

## Camada de Kohonen

### Classificação por Similaridade

Classes agrupam elementos “similares” entre si



**versus classificadores arbitrários**

### 1 - Classificação por Similaridade –

**Critérios de pertinência à uma classe**

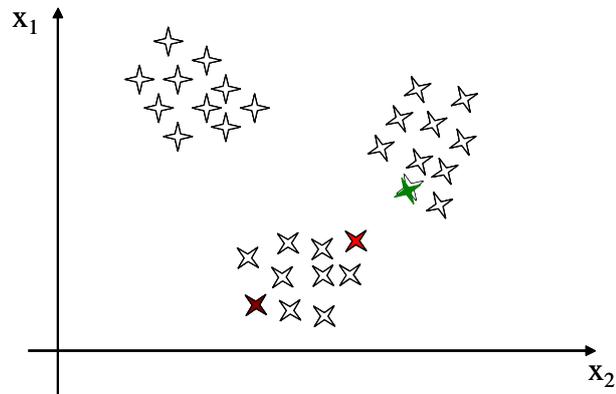
objetos físicos  $\mathcal{O}_1$   $\mathcal{O}_2$   $\longrightarrow$  objetos matemáticos  $\underline{x}_1$   $\underline{x}_2$

similaridade física  $\longrightarrow$  similaridade matemática

$$\mathcal{O}_1 \approx \mathcal{O}_2 \quad \underline{x}_1 \cong \underline{x}_2 \quad \text{ou} \quad |\underline{x}_1 - \underline{x}_2| \ll$$

## Classificadores por similaridade

### Critérios de Pertinência à uma Classe



### Critério básico

Dois elementos pertencem à mesma classe se estão próximos entre si

### Critério: k vizinhos mais próximos.

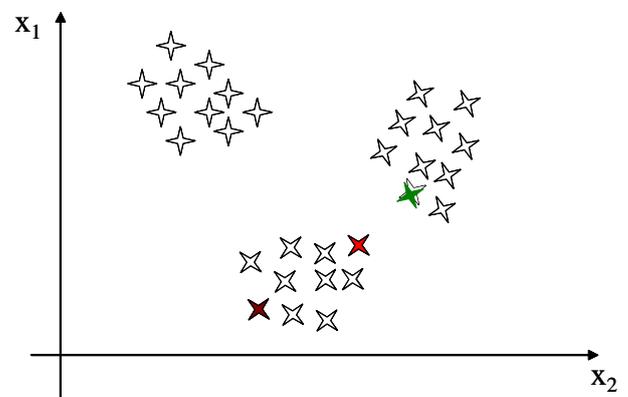
Considere os k elementos mais próximos da entrada que se pretende classificar. A entrada pertence à classe à qual pertencem a maioria dos k vizinhos.

### Critério simplificado: vizinho mais próximo.

Se  $k=1$  uma entrada pertence a uma classe se seu vizinho mais próximo é um elemento desta classe

### pouco práticos

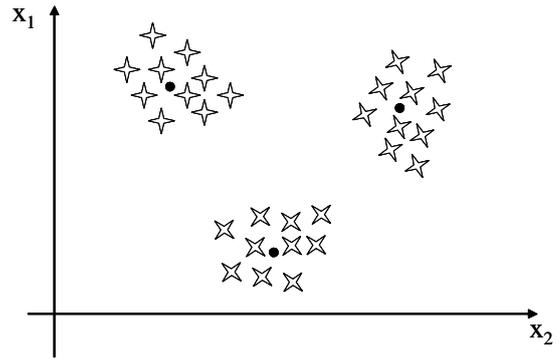
a descrição de uma classe exige a utilização de todos seus elementos (domínio da classe)



## Critérios baseados no Padrão de classe

Padrão  $\underline{w}_i$  da classe  $C_i$

$$\underline{w}_i = \underset{\forall \underline{x}_j \in C_i}{E} \underline{x}_j = \frac{1}{N_i} \sum_{\forall \underline{x}_j \in C_i, j=1}^{N_i} \underline{x}_j$$



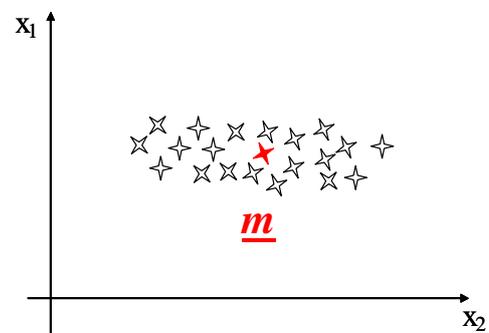
a **média** ou **baricentro** é escolhido como padrão da classe porque minimiza a **dissimilaridade** e o **erro de representação** da classe pelo seu padrão

**Dispersão média intra classe ou erro (médio quadrático) de representação da classe  $C_j$  para o seu padrão  $\vec{p}_j$**

$$\sigma_j^2 = \underset{\forall x_i \in C_j}{E} \|x_i - p_j\|^2 = \frac{1}{n_j} \sum_{\forall x_i \in C_j} \|x_i - p_j\|^2$$

por componente k

$$\sigma_{jk}^2 = \underset{\forall x_i \in C_j}{E} (x_{ik} - p_{jk})^2 = \frac{1}{n_j} \sum_{\forall x_i \in C_j} (x_{ik} - p_{jk})^2$$



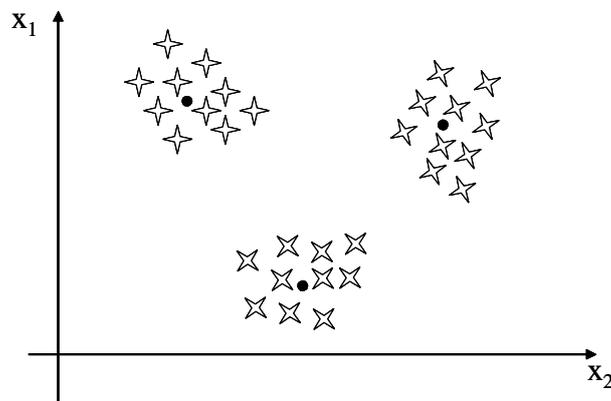
após pequena álgebra

$$\frac{\partial \sigma_{jk}^2}{\partial p_{jk}} = 0 \quad \Rightarrow \quad p_{jk} = m_{jk} \quad \Rightarrow \quad \vec{p}_j = \vec{m}_j$$

**o padrão que minimiza a dispersão intra classe - e o erro de representação da classe pelo seu padrão - é o seu baricentro**

## 1.1 - Critério de pertinência 1 :

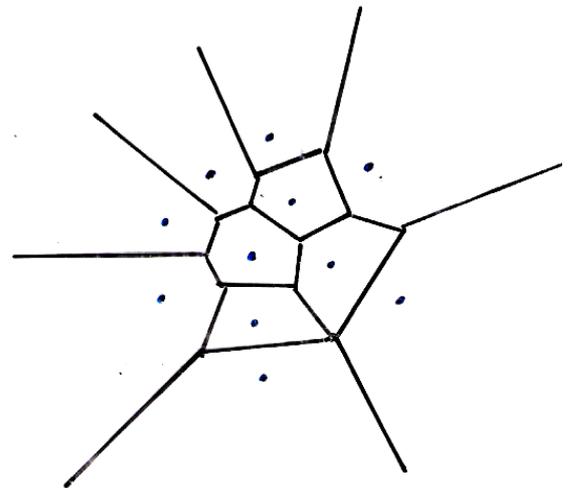
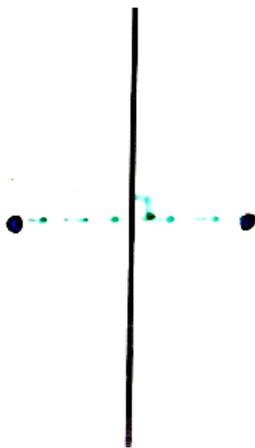
### Padrão mais similar à entrada



$$\underline{x}_i \in C_i \quad \text{sse} \quad |\underline{x} - \underline{w}_i|^2 < |\underline{x} - \underline{w}_j|^2 \quad \forall j \neq i \quad \text{eq (1)}$$

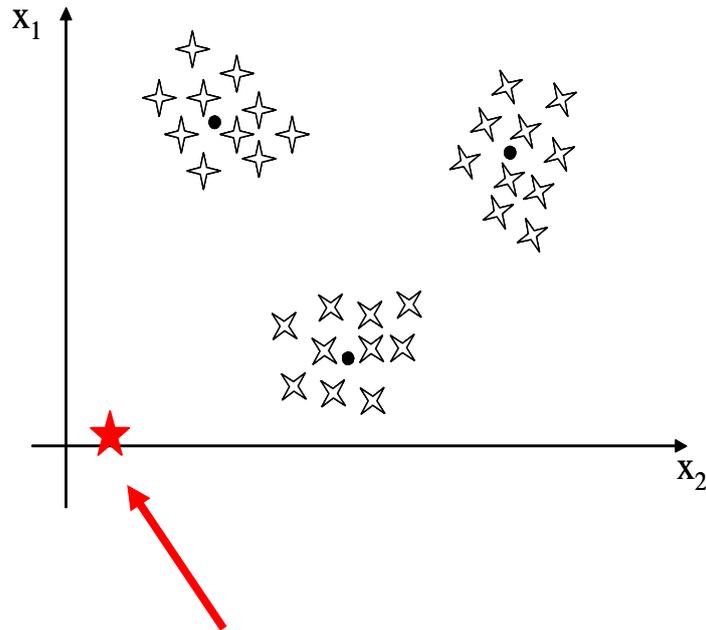
porque minimiza a dispersão intra-classes total

### Separadores para o critério 1

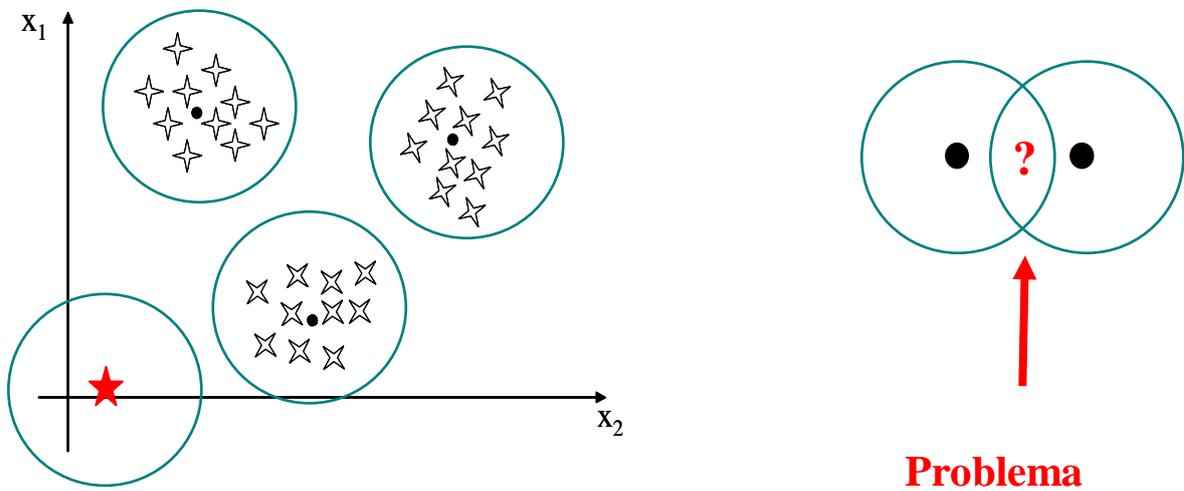


Tecelagem (diagrama) de Voronoi

**Problema:**



**1.2 - Critério de pertinência 2 – Padrão que satisfaz uma similaridade mínima**

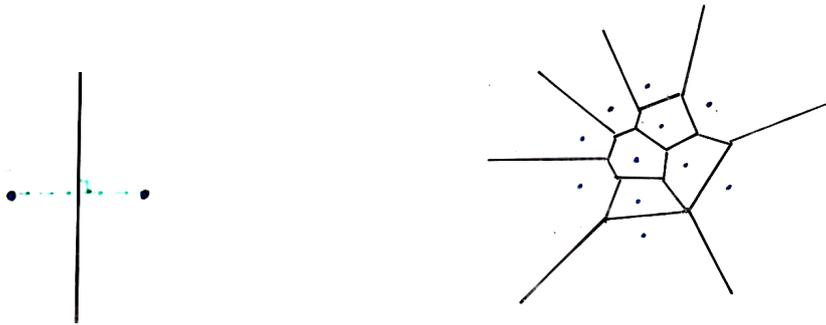


$$\underline{x}_i \in C_i \quad \text{sse} \quad |\underline{x} - \underline{w}_i|^2 < r_0^2 \quad \text{eq (2)}$$

$r_0$  – raio de similaridade

## Separadores

**Critério 1 - padrão mais similar (mais próximo) a entrada Eq(1)**

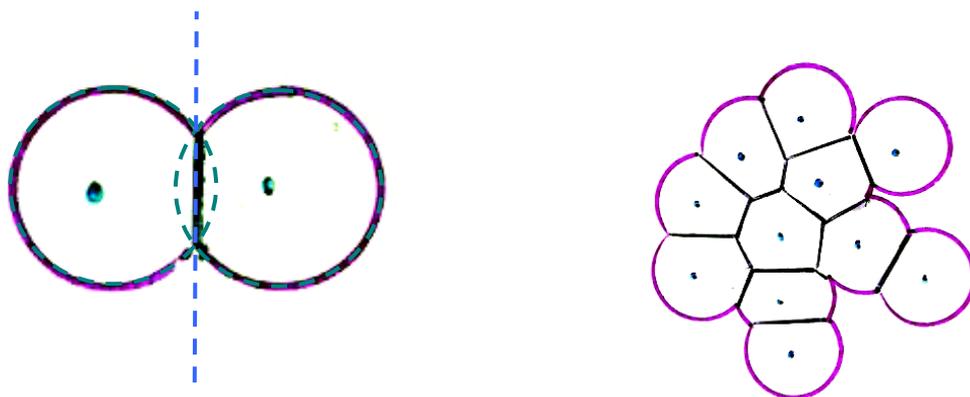


**Critério 2 - mínima similaridade exigida Eq (2)**



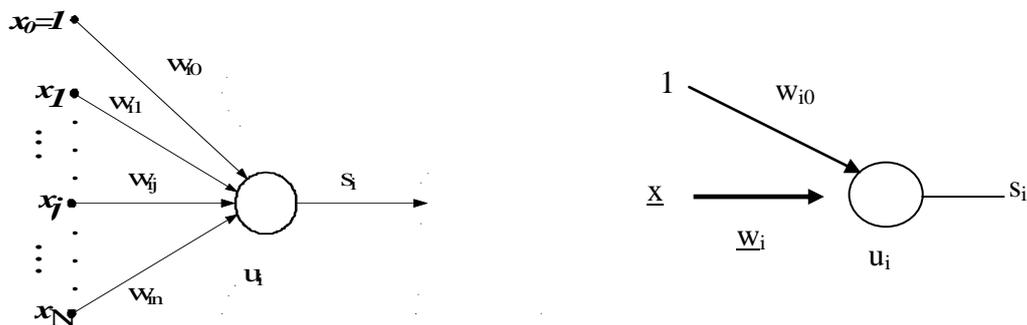
**1.3 - Critérios 1 e 2, Eq(1) + Eq(2)**

**padrão mais similar & mínima similaridade atingida**



## 2 - Redes Neurais

### Neurônio para redes feedforward



$$u_i = \sum_{j=1}^N w_{ij} x_j + w_{i0} = \underline{w}_i^t \underline{x} + w_{i0}$$

$$\tilde{y}_k = \varphi_k(u_k) = \begin{cases} u_k & \text{neurônio linear} \\ \text{tgh}(u_k) & \text{neurônio tgh} \end{cases}$$

### Neurônio como **comparador de similaridade**

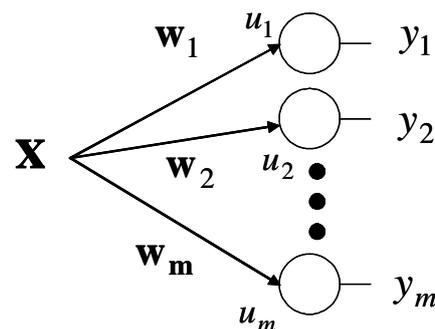
de uma entrada  $\underline{x}$  com dois padrões  $\underline{w}_i$  e  $\underline{w}_j$

$$d_i^2 = |\underline{x} - \underline{w}_i|^2 = |\underline{x}|^2 + |\underline{w}_i|^2 - 2\underline{w}_i^t \underline{x}$$

$$u_i = \underline{w}_i^t \underline{x} + w_{i0}$$

fazendo  $w_{i0} = -\frac{1}{2}|\underline{w}_i|^2$

$$u_i = \frac{1}{2}(|\underline{x}|^2 - d_i^2)$$



logo

$$u_i > u_j \iff d_i^2 < d_j^2$$

e  $u$  é um **comparador de similaridade**

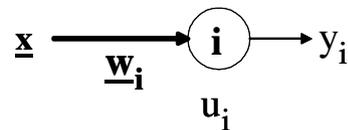
$y_i$  – depende dos outros neurônios

### Neurônio como **medidor de similaridade**

entre uma entrada  $\underline{x}$  e um padrão  $\underline{w}_i$

#### uma outra definição

$$u_i = -\|\underline{x} - \underline{w}_i\|^2 = -d_i^2 \leq 0$$



$u_i$  - medida de similaridade entre  $\underline{x}$  e  $\underline{w}_i$

$u_i = 0$  distância nula = máxima similaridade

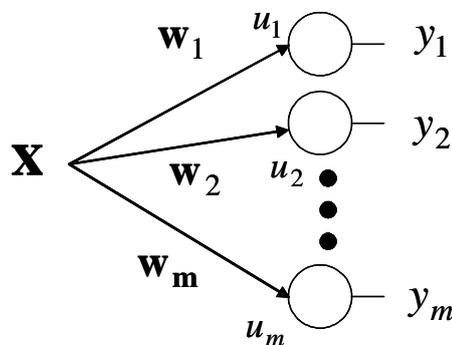
$y_i$  – depende dos outros neurônios

### 3- Camada de Kohonen

#### Template Matching

$$u_i = -d_i^2$$

**maior  $u_i$  =  
menor distância =  
maior similaridade**

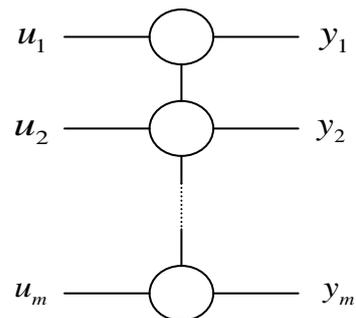


$u_i$  é uma medida de similaridade entre  $\underline{x}$  e  $\underline{w}_i$

#### Winner-takes-all

$$y_i = 1 \text{ sse } u_i > u_j \quad \forall j \neq i$$

$$y_i = 0 \text{ caso contrário}$$



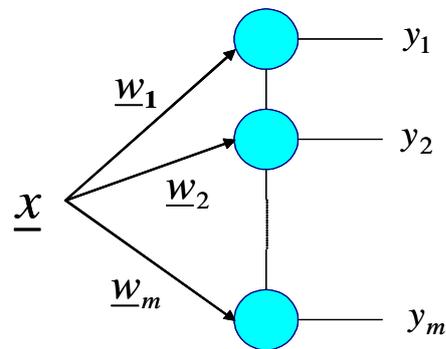
## Camada de Kohonen

[versão unidimensional, simplificada (D=0)]

Classe  $C_i$

Padrão  $\underline{w}_i$

Indicador  $y_i$



Se  $y_i = 1$  então  $\underline{x} \in C_i$

pele critério 1 (padrão mais similar a entrada).

## 4 - Treinamento da Camada de Kohonen

como um processo de otimização

### Treinamento supervisionado

- as classes de cada entrada (elemento)  $\underline{x}$  são conhecidas

Pares entrada-saída  $[\underline{x}, C(\underline{x})]$

### Treinamento como um processo de otimização:

**Treinamento em Batelada** - Todas as entradas (e suas classes) são disponíveis e processadas conjuntamente.

**Treinamento em “regra delta”** - as entradas (e suas classes) são apresentadas uma a uma.

## Alocar os padrões $\underline{x}$ das classes de forma a minimizar $F_{in}$

$$F_{in} = \sum_{\forall C_j} F_j$$

$$F_j = \sum_{\forall \vec{x} \in C_j} |\vec{x} - \vec{w}_j|^2 = \sum_{\forall \vec{x} \in C_j} \sum_{\forall k} (x_k - w_{jk})^2 = \sum_{\forall k} \sum_{\forall \vec{x} \in C_j} (x_k - w_{jk})^2$$

como já visto, após pequena álgebra

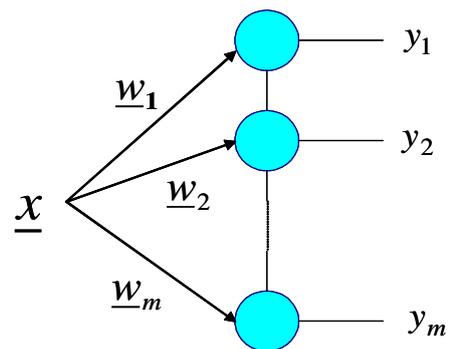
$$\frac{\partial}{\partial w_{jk}} \sum_{\forall \vec{x} \in C_j} (x_k - w_{jk})^2 = 0 \quad \Rightarrow \quad w_{jk} = m_{jk} \quad \Rightarrow \quad \vec{w}_j = \vec{m}_j$$

## Treinamento em Batelada

todas as entradas estão disponíveis  
e são processadas simultaneamente

$$\vec{w}_j = \vec{m}_j = \frac{1}{n_j} \sum_{\forall \vec{x} \in C_j} \vec{x}$$

fim do treinamento

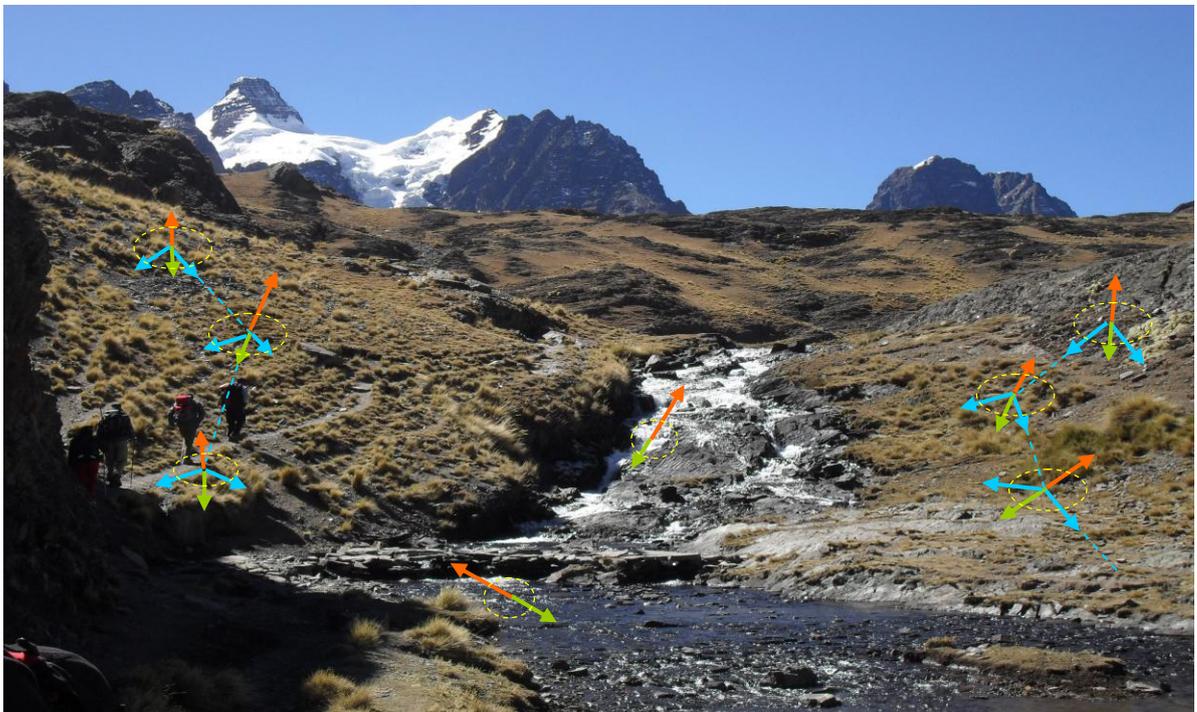
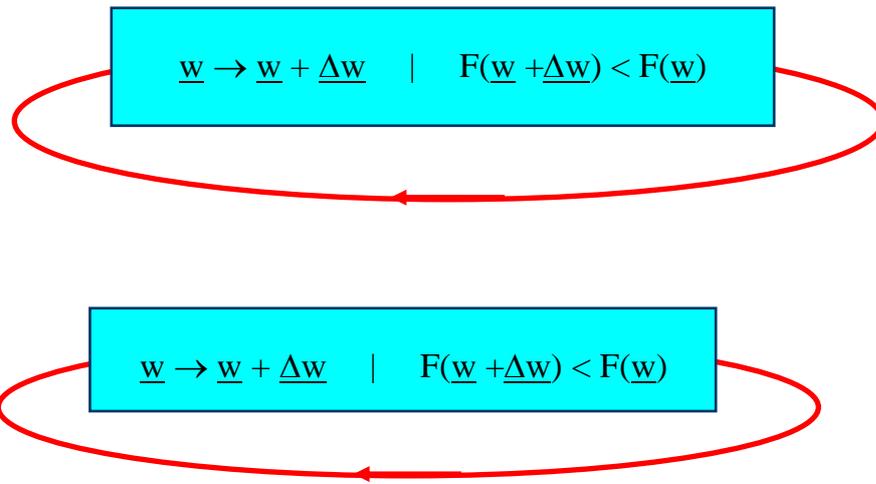


## Treinamento em “regra delta”

cada entrada é apresentada e a rede é treinada

### Otimização: Métodos Numéricos Recursivos

#### Descida continuada



Em que direção se mover para reduzir  $F_{in}$  ?

Gradiente:

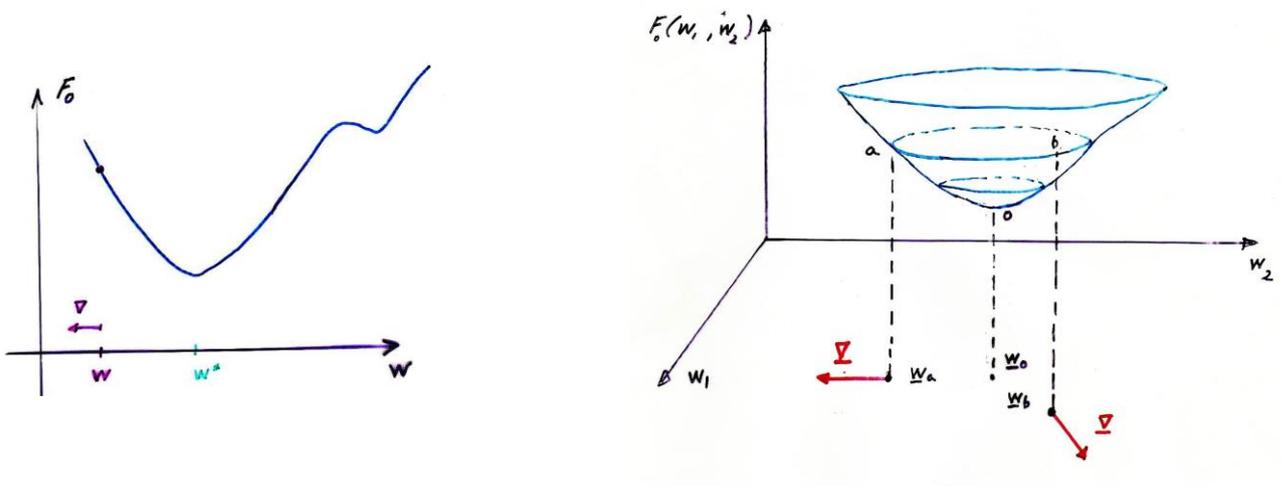
$$F_0(w_1, w_2, \dots) = F(\underline{w})$$

$$\underline{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \end{bmatrix} \quad \nabla_{\underline{w}} F_0 = \nabla = \begin{bmatrix} \frac{\partial F_0}{\partial w_1} \\ \frac{\partial F_0}{\partial w_2} \\ \vdots \end{bmatrix}$$

Mover na direção e sentido contrário do gradiente reduz F

Gradiente descendente ou descida por gradiente

$$\underline{w} \xrightarrow{\text{red arrow}} \underline{\Delta w} = -\alpha \nabla F(\underline{w}) \xrightarrow{\text{red arrow}} \underline{w} = \underline{w} + \underline{\Delta w}$$



## Que direção garante o decréscimo em $F_{in}$ ?

Pelo método do **gradiente descendente** o decréscimo é garantido se, considerando todas as entradas  $\underline{x}$  simultaneamente

$$\begin{aligned}\Delta w_{jk} &= -\alpha \frac{\partial F_{in}}{\partial w_{jk}} = -\alpha \frac{\partial F_j}{\partial w_{jk}} = -\alpha \frac{\partial}{\partial w_{jk}} \sum_{\forall \vec{x} \in C_j} |\vec{x} - \vec{w}_j|^2 = \\ &= -\alpha \frac{\partial}{\partial w_{jk}} \sum_{\forall \vec{x} \in C_j} (x_k - w_{jk})^2 = \alpha \mathbf{E}_{\forall \vec{x} \in C_j} (x_k - w_{jk})\end{aligned}$$

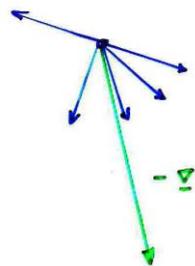
### Regra Delta

$$\Delta \vec{w}_j = \alpha (\vec{x} - \vec{w}_j)$$

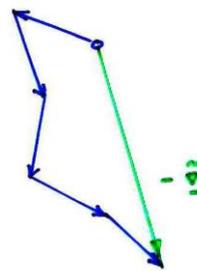
### Regra Delta

Pelo método da “regra delta” o decréscimo é garantido se, considerando todas as entradas  $\underline{x}$  uma a uma

$$\Delta \vec{w}_j = \alpha (\vec{x} - \vec{w}_j) \quad \text{ou} \quad \Delta w_{jk} = \alpha (x_k - w_{jk}) \quad \text{onde } \alpha \text{ é pequeno e positivo}$$



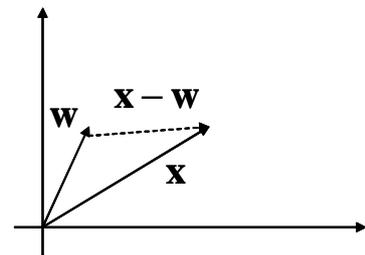
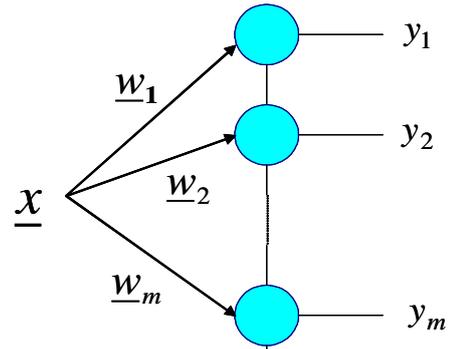
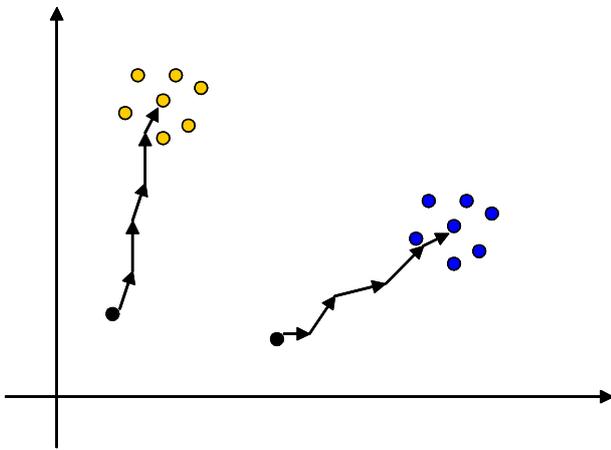
batelada



regra delta

### 4.1 - Treinamento da Camada de Kohonen por regra delta

#### Treinamento Supervisionado



#### Treinamento Supervisionado

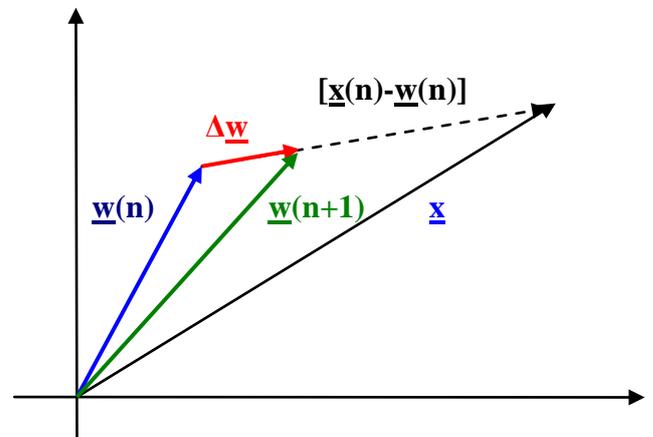
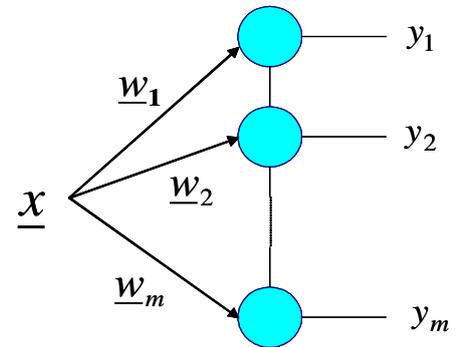
$$\underline{x}(n) \in C_i \Rightarrow y_i = 1$$

$$y_j = 0 \quad \forall j \neq i$$

$$\underline{w}_i(n+1) = \underline{w}_i(n) + \alpha [\underline{x}(n) - \underline{w}_i(n)]$$

$$= (1 - \alpha) \underline{w}_i(n) + \alpha \underline{x}(n)$$

$$\underline{w}_j(n+1) = \underline{w}_j(n) \quad \forall j \neq i$$



Onde estabiliza, qual o valor final de  $\underline{w}$  ?

Evolução do  $\underline{w}$  de uma classe

Passo n:  $\underline{x}(n)$   $\underline{w}(n)$

$x(n) \in C_i$

$$\begin{cases} \underline{w}_i(n+1) = (1-\alpha)\underline{w}_i(n) + \alpha\underline{x}(n) \\ \underline{w}_j(n+1) = \underline{w}_j(n) & \forall j \neq i \end{cases}$$

$\underline{w}(0)$

$$\underline{w}(1) = (1-\alpha)\underline{w}(0) + \alpha\underline{x}(1)$$

$$\begin{aligned} \underline{w}(2) &= (1-\alpha)\underline{w}(1) + \alpha\underline{x}(2) \\ &= (1-\alpha)[(1-\alpha)\underline{w}(0) + \alpha\underline{x}(1)] + \alpha\underline{x}(2) \\ &= (1-\alpha)^2\underline{w}(0) + \alpha(1-\alpha)\underline{x}(1) + \alpha\underline{x}(2) \end{aligned}$$

$$\begin{aligned} \underline{w}(3) &= (1-\alpha)\underline{w}(2) + \alpha\underline{x}(3) \\ &= (1-\alpha)[(1-\alpha)^2\underline{w}(0) + \alpha(1-\alpha)\underline{x}(1) + \alpha\underline{x}(2)] + \alpha\underline{x}(3) \\ &= (1-\alpha)^3\underline{w}(0) + \alpha(1-\alpha)^2\underline{x}(1) + \alpha(1-\alpha)\underline{x}(2) + \alpha\underline{x}(3) \end{aligned}$$

⋮

$$\begin{aligned} \underline{w}(n) &= (1-\alpha)^n \underline{w}(0) + \alpha[(1-\alpha)^{n-1} \underline{x}(1) + (1-\alpha)^{n-2} \underline{x}(2) + \dots + \underline{x}_n] \\ &= (1-\alpha)^n \underline{w}(0) + \alpha \sum_{i=1}^n (1-\alpha)^{n-i} \underline{x}(i) \end{aligned}$$

algumas aproximações:

$$0 < \alpha < 1 \quad 0 < (1 - \alpha) < 1 \quad n \gg 1 \quad (1 - \alpha)^n \rightarrow 0$$

$$1 + (1 - \alpha) + (1 - \alpha)^2 + (1 - \alpha)^3 + \dots = \sum_{i=0}^{\infty} (1 - \alpha)^i = \frac{1}{1 - (1 - \alpha)} = \frac{1}{\alpha}$$

$$n \gg 1 \quad \alpha \approx \frac{1}{\sum_{i=0}^{n-1} (1 - \alpha)^i}$$

$$\underline{w}(n) = (1 - \alpha)^n \underline{w}(0) + \alpha \sum_{i=1}^n (1 - \alpha)^{n-i} \underline{x}(i)$$

$$\cong \frac{\sum_{i=1}^n (1 - \alpha)^{n-i} \bar{x}(i)}{\sum_{i=0}^{n-1} (1 - \alpha)^i} = \frac{\sum_{i=0}^{n-1} (1 - \alpha)^i \bar{x}(n-i)}{\sum_{i=0}^{n-1} (1 - \alpha)^i}$$

**Média das entradas que pertencem a classe**

**ponderada geometricamente pelo tempo !**

$$\underline{w}_n \cong \frac{\sum_{i=1}^n (1 - \alpha)^{n-i} \bar{x}(i)}{\sum_{i=1}^n (1 - \alpha)^{n-i}} = \left( \frac{1}{\sum_{i=0}^{n-1} (1 - \alpha)^i} \right) \sum_{i=0}^{n-1} (1 - \alpha)^i \bar{x}(n-i)$$

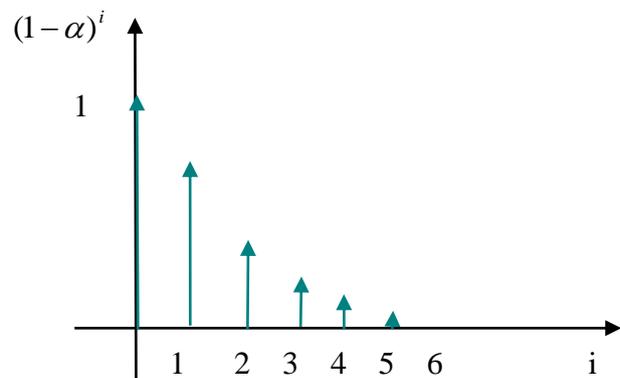
**instante atual = n**

**$\underline{x}(n-i)$  = entrada atrasada**

**de i intervalos de tempo**

**$(1 - \alpha)^i$  = ponderador**

**para a entrada  $\underline{x}(n-i)$**



**Note que se**

$$\alpha \cong 1$$

**e a estatística de  $\vec{x}$  for invariante no tempo**

$$\vec{w}_n \cong \left( \frac{1}{\sum_{i=0}^{n-1} (1-\alpha)^i} \right) \sum_{i=0}^{n-1} (1-\alpha)^i \vec{x}(n-i) \cong E[\vec{x}]$$

$$\vec{w}_n \cong E[\vec{x}]$$

## 4.2 – Tempo de medida

**Fim (prático) da soma ponderada (tempo de medida)**

Atraso zero

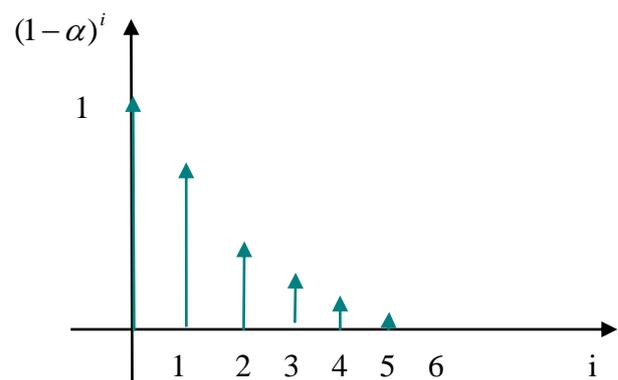
$$\text{Ponderador} = (1-\alpha)^0 = 1$$

Último atraso à ser considerado:

$$\text{Ponderador} = .02 = (1-\alpha)^i$$

$$i = \frac{\ln .02}{\ln(1-\alpha)} \approx \frac{-4}{-\alpha} \Big|_{\alpha \rightarrow 0}$$

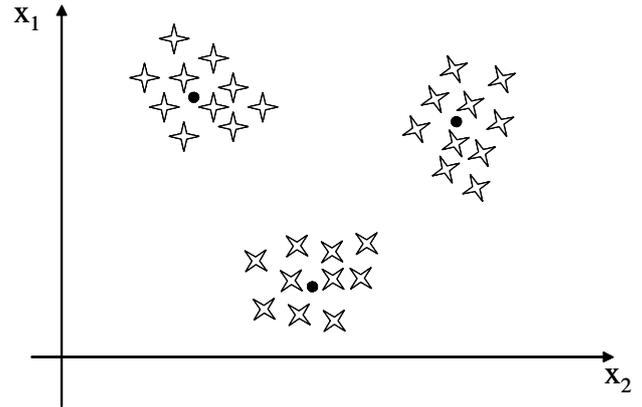
$$i \approx \frac{4}{\alpha}$$



### 4.3 - Erro na determinação do baricentro:

Se em uma classificação por similaridade cada elemento  $\underline{x}$  de uma classe  $C_k$  pode ser representado pelo padrão  $\underline{w}_k$  da classe adicionado de um vetor de ruído  $\underline{r}$  com média nula.

$$\underline{X} = \underline{W}_k + \underline{r}$$



Cada componente  $x_j$  de  $\underline{x}$  é então representada pela componente correspondente de  $\underline{w}_k$ ,  $w_j$ , adicionada de um ruído  $r_j$  de média nula e variância  $\sigma_{x_j}^2$

$$x_j = w_j + r_j$$

Com que precisão componente  $w_j$  é calculada pela camada de Kohonen, qual a sua variância  $\sigma_{w_j}^2$ ? A componente  $j$  de  $\underline{w}$ ,  $w_j$ , em um instante  $n \gg 1$  é dada por

$$w_j \cong \left( \frac{1}{\sum_{i=0}^n (1-\alpha)^i} \right) \sum_{i=0}^n (1-\alpha)^i x_j(n-i)$$

Sendo uma soma ponderada sua variância será dada por

$$\sigma_{w_j}^2 \cong \left( \frac{1}{\sum_{i=0}^{\infty} (1-\alpha)^i} \right)^2 \sum_{i=0}^{\infty} (1-\alpha)^{2i} \sigma_{x_j}^2 \quad \text{ou} \quad \sigma_{w_j}^2 \cong \frac{\alpha}{2} \sigma_{x_j}^2$$

#### 4.4 – Compromisso entre Erro vs. Tempo de Medida

Quanto menor  $\alpha$  mais precisamente o baricentro será determinado. Mas, como esperado, maior o tempo (número de passos) necessário para calculá-lo

$$\sigma_{wj}^2 \cong \frac{\alpha}{2} \sigma_{xj}^2 \quad \# \text{ passos} = i \cong \frac{4}{\alpha}$$

##### Exemplo:

$$\sigma_x = .05 \quad (5\%)$$

$$\sigma_w = .01 \quad (1\%) \text{ requerido}$$

$$\alpha = \frac{2\sigma_{wj}^2}{\sigma_{xj}^2} = .08 \quad \# \text{ passos} = \frac{4}{\alpha} = 50 \text{ passos}$$

Mas se:

$$\sigma_w = .001 \quad (.1\%) \text{ requerido}$$

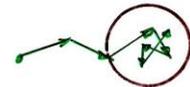
$$\alpha = \frac{2\sigma_{wj}^2}{\sigma_{xj}^2} = .0008 \quad \# \text{ passos} = \frac{4}{\alpha} = 5.000 \text{ passos !!}$$

#### 4.5 Considerações:

##### 4.5.1. Fim do treinamento ?



$$E[\Delta \underline{w}] = \underline{0}$$



$$\Delta \bar{w} = \alpha (\bar{x} - \bar{w})$$

$$E(\Delta \bar{w}) = E[\alpha (\bar{x} - \bar{w})] = \alpha [E(\bar{x}) - \bar{w}] = 0$$

$$\bar{w} = E(\bar{x})$$

### 4.5.2. Valores iniciais das sinapses

Irrelevantes ! Mas escolher o valor inicial como o primeiro elemento da classe acelera o treinamento, porque a sinapse já começa dentro da classe.



### 4.5.3 - Passo de treinamento

Escolher  $\alpha$  em função do tempo de medida e erro aceitável em  $\underline{w}$

#### 4.5.3.1 - Passo de treinamento variável no tempo e por sinapse

reduzir ao longo do tempo:

compromisso entre um ajuste rápido e erro pequeno

$$\alpha(n) = \alpha_0 e^{-\frac{n}{N_0}}$$

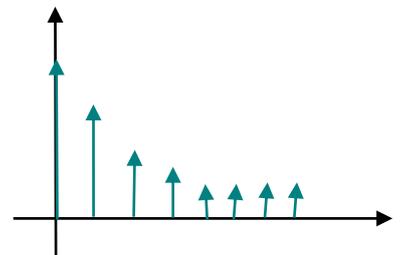
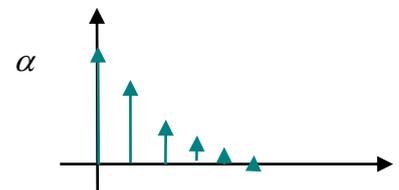
$$\alpha(0) = \alpha_0 \quad \alpha(n+1) = k \alpha(n) \quad \text{onde}$$

$$k = 1 - \frac{1}{N_0}$$

o processo acaba ( $\alpha(n) \ll \alpha_0$ ) para  $n > 4N_0$

alternativa para não acabar o treinamento:

$$\alpha(n) = \alpha_{\min} + \alpha_0 e^{-\frac{n}{N_0}}$$

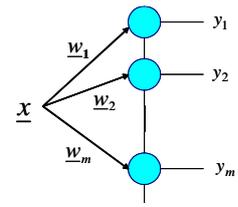


e as sinapses pouco ativadas (classes pouco populosas) ?

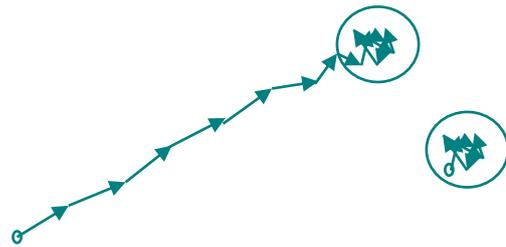
usar  $\alpha$  diferenciado por sinapse

$$\alpha_{w_j}(0) = \alpha_0 \quad \alpha_{w_j}(n_j + 1) = k \alpha_{w_j}(n_j)$$

onde  $n_j$  é o número de vezes que a sinapse  $w_j$  foi treinada.

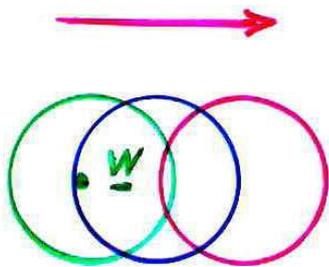


**Obs: Passo variável é menos importante se os valores iniciais das sinapses são elementos do cluster**

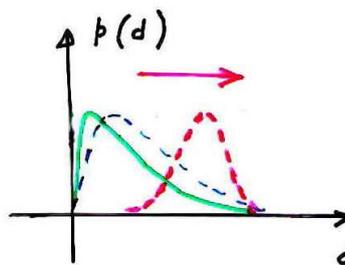


### 4.5.4 Treinamento dinâmico, adaptativo

Uma classe varia de posição no tempo. Como saber ?



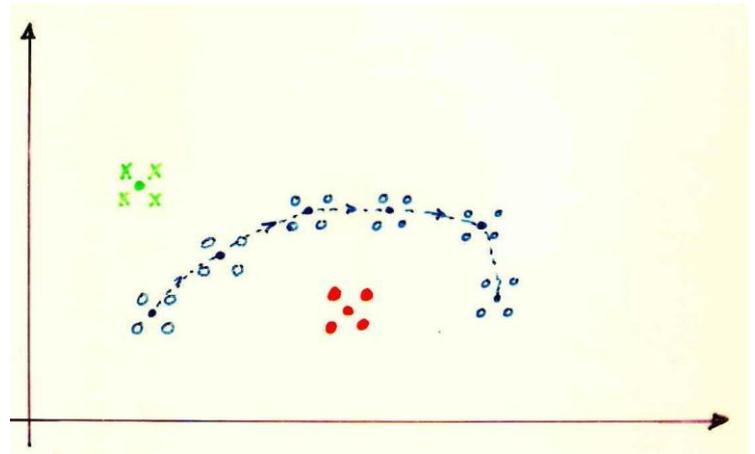
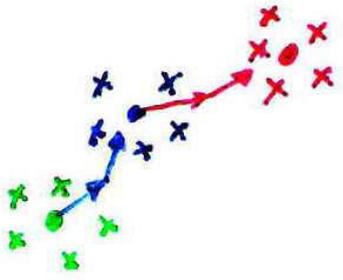
$$d^2 = | \underline{x} - \underline{w} |^2$$



**O valor médio de  $u_i = - d^2_i$  diminui**

**(o valor da distância média (ou a da moda) aumenta)**

**Como corrigir ? Ligar o treinamento até que  $d^2_i$  volte a seus valores normais.**



Se a variação do baricentro for lenta e o treinamento estiver ligado o padrão segue o baricentro

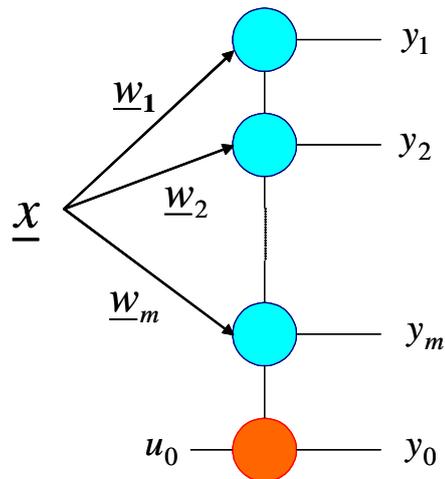
5 - E o segundo critério (similaridade mínima) ?

Camada de Kohonen (aumentada)

$$u_0 = -r_0^2$$

Se  $y_i = 1$  então

$$\underline{x} \in C_i$$



- pelos critérios
- 1 (padrao mais similar a entrada) e
  - 2 (satisfaz a similaridade minima exigida).

Se  $y_i = 1$  então

$$\underline{x} \in C_i$$

pelos critérios

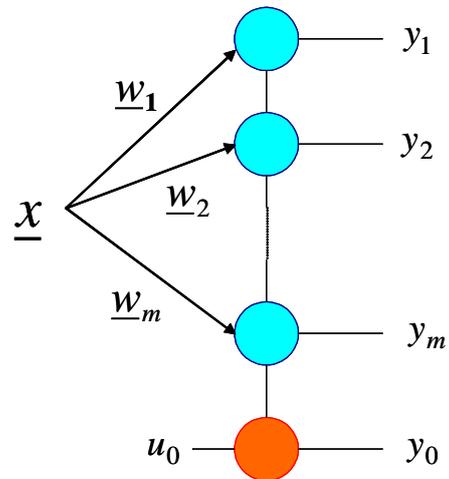
1 (centro de classe mais similar à entrada) e

2 (satisfaz à similaridade mínima)

Se  $y_0 = 1$  então

$$\underline{x} \notin C_i \quad \forall i$$

$\underline{x}$  não satisfaz ao critério 2 para nenhuma classe



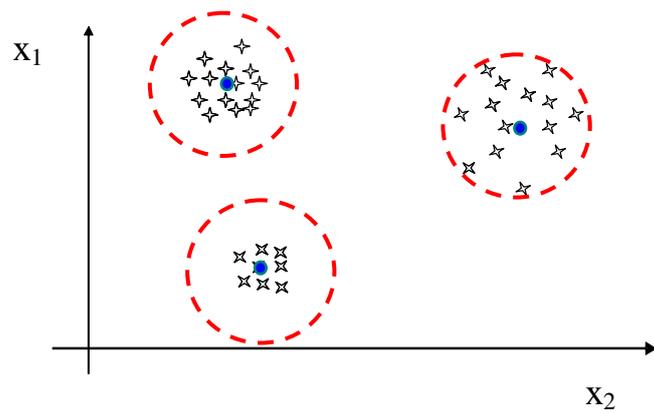
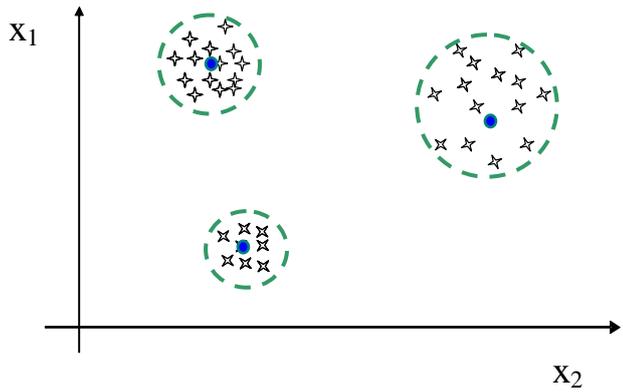
Escolha de  $u_0$

após a construção da camada de Kohonen

determinar o máximo afastamento de qualquer elemento  $\underline{x}$  do centro de sua classe

$$u_0 \approx - \underset{\forall \text{ classe } C_j}{\text{Max}} \underset{\forall \bar{x} \in C_j}{\text{Max}} (-u_j) = \underset{\forall \text{ classe } C_j}{\text{Min}} \underset{\forall \bar{x} \in C_j}{\text{Min}} (u_j)$$

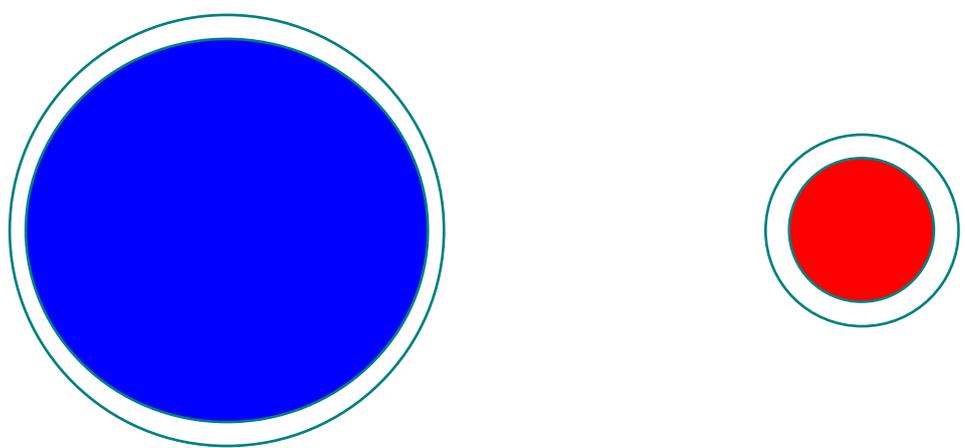
$$u_0 \approx \underset{\forall \bar{x}}{\text{Min}} (u_j)$$



### 5.1 - Classes esféricas com dimensões variáveis



### Neurônio com raio variável



## Raio de vigilância variável por neurônio

### Condição de pertinência à classe $C_i$

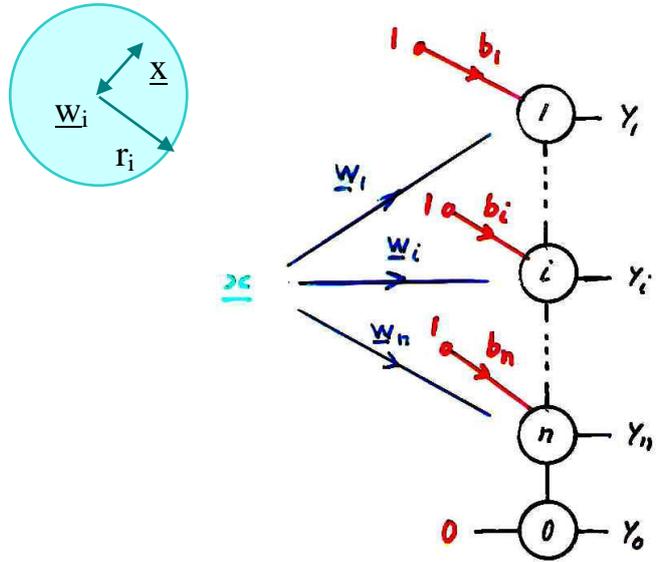
$$|\vec{x} - \vec{w}_i|^2 = d_i^2 < r_i^2$$

$$-d_i^2 > -r_i^2$$

$$-d_i^2 + r_i^2 > 0$$

$$b_i = r_i^2$$

$$u_i = -d_i^2 + b_i = -d_i^2 + r_i^2 > 0$$



### Caso dois neurônios apresentem $u > 0$

a entrada satisfaz a similaridade mínima para os dois padrões

$$u_i > u_j > 0 \quad \underline{x} \in C_i$$

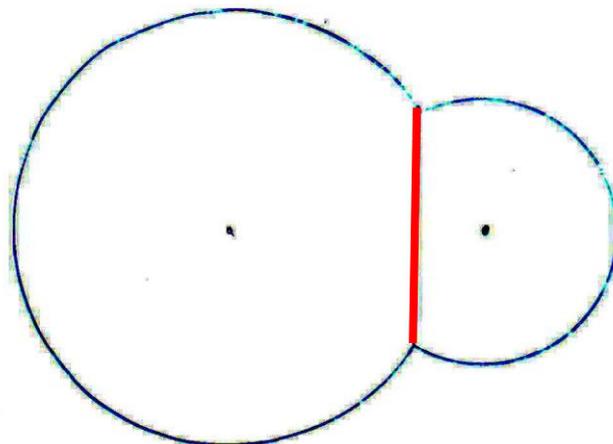
$$-d_i^2 + r_i^2 > -d_j^2 + r_j^2$$

### Separador “interno”

plano que contém

a circunferência

intersecção das esferas



## Camada de Kohonen com similaridade mínima por classe

1 - Treinar a camada de Kohonen (treinamento supervisionado)

2 – Calcular o raio de cada classe como o máximo afastamento dos elementos aos centros das respectivas classes [  $r_i = \text{Max} (-u_i)$  ]

3 – Inserir os bias  $b_i = - r_i$  e o neurônio de controle  $y_0$

