Introdução às Redes Neurais Artificiais

Competitive Learning Networks

Prof. João Marcos Meirelles da Silva

http://www.professores.uff.br/jmarcos

Departamento de Engenharia de Telecomunicações

Escola de Engenharia

Universidade Federal Fluminense

Créditos autorais

Este curso e estes slides são parcialmente adaptados da bibliografia citada e das aulas do professor Luiz Pereira Calôba - COPPE/UFRJ

www.lps.ufrj.br/∼caloba

Sumário

- Introdução
- Considerações Gerais
- Classificação por Similaridade
- Critérios de Pertinência
- Neurônio Medidor de Similaridade
- Arquitetura da Rede
- Treinamento
- Considerações Sobre o Treinamento
- *Template Matching*

Introdução

Até o presente momento, estudamos algumas redes as quais poderiam classificar um determinado padrão como sendo pertencente a mais de uma classe.

Em circunstâncias normais, um padrão deve pertencer a uma única classe.

Para garantirmos isso, devemos incluir uma estrutura adicional na rede de forma que apenas um único neurônio dispare de forma a sinalizar a classe mais provável. O mecanismo pelo qual isso é conseguido chama-se *competição*.

Introdução

A partir deste momento, estudaremos algumas redes que aprendem baseadas no mecanismo de competição. A forma mais extremada de competição em um grupo de neurônios é chamada de *Winners Take All*, que mostra-se bastante eficiente.

Neste sentido, veremos as seguintes redes:

- Redes de Kohonen
- Mapas de Kohonen (*Self Organizing Maps*)
- Redes *Learning Vector Quantization*
- Redes ART
- Redes ART Modificada

Considerações Gerais

- Aplicações em problemas de *clusterização*[a] (segmentação)
- Número de classes conhecido "*a priori*"
- Cada neurônio da camada de saída corresponde a uma única classe
- O vetor de pesos para um neurônio da camada de saída é conhecido como *code vector*
- Aprendizado não supervisionado
- Neurônios são utilizados como medidores de similaridade

[a] Neologismo.

Classificação por Similaridade

Figura 1: Há padrões “mais parecidos” com uns que com outros.

⇒ Classes agrupam elementos “similares” entre si.

Classificação por Similaridade

Objetos físicos $\rightarrow \bigotimes_1, \bigotimes_2 \quad \Rightarrow \quad$ Objetos matemáticos $\rightarrow \underline{x}_1, \underline{x}_2$

Similaridade física $\quad \Rightarrow \quad$ Similaridade matemática

$$\bigotimes_1 \approx \bigotimes_2 \quad \Rightarrow \quad \underline{x}_1 \approx \underline{x}_2$$

Critérios de Pertinência

- Critérios de Pertinência a uma classe

Critério Básico $\longrightarrow$ Dois elementos pertencem à mesma classe se estão “próximos” entre si.

Critério dos k vizinhos mais próximos $\longrightarrow$ Considere os k elementos mais próximos da entrada (ou padrão) que se pretende classificar. A entrada pertence à classe a qual pertencem a maioria dos k vizinhos.

$\Rightarrow$ Se $k = 1$, então uma entrada pertencerá à classe que tiver um elemento mais próximo.

Critérios de Pertinência

$\Rightarrow$ POUCO PRÁTICO...

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

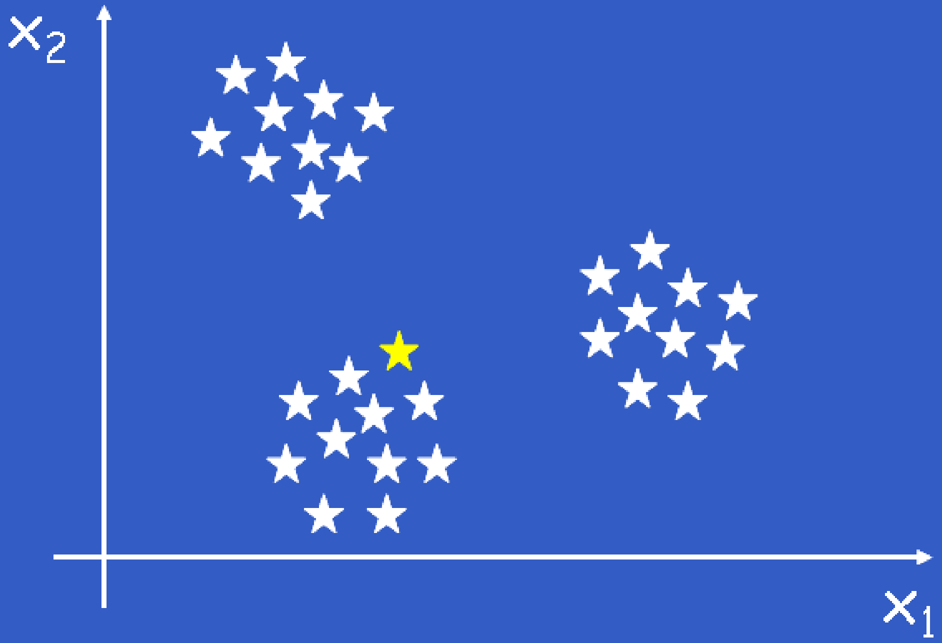


Figura 2: Vizinho mais próximo.

Critérios de Pertinência

Padrão de Classe

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

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

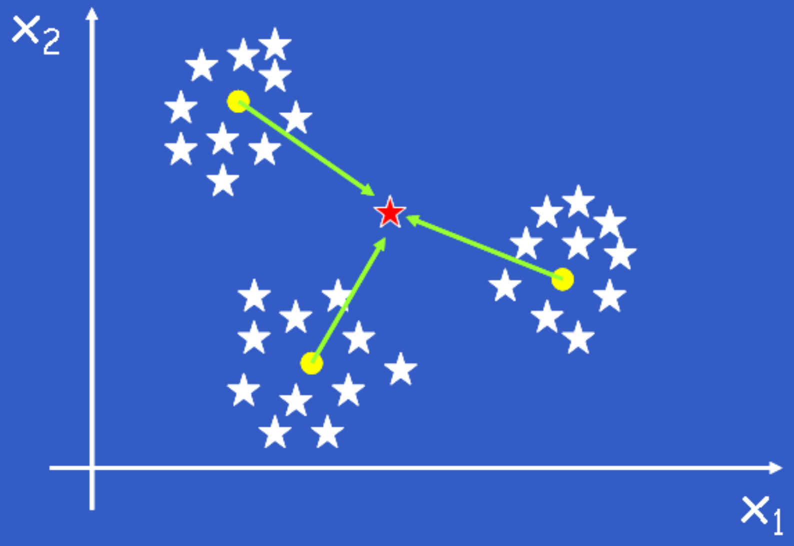
$\Rightarrow$ O baricentro é escolhido porque minimiza a dissimilaridade interna (dispersão intra-classe) ou erro de representação da classe.

Figura 3: Centros de classes.

Critérios de Pertinência

Primeiro critério de pertinência:

Centro de classe mais próximo $\longrightarrow \underline{x}_i \in C_i$ se, e somente se



$$\|\underline{x} - \underline{w}_i\|^2 < \|\underline{x} - \underline{w}_j\|^2, \forall j \neq i$$

Figura 4: 1^o critério de pertinência.

Critérios de Pertinência

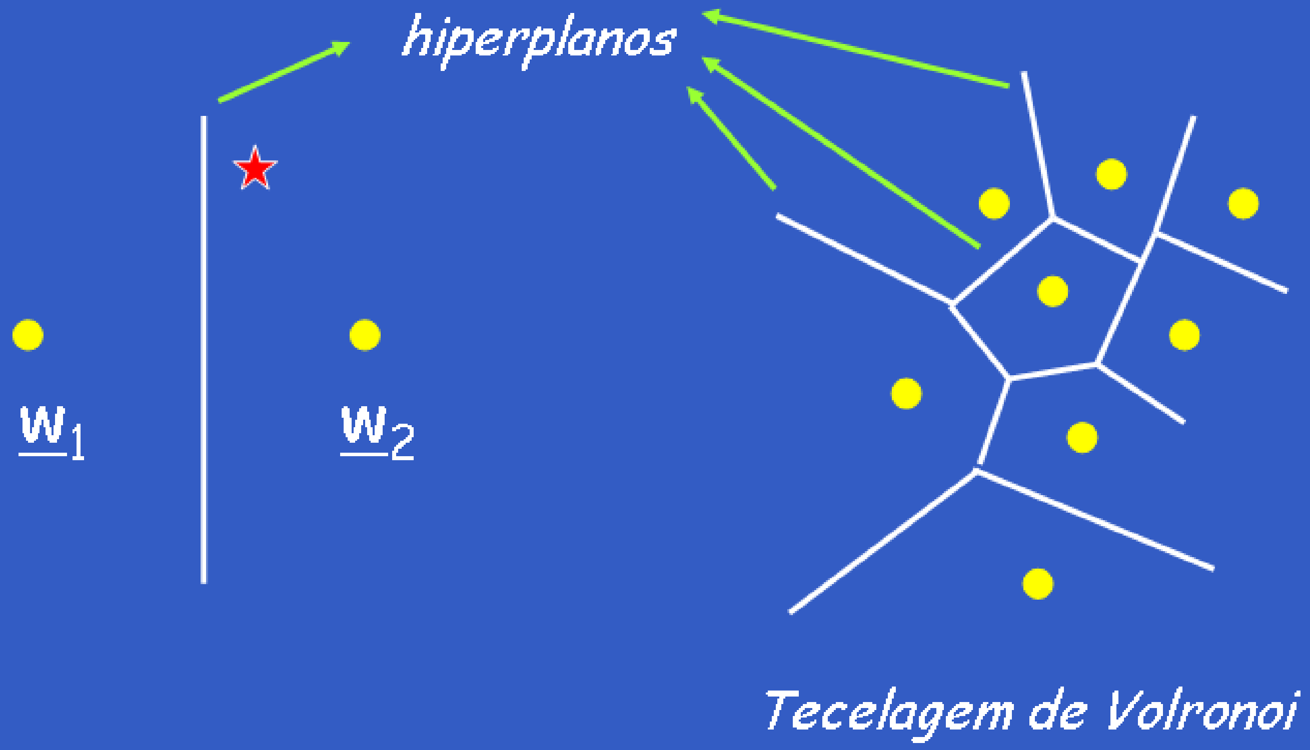


Figura 5: Separadores para o primeiro critério.

Critérios de Pertinência

Problema com o primeiro critério:

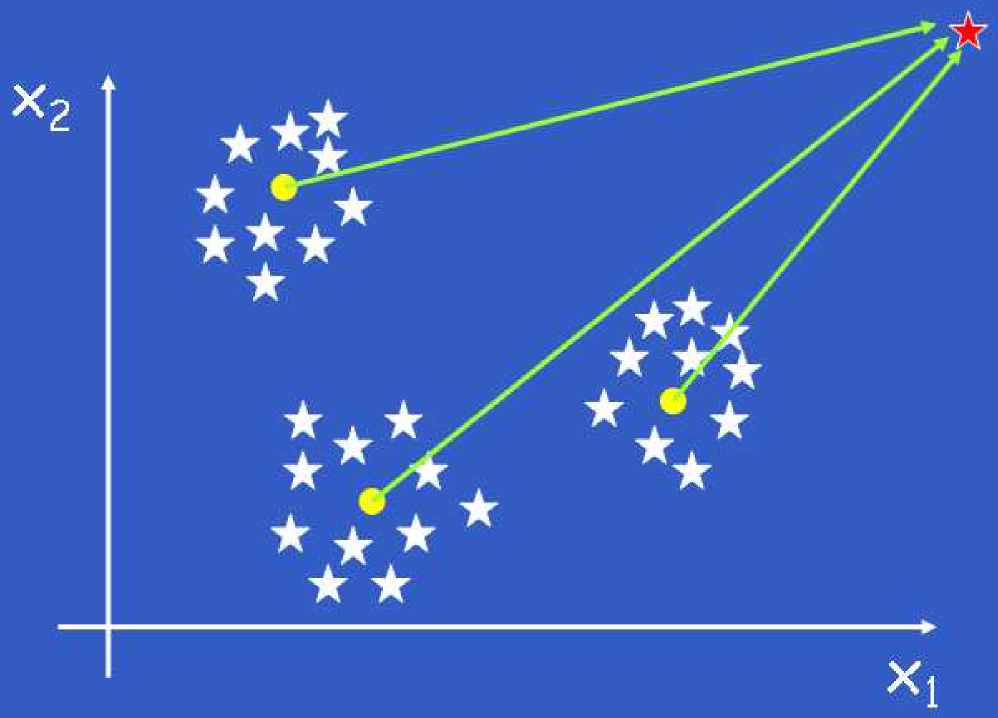


Figura 6: A qual classe pertence este padrão tão “diferente”?

Critérios de Pertinência

Solução $\rightarrow$ Adicionar um 2^o critério...o raio de similaridade mínima!

$$\underline{x}_i \in C_j \text{ sse } \|\underline{x}_i - \underline{w}_j\|^2 < r_0^2$$

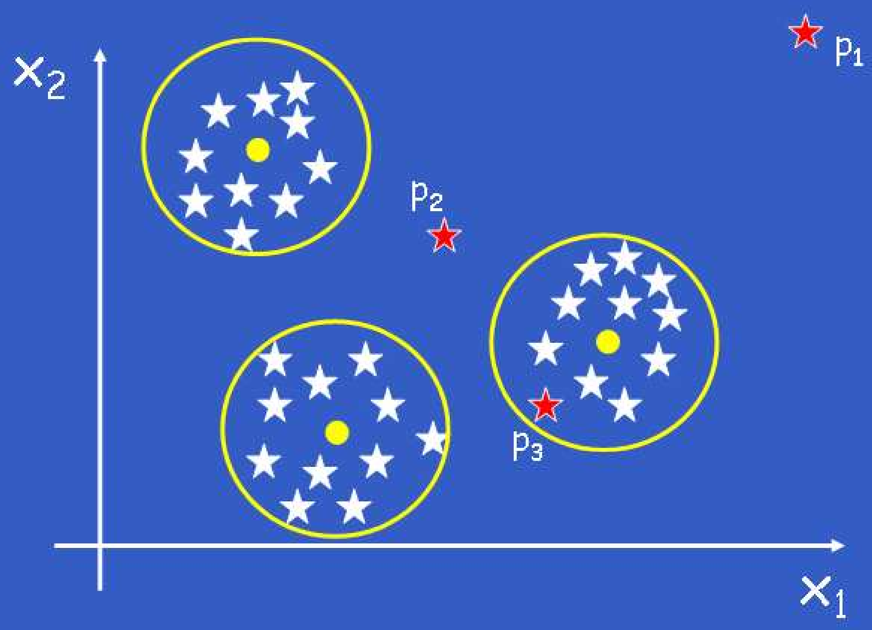


Figura 7: Apenas o padrão p_3 pertence a uma das classes.

Critérios de Pertinência

Reunindo agora os dois critérios, temos:

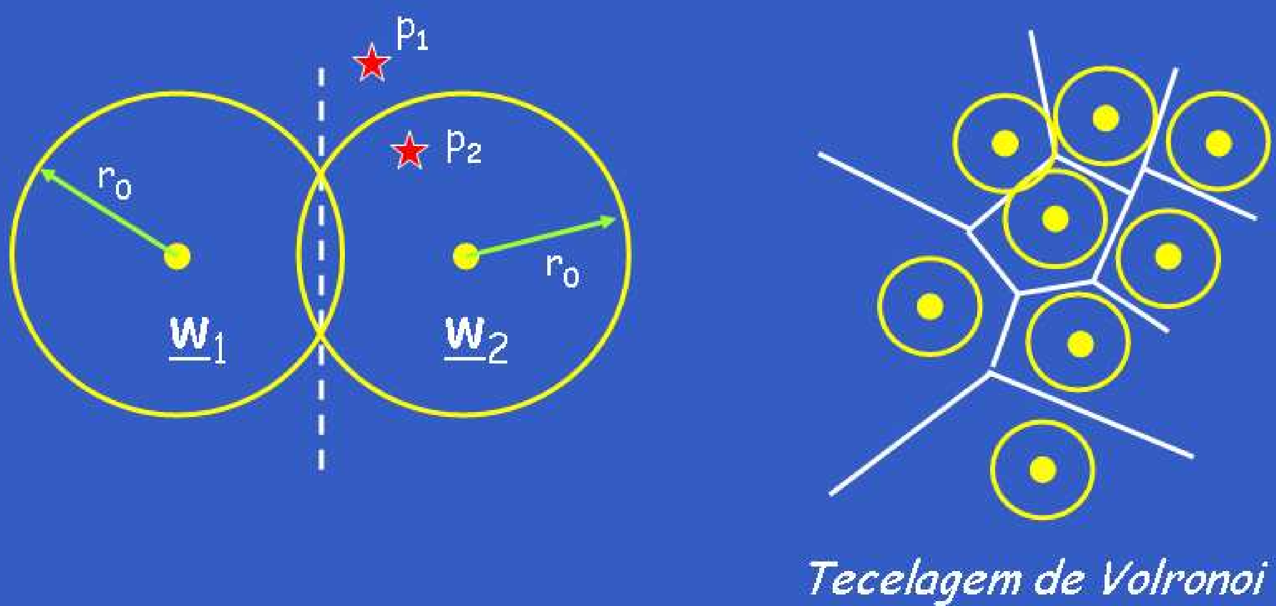


Figura 8: Separadores para os dois critérios reunidos.

Neurônio Medidor de Similaridade

- O Neurônio como Medidor de Similaridade

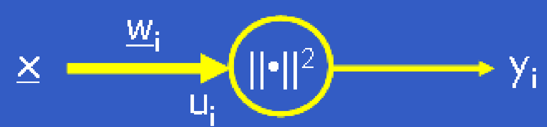


Figura 9: Novo modelo de neurônio.

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

- $u_i \rightarrow$ medida de similaridade entre $\underline{x}_i$ e $\underline{w}_i$;
- $u_i = 0 \rightarrow$ distância nula = máxima similaridade!
- y_i depende do outros neurônios.

Arquitetura da Rede

Seja:

- $\underline{x} = [x_1, x_2, \ldots, x_m]^T$
- $\underline{w}_j = [w_{j1}, w_{j2}, \ldots, w_{jm}]^T, \quad j = 1, 2, \ldots, m$

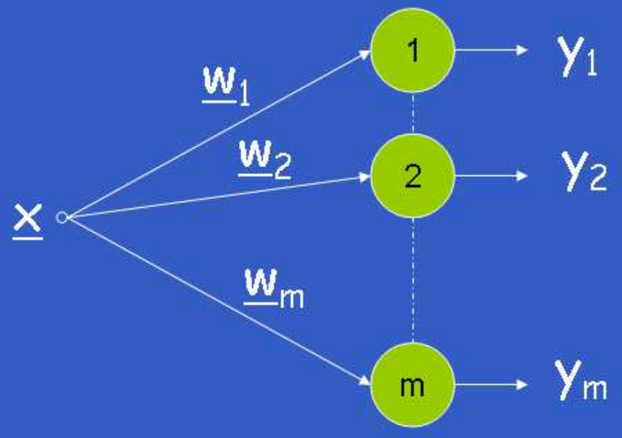


Figura 10: *Competição*.

- Classe C_i
- Padrão $\underline{w}_i$
- Indicador y_i

Se $y_i = 1$, então $\underline{x} \in C_i$ pelo critério 1 (padrão mais similar à entrada).

Arquitetura da Rede

Seja:

- $\underline{x} = [x_1, x_2, \ldots, x_m]^T$
- $\underline{w}_j = [w_{j1}, w_{j2}, \ldots, w_{jm}]^T, \quad j = 1, 2, \ldots, m$

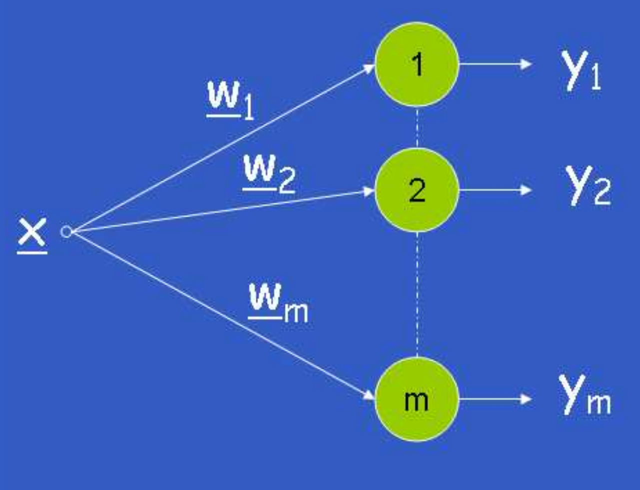


Figura 10: *Competição*.

Winner Takes All

$\Rightarrow$ Vence o neurônio cujo u_i for maior !

$y_i = 1$ sse $u_i > u_j, \forall j \neq i$

$y_i = 0$ caso contrário.

Treinamento

Os pesos da rede podem ser encontrados de três formas diferentes:

1. Se os padrões $C_i, i = 1, 2, \ldots, m$, já forem conhecidos, então $\underline{w}_i =$ baricentro $\{C_i\}$;
2. Se os padrões não forem conhecidos, mas a base de dados é rotulada, então:

$$\underline{w}_i = \frac{1}{N_i} \sum_{\forall \underline{x}_j \in C_i} \underline{x}_j$$

3. Se os padrões não forem conhecidos, e nem a base é rotulada, então $\longrightarrow$ Treinamento Iterativo (ou on-line).

Treinamento

- Treinamento da Rede de Aprendizado Competitivo

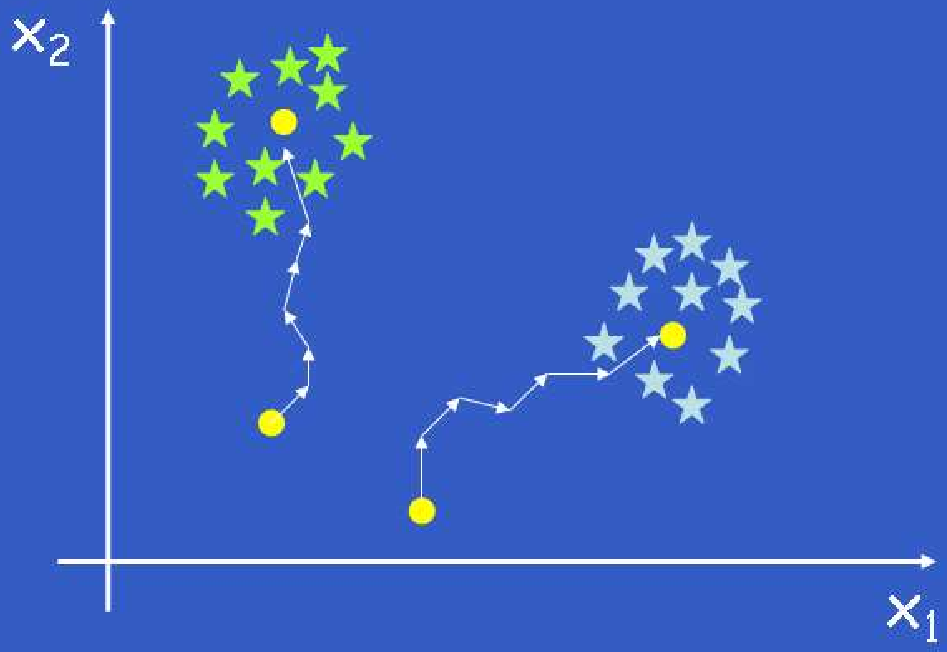


Figura 11: Treinamento iterativo (*on-line*) da Rede.

Treinamento

- Treinamento da Rede de Aprendizado Competitivo

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

Logo,

$$\begin{aligned} \underline{w}_i(n+1) &= \underline{w}_i(n) + \alpha[\underline{x}(n) - \underline{w}_i(n)] \\ &= (1-\alpha)\underline{w}_i(n) + \alpha\underline{x}(n) \\ \underline{w}_j(n+1) &= \underline{w}_j(n), \quad \forall j \neq i. \end{aligned}$$

Treinamento

- Convergência

$\Rightarrow$ Para que ponto o algoritmo converge?

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

Treinamento

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

Treinamento

Assumindo as seguintes premissas:

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

Temos:

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

Treinamento

$$\sum_{i=0}^{\infty}(1-\alpha)^i = \frac{1}{\alpha}, \qquad \text{então}$$

$$\alpha \approx \frac{1}{\sum_{i=0}^{n}(1-\alpha)^i}, \qquad n >> 1$$

$$\begin{aligned} \underline{w}(n) &= \underbrace{(1-\alpha)^n}_{0}\underline{w}(1) + \alpha\sum_{i=1}^{n}(1-\alpha)^{n-i}\underline{x}(i) \\ &\approx \frac{1}{\sum_{i=0}^{n}(1-\alpha)^{n-i}}\sum_{i=0}^{n}(1-\alpha)^{n-i}\underline{x}(i) \end{aligned}$$

$\Rightarrow$ Média das entradas que pertencem à classe ponderada geometricamente pelo tempo !.

Treinamento

$$
\begin{aligned}
\underline{w}(n) &\approx \frac{\sum_{i=0}^{n}(1-\alpha)^{n-i}\underline{x}(i)}{\sum_{i=0}^{n}(1-\alpha)^{n-i}} \\
&\approx \frac{1}{\sum_{i=0}^{n}(1-\alpha)^{i}} \sum_{i=0}^{n}(1-\alpha)^{i}\underline{x}(n-i)
\end{aligned}
$$

- n = Instante atual
- $\underline{x}(n-i)$ = entrada atrasada de i intervalos de tempo
- $(1-\alpha)^{i}$ = ponderador para a entrada $\underline{x}(n-i)$

Treinamento

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

- Para $n = 0 \rightarrow$ Ponderador = $(1-\alpha)^0 = 1$
- Último atraso a ser considerado $\rightarrow$ Ponderador = $0,02 = (1-\alpha)^i$

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

Treinamento

- Erro na determinação do baricentro

Cada componente j da entrada $\underline{x}$ tem um ruído de média nula e variância σ_{xj}^2. A componente j de $\underline{w}$, w_j, em um instante $n >> 1$ é dada por:

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

Sendo uma soma ponderada, sua variância será dada por:

$$\sigma_{wj}^2 \approx \left(\frac{1}{\sum_{i=0}^{\infty} (1-\alpha)^i} \right)^2 \sum_{i=0}^{\infty} (1-\alpha)^{2i} \sigma_{xj}^2 \approx \frac{\alpha}{2} \sigma_{xj}^2$$

Treinamento

- Erro $\times$ Tempo de Medida

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

$$\sigma_{wj}^2 \approx \frac{\alpha}{2}\sigma_{xj}^2 \qquad \# passos = i \approx \frac{4}{\alpha}$$

Exemplo: $\sigma_x = 0,1 \quad (10\%), \quad \sigma_w = 0,01 \quad (0,1\%)$ requerido

$$\alpha = \frac{2\sigma_{wj}^2}{\sigma_{xj}^2} = 0,02 \qquad \# passos = \frac{4}{\alpha} = 200 \text{ passos}$$

Treinamento

- Passo de Treinamento

Reduzir ao longo do tempo...

$$\alpha(n) = \alpha_0 e^{-\frac{n}{N_0}} \tag{1}$$

onde N_0 é uma constante.

Podemos reescrever (1) como:

$$\alpha(n+1) = k\alpha(n) \tag{2}$$

Usando a equação (1) recursivamente, podemos provar a relação (2):

$$\begin{aligned}
\alpha(0) &= \alpha_0 e^0 \\
\alpha(1) &= \alpha_0 . e^{-\frac{1}{N_0}} \\
&= \alpha(0) . e^{-\frac{1}{N_0}} \\
\alpha(2) &= \alpha_0 . e^{-\frac{2}{N_0}} \\
&= \alpha_0 . e^{-\frac{1}{N_0}} . e^{-\frac{1}{N_0}} \\
&= \alpha(1) . e^{-\frac{1}{N_0}} \\
\alpha(3) &= \alpha_0 . e^{-\frac{3}{N_0}} \\
&= \alpha_0 . e^{-\frac{2}{N_0}} . e^{-\frac{1}{N_0}} \\
&= \alpha(2) . e^{-\frac{1}{N_0}} \\
\vdots &= \vdots \\
\alpha(n+1) &= \alpha(n) . e^{-\frac{1}{N_0}}
\end{aligned}$$

Expandindo o termo $e^{-\frac{1}{N_0}}$ em série, temos que:

$$e^{-\frac{1}{N_0}} \approx 1 - \frac{1}{N_0}$$

Para $N_0 \rightarrow \infty$.

Logo:

$$\alpha(n+1) = k\alpha(n), \qquad k = 1 - \frac{1}{N_0}$$

Treinamento

- Passo de Treinamento

E as sinapses pouco treinadas (classes pouco populosas)?

Usar α diferenciado por sinapse !

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

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

Treinamento

- Treinamento dinâmico

E se um padrão variar ao longo do tempo, como saber?

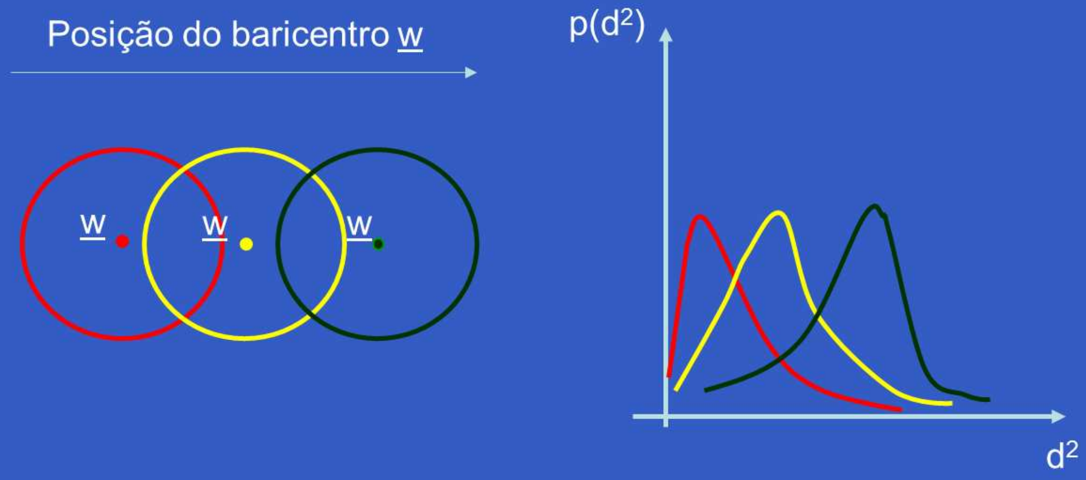


Figura 12: Padrão variante no tempo.

Treinamento

- Treinamento dinâmico

O valor médio (ou da moda) de $u_i = -d_i^2$ diminui, o que significa que a distância média aumenta.

Para corrigir, basta religar o treinamento até que d_i^2 volte a seus valores normais.

Se a variação do baricentro for lenta e o treinamento permanecer ligado, a rede poderá “rastrear” automaticamente a mudança dos padrões ao longo do tempo, adaptando-se às novas necessidades. Isto é o que chamamos de plasticidade da rede neural.

Treinamento

- Treinamento dinâmico

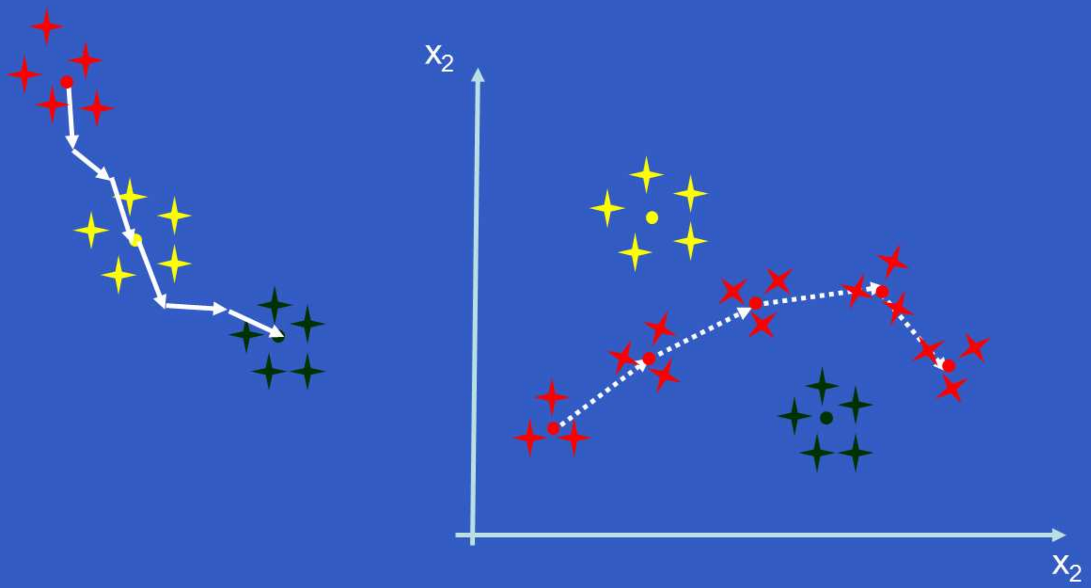


Figura 13: A rede pode rastrear mudanças nos padrões.

Treinamento

- Fim do Treinamento?

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

$$\begin{aligned} \Delta \underline{w} &= \alpha(\underline{x} - \underline{w}) \\ E(\Delta \underline{w}) &= E[\alpha(\underline{x} - \underline{w})] \\ &= \alpha[E(\underline{x}) - \underline{w})] = 0 \end{aligned}$$

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

Considerações Sobre o Treinamento

- Problemas no treinamento

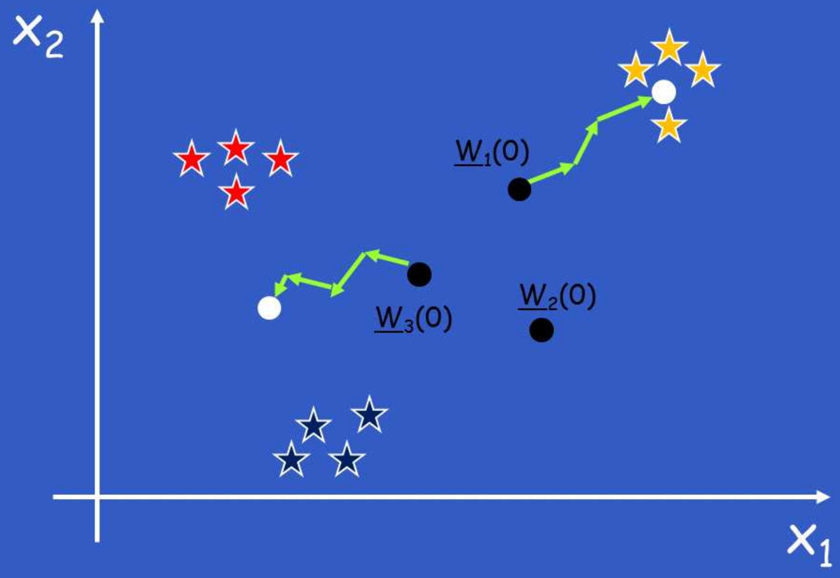


Figura 14: A inicialização do algoritmo é crítica !

Considerações Sobre o Treinamento

- Possíveis soluções

  - Utilizar um número maior de sinapses do que o número de classes;
  - Inicializar as classes utilizandos os próprios dados: $\underline{w}_i = \underline{x}(i)$;
  - Fazer com que as sinapses iniciais não sejam muito próximas umas das outras;
  - Adotar um mecanismo de consciência.

Considerações Sobre o Treinamento

- Possíveis soluções

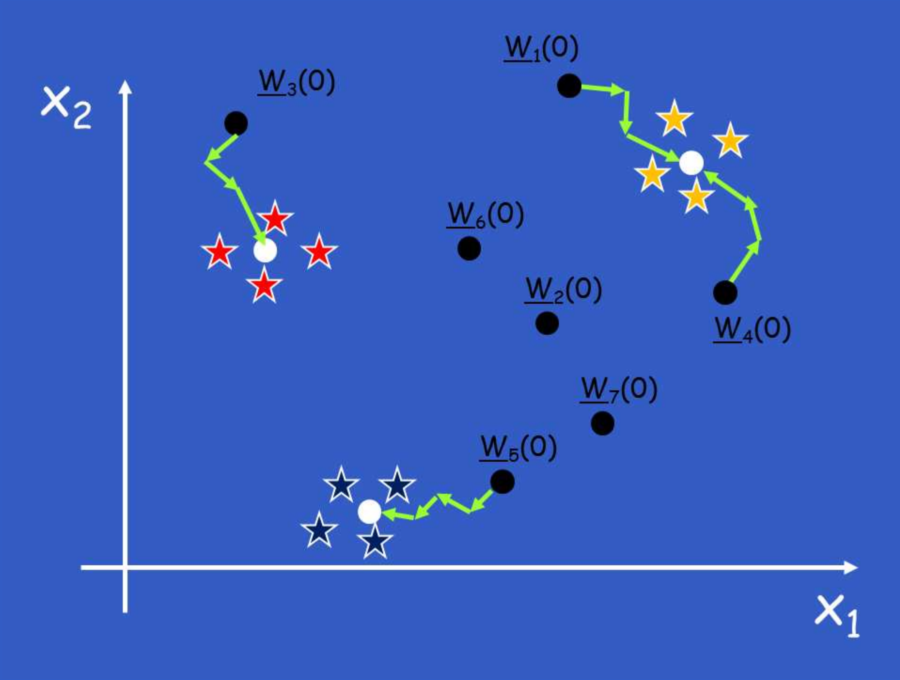


Figura 15: Inicializar várias sinapses de forma randômica !

Considerações Sobre o Treinamento

- Possíveis soluções

Para entradas “não muito próximas”, podemos adotar um critério aproximado: $d_{ij} = |\underline{x}_i - \underline{x}_j| \geq \varphi/N$, onde N é o número de classes.

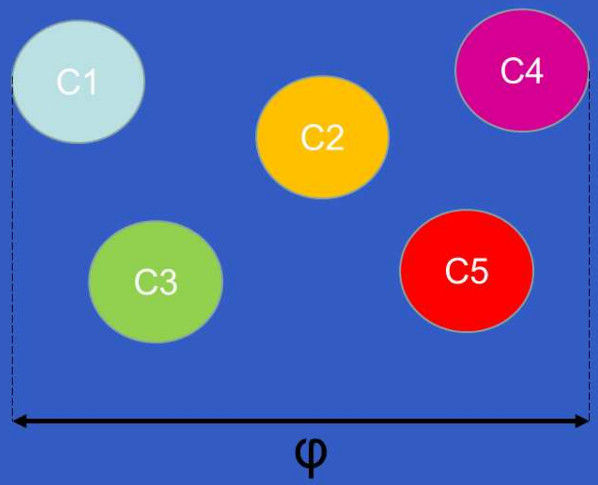


Figura 16: Distância inter-classes φ (d^{ss}_{max} - Aula de clusterização II).

Considerações Sobre o Treinamento

- Mecanismo de consciência

O neurônio que treinou muitas vezes abre mão para o segundo vencedor. Para isso, é adicionado um *threshold* w_{i0} ao neurônio que mais venceu.

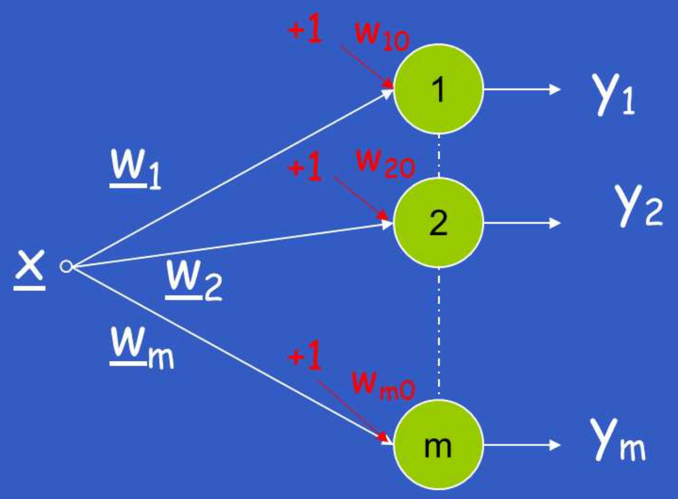


Figura 17: Mecanismo de consciência na rede de Kohonen.

Considerações Sobre o Treinamento

- Mecanismo de consciência

Logo:

$$\begin{aligned} u_j &= -d_j^2 + w_{j0} \\ u_k &= -d_k^2 + w_{k0} \end{aligned}$$

Para $u_j > u_k$, temos que

$$d_j^2 - w_{j0} < d_k^2 - w_{k0}$$

$$d_j^2 < d_k^2 - (w_{k0} - w_{j0})$$

Considerações Sobre o Treinamento

- Formas de consciência

- Abrupta

$$p_i = \begin{cases} < p^*, & w_{i0} = 0; \\ \geq p^*, & w_{i0} = \varphi \end{cases}$$

- Suave

$$w_{i0} = -\frac{\varphi}{2}\{1 - \tanh[k(p_i - p^*)]\}$$

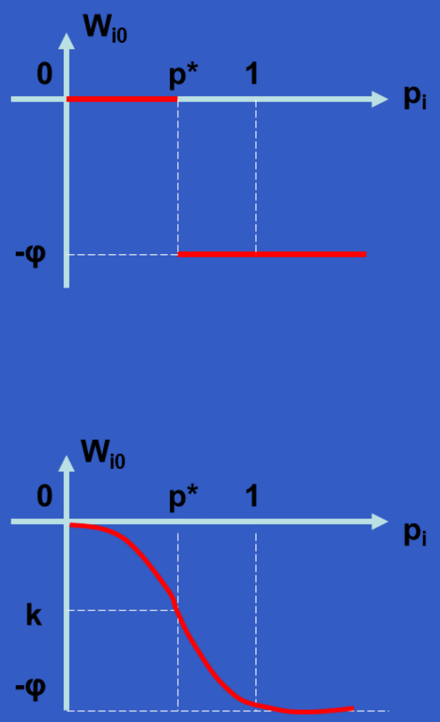


Figura 18: Formas de consciência.

Considerações Sobre o Treinamento

- Formas de consciência

Para a forma suave, podemos calcular o valor de p_i sa seguinte forma:

$$y_i \in \{0, 1\}$$

$$p_i(0) = 0$$

$$p_i(n+1) = (1-\alpha)p_i(n) + \alpha y_i(n)$$

$$p_i(n) \approx \frac{\sum_{j=1}^{n}(1-\alpha)^{n-j}y(j)}{\sum_{j=1}^{n}(1-\alpha)^{n-j}}$$

Considerações Sobre o Treinamento

- Valores Iniciais das Sinapses

A escolha de valores de classes diferentes como pesos iniciais acelera a convergência pois a sinapse já começa dentro da classe.

- Passo de Treinamento

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

- $\alpha(0) = \alpha_0$
- $\alpha(n+1) = k\alpha(n)$
- $k = 1 - 1/N_0$

$\Rightarrow$ O processo acaba ($\alpha(n) << \alpha_0$) para $n > 4N_0$.

Considerações Sobre o Treinamento

- Valores Iniciais das Sinapses

E as sinapses pouco ativadas (classes pouco populosas)?

$\Rightarrow$ Usar α diferenciado por sinapse !

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

onde n_j é o número de vezes que a sinapse $\underline{w}_j$ foi treinada.

Considerações Sobre o Treinamento

- Treinamento dinâmico, adaptativo

Uma classe varia sua posição ao longo do tempo. Como saber?

O valor médio (ou da moda) de $u_i = -d_i^2$ diminui (a distância média aumenta).

Como corrigir? "Ligando" o treinamento ate que d_i^2 volte aos seus valores normais.

Considerações Sobre o Treinamento

- Como escolher r_0?

Frequentemente, as classes geradas são geradas a partir de um padrão com ruído Gaussiano aditivo $\Rightarrow$ Logo, o raio r_0 depende do ruído !

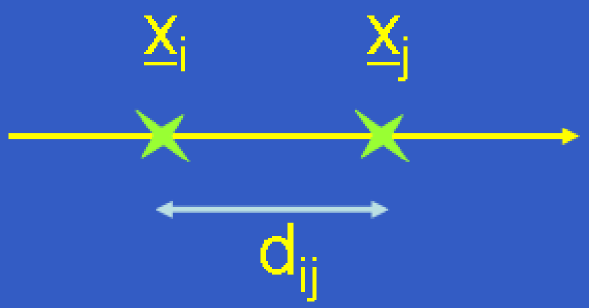
$$\begin{aligned} \underline{x} &\in C_k \\ \underline{x} &= \underline{w}_k + \underline{r} \\ x_j &= w_j + r_j \\ \mu_{r_j} = 0 \quad & \quad \sigma^2_{x_j} = \sigma^2_{r_j} \end{aligned}$$

Considerações Sobre o Treinamento

- Caracterização do ruído

$\Rightarrow$ Caso de uma única classe com dimensão 1!

$$\begin{aligned} \underline{x} &\in C_k \\ \underline{x} &= \underline{w}_k + \underline{r} \\ x_j &= w_j + r_j \\ \mu_{r_j} = 0 & \qquad \sigma^2_{x_j} = \sigma^2_{r_j} \end{aligned}$$



Considerações Sobre o Treinamento

- Caracterização do ruído

$\Rightarrow$ Medir a distância entre entradas entre classes.

Críticas Pós-Treinamento

- Neurônios não (ou pouco) treinados devem ser eliminados;
- Dois neurônios podem partilhar a mesma classe. Neste caso:

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

Fazer a média aritmética entre w_i e w_j para obter um único neurônio representando a classe !

Template Matching

Outros problemas:

- Entrada apresentada à rede está rotacionada em torno de um eixo;
- Entrada apresentada à rede está levemente deformada em relação ao padrão original.

Exemplo: Reconhecimento de objetos em imagens 2D, porém o objeto pode apresentar-se rotacionado em torno de um ou mais eixos.

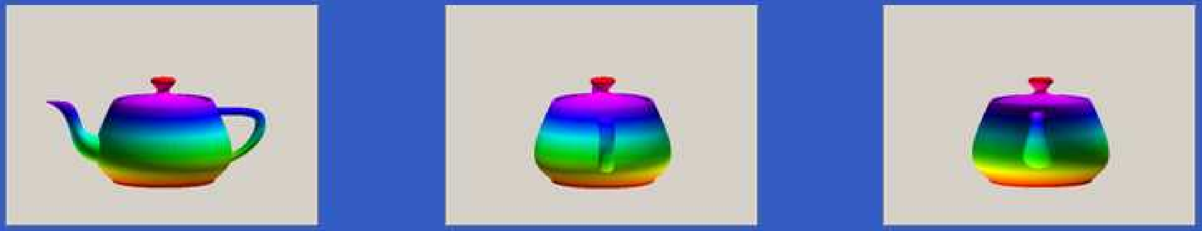


Figura 19: Bule

Template Matching

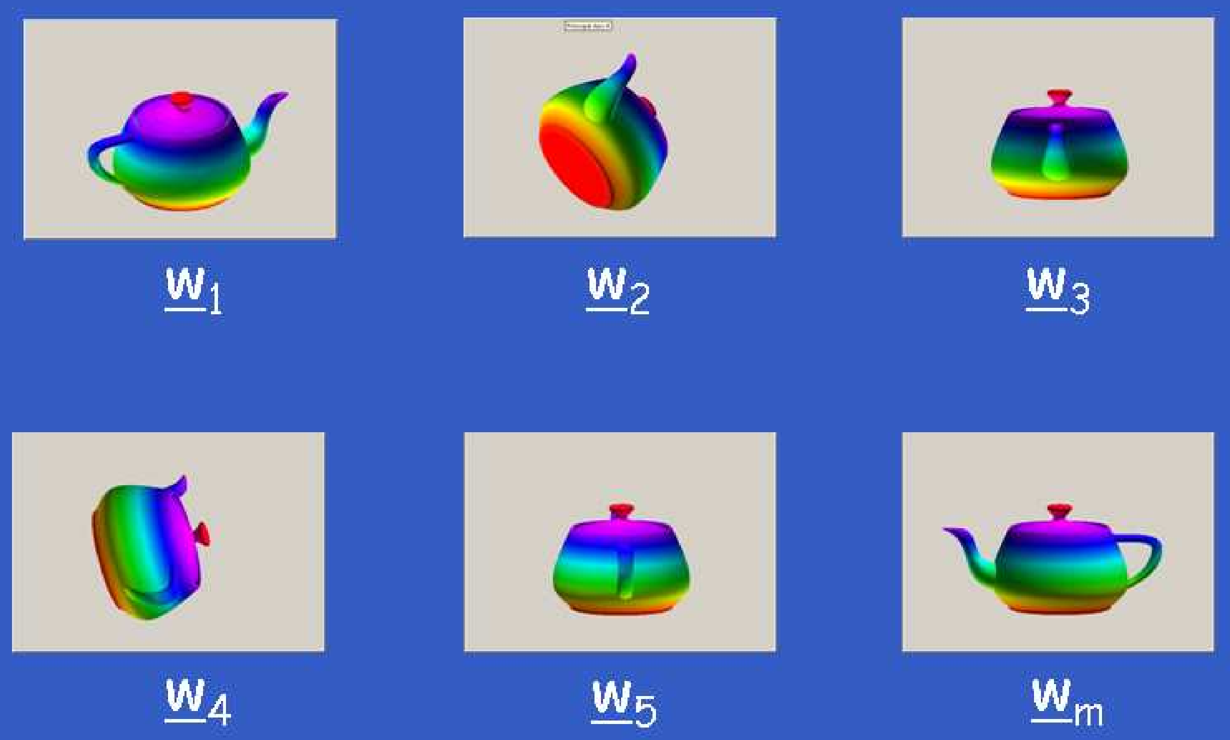


Figura 20: Template Matching do bule.

Template Matching

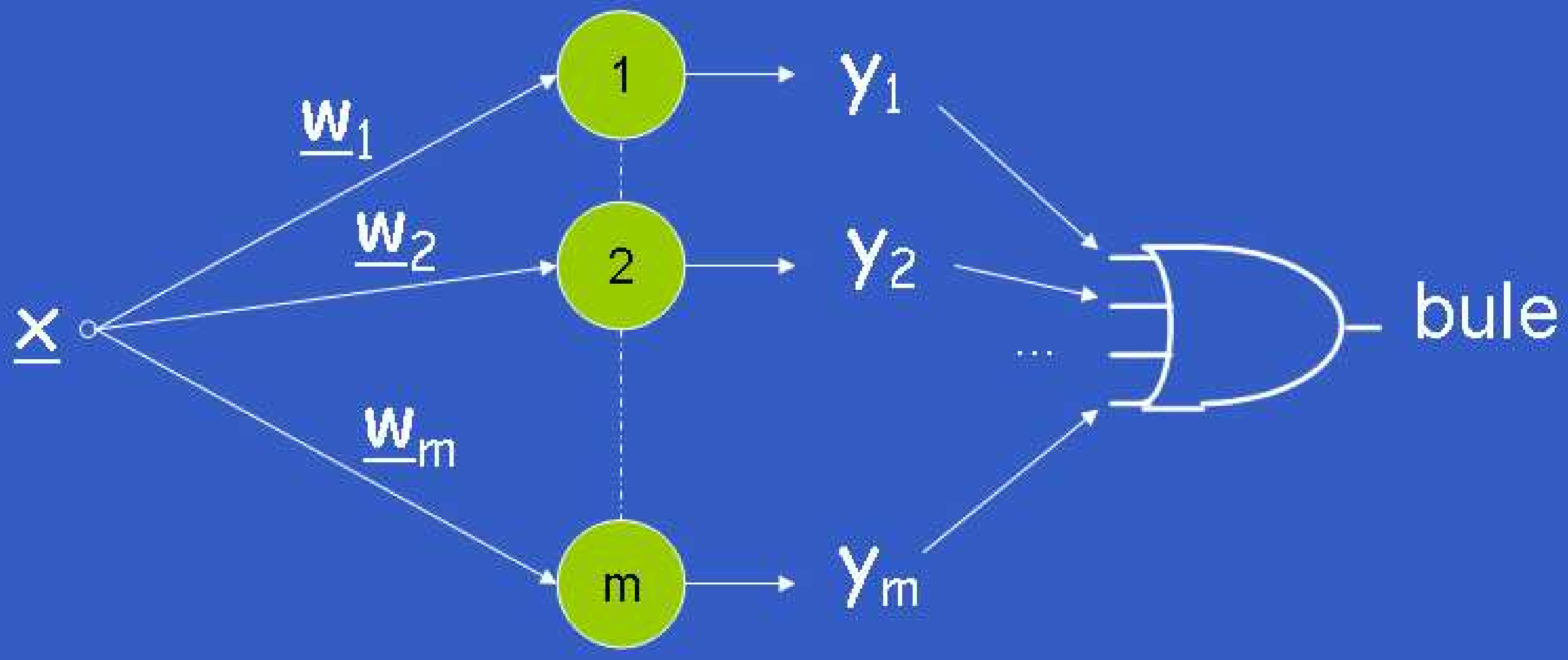


Figura 21: Reconhecimento do bule.

Template Matching

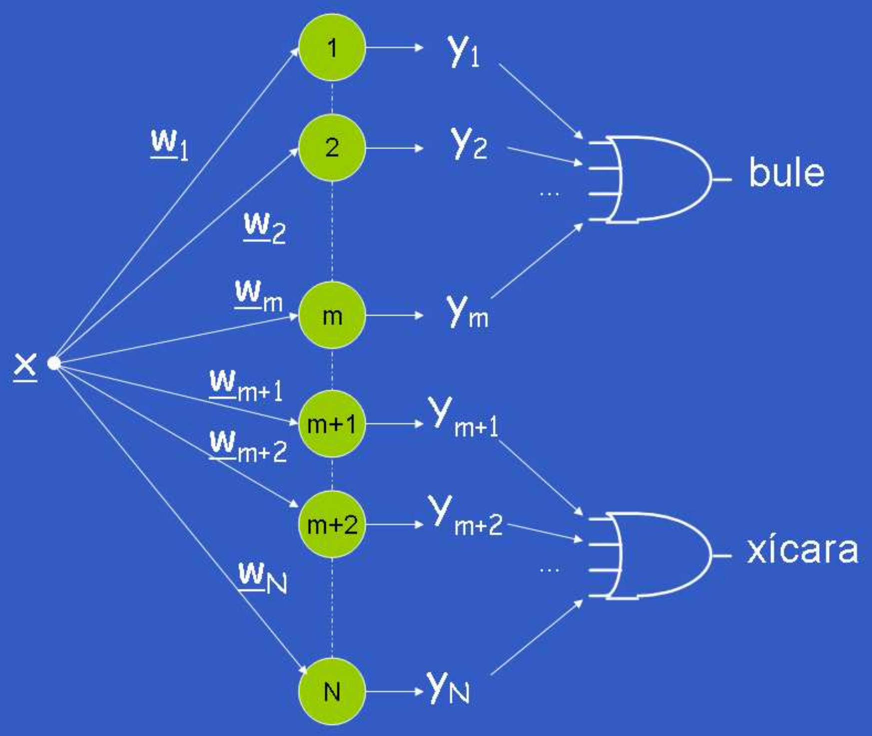


Figura 22: Reconhecendo objetos em uma cena.

Referências

1. Haykin, S., "*Neural Networks - A Comprehensive Foundation*," 2^{nd} edition - Prentice Hall, 1999.
2. Fausett, L., "*Fundamental of Neural Networks - Architectures, Algorithms and Applications*," Prentice Hall, 1994.
3. Theodoris, S., Koutroumbas, K, "*Pattern Recognition*," 4^{th} edition, Academic Press.

FIM