

## 8 – Treinamento:

### 8.1 Superfícies de Erro

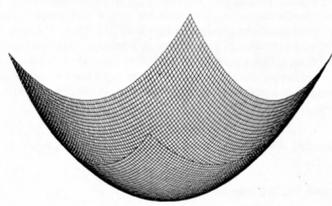


Fig. 22. Example MSE surface of linear error.



Fig. 23. Example MSE surface of sigmoid error.

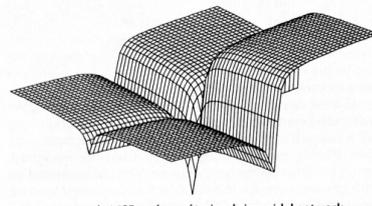
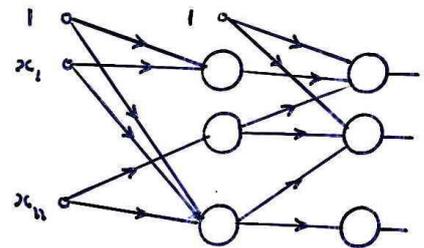


Fig. 30. Example MSE surface of trained sigmoidal network as a function of two first-layer weights.

**Mínimos locais possíveis ? SIM !**

**Vários mínimos globais**

**Comparação de soluções:  
somente via erro**



### 8.2 - Evolução do Erro durante o treinamento

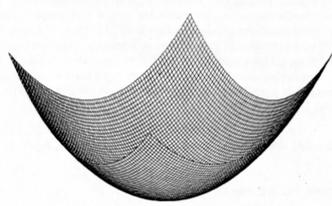


Fig. 22. Example MSE surface of linear error.



Fig. 23. Example MSE surface of sigmoid error.

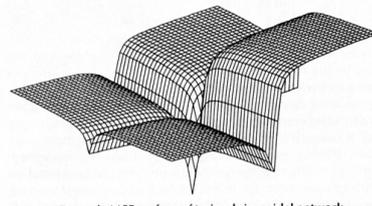
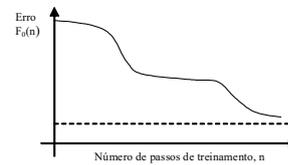
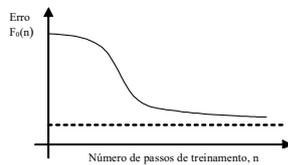
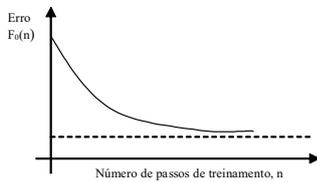
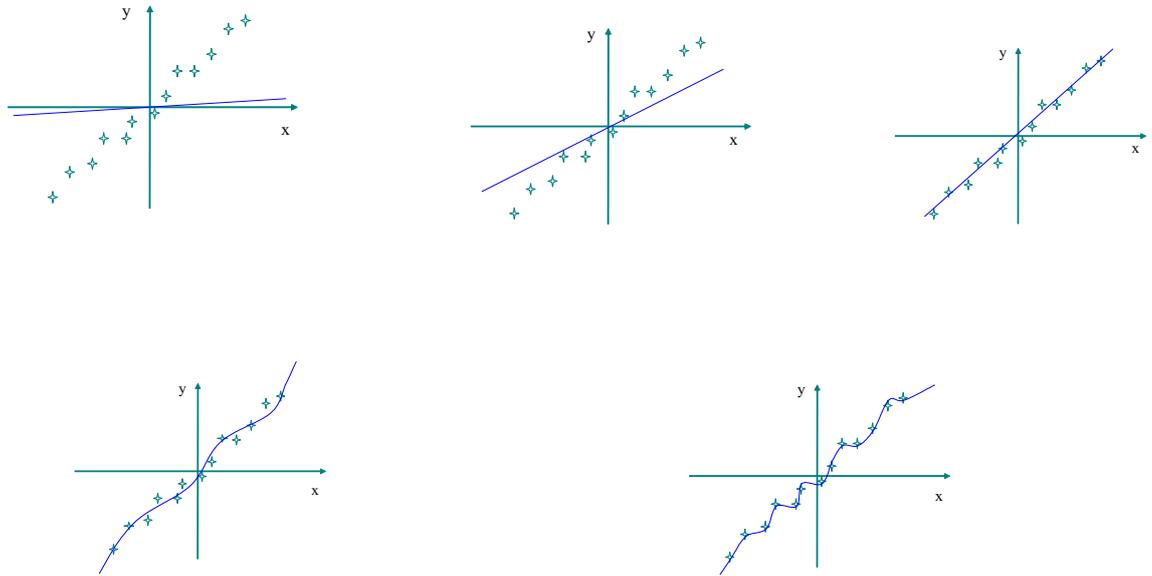


Fig. 30. Example MSE surface of trained sigmoidal network as a function of two first-layer weights.



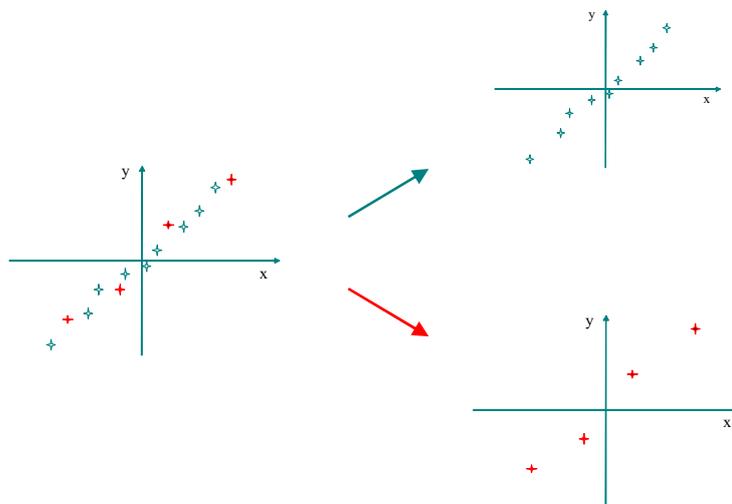
**Crítica (acompanhamento) durante o treinamento !**

### 8.3 – Overtraining (overfitting, sobre-teinamento)

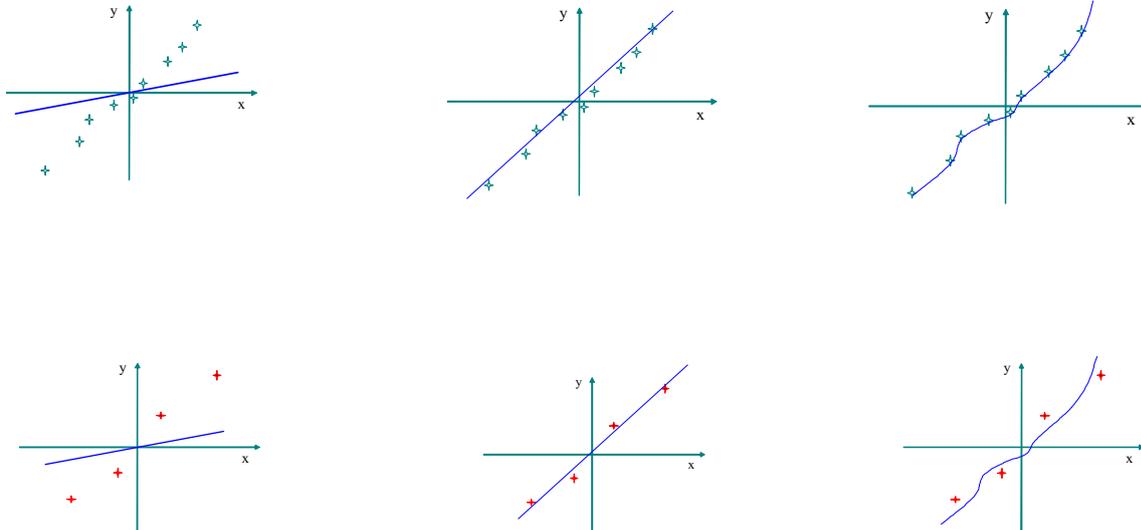


Conjunto dos pares entrada saída >>>

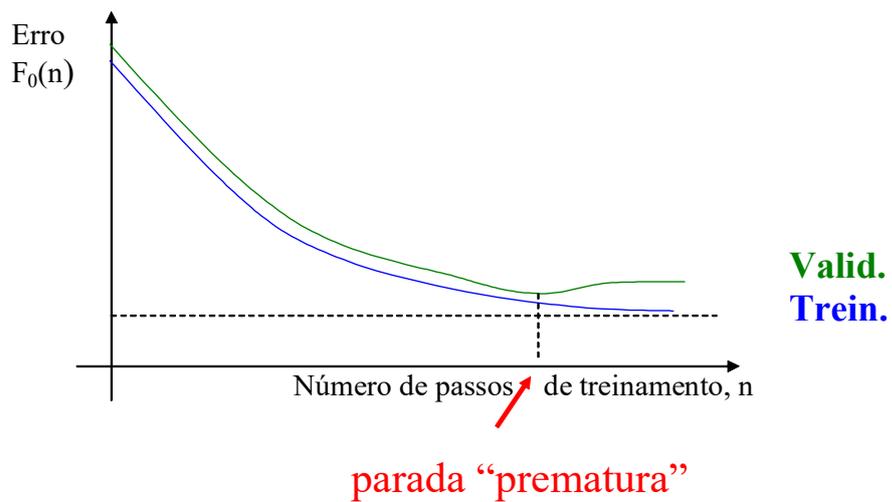
<b>Conjuntos de</b>	<b>Treinamento</b>	<b>60%</b>
	<b>Validação</b>	<b>20%</b>
	<b>(Teste</b>	<b>20%)</b>

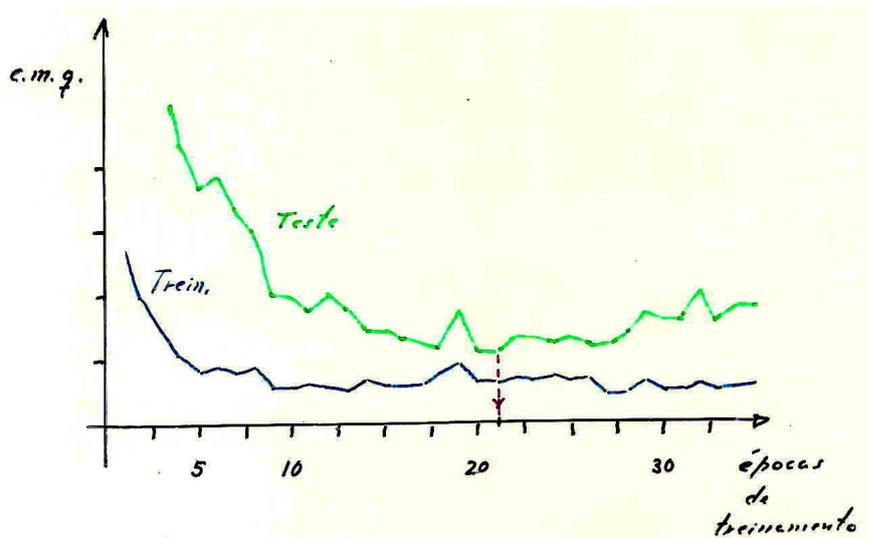


## Evolução dos erros ao longo do treinamento nos conjuntos de treinamento e de teste



## Controle do overtraining





“Save the Best”  $\epsilon_{\text{teste}}^*$   $\underline{w}^*$

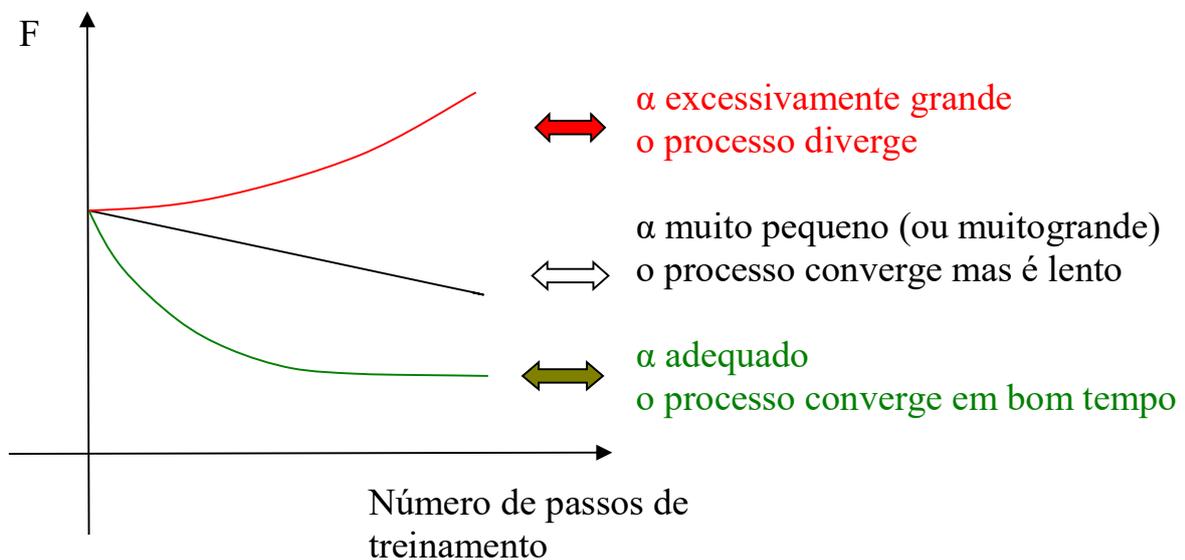
se para  $\underline{w} \neq \underline{w}^*$   $\epsilon_{\text{teste}} < \epsilon_{\text{teste}}^*$

faça  $\underline{w}^* = \underline{w}$  e  $\epsilon_{\text{teste}}^* = \epsilon_{\text{teste}}$

### 8.4 – Crítica durante o treinamento

Acompanhamento da evolução do erro nos conjuntos de treinamento, validação e teste durante o processo de treinamento

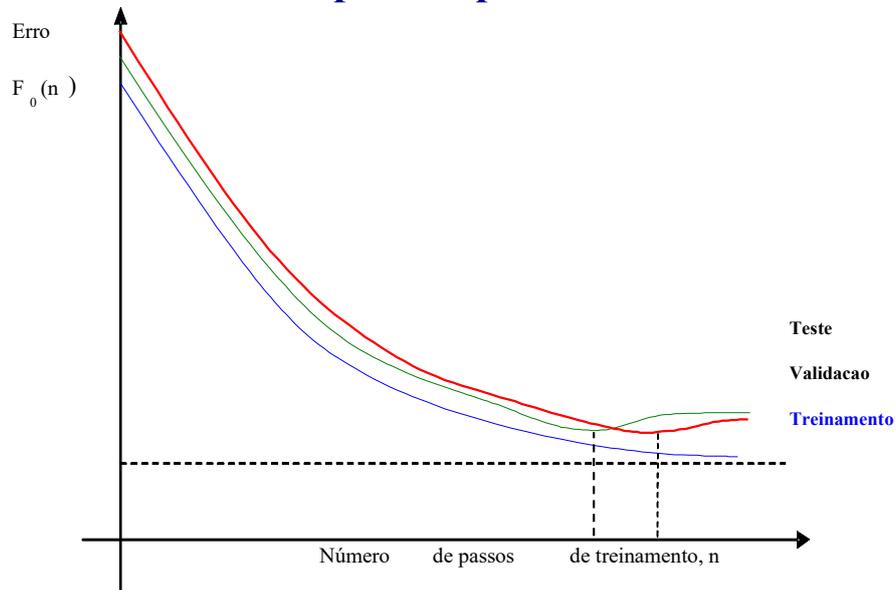
no início do treinamento: avaliação de  $\alpha$



**durante o treinamento:**

**controle do overtraining,**

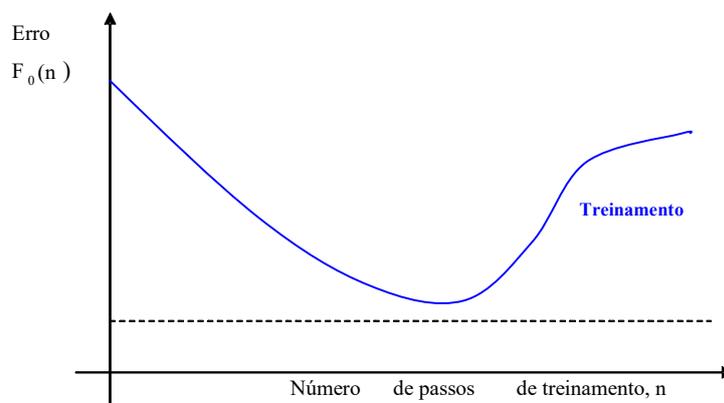
**parada prematura**



**Parada “prematura”: duas soluções**

**durante o treinamento:**

**controle de possíveis divergências**



**“Divergência tardia”**

## Que tipo(s) de erro(s) acompanhar ?

Saídas contínuas – **Erro eficaz (erms)** ou **erro médio quadrático (emq)**

(erro na variável medida) versus (erro matemático)

$$erms = \sqrt{emq}$$

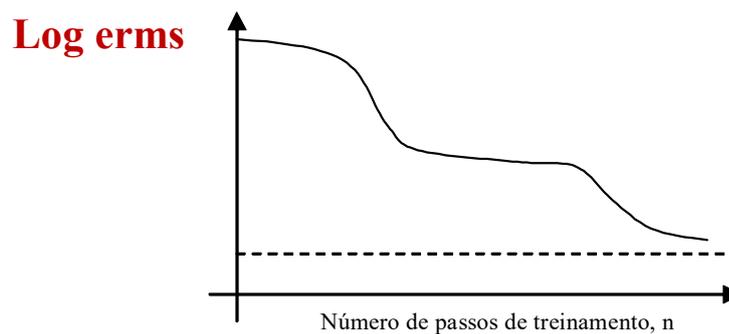
Obs: se na normalização  $y_i = \frac{1}{\sigma_{Y_i}} (Y_i - \mu_{Y_i})$

então o erro eficaz na saída real, não escalada, é:

$$erms_{Y_i} = \sigma_{Y_i} erms_{y_i}$$

## Escalas

Usualmente o erro médio quadrático (emq) e o erro eficaz (erms) variam por décadas - **usar escala log**



## Saídas Lógicas:

### Erms versus Erro de Classificação

$$F_{\text{classificação}} = \frac{1}{4} E_P [y_i - \text{sign}(\tilde{y}_i)]^2$$

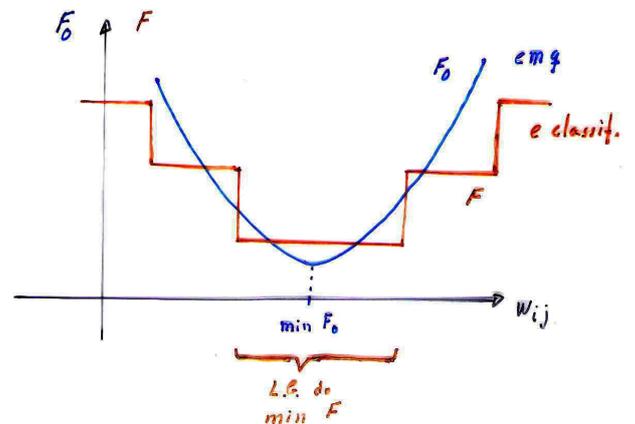
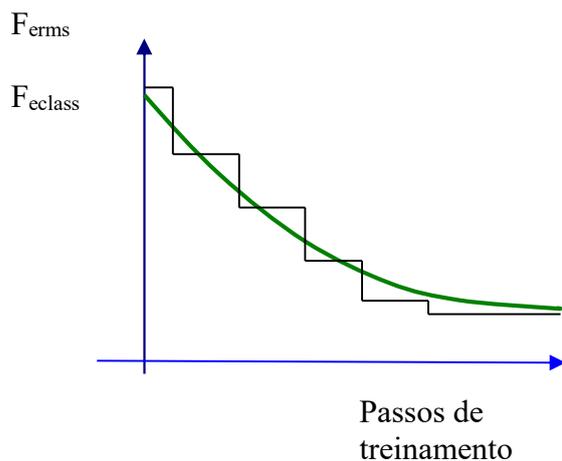
$$= \frac{1}{4P} \sum_{i=1}^P [y_i - \text{sign}(\tilde{y}_i)]^2$$

- erro percentual eficaz na classificação

### No acompanhamento do erro de saídas lógicas

Plotar os dois erros, **log erms** e erro de classificação !

**usar escala log**



## 8.5 - Validação Cruzada

baixa estatística

Caso extremo: leave one out

Conjunto dos Pares entrada saída >>>

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

Composição dos Conjuntos:

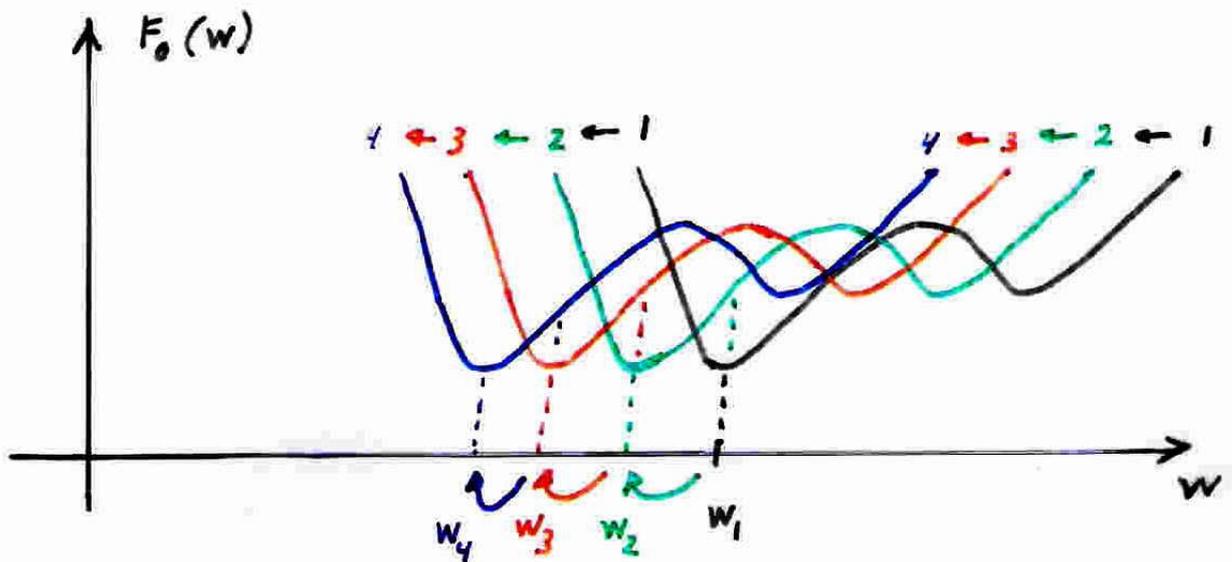
Treinamento 60%	Validação 20%	Teste 20%
<b>C+D+E *</b>	<b>B</b>	<b>A</b>
<b>B+D+E</b>	<b>C</b>	<b>A</b>
<b>B+C+E</b>	<b>D</b>	<b>A</b>
<b>B+C+D</b>	<b>E</b>	<b>A</b>
<b>C+D+E *</b>	<b>A</b>	<b>B</b>
<b>A+D+E</b>	<b>C</b>	<b>B</b>
...	...	...
<b>A+B+D</b>	<b>C</b>	<b>E</b>
<b>A+B+C</b>	<b>D</b>	<b>E</b>

Total: 20 casos ( \* 10 treinamentos a realizar)

Verificar média e dispersão do erro nos **conjunto de teste**.

## 8.6 – Treinamento dinâmico, adaptativo

### Sistemas variantes no tempo



### Variações lentas vs. rápidas

#### Variações lentas nos pares entrada-saída (no mapeamento)

a rede acompanha, “aprende” o novo mapeamento e “esquece” os pares antigos (se  $\alpha > 0$ )

#### Variações bruscas nos pares entrada –saída (no mapeamento)

a rede pode não aprender o novo mapeamento, pode ficar presa em um mínimo local, etc. Mais seguro refazer o treinamento.