

Camada de Kohonen

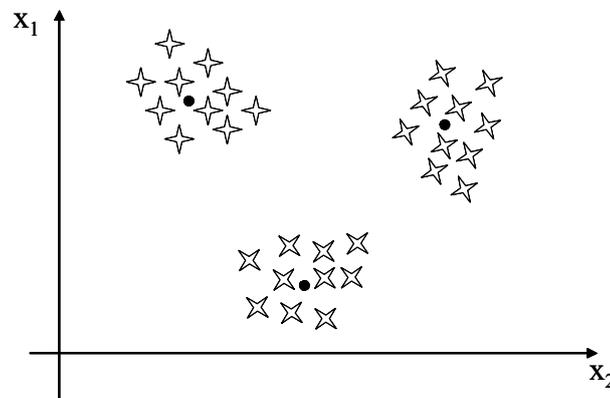
Classificação por Similaridade

Classes agrupam elementos “similares” entre si



1- Critério de pertinência:

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

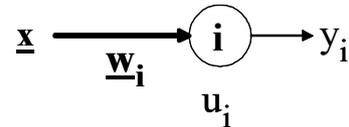
2 - Redes Neurais

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

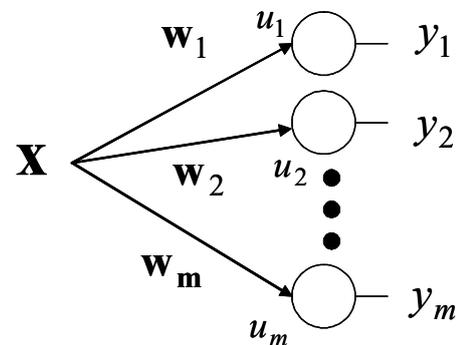
3- Camada de Kohonen

Template Matching

$$u_i = -d_i^2$$

maior u_i =
menor distância =
maior similaridade

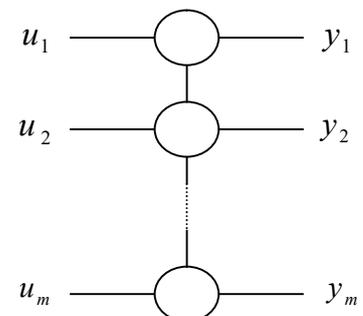
u_i é uma medida de similaridade entre \underline{x} e \underline{w}_i



Winner-takes-all

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

$$y_i = 0 \text{ caso contrário}$$



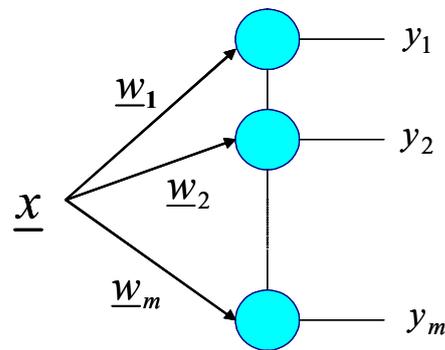
Camada de Kohonen

[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.

Objetivo:

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 anteriormente, após pequena álgebra

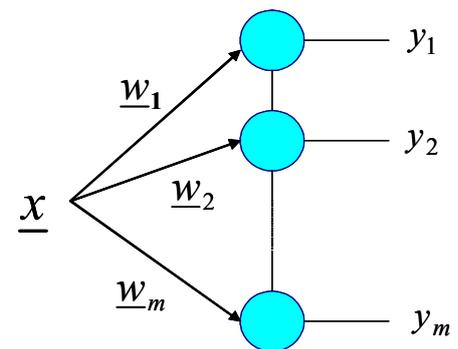
$$\frac{\partial}{\partial w_{jk}} \sum_{\forall \vec{x} \in C_j} (x_k - w_{jk})^2 = 0 \quad \Rightarrow \quad w_{jk} = m_{jk} \quad \Rightarrow \quad \vec{w}_j = \vec{m}_j$$

4.1 - Treinamento em Batelada (\underline{w}_j ótimo)

**todas as entradas (e suas classes) 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



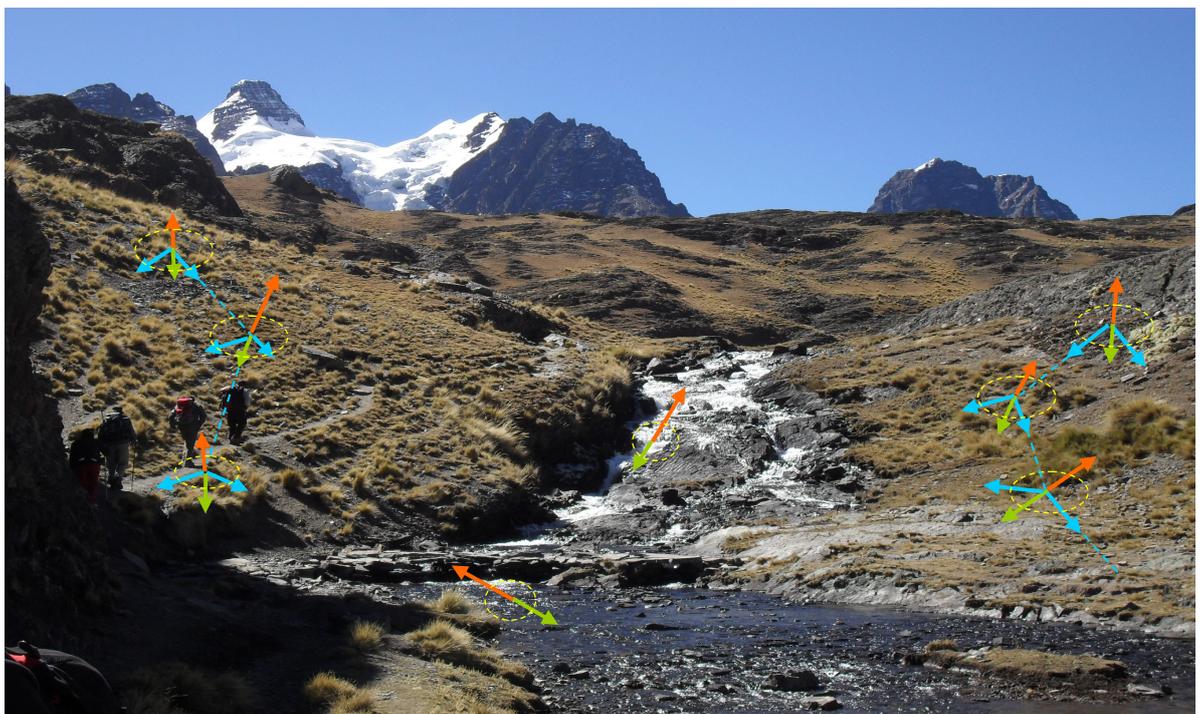
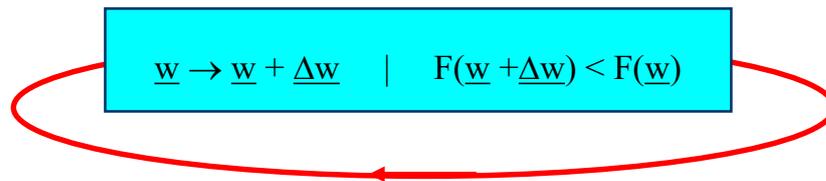
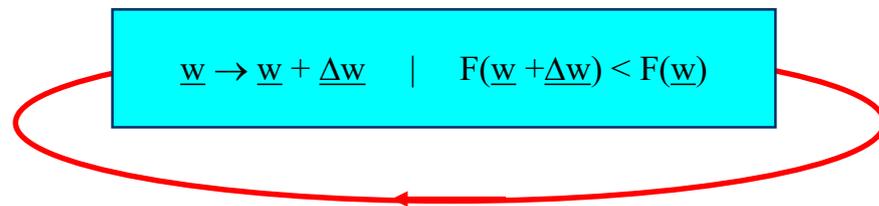
4.2 - Treinamento em “regra delta”

cada entrada é apresentada e a rede é treinada passo a passo, *on line*

dados recebidos ao longo do tempo / sistema variante no tempo

Otimização: Métodos Numéricos Recursivos

Descida continuada

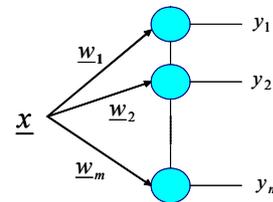


Que direção garante o decréscimo em F_{in} ?

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

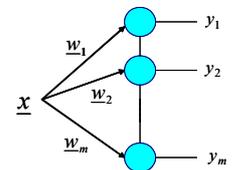
$$\Delta w_{jk} = -\alpha \frac{\partial F_{in}}{\partial w_{jk}} = -\alpha \frac{\partial F_j}{\partial w_{jk}} = -\alpha \frac{\partial}{\partial w_{jk}} \sum_{\forall \vec{x} \in C_j} |\vec{x} - \vec{w}_j|^2 = -\alpha \frac{\partial}{\partial w_{jk}} \sum_{\forall \vec{x} \in C_j} \sum_{\forall l} (x_l - w_{jl})^2 =$$

$$\Delta w_{jk} = \begin{cases} = -\alpha \frac{\partial}{\partial w_{jk}} \sum_{\forall \vec{x} \in C_j} (x_k - w_{jk})^2 = \alpha E_{\forall \vec{x} \in C_j} (x_k - w_{jk}) & \text{para } \vec{x} \in C_j \\ = 0 & \text{para } \vec{x} \notin C_j \end{cases}$$



Batelada (vários passos) - após aplicar todos as entradas \underline{x}

$$\Delta w_{jk} = \begin{cases} = \alpha E_{\forall \vec{x} \in C_j} (x_k - w_{jk}) & \text{para } \vec{x} \in C_j \\ = 0 & \text{para } \vec{x} \notin C_j \end{cases}$$



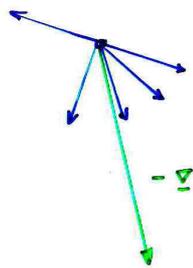
Regra Delta – após aplicar cada entrada \underline{x}

$$\Delta w_{jk} = \begin{cases} = \alpha (x_k - w_{jk}) & \text{para } \vec{x} \in C_j \\ = 0 & \text{para } \vec{x} \notin C_j \end{cases}$$

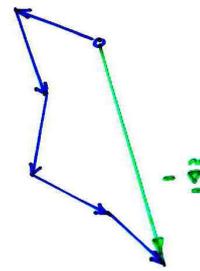
Regra Delta

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

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



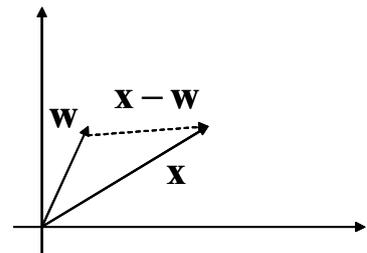
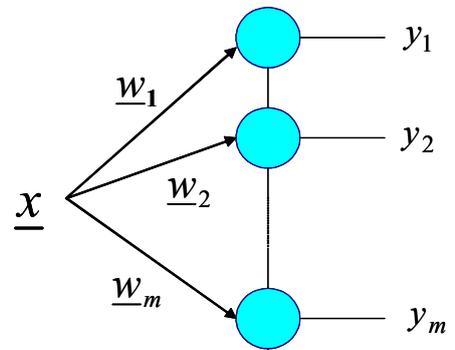
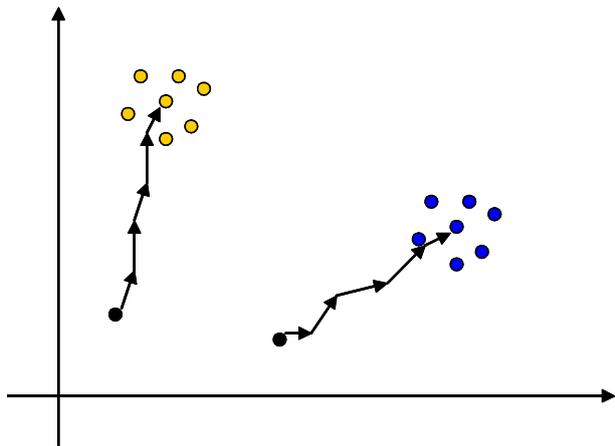
batelada



regra delta

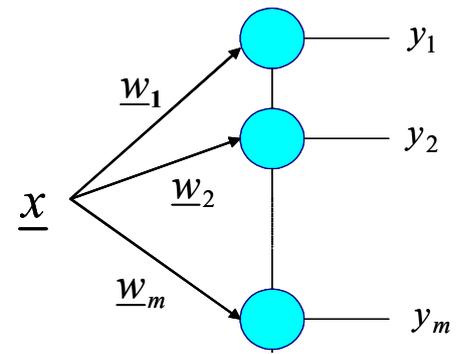
4.1 - Treinamento da Camada de Kohonen por regra delta

Treinamento Supervisionado



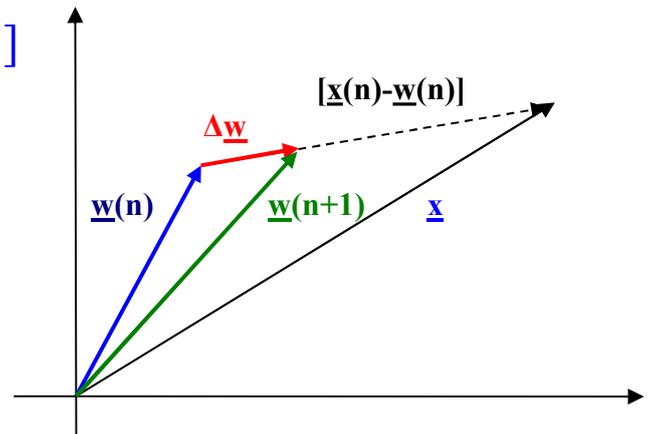
Treinamento Supervisionado

$$\underline{x}(n) \in C_i \Rightarrow \begin{aligned} y_i &= 1 \\ y_j &= 0 \quad \forall j \neq i \end{aligned}$$



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

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



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

Evolução do \underline{w} de uma classe

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

$\underline{x}(n) \in C_i$

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

$$\underline{w}(0)$$

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

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

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

⋮

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

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 !

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

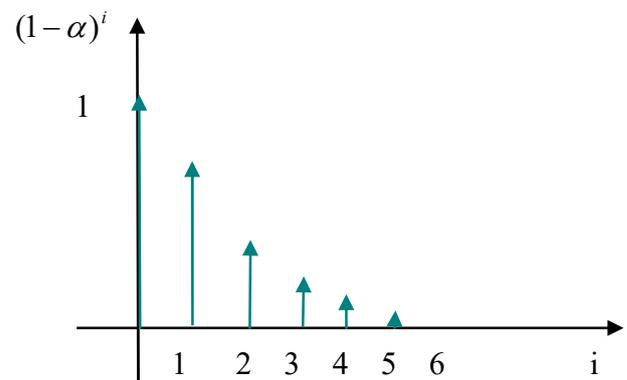
instante atual = n

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

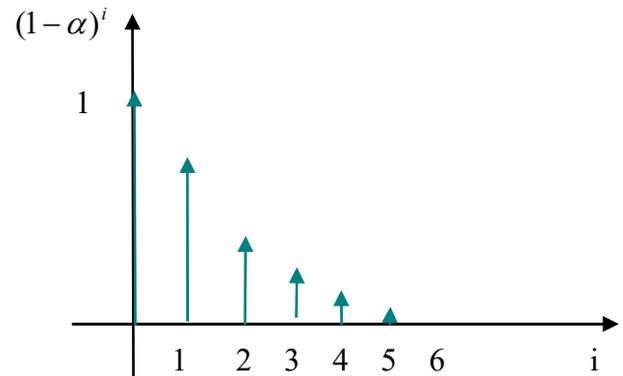
$$\text{Ponderador} = (1-\alpha)^0 = 1$$

Último atraso à ser considerado:

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

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

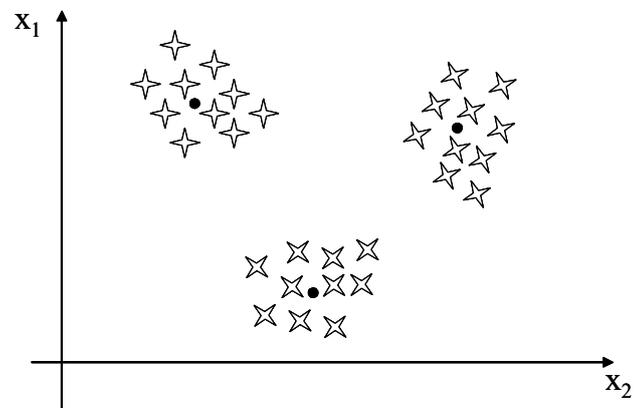
$$i \approx \frac{4}{\alpha}$$



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 α mais precisamente o baricentro será determinado. Mas, como esperado, maior o tempo (número de passos) necessário para calculá-lo

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

Exemplo:

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

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

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

Mas se:

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

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

4.5 Considerações:

4.5.1. Fim do treinamento ?



$$E[\Delta \bar{w}] = \vec{0}$$



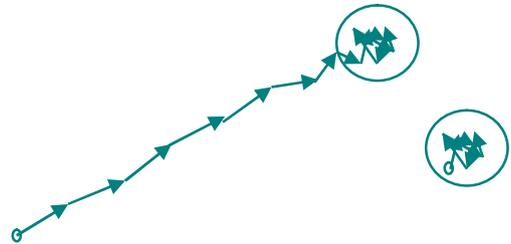
$$\Delta \bar{w} = \alpha (\vec{x} - \bar{w})$$

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

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

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 α 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

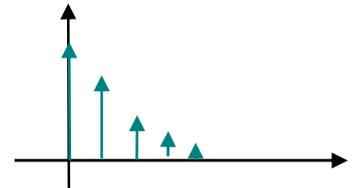
reduzir ao longo do tempo:

compromisso entre um ajuste rápido e erro pequeno

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

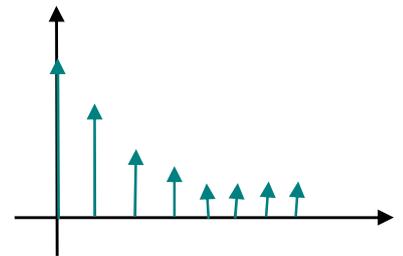
$$\alpha(0) = \alpha_0 \quad \alpha(n+1) = k \alpha(n) \quad \text{onde} \quad k = 1 - \frac{1}{N_0}$$

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



alternativa para não acabar o treinamento:

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

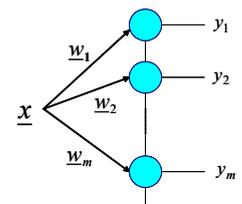


e os vetores sinapse pouco ativados (classes pouco populosas) ?

usar α 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 \underline{w}_j foi treinada.

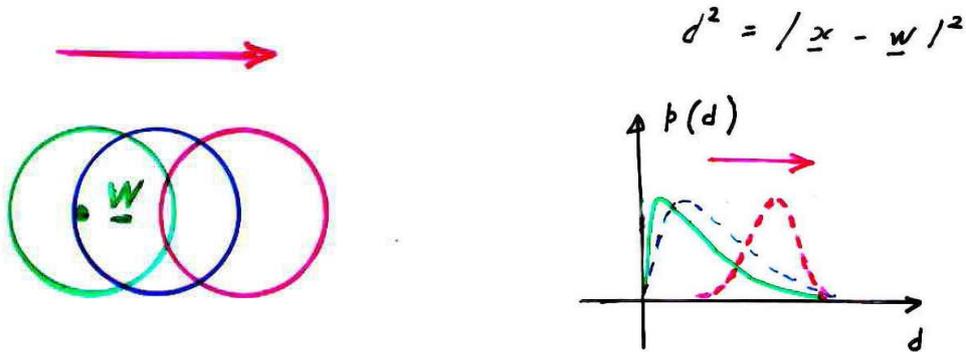


Obs: Passo variável é menos importante se os valores iniciais das sinapses são elementos do cluster



4.5.4 Treinamento dinâmico, adaptativo

Uma classe varia de posição no tempo. Como saber ?

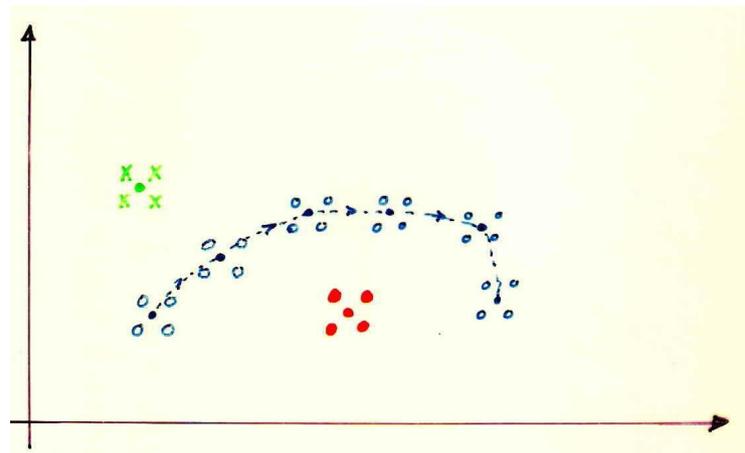
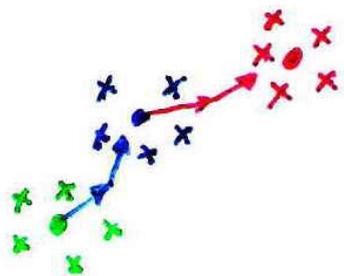


O valor médio de $u_i = -d_i^2$ diminui

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

a distribuição de d é χ_k^2

Como corrigir ? Ligar o treinamento até que d_i^2 volte a seus valores normais.



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

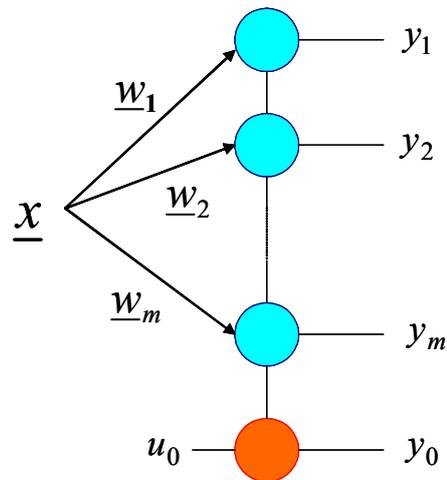
5 - E o segundo critério (similaridade mínima) ?

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

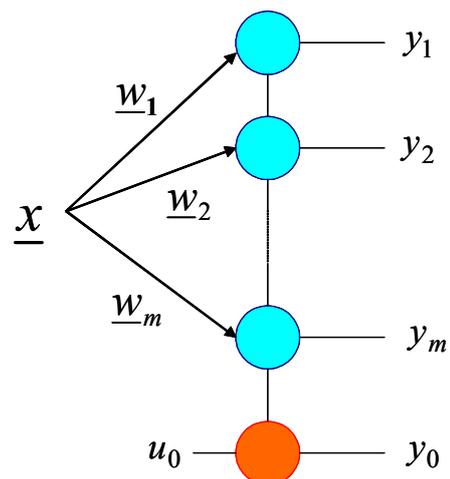
2 (satisfaz a similaridade minima exigida).

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

1 – Critério absoluto

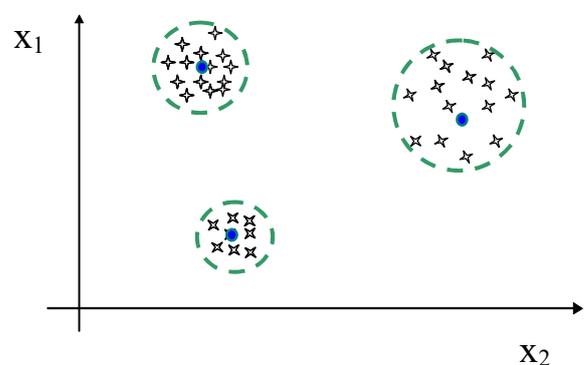
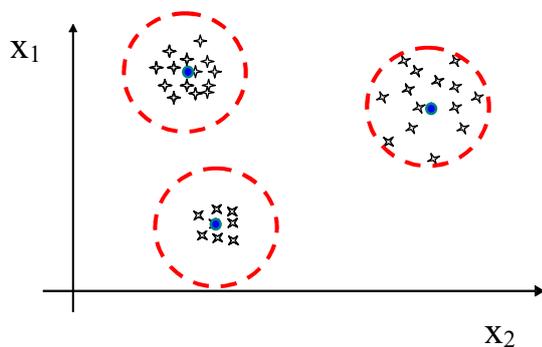
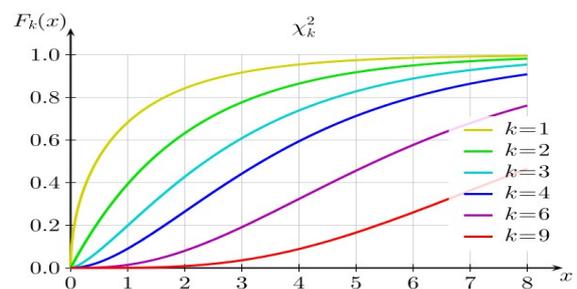
determinar o máximo afastamento de qualquer elemento \underline{x} do centro de sua classe

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

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

2 – Critério estatístico

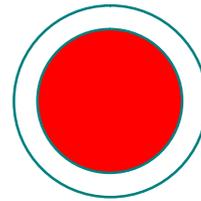
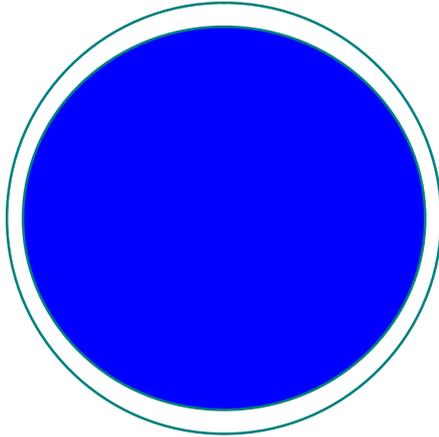
a distribuição de d é χ_k^2



5.1 - Classes esféricas com dimensões variáveis



Neurônio com raio variável



Raio de vigilância variável por neurônio

Condição de pertinência à classe C_i

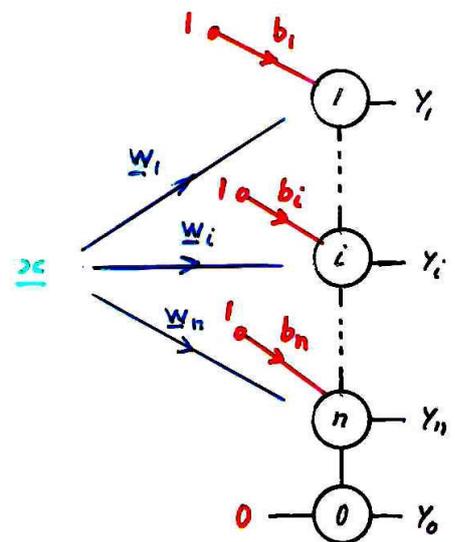
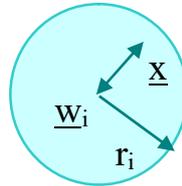
$$|\vec{x} - \vec{w}_i|^2 = d_i^2 < r_i^2$$

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

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

$$b_i = r_i^2$$

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



Caso dois neurônios apresentem $u > 0$

a entrada satisfaz a similaridade mínima para os dois padrões

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

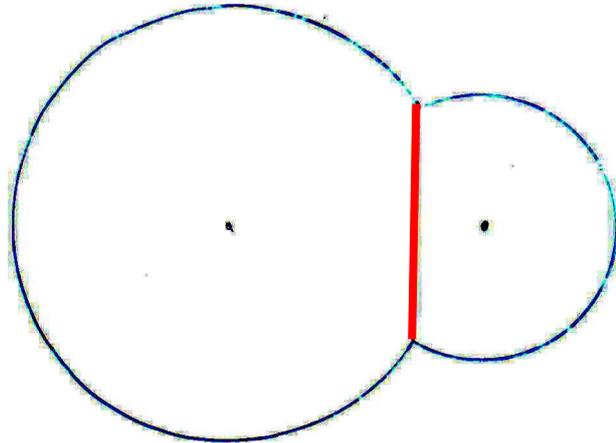
$-d_i^2 + r_i^2 > -d_j^2 + r_j^2$

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

