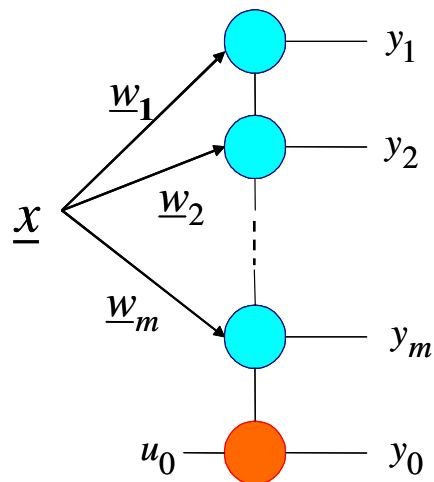


Rede ART (modificada)

Condições iniciais:

$$\mathbf{u}_0 = -\mathbf{r}_0^2 \quad \text{e}$$

todos os neurônios $\underline{w}_i(0)$
estão desativados



Obs: neurônio desativado não tem condição de ganhar a competição

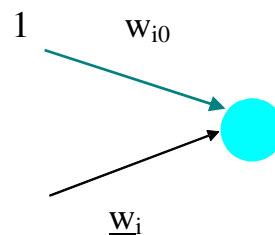
Neurônio $\underline{w}_i$ desativado:

- Não existe na rede (melhor) ou

$$- |\vec{w}_i|^2 > \text{Max} |\vec{x}|^2 + r_0^2 \quad \text{ou}$$

- usar bias $w_{i0} = -r_0^2$

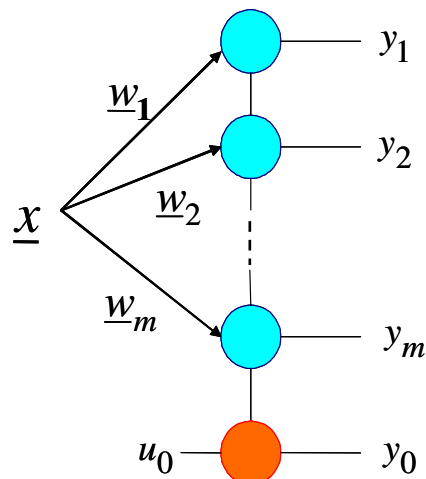
$$w_{i0} = \begin{cases} 0 & \text{neurônio ativo} \\ u_0 = -r_0^2 & \text{neurônio desativado} \end{cases}$$



Treinamento:

Apresentar entrada $\underline{x}(n)$

Neurônio vencedor y_0 ou y_i



Se $y_0(n) = 1$

Ativar um neurônio i desativado

$$\underline{w}_i(n) \text{ desativado} \Rightarrow \underline{w}_i(n+1) = \underline{x}(n)$$

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

Se $y_i(n) = 1$

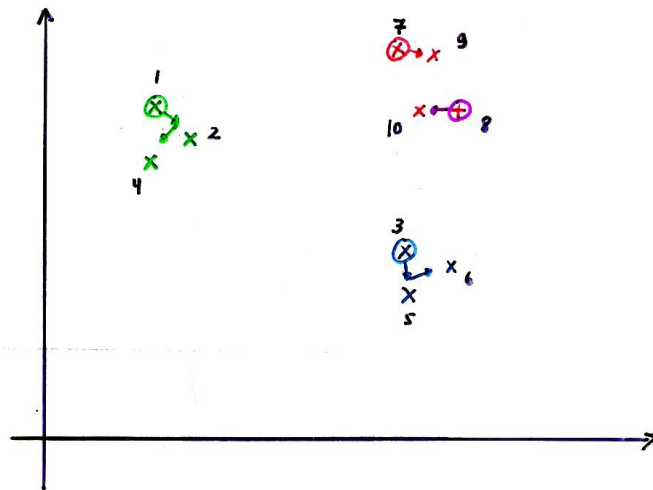
Treinar o neurônio vencedor, i

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

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

Plasticidade, Treinamento competitivo

Treinamento ART



Treinamento ART

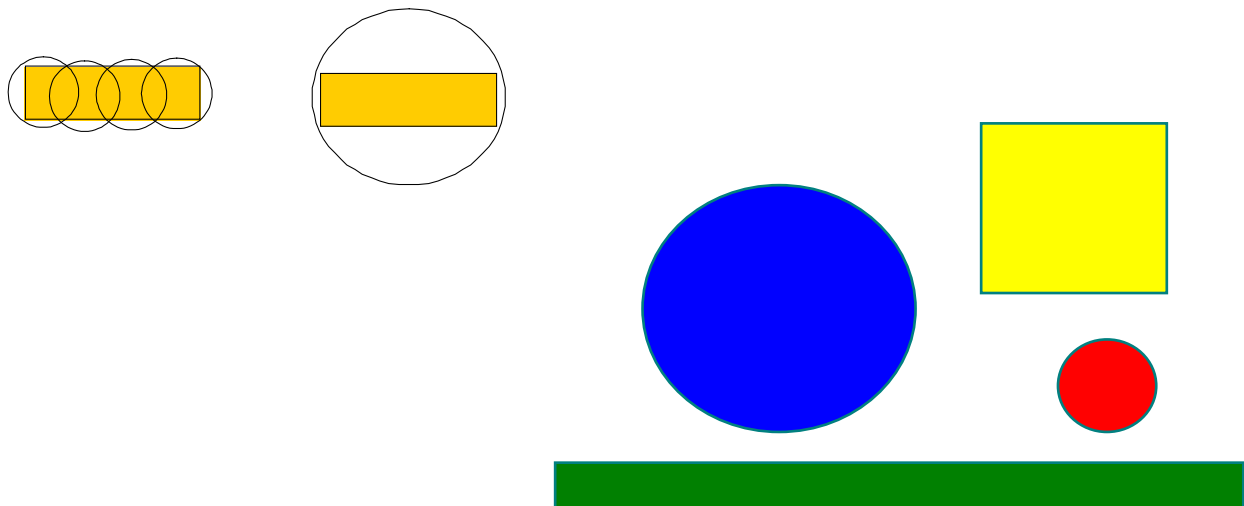
INPUT	EXEMPLARS AFTER EACH INPUT
C	C
E	C E
F	C E F
F	C E F
F	C E F F

Figure 11. An example of the behavior of the Carpenter Grossberg net for letter patterns. Binary input patterns on the left were applied sequentially starting with the upper "C" pattern. Exemplars formed by top-down connection weights after each input was presented are shown at the right.

Ponto crítico: Granularidade, escolha de r_0

muito pequeno: boa definição, mas muitos neurônios

muito grande: poucos neurônios, mas definição pobre



Como (será) discutido no pré-processamento, no caso em que

1 – as classes são geradas a partir de padrões adicionados de ruído gaussiano com média nula.

2 – o valor rms (desvio padrão) do ruído de cada componente x_i é σ_i , dependente da componente mas independente da classe.

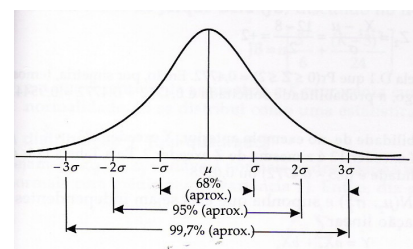
3 - o fator de escala $k_i = \frac{1}{\sigma_i}$ foi utilizado para cada componentes x_i

então as classes tem distribuição gaussiana multidimensional com desvio padrão unitário em todas as direções. Em consequência

$r_0 = 1$ captura 68 % dos elementos da classe,

$r_0 = 2$ captura 95 % dos elementos da classe,

$r_0 = 3$ captura 99,7 % dos elementos da classe, etc.



Obs 1:

Se os ruídos das componentes forem correlatos os domínios das classes serão elipsóides, mas estarão contidos nas esferas determinadas. A decorrelação dos ruídos permitirá uma melhor determinação dos domínios das classes, que passarão a ser esféricos.

Obs 2:

Casos envolvendo classes com domínios não esféricos serão vistos a frente, mas as considerações sobre o escalamento k_i e r_0 feitas aqui continuarão válidas.

ART

Esquecimento

pode ser usado na fase de operação e/ou na de treinamento.

Esquecimento (abrupto)

Se $y_i = 0$ por N entradas consecutivas

Guarde a informação para o futuro como
uma lembrança $\underline{\mathbf{m}}_k$, $\underline{\mathbf{m}}_k = \underline{\mathbf{w}}_i$ (opcional)

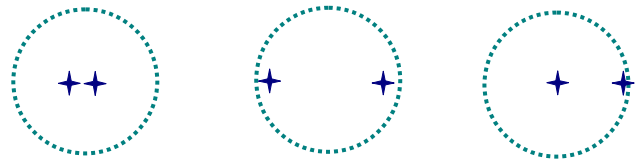
Desative o neurônio i, $\underline{\mathbf{w}}_i$

Esquecer elimina padrões não mais usados, diminui o número de sinapses e reduz o trabalho computacional

No treinamento o esquecimento pode ser usado para eliminar automaticamente padrões espúrios, porque inicialmente estes são muito pouco ativados.

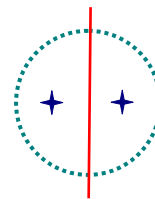
Evolução de padrões múltiplos por classe

Posições iniciais



Posição final

(com ou sem consciência)



mas a evolução a partir da posição inicial assimétrica é mais lenta sem consciência.

Lembrança

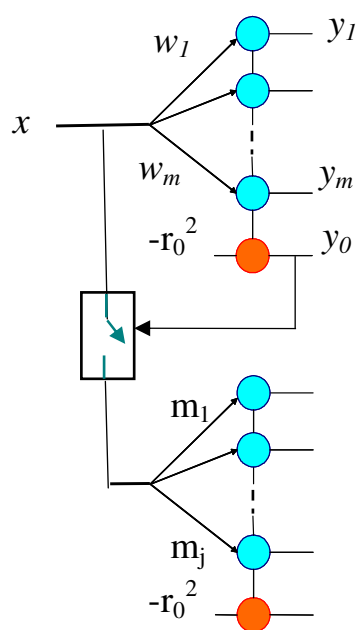
Se $y_0 = 0$ Ativar neurônio, mas

Verifique se algum padrão $\underline{\mathbf{m}}_k$ estocado serve

$$-\|\underline{\mathbf{x}} - \underline{\mathbf{m}}_k\|^2 > -r_0^2 \quad ?$$

sim = existe lembrança $\underline{\mathbf{m}}_k$, ative $\underline{\mathbf{w}}_j = \underline{\mathbf{m}}_k$

não = não existe lembrança, ative $\underline{\mathbf{w}}_j = \underline{\mathbf{x}}$



Memória Principal

Lembranças

Esquecimento / Lembrança graduais

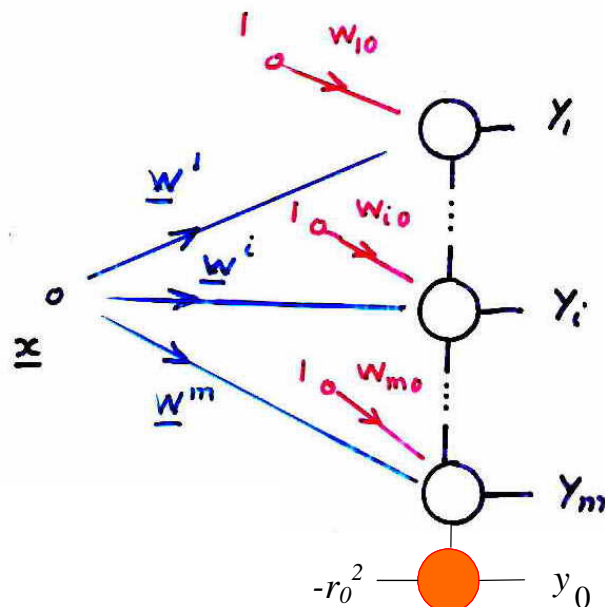
funciona como uma “consciência invertida”, neurônios pouco ativados vão tendo sua possibilidade de vencer reduzida e os muito ativados aumentada.

É adicionado um bias w_{i0} á cada neurônio, que representará um “handicap” para o(s) neurônio(s) pouco ativados e uma vantagem para os muito ativados

Isto que corresponde a alterar o raio de vigilância do neurônio para r_{i0}

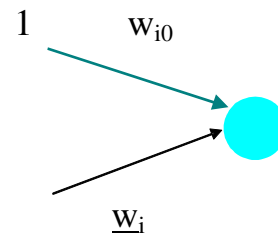
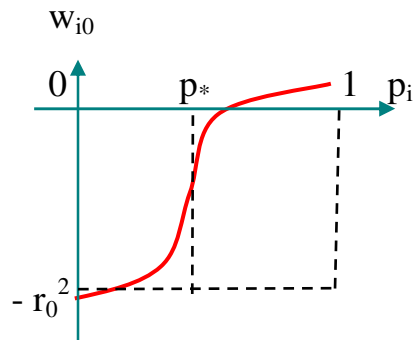
$$r_{i0}^2 = r_0^2 + w_{i0}$$

$w_{i0} < 0$ reduz o raio, $w_{i0} > 0$ aumenta o raio



w_{i0} deve variar segundo uma sigmóide em função do percentual de ativações (vitórias) do neurônio i

$$w_{i0} = a + b \tanh[c(p_i - p_*)]$$

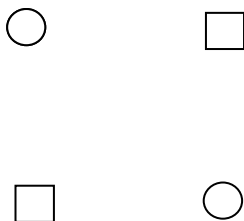


ART

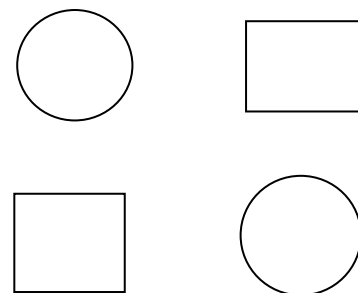
Demo:

Kohonen – Domínio das Classes

1a



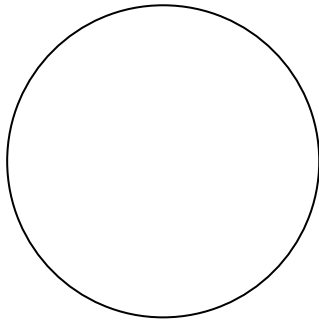
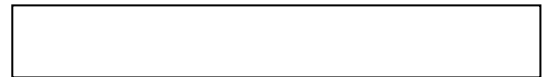
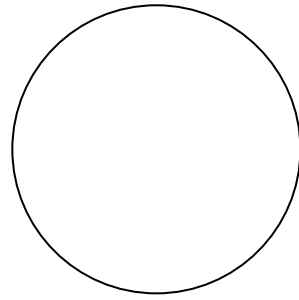
1b



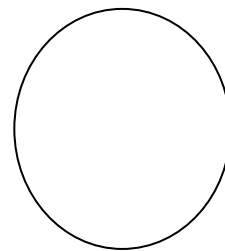
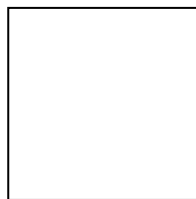
$N = 3, 4, 5$

Consciência

Kohonen vs ART r_0

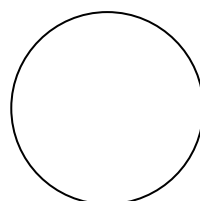
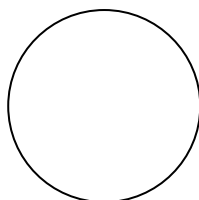
2 – esfera grande**3 – esfera e retângulo**

efeito de r_0
efeito de padrões espúrios

4 – áreas iguais**5 – efeito das populações**

5a – 50 – 50 %

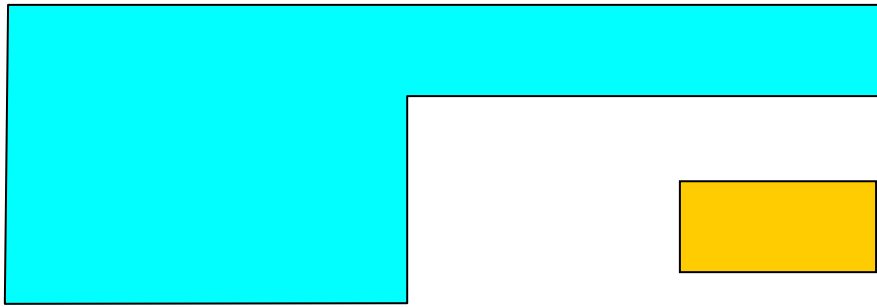
5b – 20 – 80 %



efeito da consciência

Classes Não-Esféricas

Treinamento Não-Supervisionado

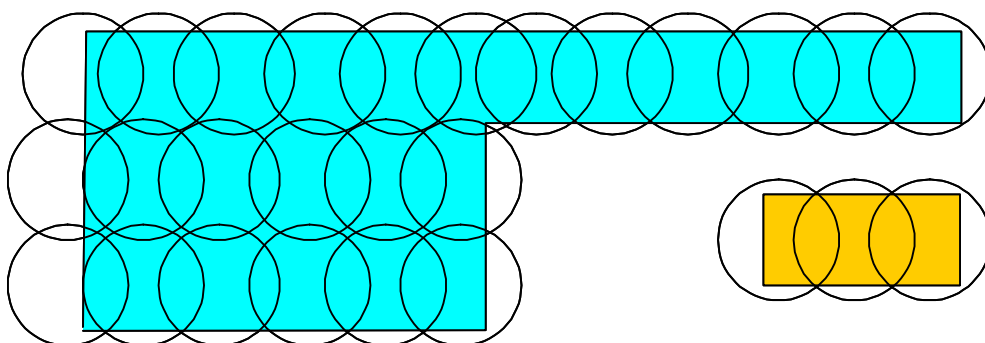


- 1a. Etapa – estabelecer o raio de similaridade a ser usado
- 2a. Etapa – ART - determinar os domínios esféricos
- 3a. Etapa – determinar os domínios vizinhos
- 4a. Etapa – OU – conecta os domínios vizinhos

- 1a. Etapa -
escolher um raio de similaridade r_0 adequado

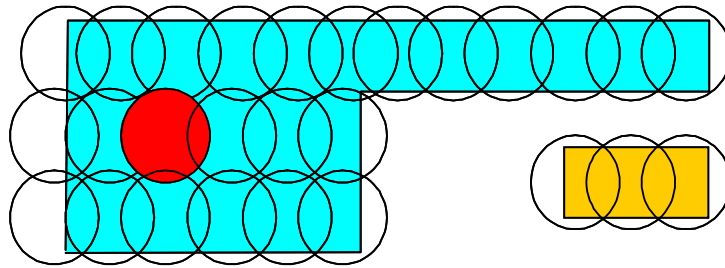


- 2a. Etapa – ART - determinar os domínios esféricos

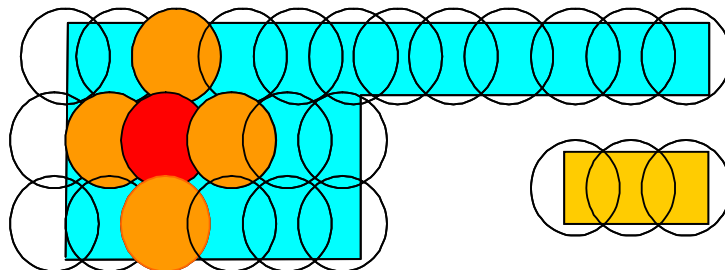


- 3a. Etapa – determinar os domínios vizinhos

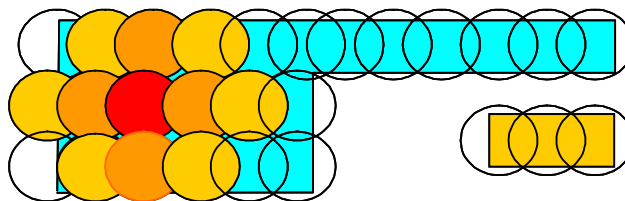
Centro de classe $\underline{w}_i$ escolhido arbitrariamente



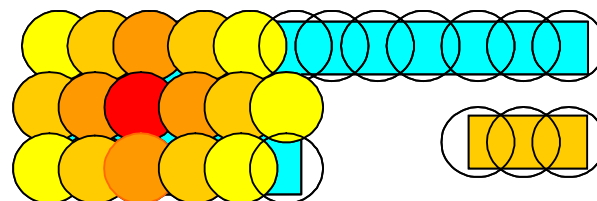
vizinhos $|\underline{w}_i - \underline{w}_j| \leq 2r_0$



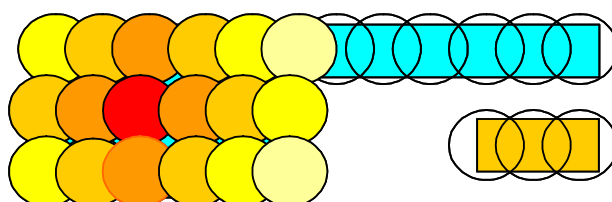
Vizinhos de vizinhos



Vizinhos de vizinhos de vizinhos



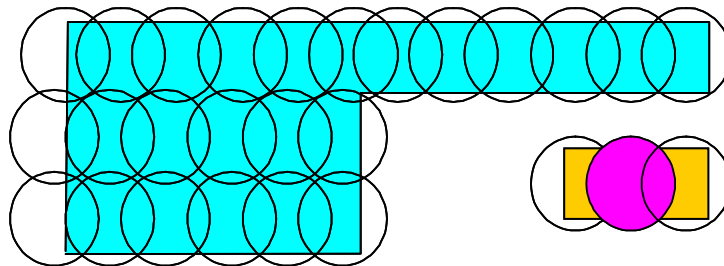
Vizinhos de vizinhos de ...



Todos pertencem à uma mesma classe.

Sobraram neurônios ?

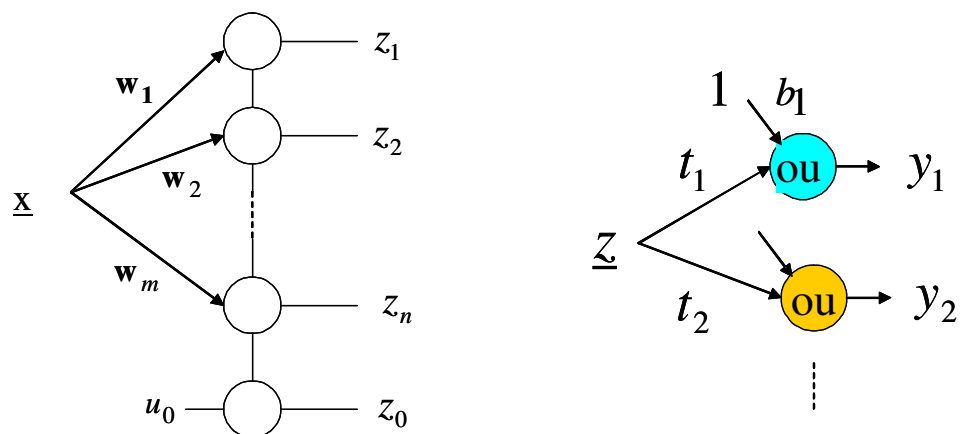
Recomeçar o processo, outra classe.



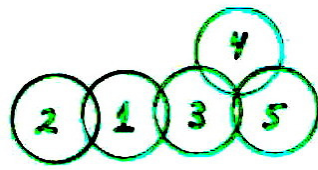
Não há mais neurônios ?

- **4a. Etapa – OU – conecta os domínios vizinhos**

Construir a segunda camada com um neurônio “ou” para cada classe

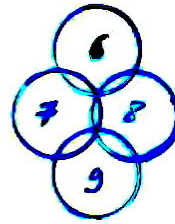


Exemplo 2



1	2	-	
	3	4	-
		5	-

$$C_1 = \{1,2,3,4,5\}$$



6	7	9	-
	8	-	

$$C_2 = \{6,7,8,9\}$$



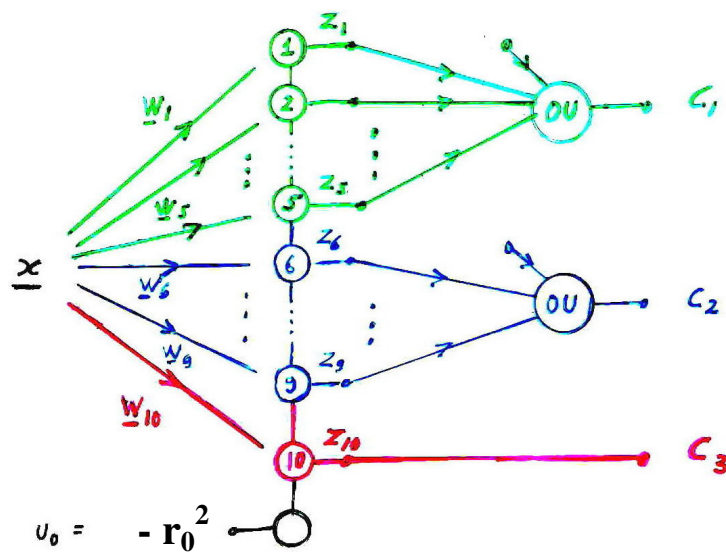
10	-

$$C_3 = \{10\}$$

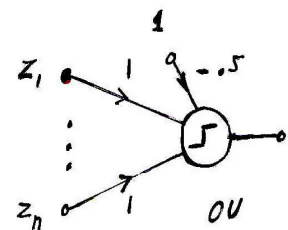
$$C_1 = \{1,2,3,4,5\}$$

$$C_2 = \{6,7,8,9\}$$

$$C_3 = \{10\}$$

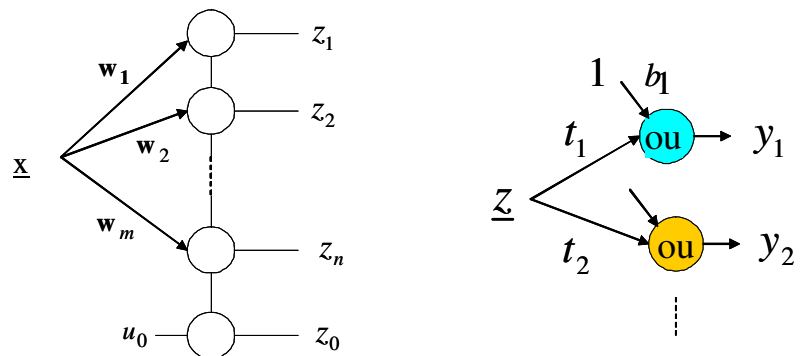


$$z_i \in \{0,1\}$$



Obs: Sendo um processo não supervisionado este método não consegue diferenciar classes cuja separação (a) é muito menor que a dimensão das classes ou (b) é da ordem de grandeza de r_0 .

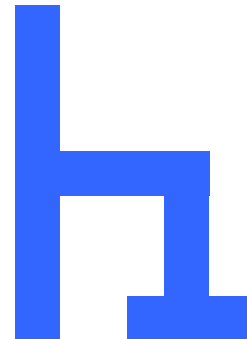
Demo:



Ex 6:



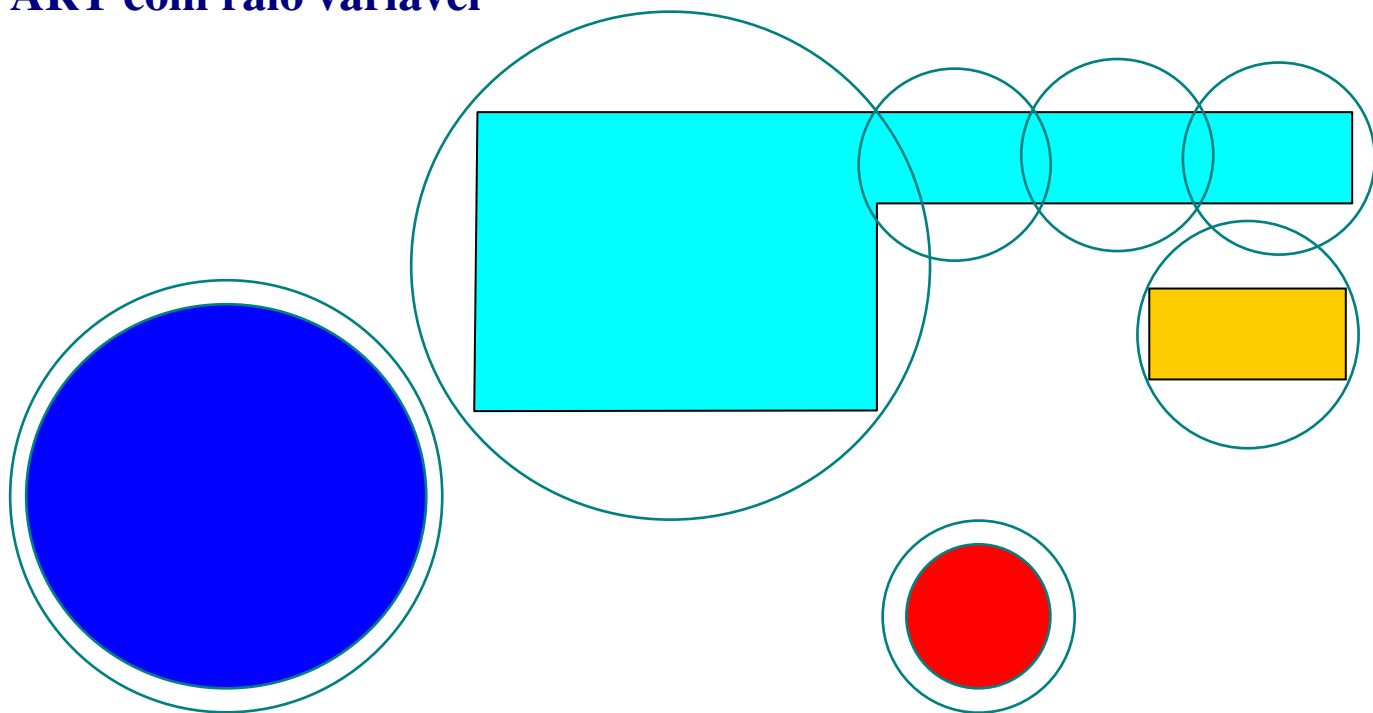
Ex 7:



Classes com dimensões variáveis



ART com raio variável



Raio de vigilância variável por neurônio

Condição de pertinência à classe C_i

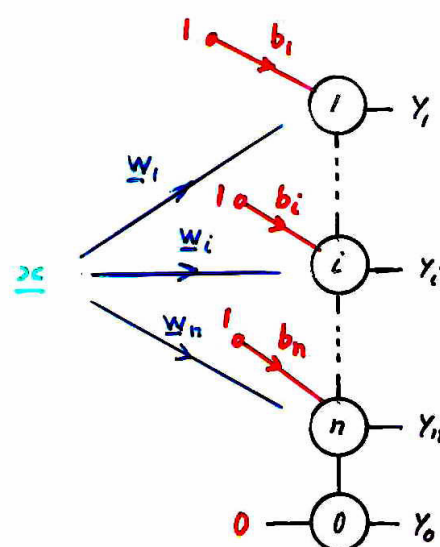
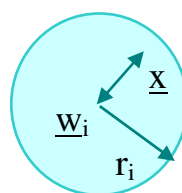
$$|\vec{x} - \vec{w}_i|^2 = d_i^2 < r_i^2$$

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

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

$$b_i = r_i^2$$

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



Caso dois neurônios apresentem $u > 0$

a entrada satisfaz a similaridade mínima para os dois padrões

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

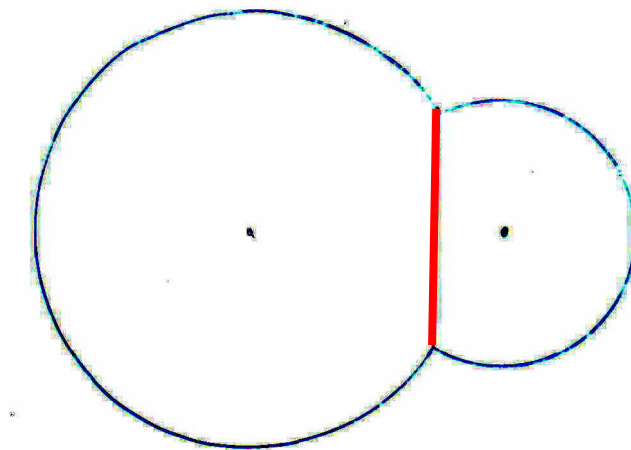
$$-d_i^2 + r_i^2 > -d_j^2 + r_j^2$$

Separador –

plano que contém

a circunferência

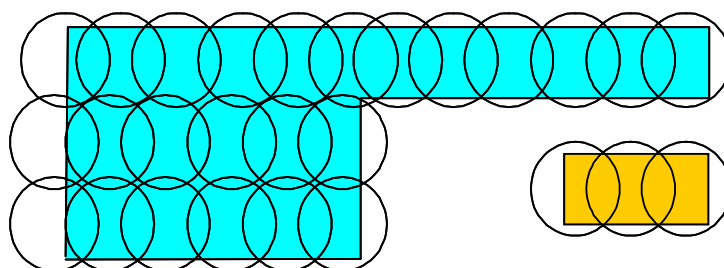
intersecção das esferas



Pré processamento para uso de esferas de raios variados

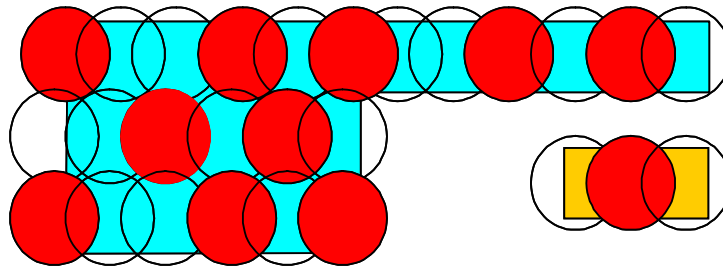
1. Determine os domínios utilizando esferas suficientemente pequenas.
Como no processo inicial para classes não esféricas.

2. Determine as classes e seus elementos



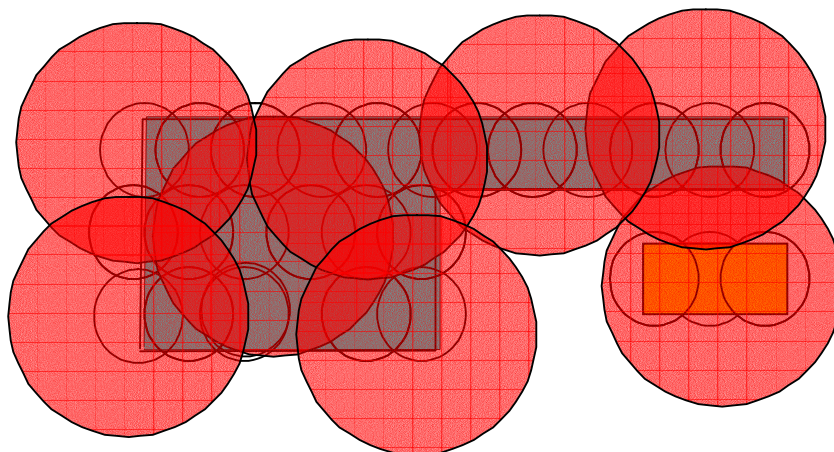
Processo aditivo

1. Escolha um conjunto de neurônios não vizinhos entre si tais que todos os neurônios são vizinhos de ao menos um neurônio do conjunto selecionado. Comece com os neurônios com maior número de vizinhos.

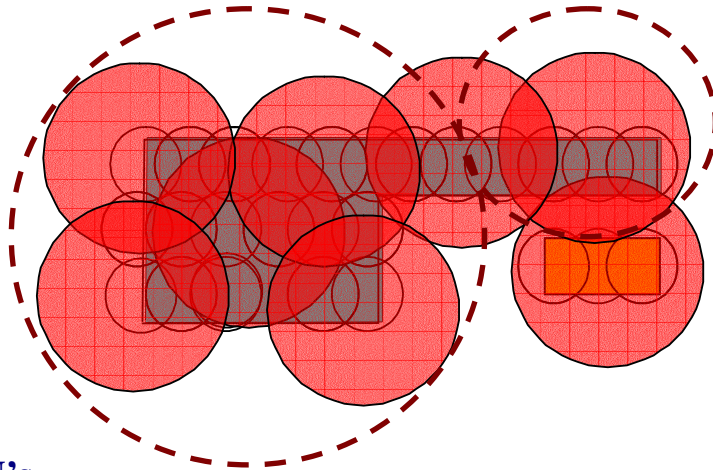


2. Para cada neurônio do conjunto selecionado e seus vizinhos recalcular o baricentro e o raio ou

- 2a. Alternativa mais simples: use o mesmo centro e triplique o raio



3. Verificar se não houve classificações erradas. Se sim, reduzir o raio do neurônio ganhador errado até que o neurônio correto ganhe.
4. Verificar se não houve elementos não classificados. Se sim, criar novos neurônios para classifica-los corretamente
5. Repetir o processo com os novos neurônios até que o número de classes esteja adequado.



6. Construir a outstar de OU's

Processo divisivo

1. Estimar o raio de cada classe (moda da distância entre elementos da classe)
2. Treinamento ART para cada classe usando seu raio
3. Verificar se houve “invasões”

SIM neurônios não invadidos - congelar $\underline{w}_i$ e r_i

neurônios invadidos - reduzir r_i , congelar $\underline{w}_i$ e r_i

verificar se há áreas “descobertas”

SIM - retornar a 2

NÃO- passo 4

NÃO passo 4

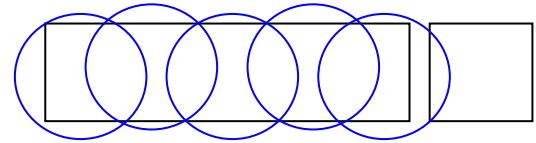
4. Construir a “outstar” de OU's

Classes

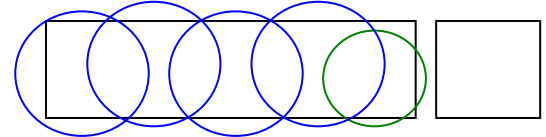


2 – Treinamento ART

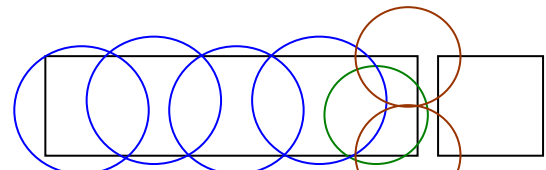
3 – Há “invasões”



3 – Redução de raio
Há áreas descobertas

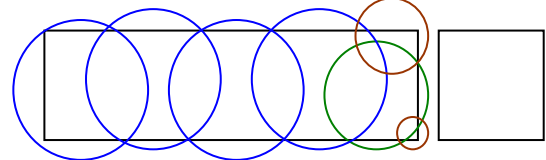


2 – ART novamente
São criados novos neurônios



3 - Há novas invasões

3 – Redução de raio
Não há áreas descobertas
Fim



A Speaker Verification Method Using LPC Singularity Location

Hardy L. C. P. Pinto

Rafael G. C. P. Pinto

Luiz P. Calôba

Representação LPC dos fonemas /a/ e /i/

Cinco locutores masculinos

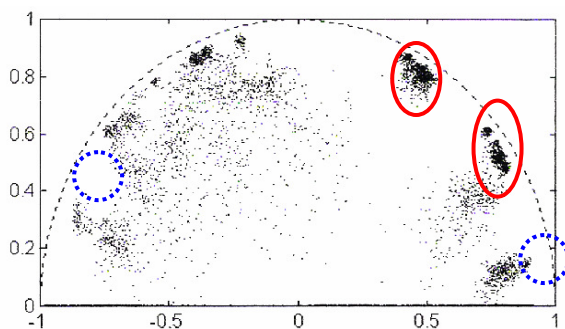


Figure 1- Roots of the predictor polynomial for segments of the sustained Portuguese phoneme /a/ for five different people of the same sex.

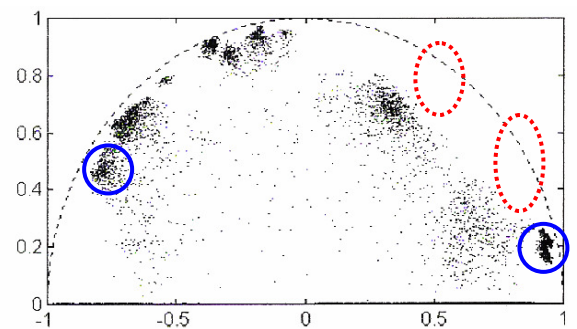


Figure 2- Same as fig. 1 for the phoneme /i/.

Diferenciação dos locutores

Fonema /a/ por dois locutores diferentes

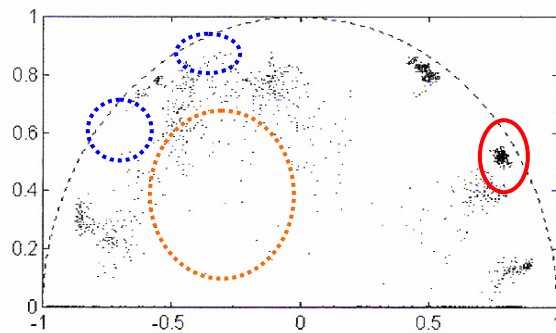


Figure 3 - Phoneme /a/ spoken by one speaker.

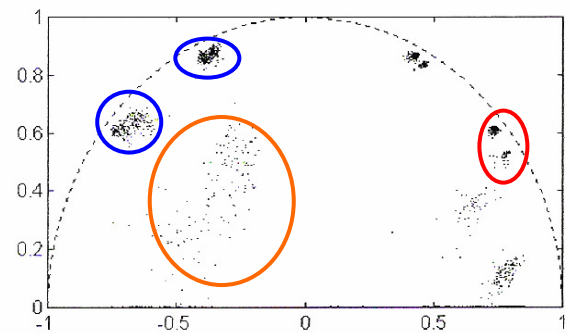


Figure 4 - Phoneme /a/ spoken by another speaker.

Rede ART modificada:

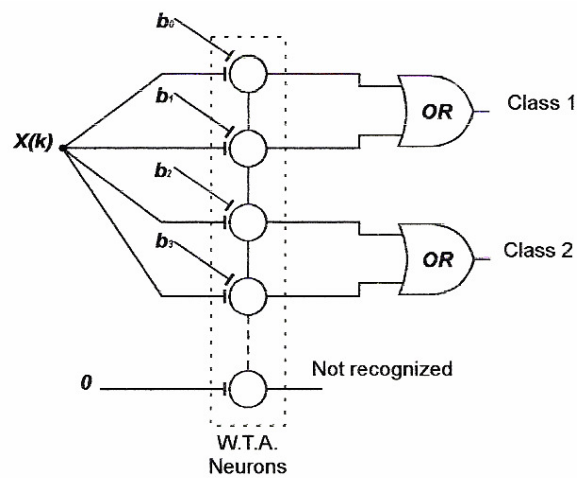


Figure 5 - Modified ART structure

Taxas de acerto de Fonemas:

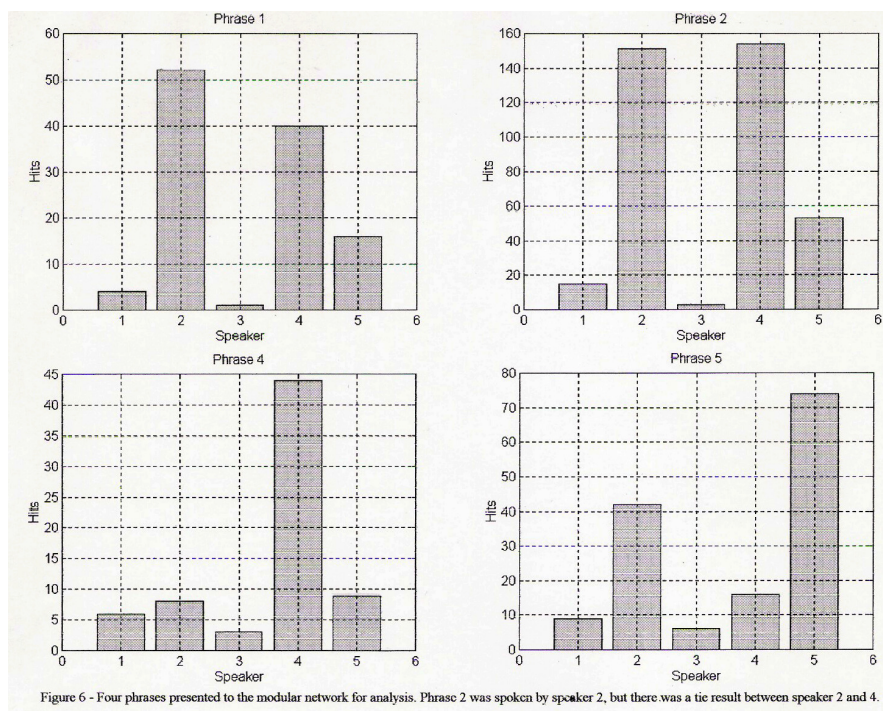
TABLE I A - PHONEME TEST SET

Phoneme	Total segments	Correct	Not recognized	Wrong classif.
/a/	548	98.91 %	0.00 %	1.09 %
/e/	600	99.17 %	0.33 %	0.50 %
/i/	551	99.64 %	0.18 %	0.18 %
/o/	601	97.84 %	0.67 %	1.49 %
/u/	549	97.45 %	1.09 %	1.46 %

TABLE I B - CONFUSION TABLE FOR THE PHONEMES

Spoken	Recognized				
	/a/	/e/	/i/	/o/	/u/
/a/	98.90 %	0.37 %	0.00 %	0.73 %	0.00 %
/e/	0.17 %	99.17 %	0.00 %	0.00 %	0.33 %
/i/	0.00 %	0.00 %	99.63 %	0.00 %	0.18 %
/o/	1.50 %	0.00 %	0.00 %	97.84 %	0.00 %
/u/	0.00 %	0.18 %	1.09 %	0.18 %	97.45 %

Votação para Locutores:



Taxa de acerto de locutores

TABLE II A - SPEAKER TEST SET (MODULAR TOPOLOGY)

Speaker	Total segments	Correct	Not recognized	Wrong classif.
2	750	75.20 %	7.87 %	16.93 %
3	749	91.32 %	3.20 %	5.48 %
4	598	70.23 %	11.71 %	18.06 %
5	752	82.71 %	6.52 %	10.77 %

TABLE II B - CONFUSION TABLE FOR SPEAKERS (MODULAR)

Speaker	Recognized Speaker				
	1	2	3	4	5
2	0.93 %	75.20 %	7.33 %	7.20 %	1.47 %
3	0.00 %	2.80 %	91.32 %	0.88 %	1.87 %
4	1.17 %	13.88 %	1.00 %	70.23 %	2.01 %
5	0.00 %	2.79 %	6.25 %	1.73 %	82.71 %

Os autores do trabalho:



Comparação **BP** vs **ART**

Velocidade de treinamento

ART / BP

Mínimos locais

ART

Velocidade de operação

ART / BP

Plasticidade

ART

Precisão

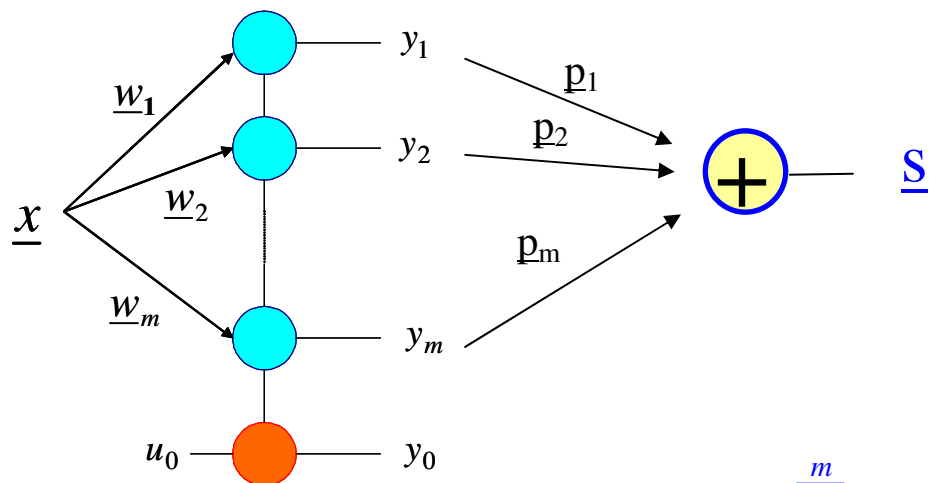
Classes esféricas

ART

Caso geral

BP

Counterpropagation (tipo feedforward)

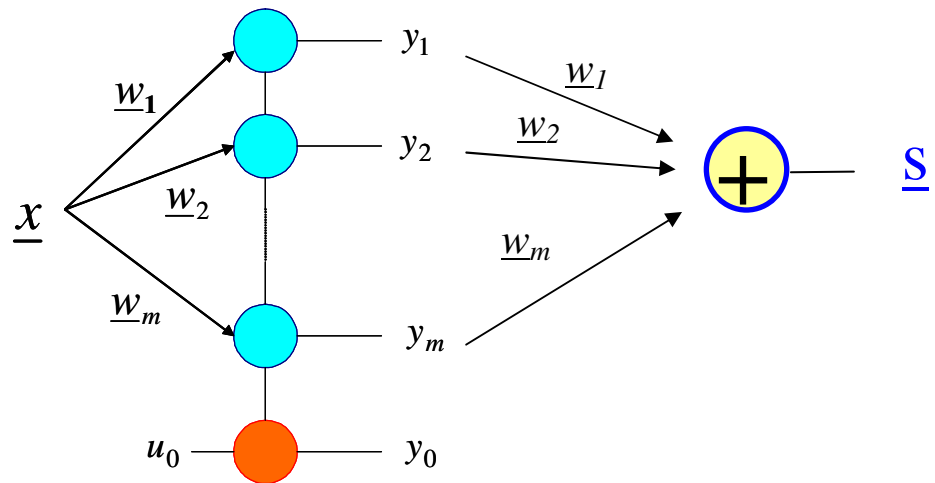


$$\underline{s} = \sum_{j=1}^m y_j \underline{p}_j$$

Memória hetero associativa

$$\underline{s} = \underline{p}_i \quad (y_i = 1)$$

Se $\underline{p}_i = \underline{w}_i$ $\underline{x} = \underline{w}_i + \underline{r}$ $\underline{s} = \underline{w}_i$



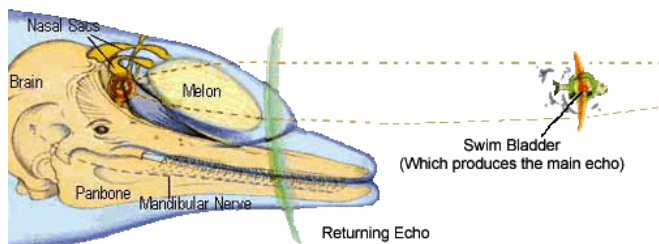
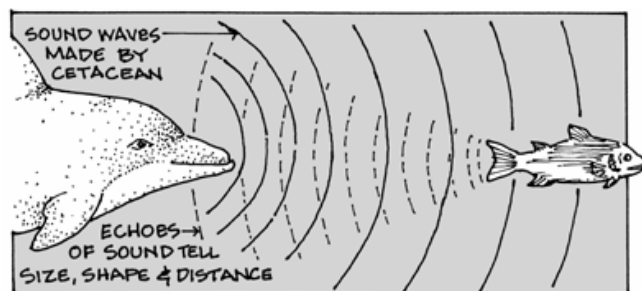
Memória endereçável por conteúdo

Memória auto associativa

Filtragem não linear

Detecção de padrões desconhecidos

Ex 1: Dolphin Ecolocation



Time to frequency patterns:

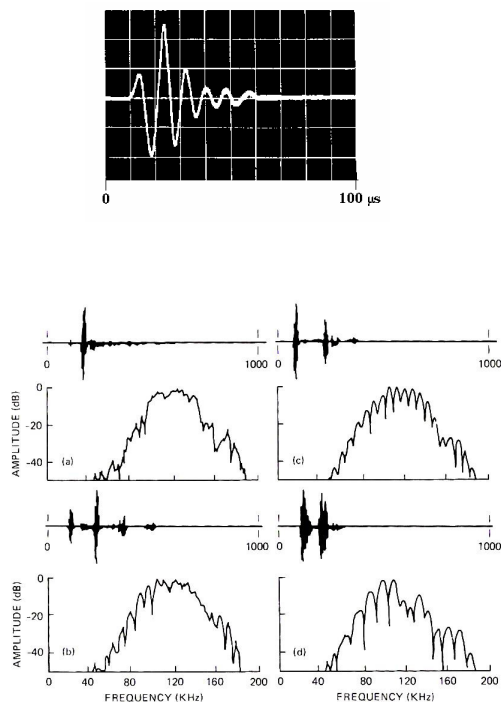
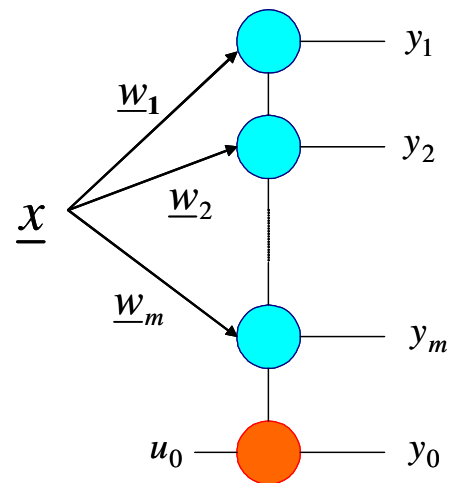
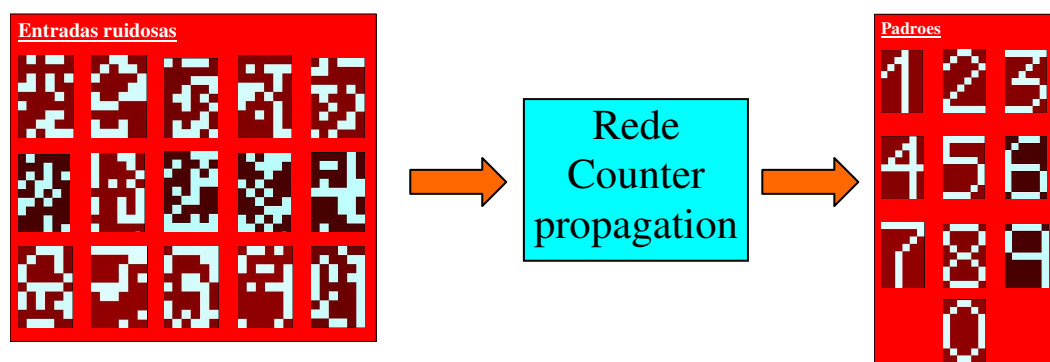


Figure 10-11. Amplitude display and Fourier transform for each item in the dolphin echolocation experiment. (From H. L. Roitblatt, et al. Dolphin Echolocation: Identification of returning echoes using a counterpropagation network. *Proc. IJCNN*. © 1989 IEEE.)



Ex 2: OCR: Figuras com ruído

Obtenções de padrões a partir de sinais ruidosos:



Compressão de informações

Compressão de imagens

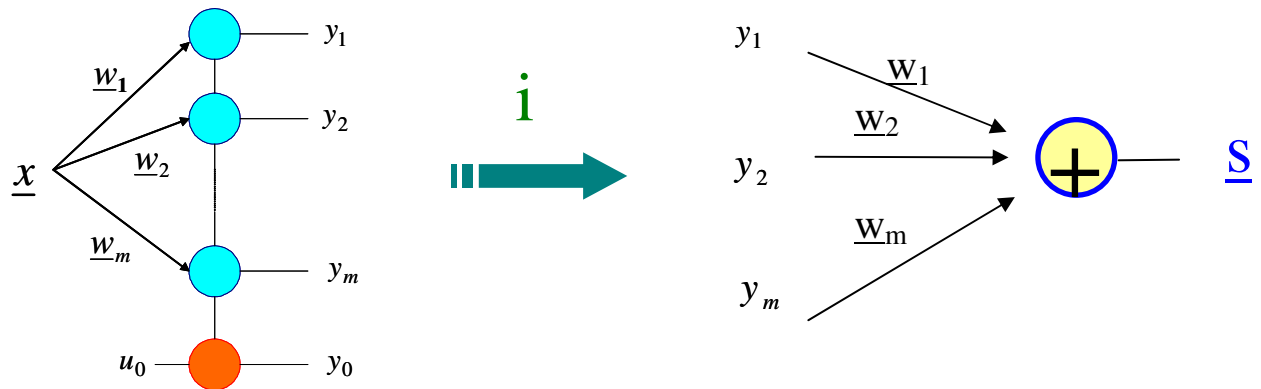
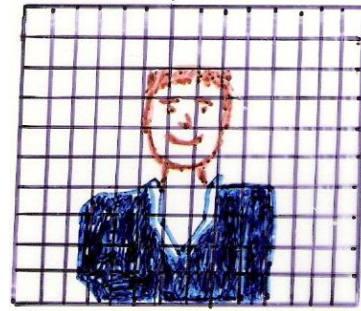


imagem P&B

BMP

1 quadro 4x4 pix = 16 pix

16 pix/quadro 8 bits/pix = 128 bits/quadro

Compressao RN

1 quadro 4x4 pix

8 padroes/quadro 3 bits/padrao = 24 bits/quadro

Taxa de compressao: $128/24 = 5.3 \times$

Exemplo de Compressão

Imagens original e comprimidas

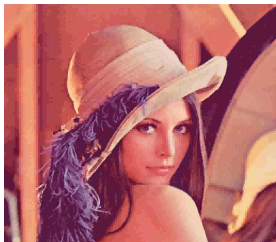
(1x)



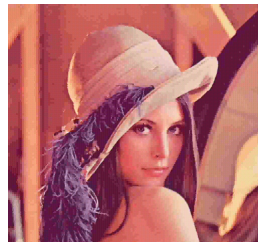
(11x)



(15x)



(28x)



Efeito do codebook (compressão aprox 20x)

codebook Lena



codebook house

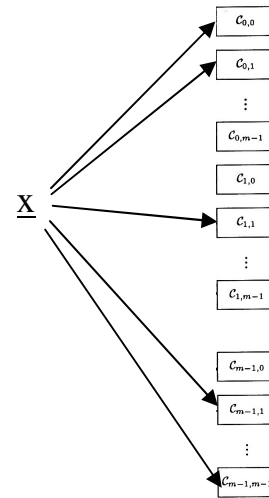


Operação com grande número de padrões

Classificador “clássico”

$N = \text{padrões}$ $\Rightarrow$

N comparações

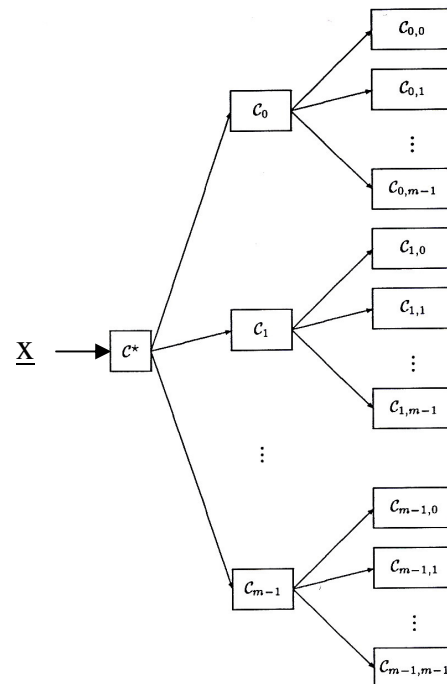


Solução

Classificador hierárquico,

Classificador em árvore,

Dendograma,



1ª camada m comparadores

2ª camada $m \cdot n$ comparadores

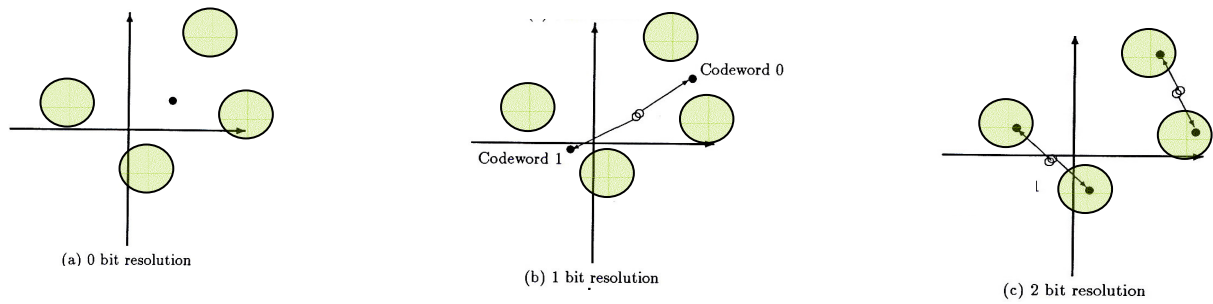
padrões = $N = m \cdot n$

comparações = $m + n$

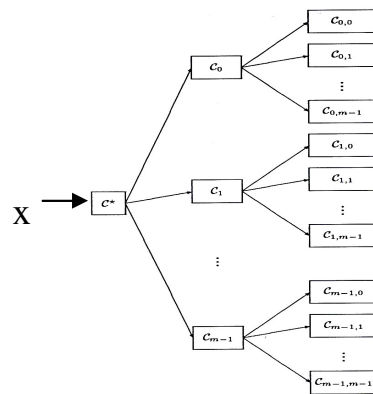
mais camadas são possíveis

Figure 12.1: Basic Structure of Tree-Structured VQ Encoding

Construção via K-means



o conjunto de treinamento
é dividido



Construção via ART

o raio de similaridade
diminui por camada

o conjunto de treinamento
é dividido

