



2.4. Redes Neuronales Artificiales. Mapas Autoorganizativos de Kohonen (SOM)

M^a Carmen Pegalajar

mcarmen@decsai.ugr.es

*Depto de Ciencias de la Computación e IA
Universidad de Granada*

Mapas Autoorganizativos de Kohonen (SOM)

- ▣ Los modelos de Mapas Autoorganizativos (SOM) fueron introducidos por T. Kohonen [Kohonen90,Kohonen01]
- ▣ Estas redes tienen la habilidad de encontrar similitudes entre los datos
- ▣ Han demostrado ser una herramienta muy valiosa en Minería de Datos: reconocimiento de patrones, análisis de imágenes, monitoreo de procesos, etc

Mapas Autoorganizativos de Kohonen (SOM)

Ventajas:

- ❑ No hacen suposiciones sobre la distribución de variables ni requieren independencia entre variables.
- ❑ Son más fáciles de implementar y pueden resolver problemas no lineales de alta complejidad.
- ❑ Manejan eficazmente datos ruidosos y faltantes, dimensiones muy pequeñas y muestras de tamaño ilimitado

Mapas Autoorganizativos de Kohonen (SOM)

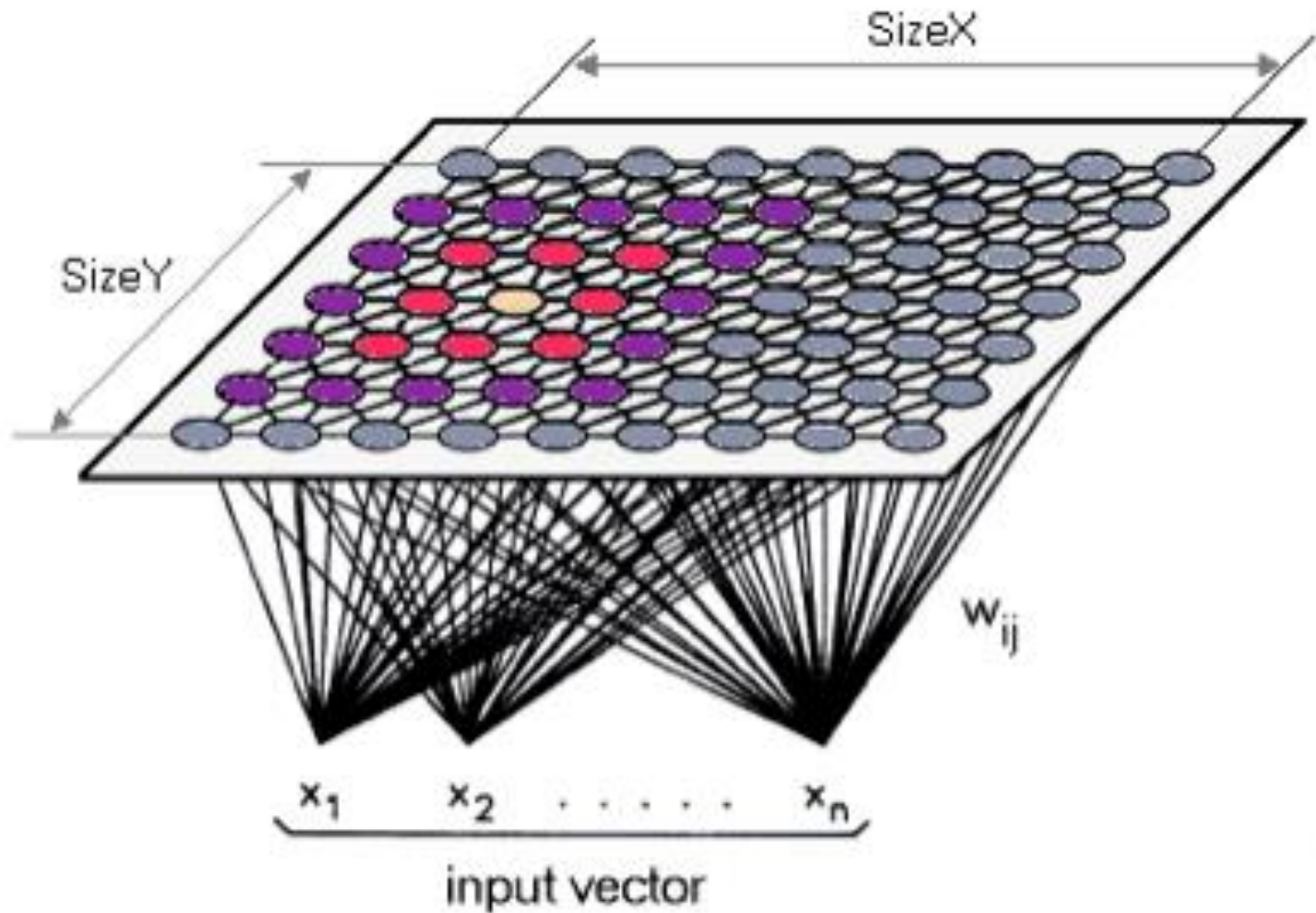
Procesos principales del SOM:

- ❑ **Competencia:** las neuronas compiten entre si para representar de la mejor manera a una entrada
- ❑ **Cooperación:** neuronas cercanas representarán a entradas similares
- ❑ **Adaptación:** tanto la neurona ganadora como sus vecinas adaptan sus pesos a favor de los valores mas altos de la función de activación.

Mapas Autoorganizativos de Kohonen (SOM)

- Son **algoritmos no-supervisados**, es decir no usan la etiqueta de clase de los datos para el entrenamiento.
 - Dicha etiqueta se utiliza para **visualizar los datos correctamente**
- Se denominan también **algoritmos competitivos** porque en el entrenamiento se activa/entrena una sola neurona cada vez

Mapas Autoorganizativos de Kohonen (SOM)



Mapas Autoorganizativos de Kohonen (SOM)

- ❑ Cuando una nueva señal de entrada llega **todas las neuronas compiten** para representarla
- ❑ La unidad que mejor se ajusta es la unidad que **gana la competición** y en conjunto con los vecinos de la rejilla aprenden la señal.
- ❑ Las **neuronas vecinas** gradualmente se especializarán para representar **señales de entrada similares**.
- ❑ Las representaciones se organizarán y ordenarán en la rejilla del mapa

Mapas Autoorganizativos de Kohonen (SOM)

1. Calcular la unidad que mejor se ajusta al vector de entrada.

$$\|x - w_c\| = \min_i \|x - w_i\|$$

En general se utiliza la distancia euclídea:

$$\|x - w_i\| = \sqrt{\sum_{j=1}^n (x_j - w_i^j)^2}$$

Mapas Autoorganizativos de Kohonen (SOM)

2. Se actualiza la unidad ganadora y los vecinos más cercanos. La magnitud de dicha atracción está regida por la tasa de aprendizaje.

$$W_j(t+1) = W_j(t) + \Theta(k, j, t) \alpha(t) (X(t) - W_j)$$

$\alpha(t)$: razón de aprendizaje

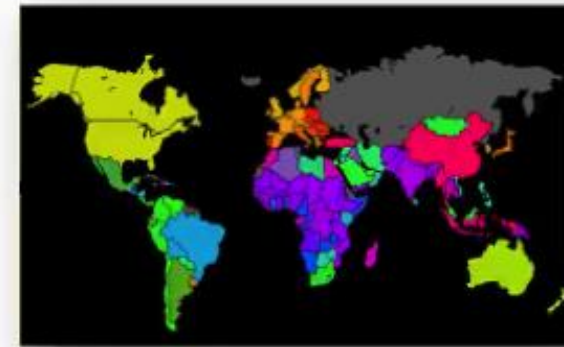
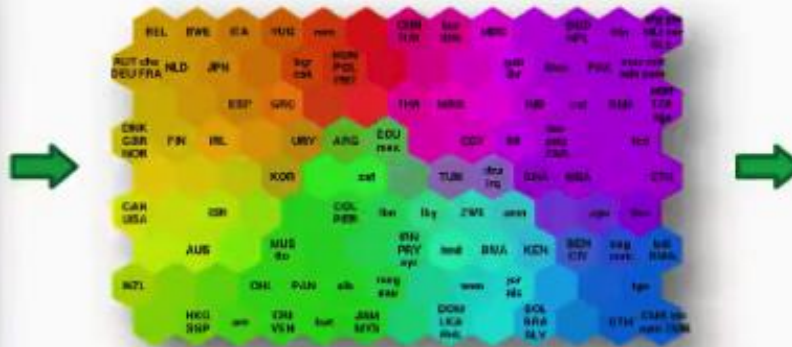
$\Theta(k, j, t)$: vecindad de la neurona k a la j , si está próxima vale 1, sino 0

3. Conforme el entrenamiento se va produciendo, la tasa de aprendizaje y la vecindad van decreciendo

Mapas Autoorganizativos de Kohonen (SOM)

EJEMPLO DE APLICACIÓN: conjunto de datos del Banco Mundial que indica el desarrollo humano

	A	B	C	D	E
1	Country	Country O	Health Ex	Education E	Inflation
2	Aruba	ABW	9.418971	5.92467022	-2.13637
3	Afghanistan	AFG	4.371774		-8.28508
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	Azerbaijan	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





2.5 Redes Neuronales Artificiales. Redes Función Base Radial (RBF)

M^a Carmen Pegalajar

mcarmen@decsai.ugr.es

*Depto de Ciencias de la Computación e IA
Universidad de Granada*

Redes RBF

Están formadas por tres capas:

- **Capa de entrada**

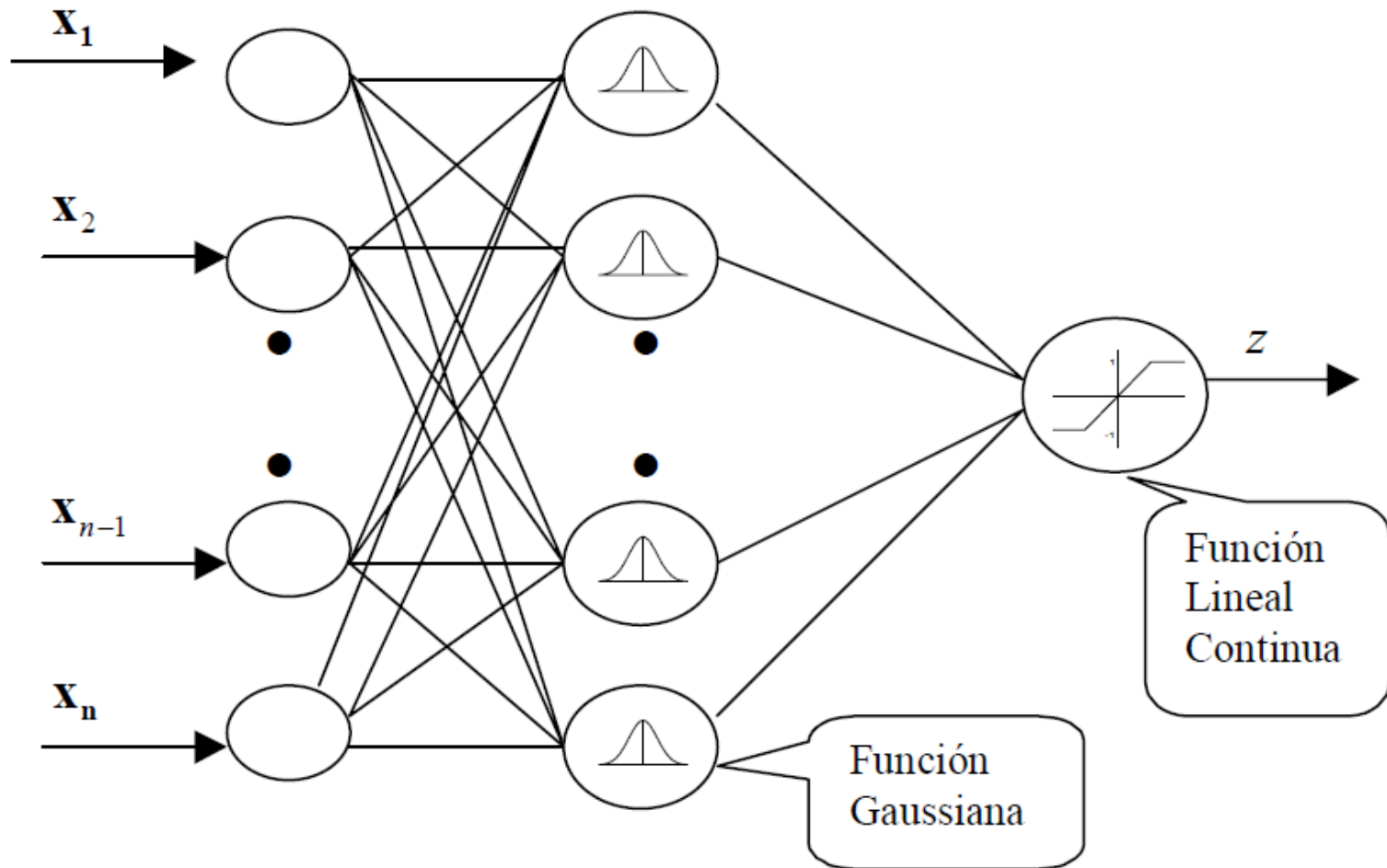
- **Capa oculta:**

Completamente interconectada a la capa de entrada y con funciones de activación radial (gaussiana)

- **Capa de salida:**

Completamente interconectada a la capa oculta y activada a través de una función lineal continua

Topología



Capa Oculta

Los nodos ocultos contienen una función base radial, que tiene como parámetros centro y ancho.

- Existe un **centro** para cada función radial
- El **ancho** es el término empleado para identificar la amplitud de la campana de gauss. Habitualmente se utiliza un ancho que es común en cada una de las funciones gaussianas

Capa Oculta

- En cada nodo de la capa oculta **se calcula la distancia radial** (dist. euclidea) *d* entre el vector de entrada x , y el centro de gravedad de ese nodo.

$$d = |x - c| = \sqrt{(x_1 - c_1)^2 + (x_2 - c_2)^2 \dots + (x_n - c_n)^2}$$

- Este valor d es un componente en la entrada para activar la función radial

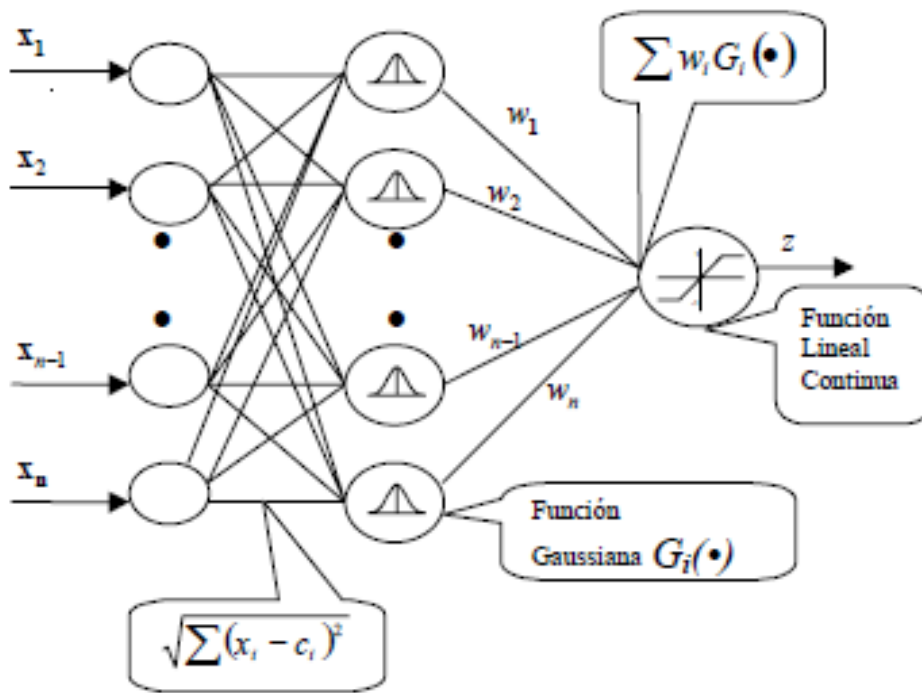
Capa Oculta

La función quedaría así:

$$G_i(.) = \exp(d^2 / a)$$

Donde a es el ancho para ese nodo oculto y d es la distancia calculada

Capa de Salida



En la capa de salida están los pesos w que nos sirven para calcular la salida lineal z

$$z = \sum w_i G(.)$$

Donde $G(.)$ es la salida de la capa oculta

2. Entrenamiento

El resultado de este tipo de entrenamiento:

- Los valores de entrada deben ser transformados a una escala
- En la capa oculta, si los valores de entrada:
 - Se parecen a un centro, su distancia tenderá a cero y de este modo la función gaussiana se disparará a las vecindades de uno
 - No se parecen a un centro la distancia será mayor y la función radial parecería tender a cero.

Este proceso es una clasificación no lineal de las entradas

2. Entrenamiento

- ▣ Los valores obtenidos en las salidas de la capa oculta serían transformados por la función lineal que permite aproximar los valores de Z a los valores deseados.
- ▣ El tiempo de entrenamiento es substancialmente inferior al requerido por otros algoritmos

2. Entrenamiento

El entrenamiento tiene que determinar todos los parámetros de la red:

- Parámetros de la capa de salida: Pesos w
- Parámetros Capa Oculta: Centros c y anchos a

2. Entrenamiento

Tiene un **Aprendizaje Híbrido**:

- **Fase No Supervisada:** Determinación de parámetros de la capa oculta
- **Fase Supervisada:** Determinación de pesos en la capa de salida

2.1 Fase No supervisada

Determinación de Centros	Determinación de Desviaciones
<ul style="list-style-type: none">• Algoritmos de K-medias• Mapas de Kohonen	<p>Se deben calcular de manera que:</p> <ul style="list-style-type: none">• Cada neurona de la capa Oculta se active en una región de espacios de entradas• El solapamiento de las zonas de activación de una neurona sea lo más ligero posible, para suavizar así la interpolación

Determinación de anchura

Se suele escoger una única anchura.

Cuando es distinta, se utilizan varias aproximaciones:

- ▣ **Media Uniforme** de las distancias euclídeas del centro C_i a los p centros más cercanos.

$$a_i = \frac{1}{p} \sum_p \|C_i - C_p\|$$

- ▣ **Media geométrica** de la distancia del centro C_i a sus dos vecinos más cercanos, C_t y C_s :

$$a_i = \sqrt{\|C_i - C_t\| \|C_i - C_s\|}$$

2.2 Fase Supervisada

- Minimización de error dada por la salida de la red (LMS):

$$E = \frac{1}{N} \sum_{n=1}^N e(n)$$

Donde:

$$e(n) = \frac{1}{2} \sum_{k=1}^r (y_k(n) - z_k(n))^2$$

$$\Delta w_{ik} = \gamma \cdot (y_k - z_k) \cdot G_i(n)$$