

Pós-processamento dos resultados

1 – Crítica dos Resultados - pós treinamento – **indispensável**

Geral

Classes k com grande $F_{in}(k)$,

ou grande dispersão $\sigma_k = \sqrt{\frac{F_{in}(k)}{n_k}}$ devem ser bipartidas

Classes com baricentros muito próximos devem ser agrupadas
comparar a distância entre baricentros com σ_k de cada classe

Verificar o “agrupamento natural” pela evolução de $F_{in}(N)$

Específico

depende do processo usado.

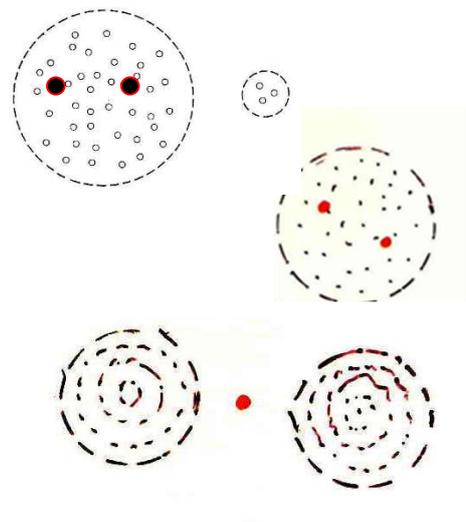
Para ART e Camada de Kohonen

a - Neurônios não (**ou pouco**) treinados - eliminar

b – Classes com baixa população

c - Dois neurônios partilham a mesma classe

d – Um neurônio atende duas classes



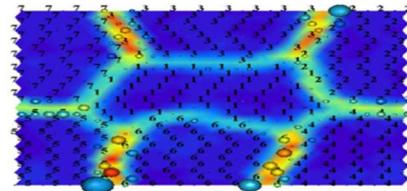
Para o SOM

Identificação da classe dos neurônios

Supervisionado - classes com mais acionamentos

vizinhos mais próximos

Não supervisionados – Distância aos vizinhos mais próximos



2 – Apresentação dos Resultados

2.1 Resultado da Classificação

Classe x Não Classe

		Resultado do Classificador	
		Pertence à classe	Não pertence à classe
Classificação verdadeira	Pertence à classe	VP Verdadeiros Positivos	FN Falsos Negativos
	Não pertence à classe	FP Falsos Positivos	VN Verdadeiros Negativos

Múltiplas classes - Tabela de Confusão

Classes corretas	Resultado da classificação					
	C_1	C_2	...	C_i	...	C_N
C_1						
C_2						
...						
C_j				a_{ij}		
...						
C_N						

Classes corretas	Resultado da classificação					
	C_1	C_2	...	C_j	...	C_N
C_1	VP			FP		
C_2		VP		FP		
...			VP	FP		
C_j	FN	FN	FN	VP	FN	FN
...				FP	VP	
C_N				FP		VP

a_{ij} - % de elementos de C_j classificados como C_i

Classificação em relação à C_j

3 - Parâmetros de qualidade do classificador

Critério usualmente otimizado:

Erro de classificação
$$e_{class} = \frac{FP + FN}{VP + VN + FP + FN}$$

Outros critérios de qualidade

Taxa de acerto
$$TA = \frac{VP + VN}{VP + VN + FP + FN} = 1 - e_{class}$$

Sensibilidade
$$S = \frac{VP}{VP + FN}$$

Especificidade
$$E = \frac{VN}{VN + FP}$$

Valor preditivo positivo
$$VPP = \frac{VP}{VP + FP}$$

Valor preditivo negativo
$$VPN = \frac{VN}{VN + FN}$$

Falsos Alarmes
$$FA = \frac{FP}{VN + FP} = 1 - E$$

Falsas Perdas
$$FP = \frac{FN}{VP + FN} = 1 - S$$

4 – Análise dos erros

Anomalias

Intrusos, *outliers* - eliminar

Baixa população

Balancear as populações (replicar pares) ou
Ponderar os erros

Mapeamentos complexos

Inserção de mais neurônios **na região**
vs overtraining

Efeito de fronteira / fusão (1° vs 2° ganhador) (u_1 vs u_2)

5 – Conjuntos de Treinamento e Teste

Generalização da Classificação – evitando overtraining

Conjunto dos pares entrada saída >>>

Conjuntos de	Treinamento	~ 70 %
	Teste	~ 30%

6 - Validação Cruzada – validação do processo

baixa estatística

Conjunto dos Pares entrada saída >>>

Conjunto A	20%
Conjunto B	20%
Conjunto C	20%
Conjunto D	20%
Conjunto E	20%

Composição dos Conjuntos de Treino e Teste:

Treinamento 60%	Teste 20%
A+B+C	D+E
A+B+D	C+E
A+B+E	C+D
A+C+D	B+E
A+C+E	B+D
...	...

Treinamento supervisionado ?

Verificar média e dispersão do erro nos conjuntos de treinamento e teste.

Treinamento não supervisionado

Verificar coerência das classificações nos conjuntos de treinamento e teste.

Centros de classe ~ iguais ?

7 – Relevância das entradas

em um classificador uma entrada é muito relevante se sua ausência provoca um percentual elevado de modificações nas saídas (na classificação) para o conjunto das P entradas disponíveis.

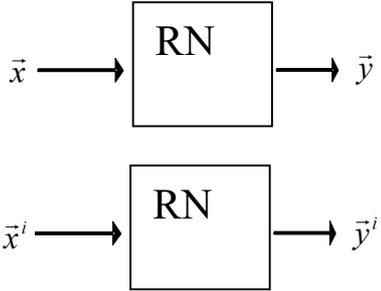
relevância de uma entrada x_i em relação à saída y_j (à classe C_j)

$$\Delta x_i \Rightarrow \Delta y_j \begin{cases} \Delta y_j \gg & \text{muito relevante} \\ \Delta y_j \ll & \text{pouco relevante} \end{cases}$$

7.1 - Medida da relevância

$$\text{considere } \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_i \\ \dots \\ x_n \end{bmatrix} \text{ e } \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ \mu(x_i) \\ \dots \\ x_n \end{bmatrix} \text{ onde } \mu(x_i) = \begin{cases} 0 & \text{se } x_i \text{ é variável contínua} \\ 0,5 & \text{se } x_i \in \{0,1\} \text{ é variável lógica} \end{cases}$$

aplicando \bar{x} e \bar{x}^i na rede obteremos respectivamente \bar{y} e \bar{y}^i

$$\text{onde } \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_j \\ \dots \\ y_m \end{bmatrix} \text{ e } \begin{bmatrix} y_1^i \\ y_2^i \\ \dots \\ y_j^i \\ \dots \\ y_m^i \end{bmatrix}$$


realizando esta operação para P entradas,

$$r(x_i, y_j) = \frac{1}{2P} \sum_{\forall P \text{ entradas}} (y_j - y_j^i)^2$$

$r(x_i, y_j)$ é a relevância de x_i em relação à y_j e mede o percentual de vezes em que a classe C_j é afetada pela eliminação da entrada x_i

$$r(x_i) = \sum_{j=1}^m r(x_i, y_j) = \frac{1}{2P} \sum_{\forall P \text{ entradas}} \sum_{j=1}^m (y_j - y_j^i)^2$$

$r(x_i)$ é a relevância total de x_i em relação à classificação, e mede o percentual de vezes em que a classificação é afetada pela eliminação da entrada x_i .

Obs: o fator 2 em $r(\cdot)$ é porque devido ao winner-takes-all mudam 0 ou 2 saídas.

7.2 – Estimativa da relevância das entradas pela diferença de valor das respectivas sinapses

$$\underline{x}_i \in C_i \quad \text{sse}$$

$$u_i = -|\underline{x} - \underline{w}_i|^2 > u_j = -|\underline{x} - \underline{w}_j|^2 \quad \forall j \neq i \quad \Rightarrow$$

$$\sum_{l=1}^n (x_l - w_{li})^2 < \sum_{l=1}^n (x_l - w_{lj})^2 \quad \forall j \neq i$$

$$r_l = E_{\forall i, j} |w_{li} - w_{lj}|$$

7.3 - Poda de Entradas

- **eliminar as entradas com baixa relevância**
- **retreinar a rede podada.**

8 – Redimensionar a rede

- **aglutinar classes com baricentros muito próximos**
- **dividir classes com muitos erros (hierárquico)**