

# Idea de los Mapas Autoorganizados

# Plan de Ataque

# Plan de Ataque

---

En esta sección aprenderemos:

- ¿Cómo funcionan los Mapas Autoorganizados?
- Clustering con K-Means
- ¿Cómo aprenden los Mapas Autoorganizados? (Parte 1)
- ¿Cómo aprenden los Mapas Autoorganizados? (Parte 2)
- Ejemplo real de SOM
- Lecturas acerca de SOM Avanzadas

# ¿Cómo funcionan los SOM?

# Mapas Autoorganizados

No Supervisada	Redes neuronales artificiales	Utilizado para regresión y clasificación
✓	Redes neuronales convolucionales	Utilizado para la visión por ordenador
✓	Redes neuronales recurrentes	Utilizado para el análisis de series temporales
→	Mapas autoorganizados	Utilizado para la detección de características
	Máquinas de Boltzmann Profundas	Utilizado para los sistemas de recomendación
	AutoEncoders	Utilizado para los sistemas de recomendación

# ¿Cómo funcionan los SOM?



Image Source: aka.fi

# ¿Cómo funcionan los SOM?

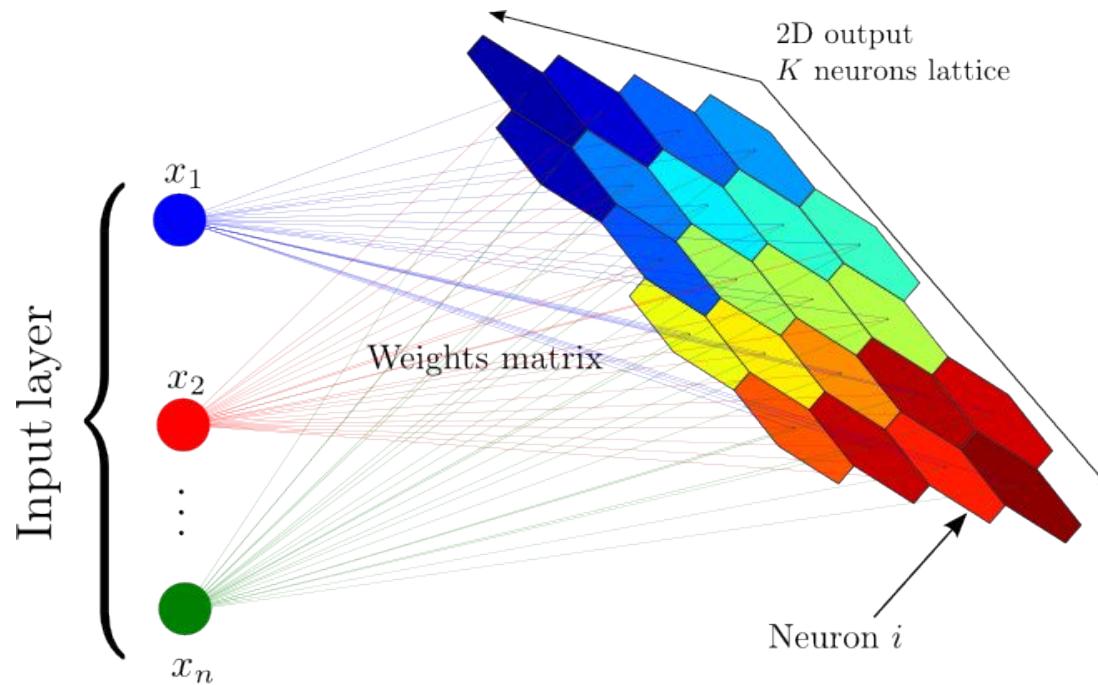


Image Adapted From: [arxiv.org/pdf/1312.5753.pdf](https://arxiv.org/pdf/1312.5753.pdf)

# ¿Cómo funcionan los SOM?

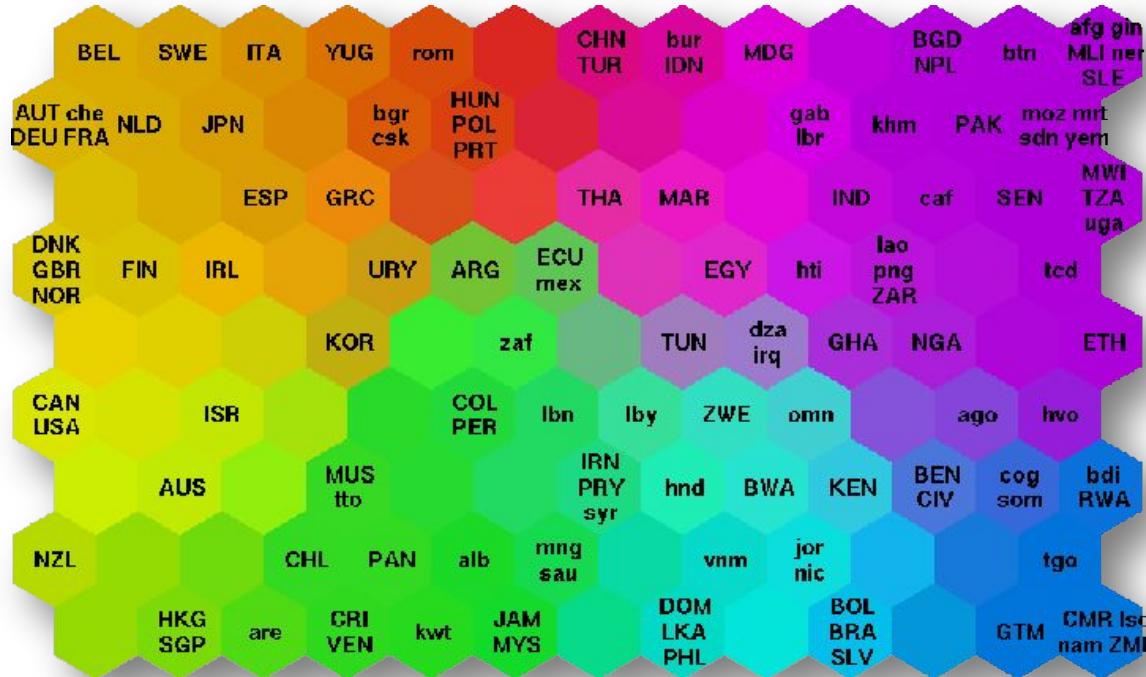
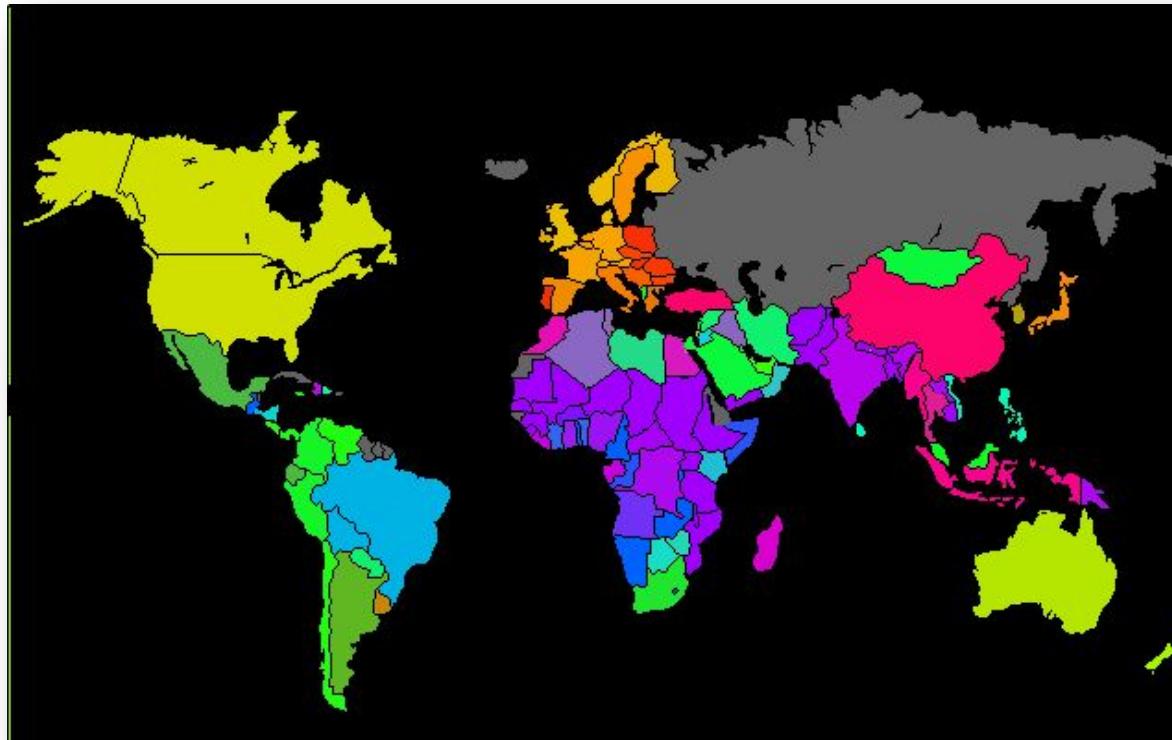


Image Source: cis.hut.fi

# ¿Cómo funcionan los SOM?



*Image Source: cis.hut.fi*

# ¿Cómo funcionan los SOM?

	A	B	C	D	E
1	Country	Country C	Health Ex	Education	Inflation
2	Aruba	ABW	9.418971	5.92467022	-2.13637
3	Afghanist	AFG	4.371774		-8.28308
4	Angola	AGO	5.791339		13.73145
5	Albania	ALB	6.75969		2.280502
6	Andorra	AND	4.57058	3.1638701	
7	Arab Wor	ARB	4.049924		3.524814
8	United Ar	ARE	7.634758		
9	Argentina	ARG	4.545323	4.88997984	6.282774
10	Armenia	ARM		3.84079003	3.406767
11	American	ASM	4.862062		
12	Antigua a	ATG	9.046056	2.55447006	-0.55016
13	Australia	AUS	11.19444	5.09262991	1.820112
14	Austria	AUT	5.85024	5.7674098	0.506313
15	Azerbaija	AZE	6.964187	3.22430992	1.401056
16	Burundi	BDI	10.39434	6.3197999	10.98147
17	Belgium	BEL	4.46431	6.41535997	-0.05315
18	Benin	BEN	7.405431	4.22204018	2.15683

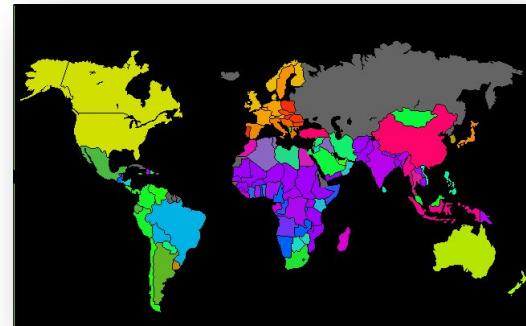
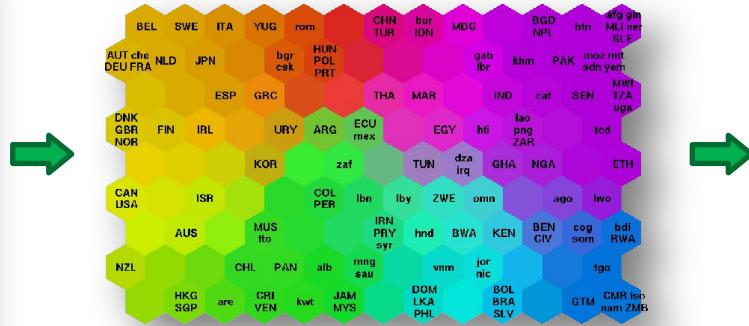


Image Source: cis.hut.fi

# ¿Cómo funcionan los SOM?

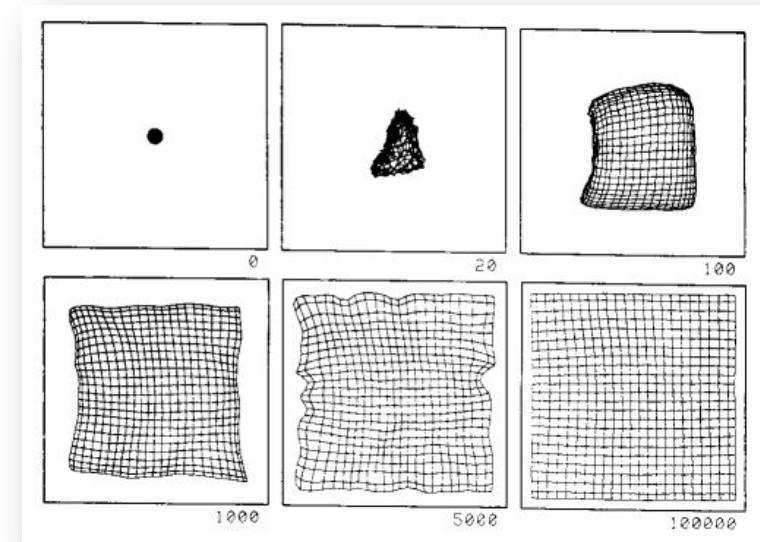
Lecturas Adicionales:

*The Self-Organizing Map*

de Tuevo Kohonen (1990)

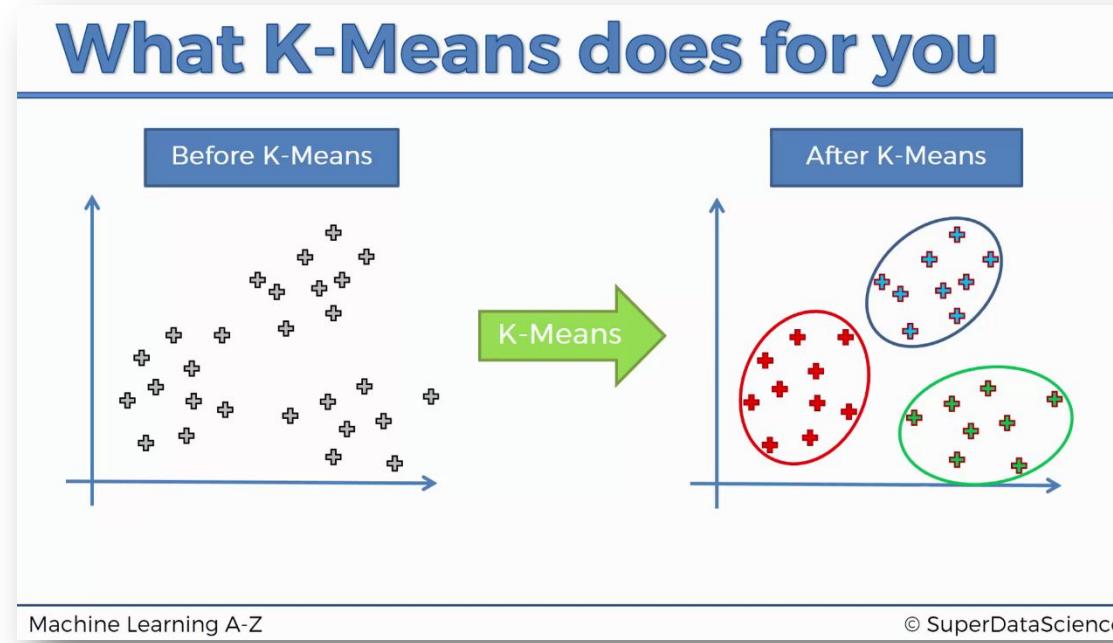
Link:

<http://sci2s.ugr.es/keel/pdf/algorithm/articulo/1990-Kohonen-PIEEE.pdf>



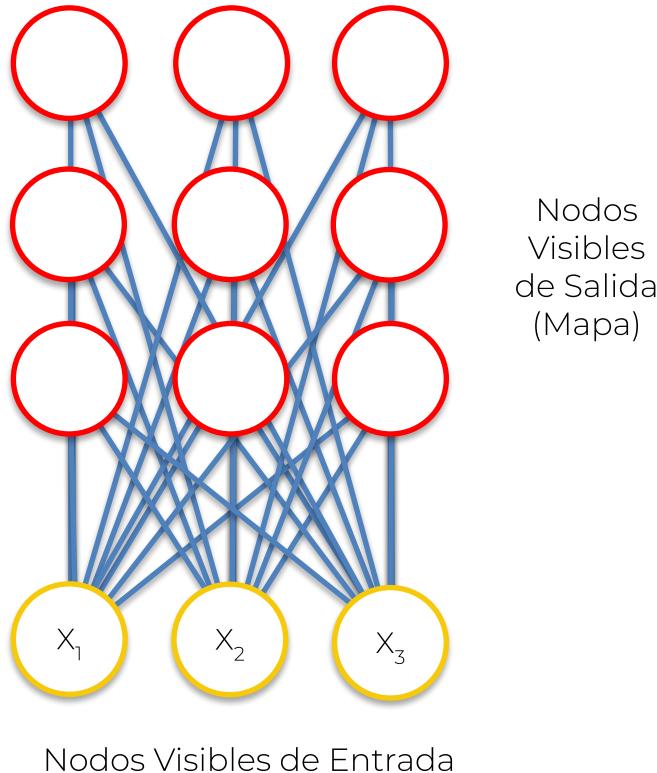
# ¿Por qué necesitamos revisar K-means?

# ¿Por qué revisar K-means?

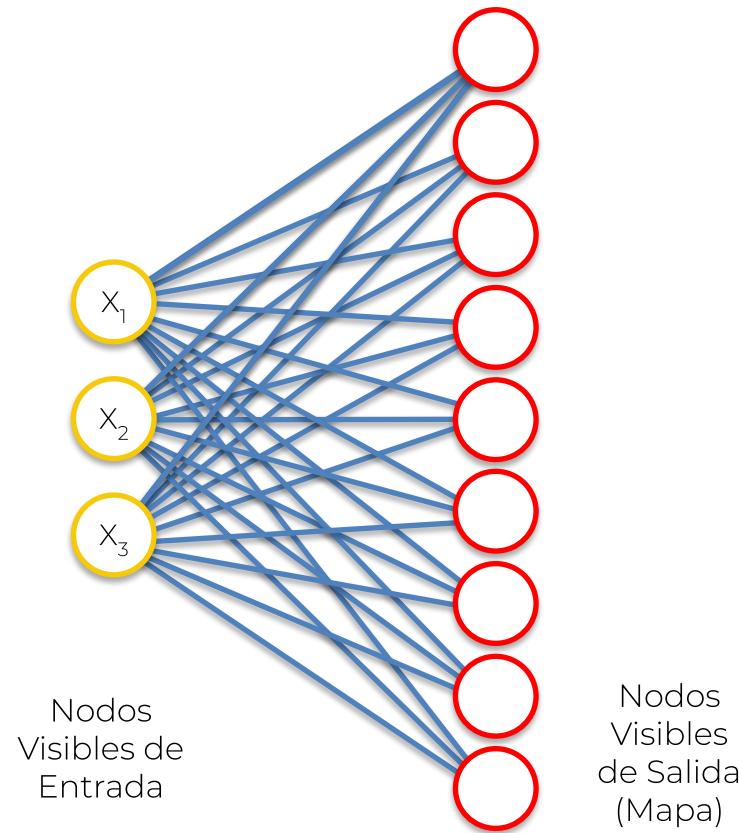


# ¿Cómo aprenden los SOM?

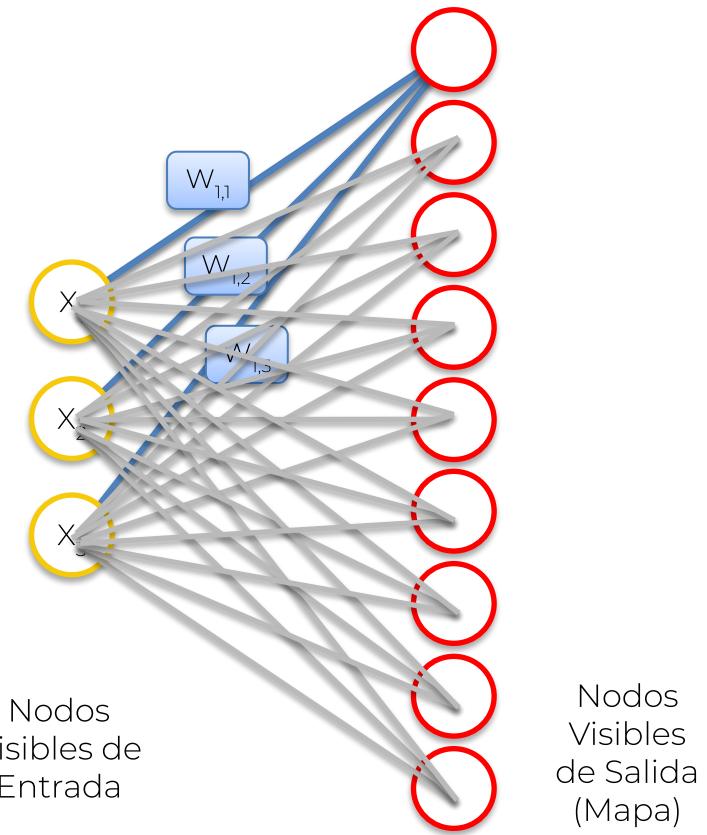
# ¿Cómo aprenden los SOM?



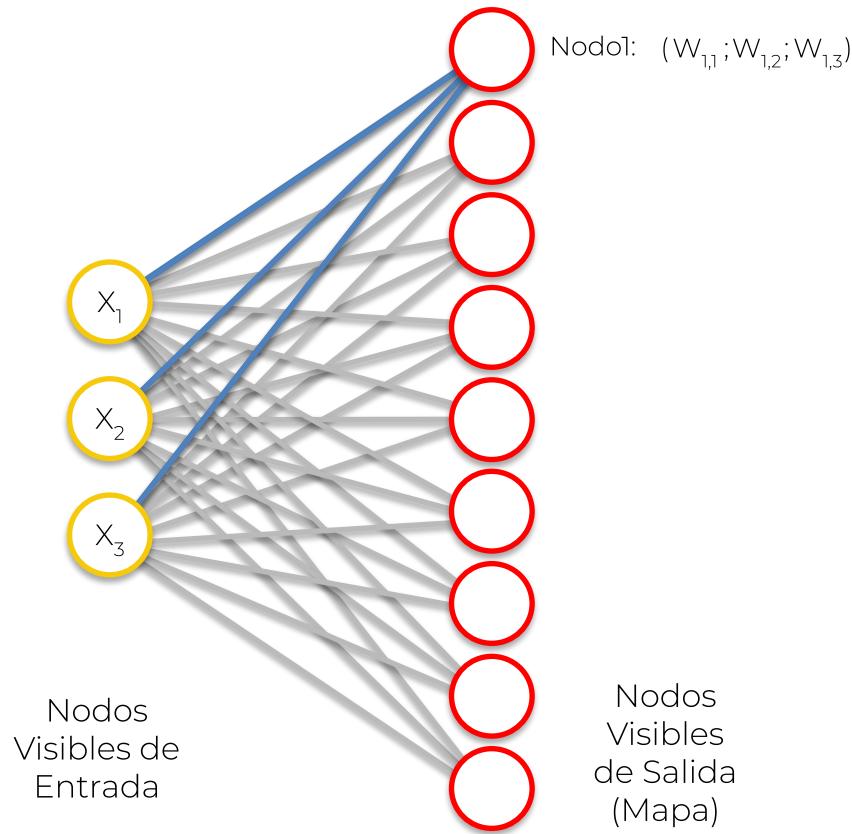
# ¿Cómo aprenden los SOM?



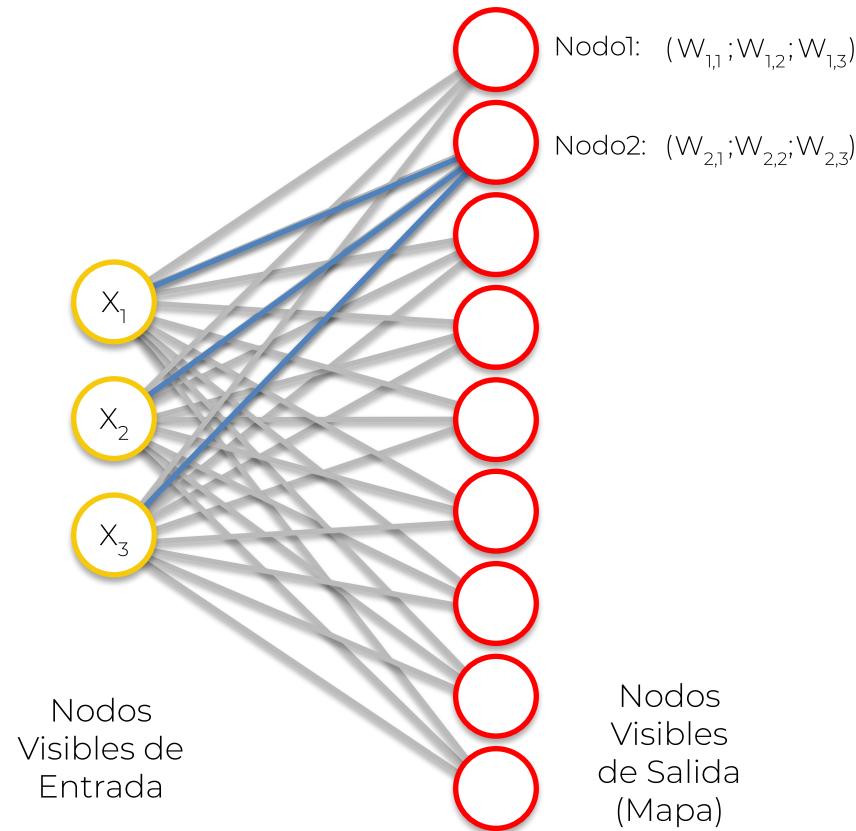
# ¿Cómo aprenden los SOM?



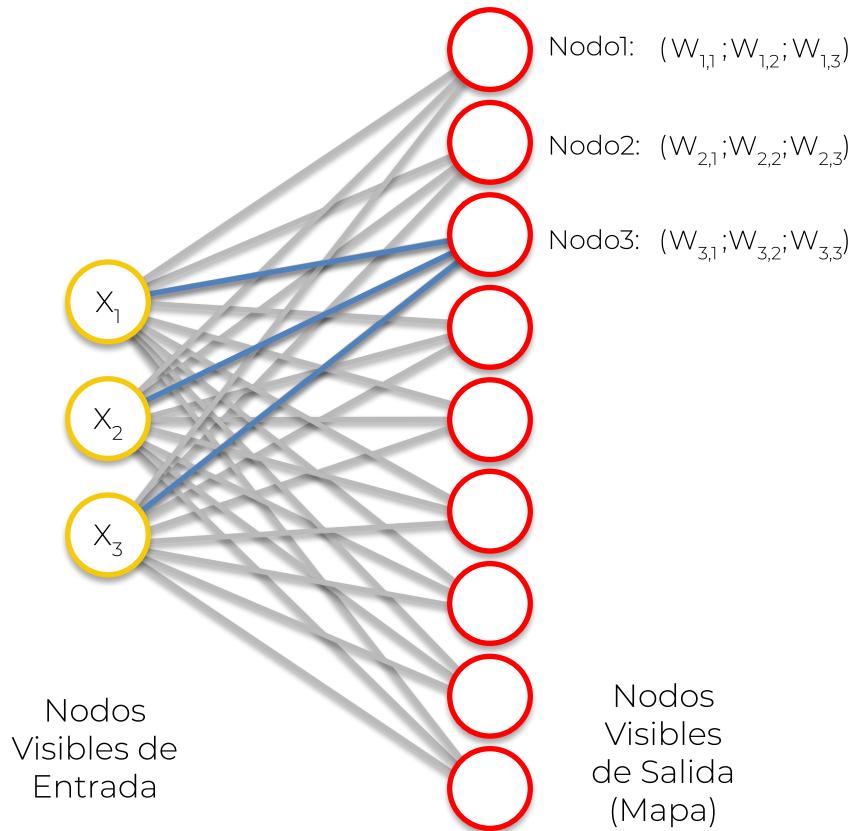
# ¿Cómo aprenden los SOM?



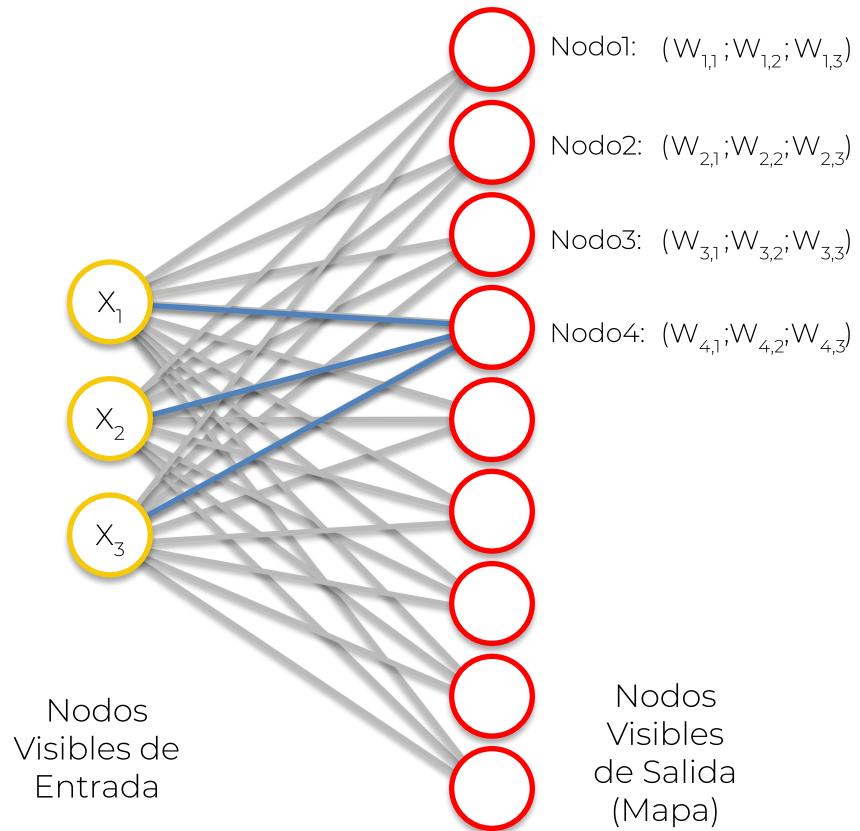
# ¿Cómo aprenden los SOM?



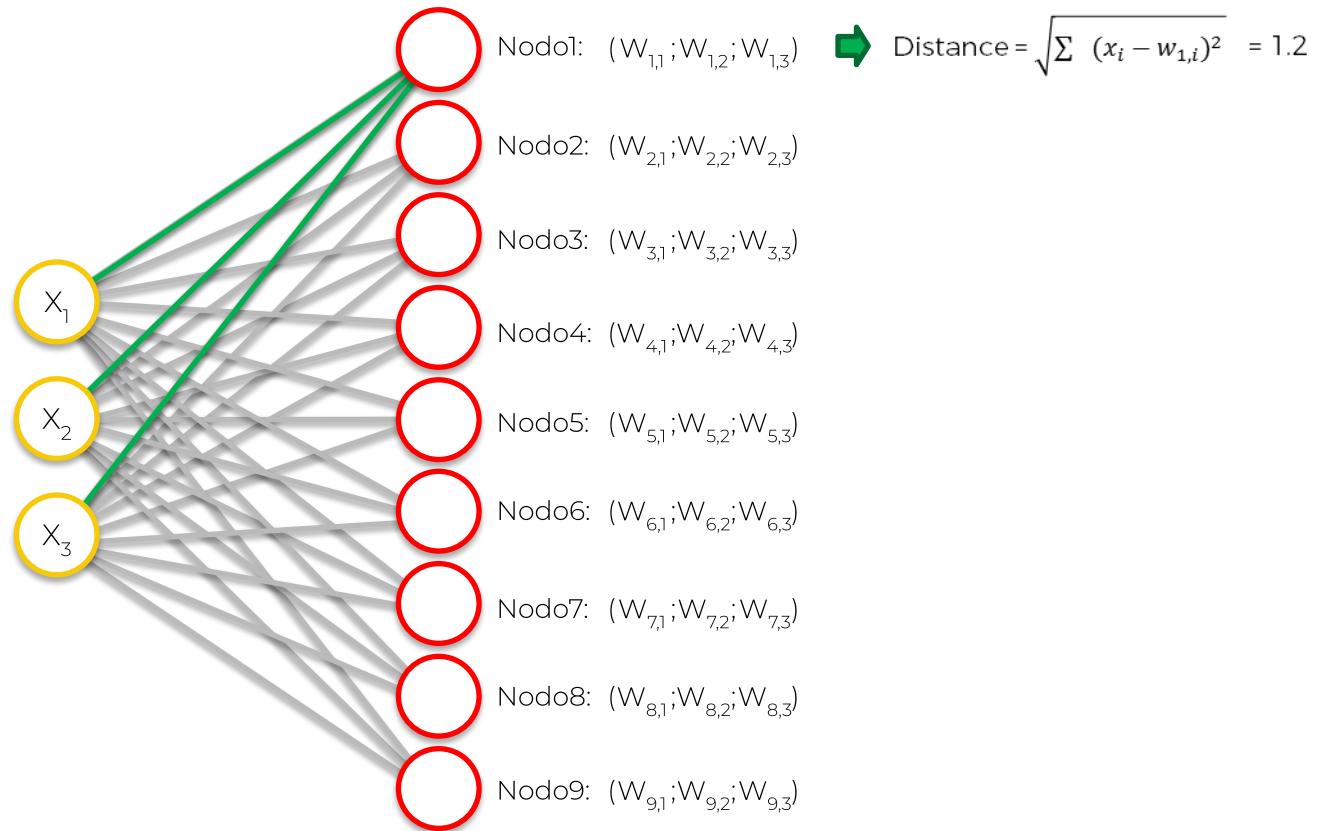
# ¿Cómo aprenden los SOM?



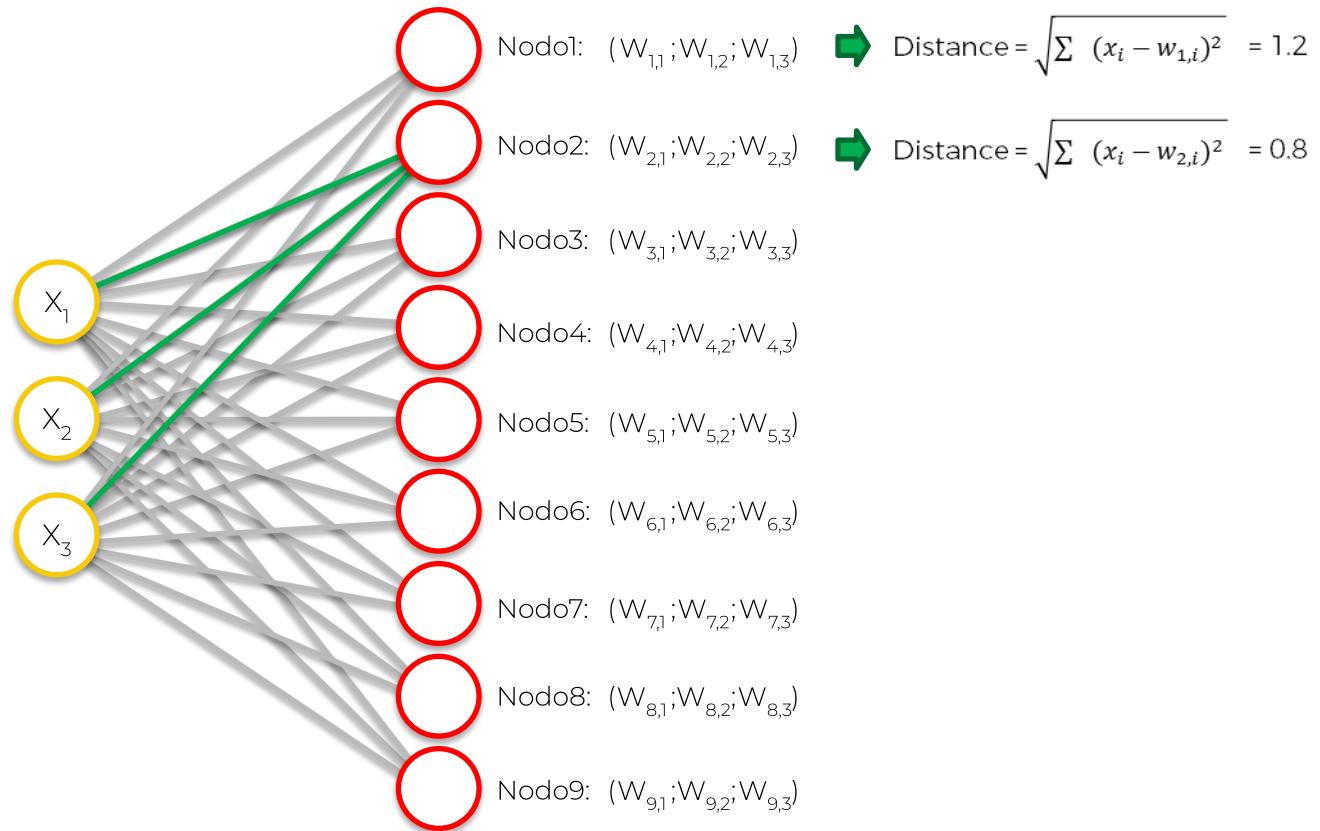
# ¿Cómo aprenden los SOM?



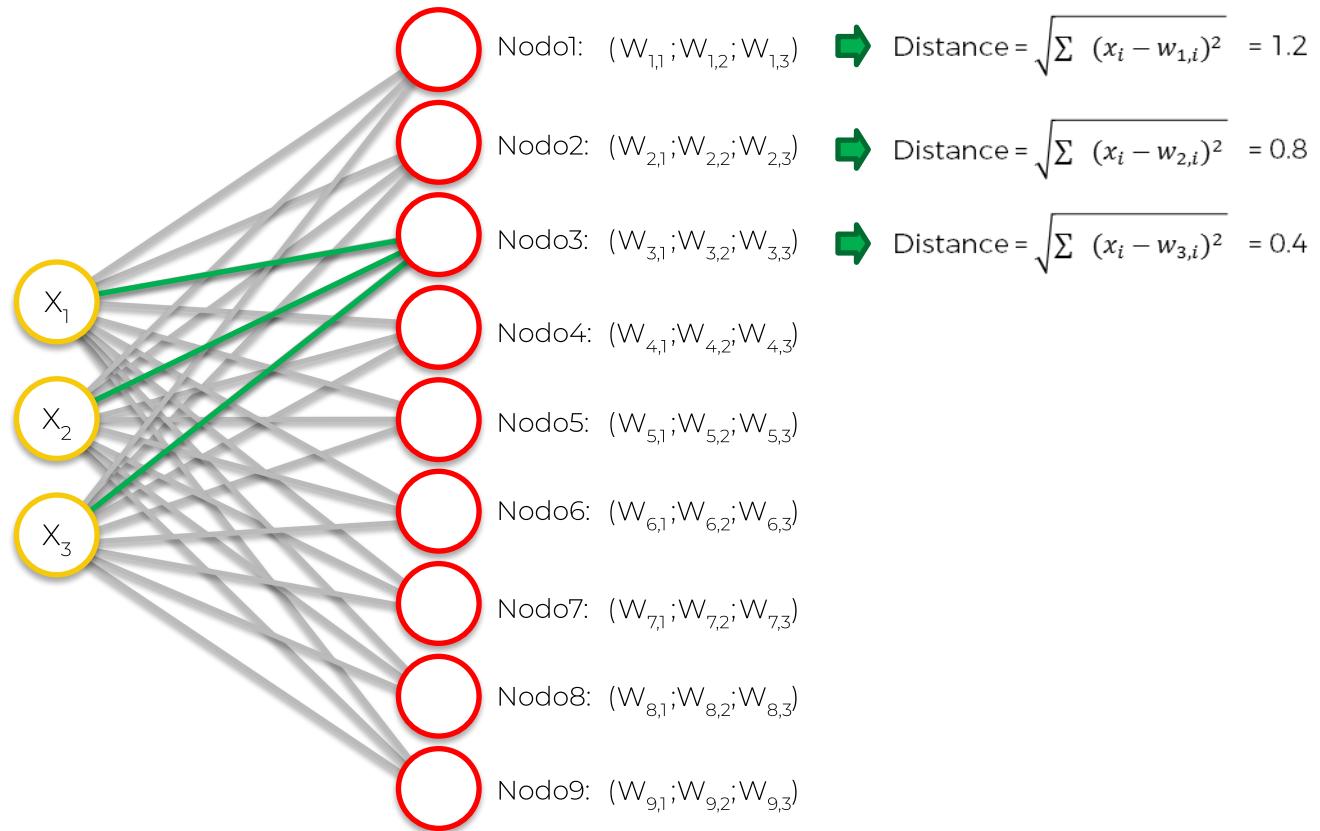
# ¿Cómo aprenden los SOM?



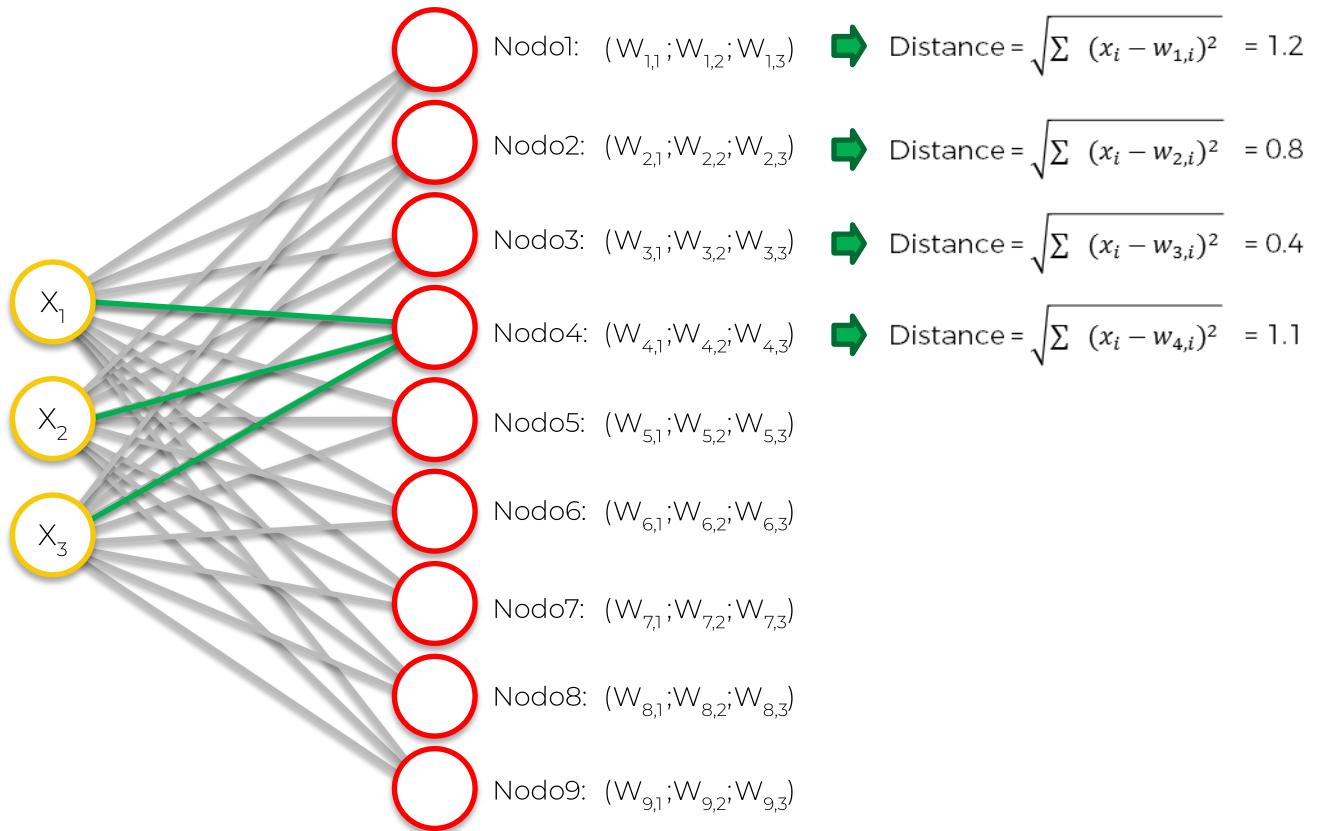
# ¿Cómo aprenden los SOM?



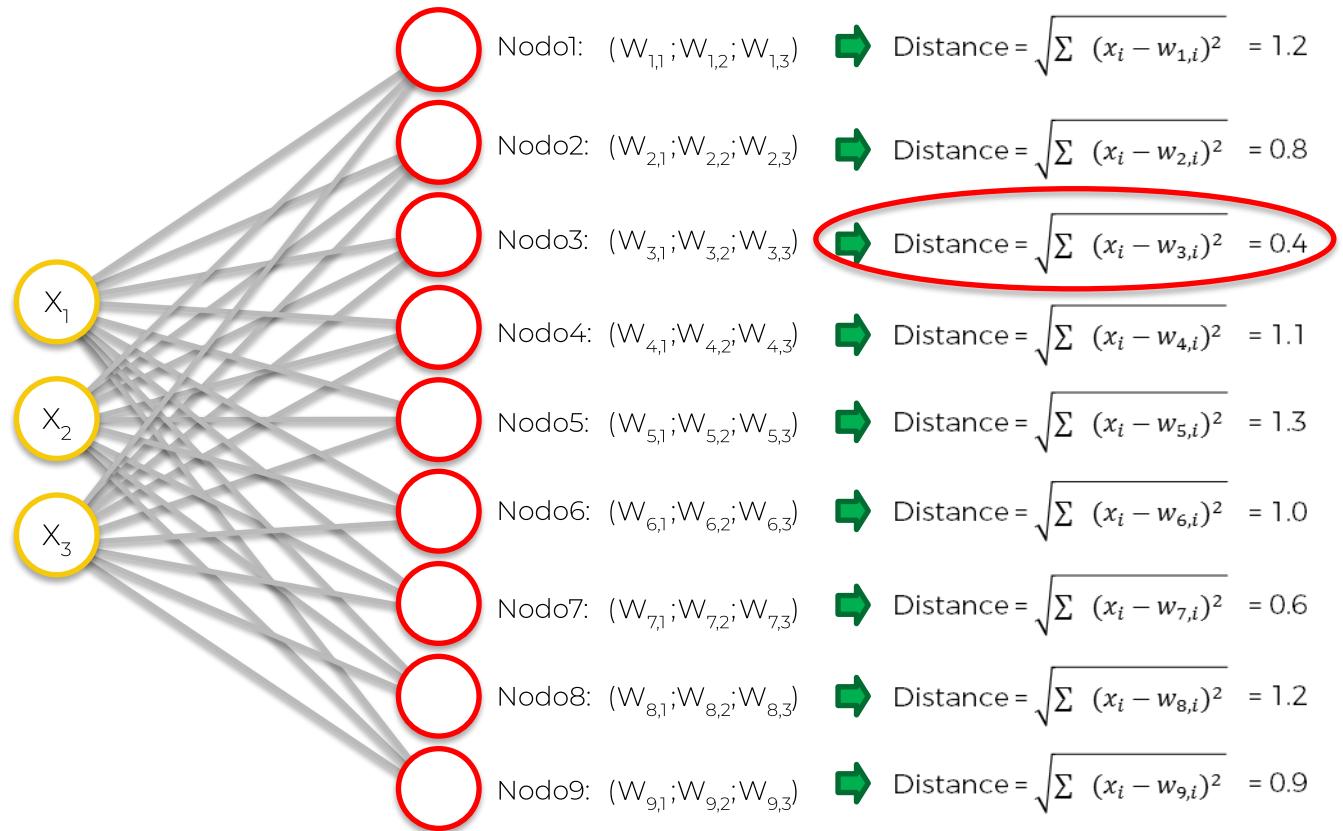
# ¿Cómo aprenden los SOM?



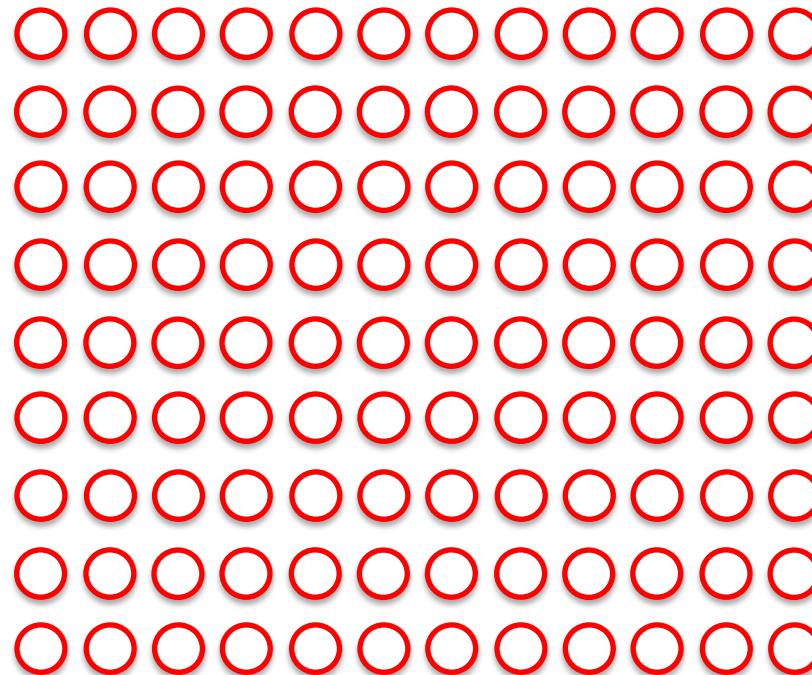
# ¿Cómo aprenden los SOM?



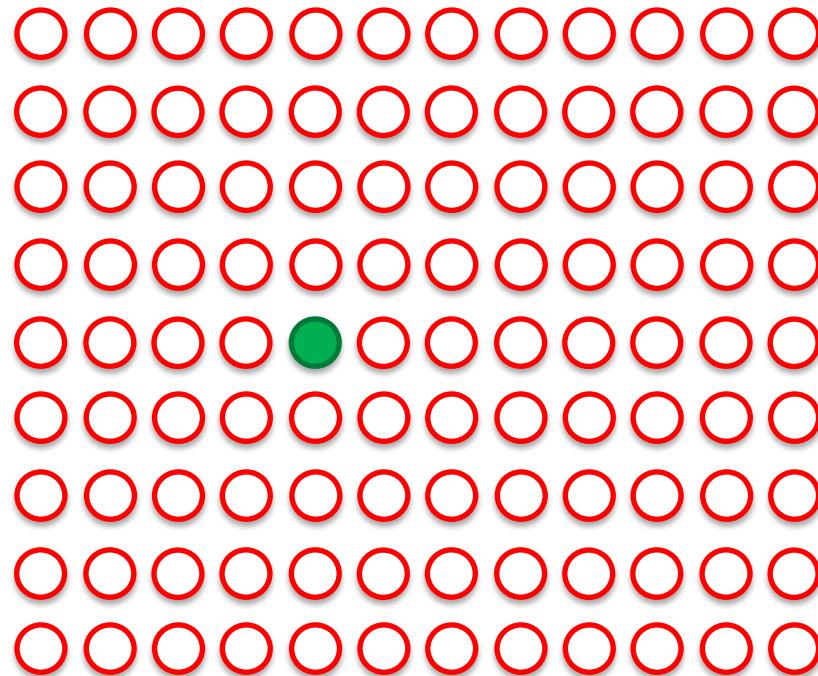
# ¿Cómo aprenden los SOM?



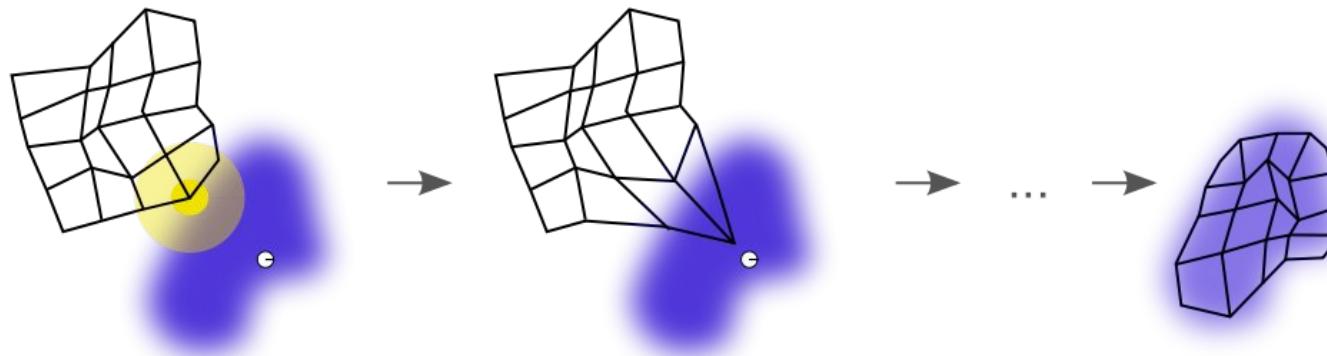
# ¿Cómo aprenden los SOM?



# ¿Cómo aprenden los SOM?

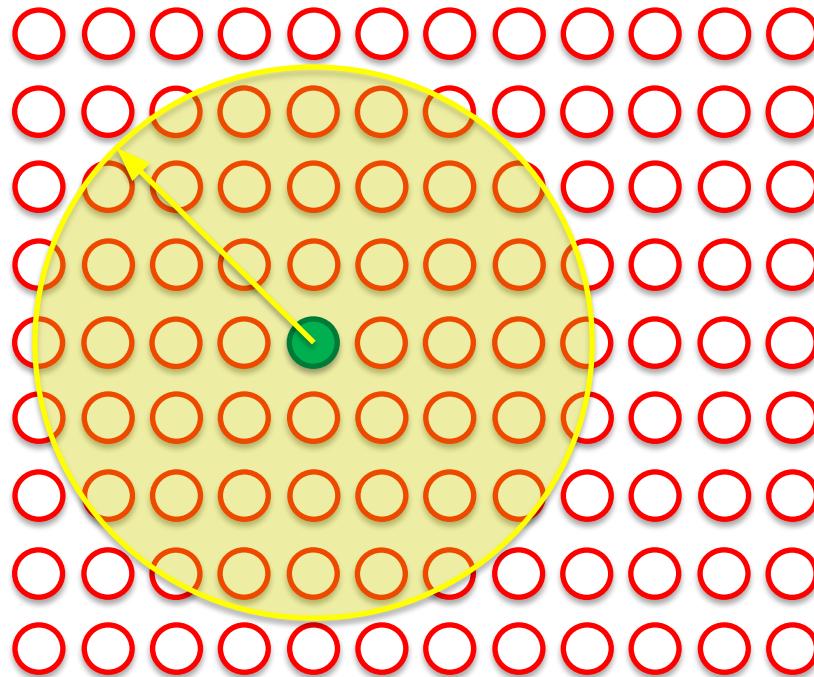


# ¿Cómo aprenden los SOM?

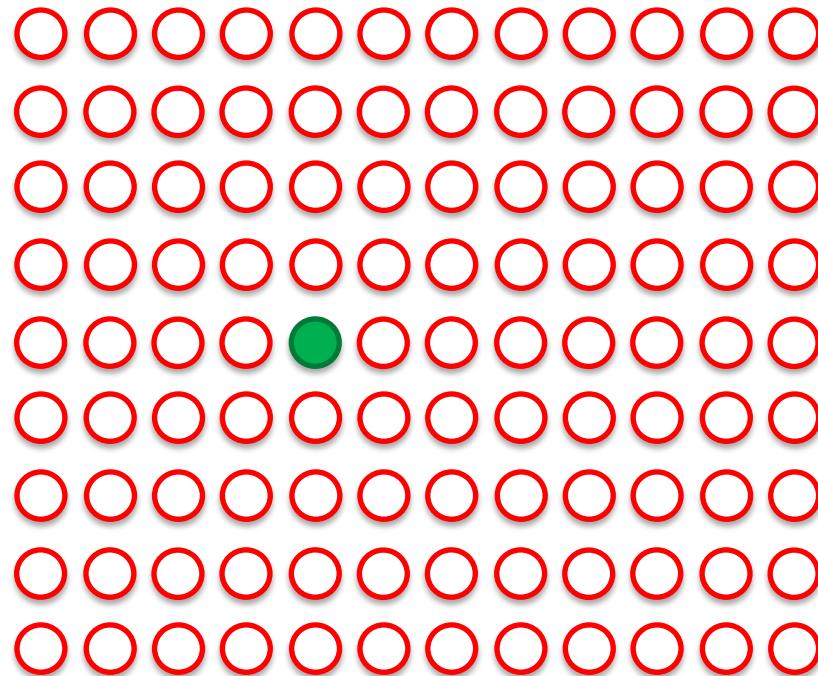


*Image Source: Wikipedia*

# ¿Cómo aprenden los SOM?



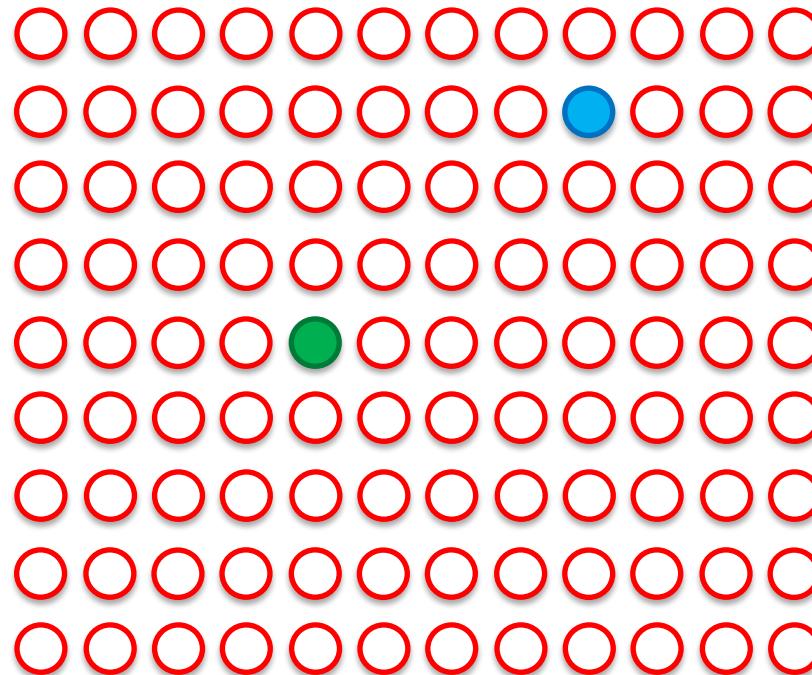
# ¿Cómo aprenden los SOM?



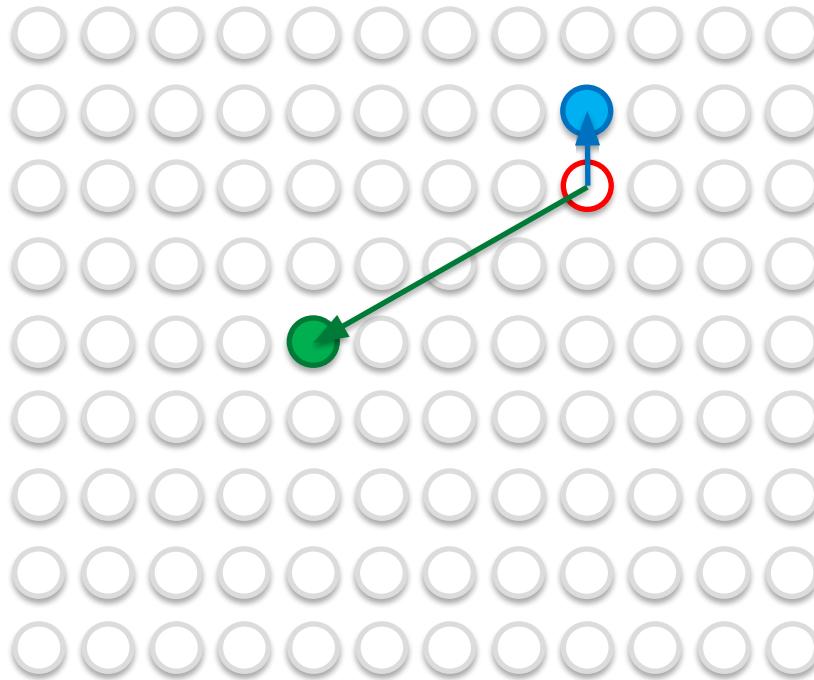
# ¿Cómo aprenden los SOM?



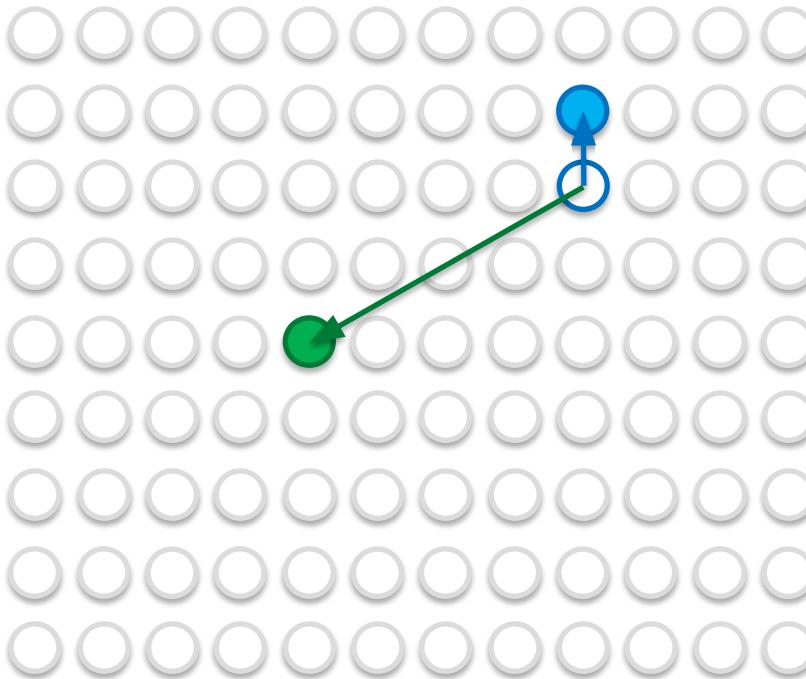
# ¿Cómo aprenden los SOM?



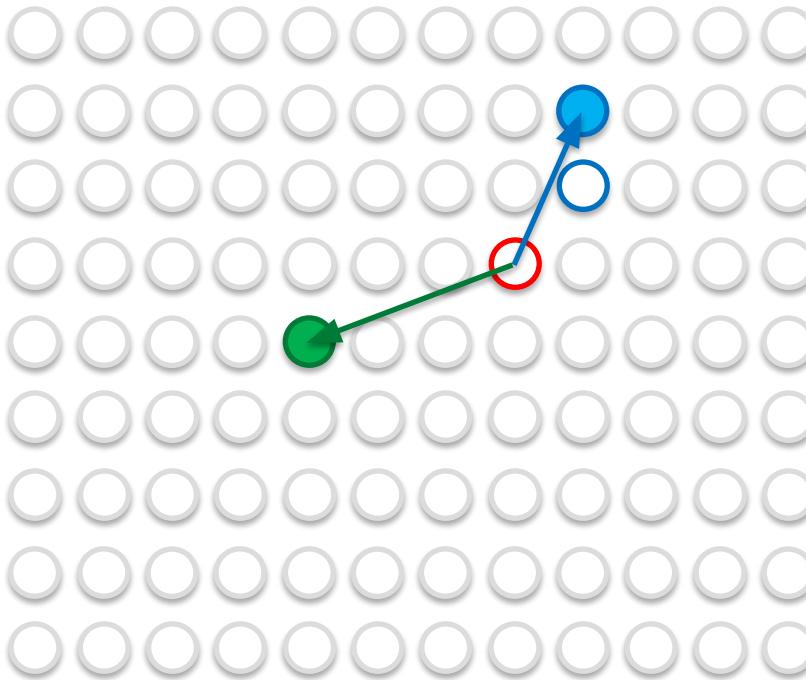
# ¿Cómo aprenden los SOM?



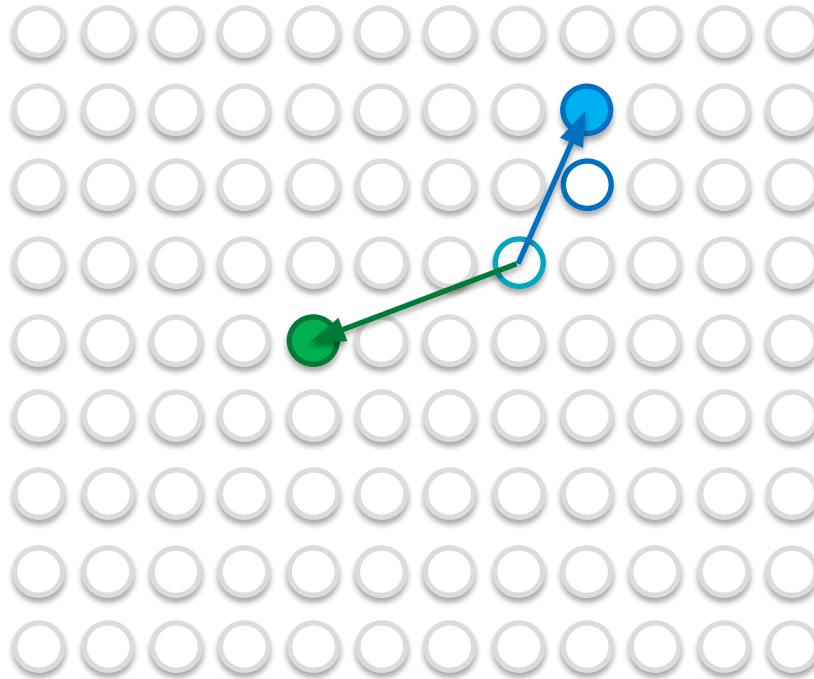
# ¿Cómo aprenden los SOM?



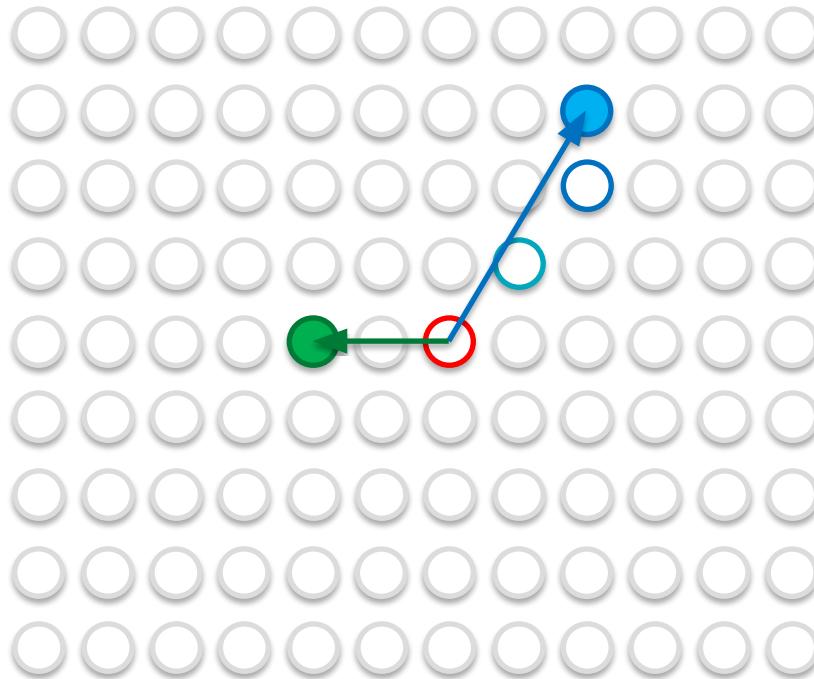
# ¿Cómo aprenden los SOM?



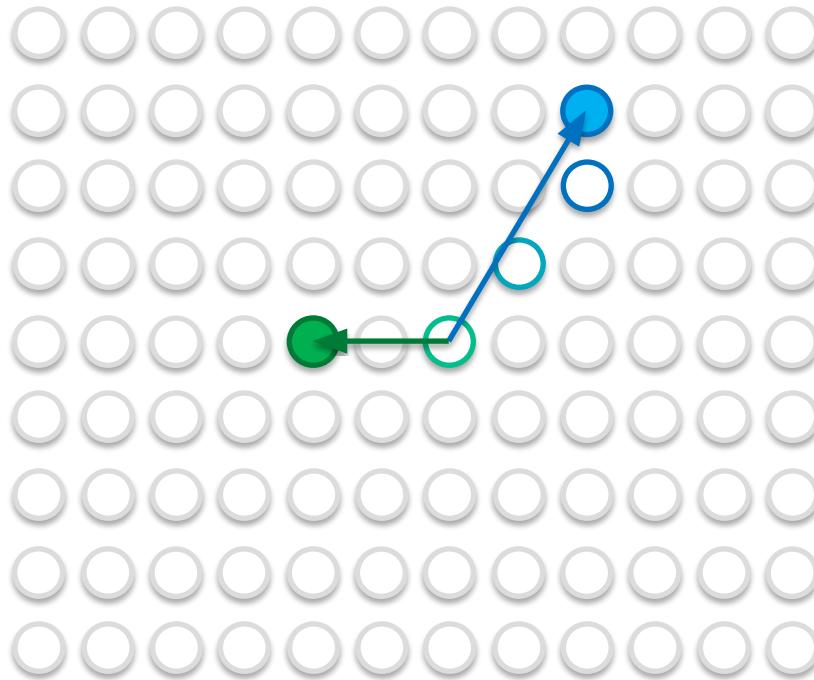
# ¿Cómo aprenden los SOM?



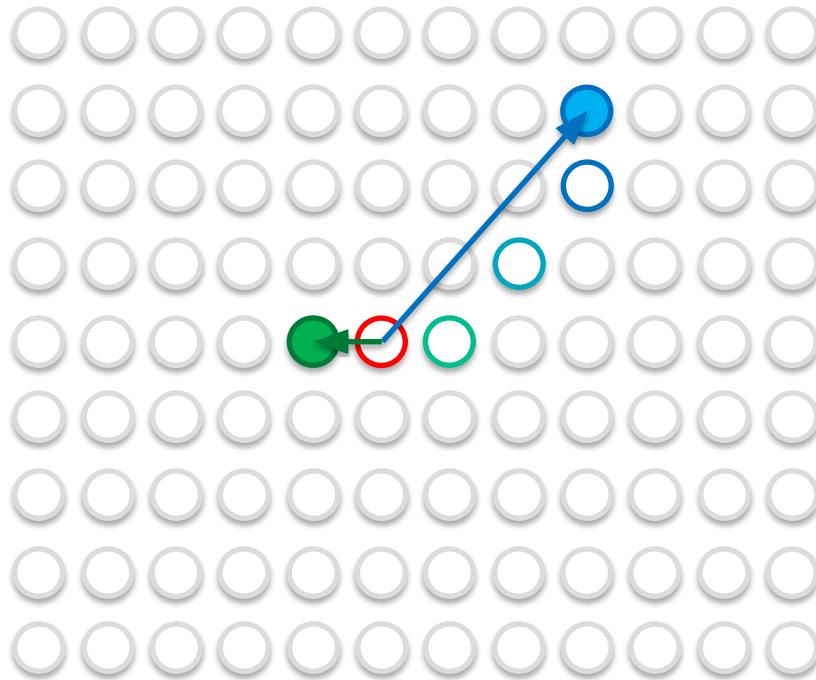
# ¿Cómo aprenden los SOM?



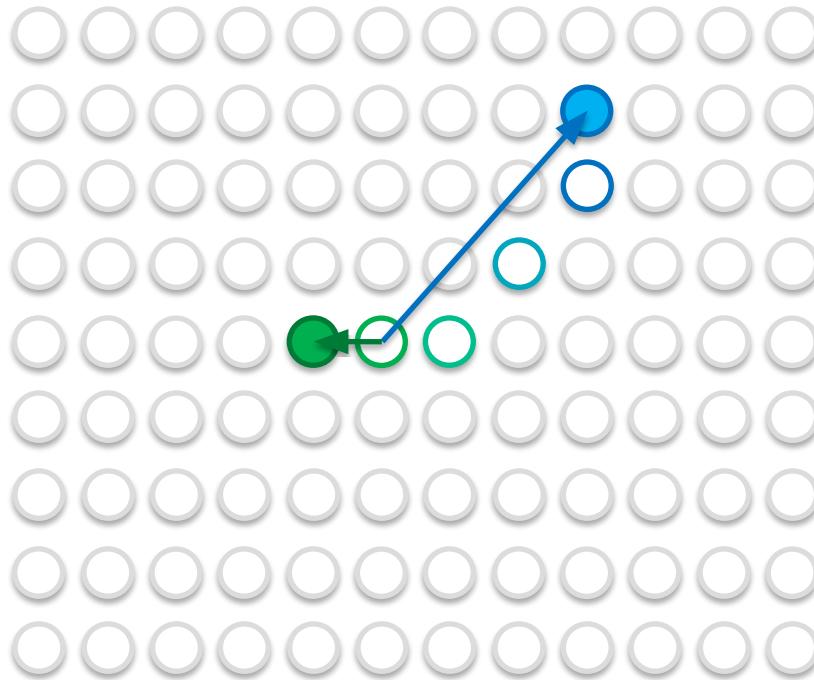
# ¿Cómo aprenden los SOM?



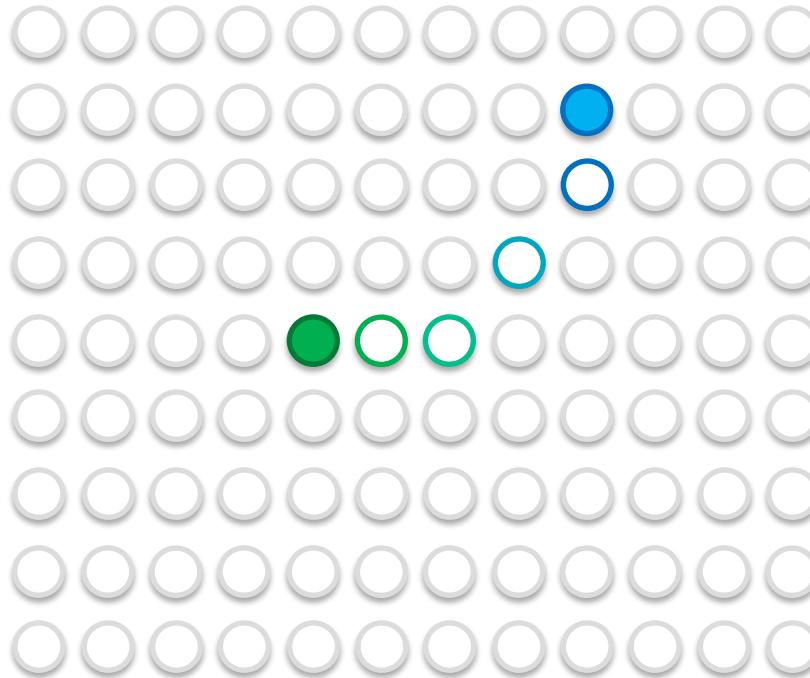
# ¿Cómo aprenden los SOM?



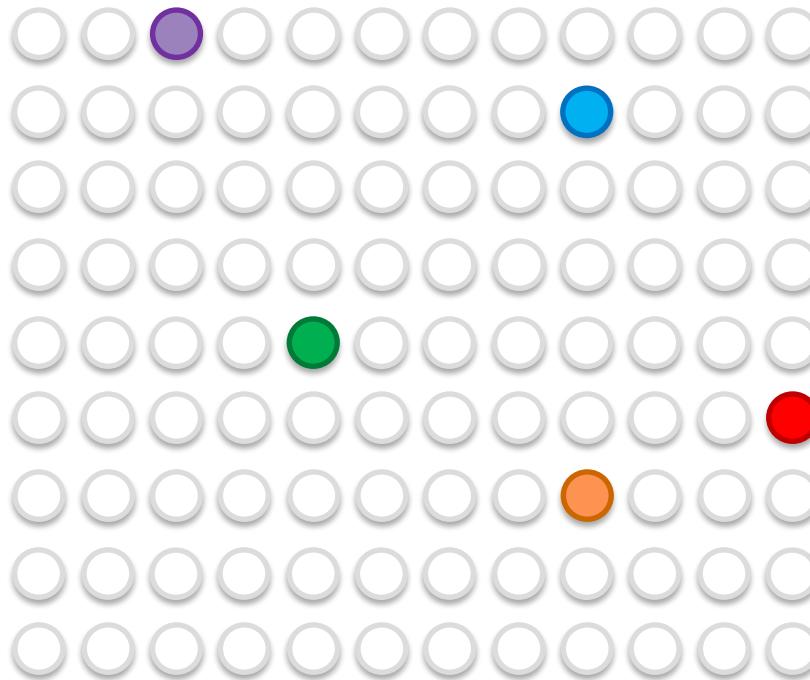
# ¿Cómo aprenden los SOM?



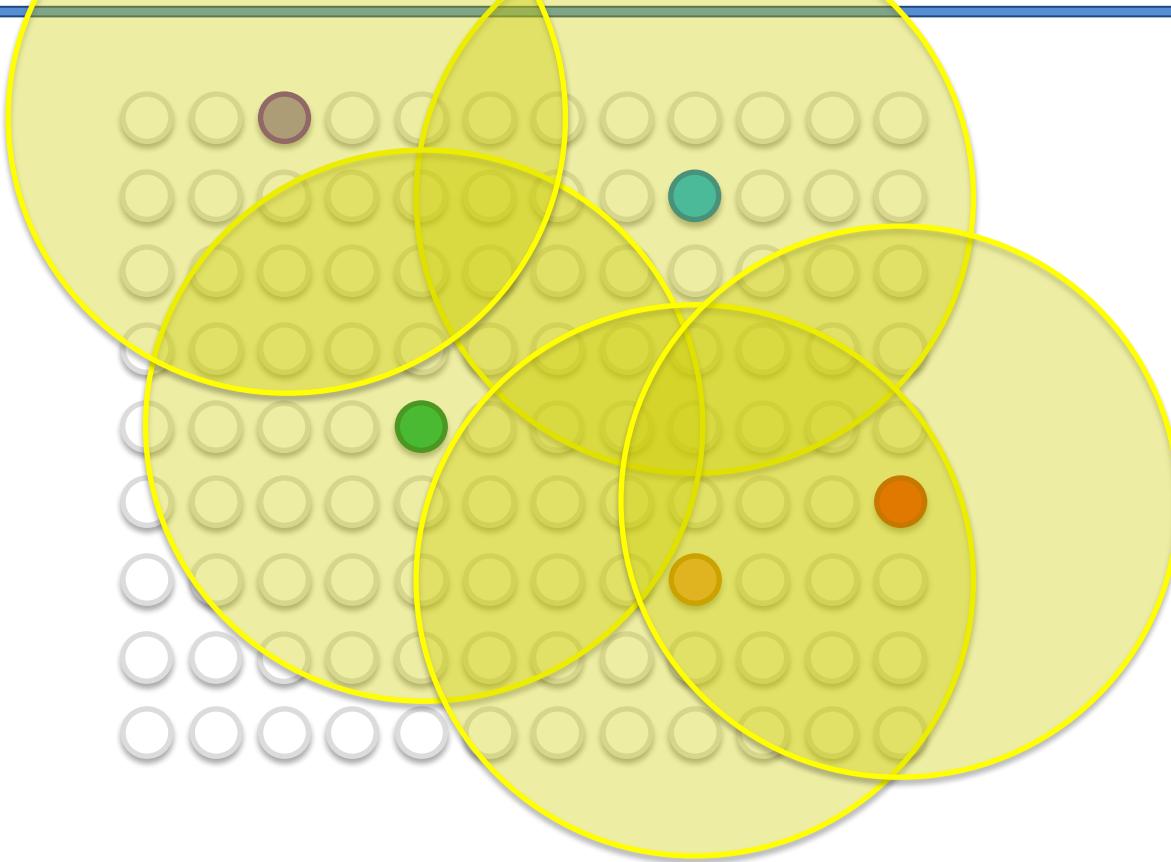
# ¿Cómo aprenden los SOM?



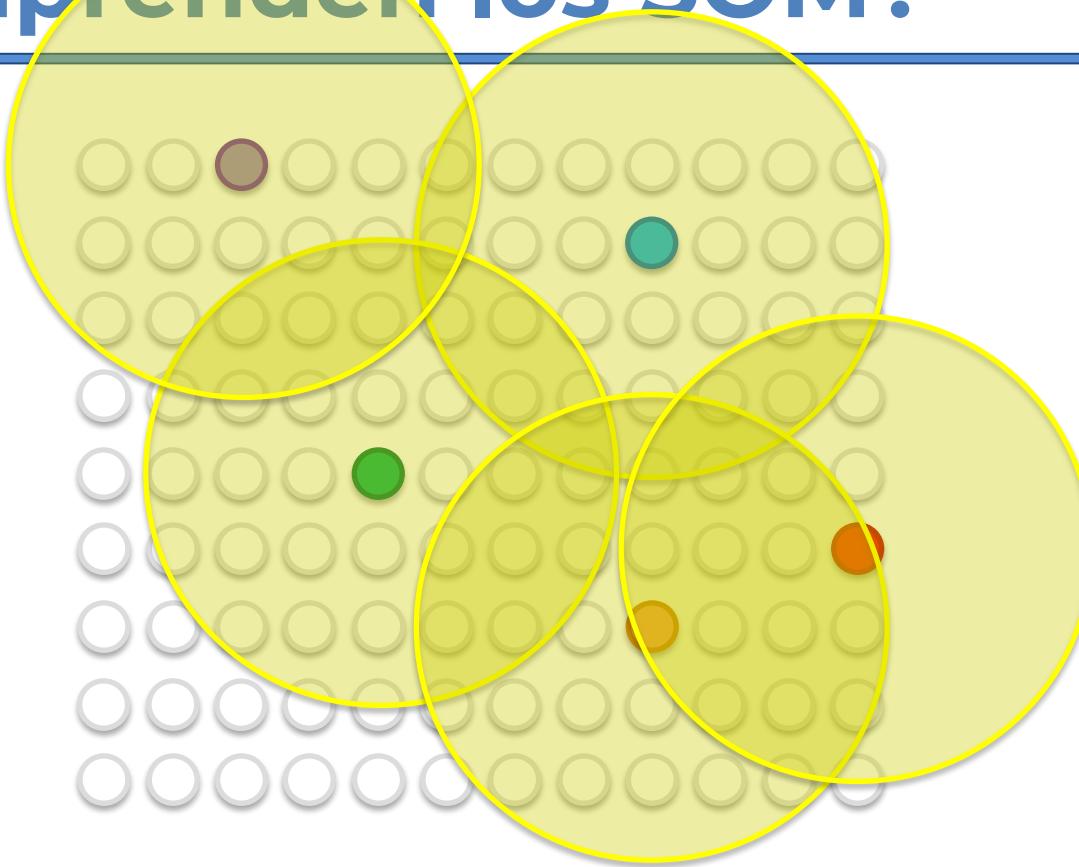
# ¿Cómo aprenden los SOM?



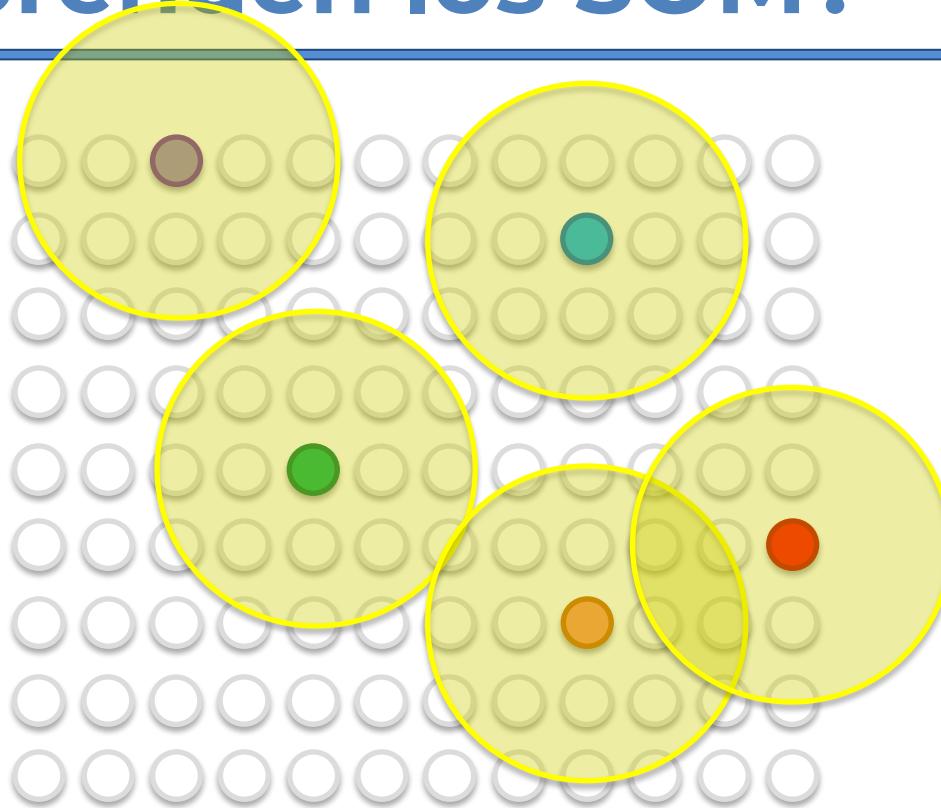
# ¿Cómo aprenden los SOM?



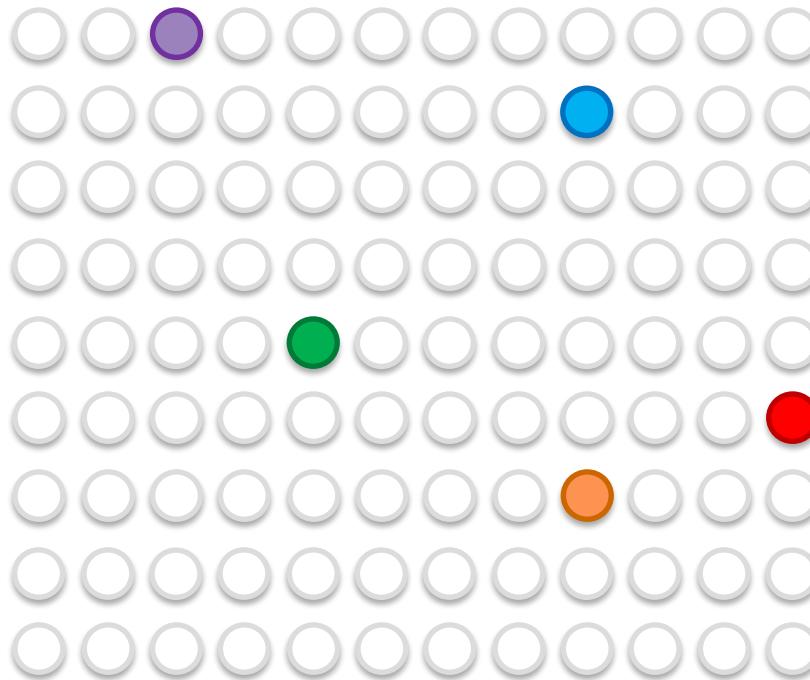
# ¿Cómo aprenden los SOM?



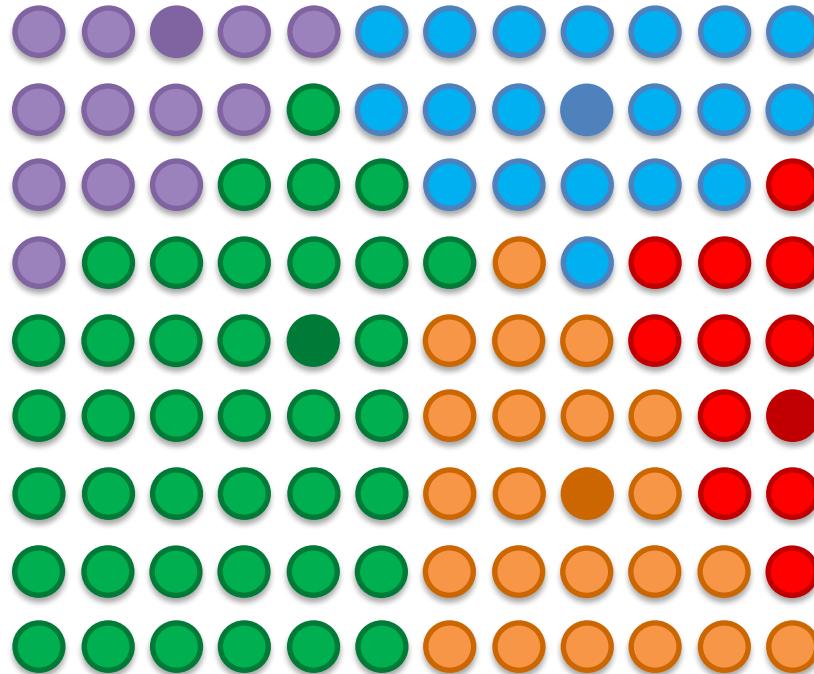
# ¿Cómo aprenden los SOM?



# ¿Cómo aprenden los SOM?



# ¿Cómo aprenden los SOM?



# ¿Cómo aprenden los SOM?

Importante recordar que:

- Los SOMs retienen la topología del conjunto de entrada
- Los SOMs revelan correlaciones que no se identifican fácilmente
- Los SOMs clasifican datos sin supervisión alguna
- No hay vector de variable target -> no hay propagación hacia atrás
- No hay conexiones laterales entre los nodos

# ¿Cómo aprenden los SOM?

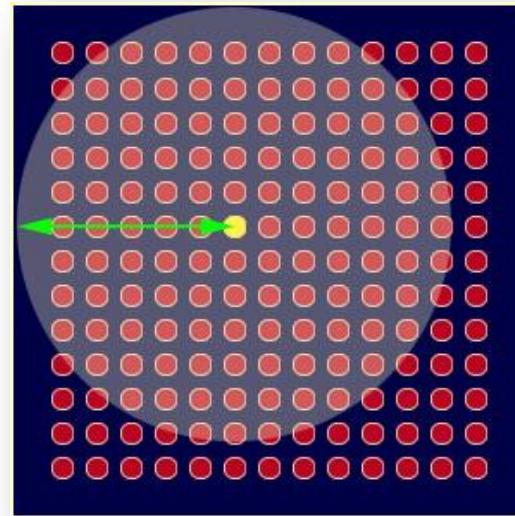
Lecturas Adicionales:

*Kohonen's Self Organizing Feature Maps*

de Mat Buckland (2004?)

Link:

<http://www.ai-junkie.com/ann/som/som1.html>



# Ejemplo real de SOM

# Cómo leer un SOM avanzado

# Cómo leer un SOM Avanzado

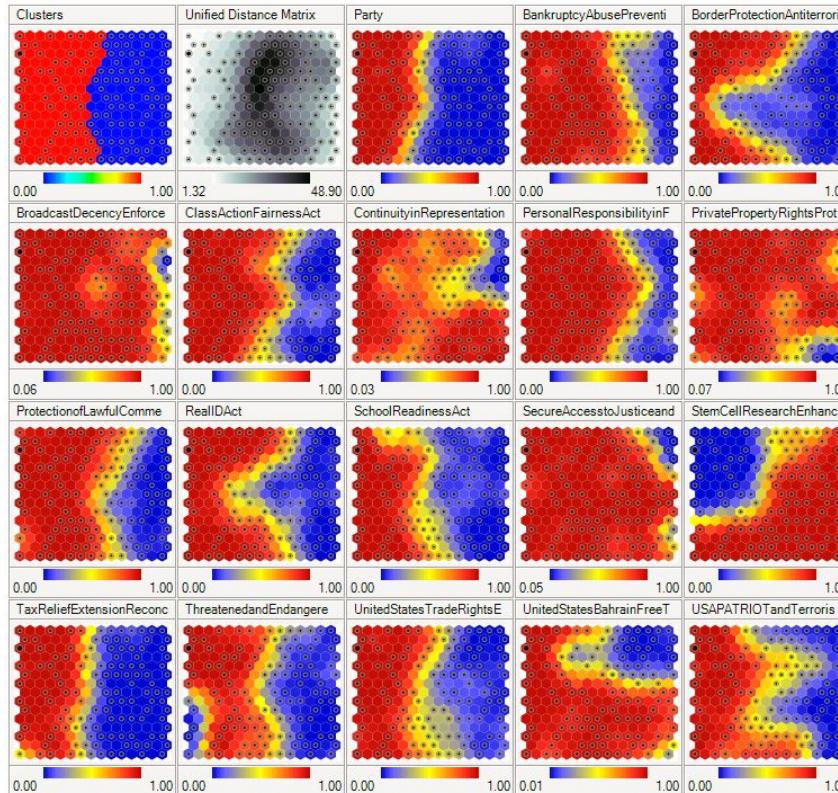


Image Source: Wikipedia

# Cómo leer un SOM Avanzado

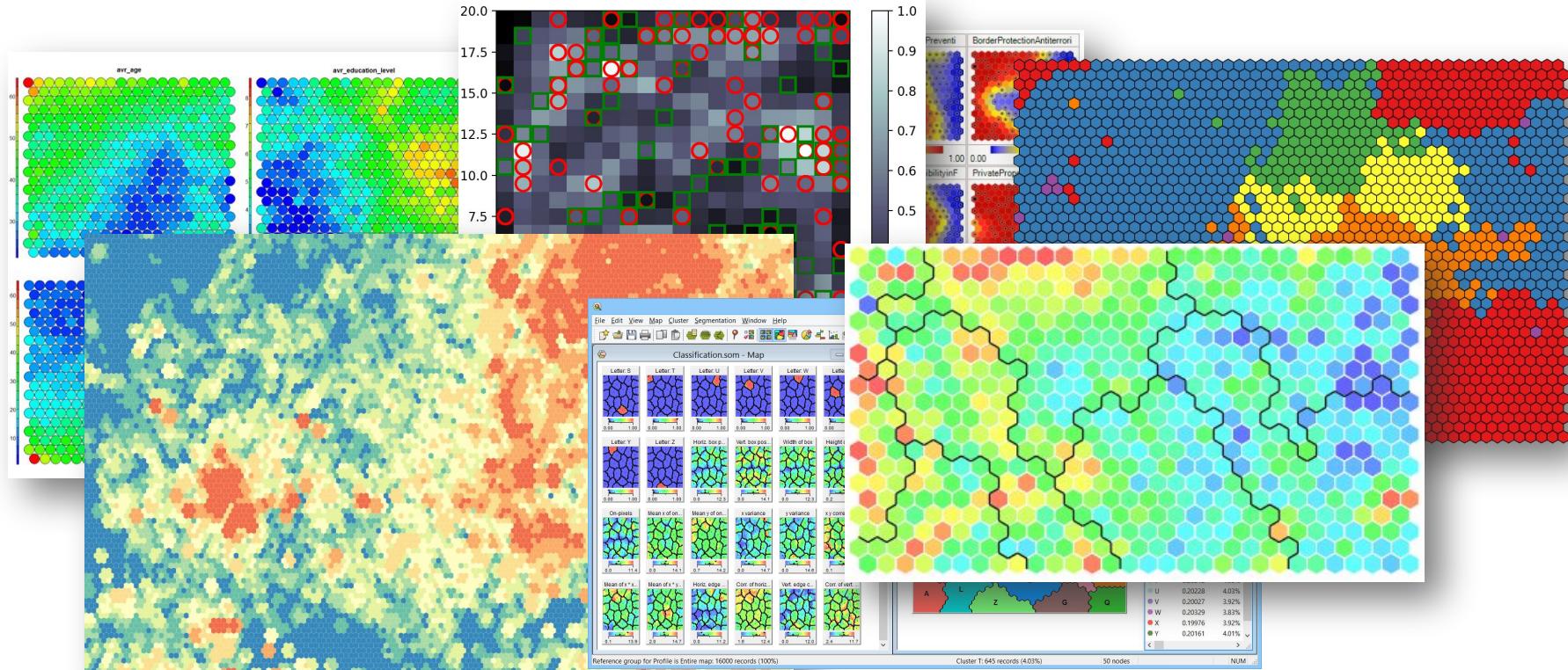


Image Sources in order of appearance: Wikipedia, boylelab.org, R-Bloggers, This Course, stackoverflow.com, Viscovery.com, visualcinnamon.com

# Cómo leer un SOM Avanzado

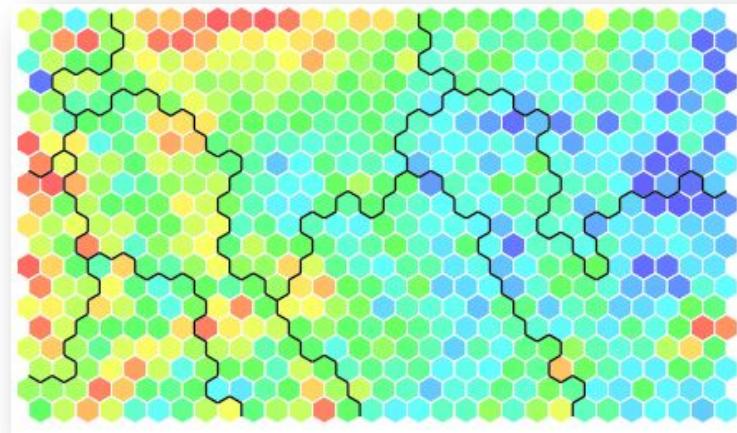
Lecturas Adicionales:

*SOM - Creating hexagonal heatmaps with D3.js*

de Nadieh Bremer (2003)

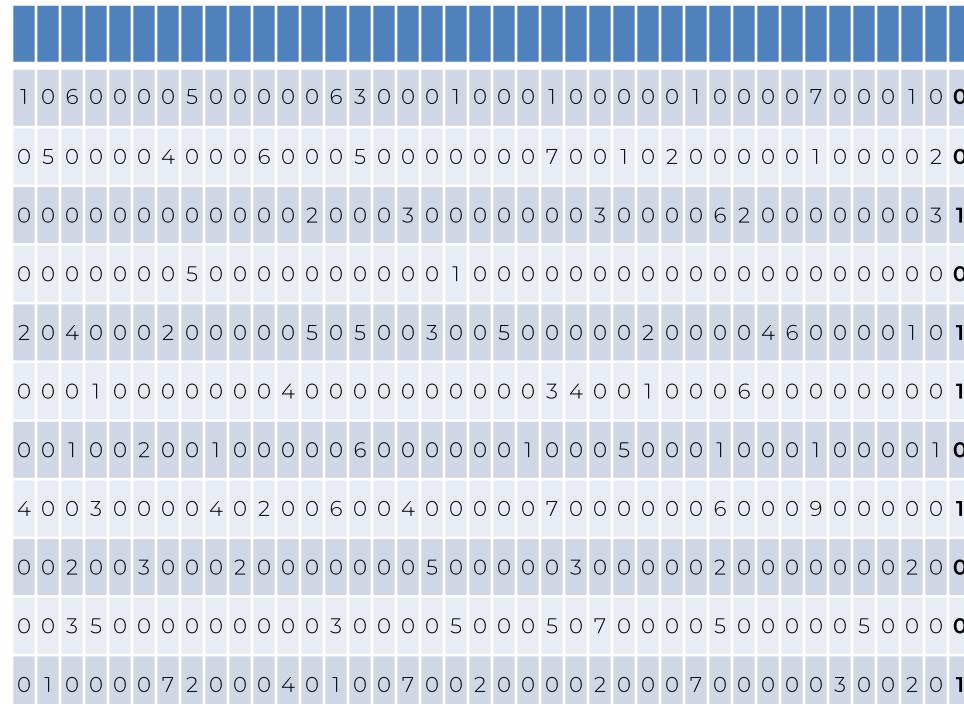
Link:

<https://www.visualcinnamon.com/2013/07/self-organizing-maps-creating-hexagonal.html>

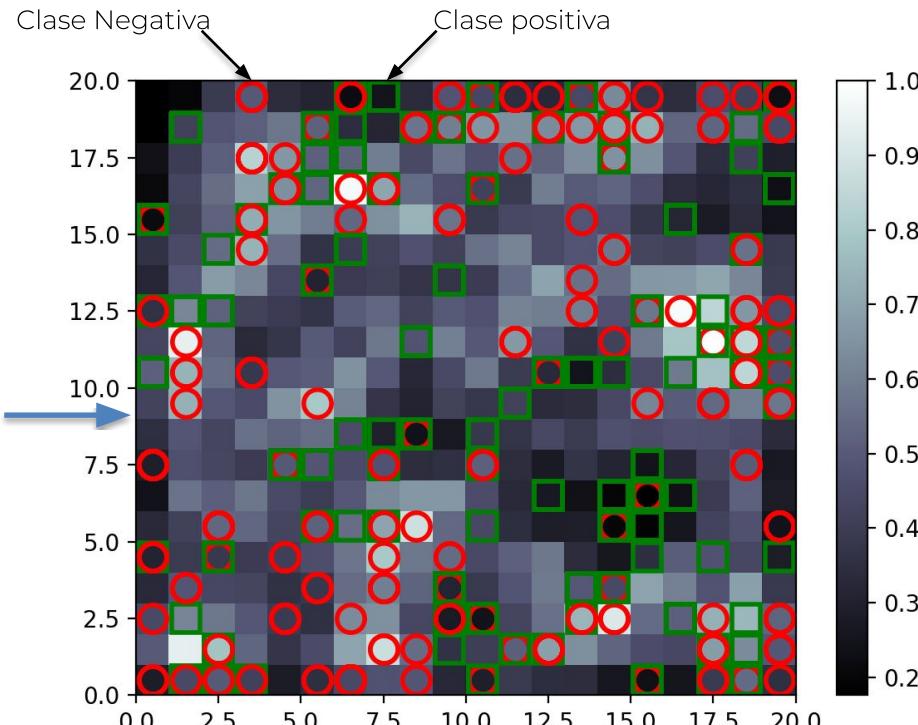


# **EXTRA STUFF**

# Mapas Autoorganizados



Dataset de grandes dimensiones



Mapa Autoorganizado

# Training a SOM

# Entrenar con SOM

**PASO 1:** Empezamos con un dataset compuesto de  $n\_features$  variables independientes.

**PASO 2:** Preparamos una parrilla compuesta de nodos, cada uno con un vector de pesos de  $n\_features$  elementos.

**PASO 3:** Aleatoriamente inicializamos los valores del vector de pesos a números pequeños cercanos a 0 (pero no 0).

**PASO 4:** Seleccionar una observación aleatoria del dataset.

**PASO 5:** Calcular la distancia Euclídea desde dicho punto a las diferentes neuronas de la red.

**PASO 6:** Seleccionar la neurona con la menor distancia al punto. Dicha neurona es el nodo ganador.

**PASO 7:** Actualizar los pesos del nodo ganador para moverlo más cerca del punto.

**PASO 8:** Utilizar una función Gaussiana al vecindario del punto de media el nodo ganador y actualizar los pesos de los vecinos para moverlos más cerca del punto. El radio de los vecinos afectados es la desviación típica de la Gaussiana.

**PASO 9:** Repetir los pasos 1 a 5 y actualizar los pesos después de cada observación (*Reinforcement Learning*) o después de un conjunto de observaciones (*Batch Learning*), hasta que la red neuronal converja en un punto donde los vecindarios no cambien.

# SOM en dimensiones pequeñas

Fijémonos que en dos o tres dimensiones (cuando  $n\_features = 2$  o  $3$ ), el SOM es muy similar a K-Means

# SOM in low dimensions

[https://www.youtube.com/watch?v=abF\\_FdCb5OI](https://www.youtube.com/watch?v=abF_FdCb5OI)

## Self-Organizing Maps : Origins

- Ideas first introduced by C. von der Malsburg (1973), developed and refined by T. Kohonen (Finland ) (1982)
- Neural network algorithm using unsupervised competitive learning
- Primarily used for organization and visualization of complex data
- Neurons are arranged on a flat grid (map, lattice, 2 dimensional array).
- There is no hidden layer.** Only an input and output layer.
- Each neuron on the grid is an output neuron
- Topological relationships within the training set are maintained



Teuvo Kohonen

## Self-Organizing Maps : Introduction

The neurons in the output layer are arranged on a map

