

## **REDES AUTO-ORGANIZÁVEIS**

### **SELF-ORGANIZING MAP (SOM)**

## **APRENDIZADO COMPETITIVO**

- Os algoritmos de aprendizado não-supervisionado são geralmente baseados em uma forma de competição entre os neurônios. O método mais comum é chamado **aprendizado competitivo**.

## APRENDIZADO COMPETITIVO

- aprendizado competitivo é uma forma de aprendizado que divide o conjunto de padrões de entrada em grupos inerentes aos dados. Em sua forma mais simples - Winner-Take-All

## APRENDIZADO COMPETITIVO

### Características Básicas

- Regra de Propagação  $y_j = \mathbf{XW}_j$
- Função de ativação: degrau (para o neurônio vencedor)
- Topologia: uma única camada de processadores
- Algoritmo de aprendizado: não-supervisionado  $\Delta w_{ij} = \alpha(x_i - w_{ij})$
- Valores de entrada: binário/contínuo

*OBS.: Regras de Propagação também pode ser*

$$y_j = \|x(n) - w_j\|, \quad j=1, 2, \dots, l$$

## APRENDIZADO COMPETITIVO

### Funcionamento

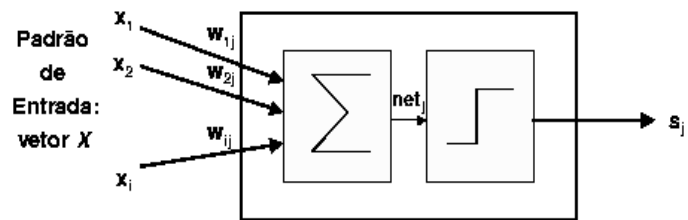
- Os vetores  $\mathbf{X}$  e  $\mathbf{W}_j$  devem ser normalizados;
- Somente o neurônio vencedor é ativado  
(neurônio com maior valor  $y$ )  
 $s_j = 1$  se  $y_j > y_k, \forall j \neq k$   
 $s_j = 0$  caso contrario
- Somente os pesos do neurônio vencedor e seus vizinhos pre-definidos são atualizados.

## APRENDIZADO COMPETITIVO

- Não existe conhecimento a respeito da classe a que o padrão pertence;
- Aprendizado depende das entradas e de suas densidades de probabilidade;
- Precisa de um grande conjunto de dados redundância para adquirir conhecimento das propriedades estatísticas dos padrões;
- Dependência do histórico de apresentação dos padrões

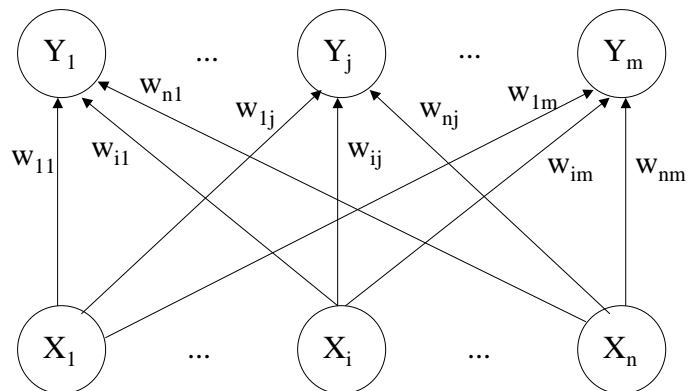
## MAPA DE KOHONEN

### • *Elemento Processador*

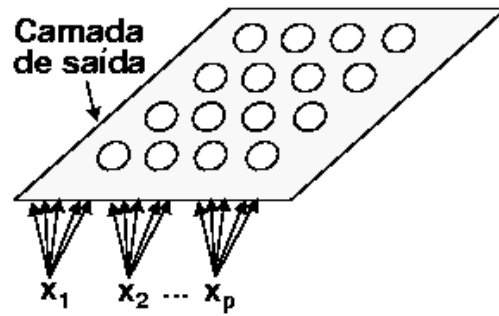


## MAPA DE KOHONEN

### Topologia



## MAPA DE KOHONEN



## MAPA DE KOHONEN

Dada uma amostra  $X$  do espaço de entrada representando um padrão de ativação aplicado à rede, três processos estarão envolvidos na formação do mapa auto-organizável:

*Competição*

*Cooperação*

*Adaptação Sináptica*

## MAPA DE KOHONEN

**Competição:** Para cada padrão de entrada, os neurônios na rede computam seus respectivos valores de uma função discriminante. Esta função discriminante fornece a base para a competição entre os neurônios e aquele que melhor-equivale ao valor da função discriminante é declarado o *neurônio vencedor*.

## MAPA DE KOHONEN

**Cooperação:** O neurônio vencedor determina a localização espacial de uma vizinhança topológica de neurônios excitados, fornecendo assim a base para a cooperação entre neurônios vizinhos.

## MAPA DE KOHONEN

**Adaptação Sináptica:** Este último mecanismo permite aos neurônios excitados incrementar seus valores individuais da função discriminante em relação ao padrão de entrada através de ajustes adequados aplicados à seus pesos sinápticos.

## MAPA DE KOHONEN

**Adaptação Sináptica (cont.):** Este ajuste é feito de tal maneira que a resposta do neurônio vencedor à uma subsequente aplicação de um padrão similar é desenvolvido, e pode ser alcançado através da fórmula

$$w_j(n+1) = w_j(n) + \eta(n)h_{j,i(x)}(n)(x(n) - w_j(n))$$

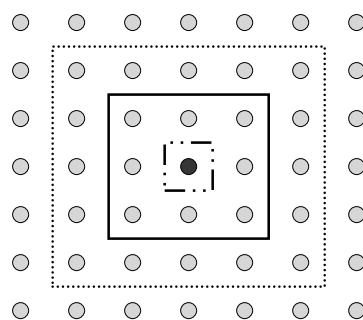
onde  $\eta(n)$  é um *parâmetro de taxa de aprendizado*, e  $h_{j,i(x)}(n)$  uma *função de vizinhança* em torno de um neurônio vencedor.

## MAPA DE KOHONEN

Assim, após repetidas apresentações de dados de treinamento, os pesos sinápticos tendem a seguir a distribuição do vetor de entrada devido à esta atualização da vizinhança. O algoritmo portanto leva a uma ordenação topológica do mapa de característica em relação ao espaço de entrada, no sentido que os neurônios adjacentes terão vetores de pesos similares.

## MAPA DE KOHONEN

**Função de Vizinhança**



R = 2    ·····

R = 1    ———

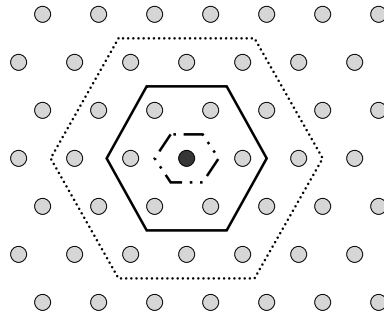
R = 0    - - - -

Vizinhança de grade retangular



## MAPA DE KOHONEN

Função de  
Vizinhança



R = 2 .....  
R = 1 ———  
R = 0 - - - -

Vizinhança de grade Hexagonal

## MAPA DE KOHONEN

**Análise qualitativa 1:**

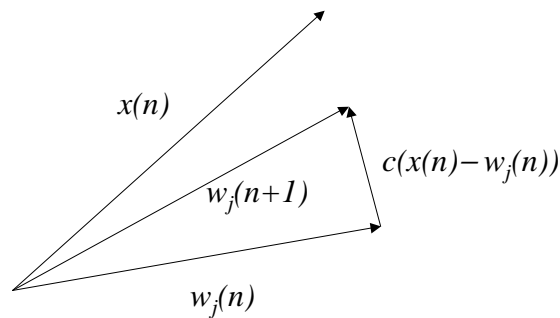
$$w_j(n+1) = w_j(n) + \eta(n)h_{j,i(x)}(n)(x(n) - w_j(n))$$

- Se  $x_i(n) > w_{ij}(n)$ ,  $w_{ij}(n+1) = w_{ij}(n) + \text{valor positivo}$   
Então, valor do  $w_{ij}$  cresce.
- Se  $x_i(n) < w_{ij}(n)$ ,  $w_{ij}(n+1) = w_{ij}(n) + \text{valor negativo}$   
Então, valor do  $w_{ij}$  diminui.
- Portanto,  $w_{ij}$  aproxima  $x_i$

## MAPA DE KOHONEN

### *Análise qualitativa 2:*

$$w_j(n+1) = w_j(n) + \eta(n)h_{j,i(x)}(n)(x(n) - w_j(n))$$



## MAPA DE KOHONEN - ALGORITMO

- Passo 0: Inicialize os pesos  $w_{ij}$ , inicialize os parâmetros de vizinhança; inicialize os parâmetros de taxa de aprendizagem
- Passo 1: Enquanto a condição de termina é falsa, faça passos 2-8
- Passo 2: Para cada vetor de entrada  $\mathbf{x}$ , faça passos 3-5
- Passo 3: Para cada  $j$ , calcula função discriminante;
- Passo 4: Encontra a índice  $J$  tal que  $D(J)$  é a mínima;
- Passo 5: Para todos unidades  $j$  em uma vizinhança específica do  $J$ , e para todos  $i$ :
- $$w_{ij}(\text{new}) = w_{ij}(\text{old}) + \alpha[x_i - w_{ij}(\text{old})]$$
- Passo 6: Atualiza a taxa de aprendizagem;
- Passo 7: Reduz região de vizinhança no tempo específico;
- Passo 8: Testa a condição de termina.

## MAPA DE KOHONEN

### Exemplo:

Usa o SOM para clusteriza 4 vetores:

(1, 1, 0, 0), (0, 0, 0, 1), (1, 0, 0, 0) e (0, 0, 1, 1)

Suponhamos que o numero máximo de clusters podem ser formados são  $m = 2$  e a taxa de aprendizagem é  $\alpha(0) = 0.6$

$$\alpha(t+1) = 0.5 \alpha(t)$$

## MAPA DE KOHONEN

Passo 0: Raio inicial  $R = 0$ , taxa de aprendizagem inicial

$$\alpha(0) = 0.6$$

Inicialize a matriz de pesos,

$$\begin{bmatrix} 0.2 & 0.8 \\ 0.6 & 0.4 \\ 0.5 & 0.7 \\ 0.9 & 0.3 \end{bmatrix}$$

## MAPA DE KOHONEN

Passo 1: Começa o treinamento

Passo 2: Para o primeiro vetor (1, 1, 0, 0), faça Passos 3-5

Passo 3:  $D(1) = (0.2-1)^2 + (0.6-1)^2 + (0.5-0)^2 + (0.9-0)^2 = 1.86$

$$D(2) = (0.8-1)^2 + (0.4-1)^2 + (0.7-0)^2 + (0.3-0)^2 = 0.98$$

Passo 4: O vetor de entrada é mais próximo o nó 2.

Então,  $J = 2$

Passo 5: Os pesos da unidade vencedor é atualizados:

$$w_{i2}(\text{new}) = w_{i2}(\text{old}) + 0.6[x_i - w_{i2}(\text{old})]$$

Isso dá a matriz de pesos:

$$\begin{bmatrix} 0.2 & 0.92 \\ 0.6 & 0.76 \\ 0.5 & 0.28 \\ 0.9 & 0.12 \end{bmatrix}$$

## MAPA DE KOHONEN

Passo 2: Para o segundo vetor (0, 0, 0, 1), faça Passos 3-5

Passo 3:  $D(1) = (0.2-0)^2 + (0.6-0)^2 + (0.5-0)^2 + (0.9-1)^2 = 0.66$

$$D(2) = (0.92-0)^2 + (0.76-0)^2 + (0.28-0)^2 + (0.12-1)^2 = 2.2768$$

Passo 4: O vetor de entrada é mais próximo a unidade 1.

Então,  $J = 1$

Passo 5: Atualiza a primeira coluna da matriz de pesos:

$$\begin{bmatrix} 0.08 & 0.92 \\ 0.24 & 0.76 \\ 0.20 & 0.28 \\ 0.96 & 0.12 \end{bmatrix}$$

## MAPA DE KOHONEN

Passo 2: Para o terceiro vetor (1, 0, 0, 0), faça passos 3-5

$$\text{Passo 3: } D(1) = (0.08-1)^2 + (0.24-0)^2 + (0.20-0)^2 + (0.96-0)^2 = 1.8656$$

$$D(2) = (0.92-1)^2 + (0.76-0)^2 + (0.28-0)^2 + (0.12-0)^2 = 0.6768$$

Passo 4: O vetor de entrada é mais próximo a unidade 2.

$$\text{Então, } J = 2$$

Passo 5: Atualiza a segunda coluna da matriz de pesos:

$$\begin{bmatrix} 0.08 & 0.968 \\ 0.24 & 0.304 \\ 0.20 & 0.112 \\ 0.96 & 0.048 \end{bmatrix}$$

## MAPA DE KOHONEN

Passo 2: Para o quarto vetor (0, 0, 1, 1), faça Passos 3-5

$$\text{Passo 3: } D(1) = (0.08-0)^2 + (0.24-0)^2 + (0.20-1)^2 + (0.96-1)^2 = 0.7056$$

$$D(2) = (0.968-0)^2 + (0.304-0)^2 + (0.112-1)^2 + (0.048-1)^2 \\ = 2.724$$

Passo 4: O vetor de entrada é mais próximo a unidade 1.

$$\text{Então, } J = 1$$

Passo 5: Atualiza a primeira coluna da matriz de pesos:

$$\begin{bmatrix} 0.032 & 0.968 \\ 0.096 & 0.304 \\ 0.680 & 0.112 \\ 0.984 & 0.048 \end{bmatrix}$$

## MAPA DE KOHONEN

Passo 6: Reduz a taxa de aprendizagem  $\alpha = 0.5*(0.6) = 0.3$

Agora a equação para atualizar os pesos é

$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + 0.3[x_i - w_{ij}(\text{old})]$$

A matriz de pesos após a segunda época de treinamento é

$$\begin{bmatrix} 0.016 & 0.980 \\ 0.047 & 0.360 \\ 0.630 & 0.055 \\ 0.999 & 0.024 \end{bmatrix}$$

.....

## MAPA DE KOHONEN

O resultado de 100 iterações (épocas) são (a taxa de aprendizagem diminui de 0.6 para 0.01):

Iteração 0

$$\begin{bmatrix} 0.2 & 0.8 \\ 0.6 & 0.4 \\ 0.5 & 0.7 \\ 0.9 & 0.3 \end{bmatrix}$$

Iteração 1

$$\begin{bmatrix} 0.032 & 0.968 \\ 0.096 & 0.304 \\ 0.680 & 0.112 \\ 0.984 & 0.048 \end{bmatrix}$$

Iteração 2

$$\begin{bmatrix} 0.016 & 0.980 \\ 0.047 & 0.360 \\ 0.630 & 0.055 \\ 0.999 & 0.024 \end{bmatrix}$$

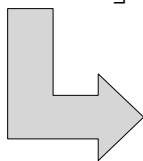
## MAPA DE KOHONEN

$$\begin{array}{l} \text{Iteração 10} \\ \text{Iteração 50} \end{array} \begin{bmatrix} 1.5e-7 & 1.0000 \\ 4.6e-7 & 0.3700 \\ 0.6300 & 5.4e-7 \\ 1.0000 & 2.3e-7 \end{bmatrix} \begin{bmatrix} 1.9e-19 & 1.0000 \\ 5.7e-15 & 0.4700 \\ 0.5300 & 6.6e-15 \\ 1.0000 & 2.8e-15 \end{bmatrix}$$

$$\text{Iteração 100:} \begin{bmatrix} 6.7e-17 & 1.0000 \\ 2.0e-16 & 0.4900 \\ 0.5100 & 2.3e-16 \\ 1.0000 & 1.0e-16 \end{bmatrix}$$

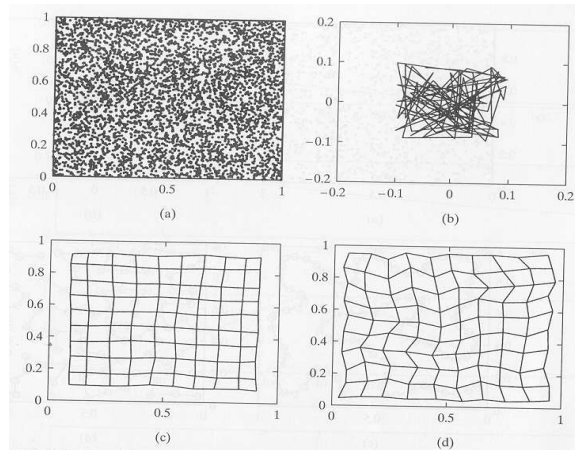
## MAPA DE KOHONEN

$$\text{Iteração 100:} \begin{bmatrix} 6.7e-17 & 1.0000 \\ 2.0e-16 & 0.4900 \\ 0.5100 & 2.3e-16 \\ 1.0000 & 1.0e-16 \end{bmatrix}$$


$$\begin{bmatrix} 0 & 1.0 \\ 0 & 0.5 \\ 0.5 & 0 \\ 1.0 & 0 \end{bmatrix}$$

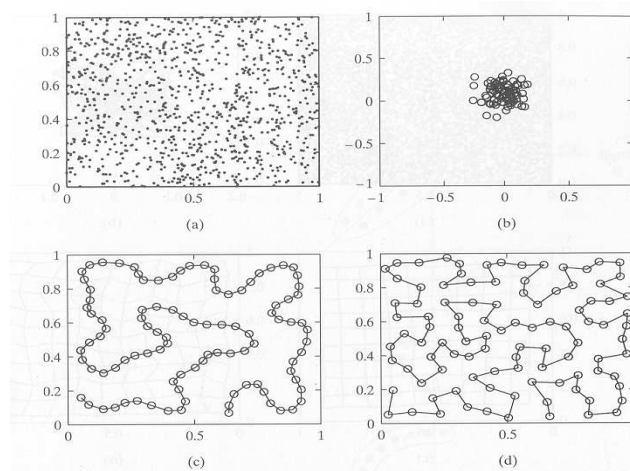
A primeira coluna representa a media de dois vetores que são colocados na cluster 1 e a segunda coluna representa a media de dois vetores que são colocados na cluster 2

## MAPA DE KOHONEN – APLICAÇÃO - 1



**FIGURE 9.8** (a) Input data distribution. (b) Initial condition of the two-dimensional lattice. (c) Condition of the lattice at the end of the ordering phase. (d) Condition of the lattice at the end of the convergence phase.

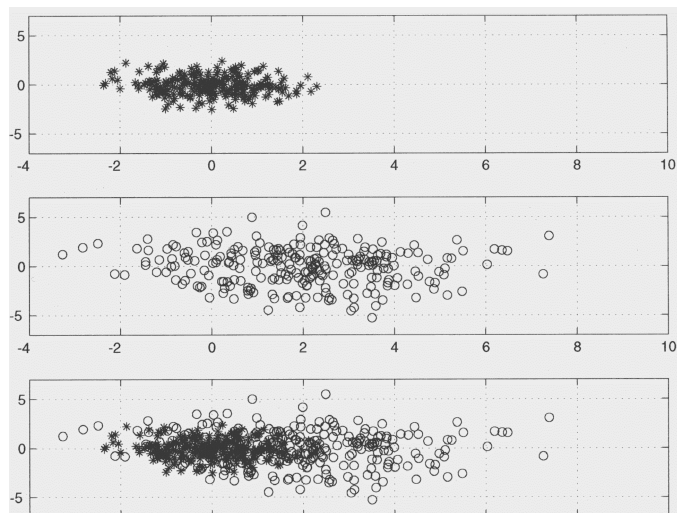
## MAPA DE KOHONEN – APLICAÇÃO - 1



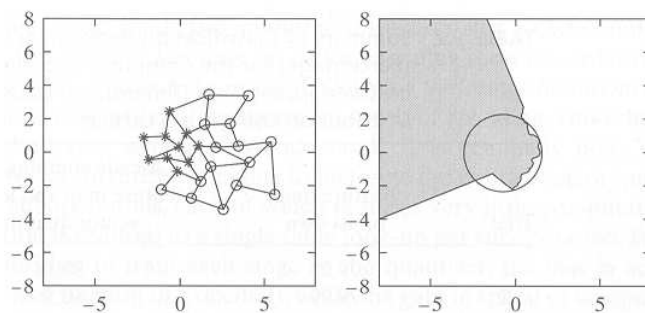
**FIGURE 9.9** (a) Two-dimensional input data distribution. (b) Initial condition of the one-dimensional lattice. (c) Condition of the lattice at the end of the ordering phase. (d) Condition of the lattice at the end of the convergence phase.



## MAPA DE KOHONEN – APLICAÇÃO - 2



## MAPA DE KOHONEN – APLICAÇÃO - 2



## MAPA DE KOHONEN – APLICAÇÃO - 3

TABLE 9.3 Animal Names and Their Attributes

Animal		Dove	Hen	Duck	Goose	Owl	Hawk	Eagle	Fox	Dog	Wolf	Cat	Tiger	Lion	Horse	Zebra	Cow
is	small	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0
	medium	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	big	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
has	2 legs	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	4 legs	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	hair	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	hooves	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
	mane	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0
	feathers	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
likes to	hunt	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0
	run	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0
	fly	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
	swim	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

## MAPA DE KOHONEN – APLICAÇÃO - 3

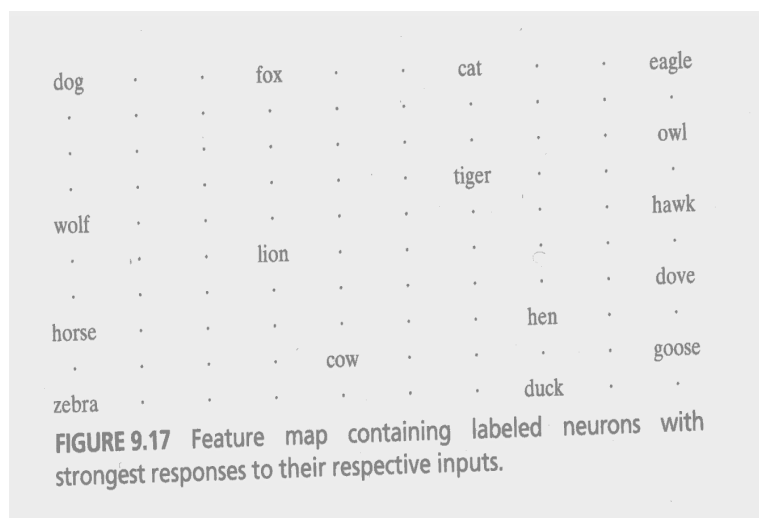


FIGURE 9.17 Feature map containing labeled neurons with strongest responses to their respective inputs.

## MAPA DE KOHONEN – APLICAÇÃO - 3

dog	dog	fox	fox	fox	cat	cat	cat	eagle	eagle
dog	dog	fox	fox	fox	cat	cat	cat	eagle	eagle
wolf	wolf	wolf	fox	cat	tiger	tiger	tiger	owl	owl
wolf	wolf	lion	lion	lion	tiger	tiger	tiger	hawk	hawk
wolf	wolf	lion	lion	lion	tiger	tiger	tiger	hawk	hawk
wolf	wolf	lion	lion	lion	owl	dove	hawk	dove	dove
horse	horse	lion	lion	lion	dove	hen	hen	dove	dove
horse	horse	zebra	cow	cow	cow	hen	hen	dove	dove
zebra	zebra	zebra	cow	cow	cow	hen	hen	duck	goose
zebra	zebra	zebra	cow	cow	cow	duck	duck	duck	goose

**FIGURE 9.18** Semantic map obtained through the use of simulated electrode penetration mapping. The map is divided into three regions representing: birds, peaceful species, and hunters.