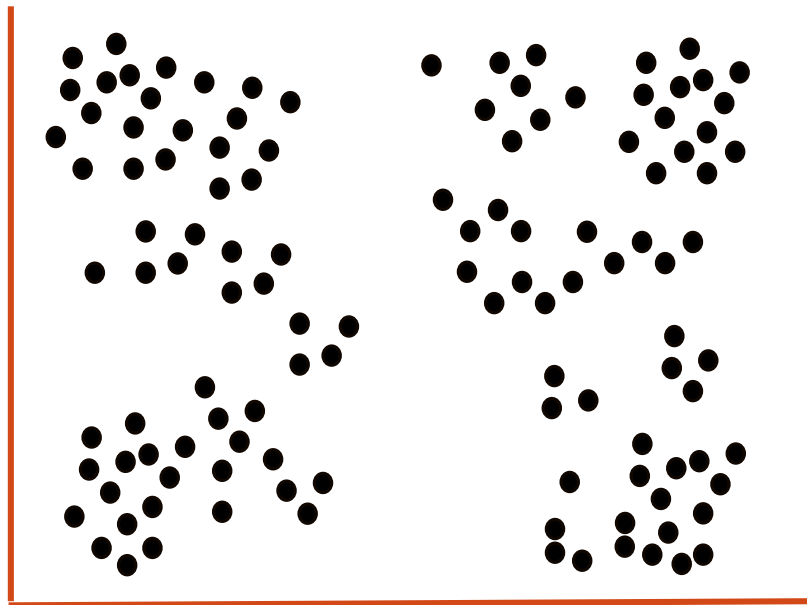


Este modelo de malla se adapta mejor.



Aprendizaje No Supervisado

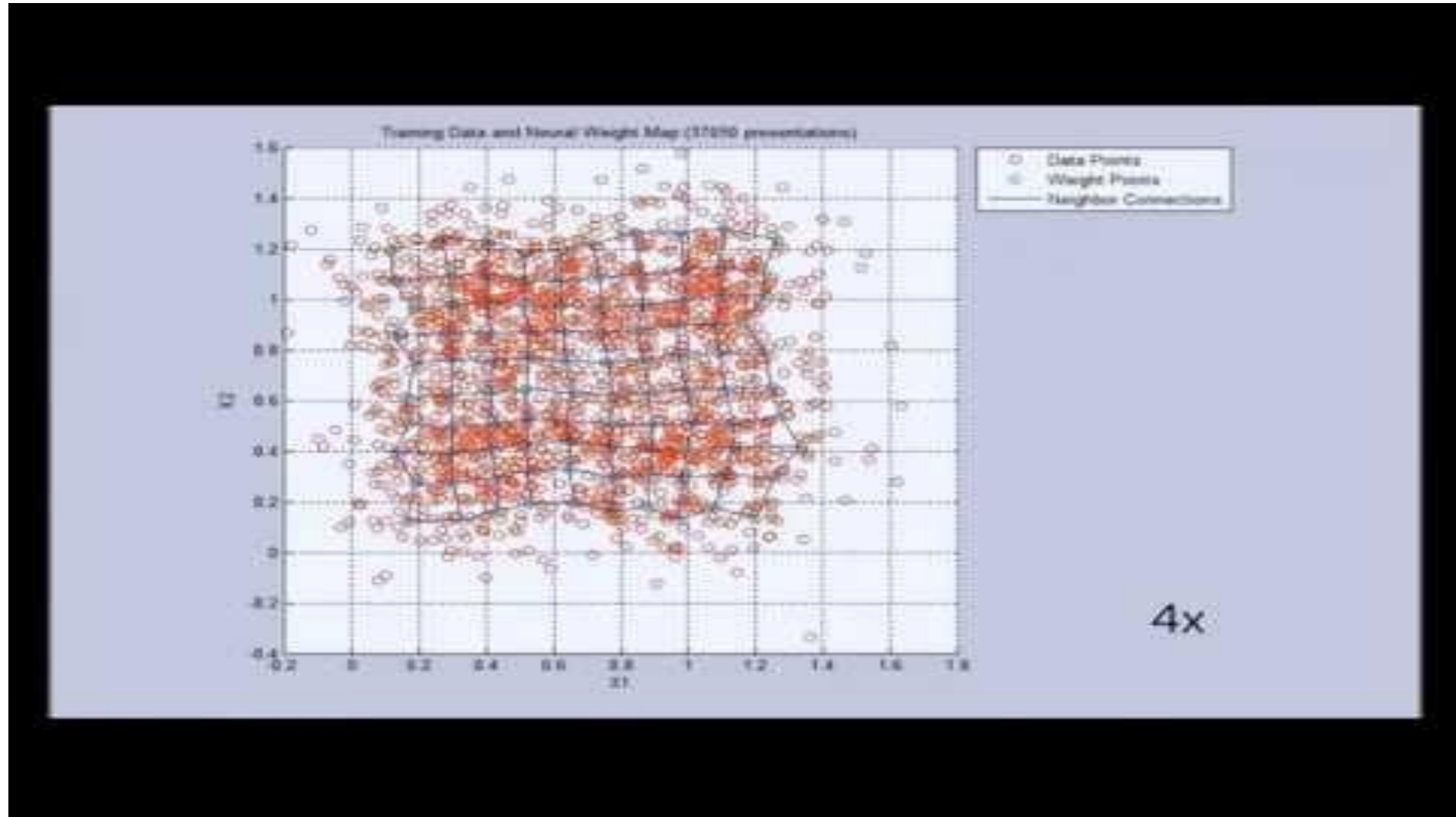
¿Y si tenemos más puntos que tienden a una distribución uniforme como sería nuestra malla? ... la topología de la malla debe adaptarse a los datos.



Para dos dimensiones,
hacerlo visualmente es
relativamente fácil,

**¿Pero cómo sería para
puntos
multidimensionales?**

Mapas Auto-Organizados





Mapas Auto-Organizados

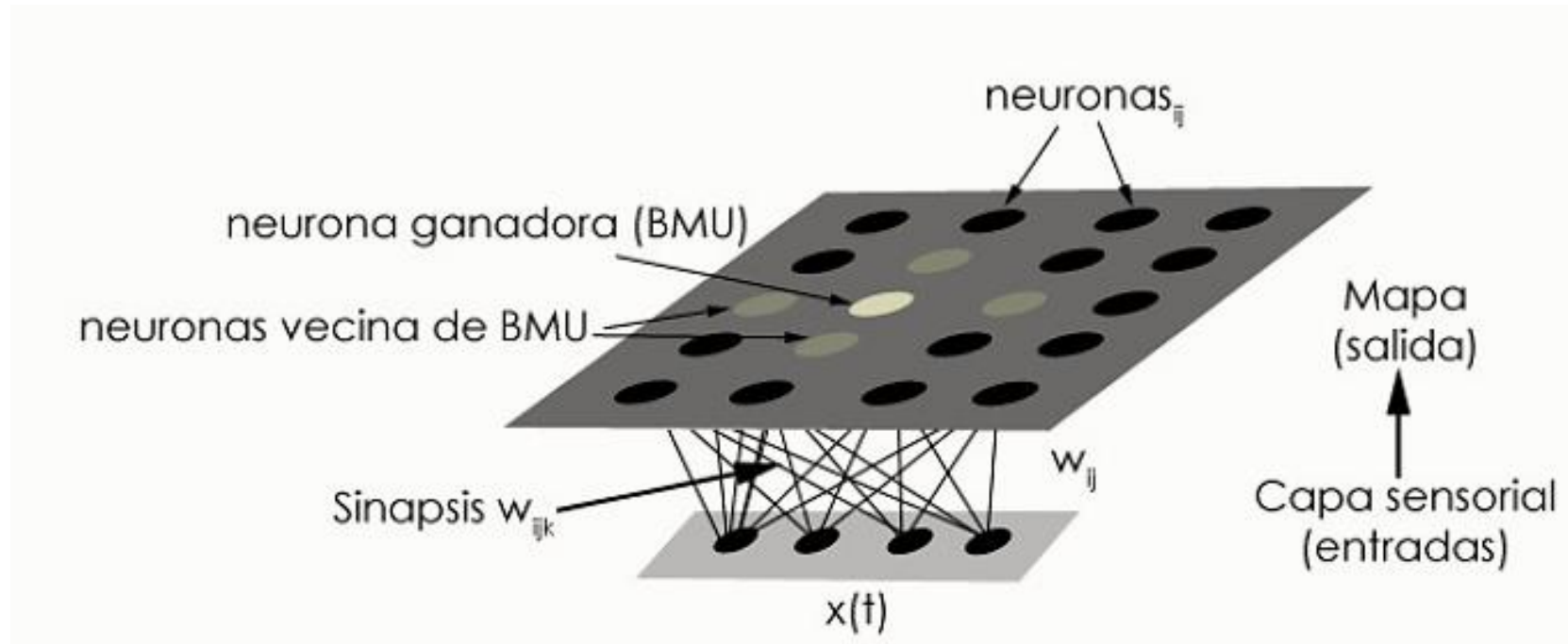
Los Mapas Auto-Organizados (Self Organizing Maps, SOM) fueron desarrollados por el finlandés Teuvo Kohonen en 1982, por el cual también son conocidos como **Redes de Kohonen**.

Permiten encontrar individuos de una población que comparten características comunes.

Esto los hace ideales para explorar espacios vectoriales en los que se desconoce la estructura de clasificación de los vectores.



Mapas Auto-Organizados: Arquitectura



Estructura de un mapa auto-organizado



Mapas Auto-Organizados

- ❖ La idea básica del modelo es crear una imagen de un espacio multidimensional de entrada en un espacio de salida de menor dimensionalidad.
- ❖ Se trata de un modelo con dos capas de neuronas, una de entrada y otra de salida (procesamiento).
- ❖ Las neuronas de la primera capa se limitan a recoger y canalizar la información.
- ❖ La segunda capa está conectada a la primera a través de los pesos sinápticos y realiza la tarea importante: **una proyección no lineal del espacio multidimensional de entrada, preservando las características esenciales de estos datos en forma de relaciones de vecindad.**
- ❖ El resultado final es la creación del llamado mapa auto-organizado donde se representan los rasgos más sobresalientes del espacio de entrada



Algoritmo de Aprendizaje: resumen

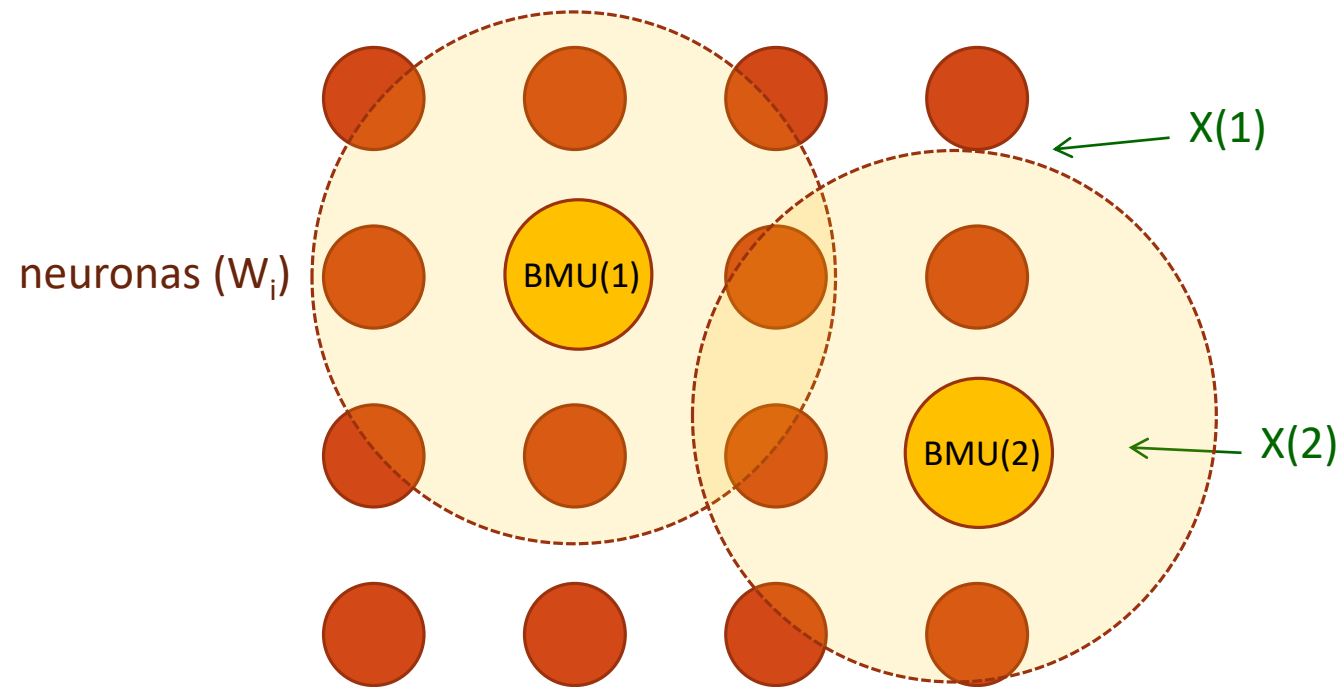
1. Inicializar todas las neuronas de la capa de salida.
2. Repetir mientras las neuronas no cambien mucho o se exceda un umbral
 3. Escoger la siguiente entrada
 4. Determinar la neurona más cercana a la entrada
 5. Actualizar los pesos de esta neurona y las neuronas cercanas (en una vecindad especificada)
6. Finalmente, formar los grupos asignando cada entrada a su neurona más cercana.



Algoritmo de Aprendizaje: paso a paso

1. Todos los pesos de las neuronas W_i de salida reciben, inicialmente, valores aleatorios.
2. Se presenta el vector de entrenamiento $X(t)$ a la red.
3. Se determina cuál de las neuronas está más cerca de la entrada. La neurona ganadora es también llamada **BMU** (Best Matching Unit)

Algoritmo de Aprendizaje: paso a paso



Algoritmo de Aprendizaje: paso a paso



3. Para medir la cercanía entre el vector de pesos de las neuronas y el vector de entrenamiento, normalmente se usa la distancia euclidiana:

$$\|X - W\| = d(X, W) = \sqrt{\sum_{j=1}^D (x_j - w_j)^2}$$

Esta sujeto al tipo de aplicación.

Ej. En text-mining se suele usar la distancia coseno.

El BMU $W_c(t)$ es la que obtiene la mínima distancia a $X(t)$:

$$\|X(t) - W_c(t)\| = \min_i \{\|X(t) - W_i(t)\|\}$$



Algoritmo de Aprendizaje: paso a paso

4. Los vectores de pesos del BMU y sus vecinos (en sentido topológico) se actualizan para moverse cerca del vector $X(t)$ en el espacio de datos:

$$W_i(t + 1) = \begin{cases} W_i(t) + h_i(t)[X(t) - W_i(t)] & i \in N_c(t) \\ W_i(t) & i \notin N_c(t) \end{cases}$$

En donde la magnitud de dicha atracción está regida por la tasa de aprendizaje h , el cual depende tanto del tiempo t como de la distancia entre W_i y W_c .

N_c representa el conjunto de las neuronas vecinas de W_c para el tiempo t .



Algoritmo de Aprendizaje: paso a paso

4. La tasa de aprendizaje h se calcula de la siguiente forma:

$$h_i(t) = \alpha(t) * e^{-\frac{d(W_i, W_c)}{2\sigma^2}}$$

↑
taza de aprendizaje

← distancia entre el BMU y la neurona vecina

← ancho de del kernel

$$\alpha(t) = 0.8 \cdot \left(1.0 - \frac{t}{t_{Max}}\right)$$

← maximo número de iteraciones

$$\sigma(t) = 0.25 \cdot (H + W) \cdot \left(1.0 - \frac{t}{t_{Max}}\right)$$

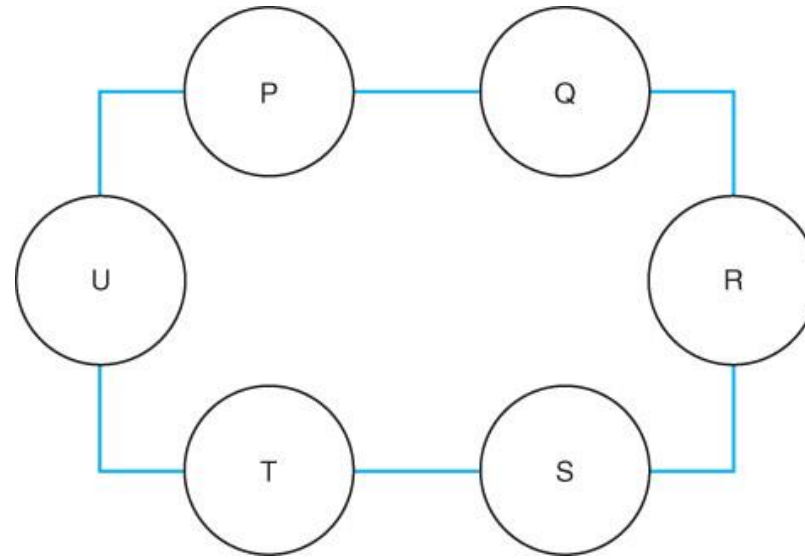
↑

alto y ancho del mapa neuronal



Algoritmo de Aprendizaje: ejemplo

Dada la siguiente topología del SOM



Y los pesos para cada neurona:

	P	Q	R	S	T	U
w_1	-1	0	3	-2	3	4
w_2	2	4	-2	-3	2	-1



Algoritmo de Aprendizaje: ejemplo

Usando la distancia Euclideana, nosotros podemos encontrar la neurona ganadora para el dato de entrada $x = \{x_1 = 2, x_2 = -4\}$

$$\|x - w_P\| = \sqrt{(2+1)^2 + (-4-2)^2} = \sqrt{45},$$

$$\|x - w_Q\| = \sqrt{(2-0)^2 + (-4-4)^2} = \sqrt{68},$$

$$\|x - w_R\| = \sqrt{(2-3)^2 + (-4+2)^2} = \sqrt{5},$$

$$\|x - w_S\| = \sqrt{(2+2)^2 + (-4+3)^2} = \sqrt{17},$$

$$\|x - w_T\| = \sqrt{(2-3)^2 + (-4-2)^2} = \sqrt{37},$$

$$\|x - w_U\| = \sqrt{(2-4)^2 + (-4+1)^2} = \sqrt{13}.$$



Algoritmo de Aprendizaje: ejemplo

Por lo tanto, la neurona ganadora es R, ya que éste tiene la distancia mas pequeña a x.

El siguiente paso consiste en actualizar los pesos de la vecindad con la fórmula vista en el paso 4. En donde la tasa de aprendizaje queda definido de la siguiente manera:

$$h_{ck}(t) = \begin{cases} 0.5 & \text{Si la neurona es el BMU.} \\ 0.25 & \text{Si la neurona es el vecino inmediato al BMU.} \\ 0 & \text{En otros casos.} \end{cases}$$



Algoritmo de Aprendizaje: ejemplo

Vamos a empezar con el BMU, es decir la neurona R, los nuevos pesos serían:

$$\begin{pmatrix} 3 \\ -2 \end{pmatrix} + 0.5 \left\{ \begin{pmatrix} 2 \\ -4 \end{pmatrix} - \begin{pmatrix} 3 \\ -2 \end{pmatrix} \right\} = \begin{pmatrix} 3 \\ -2 \end{pmatrix} + 0.5 \begin{pmatrix} -1 \\ -2 \end{pmatrix} = \begin{pmatrix} 2.5 \\ -3 \end{pmatrix}.$$

Los vecinos inmediatos de R son Q y S, entonces los nuevos pesos serían:

$$\begin{pmatrix} 0.5 \\ 2 \end{pmatrix} \quad \begin{pmatrix} -1 \\ -3.25 \end{pmatrix}$$



Presentando el SOM

Matriz de neuronas: para cada neurona visualizamos su vector de pesos W_i (datos en 3D, colores, espacios continuos)

Entonces, para un mapa de color con 3 entradas, si los pesos de la neurona son (0.7, 0.2, 0.3), mostraríamos un color rojizo con 0.7 de rojo, 0.2 de verdes y 0.3 de azul.

Para un mapa de puntos en el plano con dos entradas, dibujaríamos un punto para cada neurona en posición (W_x , W_y).



Presentando el SOM

Matriz de neuronas: para cada neurona visualizamos su vector de pesos W_i (datos en 3D, colores, espacios continuos)

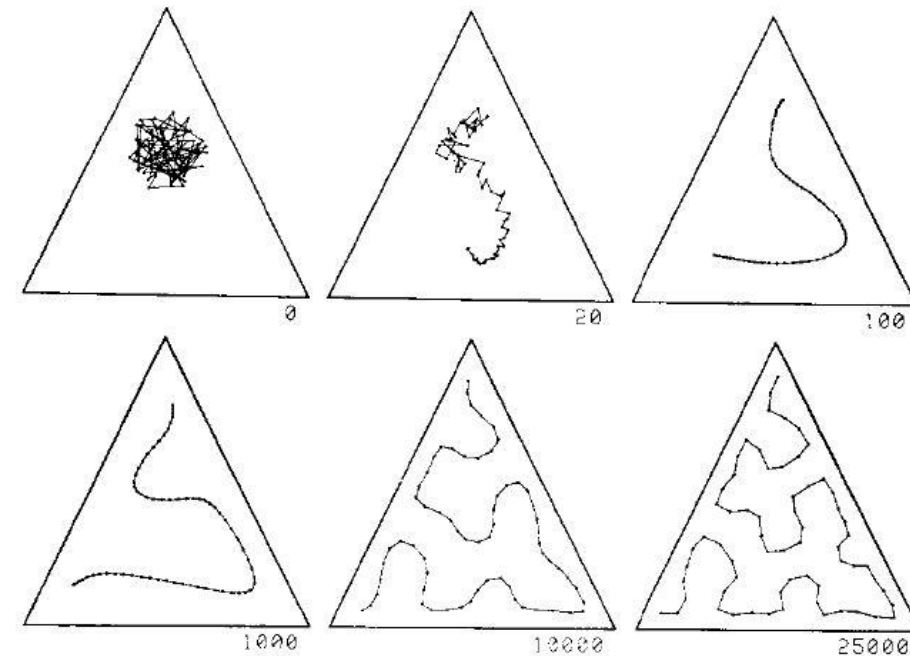
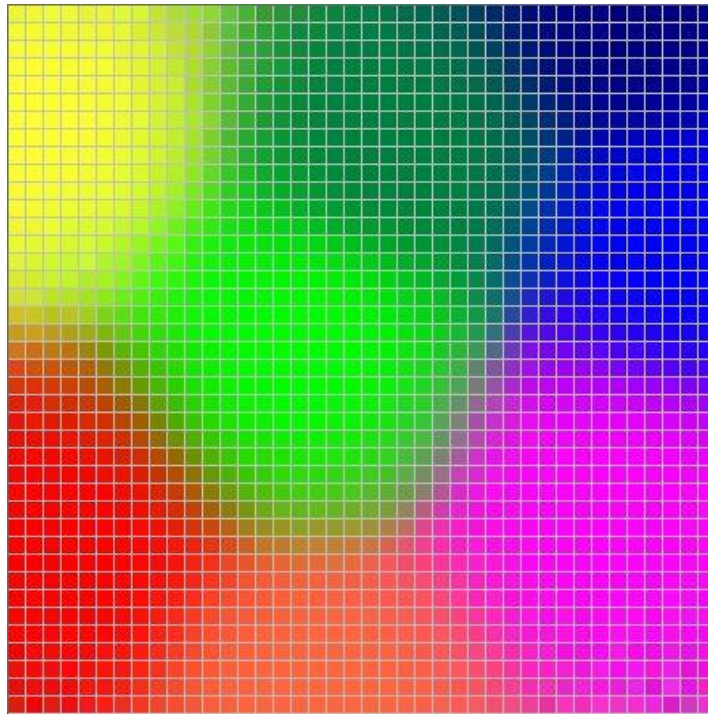
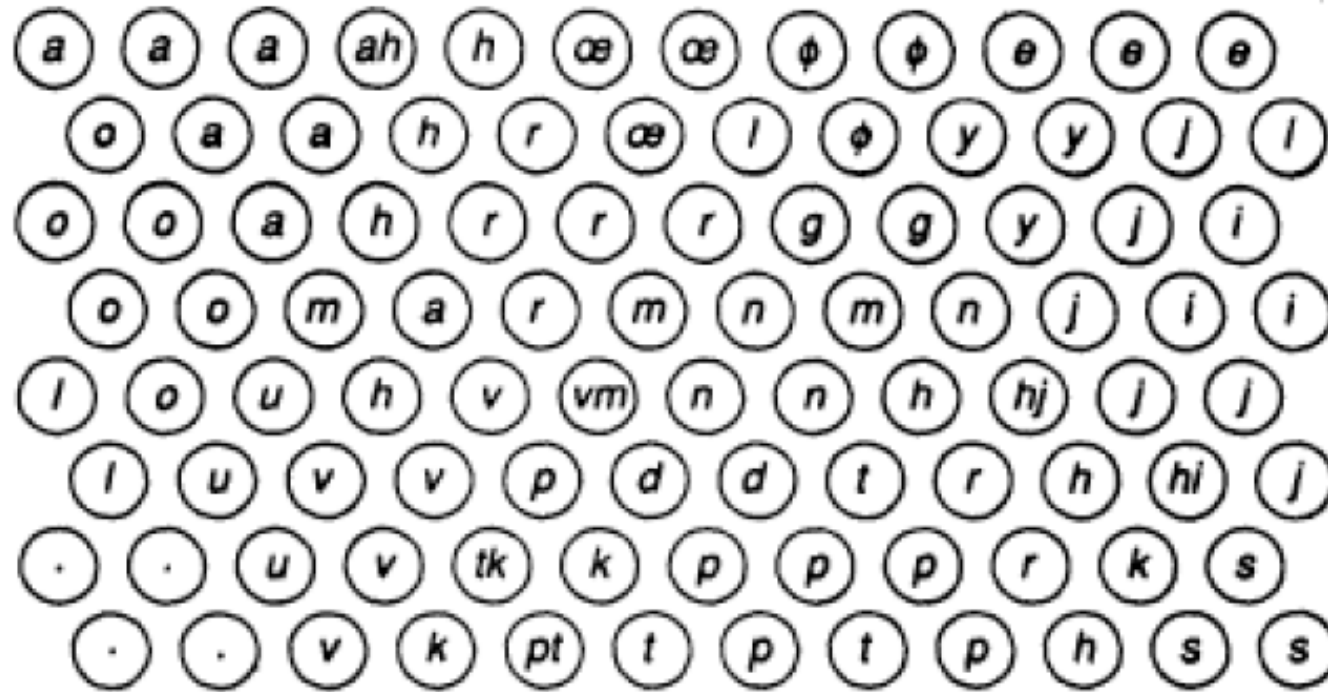


Fig. 4. Weight vectors during the ordering process, one-dimensional array.



Presentando el SOM

Mapa de centroides: representar cada neurona por un elemento del conjunto de entrenamiento X que está más cerca del vector W_i . Dicho elemento también se le conoce como centroide.



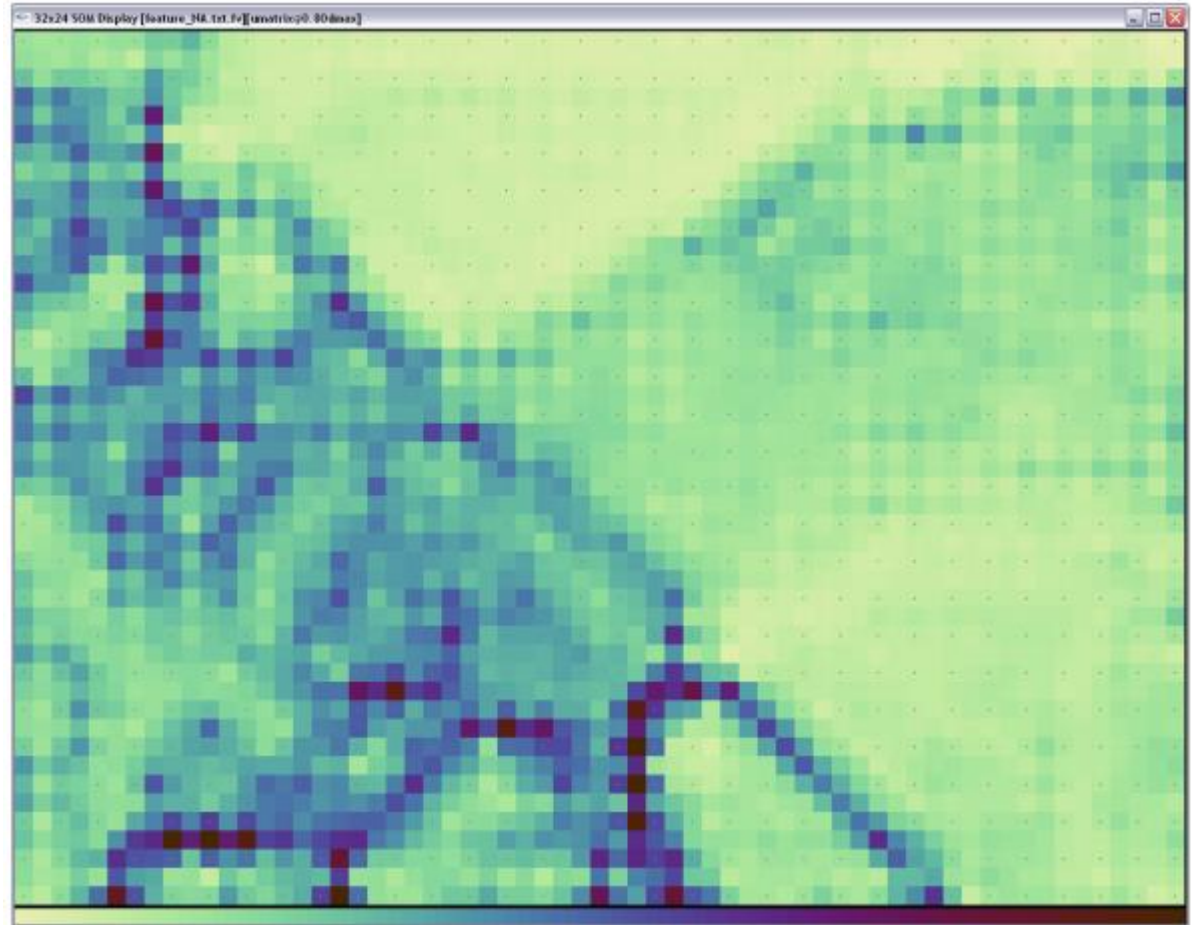


Presentando el SOM

Matriz Unificada De Distancias

Cuando se genera la matriz se calcula, a su vez, una matriz de distancias entre los vectores de referencia de neuronas adyacentes en el mapa bidimensional.

Después se selecciona algún tipo de representación gráfica, por ejemplo una escala de grises. Los colores en la figura se seleccionan de modo que cuanto más oscuro es el color entre dos neuronas, menor es la distancia entre ellas.

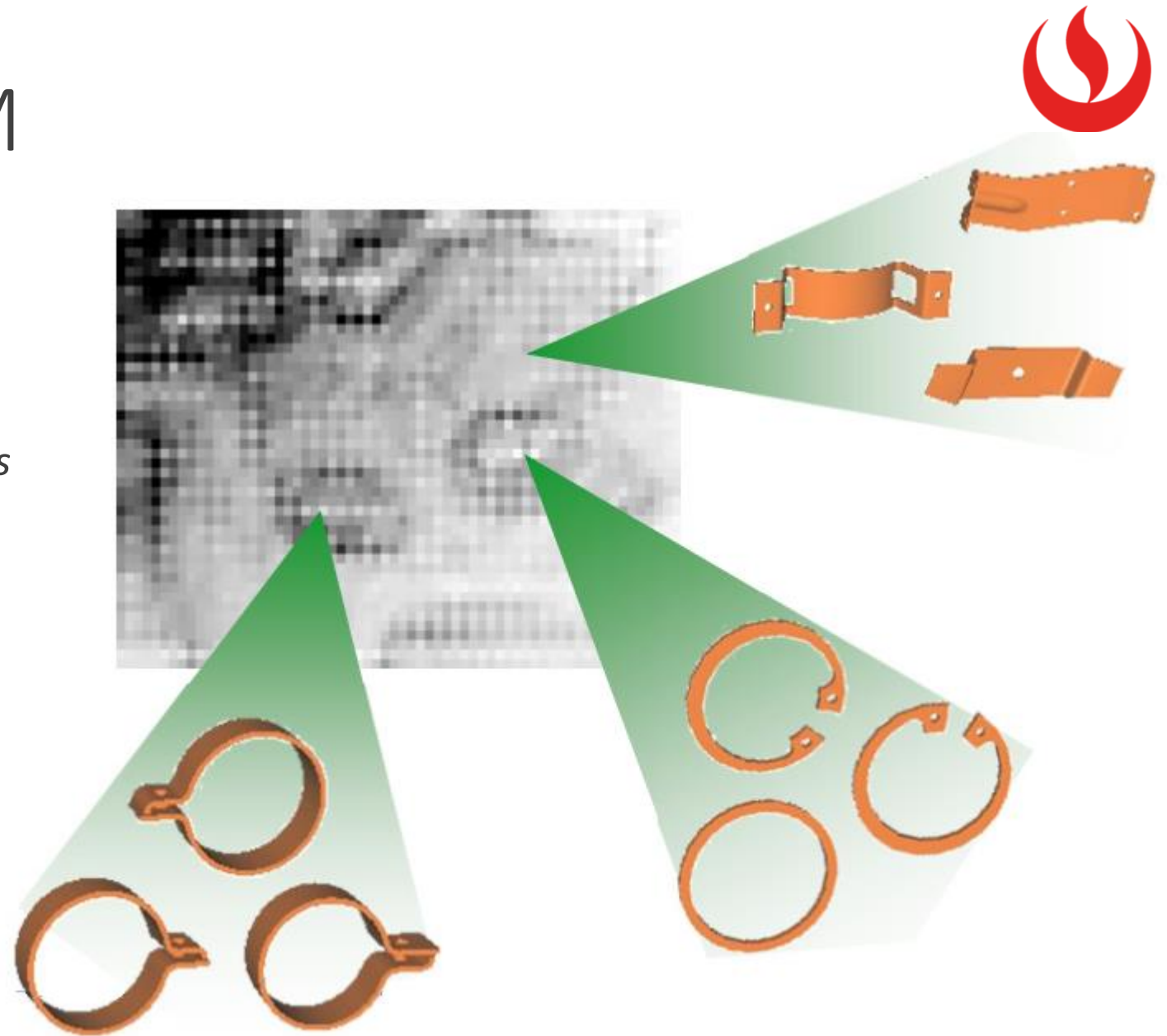


Presentando el SOM

Grupos de objetos y estructura del SOM

*Umatrix y thumbnails escogidos de objetos
3D CAD*

*Objetos son mapeados a los nodos
SOM de acuerdo al vecino más cercano*



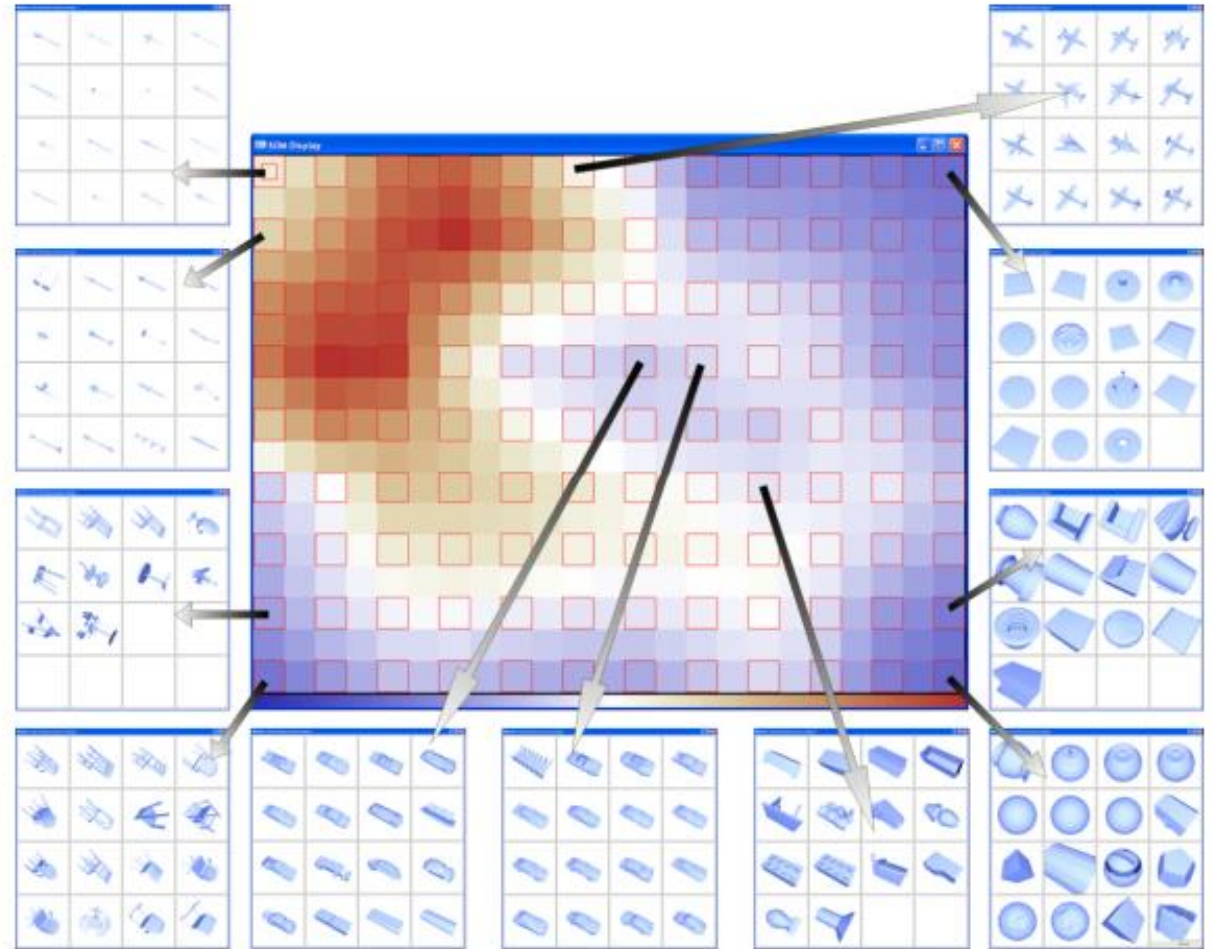


Presentando el SOM

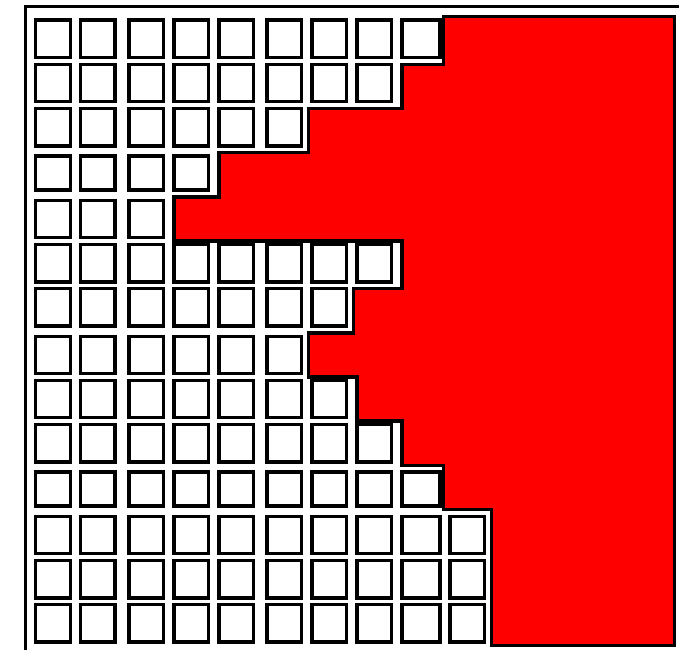
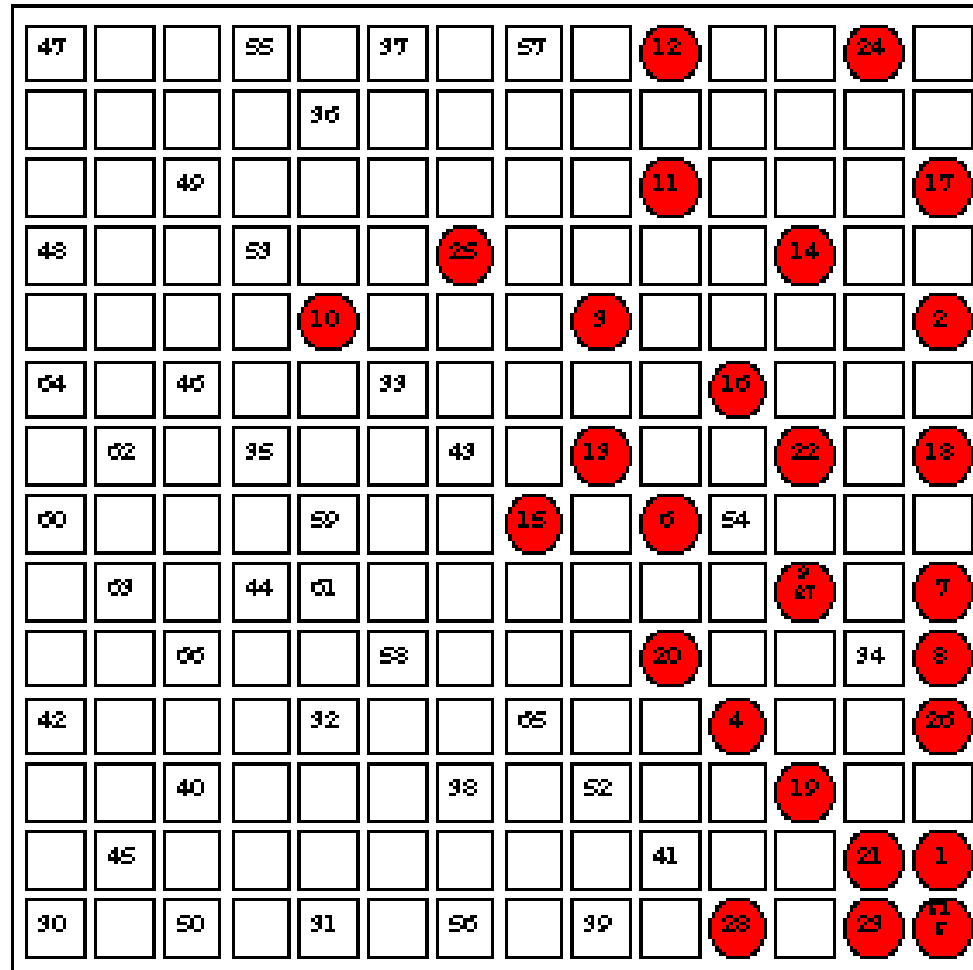
Grupos de objetos y densidad de objetos

Mapa de densidad suavizado y algunos thumbnails de una colección de datos 3D genérica

(color indica densidad relativa suavizada del vecino más cercano)



Aplicación: Estudio de la quiebra





Aplicación: Estudio de la quiebra

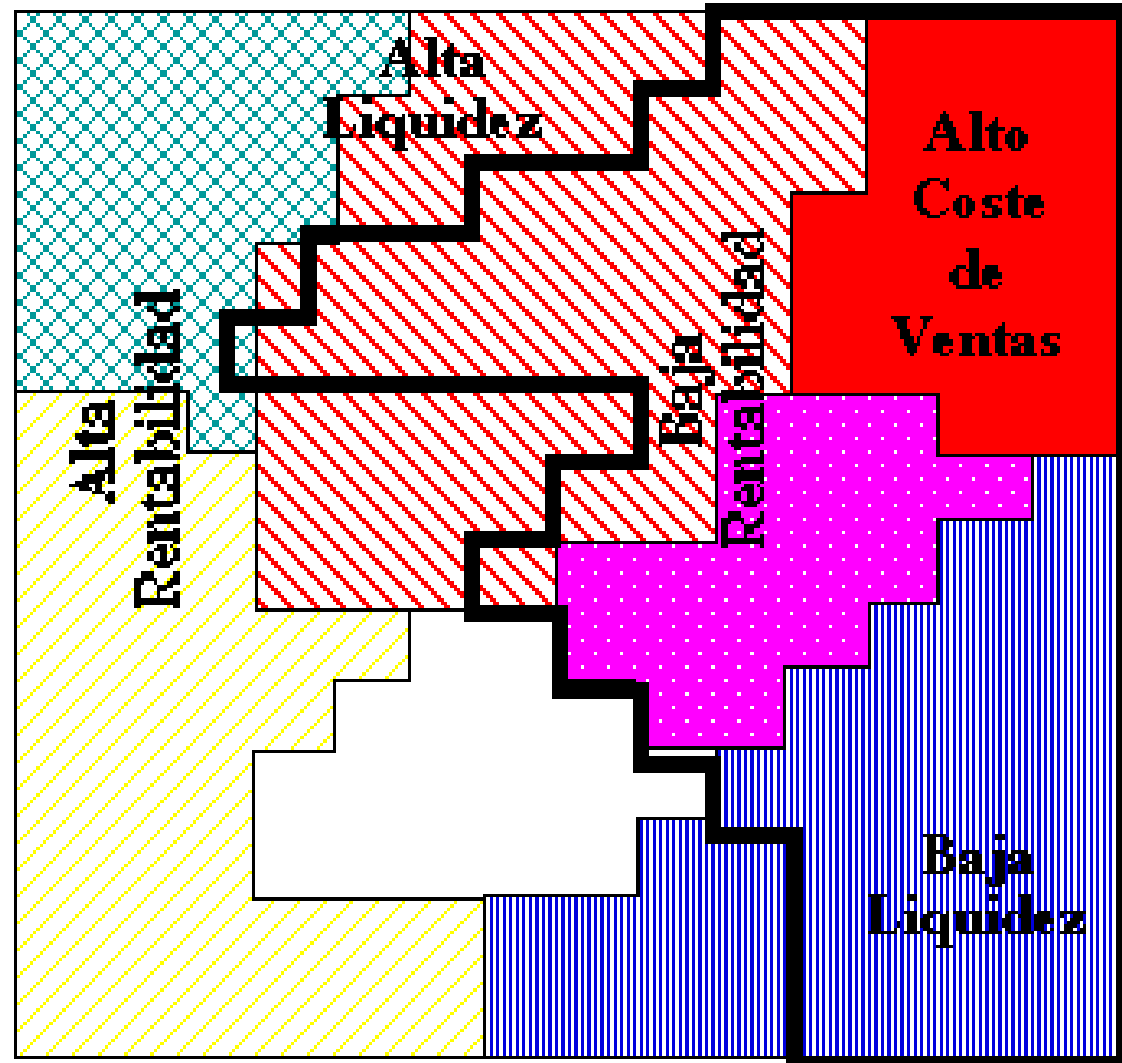
En el mapa se observa cómo aquellos bancos que quebraron al año siguiente, se ubican en la zona derecha del mapa, y los solventes se alojan en la parte izquierda del mapa.

Esta figura recibe el nombre de "**mapa de solvencia**", Serrano y Martín (1994). Otros mapas auxiliares del modelo permiten conocer qué rasgos financieros dominan en cada zona del mapa: zonas de alta y baja rentabilidad, liquidez, endeudamiento, etc, véase Serrano (1996).

Aplicación: Estudio de la quiebra



El siguiente mapa muestra los rasgos financieros que caracterizan a cada zona del mapa.





Aplicación: Estudio de la quiebra

La red de kohonen descrita puede ser de gran utilidad en el análisis de información contable de empresas. Al introducir información contable de un conjunto de empresas se producirá una autoorganización de las mismas, de forma que empresas con características financieras similares se colocarán próximas en el mapa. **La situación de una empresa vendrá determinada por su ubicación en el mapa,** teniendo en cuenta que una empresa puede excitar a más de una neurona y esto con diferentes intensidades



Aplicación: Estudio de la quiebra

Además, este modelo permite, estudiar la **evolución temporal de una empresa** introduciendo información procedente de varios ejercicios, situarla en relación con sus competidores, elaborar **mapas sectoriales**, introducir ratios financieros o partidas como la cifra de ventas o el activo, información cualitativa, etc. A priori, son muchas las aplicaciones que puede tener, como análisis exploratorio de datos o mediante su integración en un sistema de ayuda a la toma de decisiones.



Aplicación: Mapa de Movimientos

Mapa de movimiento: recuperacion basado en imagenes y segmentacion de datos de movimiento.

- Sakamoto, Kuriyama, Kenko
- SCA: Symposium on Computer Animation 2004

Objetivo:

Presentar al usuario una cuadrícula de posturas para seleccionar un clip de datos de movimiento de una gran base de datos.

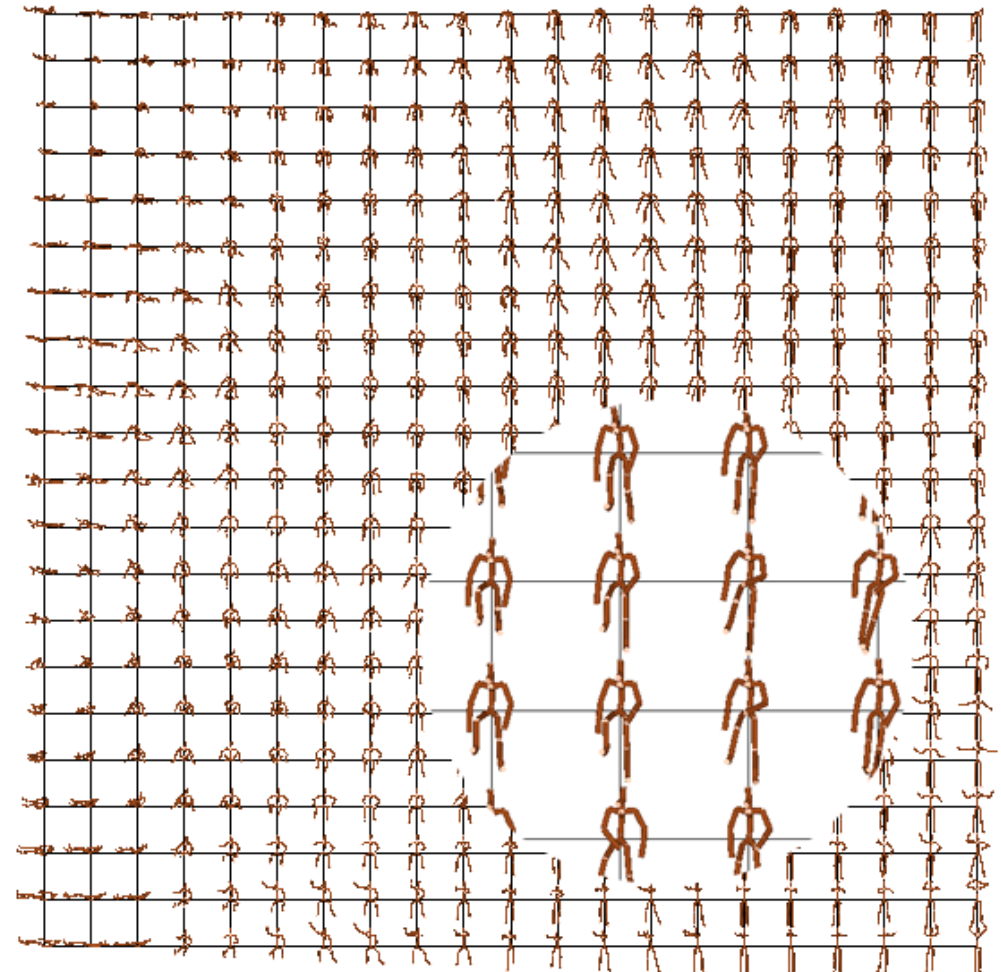
Realice la agrupación en el SOM en lugar de los datos abstractos.



Aplicación: Mapa de Movimientos

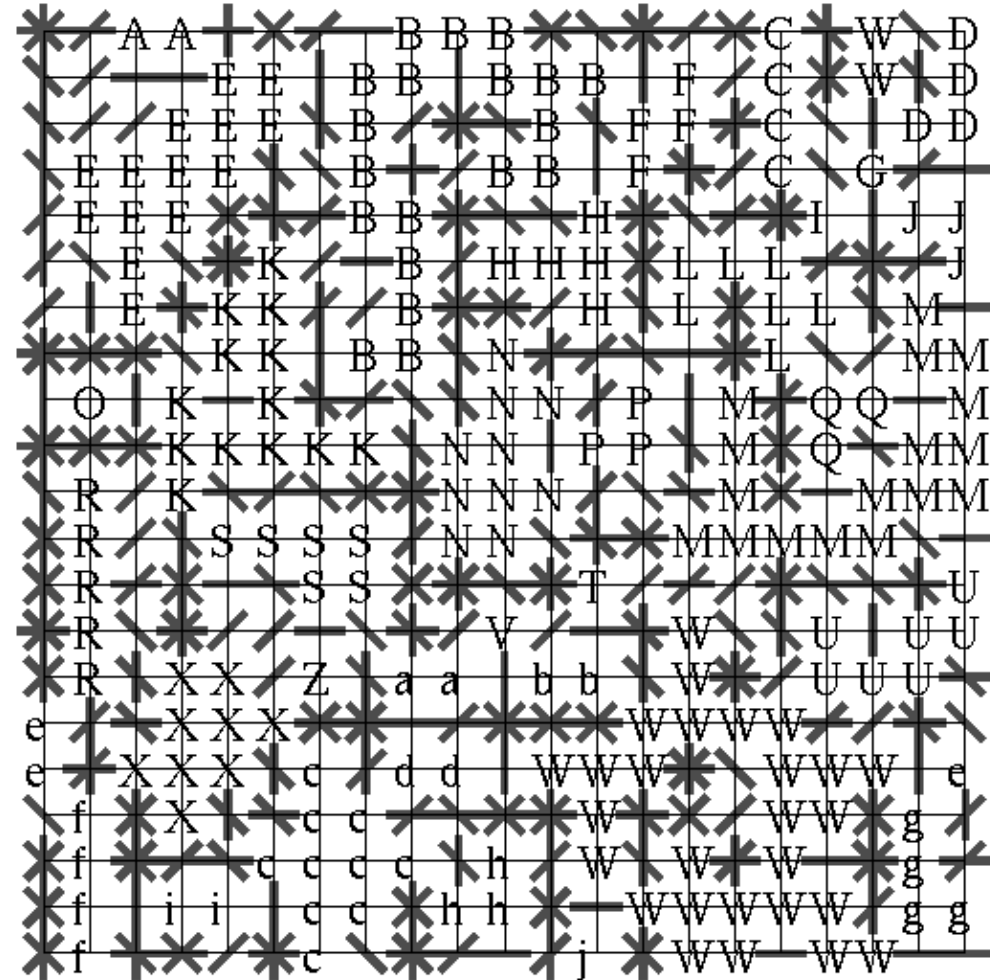
Ejemplo:

436 muestras de postura a partir de 55K frames de 51 archivos de movimiento.





Clustering basado en SOM





Aplicación: Mapa de Movimientos

Proceso:

Se crea un mapa de muestras de postura a partir de todos los archivos de movimiento juntos.

Cada muestra similar a su muestra más cercana supera un umbral determinado para reducir el tiempo de cálculo.

Se calcula un SOM estándar.

Cada postura se conecta a una tabla hash de los archivos de movimiento que contienen posturas similares.

Agrupar el SOM permite mostrar un mapa simplificado al usuario.



Aplicación: Mapa de Movimientos

Mapa simplificado después de la agrupación de SOM:

17 estilos de baile

