

Redes neuronales

Departamento de Computación - FCEyN - UBA

Trabajo práctico 2

Aprendizaje no supervisado

Matías Battocchia

27 de junio de 2018

# Introducción

El trabajo consiste en clasificar documentos de manera no supervisada. Los datos de entrada son 900 documentos distribuidos en 9 categorías uniformemente. Los datos cuentan con 850 atributos que son frecuencias de palabras.

El código del trabajo es accesible desde el repositorio <https://github.com/matiasbattocchia/redes-neuronales>.

# Aprendizaje hebbiano

El aprendizaje hebbiano es una teoría del **aprendizaje asociativo**. Inspirada en una base biológica, sostiene que las células que se activan simultáneamente *cuando una de las células contribuye a activar a la otra* tienden a reforzar su conexión sináptica.

Se implementó un perceptrón simple sin sesgo (*bias*), de unidades lineales, cantidad de unidades de entrada igual a la dimensión de los datos (850), cantidad de unidades de salida equivalente a la dimensión deseada (3). El valor de la unidad de salida  $y_i$  está dado por

$$y_i = \sum_j w_{ij}x_j \quad (1)$$

donde  $w_{ij}$  es el peso entre la unidad de salida y la unidad de entrada  $x_j$ . En principio se podría utilizar la **regla de Hebb** para actualizar los pesos,

$$\Delta w_{ij} = \eta y_i x_j \quad (2)$$

donde  $\eta$  es el factor de aprendizaje; sin embargo esta regla hace que los pesos crezcan o decrezcan exponencialmente. Usualmente se utilizan otras reglas como las presentadas a continuación que estabilizan el aprendizaje.

La **regla de Oja** es un algoritmo que se desprende del de Hebb, normalizando los pesos al actualizarlos.

$$\Delta w_{ij} = \eta y_i(x_j - y_i w_{ij}) \quad (3)$$

En la derivación de la regla se usa que  $|\eta| \ll 1$  por lo tanto hay una restricción a la hora de escoger el valor del factor de aprendizaje.

La regla de Oja extrae la *primera* componente principal de los datos. Solo tendría sentido aplicarla cuando la dimensión de salida es una. Aunque no de una manera muy efectiva, es capaz de reducir la dimensionalidad separando los datos (figura 2). Suponemos que como los pesos se inicializan al azar, se terminan creando *proyecciones al azar* en torno a la componente principal.

**La regla de Sanger** es una combinación de la regla de Oja y el proceso de ortonormalización de Gram–Schmidt. Generaliza la regla de Oja a múltiples salidas y obtiene *las primeras* componentes principales.

$$\Delta w_{ij} = \eta y_i \left( x_j - \sum_{k=1}^i w_{kj} y_k \right) \quad (4)$$

En general no hay mucho que variar: las reglas utilizadas son métodos que convergen. La inicialización de pesos y la elección del factor de aprendizaje solo acortan o dilatan la cantidad de épocas necesarias para alcanzar un estado estable como el de la figura 4: comparar la similitud entre las figuras 5 (1000 épocas,  $\eta = 1 \times 10^{-4}$ ) y 3 (100 épocas,  $\eta = 1 \times 10^{-3}$ ).

Los datos fueron separados en conjuntos de entrenamiento (90 %) y prueba (10 %). Las figuras muestran los datos de prueba.

Los datos fueron estandarizados (media cero y varianza uno). La transformación se calculó usando el conjunto de entrenamiento. Transformar la media de datos esparzos arruina su esparsidad. Todas las dimensiones tienen una varianza similar, no hubiera sido necesario transformarlas, sin embargo los resultados fueron más satisfactorios aplicando preprocesamiento.

La inicialización de los pesos fue al azar, según distribución uniforme en  $[-0,1; 0,1]$ . En cada época las muestras se escogieron al azar. El aprendizaje por lotes de a diez muestras dio resultados similares una muestras por vez.

Se encontró a los datos mejor agrupados en una cantidad *intermedia* de épocas (contrastar figura 1 con 2 para Oja, figuras 3 y 4 para Sanger). A orden de 10 iteraciones se puede encontrar un gran *cluster*, en 100 iteraciones se contabilizan alrededor de seis, en 1000 iteraciones esta cantidad disminuye a tres.

**Regla de Oja**  $T=100$   $\eta=0.0001$

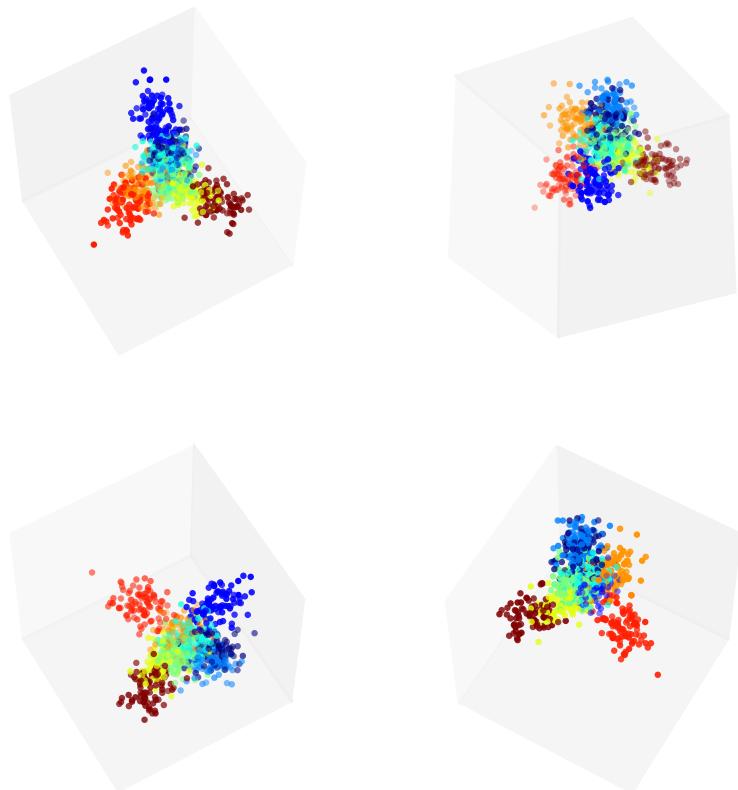


Figura 1: Regla de Oja,  $\eta = 0,0001$ ,  $T = 100$ . Distintas vistas.

Regla de Oja  $T=1000$   $\eta=0.0001$

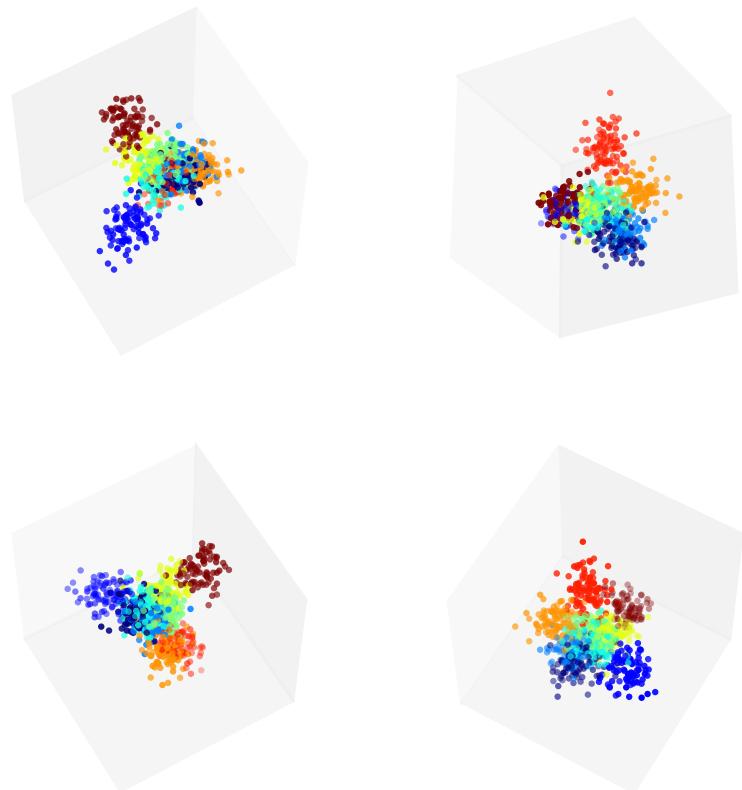


Figura 2: Regla de Oja,  $\eta = 0,0001$ ,  $T = 1000$ . Distintas vistas.

Regla de Sanger  $T=100$   $\eta=0.001$

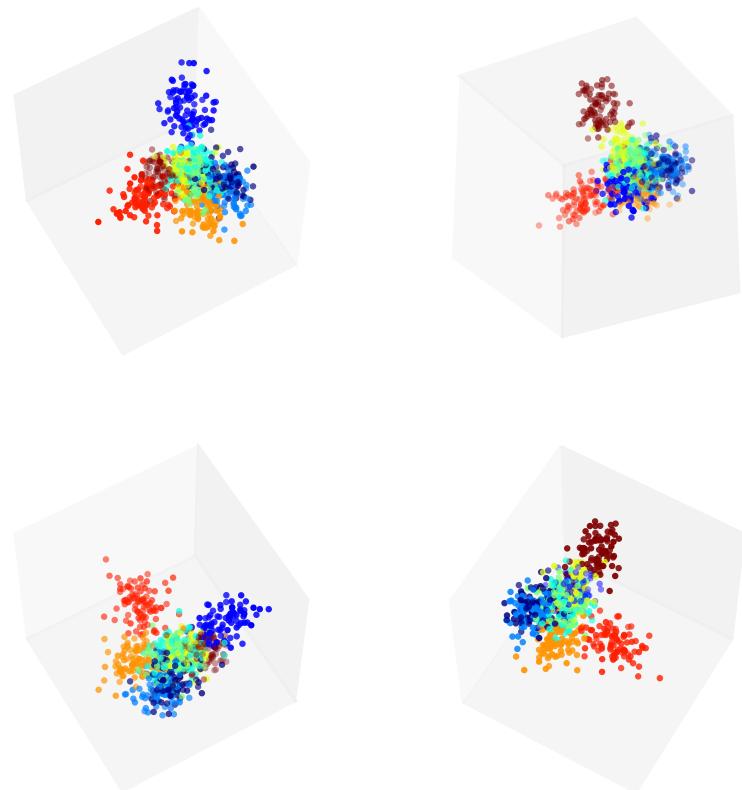


Figura 3: Regla de Sanger,  $\eta = 0,001$ ,  $T = 100$ . Distintas vistas.

**Regla de Sanger**  $T=100$   $\eta=0.0001$

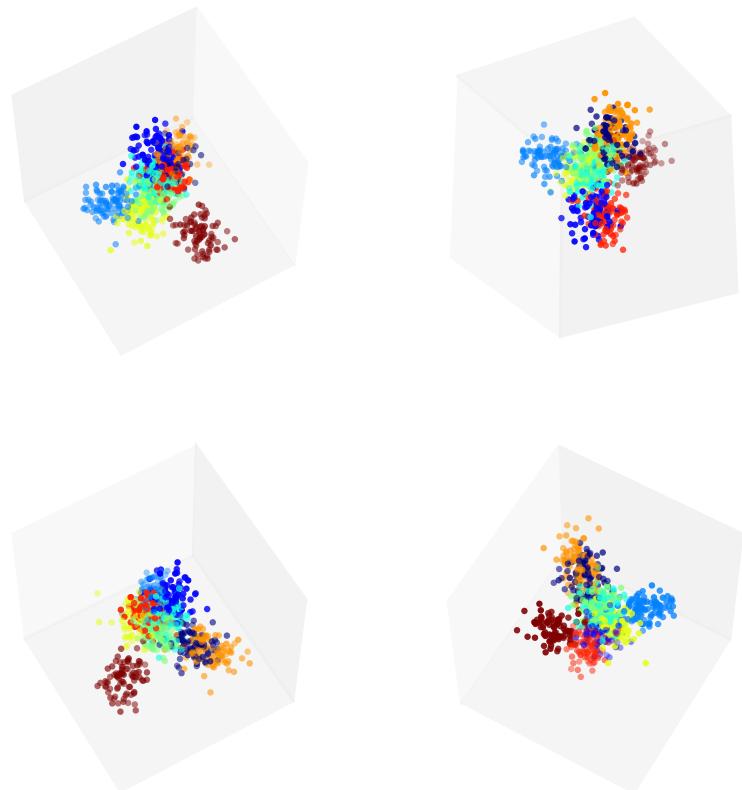


Figura 4: Regla de Sanger,  $\eta = 0,0001$ ,  $T = 100$ . Distintas vistas.

Regla de Sanger  $T=1000$   $\eta=0.0001$

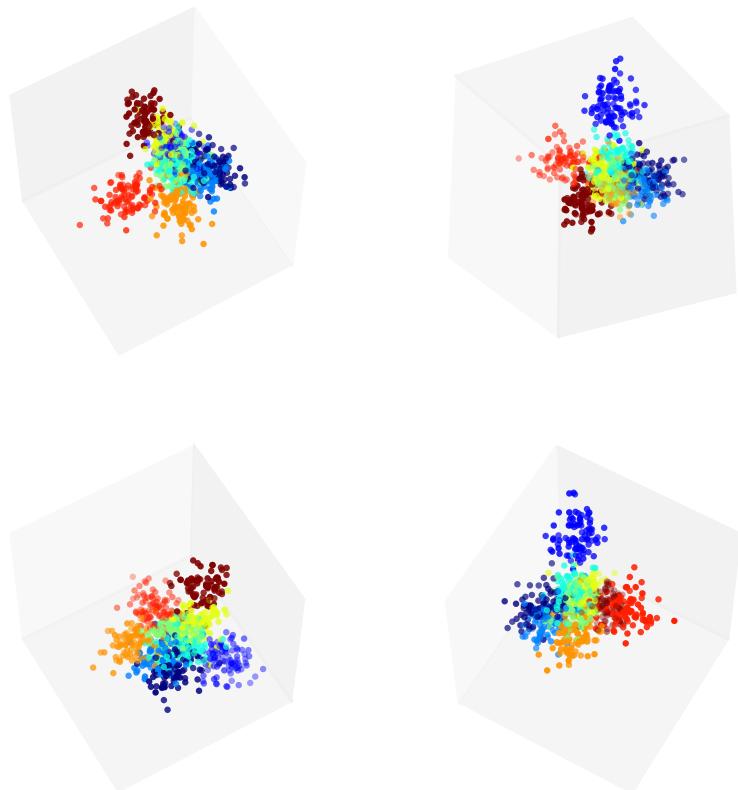


Figura 5: Regla de Sanger,  $\eta = 0,0001$ ,  $T = 1000$ . Distintas vistas.

# Mapas auto-organizados

Los mapa auto-organizados mapean espacios muestrales a espacios de menores dimensiones conservando la topología, a través de un **aprendizaje competitivo**. Como particularidad respecto a otras redes neuronales artificiales, las unidades de salida se disponen bajo cierta geometría. En este trabajo se usó una red cuadrada bidimensional de parámetro  $a = 1$  (distancia vertical y horizontal entre nodos).

Durante el entrenamiento se computa la distancia entre la muestra de ejemplo y todos los vectores de peso—a cada unidad de salida le corresponde un vector de peso. La unidad cuyo vector de peso es más similar a la muestra, *más cercano*, se elige como la *unidad ganadora*  $\hat{y}$ . Luego los pesos se actualizan con la fórmula

$$\Delta w_{ij} = \eta \theta(\mathbf{y}_i, \hat{\mathbf{y}}) (x_j - w_{ij}) \quad (5)$$

donde  $\theta$  es una función que pondera la actualización en base a la posición de la unidad de salida respecto a la de la unidad ganadora *en la red*. La intención es que solo los pesos de la unidad ganadora y su vecindario se ajusten hacia el vector de entrada. Tanto  $\eta$  como  $\theta$  pueden tener una dependencia temporal, en este trabajo  $\eta = \eta_0(T - t)/T$ , con  $\eta_0$  el factor de aprendizaje inicial,  $T$  la cantidad de épocas y  $t$  la época considerada.

La función de vecindario usada fue la curva gaussiana

$$\theta(\mathbf{y}_i, \hat{\mathbf{y}}) = e^{-\frac{1}{2} \frac{|\mathbf{y}_i - \hat{\mathbf{y}}|^2}{\sigma^2}} \quad (6)$$

con desviación estándar  $\sigma$  proporcional al parámetro de red  $a$  según  $\text{FWHM} = 2\sqrt{2 \ln 2}\sigma = ka$ ; de modo de relacionar la anchura a media altura de la curva

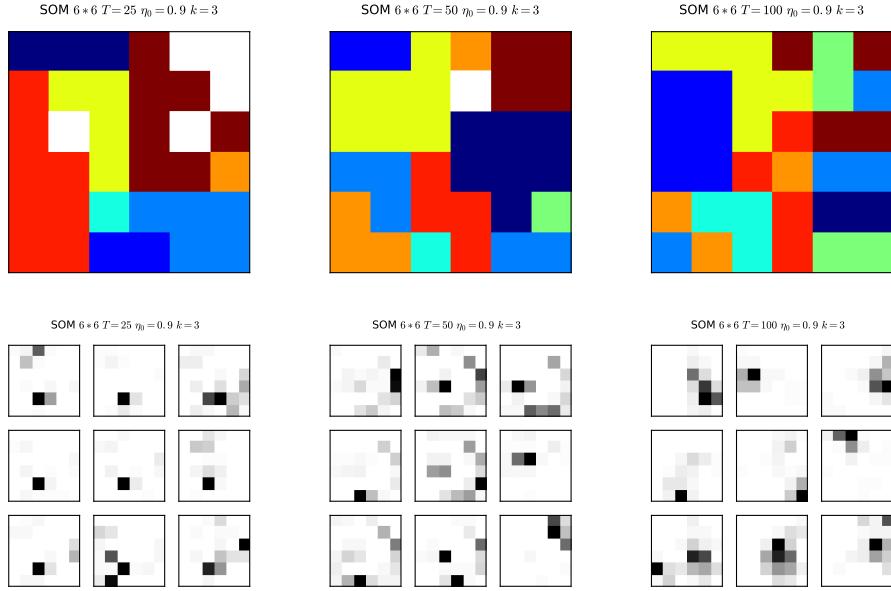


Figura 6: SOM  $6 \times 6$ ,  $\eta_0 = 0,9$ , FWHM = 3,  $T = 100$  a tiempos  $t_1 = 25$  (izquierdo),  $t_2 = 50$  (centro),  $t_3 = 100$  (derecha). Mapas compuestos (arriba) y separados por categoría (abajo).

(FWHM) con  $k$ , una constante de proporcionalidad, y el parámetro de red.

Se usaron dos medidas de distancia: euclidiana y coseno. Cuando la dimensión de los datos es alta (puntualmente, 850 atributos) y además esparza (como sucede cuando los atributos son la ocurrencia de las palabras) la distancia euclidiana deja de ser representativa de la similaridad. La distancia coseno suele funcionar mejor al comparar documentos ya que funciona como una medida de la coincidencia de palabras.

La inicialización de los pesos tiene gran influencia sobre los resultados. Se comenzó con una distribución uniforme en el intervalo  $[-0,1; 0,1]$ ; finalmente se optó por un muestreo de los valores 0 con probabilidad 0,9 y 1 con probabilidad 0,1, que genera vectores de pesos parecidos a las muestras.

Se eligió trabajar con mapas de tamaño  $6 \times 6$  para que en promedio le correspondiera cuatro nodos del mapa a cada una de las nueve categorías. Se exploró el espacio formado por el parámetros cantidad de épocas, factor de aprendizaje inicial, la extensión espacial del vecindario (FWHM). La cantidad de épocas se ajustó para que el mapa alcanzara a extenderse sobre el

espacio muestral, es decir que todos los nodos tuvieran similaridad con al menos un documento, lo que normalmente se consiguió entre 25 y 100 épocas dependiendo de los demás factores; el factor de aprendizaje afectó considerablemente a la cantidad de épocas necesarias para que esto sucediese. Se prefirieron valores entre 1 y 0,5. La extensión del vecindario resultó un parámetro sensible ya que si es pequeña (del orden del parámetro de red) el mapa puede ser incapaz de desenvolverse, en el otro extremo, cuando es grande (del orden del tamaño de la red) el mapa puede no estabilizarse.

En esta sección se utilizó el conjunto de datos sin particionar. El preprocesamiento utilizado en la sección anterior no contribuyó a mejores resultados y se lo terminó por evitar.

En la figura 6 se muestra el progreso de una corrida de 100 épocas, factor de aprendizaje 0,9 y FWHM 3. La figura 7 es similar, se diferencia en que el parámetro FWHM es 6. Las subfiguras de la fila superior muestran, por medio de colores, la categoría mayoritaria de cada nodo de la red. La fila inferior muestra los mapeos separados por categoría. En ambos tipos de diagrama el blanco significa la ausencia de activaciones. Las columnas son estados de la red a  $t_1 = 25$ ,  $t_2 = 50$ ,  $t_3 = 100$  épocas.

En ambas corridas puede observarse que en la etapa más temprana ( $t_1$ ) todas las categorías son acaparadas por un par de nodos. Esto sucede cuando el mapa se inicializa lejos de las muestras: un nodo estará ligeramente más cerca de las muestras que el resto; como el aprendizaje es competitivo se trata del nodo que más se irá ajustando a las muestras. Gracias a la función vecindario, este nodo irá arrastrando consigo a los demás, hasta que otros nodos también comiencen a ser elegidos como ganadores, dando lugar a la etapa siguiente ( $t_2$ ) en la que la red comienza a cubrir el espacio ocupado por las muestras. En la etapa tardía ( $t_3$ ) la red se ajusta más finamente y se estabiliza debido a que el factor de aprendizaje disminuye a cero con el correr de las épocas.

Siguiendo la evolución de los mapeos descompuestos por categoría, puede observarse que comienzan compartiendo algunos pocos nodos y van desdoblándose para ocupar distintas zonas del mapa. Lo ideal sería que las categorías no se solapasen o activasen las mismas unidades, lo que contrariamente

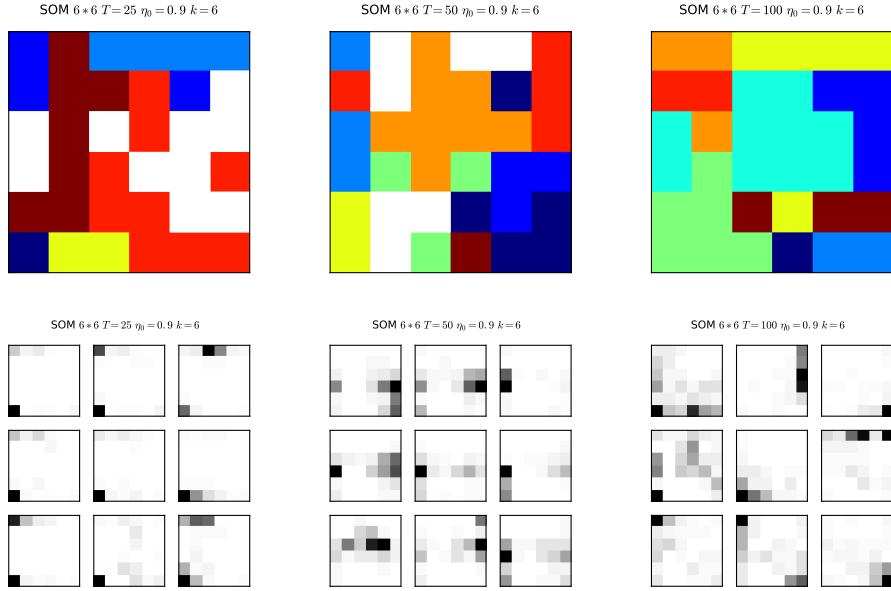


Figura 7: SOM  $6 \times 6$ ,  $\eta_0 = 0,9$ , FWHM = 6,  $T = 100$  a tiempos  $t_1 = 25$  (izquierdo),  $t_2 = 50$  (centro),  $t_3 = 100$  (derecha). Mapas compuestos (arriba) y separados por categoría (abajo).

sucede por falta de resolución del modelo (algoritmos y parámetros escogidos) y también por la propia naturaleza de los datos, ya que si dos categorías son parecidas se espera que ocupen lugares cercanos en el mapa.

## Reducción de dimensionalidad

Se realizaron experimentos aplicando reducción de dimensionalidad como paso previo al mapa auto-organizado. La reducción se logró mediante estandarización seguida por la regla de Sanger con factor de aprendizaje  $\eta = 1 \times 10^{-4}$  y 1000 épocas—los valores que mejor funcionaron en la sección anterior.

Para el mapa en esta ocasión la medida de distancia fue la euclídea, considerando la baja dimensionalidad de los datos transformados. Los parámetros fueron: épocas 100, factor de aprendizaje 0,9 y FWHM 3.

La figura 8 muestra una corrida usando Sanger con 9 unidades de salida,

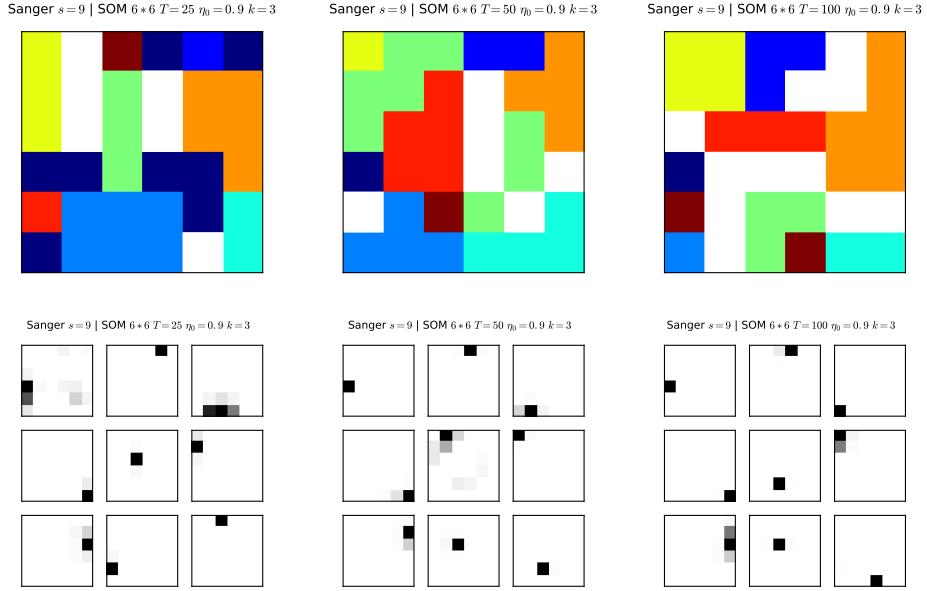


Figura 8: Sanger  $s = 9$ ,  $\eta = 1 \times 10^{-4}$ ,  $T = 1000$  seguido por SOM  $6 \times 6$ ,  $\eta_0 = 0,9$ , FWHM = 3,  $T = 100$  a tiempos  $t_1 = 25$  (izquierda),  $t_2 = 50$  (centro),  $t_3 = 100$  (derecha). Mapas compuestos (arriba) y separados por categoría (abajo).

siguiendo el estilo de las figuras anteriores. Se observa que la activación de unidades por categoría fue mucho más localizada, debido a la *clusterización* de los datos, de manera tal que quedaron unidades sin ser activadas.

Una estrategia que no se implementó y que parece prometedora es usar Sanger no para reducir la dimensionalidad sino para encontrar cierto número de componentes principales y con copias o variantes de las cuales inicializar los pesos del mapa.