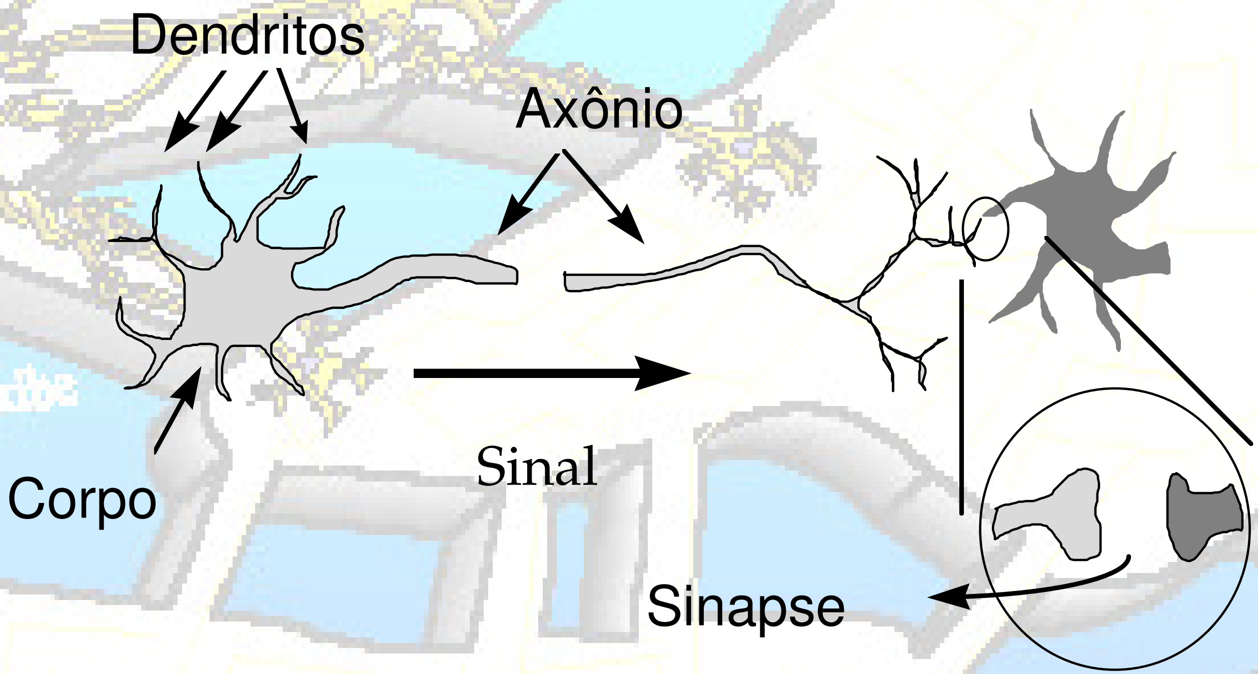
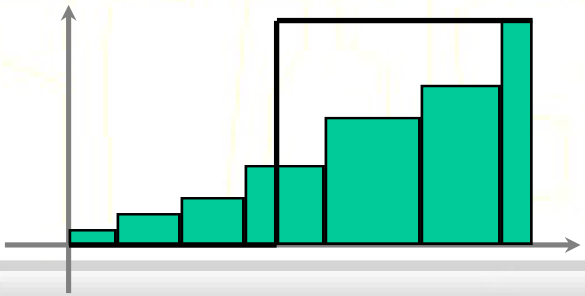
Perceptrons e *Backpropagation*

Germano C. Vasconcelos

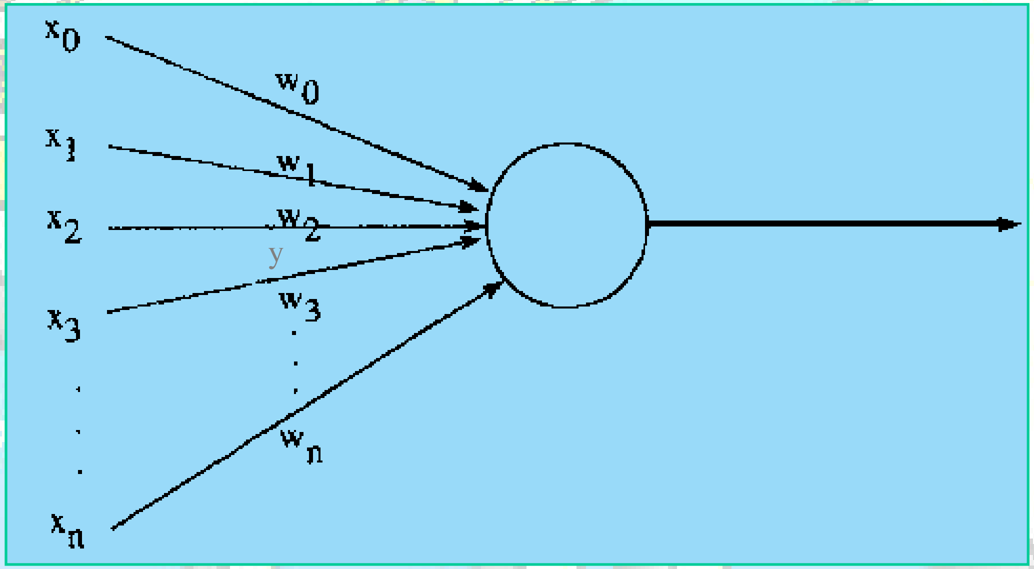
Centro de Informática - UFPE

Neurônio Natural



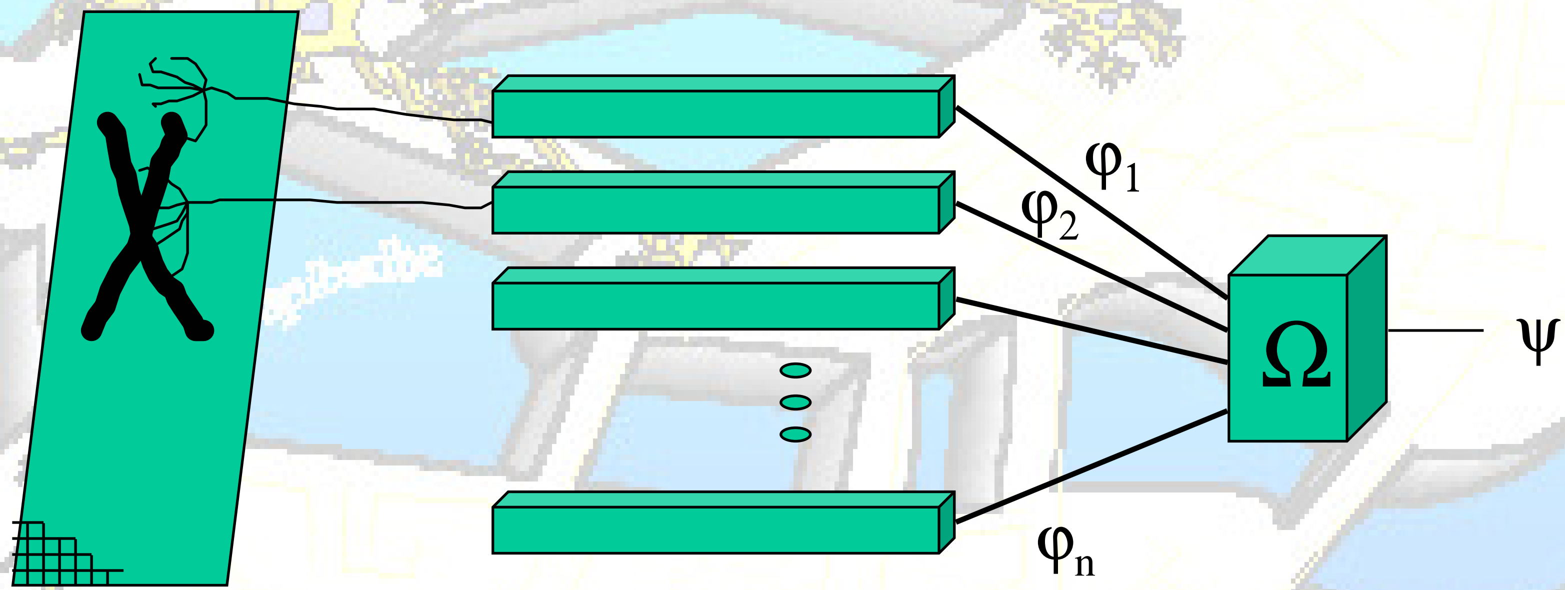


Neurônio McCulloch-Pitts

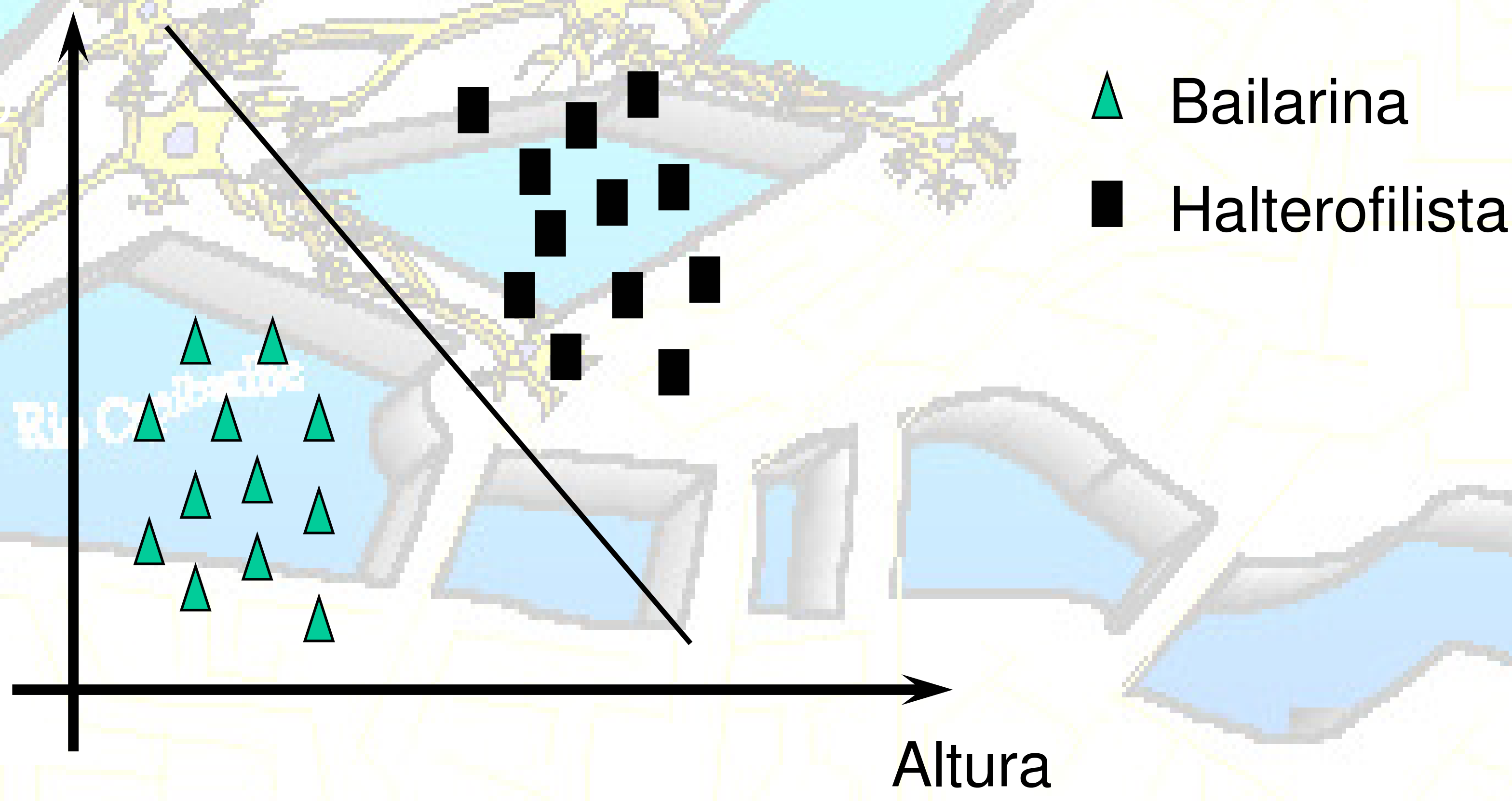


$$y = f_h\left[\sum_{i=1}^{n} w_i x_i - \theta\right]$$

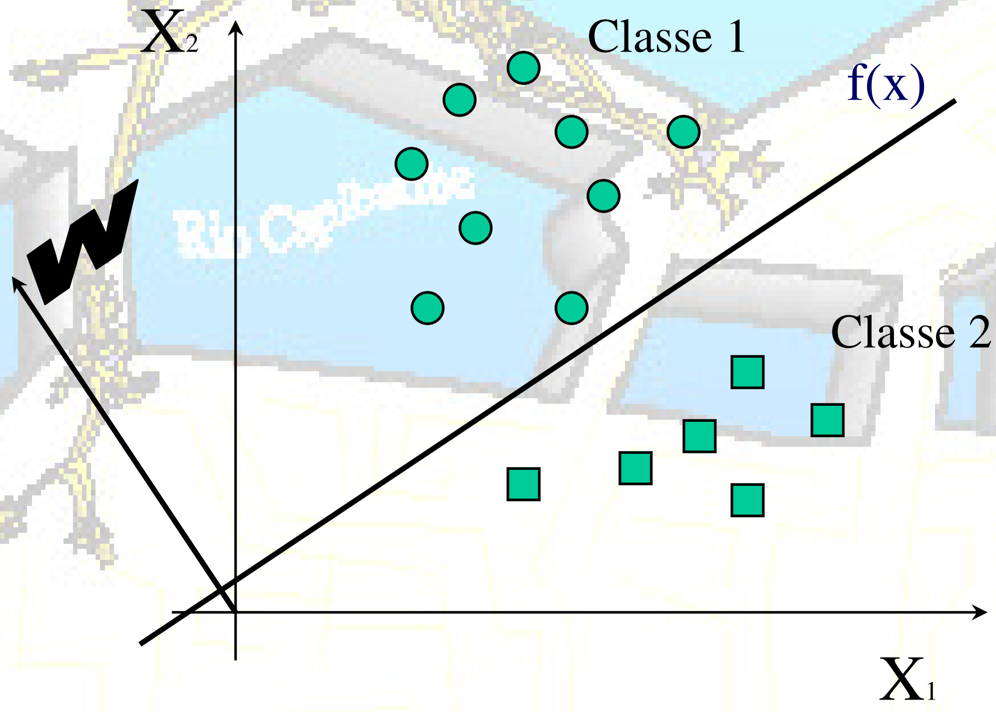
Perceptrons



Reconhecimento de Padrões



Uma Visão Matemática do Perceptron



$f(x)=\Sigma w_i . x_i - \theta$

$f(x)=(|W|.|X| \cos\Phi) - \theta$

Considere o ponto onde

$f(x) = 0$:

$w_1.x_1 + w_2.x_2 - \theta = 0$

↓

$x_2 = -w_1/w_2.x_1 + \theta/ w_2$

$(y = m.x +c)$

Aprendizagem no Perceptron

- Se um padrão é ***corretamente*** classificado

$W(t+1) = W(t)$ se $\Sigma w_i.x_i - \theta \geq 0$ e X pertence a C1

$W(t+1) = W(t)$ se $\Sigma w_i.x_i - \theta < 0$ e X pertence a C2

Se um padrão é ***incorretamente*** classificado

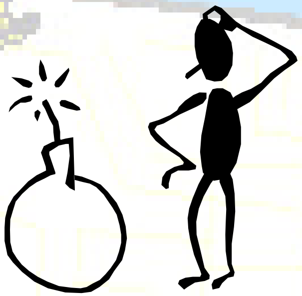
$W(t+1) = W(t) - \eta X$ se $\Sigma w_i.x_i - \theta \geq 0$ e X pertence a C2

$W(t+1) = W(t) + \eta X$ se $\Sigma w_i.x_i - \theta < 0$ e X pertence a C1

Características do Perceptron

Simples Operação

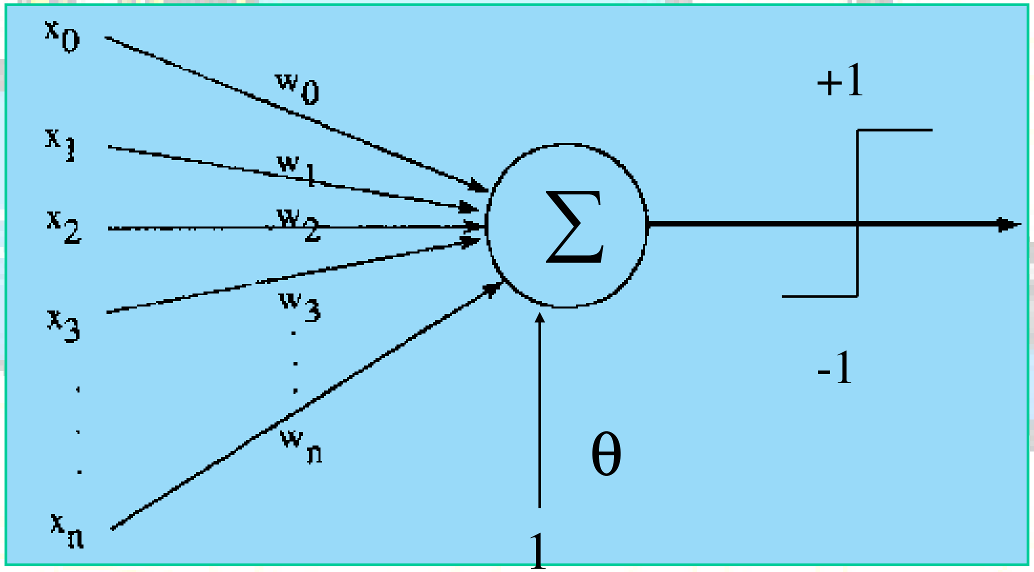
Convergência Garantida



Capaz de resolver apenas problemas linearmente separáveis

Adaline (Adaptive Linear Neuron)

Bernard Widrow 1960



Aprendizagem no Adaline

- $e_i = (d_i - y_i)$

- C (custo) $= \frac{1}{2} \sum_p (d_i - y_i)^2$

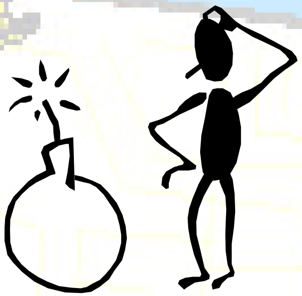
- $W(t+1) = W(t) + \eta \, e \, X(t)$

(Regra de Widrow-Hoff)

Características do Adaline

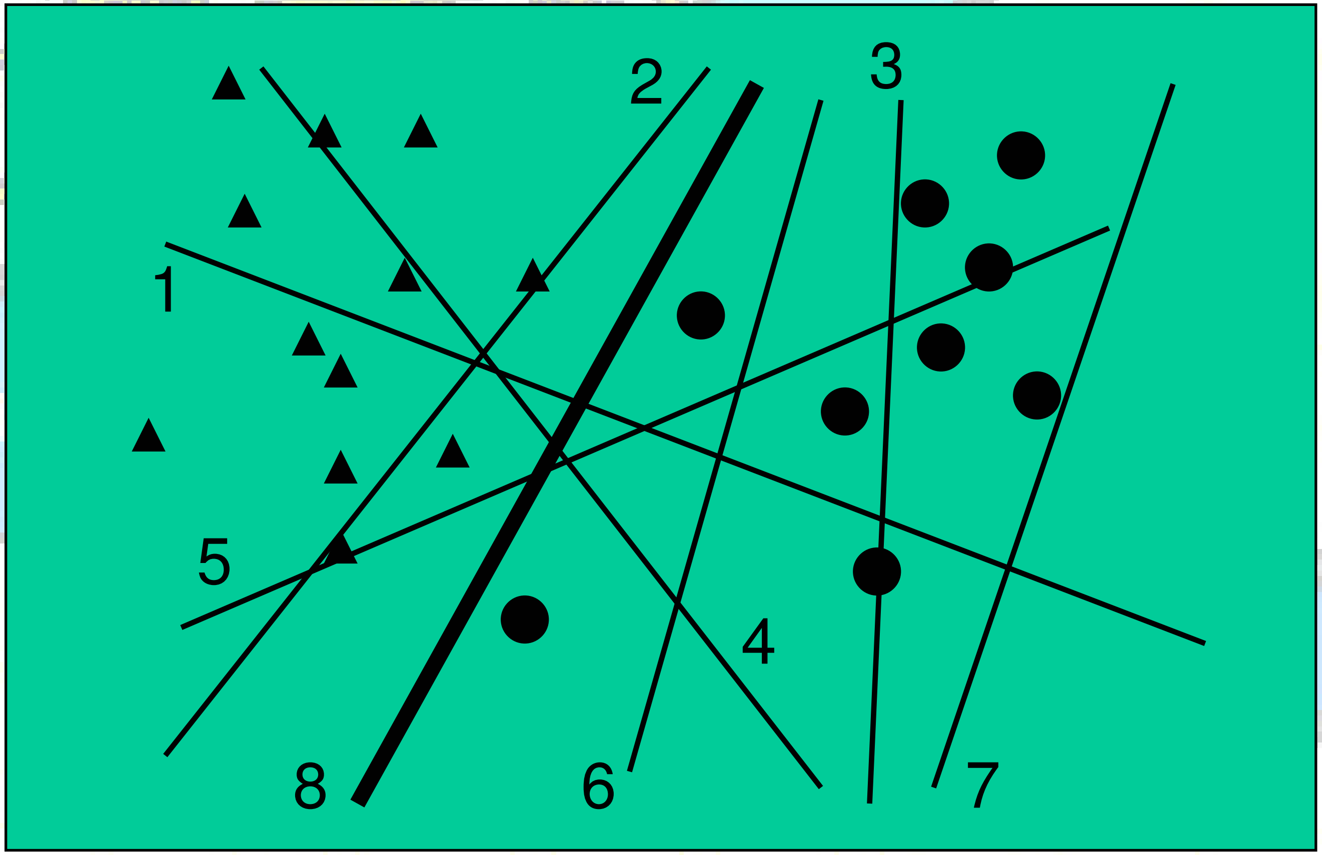
Simples Operação

Convergência Garantida

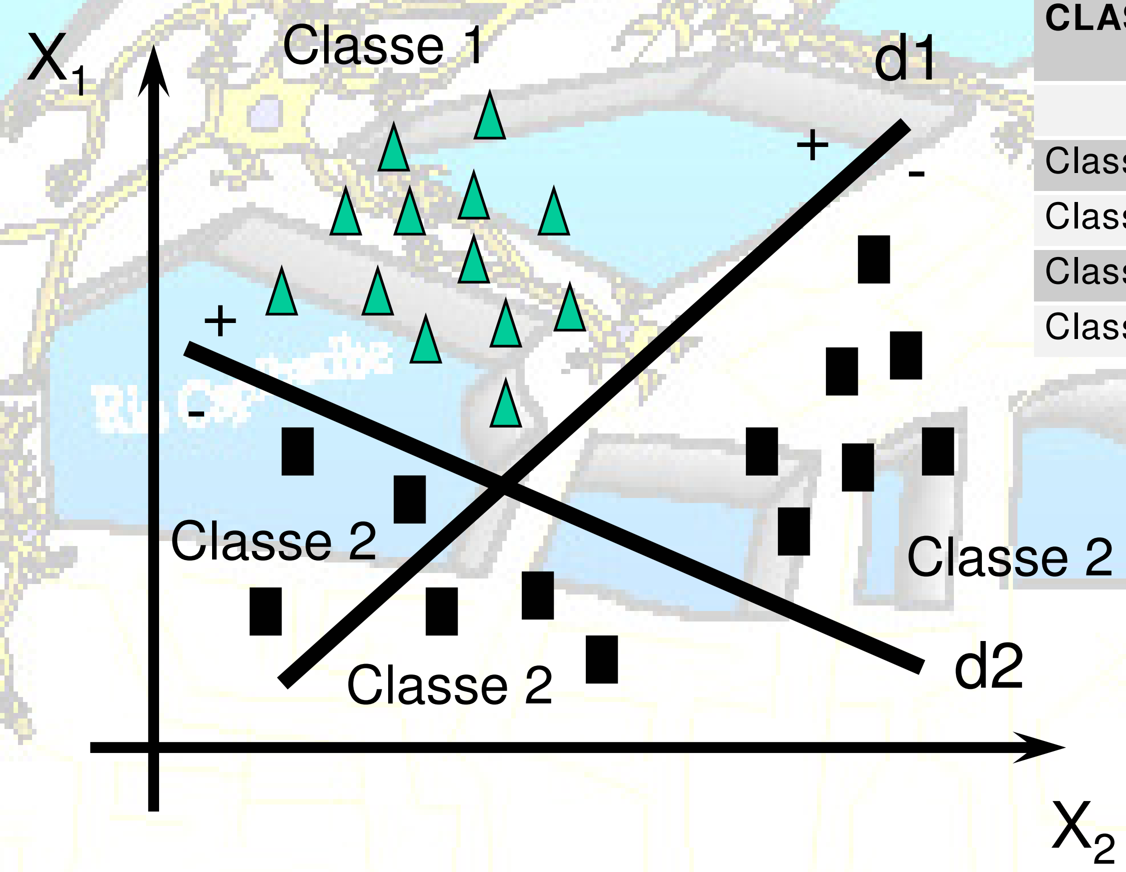


Capaz de resolver apenas problemas linearmente separáveis

Visualização do Treinamento

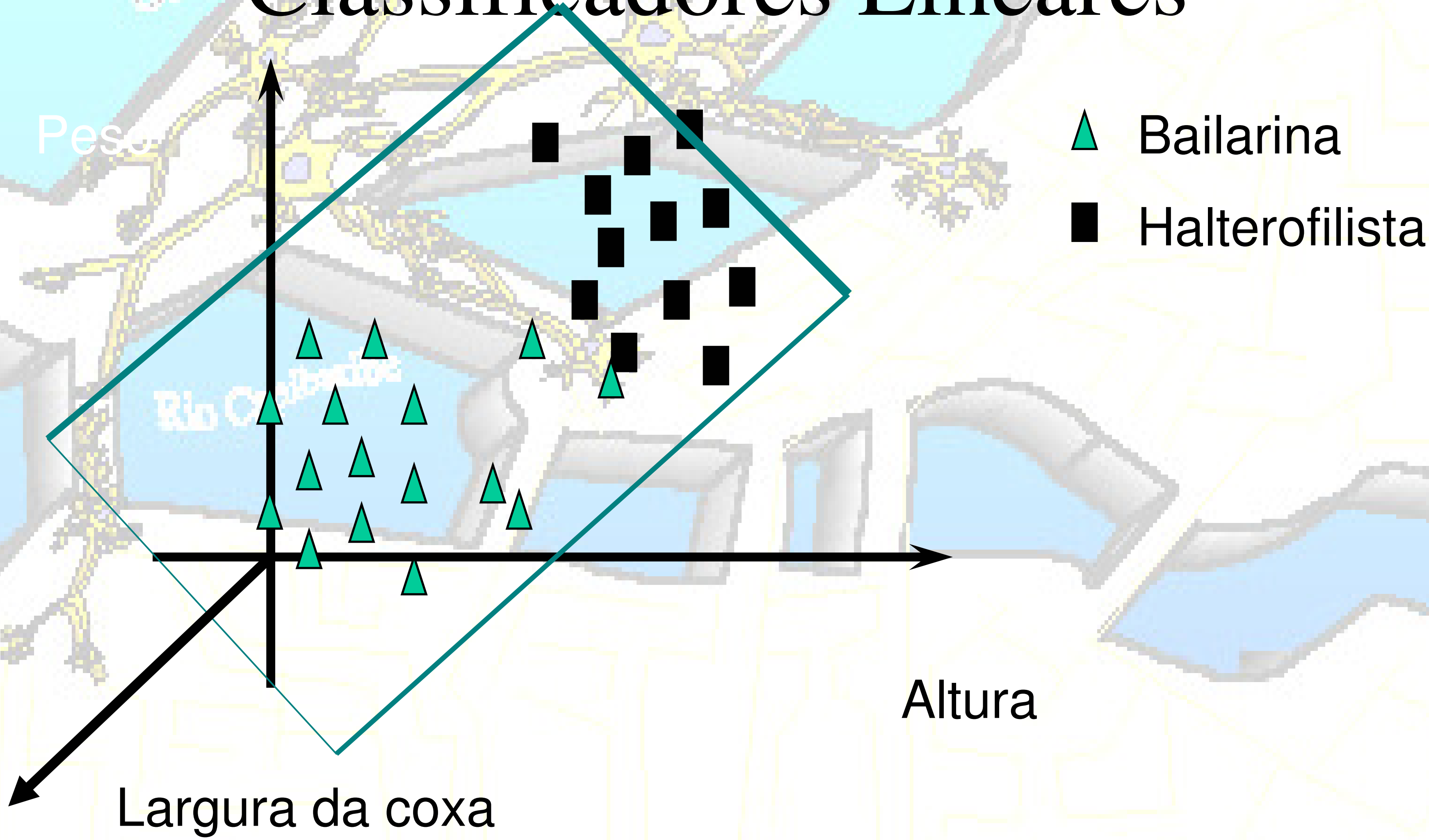


Classificadores lineares

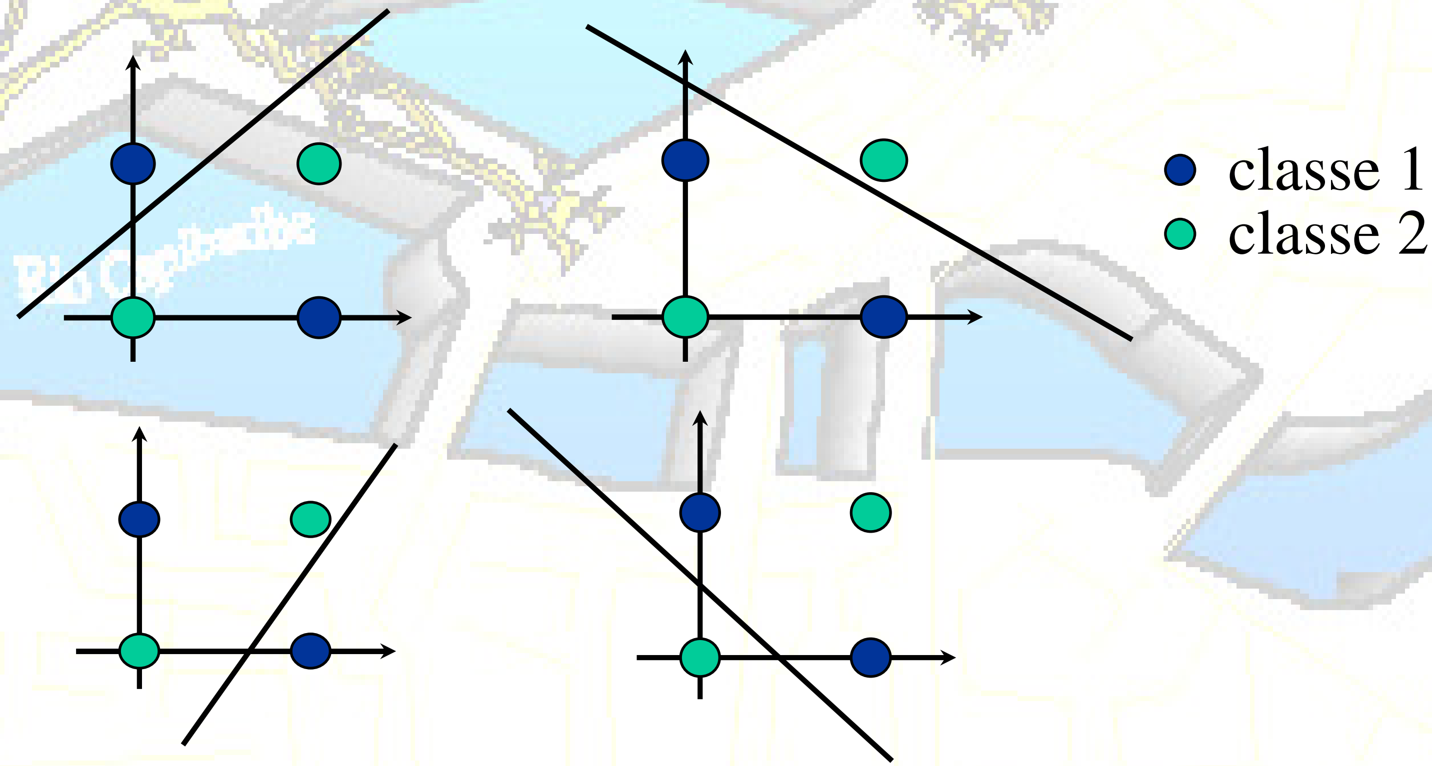


CLASSIFICAÇÃO	SINAL DA LINHA DE DECISÃO	
	d1	d2
Classe 1	+	+
Classe 2	+	-
Classe 2	-	+
Classe 2	-	-

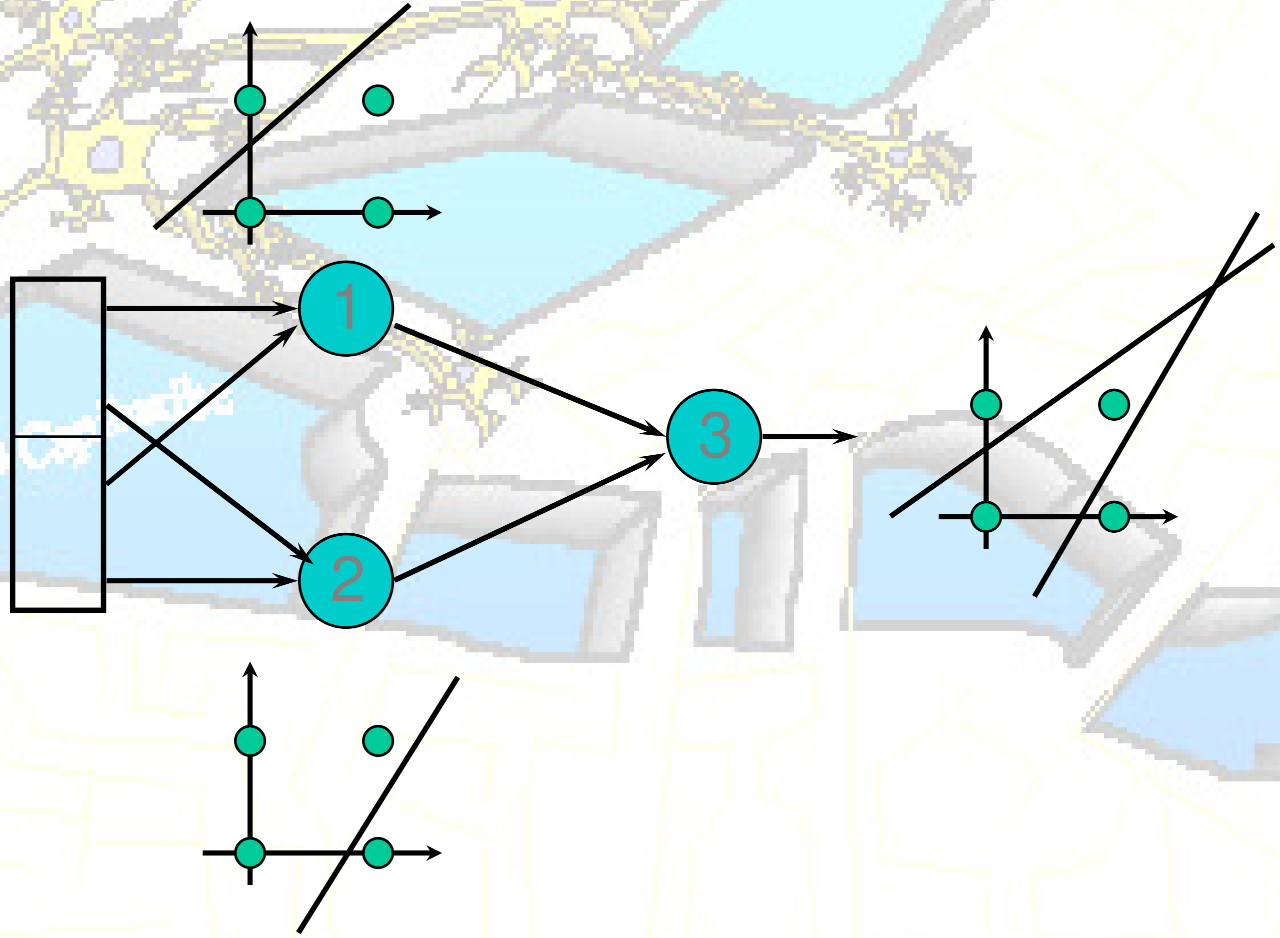
Classificadores Lineares



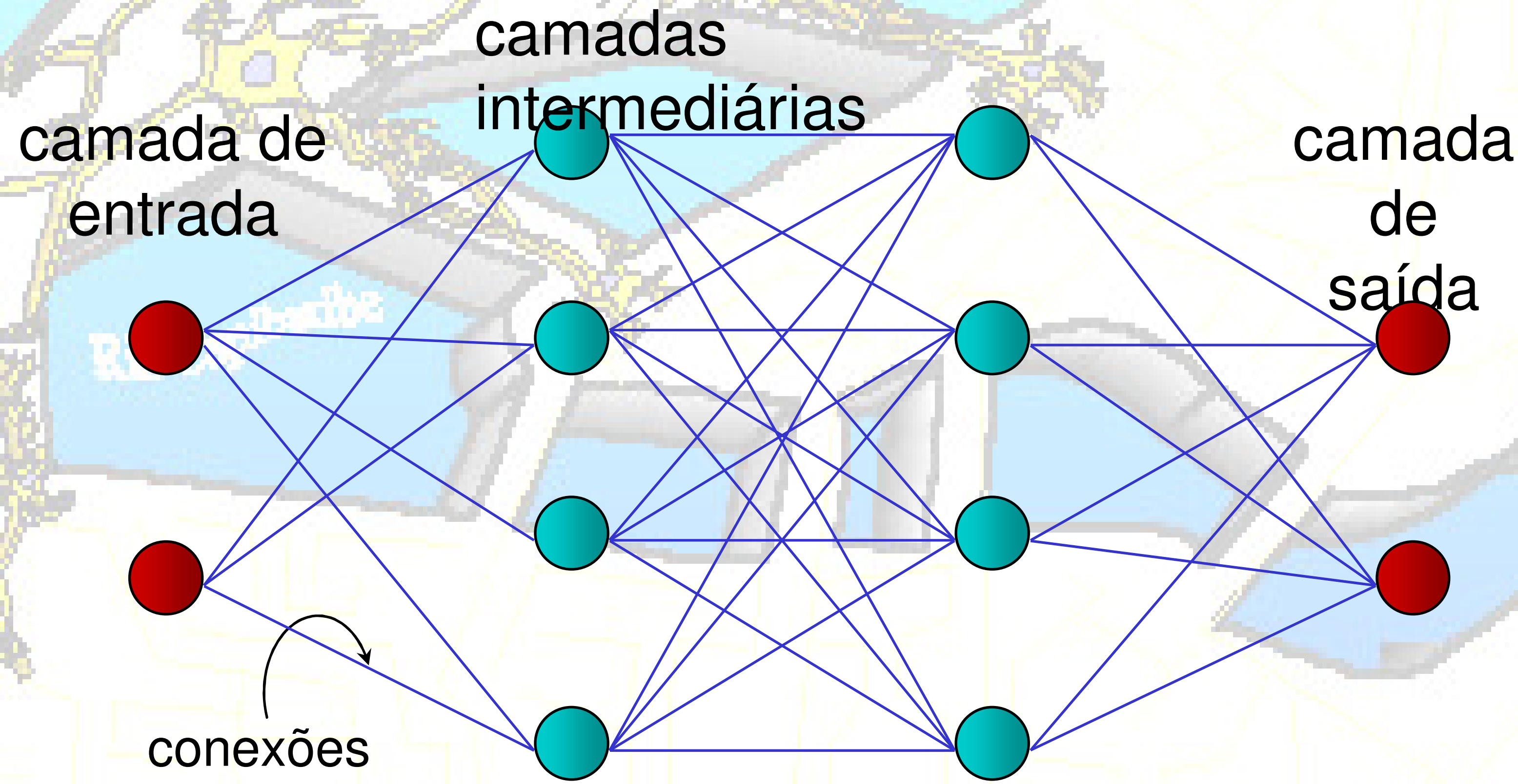
O Problema do Ou-exclusivo (XOR)



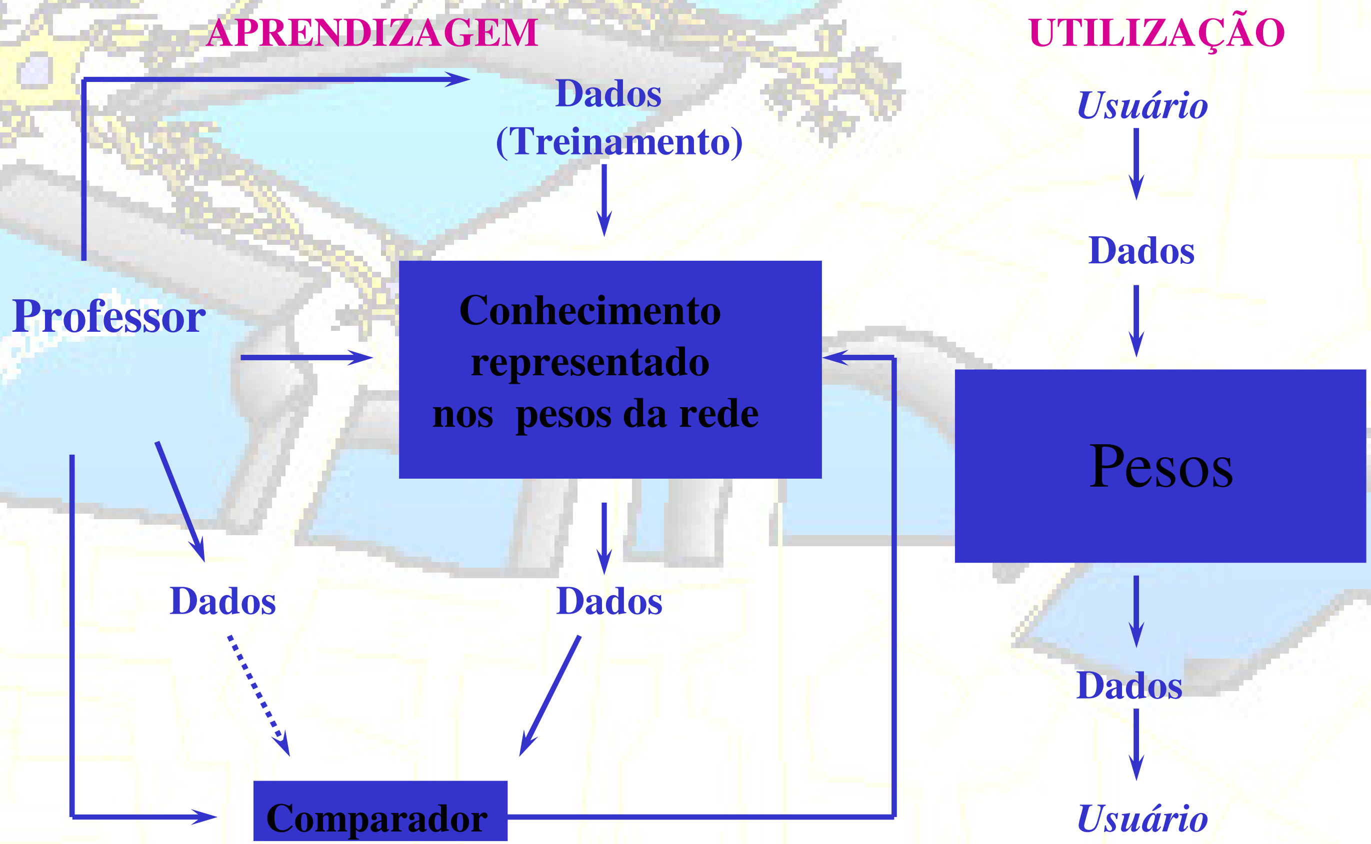
Solução para o XOR



Multilayer Perceptron (MLP) e Backpropagation



Funcionamento do MLP



Funcionamento do MLP

- Duas fases de operação
  - Passo para frente (*forward pass*)
  - Passo para trás (*backward pass*)

- Dado um conjunto de pares (X_p, Y_p), construir um mapeamento $F(W; X_p) \Rightarrow Y_p$

Como construir $F(W; X_p) \Rightarrow Y_p$?

Regra Delta Generalizada ou Error-Back Propagation

O erro na camada de saída :

$$E_p = \frac{1}{2}\sum_{j=1}^{n}(t_{pj} - o_{pj})^2$$

Para minimizar o erro :

$$\Delta_p W_{ji} \propto -\frac{\partial E_p}{\partial W_{ji}}$$

$$\frac{\partial E_p}{\partial W_{ji}} = \frac{\partial E_p}{\partial net_{pj}} \frac{\partial net_{pj}}{\partial W_{ji}}$$

$$\frac{\partial net_{pj}}{\partial W_{ji}} = \frac{\partial}{\partial W_{ji}} \sum_k W_{jk} O_{pk} = O_{pi}$$

$$\delta_{pj} = -\frac{\partial E_p}{\partial net_{pj}} \longrightarrow -\frac{\partial E_p}{\partial W_{ji}} = \delta_{pj} O_{pi}$$

$$\Delta_p W_{ji} = \eta \delta_{pj} . O_{pi}$$

$$\delta_{pj} = -\frac{\partial E_p}{\partial net_{pj}} = -\frac{\partial E_p}{\partial O_{pj}} \frac{\partial O_{pj}}{\partial net_p}$$

$$\frac{\partial O_{pj}}{\partial net_{pj}} = f_j^{\prime}(net_{pj})$$

$$\frac{\partial E_p}{\partial O_{pj}}$$

(2 casos precisam ser considerados!)

Primeiro caso : j é uma unidade de saída

$$\frac{\partial E_p}{\partial O_{pj}} = -(t_{pj} - o_{pj})$$

$$\delta_{pj} = (t_{pj} - o_{pj})\, f_j^{\prime}(net_{pj})$$

Segundo caso : *j* é uma unidade intermediária

$$\frac{\partial E_p}{\partial o_{pj}} = \sum_k \frac{\partial E_p}{\partial net_{pk}} \frac{\partial net_{pk}}{\partial o_{pj}}$$

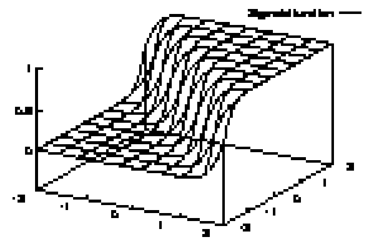
$$\frac{\partial E_p}{\partial o_{pj}} = -\sum_k \frac{\partial E_p}{\partial net_{pk}} \frac{\partial}{\partial o_{pj}} \sum_i w_{ik} o_{pi}$$

$$\Rightarrow \frac{\partial E_p}{\partial o_{pj}} = - \sum \delta_{pk} w_{jk}$$

$$\delta_{pj} = f_j'(net_{pj}) \sum \delta_{pk} w_{jk}$$

E a função de ativação f ?

Considerando uma função sigmoid



$$f'(net_{pj})\ ?$$

$$o_{pj} = f(net_{pj}) = \frac{1}{1 + \exp(-net_{pj})}$$

$$f'(net_{pj}) = o_{pj}(1 - o_{pj})$$

Características do MLP

- Aproximador Universal de Funções
  - Uma única camada intermediária é capaz de aproximar qualquer função contínua definida em um hipercubo
- Alta capacidade de generalização
- Convergência para mínimo global não garantida
- Em alguns casos, lento na aprendizagem

De uma maneira geral...

Uma rede neural pode ser vista como um conjunto de funções $Y_k(X_p;W)$, tal que dado $X_p \Rightarrow Y_p$

No caso de classificação

$$Y_k = \begin{cases} 1 & \text{se } X_p \in k \\ 0, & \text{caso contrário} \end{cases}$$

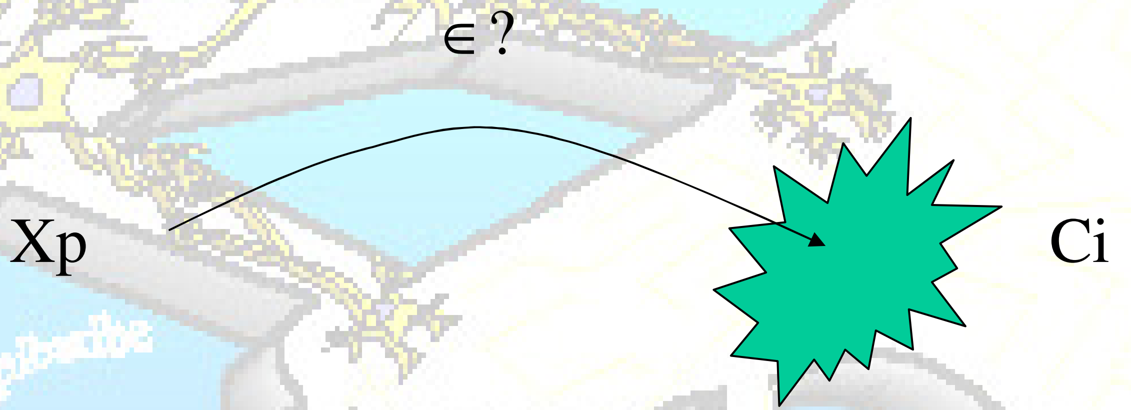
No caso em que Y_k são variáveis contínuas

⇒ problema de regressão

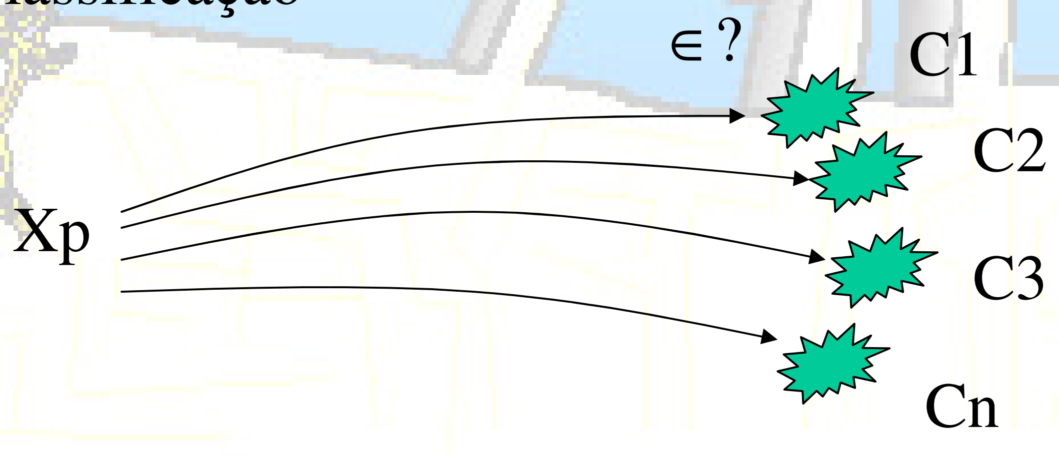
⇒ ou problema de aproximação de funções

Reconhecimento de Padrões

Verificação

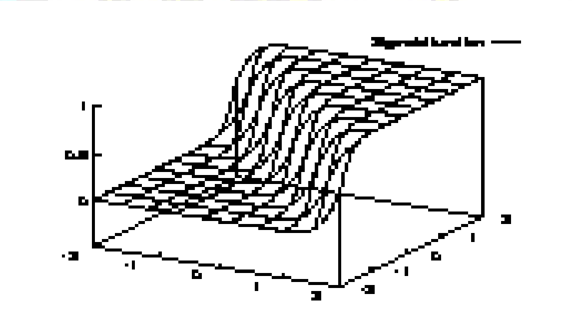


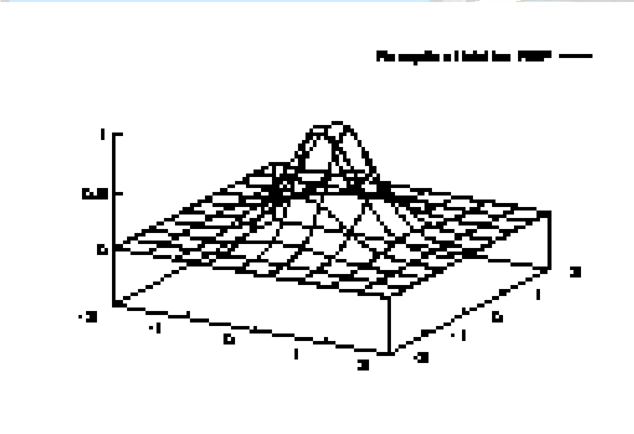
Classificação

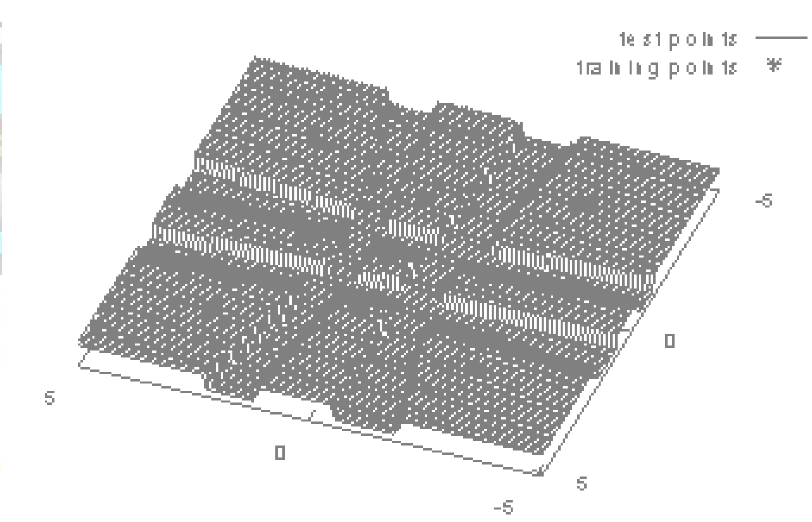


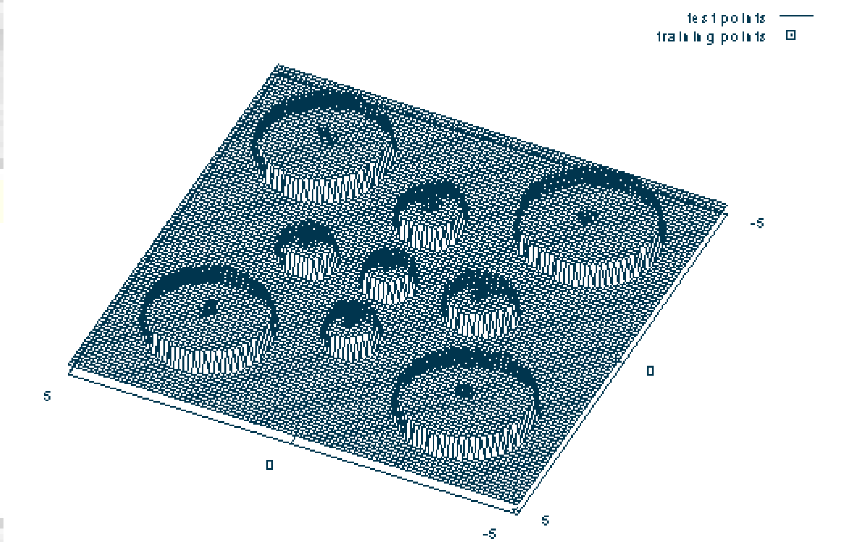
$Y_k = Y_k(X_p;W)$
$k=1,2,\ldots,n$

Reconhecimento de Padrões

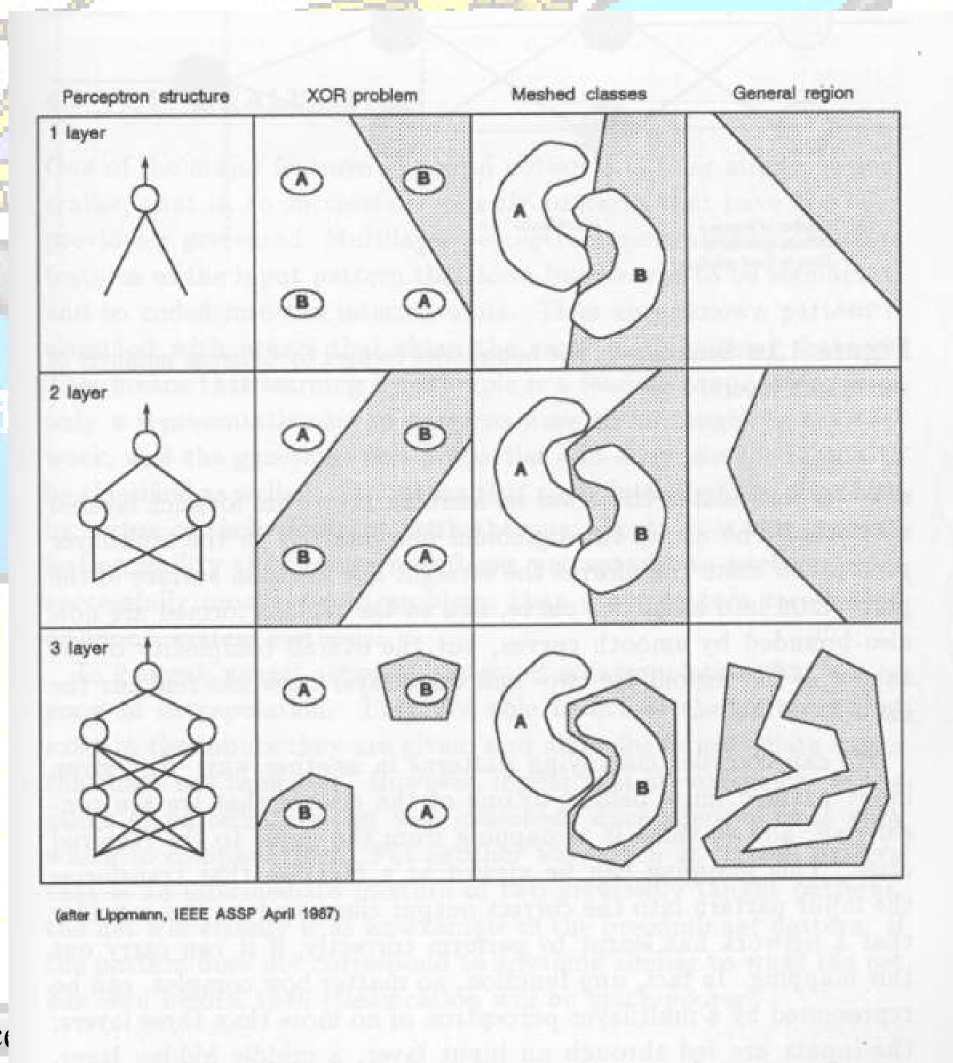




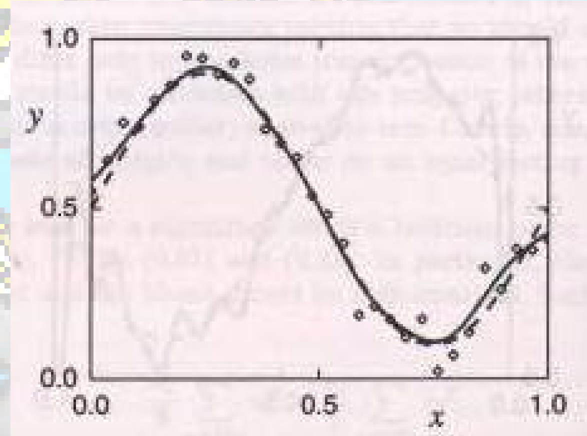


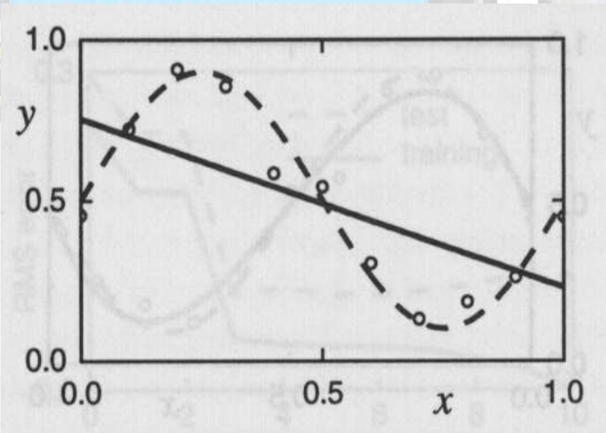


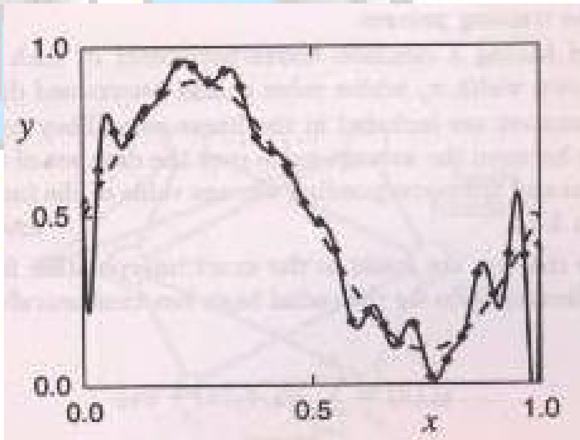
Complexidade Funcional do MLP x Número de Camadas



Complexidade Funcional versus Over-fitting







Treinamento com Validação Cruzada

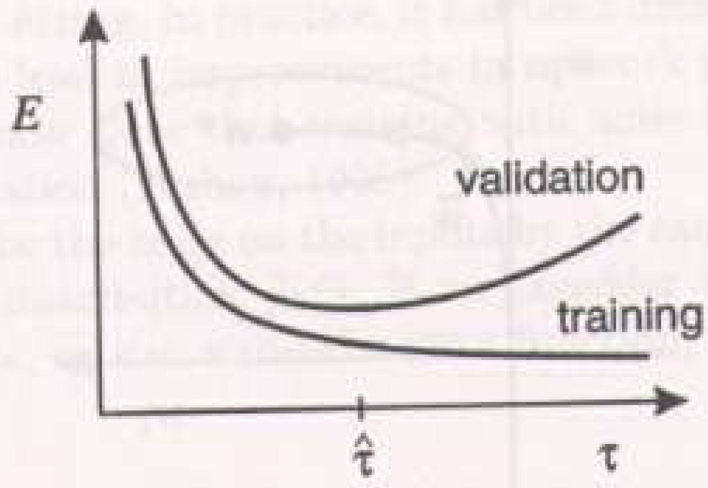
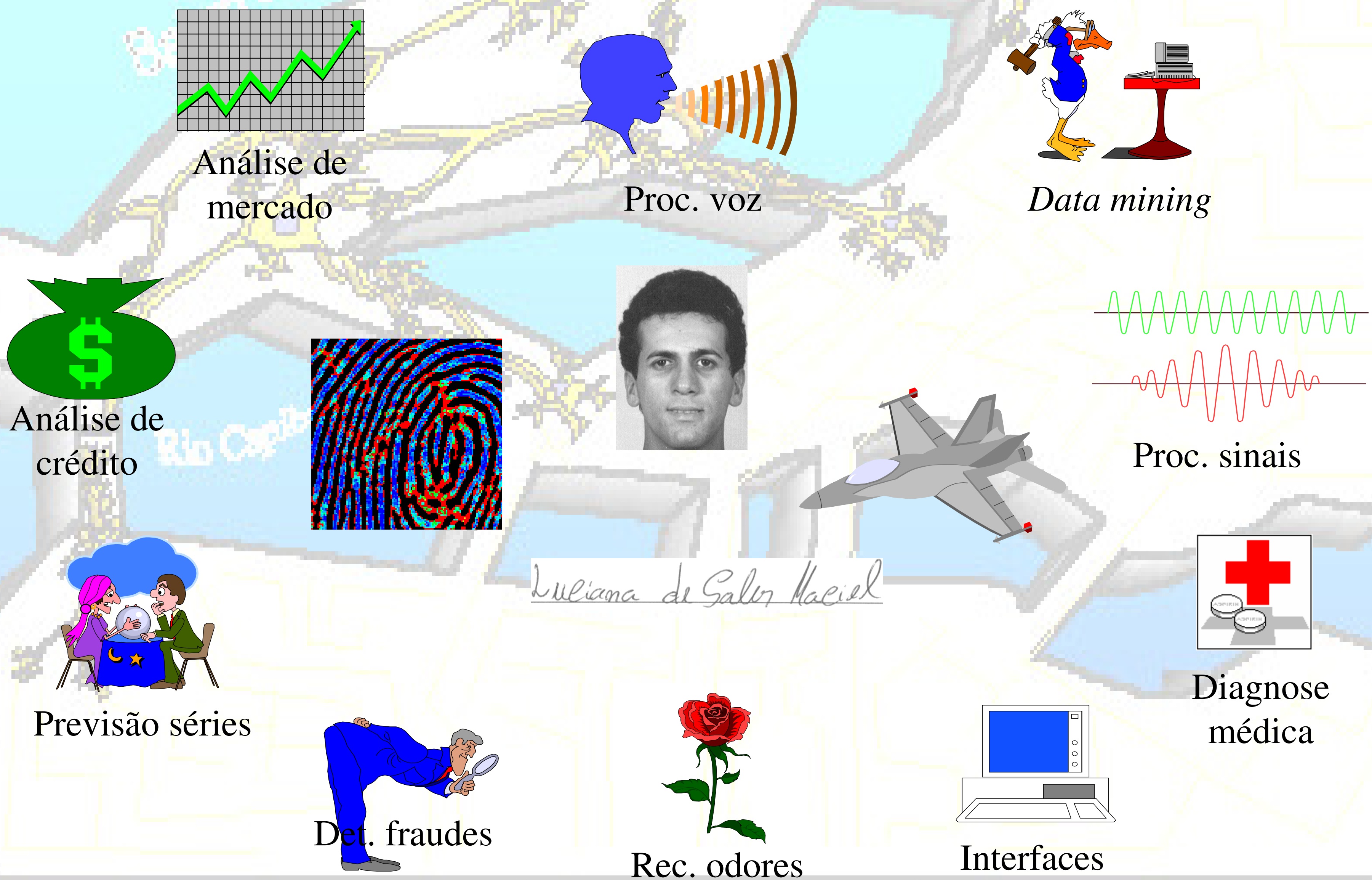


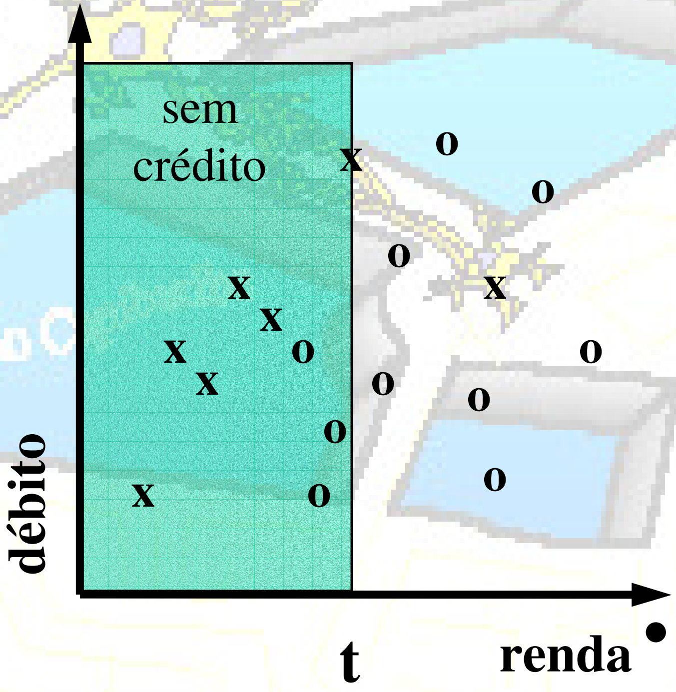
Figure 9.7. A schematic illustration of the behaviour of training and validation set errors during a typical training session, as a function of the iteration step τ. The goal of achieving the best generalization performance suggests that training should be stopped at the point $\hat{\tau}$ corresponding to the minimum of the validation set error.

Aplicações do MLP



Complexidade Funcional (I)

Análise de crédito

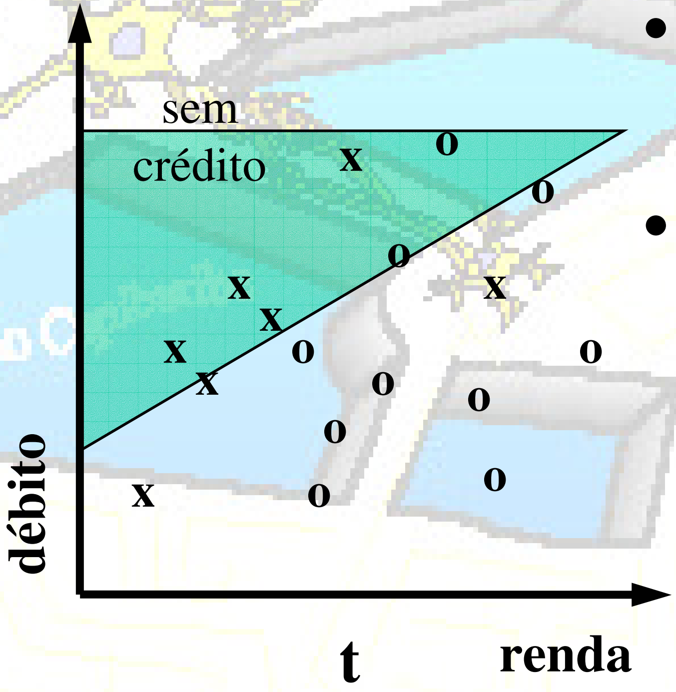


x: exemplo recusado
o: exemplo aceito

- Um hiperplano paralelo de separação: pode ser interpretado diretamente como uma regra:
  - se a renda é menor que *t*, então o crédito não deve ser liberado
- Exemplo:
  - árvores de decisão;
  - indução de regras

Complexidade Funcional (II)

Análise de crédito

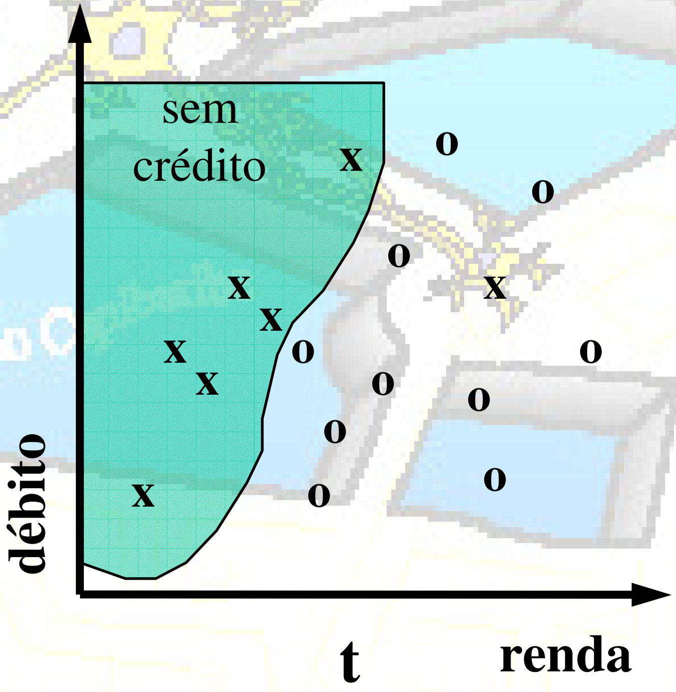


x: exemplo recusado
o: exemplo aceito

- Hiperplano oblíquo: melhor separação:
- Exemplos:
  - regressão linear;
  - perceptron;

Complexidade Funcional (III)

Análise de crédito

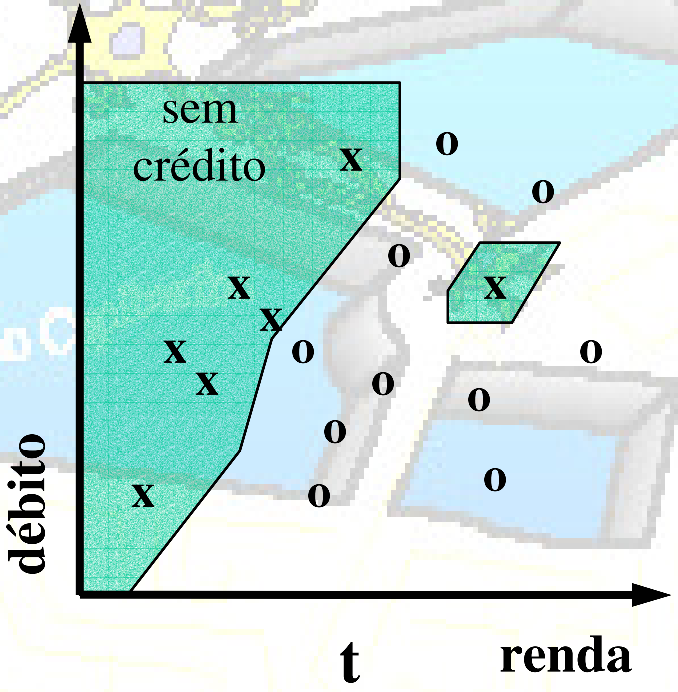


x: exemplo recusado
o: exemplo aceito

- Superfície não linear: melhor poder de classificação, pior interpretação;
- Exemplos:
  - perceptrons multicamadas;
  - regressão não-linear;

Complexidade Funcional (IV)

Análise de crédito

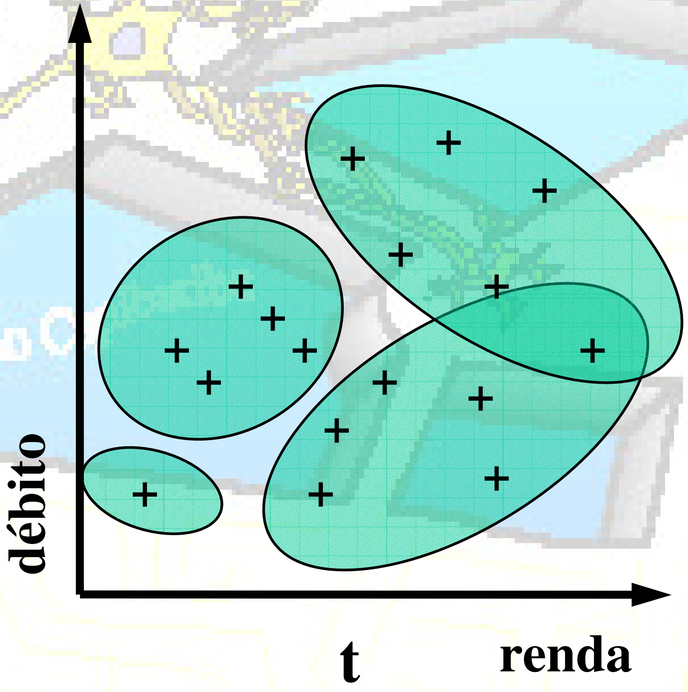


x: exemplo recusado
o: exemplo aceito

- Métodos baseado em exemplos;
- Exemplos:
  - k-vizinhos mais próximos;
  - raciocínio baseado em casos;

Complexidade Funcional (V)

Análise de crédito



+: exemplo

- Agrupamento
- Exemplo:
  - *vector quantization;*