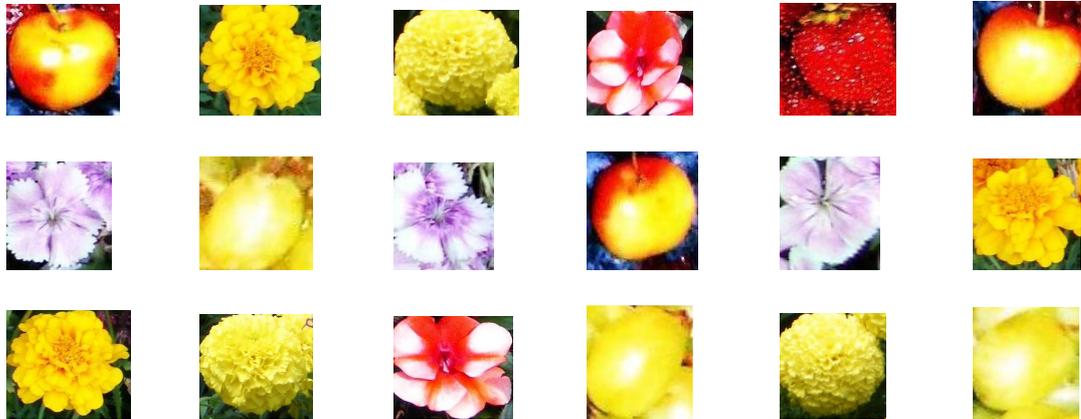


Camada de Kohonen

Classificação por Similaridade

Classes agrupam elementos “similares” entre si



versus classificadores arbitrários

1 - Classificação por Similaridade – Critérios de pertinência

objetos físicos $O_1 O_2$ \longrightarrow objetos matemáticos $\underline{x}_1 \underline{x}_2$

similaridade física \longrightarrow similaridade matemática

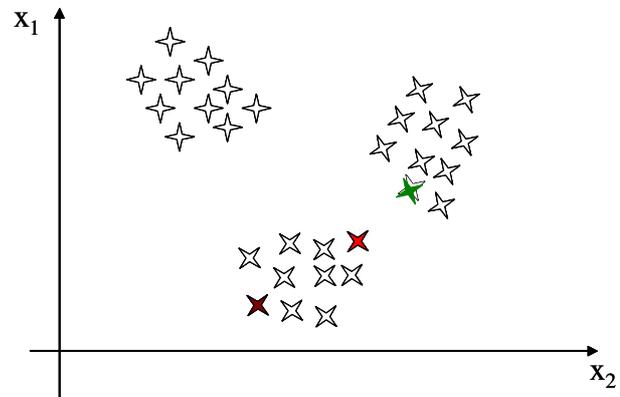
$$O_1 \approx O_2$$

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

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.

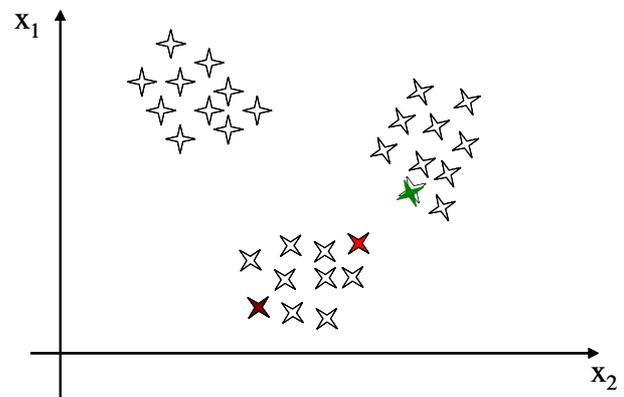
Considere os k elementos mais próximos da entrada que se pretende classificar. A entrada pertence à classe à qual pertencem a maioria dos k vizinhos.

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 prático

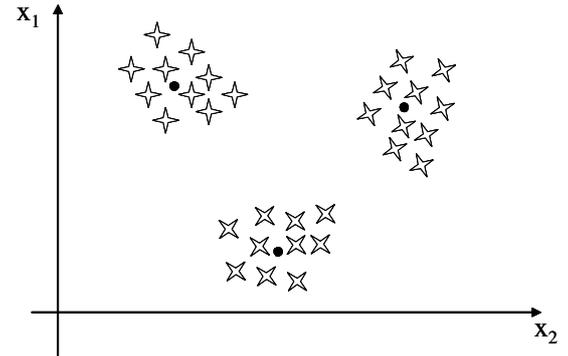
a descrição de uma classe exige a utilização de todos seus elementos (domínio da classe)



Padrão de classe:

Padrão \underline{w}_i da classe C_i

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



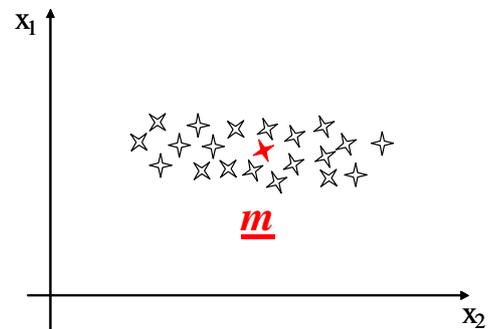
a **média** ou **baricentro** é escolhido como padrão da classe porque minimiza a **dissimilaridade** ou **erro de representação** da classe

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

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



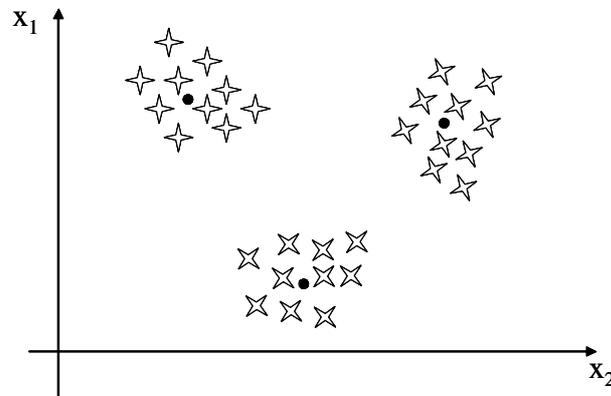
após pequena álgebra

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

o padrão que minimiza a dispersão intra classe (ou erro de representação) de uma classe é o seu baricentro

1.1 - Critério de pertinência 1 :

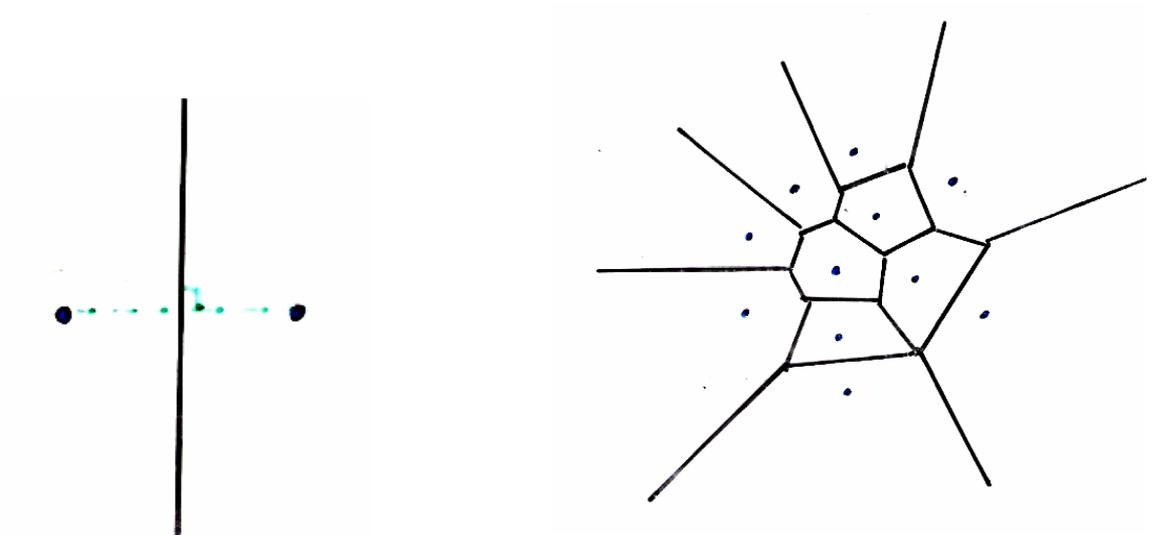
Padrão mais similar à entrada



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

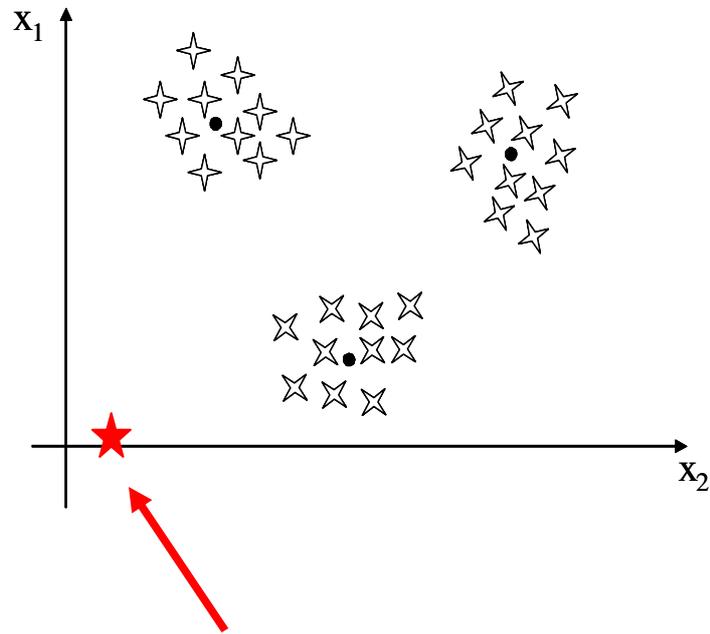
porque minimiza a dispersão intra-classes total

Separadores para o critério 1

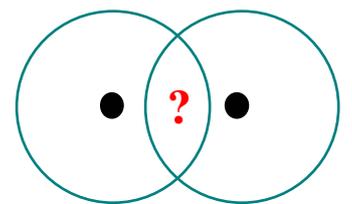
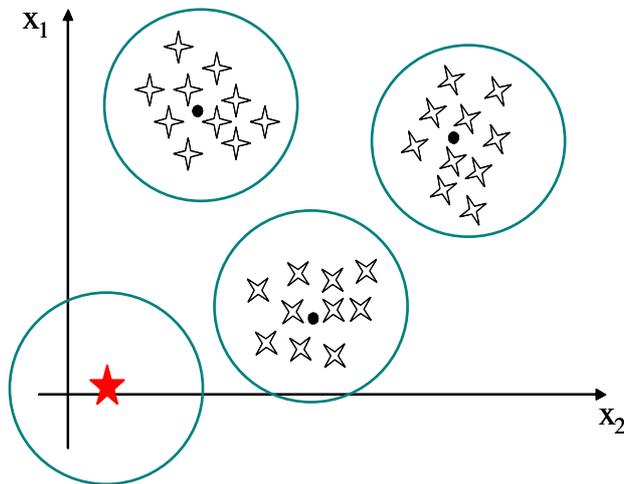


Tecelagem (diagrama) de Voronoi

Problema:



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



Problema

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

sse

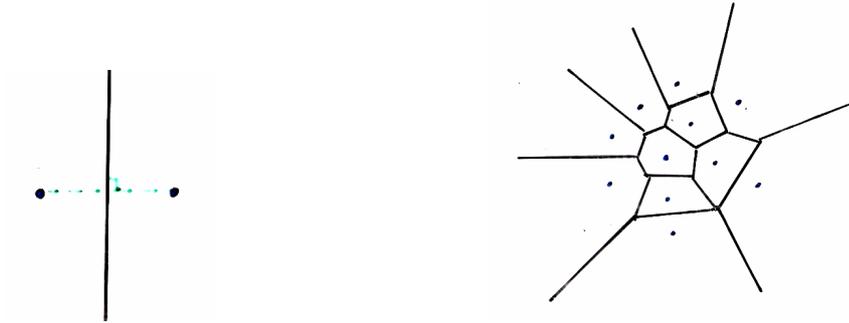
$$|\underline{x} - \underline{w}_i|^2 < r_0^2$$

eq (2)

r_0 – raio de similaridade

Separadores

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

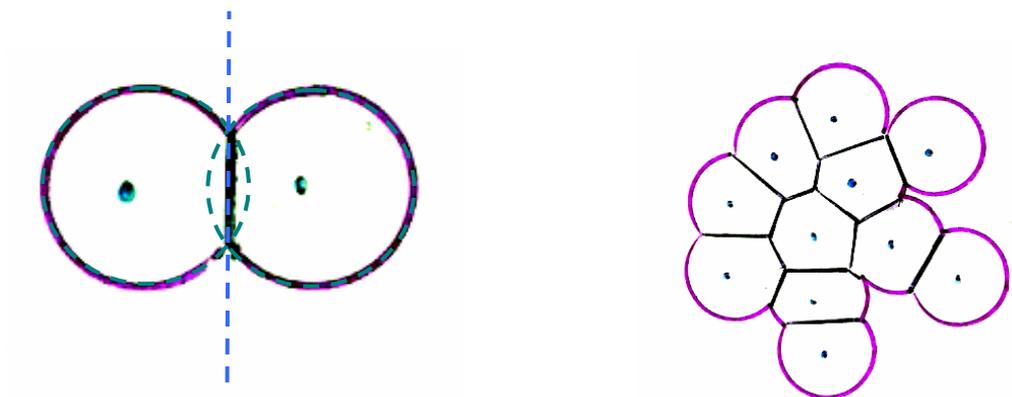


Critério 2 - mínima similaridade exigida Eq (2)



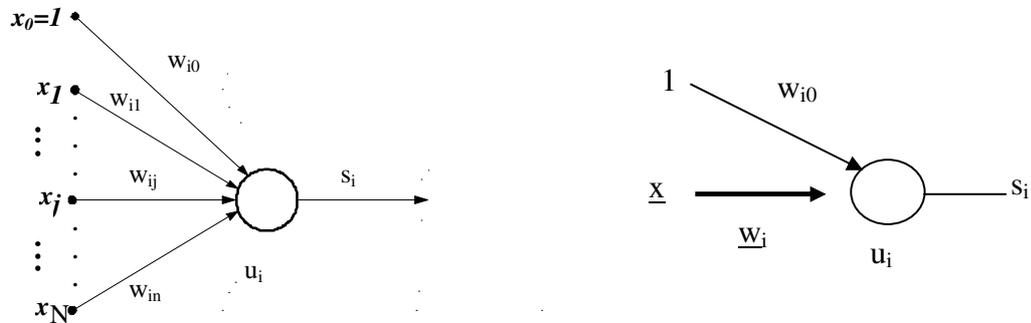
1.3 - Critérios 1 e 2, Eq(1) + Eq(2)

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

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

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

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

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

logo

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

e u é um **comparador de similaridade**

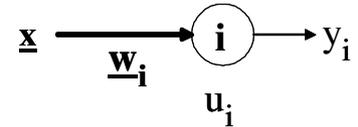
y_i – depende dos outros neurônios

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

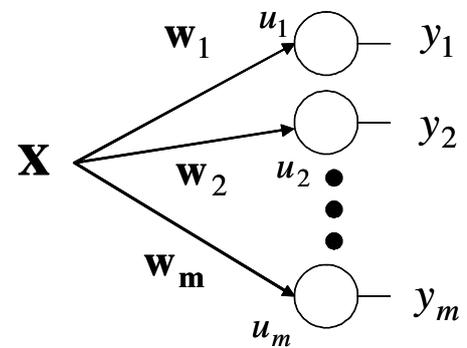
3- Camada de Kohonen

Template Matching

$$u_i = -d_i^2$$

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

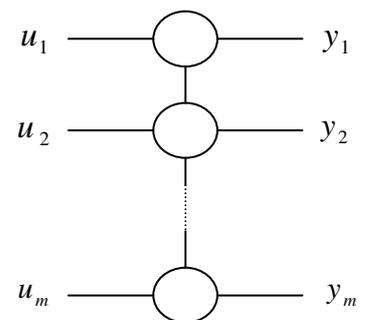
u_i é uma medida de similaridade entre \underline{x} e \underline{w}_i



Winner-takes-all

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

$$y_i = 0 \text{ caso contrário}$$



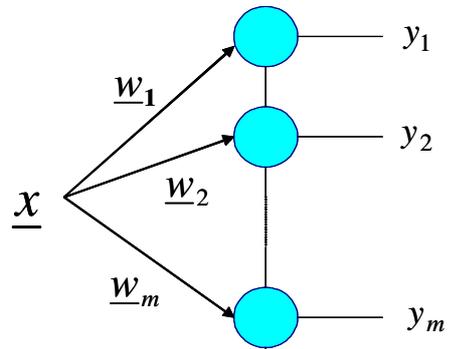
Camada de Kohonen

[versão unidimensional, simplificada ($D=0$)]

Classe C_i

Padrão \underline{w}_i

Indicador y_i

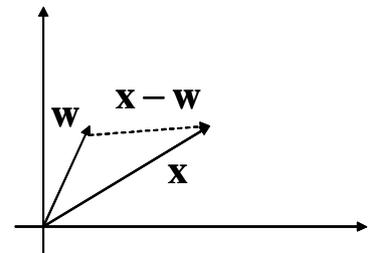
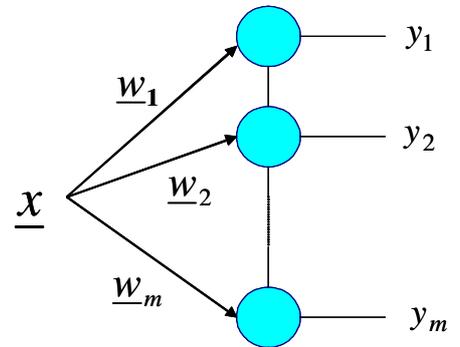
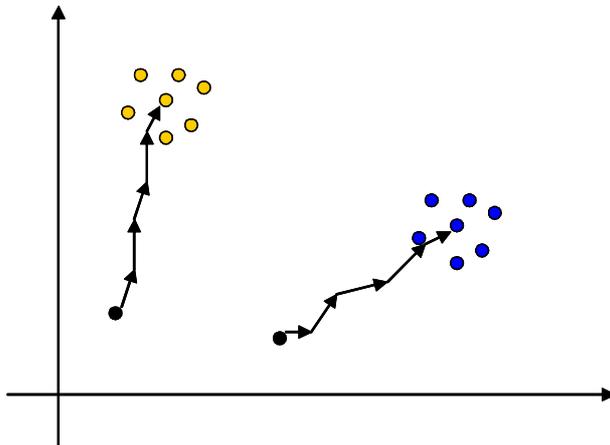


Se $y_i = 1$ então $\underline{x} \in C_i$

pele critério 1 (padrão mais similar a entrada).

4 - Treinamento da Camada de Kohonen

Treinamento Supervisionado



Treinamento Supervisionado

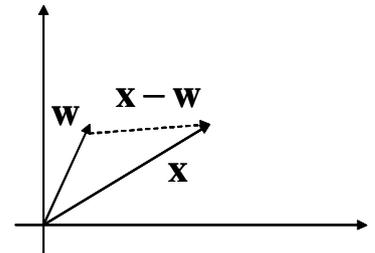
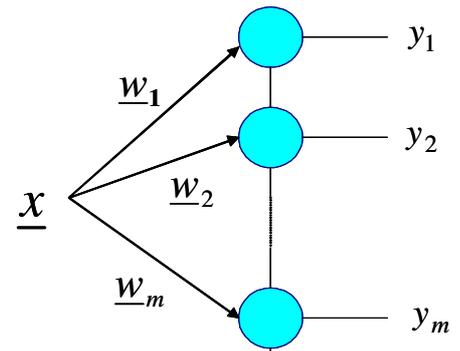
$$\underline{x}(n) \in C_i \quad \Rightarrow \quad y_i = 1$$

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

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

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

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



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

Evolução do \underline{w} de uma classe

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

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

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

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

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

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

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

⋮

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

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 !

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

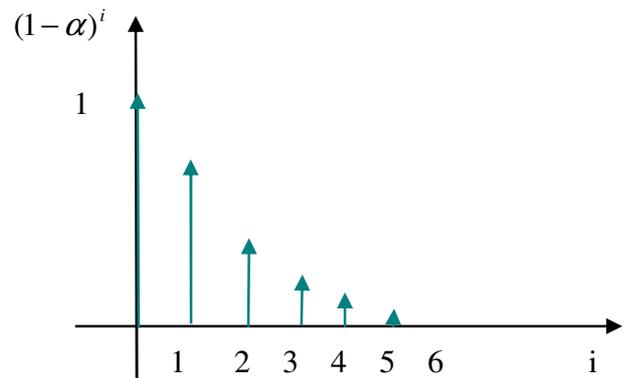
instante atual = n

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

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

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

4.1 – Tempo de medida

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

Atraso zero

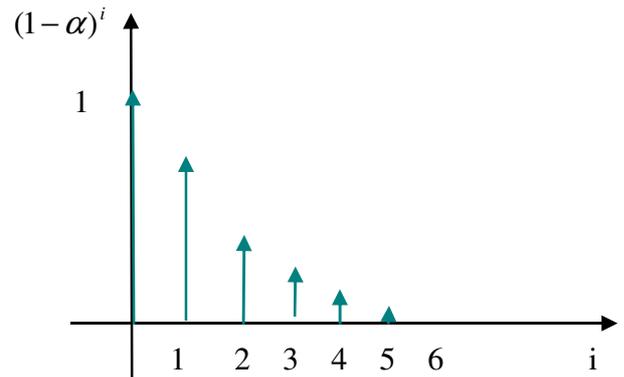
$$\text{Ponderador} = (1-\alpha)^0 = 1$$

Último atraso à ser considerado:

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

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

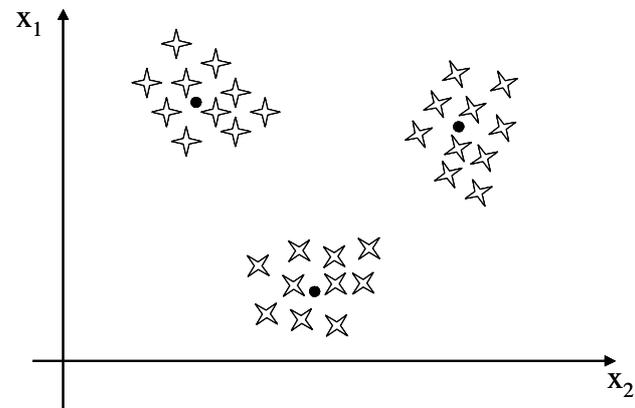
$$i \approx \frac{4}{\alpha}$$



4.2 - 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

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

4.3 – 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:

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

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

$$\alpha = \frac{2\sigma_{w_j}^2}{\sigma_{x_j}^2} = .08$$

$$\# \text{ passos} = \frac{4}{\alpha} = 50 \text{ passos}$$

Mas se:

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

$$\alpha = \frac{2\sigma_{w_j}^2}{\sigma_{x_j}^2} = .0008$$

$$\# \text{ passos} = \frac{4}{\alpha} = 5.000 \text{ passos !!}$$

4.4 Considerações:

4.4.1. Fim do treinamento ?



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



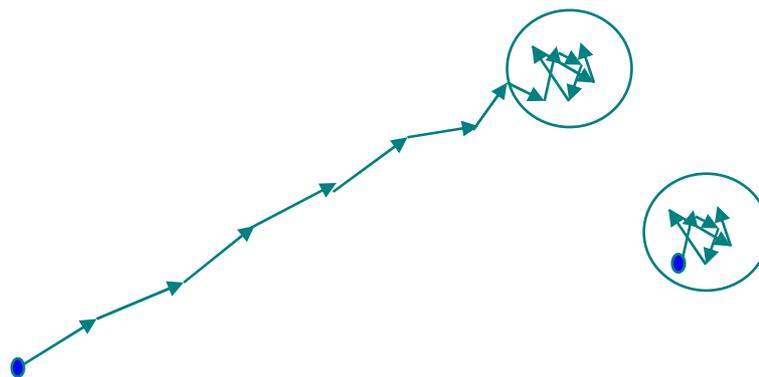
$$\Delta \bar{w} = \alpha (\bar{x} - \bar{w})$$

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

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

4.4.2. 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.



4.4.3. Passo de treinamento

reduzir ao longo do tempo $\alpha(n) = \alpha_0 e^{-\frac{n}{N_0}}$

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

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

e as sinapses pouco ativadas (classes pouco populosas) ?

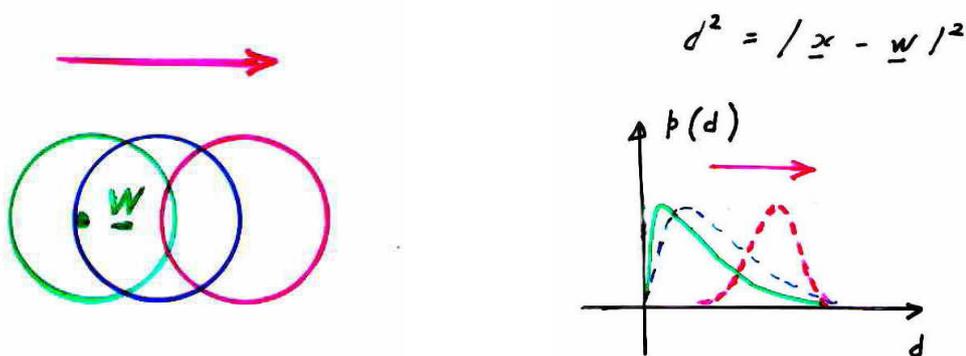
usar α diferenciado por sinapse

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

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

4.4.4 Treinamento dinâmico, adaptativo

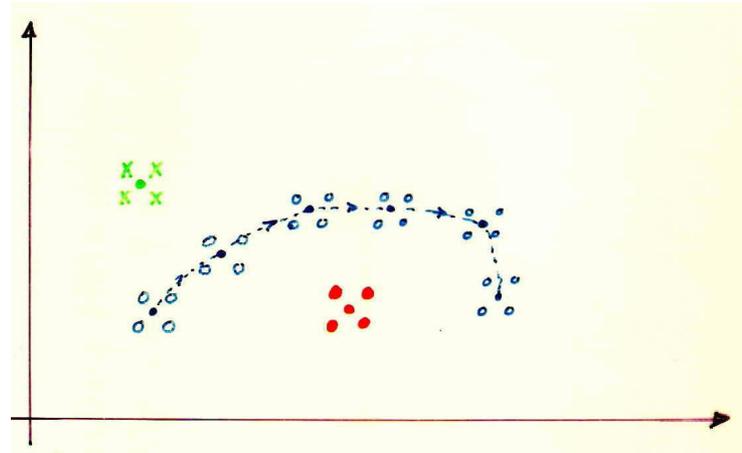
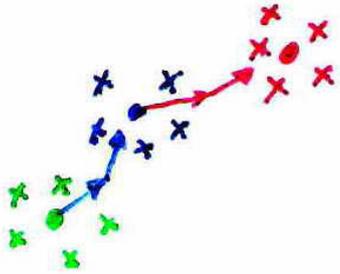
Uma classe varia de posição no tempo. Como saber ?



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

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

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

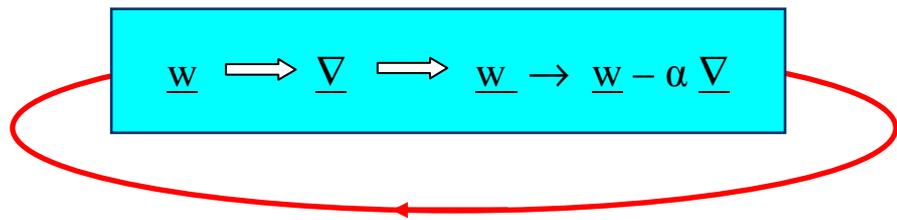


Se a variação do baricentro for lenta e o treinamento estiver ligado o padrão segue o baricentro

4.5 -O Treinamento de Kohonen como um processo Gradiente Descendente

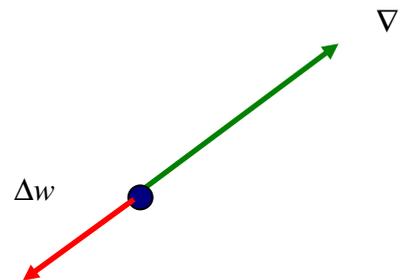
Método do Gradiente Descendente ou

Método da Descida pelo Gradiente



$$\Delta \vec{w} = -\alpha \vec{\nabla}$$

$$\Delta w_i = -\alpha \frac{\partial F}{\partial w_i}$$



Algoritmo**Até que o critério de parada seja satisfeito**Para cada variável w_i

$$\text{Calcular } \frac{\partial F}{\partial w_i}$$

$$\text{Fazer } w_{i \text{ novo}} = w_{i \text{ antigo}} - \alpha \frac{\partial F}{\partial w_i}$$

Fim (natural) do processo

$$\underline{w} = \underline{w}_0$$

$$E(\Delta \underline{w}) = \underline{0} \quad \Rightarrow \quad \nabla(\underline{w}_0) = \underline{0}$$

Em nosso caso, a função à minimizar é**Dispersão total intra classe para todas as classes:**

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

$$F_j = \sum_{\forall \vec{x} \in C_j} |\vec{x} - \vec{m}_j|^2 = \sum_{\forall \vec{x} \in C_j} \sum_{l=1}^n (x_l - m_{jl})^2$$

$$\frac{\partial F_{in}}{\partial m_{jl}} = -2 \sum_{\forall \vec{x} \in C_j} (x_l - m_{jl})$$

Solução analítica

$$\nabla F_{in} = \left[\frac{\partial F_{in}}{\partial m_{jl}} = -2 \sum_{\forall \vec{x} \in C_j} (x_l - m_{jl}) \right] = 0$$

$$\sum_{\forall \vec{x} \in C_j} (x_l - m_{jl}) = 0 \quad \sum_{\forall \vec{x} \in C_j} x_l - n_j m_{jl} = 0 \quad m_{jl} = \frac{1}{n_j} \sum_{\forall \vec{x} \in C_j} x_l = \frac{E}{\forall \vec{x} \in C_j} x_l$$

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

Solução Numérica Recursiva

$$\Delta m_{ij} = -\alpha \frac{\partial F_{in}}{\partial m_{jl}} = 2\alpha \sum_{\forall \vec{x} \in C_j} (x_l - m_{jl})$$

$$\Delta \vec{m}_j = 2\alpha \sum_{\forall \vec{x} \in C_j} (\vec{x} - \vec{m}_j)$$

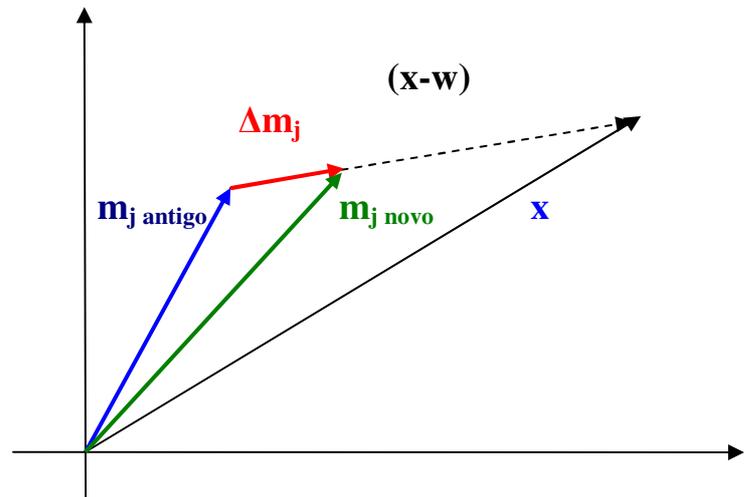
Este é um processo por batelada.

Alternativa: Calcular o acréscimo devido à cada entrada e aplicar imediatamente (processo por “regra delta”). Plástico e reduz a quantidade de memória necessária.

Processo por “regra delta”

$$\Delta m_{ij} = -\alpha \frac{\partial F_{in}}{\partial m_{jl}} = 2\alpha (x_l - m_{jl}) \Big|_{\forall \vec{x} \in C_j}$$

$$\Delta \vec{m}_j = 2\alpha (\vec{x} - \vec{m}_j) \Big|_{\forall \vec{x} \in C_j}$$



Fim do processo recursivo:

$$E(\Delta \vec{m}_j) = 2\alpha E(\vec{x} - \vec{m}_j) \Big|_{\forall \vec{x} \in C_j} = \vec{0}$$

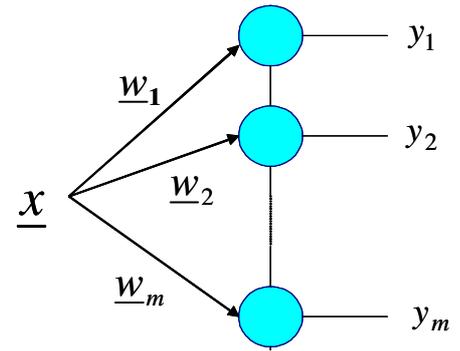
$$\vec{m}_j = \frac{1}{n_j} \sum_{\forall \vec{x} \in C_j} \vec{x} = E_{\forall \vec{x} \in C_j} \vec{x}$$

5 – LVQ – Learning Vector Quantization

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

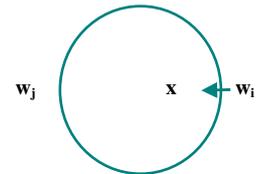
e

neurônio vencedor y_j



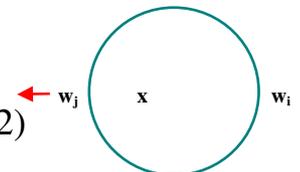
$$j = i \quad \text{ou} \quad j \neq i$$

$$\text{se } j = i \quad \vec{w}_j(n+1) = \vec{w}_j(n) + \alpha [\vec{x}(n) - \vec{w}_j(n)] \quad (\text{eq 1})$$



se a entrada pertence à classe vencedora, aproxima a sinapse vencedora da entrada e conseqüentemente do centro da classe

$$\text{se } j \neq i \quad \vec{w}_j(n+1) = \vec{w}_j(n) - \alpha [\vec{x}(n) - \vec{w}_j(n)] \quad (\text{eq 2})$$



se a entrada não pertence à classe vencedora, afasta a sinapse vencedora da entrada e, conseqüentemente, do centro w_j da classe a que a entrada pertence. Isto aumenta a chance da classe certa vencer em uma próxima competição.

Note que se as sinapses tiverem sido inicializadas com um elemento da classe a probabilidade do segundo caso ($j \neq i$) ocorrer é muito pequena. Isto também ocorre após a fase inicial do treinamento, quando as sinapses já foram arrastadas para dentro de suas respectivas classes.

Obs:

1 - O passo de aprendizagem deve decrescer com o tempo

$$\alpha = \alpha(n) = \alpha_0 \exp\left(-\frac{n}{\tau_\alpha}\right) \quad \alpha_0 \approx 0,1 \quad \tau_\alpha \approx 1000$$

2 - O equilíbrio é atingido quando $E[\Delta\vec{w}_i] = \vec{0}$, o que ocorre quando

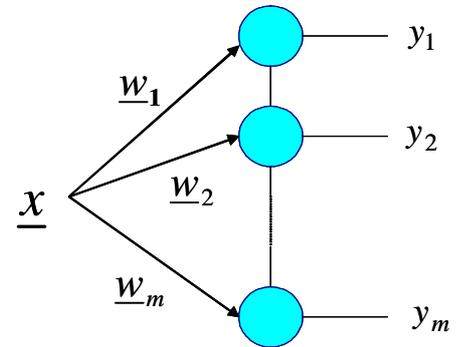
$$\vec{w}_i = \vec{m}_i = E_{\forall \vec{x} \in C_i}(\vec{x})$$

3 – No fim do treinamento todas as entradas estarão na classe correta e apenas a eq. 1 será utilizada. Em consequência este processo corresponde a minimizar a função objetivo dispersão intra-classe total

$$F = \sum_{\forall C_i} F(\vec{w}_i) \quad F(\vec{w}_i) = \sum_{\forall \vec{x} \in C_i} |\vec{x} - \vec{w}_i|^2$$

5a – Um método alternativo (???)**LVQ – Learning Vector Quantization**

$\vec{x}(n) \in C_i$ aproxima o padrão vencedor de $\vec{x}(n)$ e afasta os demais padrões de $\vec{x}(n)$



$$\vec{x}(n) \in C_i \quad \begin{cases} y_i = 1 \\ y_j = 0 \quad \forall j \neq i \end{cases}$$

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

$$\vec{w}_j(n+1) = \vec{w}_j(n) - \alpha[\vec{x}(n) - \vec{w}_j(n)] \quad \forall j \neq i$$

ou, usando uma fórmula única

$$\vec{x}(n) \in C_i \quad \begin{cases} y_i = 1 \\ y_j = 0 \quad \forall j \neq i \end{cases}$$

$$\vec{w}_j(n+1) = \vec{w}_j(n) + (2y_i - 1)\alpha [\vec{x}(n) - \vec{w}_j(n)] \quad \forall j$$

O passo de aprendizagem deve decrescer com o tempo

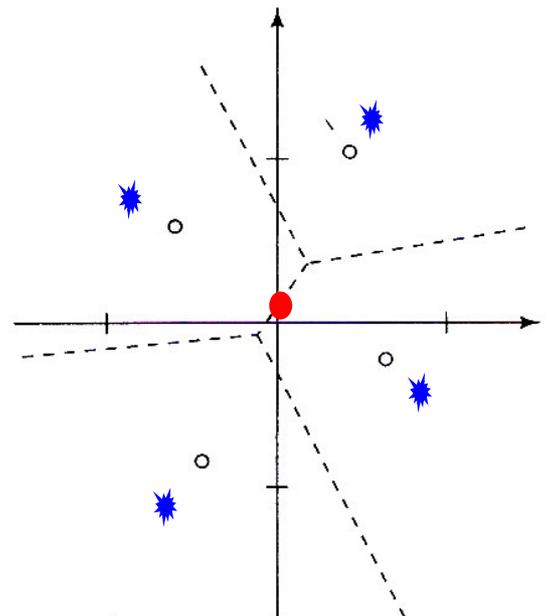
$$\alpha = \alpha(n) = \alpha_0 \exp\left(-\frac{n}{\tau_\alpha}\right) \quad \alpha_0 = 0,1 \quad \tau_\alpha = 1000$$

Este processo melhora as fronteiras de classificação (Haykin).

O equilíbrio é atingido quando $E[\Delta\vec{w}_i] = \vec{0}$, o que ocorre quando

$$\begin{aligned} \vec{w}_i &= p_i \mathbf{E}_{\forall \vec{x} \in C_i}(\vec{x}) - (1-p_i) \mathbf{E}_{\forall \vec{x} \notin C_i}(\vec{x}) = \\ &= 2p_i \mathbf{E}_{\forall \vec{x} \in C_i}(\vec{x}) - \mathbf{E}_{\forall \vec{x}}(\vec{x}) = \\ &= 2p_i \vec{m}_i - \vec{m} \end{aligned}$$

onde p_i é a probabilidade de uma entrada \vec{x} pertencer à classe C_i , \vec{m}_i é o baricentro de C_i e \vec{m} é o baricentro de todas as entradas, normalmente $\vec{m} = \vec{0}$. O efeito é deslocar os padrões das classes (★) para longe do centro de todas as entradas (●) mas também dos baricentros (○) das mesmas.



Este processo corresponde a otimizar a função objetivo

$$F(\vec{w}_i) = \sum_{\forall \vec{x} \in C_i} |\vec{x} - \vec{w}_i|^2 - \sum_{\forall \vec{x} \notin C_i} |\vec{x} - \vec{w}_i|^2$$

isto é, minimizar a dispersão intra classe de C_i e maximizar a distância de w_i aos elementos das outras classes.

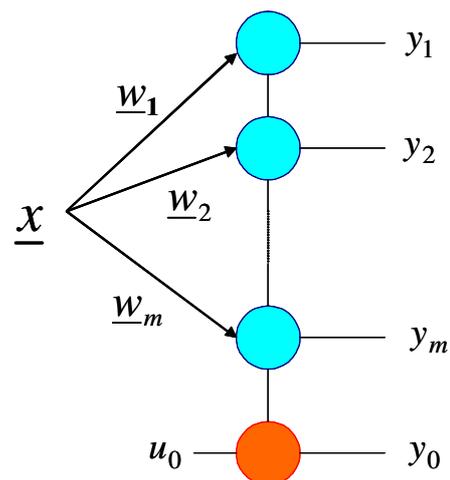
6 - E o segundo critério (similaridade mínima) ?

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

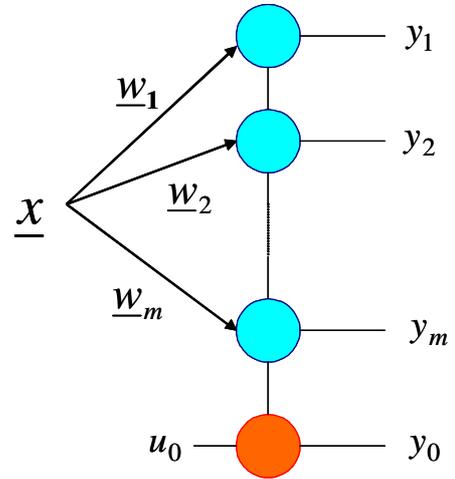
2 (satisfaz a similaridade minima exigida).

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

7 – HVQ – Hierarchical Vector Quantization

