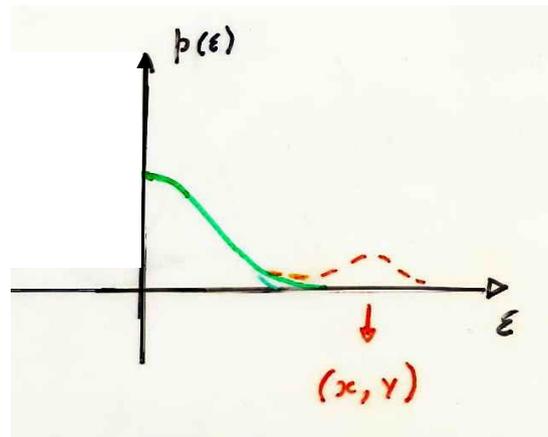


Processos Pós-treinamento

1 - Histograma dos erros (mandatório)

$$(\underline{x}, \underline{y}) \longrightarrow \underline{\tilde{y}} \longrightarrow \varepsilon = |\underline{y} - \underline{\tilde{y}}|$$

$\varepsilon >$ valor esperado \longrightarrow **Provável anomalia no par $(\underline{x}, \underline{y})$**



Examinar conjuntos de treinamento, teste e validação

Anomalias possíveis:

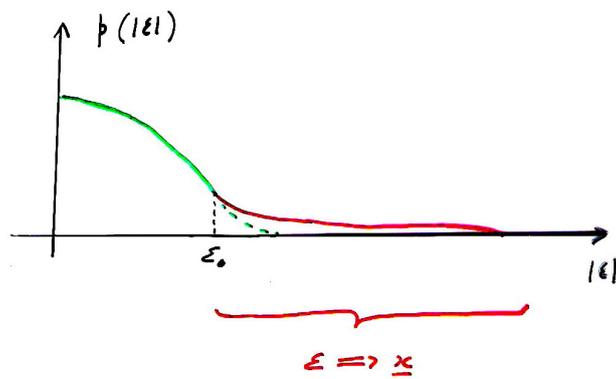
Intrusos,

Regiões de baixa população

Mapeamento localmente complexos

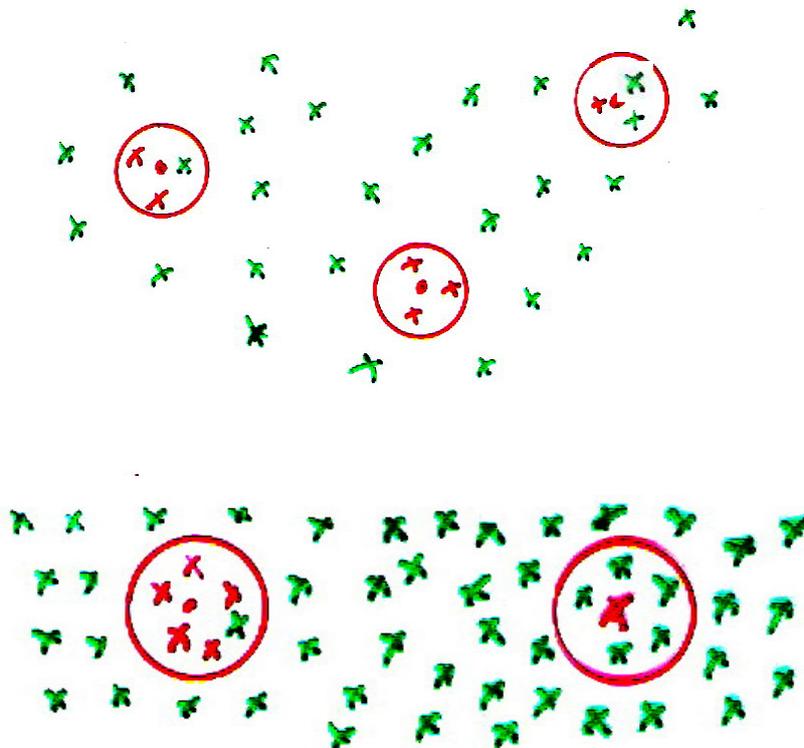
Localização das anomalias

$$(\underline{x}, y) \Rightarrow (\underline{x}, y, \bar{y}, \epsilon) \Rightarrow (\underline{x}, \epsilon)$$



$$|\epsilon| > \epsilon_0 \quad \underline{x} \in X_0$$

$$\text{Clusterizer} \left[\begin{array}{c} \underline{x} \\ \underline{y} \end{array} \right] \quad \forall \underline{x} \in X_0$$



Correção das anomalias

Análise das anomalias

Intrusos - eliminar

Baixa população

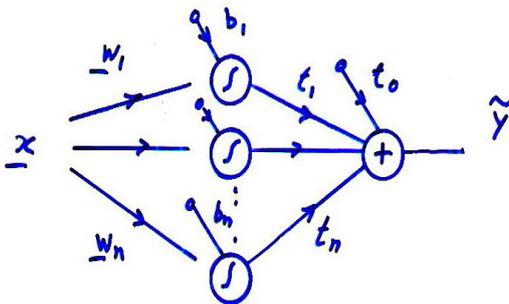
Balancar as populações (replicar pares) ou Ponderar os erros

Mapeamentos complexos

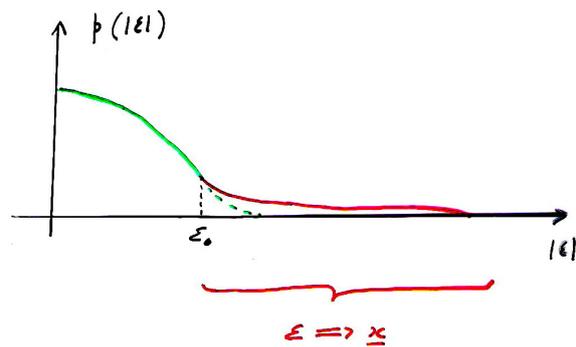
Inserção de mais neurônios na região (neurônios RBF ou tgh com $\underline{w} \sim \underline{x}_{\text{crítico}}$)

2 – Correção de erros - em pós processamento

Correção de erros em redes BP ou RBF



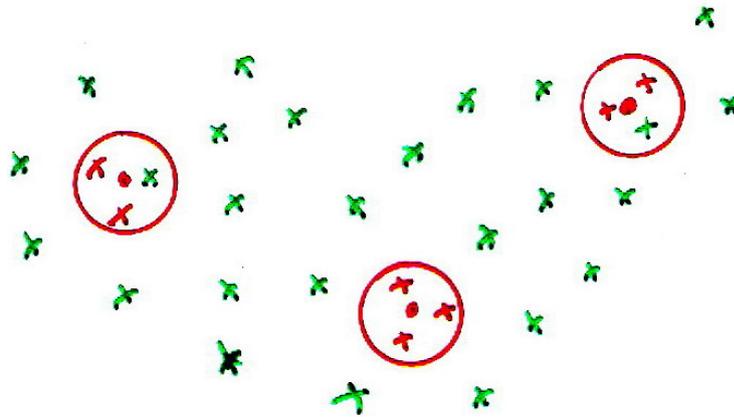
$$(\underline{x}, y) \Rightarrow (\underline{x}, y, \tilde{y}, \epsilon) \Rightarrow (\underline{x}, \epsilon)$$



$$|\epsilon| > \epsilon_0 \quad \underline{x} \in X_0$$

$$\text{Clusterizer } \underline{x} \in X_0$$

Clusterizar todo $\begin{bmatrix} \underline{x} \\ \underline{y} \end{bmatrix}$ tal que $\epsilon > \epsilon_0$



Identificar **toda** a população das regiões de grandes erros, inclusive os pares com pequenos erros

(identificar e eliminar intrusos, se houver)

Rede aumentada

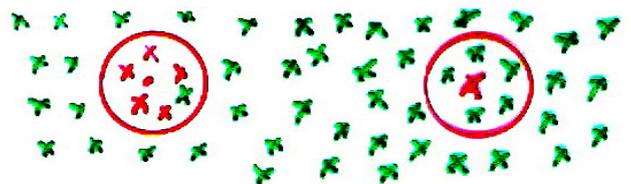
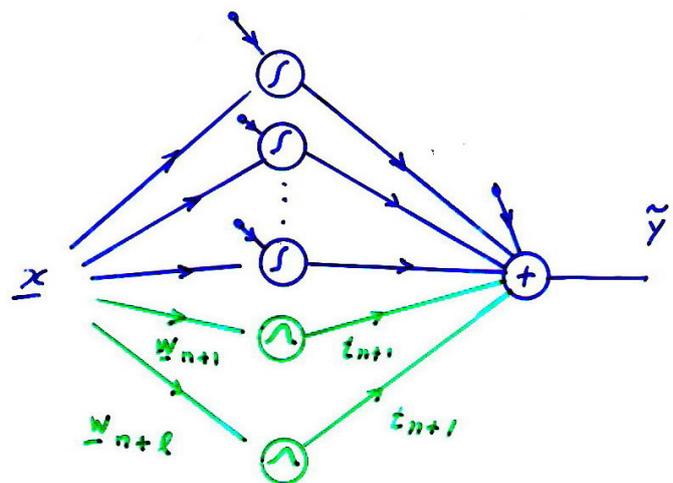
Valores iniciais dos neurônios RBF

\underline{w} – centros das classes de erro

σ – 0,5 diâmetro das classes de erro

Treinamento por BP

Iniciar com os pares $(\underline{x}, \underline{y})$ que compõem as classes de erro elevado e seus vizinhos (toda a população das regiões de grandes erros)



O processo pode ser aplicado em qualquer tipo de rede, “BP”, RBF ou outros.

3 - Pós Treinamento - Relevância das entradas

Relevância da entrada x_i , $r(x_i)$

$$\Delta x_i \implies \Delta \tilde{y}_j$$

Grande muito relevante
Pequeno pouco relevante

3.1 Variação do erro na saída

Erro na saída (rede treinada com todas as entradas) E_0

Erro na saída (rede re-treinada sem a entrada x_i) E_{-x_i}

3.2 Sem retreinamento

Alternativa mais simples, sem retreinamento da rede

$$\begin{array}{ccc}
 \begin{array}{c} x_1 \\ x_2 \\ \dots \\ x_i \\ \dots \\ x_n \end{array} & \implies & \begin{array}{c} \tilde{y}_1 \\ \tilde{y}_2 \\ \dots \\ \tilde{y}_j \\ \dots \\ \tilde{y}_n \end{array} \\
 \underline{x} & & \underline{\tilde{y}} = \varphi(\underline{x}) \\
 \\
 \begin{array}{c} x_1 \\ x_2 \\ \dots \\ \mu(x_i) \\ \dots \\ x_n \end{array} & \implies & \begin{array}{c} \tilde{y}_{1i} \\ \tilde{y}_{2i} \\ \dots \\ \tilde{y}_{ji} \\ \dots \\ \tilde{y}_{ni} \end{array} \\
 \underline{x}_i & & \underline{\tilde{y}}_i = \varphi(\underline{x}_i)
 \end{array}$$

usualmente $\mu(x_i) = 0$

Relevância de x_i

$$r(x_i) = E(y - \tilde{y}_i)^2 - E(y - \tilde{y})^2$$

3.3 – Estimativa da relevância pela variação das sinapses da entrada

Entradas não relevantes presentes

$$\Delta_p w_{ij} = \alpha v_j \delta_i$$

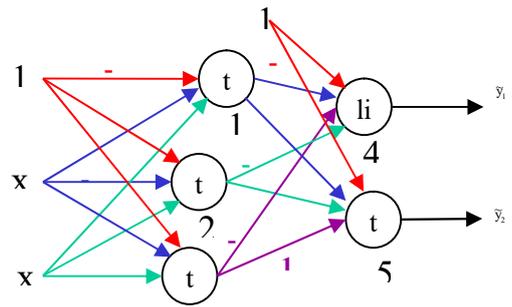
$$E(\Delta w_{ij}) = E_p(\Delta_p w_{ij}) = \alpha E_p(v_j \delta_i) = 0$$

porque $r(v_j \delta_i) = 0$

então $E(w_{ij}) = w_{ij\text{ inicial}}$ **não treinam**

$$r(x_i) = \sum_{j=1}^N (w_{ji\text{ inicial}} - w_{ji\text{ final}})^2$$

Σ para todas as sinapses conectadas à entrada x_i



3.4 - Poda de Entradas

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

Aplicações - Relevância

Classificação de Navios pela assinatura magnética



Banco de dados

FEUP-IPB >>>>> textos extraídos de jornais,
lidos por um locutor profissional.

21 minutos de fala >>>>> 18700 segmentos >>>>>

13.700 vetores para treino

3000 vetores para teste e

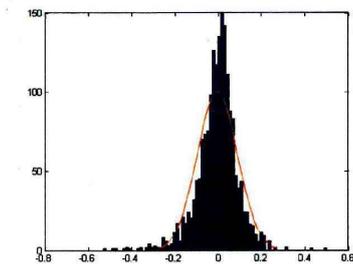
2000 vetores para validação.

etiquetados semi-automaticamente

Configuração da rede:

10 neurônios tipo tangente hiperbólica na camada intermediária,
um neurônio linear na última camada
treinamento backpropagation.

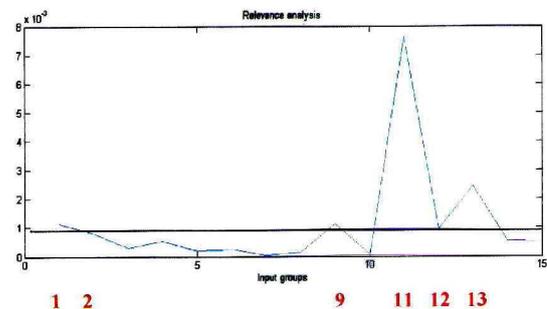
Erro



Erro: média = 0.9 ms

desvio padrão = 23.1 ms

Relevância das entradas:



Entradas mais relevantes: 1, 2, 9, 11, 12, 13

Comparação dos resultados:

0 - Processo manual, 15 entradas.

1 - Rede neural, configuração com 15 entradas.

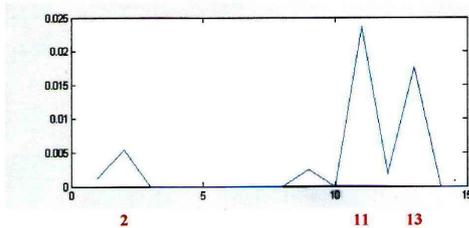
2 - Rede neural: configuração com 15 entradas mas com as entradas menos relevantes substituídas pelas respectivas média, **sem retreinamento**.

3 - Rede neural: configuração com apenas as 6 as entradas mais relevantes, **com retreinamento**.

Caso	Configuração	Média	Desvio padrão
0	Processo manual, 15 entradas		35
1	Rede neural, 15 entradas	0.9	23.1
2	Rede neural original com apenas as 6 entradas mais relevantes, sem retreino.	-10.9	24.5
3	Rede neural com apenas as 6 entradas mais relevantes, com retreino	1.0	23.6

Repetindo o processo:

Relevância das entradas:



Entradas mais relevantes: **2, 11, 13**

Comparação dos resultados:

1 - Configuração original, **15** entradas.

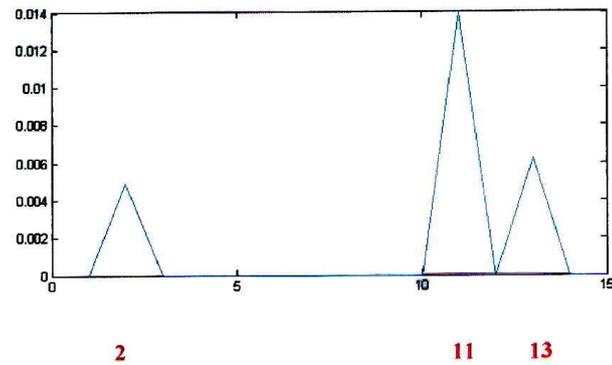
3 - Rede neural: configuração com apenas as **6** as entradas mais relevantes, com retreinamento.

4 - Rede neural: configuração com apenas as **3** as entradas mais relevantes, com retreinamento.

Caso	Configuração	Média	Desvio padrão
1	Rede neural, 15 entradas.	0.9	23.1
3	Rede neural com apenas as 6 entradas mais relevantes.	1.0	23.6
4	Rede neural com apenas as 3 entradas mais relevantes.	1.6	24.1

Repetindo o processo mais uma vez:

Relevância das entradas:



Entradas mais relevantes:

continuam as mesmas 2, 11, 13

Processo de poda encerrado.