

Clasificadores Probabilísticos en Aprendizaje Automático

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Trabajo Práctico

Integrante	LU	Correo electrónico
Lopez Valiente, Patricio	457/15	patriciolopezvaliente@gmail.com

Reservado para la catedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

Índice

1. Introducción	3
2. Desarrollo	3
2.1. Técnicas utilizadas	3
2.1.1. Aprendizaje Hebbiano	3
2.1.2. Mapas Auto Organizados	4
2.2. Detalles de Uso	5
3. Experimentos	5
3.1. Aprendizaje Hebbiano	5
3.2. Mapas Auto Organizados	13

1. Introducción

En este trabajo presentaremos técnicas de aprendizaje no supervisado como el Aprendizaje Hebbiano y los mapas auto organizados. Dentro de las muchas aplicaciones que poseen estos métodos, se analizara su utilidad a la hora de clasificar datos. Estos métodos, en particular el método SOM, están ampliamente difundidos por su capacidad de adaptarse a una gran cantidad de problemas sin requerir prácticamente trabajo humano, es decir, no requieren gran trabajo de código.

El objetivo principal de este informe es analizar los métodos de Aprendizaje no supervisado utilizados para clasificar nuestros datos.

2. Desarrollo

2.1. Técnicas utilizadas

2.1.1. Aprendizaje Hebbiano

Esta regla de aprendizaje es la base de muchas otras, en particular en este trabajo se utilizaron las variantes de Oja y Sanger, con estas reglas se busca extraer características de los datos de entrada.

El fundamento de la regla de Hebb es que si dos neuronas N_i y N_j toman el mismo estado simultáneamente, el peso de la conexión entre ambas se incrementa. Esto puede explicarse porque la regla de aprendizaje de Hebb se originó a partir de la neurona biológica clásica, que solamente puede tener dos estados: activa o inactiva.

El aprendizaje Hebbiano consiste básicamente en el ajuste de los pesos de las conexiones, de acuerdo con la correlación de los valores de activación de las dos neuronas conectadas. Este algoritmo pretende medirlas características de los datos de entrada, cuando un peso contribuye en la activación de una neurona, el peso se incrementa. Y si contribuye a la inhibición éste se decremente. De forma práctica si las dos unidades son activas (salida positiva), se produce un reforzamiento de la conexión. Si por el contrario, una es activa y la otra pasiva (salida negativa), se produce un debilitamiento de la conexión. Por tanto, la modificación de los pesos se realiza en función de las salidas de las neuronas, obtenidos tras ingresar cierta entrada.

Las dos reglas de actualización utilizadas son:

- Oja: $\Delta W_{ij} = \eta y_i(x_j - \sum_{k=1}^M y_k W_{kj})$
- Sanger: $\Delta W_{ij} = \eta y_i(x_j - \sum_{k=1}^i y_k W_{kj})$

Las dos reglas fueron implementadas en forma matricial, de la siguiente forma:

- Oja: $\Delta W = \eta Y(X^T - Y^T W)$
- Sanger: $\Delta W = \eta(YX^T - LT(YY^T)W)$, donde LT es la función que asigna 0 a todos los elementos sobre la diagonal.

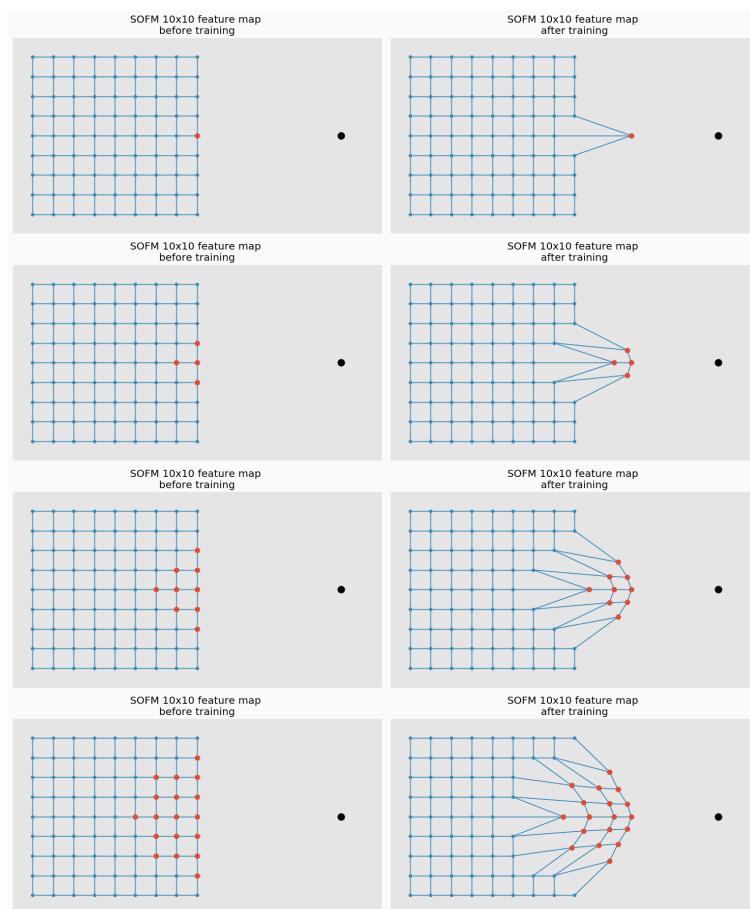
2.1.2. Mapas Auto Organizados

La idea principal de los Mapas Auto Organizados(SOM) consiste en tener una grilla de neuronas, las cuales están son activadas por las unidades de entradas, la neurona activada(la ganadora) es la que se encuentra a menor distancia de la entrada, esto lo consigue obteniendo $\min_j(x, W_j)$, luego empuja la unidad ganadora hacia la entrada, los SOM ademas poseen el concepto de vecindad, por lo que las neuronas que se encuentran en dentro de un determinado radio, con centro en la ganadora también son empujadas hacia la entrada, el factor de empuje es proporcional a la proximidad a la ganadora, y es siempre menor al de la ganadora.

Independientemente del problema a tratar las SOM se componen de grillas, arreglos de neuronas, conectadas siguiendo alguna forma geométrica, generalmente rectangulares o hexagonales, en este trabajo se utilizaron rectangulares.

Luego en la fase de entrenamiento se introducen entradas, se encuentra la neurona ganadora, la mas cercana a la entrada, y se procede a actualizar los pesos de la ganadora y las neuronas vecinas, de acuerdo al radio y factor de aprendizaje de la vecindad, el cual depende de la proximidad a la ganadora.

Figura 1. Actualización pesos por radio



En la [Figura 1](#) se ve como se actualizan los pesos de la grilla en función del radio

de la vecindad. En esta vemos que mientras mayor es la vecindad mayor es impacto en la topología de la red. Es por esto que el factor de aprendizaje y la vecindad deben ser decrecientes con respecto a las épocas transcurridas, ya que como se realiza un mapa topológico, los ajustes son cada vez más finos, por lo tanto si se mantuvieran estables, podrían generar inestabilidad.

La reducción de factor de aprendizaje se produce al terminar una época de entrenamiento. Además el radio de vecindad inicial, es reducido cada una determinada cantidad de épocas, en este trabajo cada 200 épocas. Continuando, el factor de aprendizaje de la vecindad también se reduce en función de las épocas transcurridas.

2.2. Detalles de Uso

La implementación de los Algoritmos realizados fue hecha en python3, además se provee un README con detalle de los módulos necesarios para su ejecución y información de las funcionalidades provistas.

3. Experimentos

Para la fase de experimentación se utilizaron todos los datos provistos por la cátedra, de los cuales el 10 % se los utilizó para validación. Además se utilizó colores para representar a las categorías, de la forma:

Color de categoría									
C1	C2	C3	C4	C5	C6	C7	C8	C9	
Rojo	Azul	Verde	Violeta	Naranja	Amarillo	Marrón	Rosa	Gris Oscuro	

3.1. Aprendizaje Hebbiano

Para este problema se experimentó con Redes Hebbianas, tanto con la regla de Oja, como la de Sanger. Además se experimentó con los parámetros de la red de la siguiente forma:

- $\eta = 0,1$ y $\eta = 0,9$.
- 100 y 1000 épocas.
- Se utilizaron Pesos iniciales uniformemente generados con valores comprendidos entre $-0,1$ y $0,1$.

Se representó la clasificación obtenida en gráficos 3D en distintas vistas para cada combinación de parámetros, señalando en cada caso la categoría de la entrada con el color correspondiente. Los resultados obtenidos fueron los siguientes:

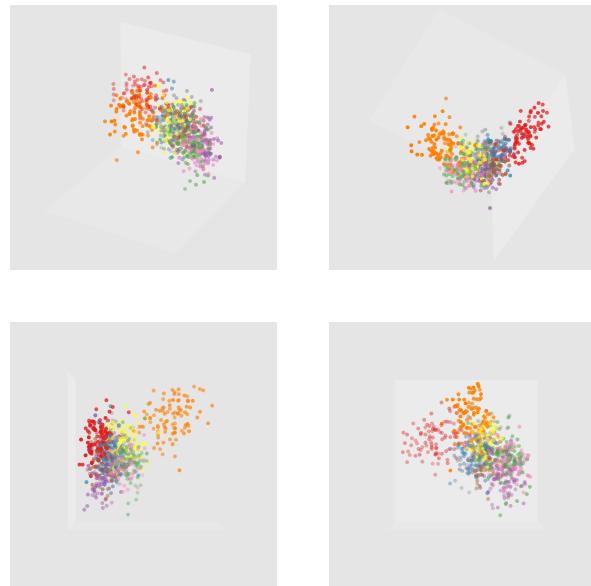
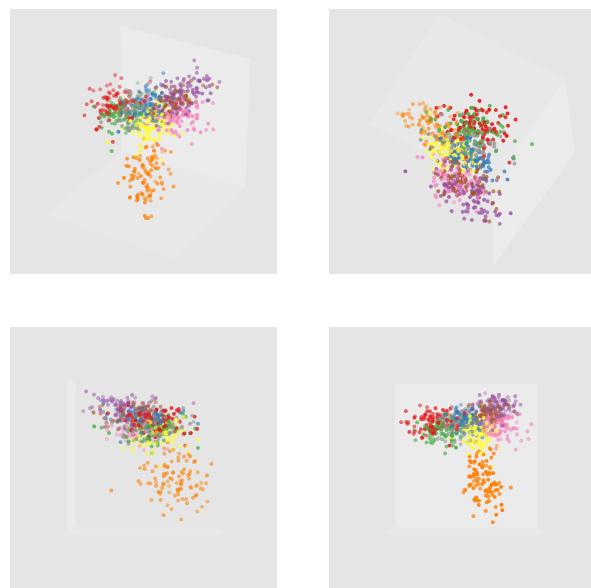
Figura 2. $\eta = 0,001$, 100 épocas, Oja**Figura 3.** $\eta = 0,001$, 1000 épocas, Oja

Figura 4. $\eta = 0,0001$, 100 épocas, Oja

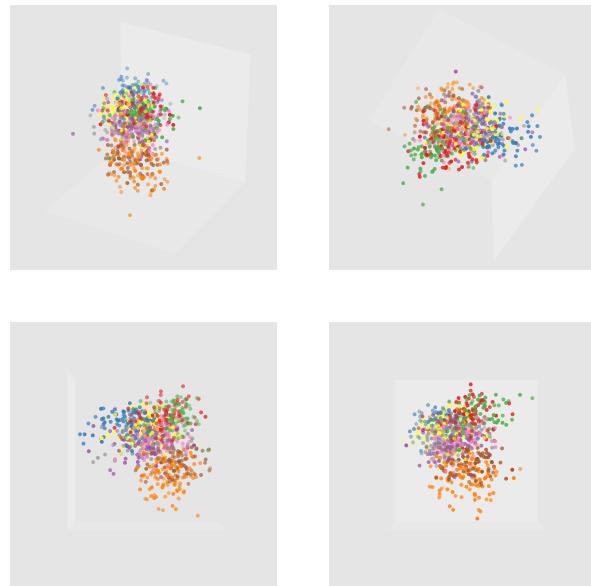
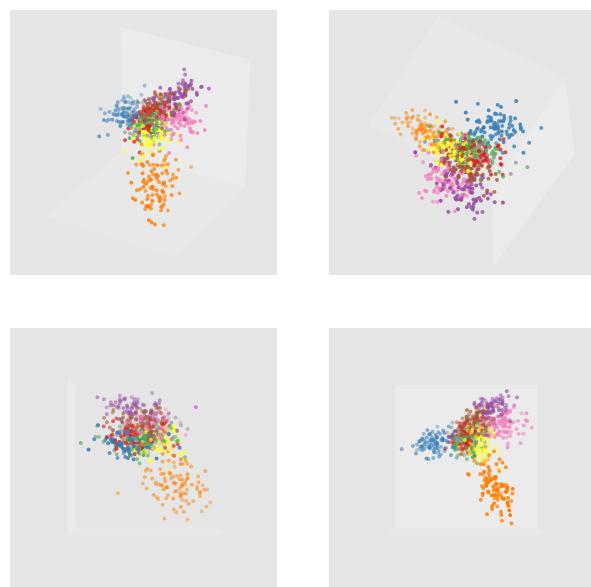


Figura 5. $\eta = 0,0001$, 1000 épocas, Oja



En la Figura 2 a la Figura 5 vemos la clasificación del espacio utilizando la

regla de Oja, se puede apreciar como con un mayor numero de épocas los datos se diferencias mas, es decir el lugar asignado en el espacio vectorial es mas claro para cada categoría, por lo tanto las áreas de cada categoría se hacen mas visibles, ademas se puede apreciar como con un η menor se logra una mejor clasificación, aunque este parámetro, impacta en bastante menor medida que las épocas.

Figura 6. $\eta = 0,001$, 100 épocas, Sanger

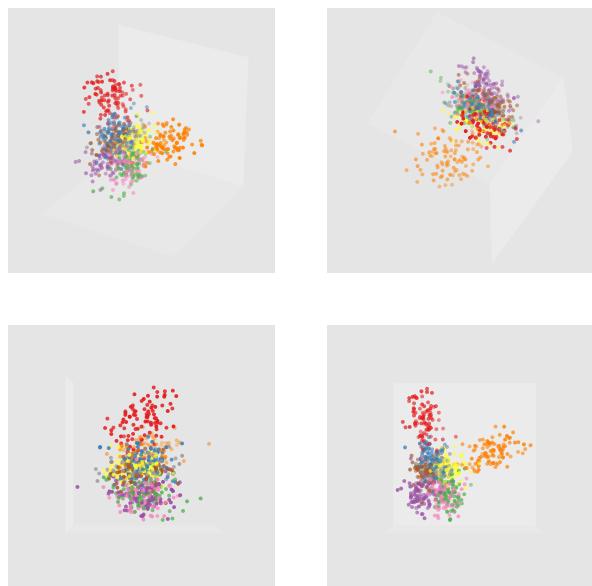


Figura 7. $\eta = 0,001$, 1000 épocas, Sanger

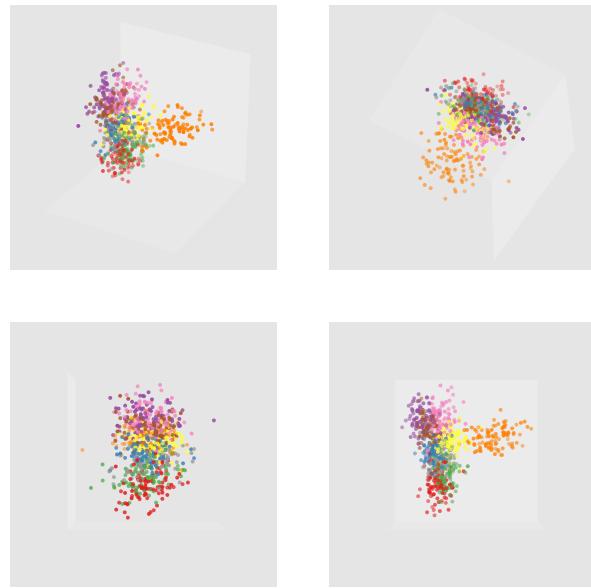


Figura 8. $\eta = 0,0001$, 100 épocas, Sanger

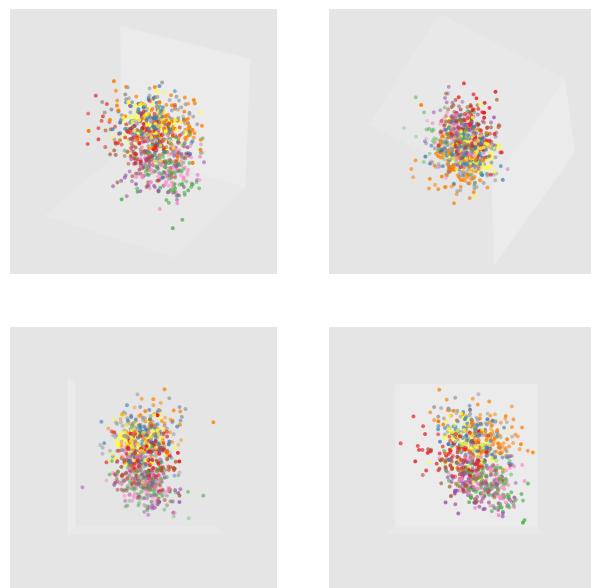
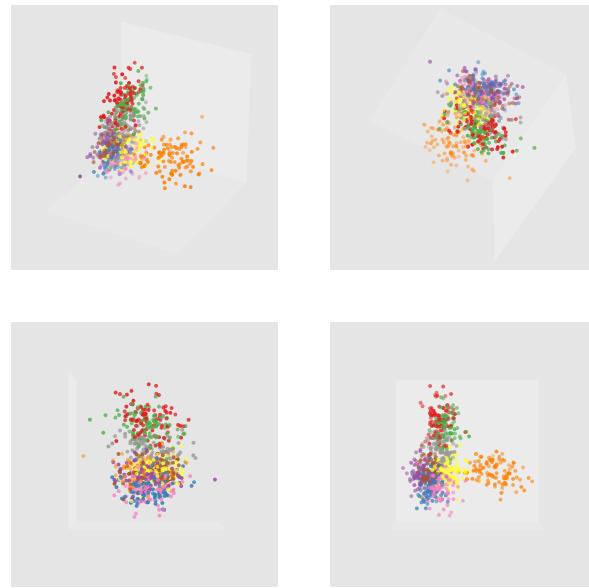


Figura 9. $\eta = 0,0001$, 1000 épocas, Sanger



En la [Figura 6](#) a la [Figura 9](#) vemos la clasificación del espacio utilizando la regla de Sanger, igualmente que con la regla de Oja se puede apreciar como con un mayor numero de épocas los datos se diferencian mas. En el caso del η , para el caso $\eta = 0,0001$ utilizando 1000 épocas se aprecia una leve mejora con respecto al η mayor y mismo numero de épocas, ya que parece diferenciar un poco mas el espacio. Sin embargo utilizando 100 épocas, la disminución del η parece ser perjudicial para la caracterización del espacio.

En la [Figura 10](#) a la [Figura 13](#) se muestran las proyecciones de los datos de Testing:

Figura 10. $\eta = 0,001$, 1000 épocas, Oja, Testing Set

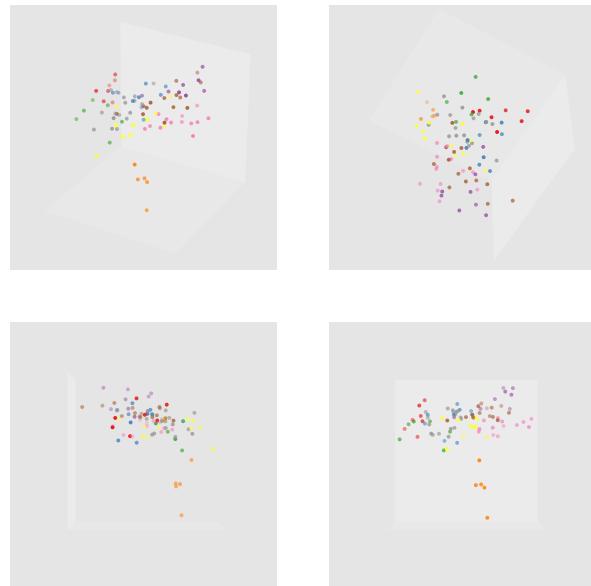


Figura 11. $\eta = 0,0001$, 1000 épocas, Oja, Testing Set

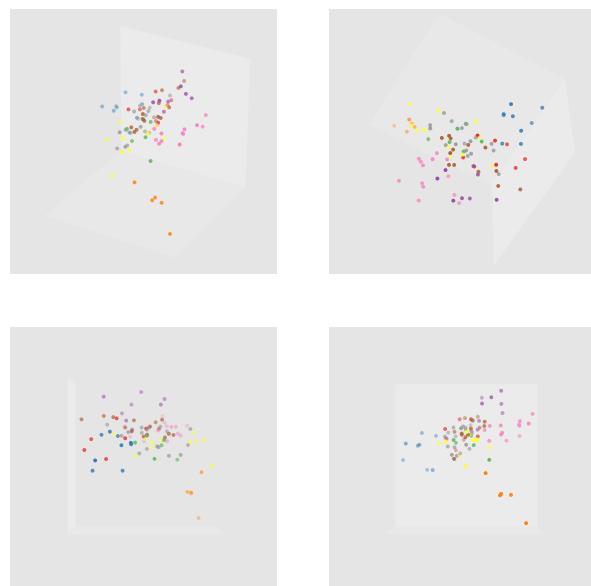
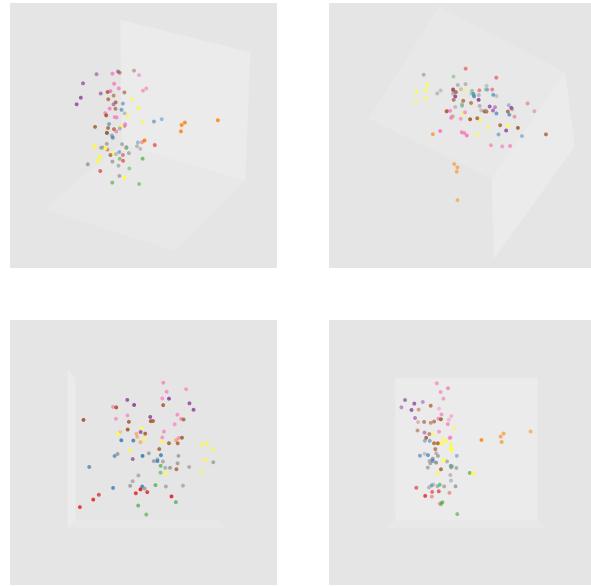
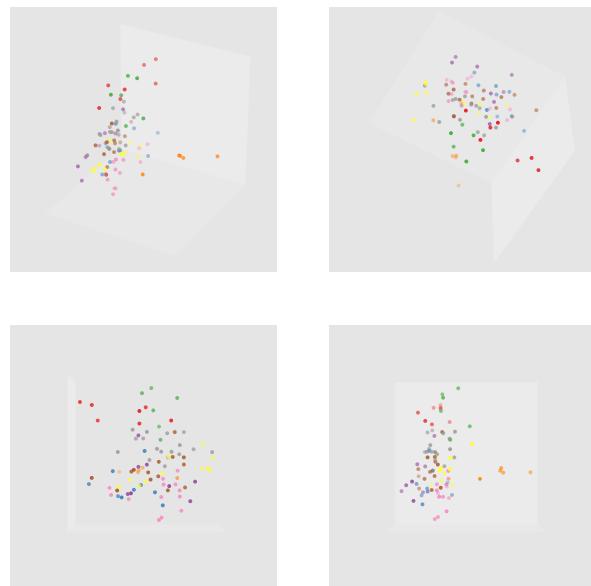


Figura 12. $\eta = 0,001$, 1000 épocas, Sanger, Testing Set**Figura 13.** $\eta = 0,0001$, 1000 épocas, Sanger, Testing Set

Estos modelos parecen buscar las características principales que componen las

muestras y proyectarlas de tal forma que el nuevo espacio generado que se diferencien mejor. Si tuviéramos que comprimir la información de las muestras o reducir de dimensión las entradas por limitaciones en tiempos de ejecución, estas dos técnicas serían dos buenas candidatas, ya que permitirían reducir las dimensiones a gusto, capturando las características de la muestra.

3.2. Mapas Auto Organizados

Para este problema se experimento con redes SOM, utilizando como grilla control $Gr = (6x6)$, con los siguientes parámetros y modelos:

- $\eta = 0,1$ y $\eta = 0,9$.
- Radio inicial $R = 0$ y $R = 3$.
- Utilizando las entradas(n) default, sus proyecciones sobre las primeras 3 componentes principales, y sus proyecciones sobre las primeras 9 componentes principales.
- Grillas de ($3x3, 6x6$ y $9x9$) para $n \in 3, 9, 850$, $\eta = 0,1$ y $R = 3$. Ademas se experimento sobre los tiempos de ejecución de estas configuraciones.

Se representó la clasificación obtenida Mostrando la categoría que mas activa cada neurona de la Grilla de salida para cada combinación de parámetros, señalando en cada caso la categoría de la entrada con el color correspondiente. Los resultados obtenidos fueron los siguientes:

Figura 14. $Grilla = 6x6, n = 850$



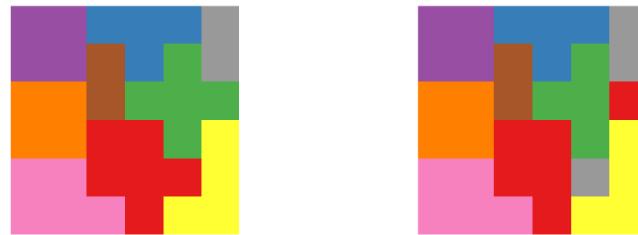
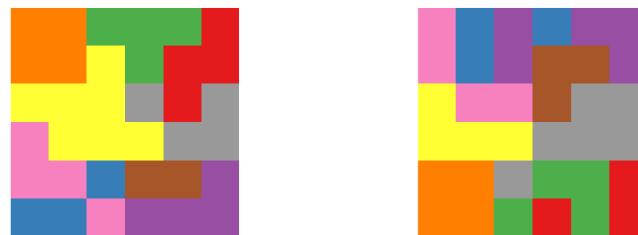
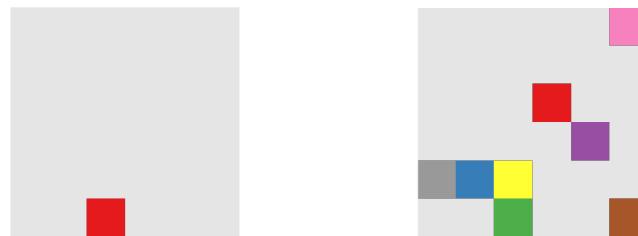
(a) $\eta = 0,1$, R=0

(b) $\eta = 0,9$, R=0



(c) $\eta = 0,1$, R=3

(d) $\eta = 0,9$, R=3

Figura 15. $Grilla = 6x6, n = 3$ (a) $\eta = 0,1, R=0$ (b) $\eta = 0,9, R=0$ (c) $\eta = 0,1, R=3$ (d) $\eta = 0,9, R=3$ **Figura 16.** $Grilla = 6x6, n = 9$ (a) $\eta = 0,1, R=0$ (b) $\eta = 0,9, R=0$ (c) $\eta = 0,1, R=3$ (d) $\eta = 0,9, R=3$

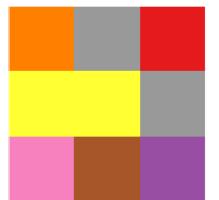
En la [Figura 14](#) a la [Figura 16](#) vemos las Topologías obtenidas para las distintas configuraciones de la Grilla control de $6x6$, estas muestran se pueden interpretar de

forma topologica en primera instancia y de esta forma ver las distribuciones tomadas por las categorías y las relaciones entre ellas, de esta forma se puede ver que hay categorías mas fuertemente vinculadas que otras, ya que habitualmente comparten frontera, por lo que se puede suponer por ejemplo que se refieren a tópicos con similitudes, por ejemplo la Categoría Roja comparte prácticamente siempre frontera con la Categoría Verde y la Marrón.

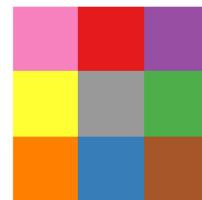
Esta forma de representar las categorías parece ser mas fiable que la del experimento 1, ya que aplica el concepto de vecindad, por lo que uno podría decir que una muestra nueva que activa una determinada neurona debe pertenecer a la categoría de mayor activación de esa neurona, que pertenece a alguna de las categorías de frontera inmediata de la neurona activada en menor probabilidad, o que es una categoría distinta a las de muestra que tiene fuerte relación con la categoría de mayor activación de la neurona activada y las de frontera inmediata de esta.

También se observa que la configuracion mas estable a la hora de "mapear" las categorías en todos los casos es $\eta = 0,1$, $R = 3$.

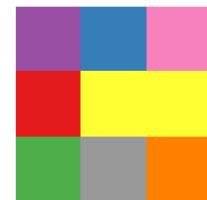
Figura 17. $\eta = 0,1$, $R=3$



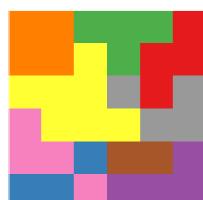
(a) $Gr = 3 \times 3$, $n = 3$



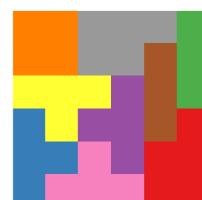
(b) $Gr = 3 \times 3$, $n = 9$



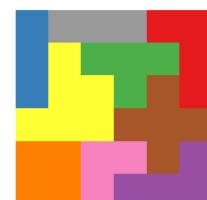
(c) $Gr = 3 \times 3$, $n = 850$



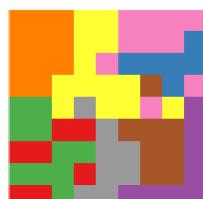
(d) $Gr = 6 \times 6$, $n = 3$



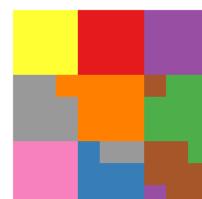
(e) $Gr = 6 \times 6$, $n = 9$



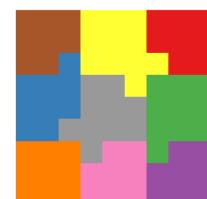
(f) $Gr = 6 \times 6$, $n = 850$



(g) $Gr = 9 \times 9$, $n = 3$



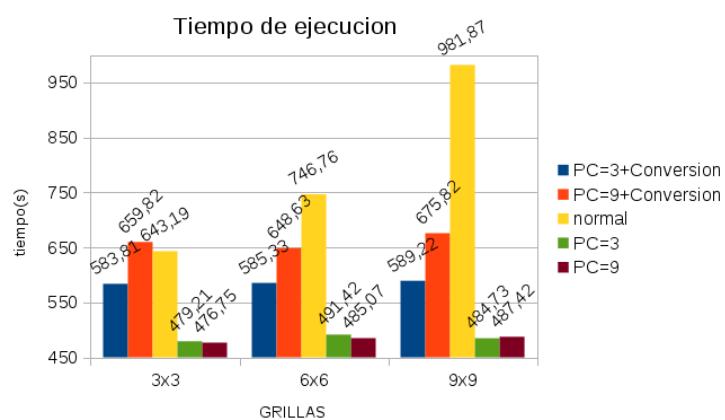
(h) $Gr = 9 \times 9$, $n = 9$



(i) $Gr = 9 \times 9$, $n = 850$

En la [Figura 17](#) vemos como impacta el tamaño de la grilla sobre la topología obtenida, se puede apreciar como al utilizar la grilla de 3×3 la clasificación se simplifica demasiado y parece perderse información sobre las relaciones entre las distintas categorías. En el caso de la grilla de 9×9 en las redes de $n = 9$ y $n = 850$ parece tener un efecto adverso. Sin embargo para $n = 3$ pareciera que como la grilla posee mayor granularidad, las fronteras entre categorías se vuelven mas específicas. En el caso de la grilla 6×6 es la que se comporta de manera mas estable para las distintas configuraciones, por lo que junto con $\eta = 0,1$ y $R = 3$ son los candidatos a parámetros de entrenamiento mas eficaces.

Figura 18. Tiempos de Training



En la [Figura 18](#) se muestra los tiempos de ejecución de las configuraciones utilizadas en la [Figura 17](#), considerando en $n = 3$ y $n = 9$ los costos de obtener las Componentes Principales(PC) y convertir la muestra, y por otra parte solo considerando la fase de entrenamiento, es decir sin considerar costo de obtener las Componentes Principales(PC) y convertir la muestra.

Podemos ver como los tiempos de ejecución crecen rápidamente a medida que aumenta el tamaño de la grilla para $n = 850$, por lo que para un n "grande" el tamaño de la grilla es limitante.

El resto de las mediciones se mantienen uniformes independientemente del tamaño de la grilla. Ademas se aprecia que el proceso de obtener las PC toma cerca de 30 % del tiempo de ejecución en todos los casos.

Es fácil ver que si la grilla a utilizar es pequeña($3 \times 3, 6 \times 6$) es prácticamente indiferente la cantidad de entradas a usar, y debería solo esto depender de la eficacia de la Red, en el caso de grillas "grandes", el tiempo de ejecución empieza a ser un factor a considerar. Por lo que es mejor utilizar un n pequeño, inclusive si significa pagar los costos de conversión del espacio.

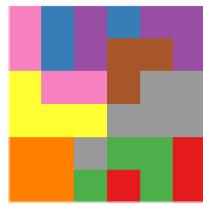
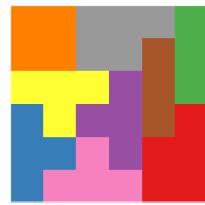
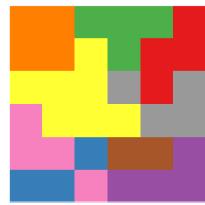
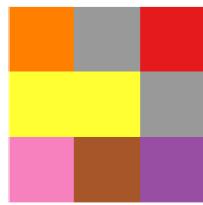
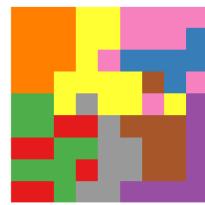
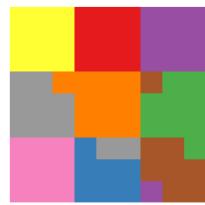
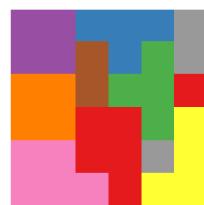
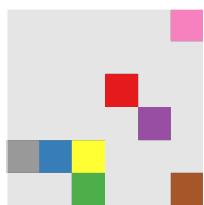
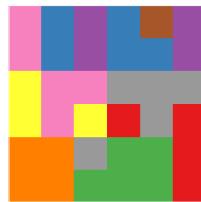
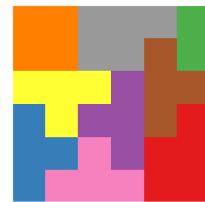
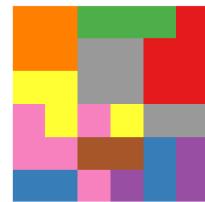
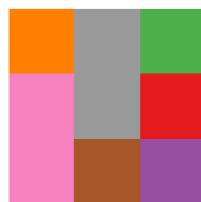
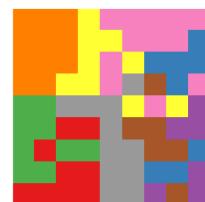
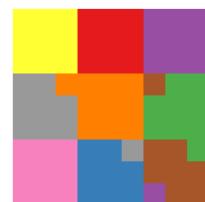
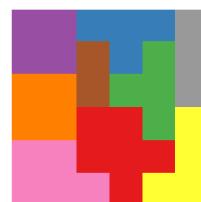
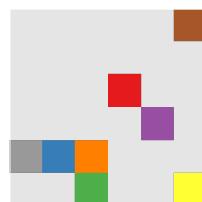
Figura 19. Training Set(a) $Gr = 6 \times 6$, $n = 3$, $\eta = 0,9$, R=3(b) $Gr = 6 \times 6$, $n = 9$, $\eta = 0,1$, R=3(c) $Gr = 6 \times 6$, $n = 3$, $\eta = 0,1$, R=3(d) $Gr = 3 \times 3$, $n = 3$, $\eta = 0,1$, R=3(e) $Gr = 9 \times 9$, $n = 3$, $\eta = 0,1$, R=3(f) $Gr = 9 \times 9$, $n = 9$, $\eta = 0,1$, R=3(g) $Gr = 6 \times 6$, $n = 3$, $\eta = 0,9$, R=0(h) $Gr = 6 \times 6$, $n = 9$, $\eta = 0,9$, R=0

Figura 20. Testing Set(a) $Gr = 6 \times 6$, $n = 3$, $\eta = 0,9$, R=3(b) $Gr = 6 \times 6$, $n = 9$, $\eta = 0,1$, R=3(c) $Gr = 6 \times 6$, $n = 3$, $\eta = 0,1$, R=3(d) $Gr = 3 \times 3$, $n = 3$, $\eta = 0,1$, R=3(e) $Gr = 9 \times 9$, $n = 3$, $\eta = 0,1$, R=3(f) $Gr = 9 \times 9$, $n = 9$, $\eta = 0,1$, R=3(g) $Gr = 6 \times 6$, $n = 3$, $\eta = 0,9$, R=0(h) $Gr = 6 \times 6$, $n = 9$, $\eta = 0,9$, R=0

Por ultimo en la [Figura 19](#) y [Figura 20](#) vemos las únicas configuraciones que evidenciaron diferencias entre el set de Training y el set de Testing, como se puede apreciar fácilmente las diferencias son mínimas, y se puede deber a la distribución de las muestras. Ademas ya que la gran mayoría de las configuraciones no evidenciaron diferencias, se podría decir que son diferencias suficientemente pequeñas como para señalar que el modelo SOM se comporto de la misma manera para ambos Sets, por lo que se esperaría un comportamiento similar para muestras foráneas, esto claramente toma como suposición que los datos provistos por la cátedra son una muestra representativa.