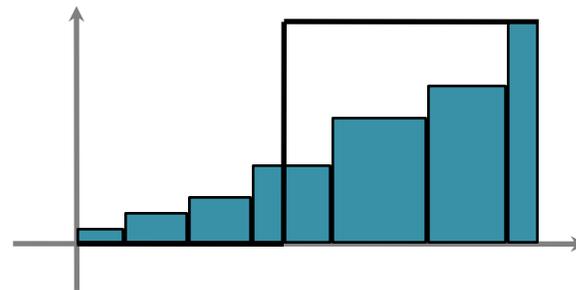
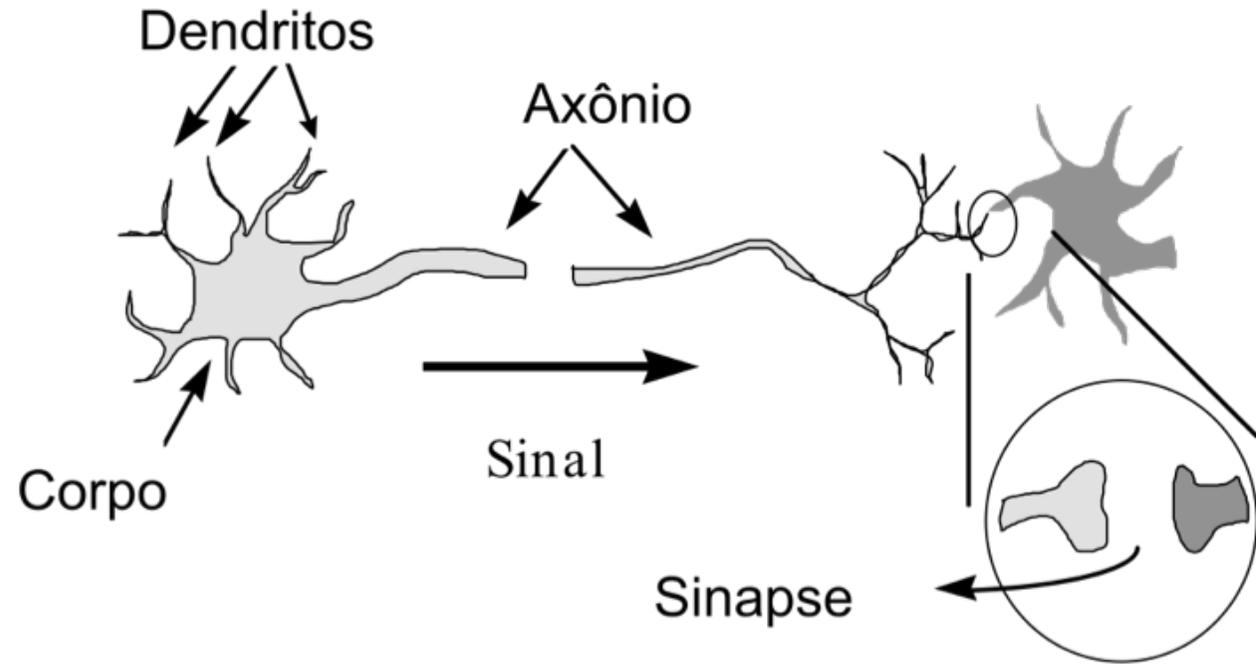


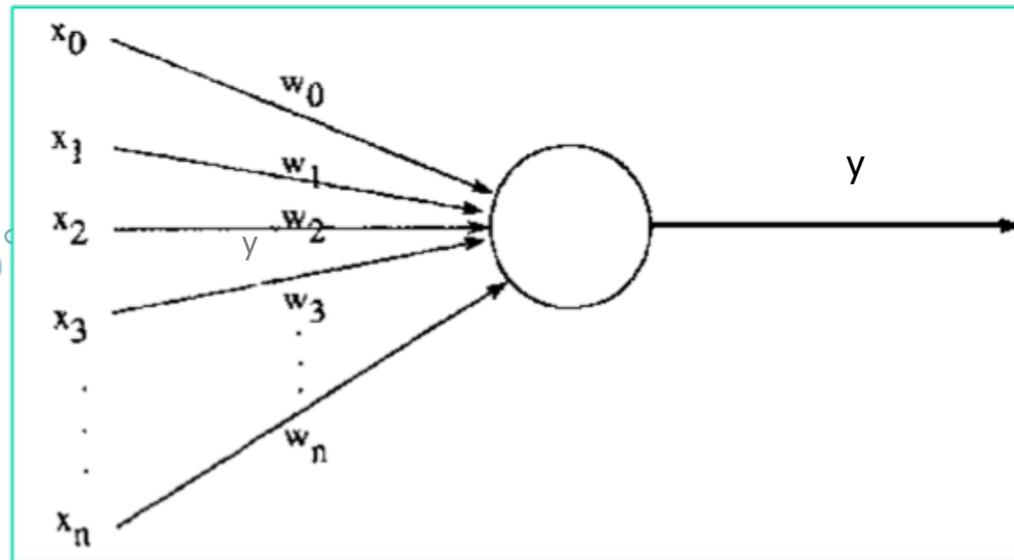
Perceptrons e Backpropagation

Germano C. Vasconcelos
Centro de Informática – UFPE

Neurônio Natural



Neurônio McCulloch–Pitts

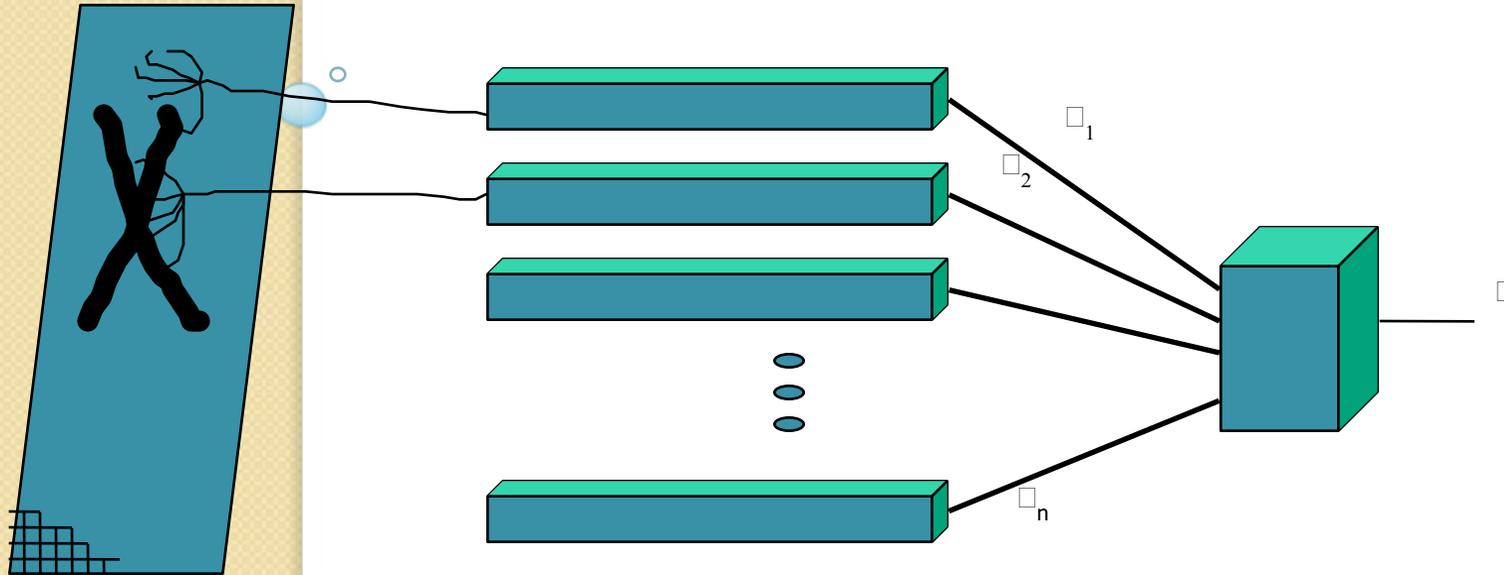


$$y = f_h \left[\sum_{i=1}^n w_i x_i - \theta \right]$$

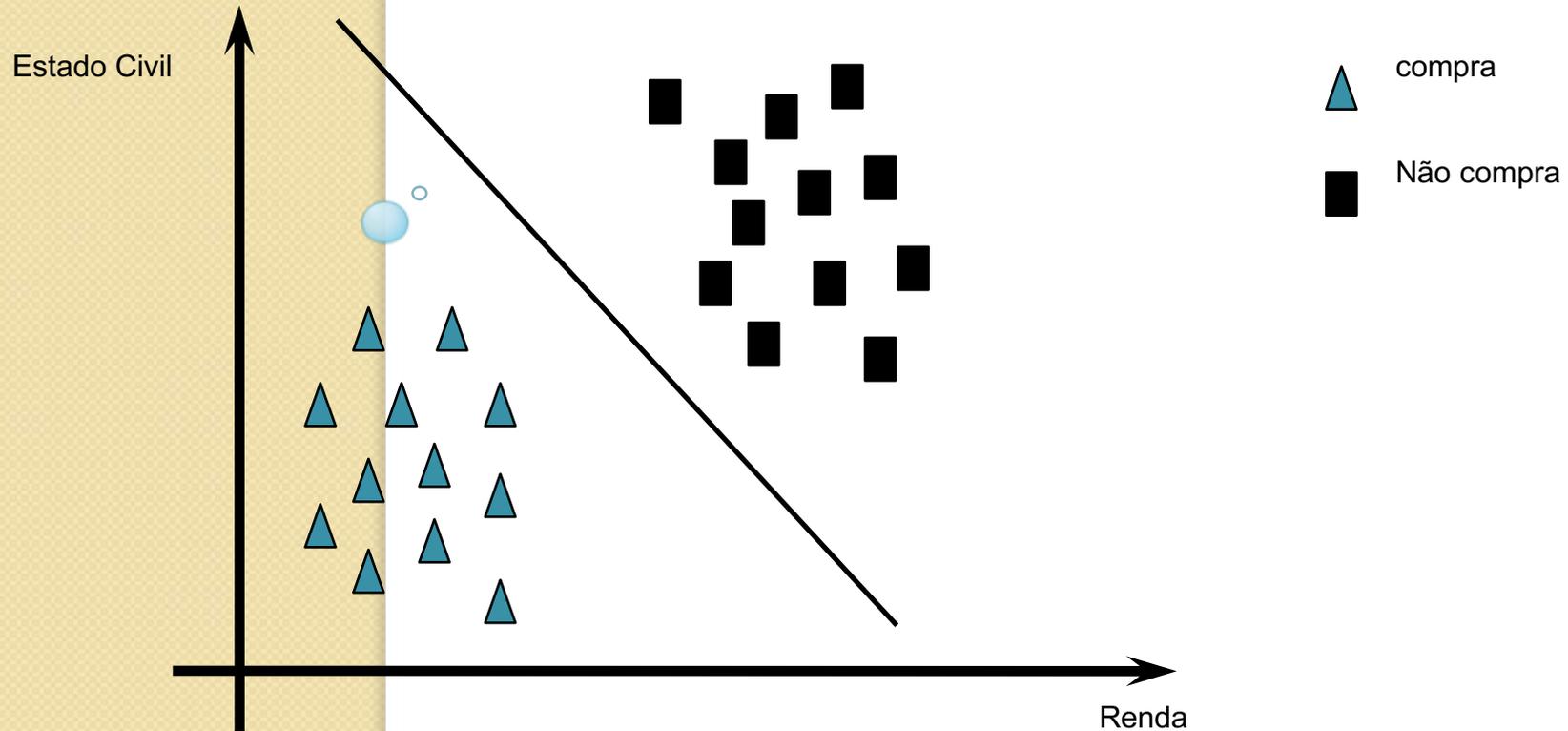
↑
 net_{pj}

Perceptrons

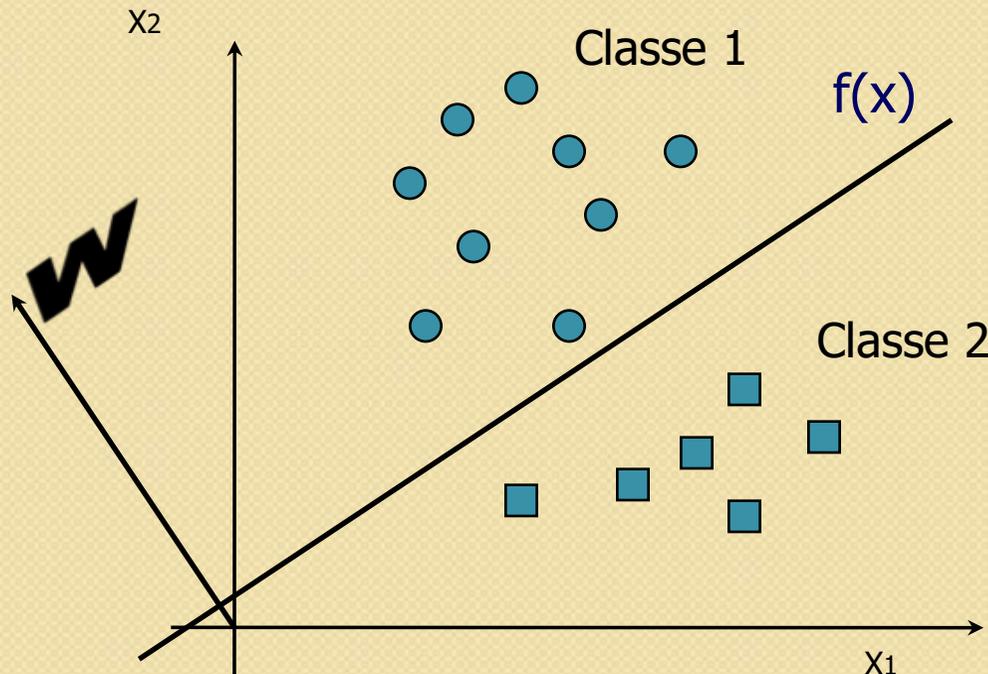
Rosemblatt [1958-1962]



Reconhecimento de Padrões



Visão Matemática do MCP



$$f(x) = \sum w_i \cdot x_i - \theta$$
$$f(x) = (|W| \cdot |X| \cos \phi) - \theta$$

Considere o ponto
onde

$$f(x) = 0:$$

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



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

$$(y = m \cdot x + c)$$

Aprendizagem no Perceptron

- Se um padrão é corretamente classificado

$$W(t+1) = W(t) \text{ se } \sum w_i \cdot x_i - \theta \geq 0 \text{ e } X \text{ pertence a } C1$$

$$W(t+1) = W(t) \text{ se } \sum w_i \cdot x_i - \theta < 0 \text{ e } X \text{ pertence a } C2$$

Se um padrão é incorretamente classificado

$$W(t+1) = W(t) - \eta X \text{ se } \sum w_i \cdot x_i - \theta \geq 0 \text{ e } X \text{ pertence a } C2$$

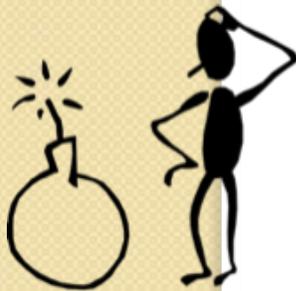
$$W(t+1) = W(t) + \eta X \text{ se } \sum w_i \cdot x_i - \theta < 0 \text{ e } X \text{ pertence a } C1$$

Características do Perceptron



Simple Operação

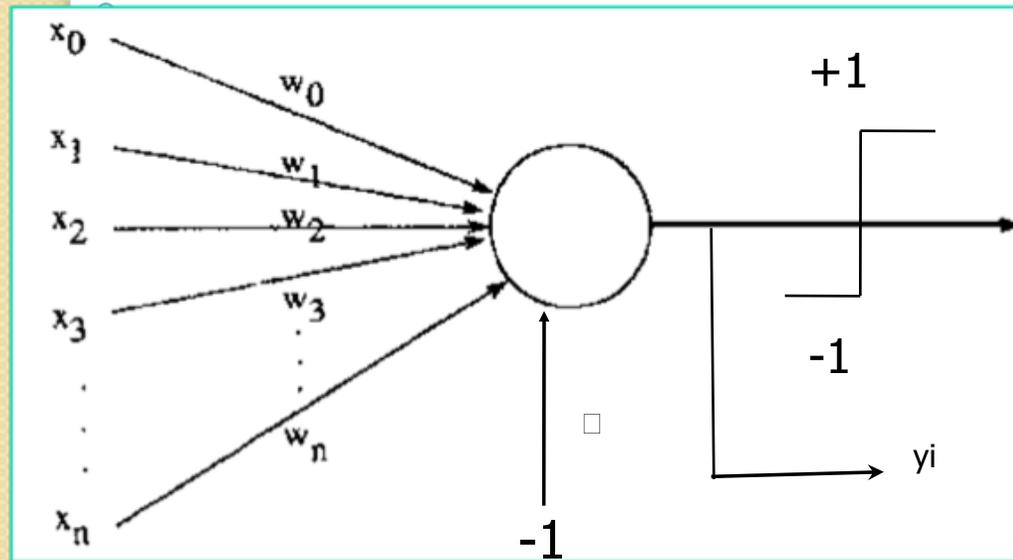
Convergência Garantida



Capaz de resolver apenas problemas linearmente separáveis

Adaline (Adaptive Linear Neuron)

Bernard Widrow 1960



Aprendizagem no Adaline

- $e_i = (d_i - y_i)$
- $e_i = C(\text{custo}) = \frac{1}{2} \sum_p (d_i - y_i)^2$ Mean Square Error (MSE) ou Erro Médio Quadrático
- $W_i(t+1) = W_i(t) + \eta e_i X(t)$

(Regra de Widrow–Hoff ou Regra Delta)

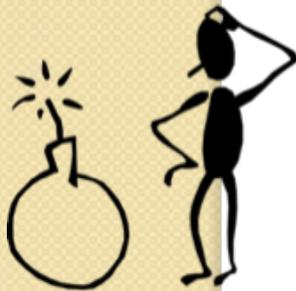
Características do Adaline



Simple Operação

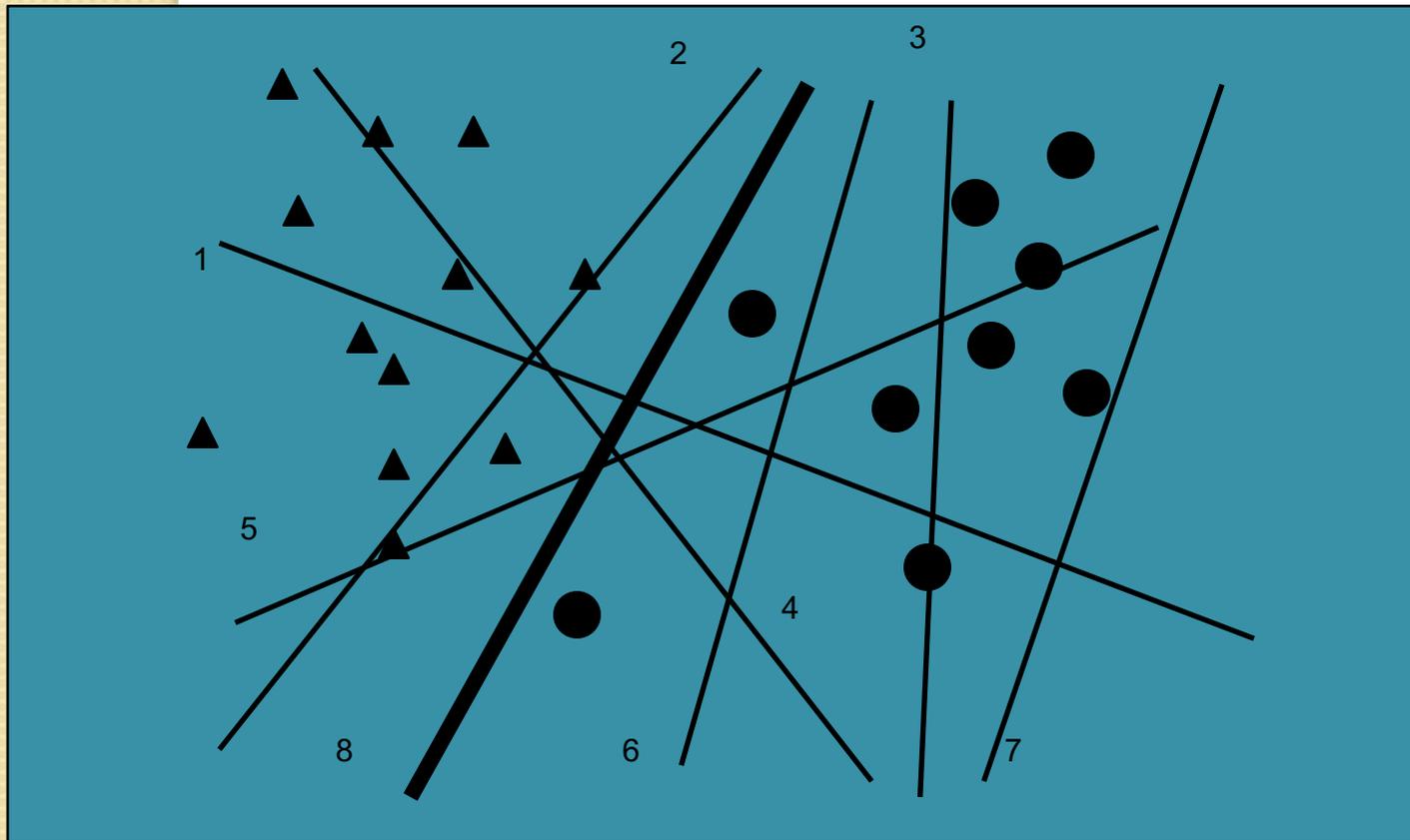


Convergência Garantida

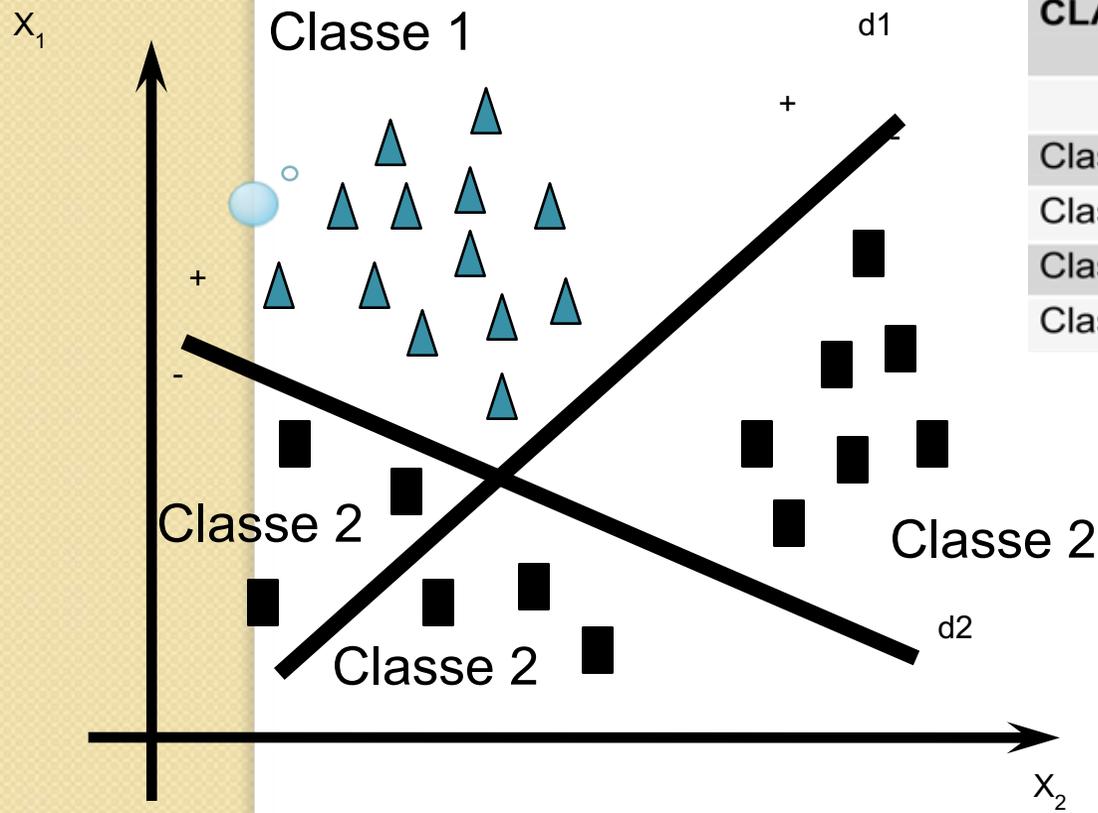


Capaz de resolver apenas problemas linearmente separáveis

Visualização do Treinamento

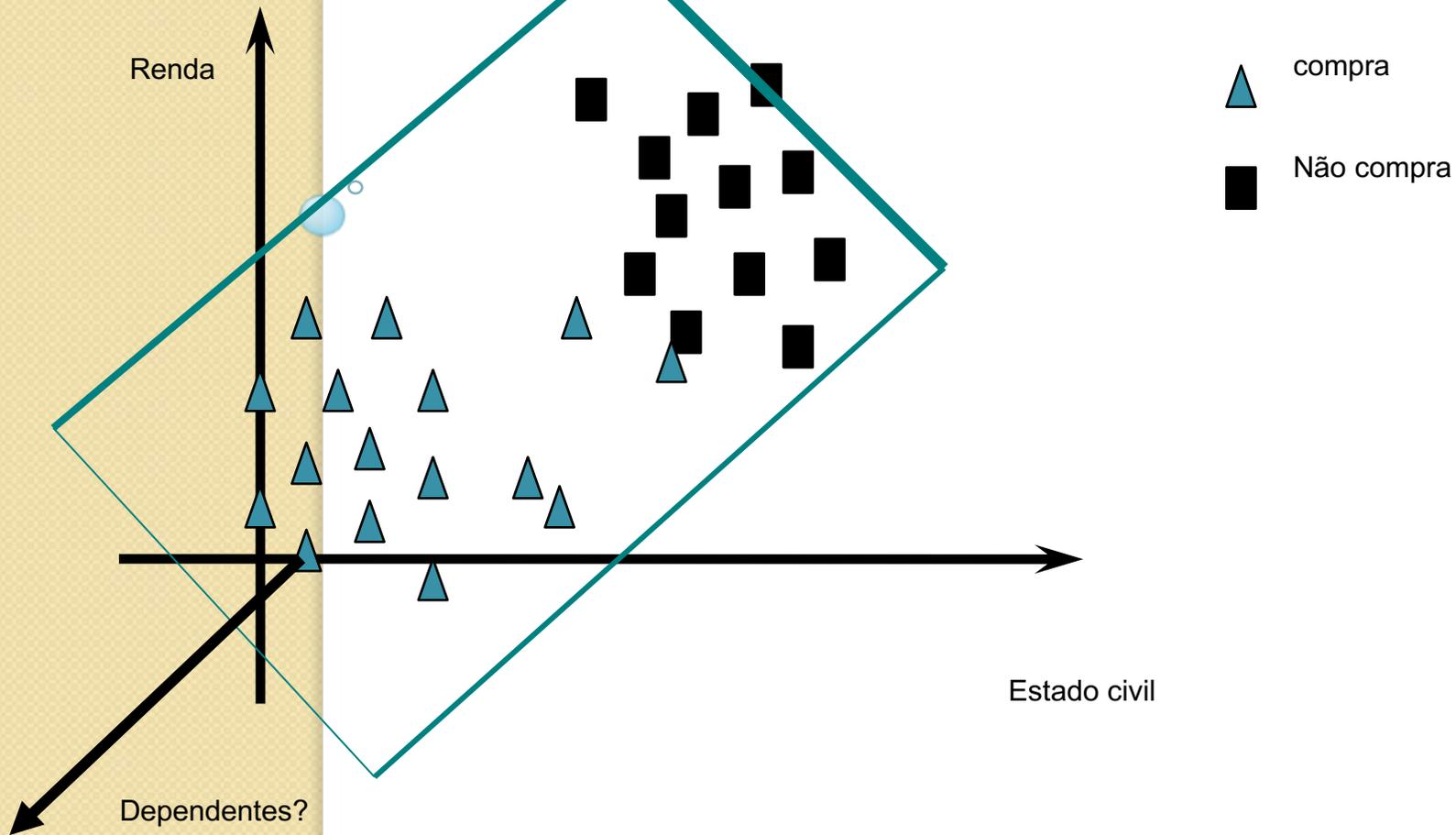


Classificadores lineares

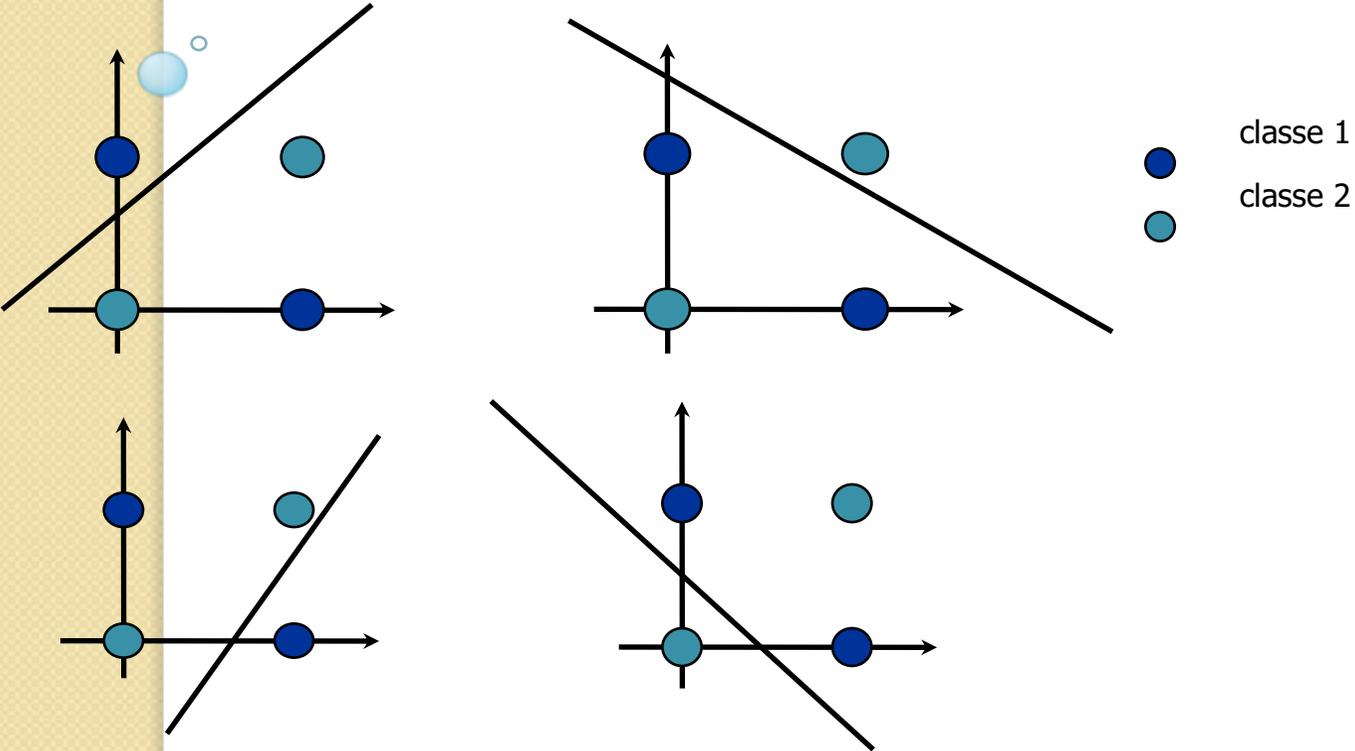


CLASSIFICAÇÃO	SINAL DA LINHA DE DECISÃO	
	d1	d2
Classe 1	+	+
Classe 2	+	-
Classe 2	-	+
Classe 2	-	-

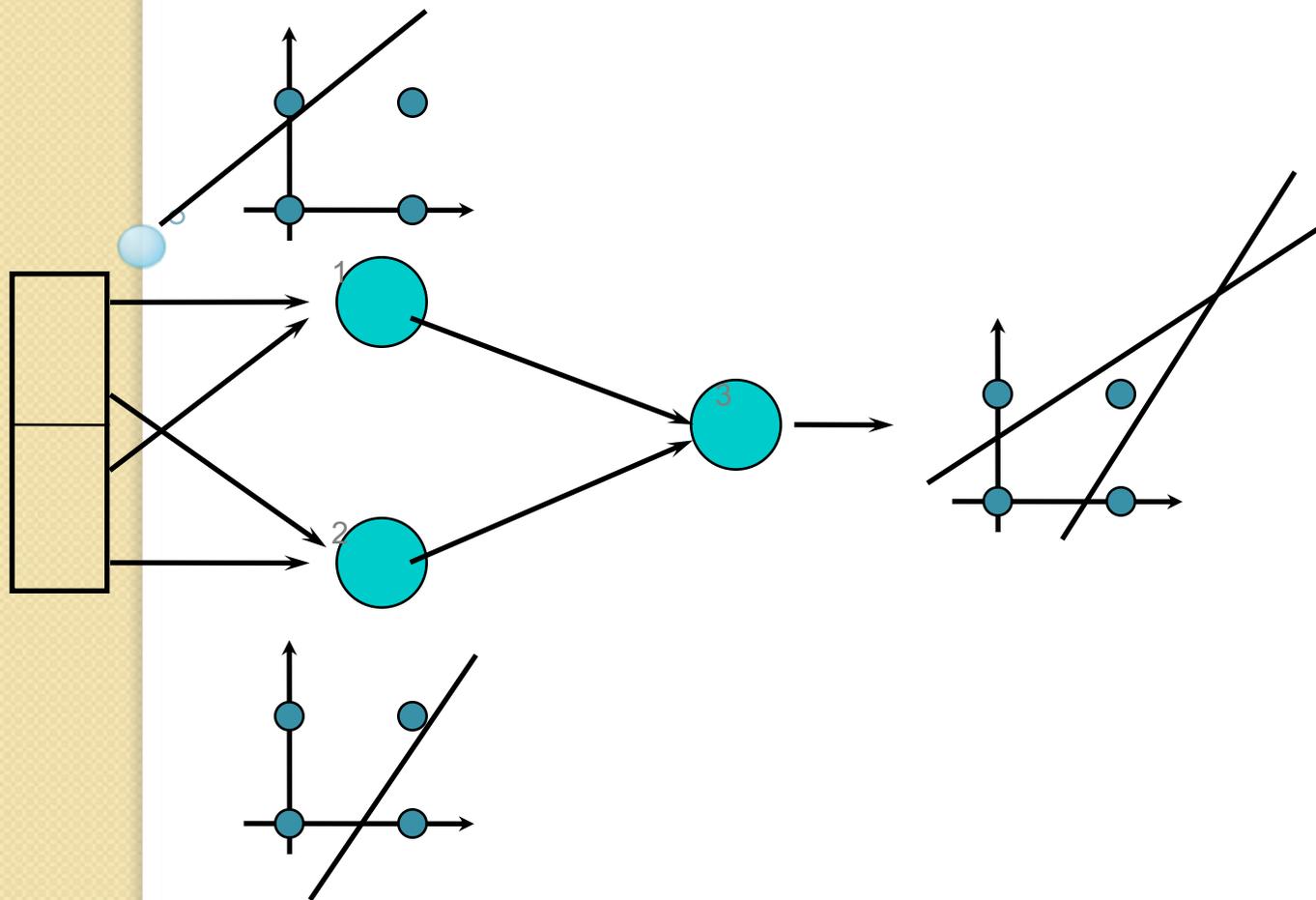
Classificadores Lineares



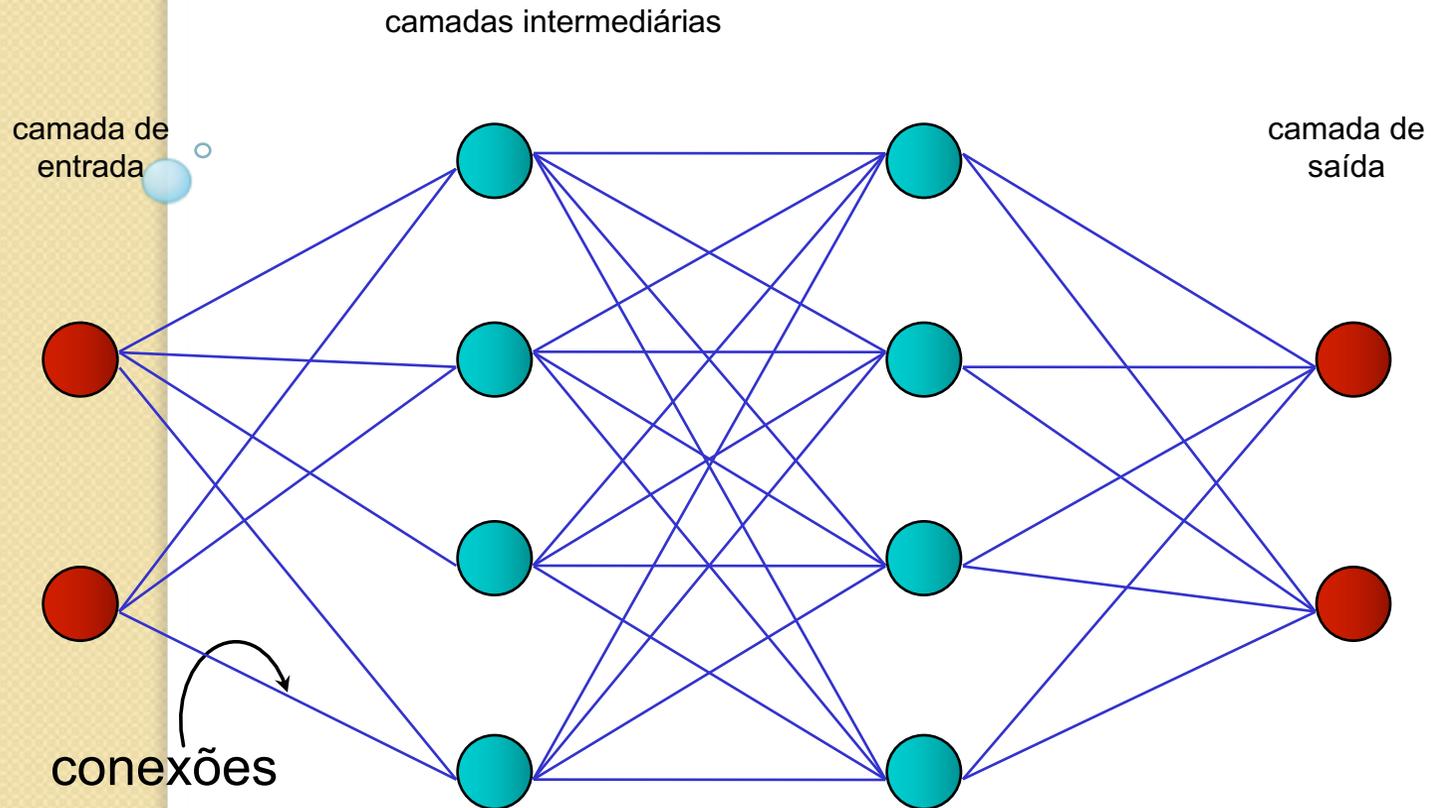
O Problema do Ou-exclusivo (XOR)



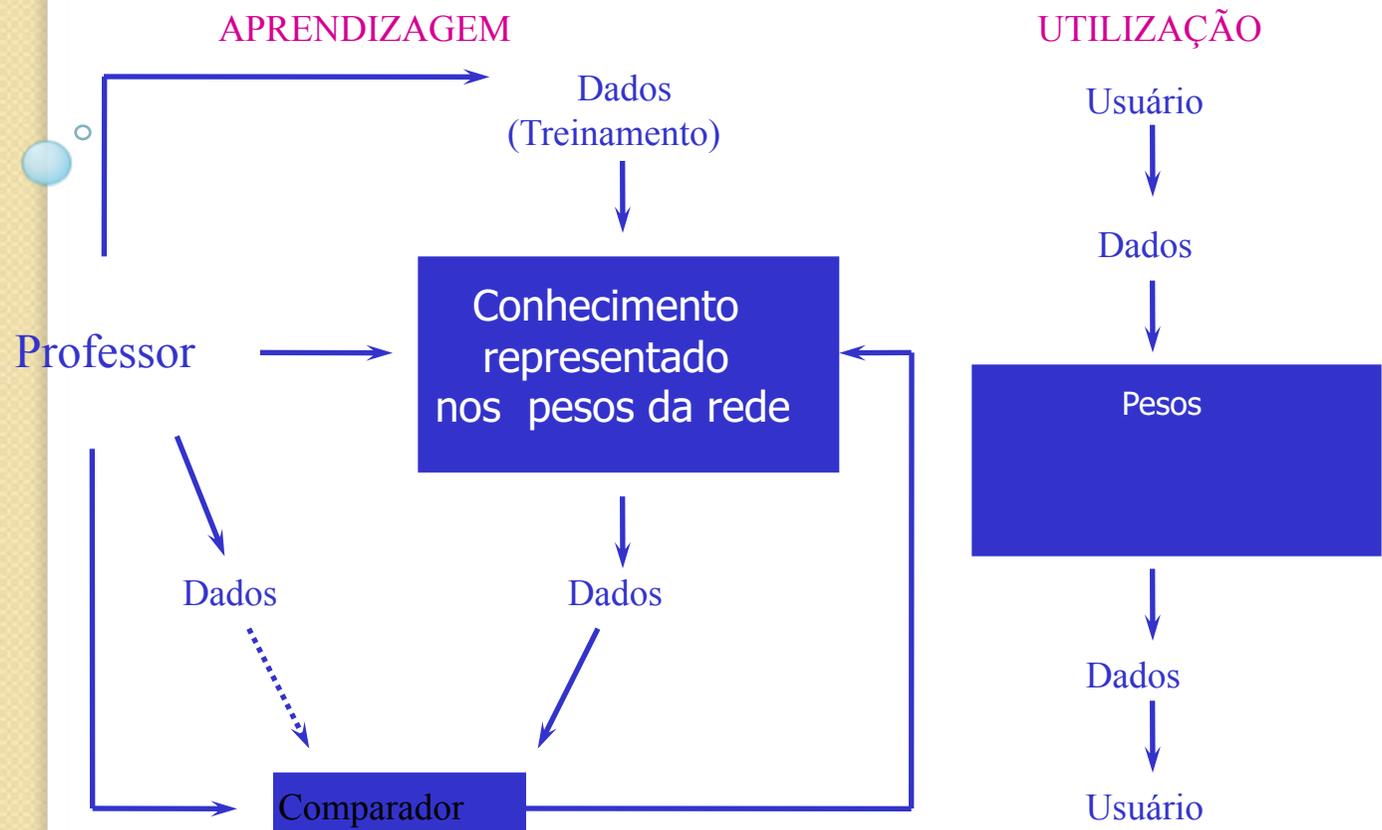
Solução para o XOR



Multilayer Perceptron (MLP) e Backpropagation (Regra Delta Generalizada)

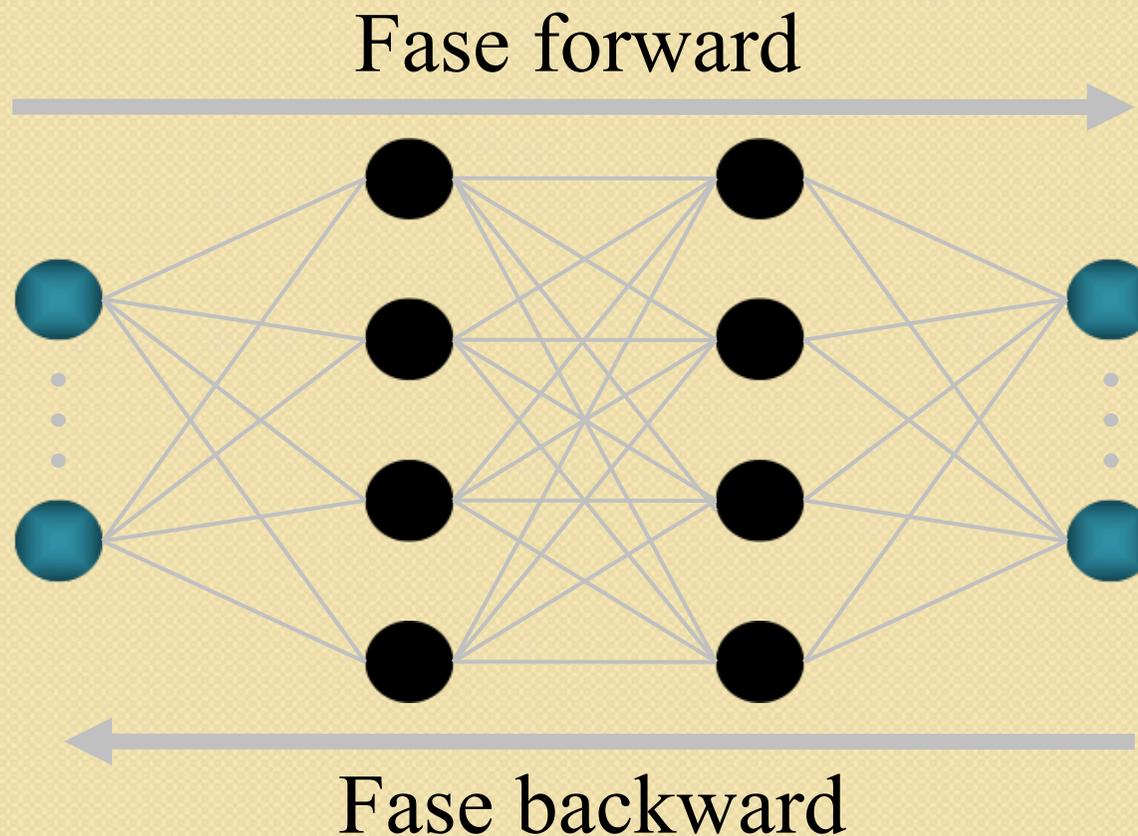


Funcionamento do MLP



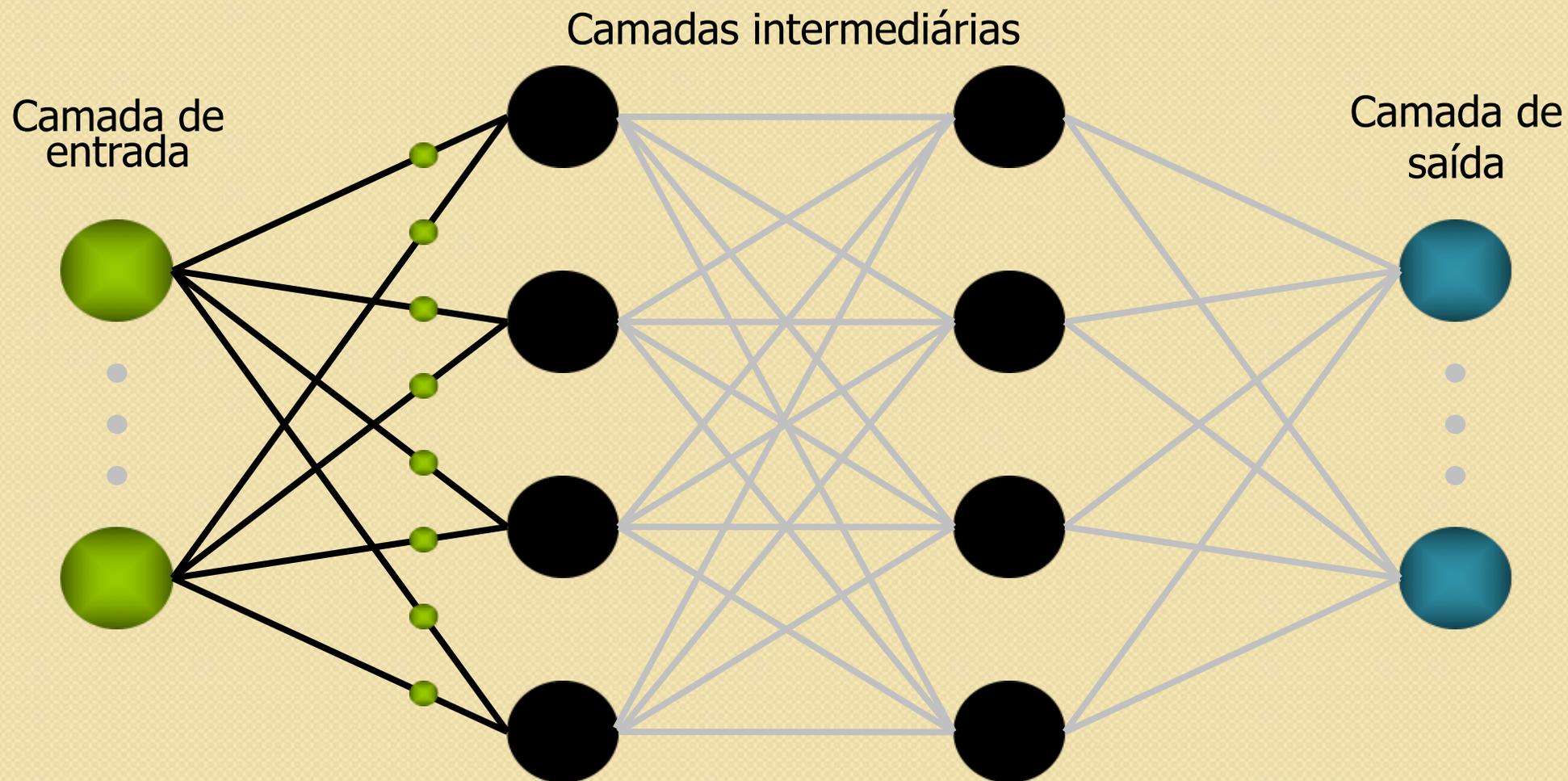
Algoritmo Backpropagation

- Treinamento em duas etapas:



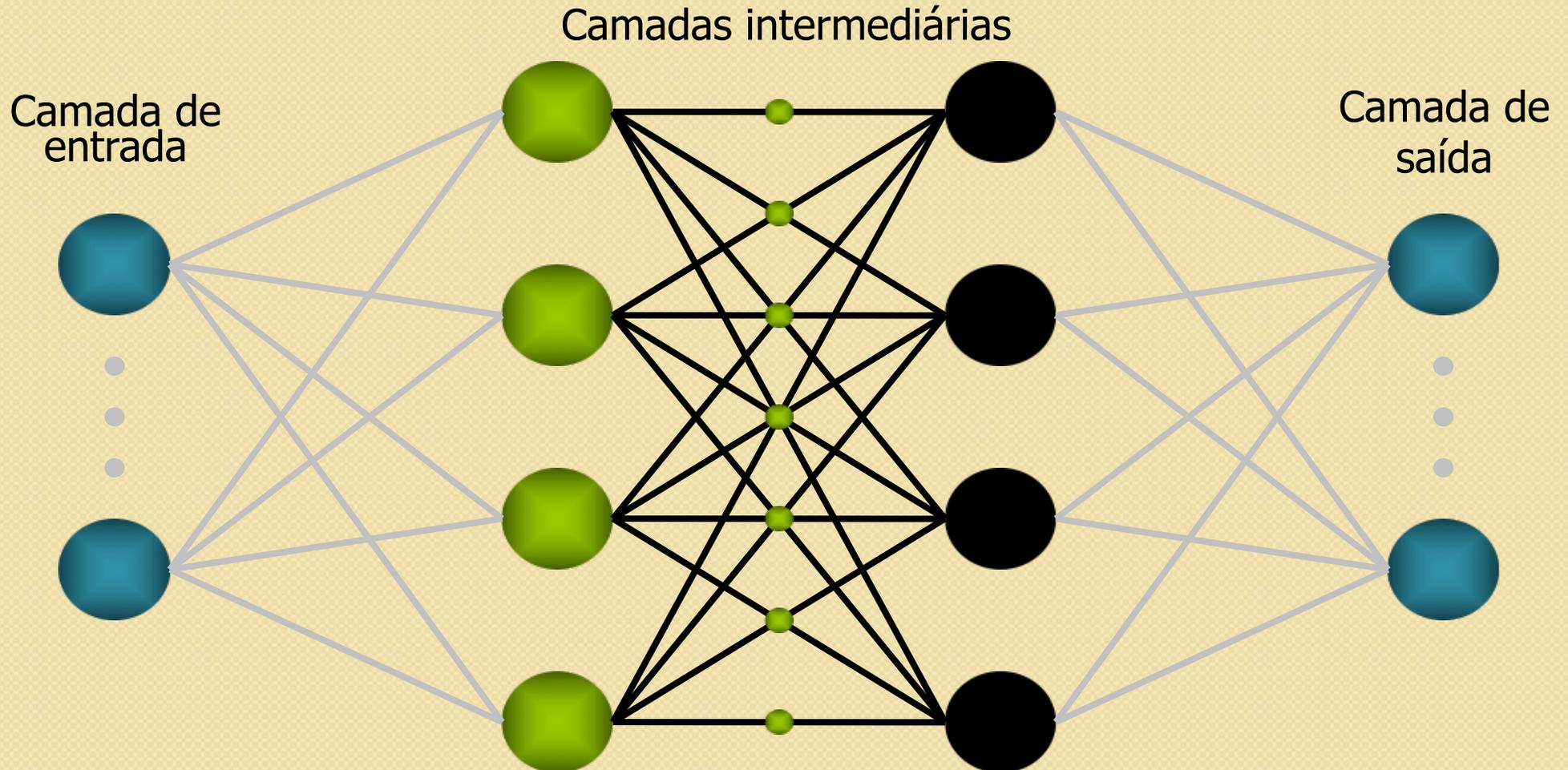
Fase forward

Entrada é apresentada à primeira camada da rede e propagado em direção às saídas



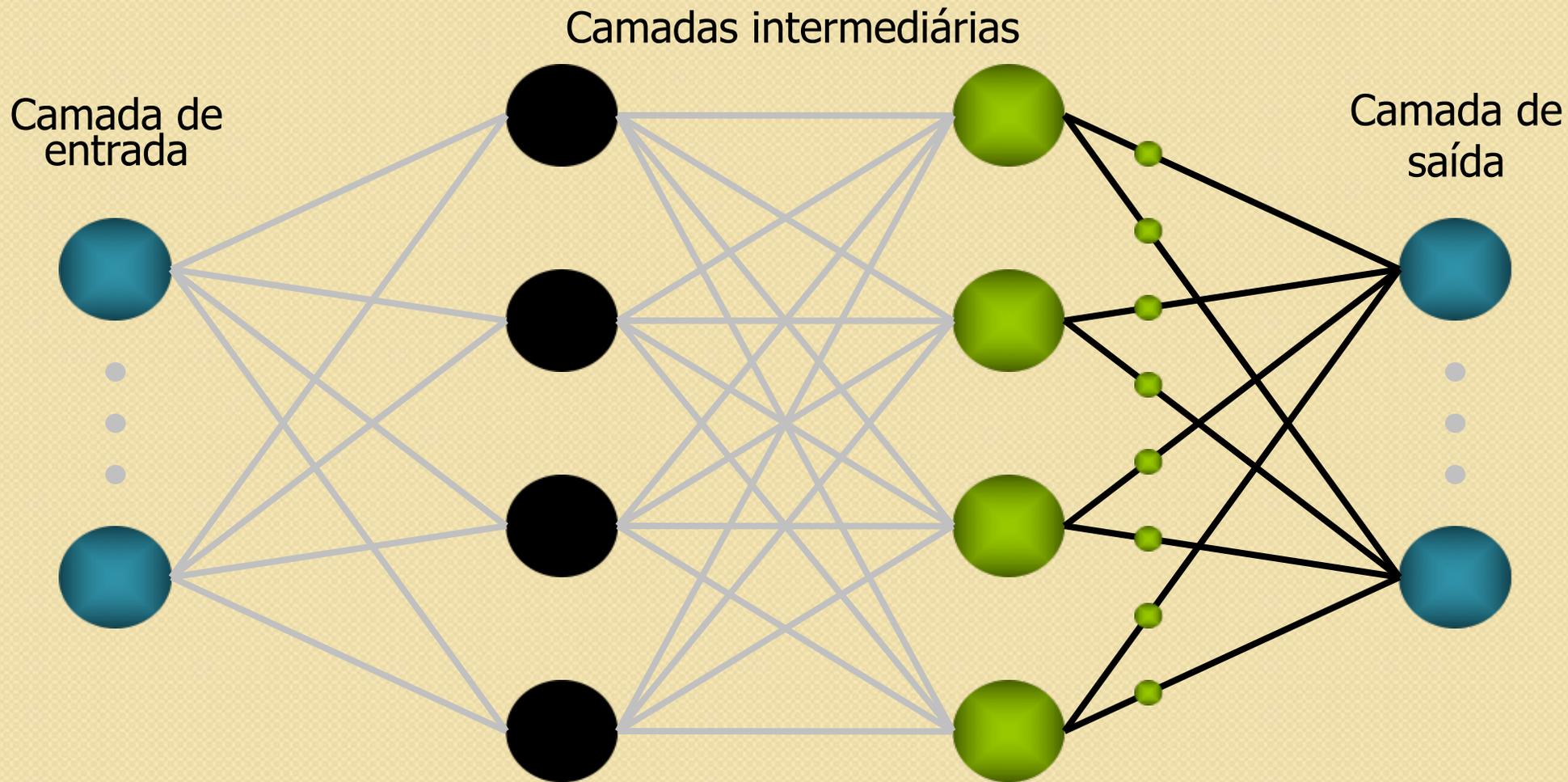
Fase forward

Os neurônios da camada i calculam
seus sinais de saída e propagam
à camada $i + 1$



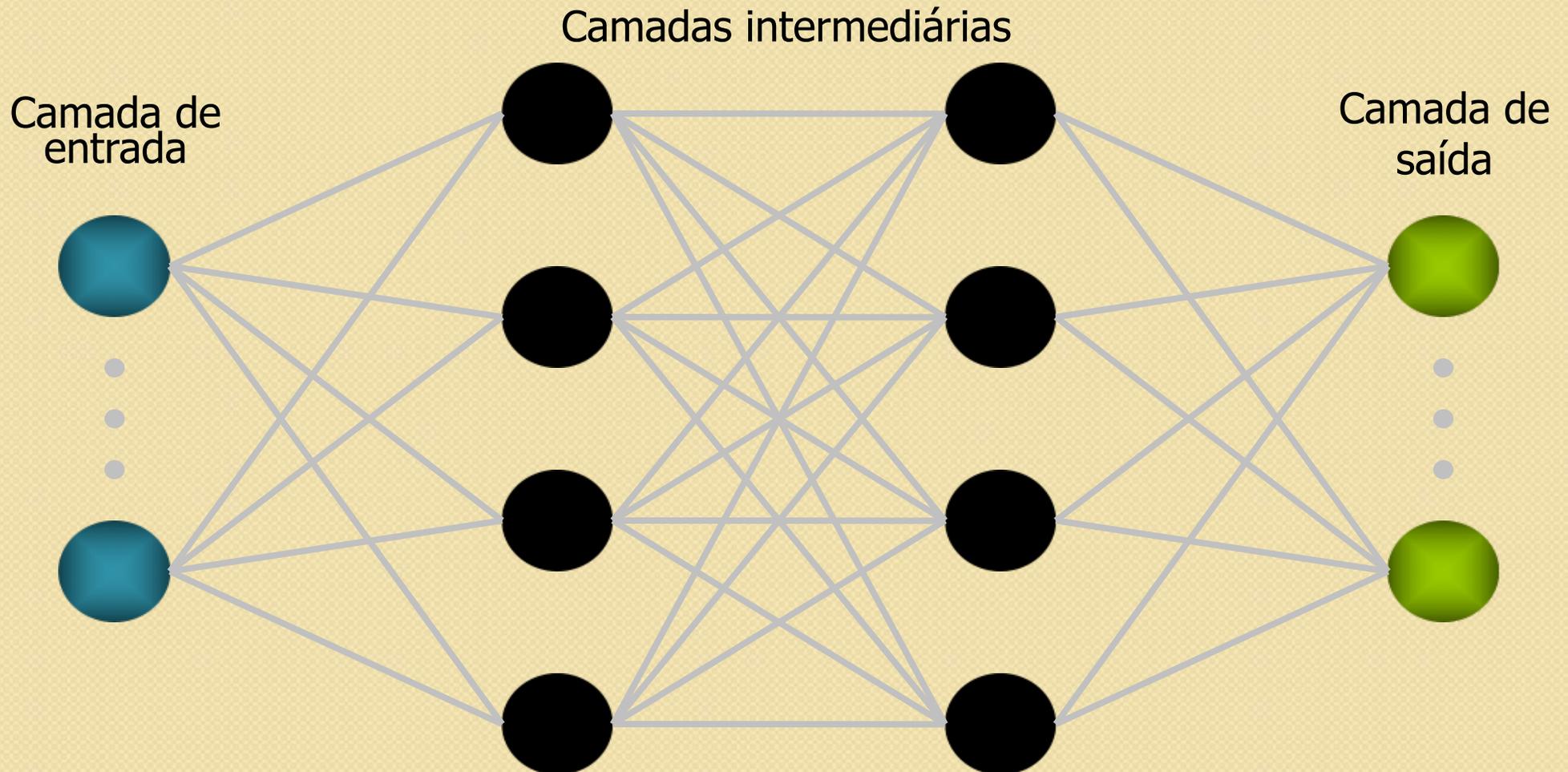
Fase forward

A última camada oculta calcula seus sinais de saída e os envia à camada de saída

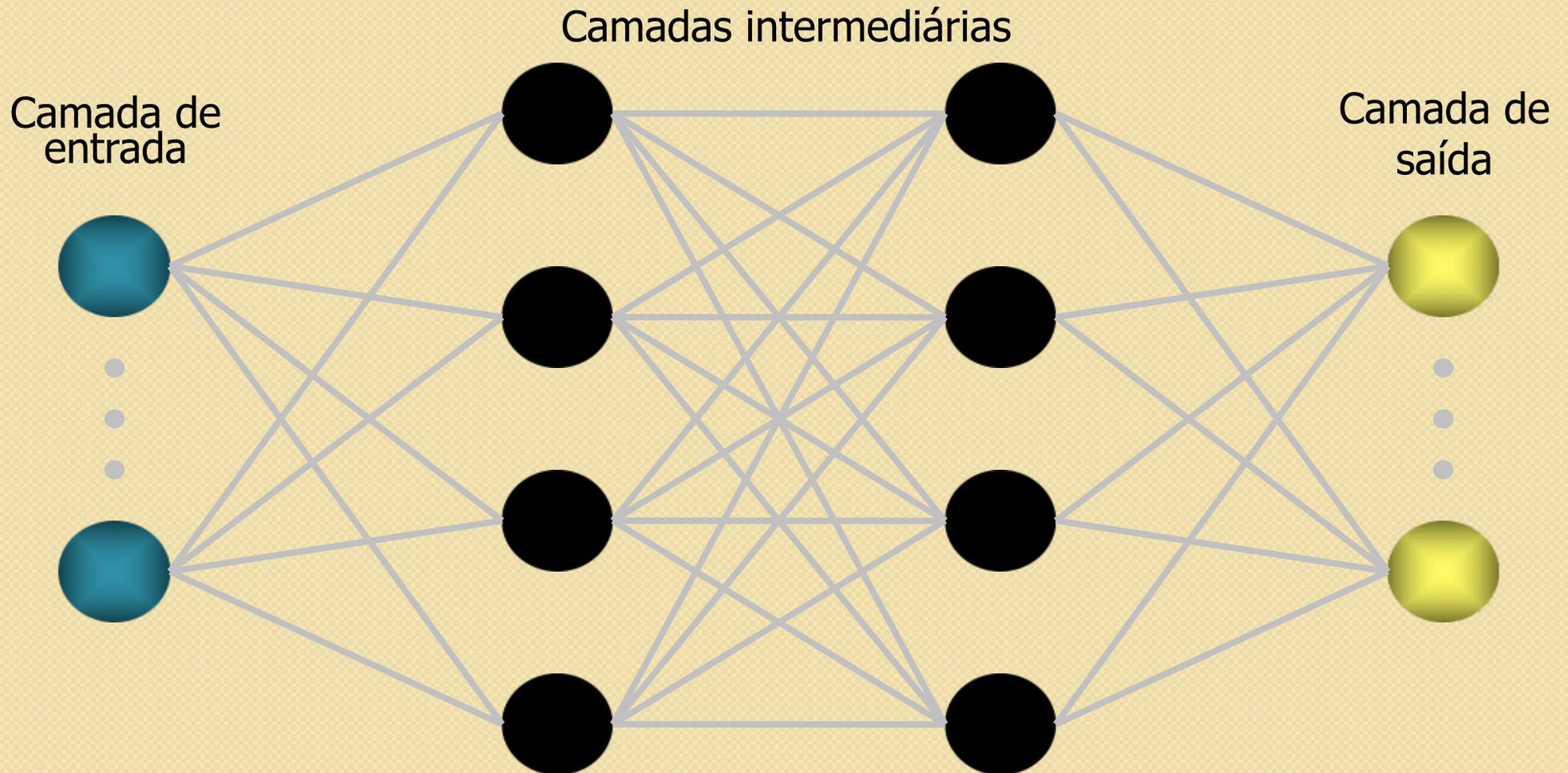


Fase forward

A camada de saída calcula os valores de saída da rede

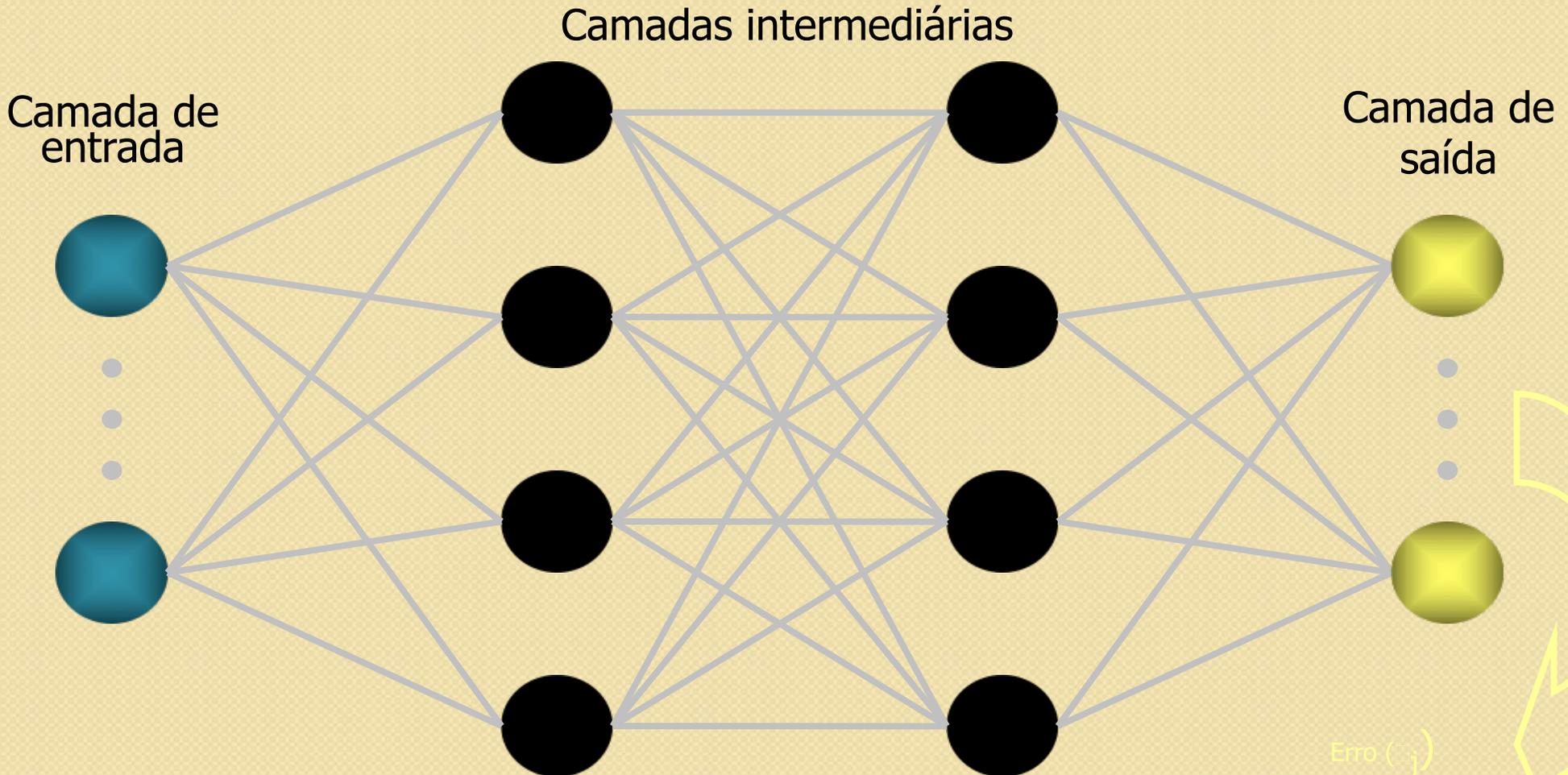


Fase backward



Fase backward

A camada de saída
calcula o erro da rede: E_j

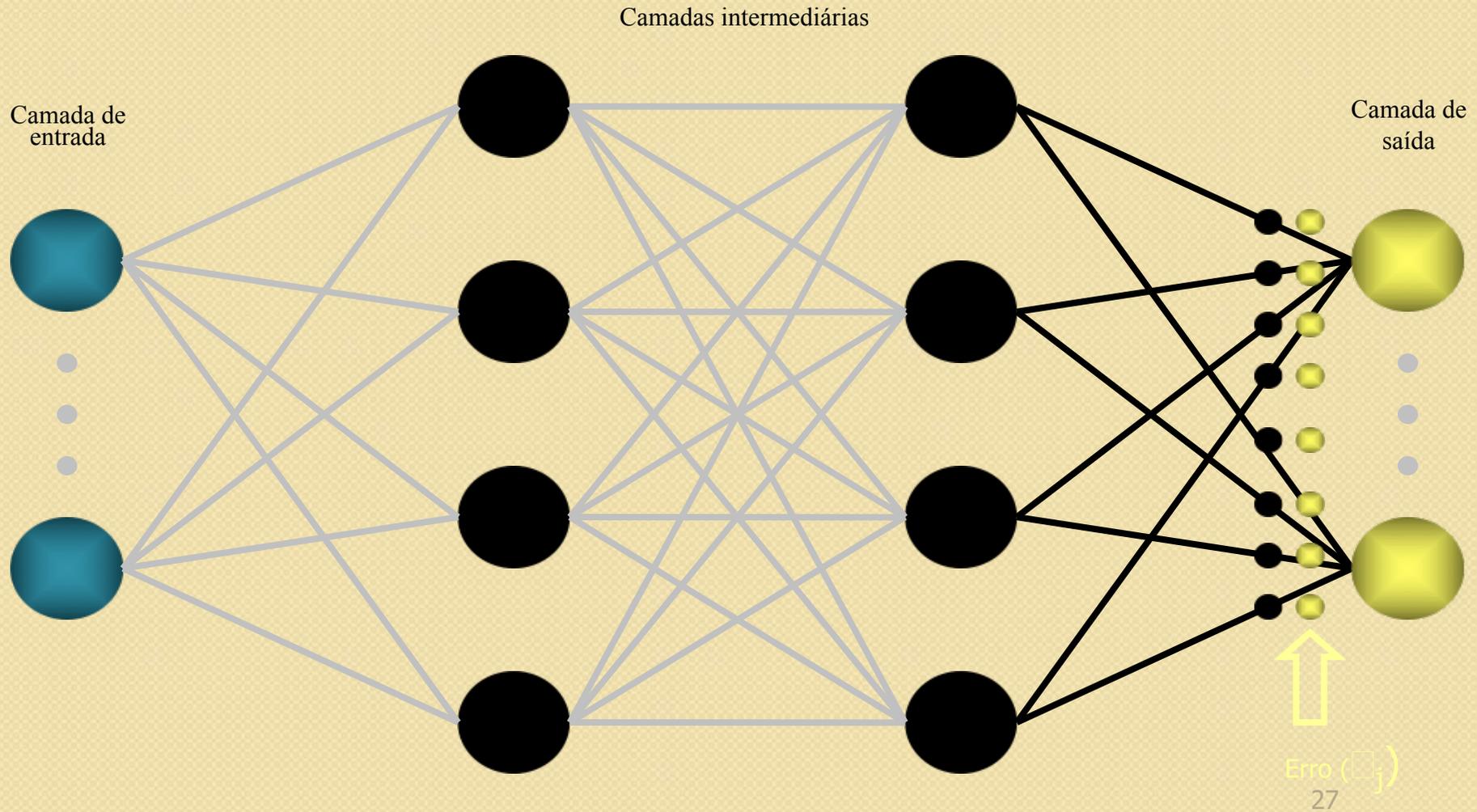


Calcula o termo de correção dos pesos

(a atualização será feita depois)

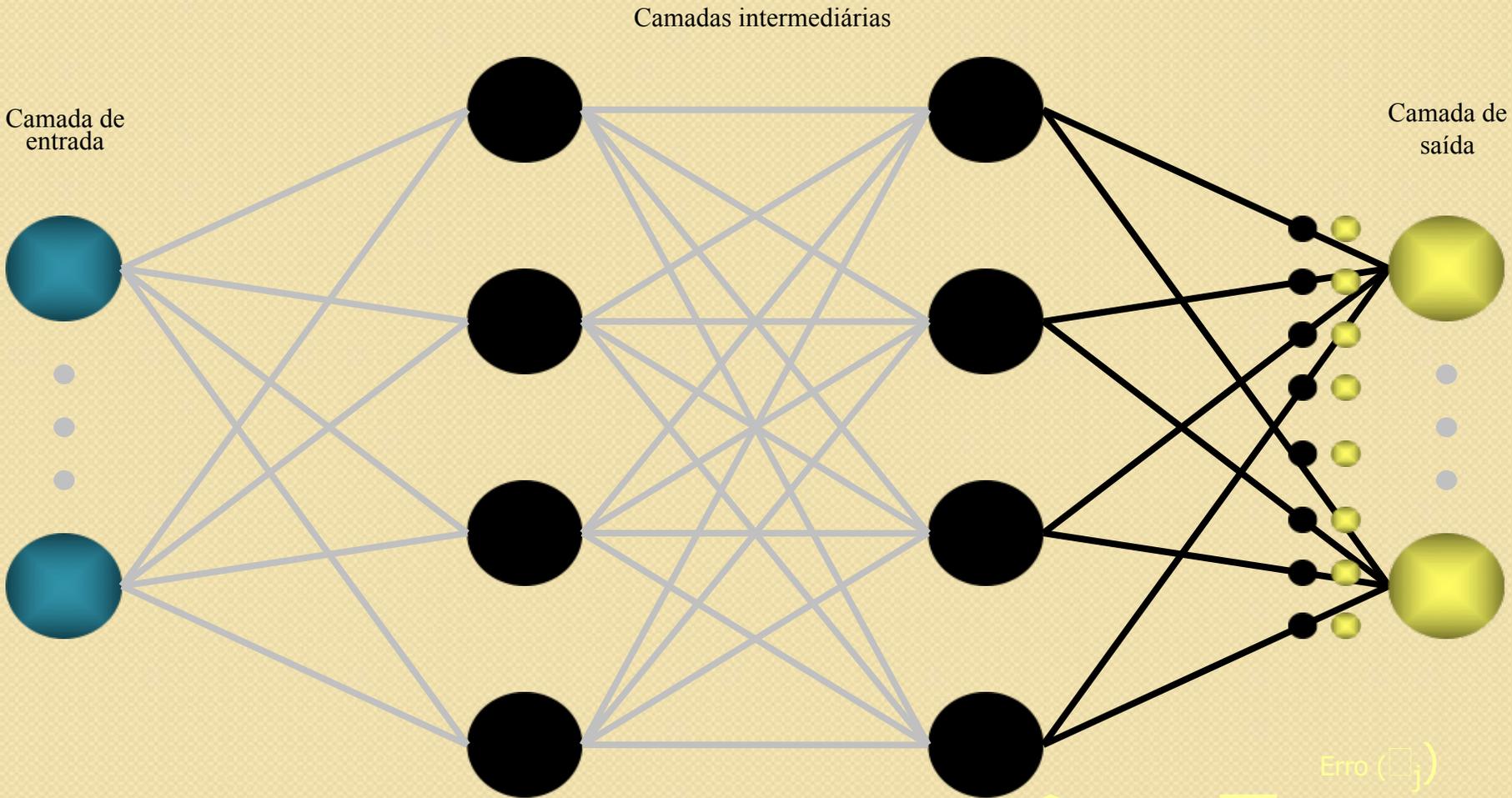
$$\Delta w_{ji} = \Delta_j x_i$$

Fase backward



Envia o erro para a última camada oculta

Fase backward

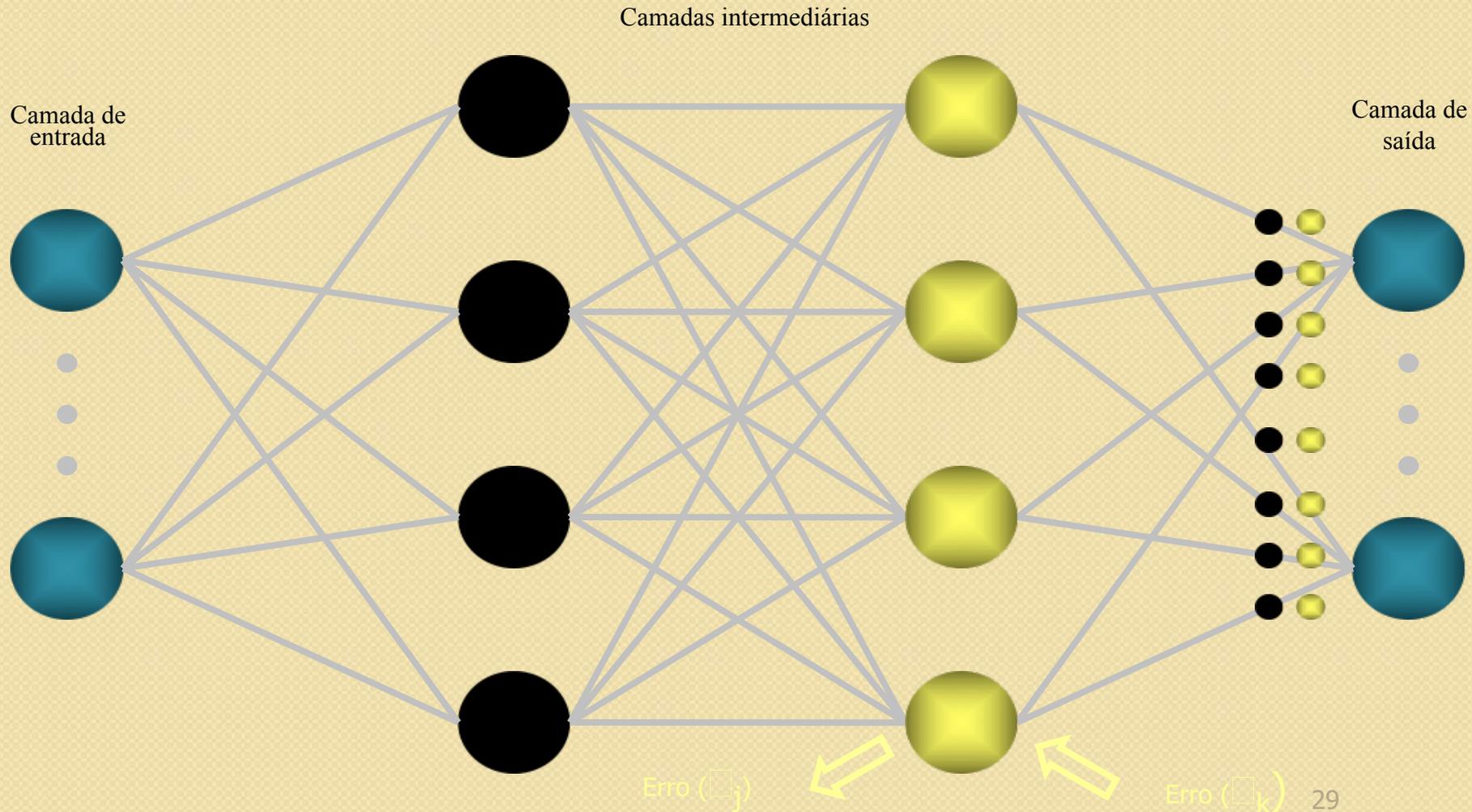


Erro (e_j)

A camada oculta calcula o seu erro

$$e_j = f'(u_j) \cdot \sum_k W_{jk} e_k$$

Fase backward

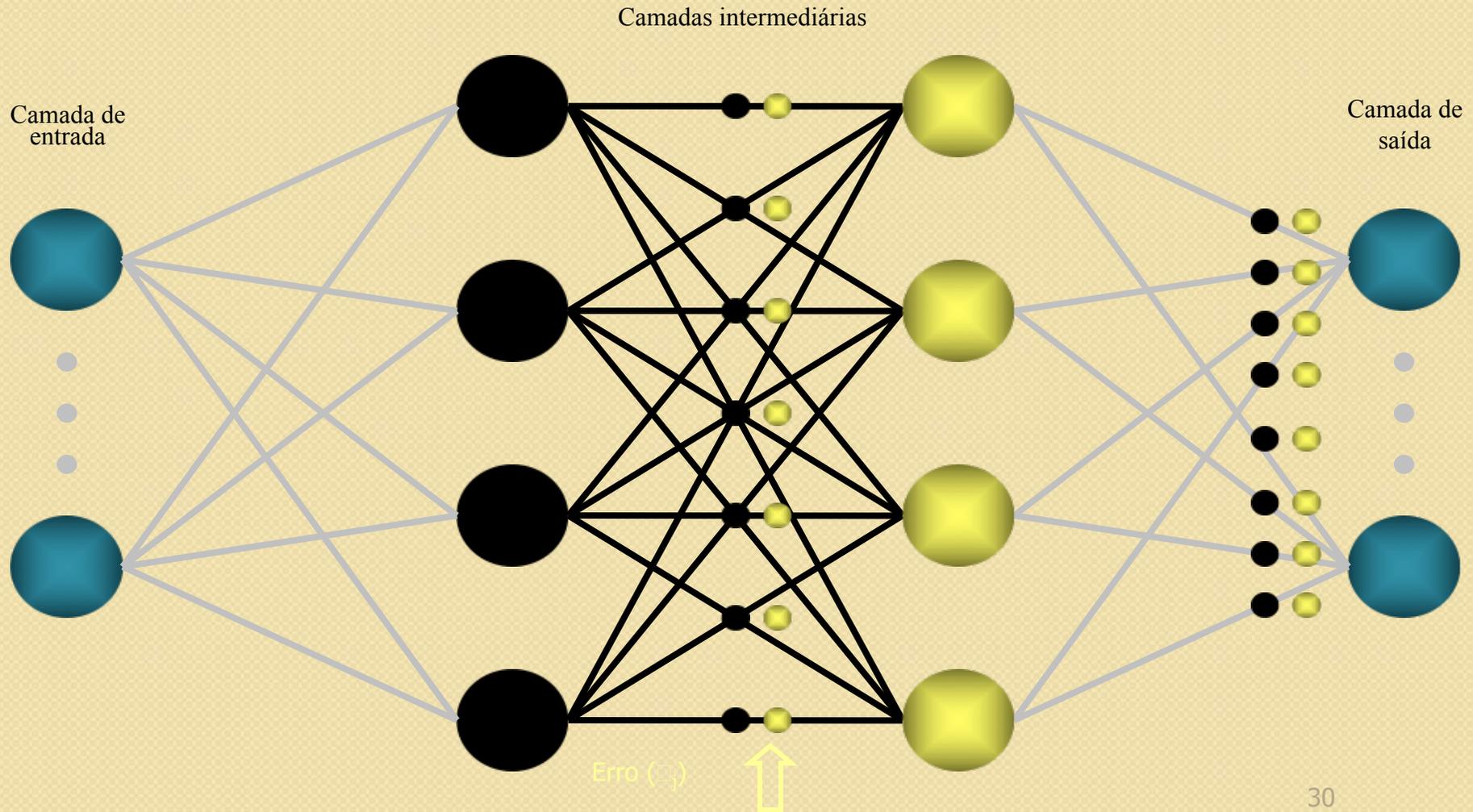


Calcula o termo de correção dos pesos

(a atualização será feita depois)

$$\Delta w_{ij} = \Delta \delta_j x_i$$

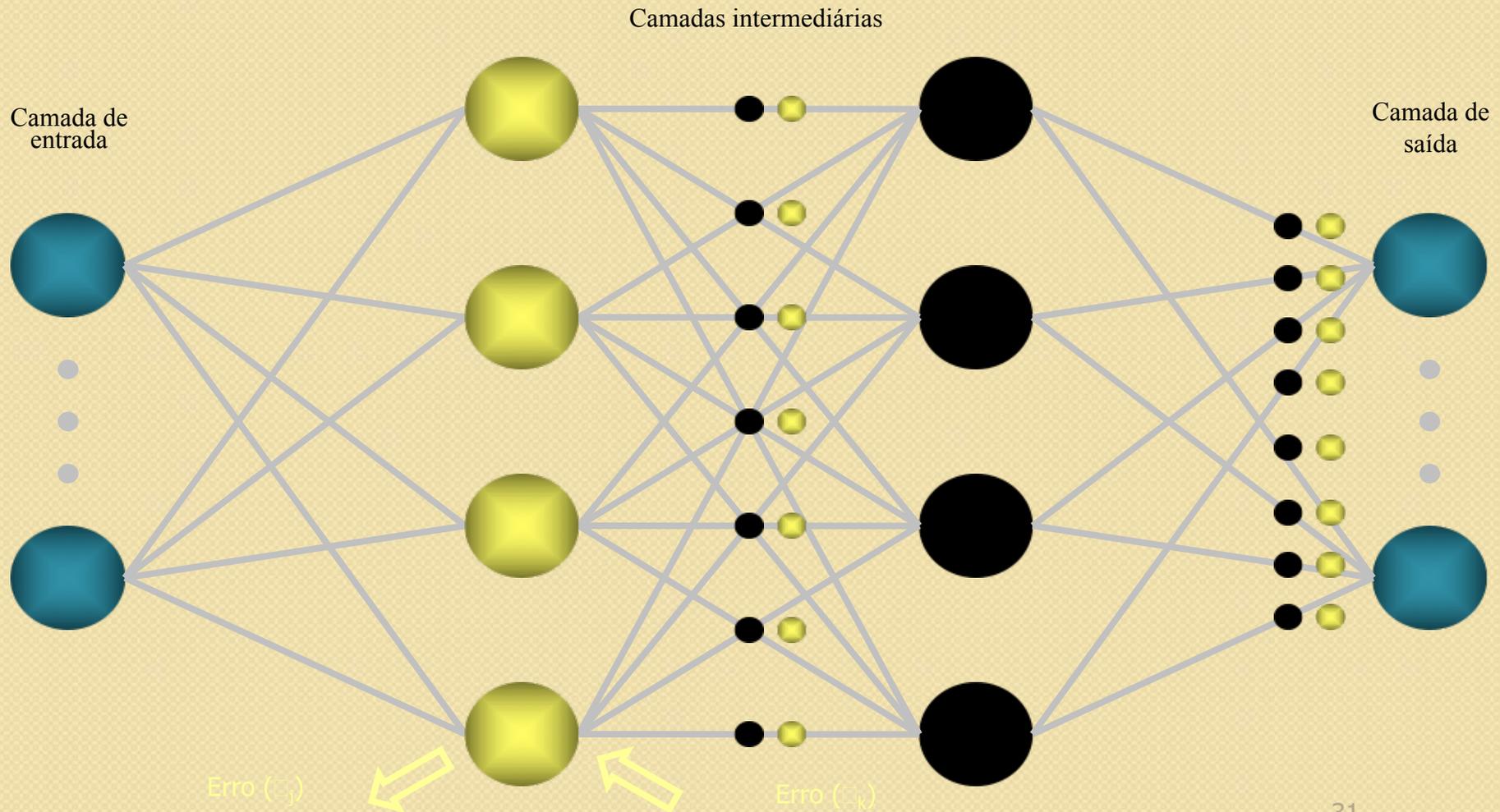
Fase backward



A camada oculta calcula o seu erro

$$\delta_j = f'(u_j) \cdot \delta_k W_{jk}$$

Fase backward

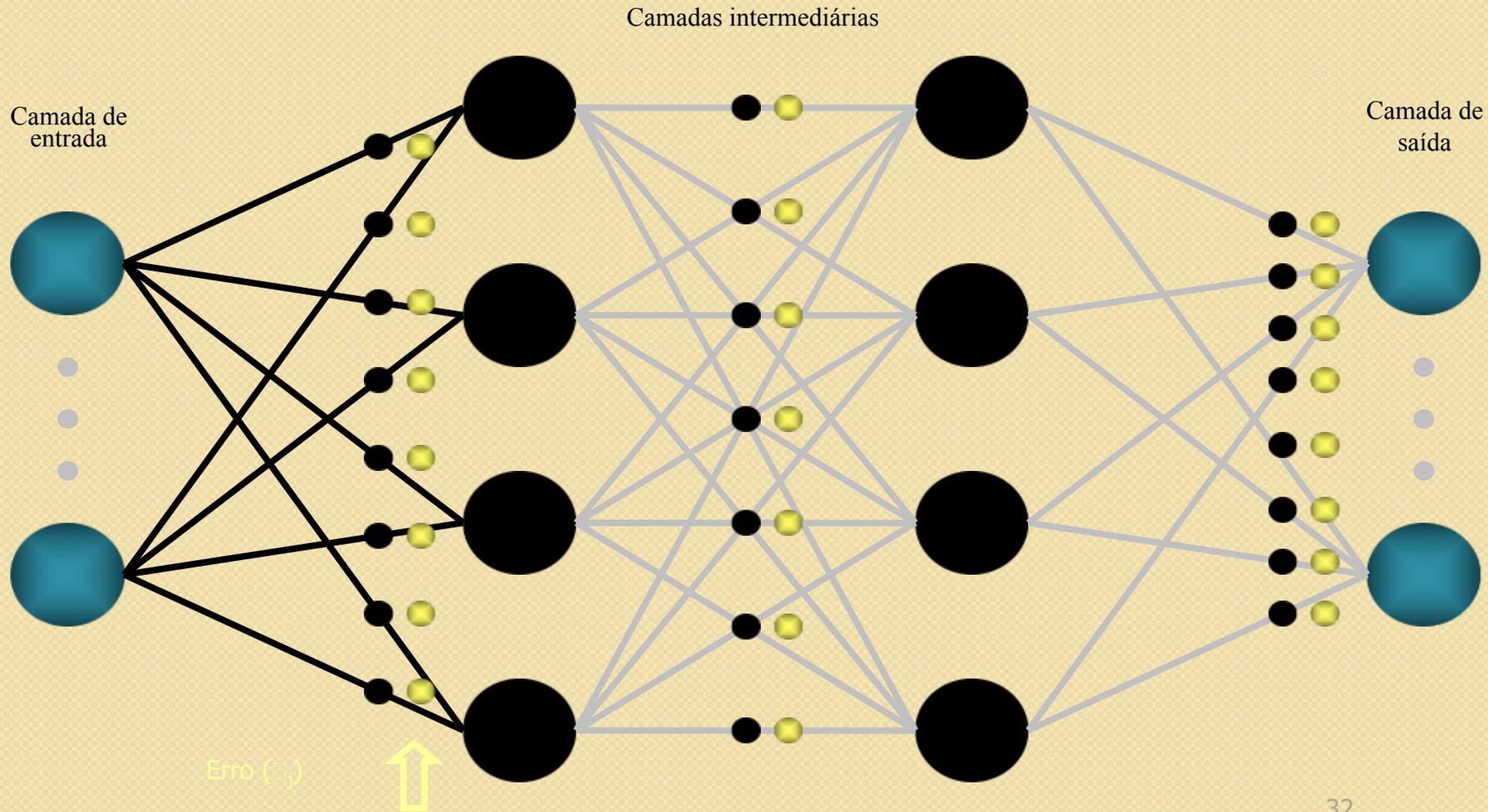


Calcula o termo de correção dos pesos

(a atualização será feita depois)

$$\Delta w_{ij} = \Delta_j x_i$$

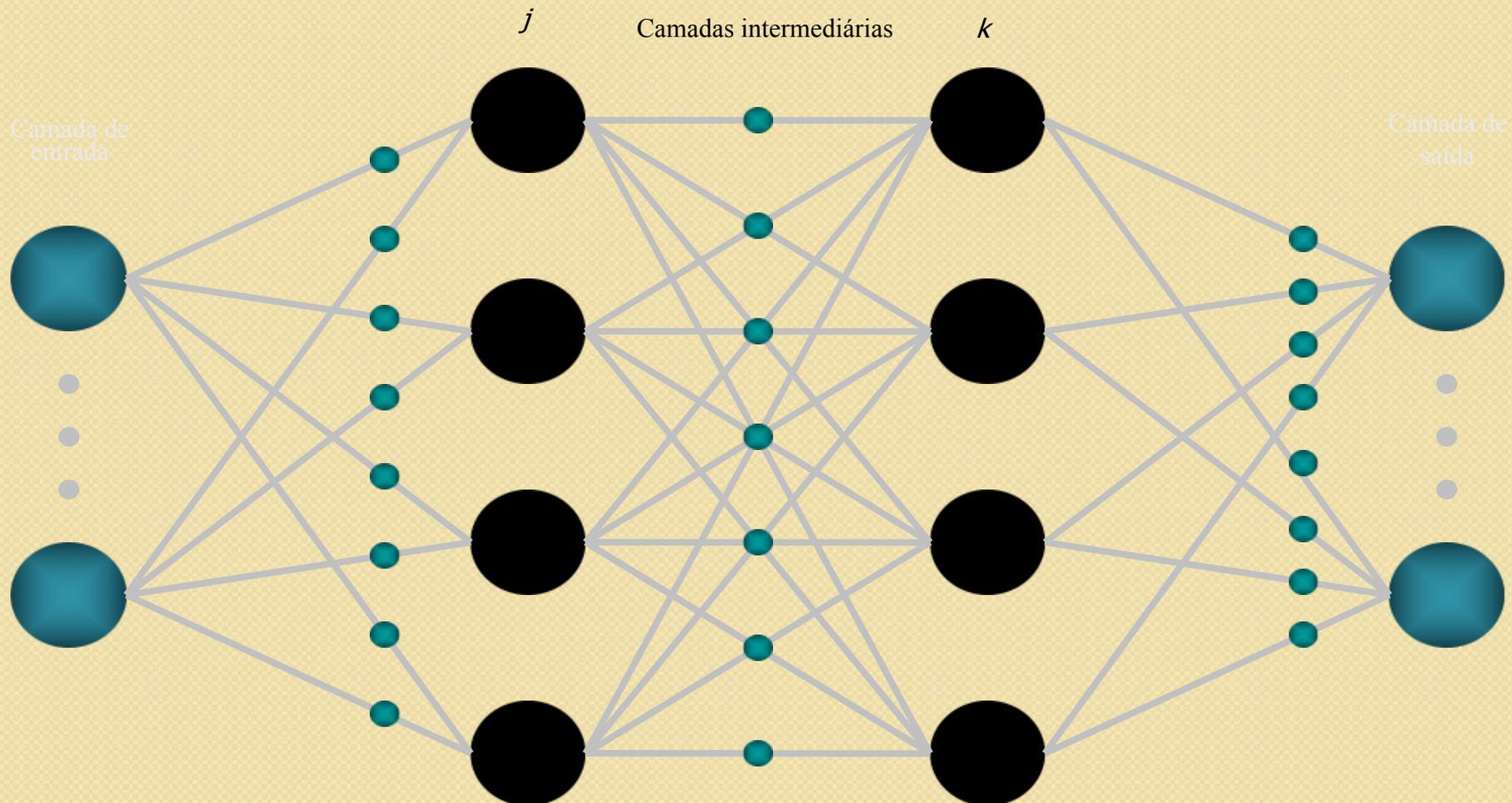
Fase backward



Cada unidade atualiza seus pesos

$$w_{ij}(\text{novo}) = w_{ij}(\text{velho}) + \Delta w_{jk}$$

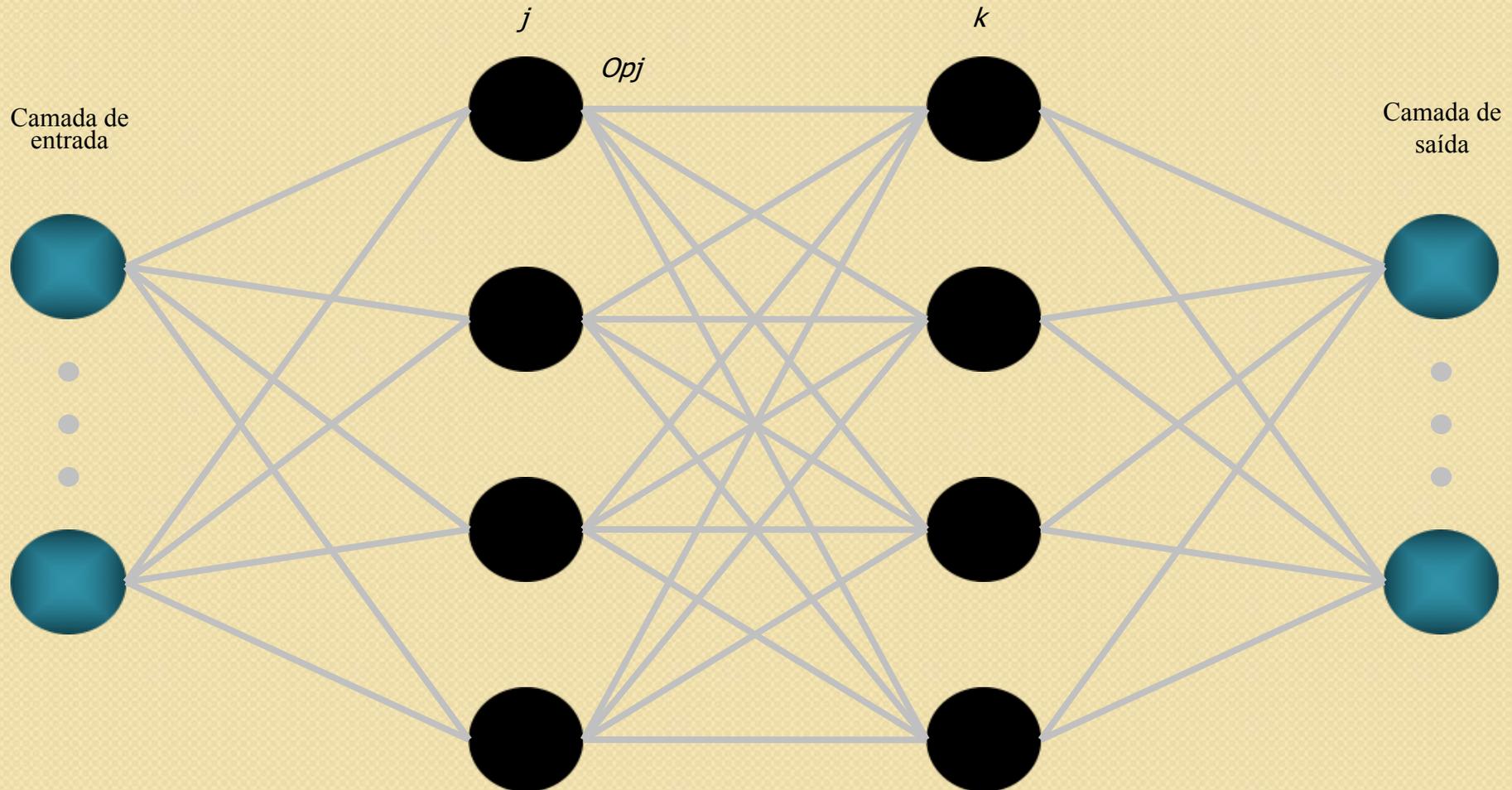
Fase backward



Repete-se o processo enquanto
enquanto a rede não aprender
o padrão de entrada

Error-Backpropagation

$$net_{pk} = \sum o_{pj} \cdot w_{jk}$$



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 :

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

Regra Delta Generalizada ou Error-Back Propagation


$$\Delta_p W_{ji} \propto \delta_j \cdot O_{pi}$$


$$\Delta_p W_{ji} = \eta \cdot \delta_j \cdot O_{pi}$$

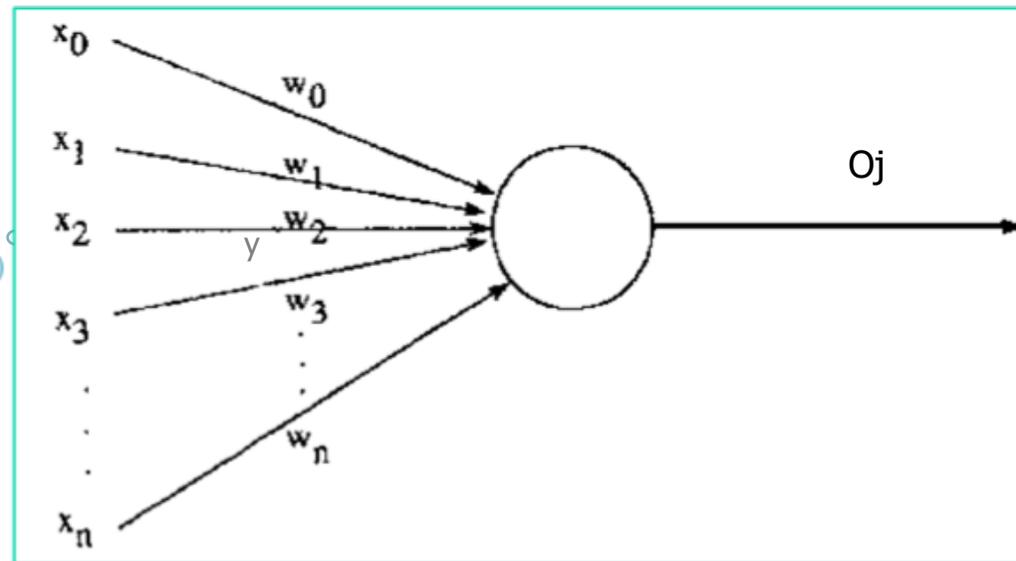
Regra Delta Generalizada ou Error-Back Propagation



Dois casos precisam ser considerados para δ_{pj}

Neurônio McCulloch-Pitts

$W \rightarrow net_{pj} \rightarrow O_j \rightarrow E$



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

\uparrow
 net_{pj}

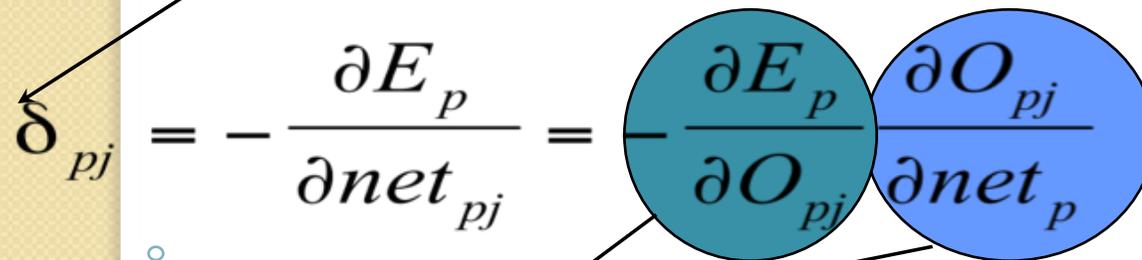
$$net_{pj} = \sum W_{jk} O_{pk}$$

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

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

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

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

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


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

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

(2 casos precisam ser considerados!)

Primeiro caso : j é uma unidade de saída

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

$$\delta_{pj} = (t_{pj} - o_{pj}) f_j'(net_{pj})$$

Segundo caso : j é uma unidade intermediária

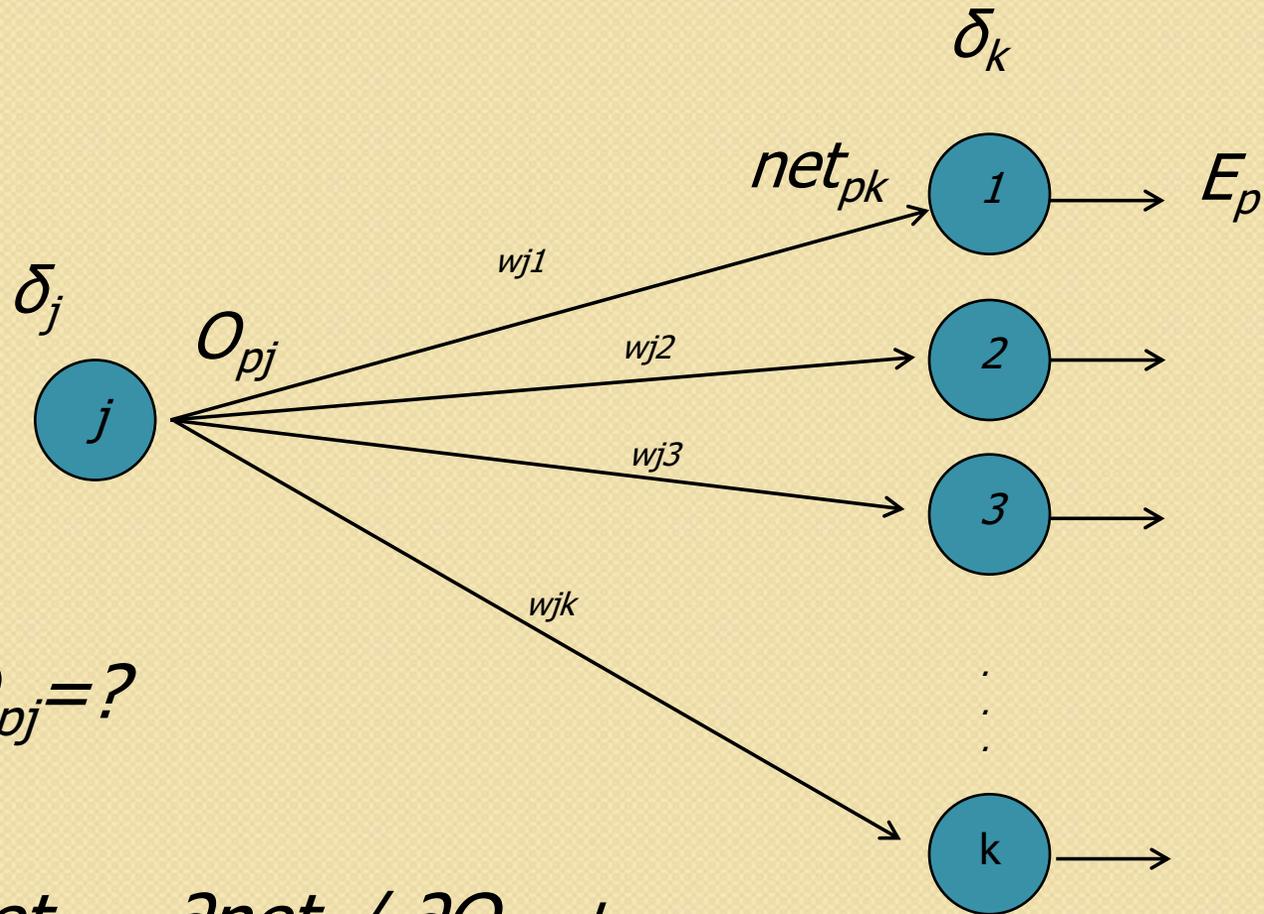
$$\frac{\partial E_p}{\partial O_{pj}} = \sum_k \frac{\partial E_p}{\partial net_{pk}} \frac{\partial net_{pk}}{\partial O_{pj}}$$

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

$$\Rightarrow \frac{\partial E_p}{\partial O_{pj}} = - \sum_k \delta_{pk} w_{jk}$$

$$\delta_{pj} = f'_j(net_{pj}) \sum_k \delta_{pk} w_{jk}$$

Saida da unidade j influencia cada uma das unidades na camada seguinte

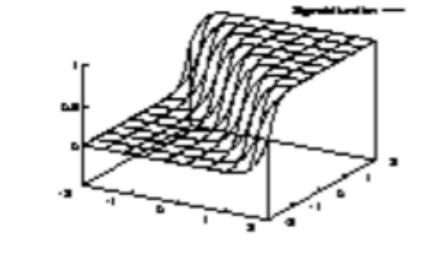


$$\frac{\partial E_p}{\partial O_{pj}} = ?$$

1 $\frac{\partial E_p}{\partial net_{pk}} \cdot \frac{\partial net_{pk}}{\partial O_{pk}} + \dots$

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

Outra Função Erro: Entropia Cruzada (Cross-Entropy)

$$C = -\frac{1}{n} \sum_x [y \ln a + (1 - y) \ln(1 - a)],$$

De uma maneira geral...

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



No caso de **classificação**

$$Y_k = 1 \text{ se } X_p \in k$$

0, caso contrário

No caso em que Y_k são variáveis contínuas

- problema de regressão
- ou problema de **aproximação de funções**

Reconhecimento de Padrões

Verificação

x_p

$\square?$

C_i

Classificação

$\square?$

C_1

C_2

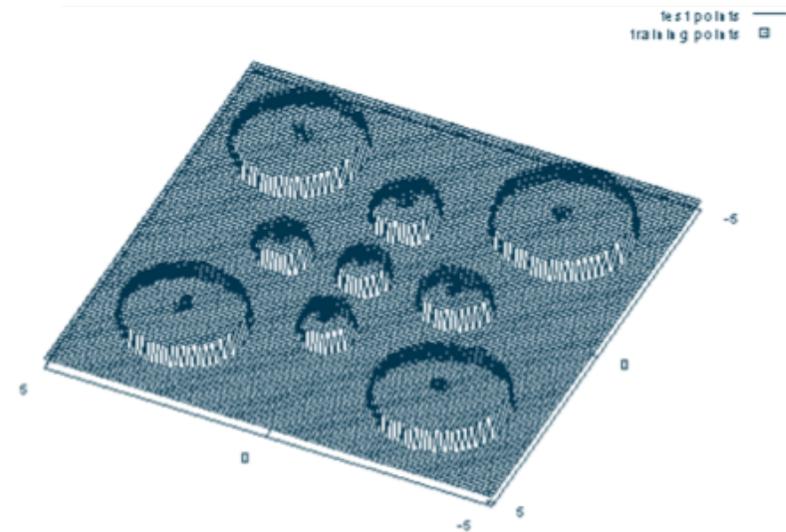
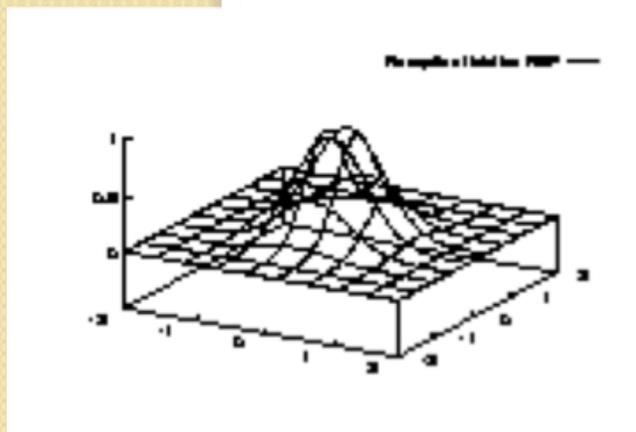
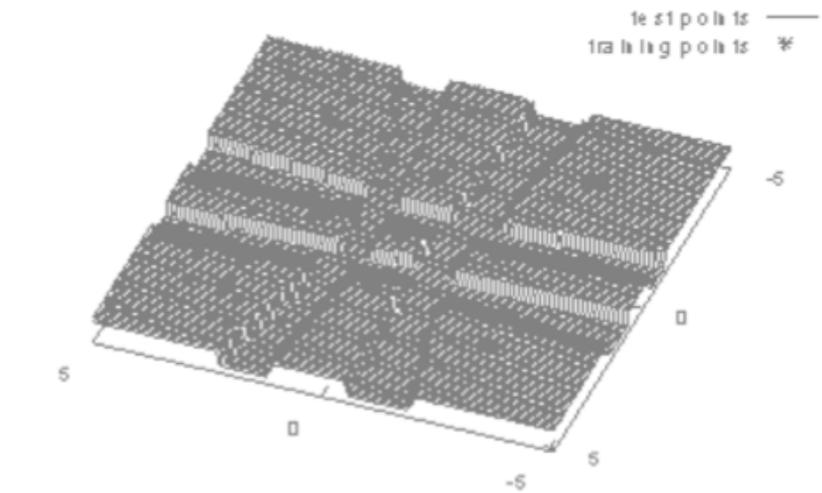
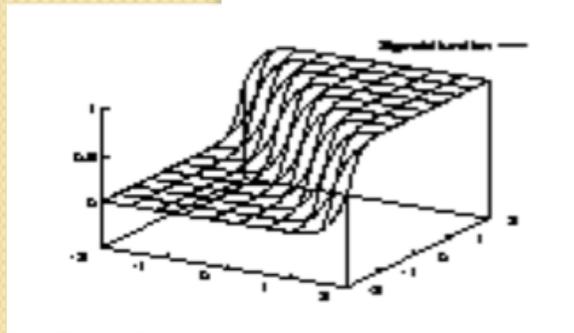
C_3

C_n

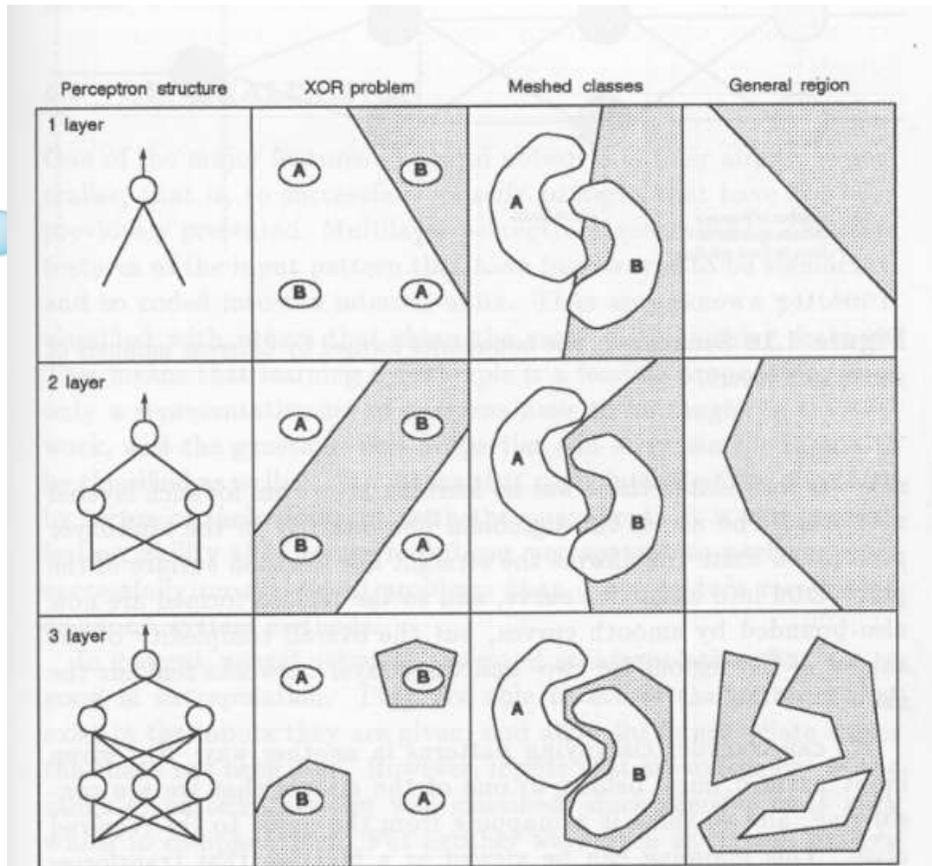
x_p

$$y_k = y_k(x_p; w) \\ k=1, 2, \dots, n$$

Reconhecimento de Padrões

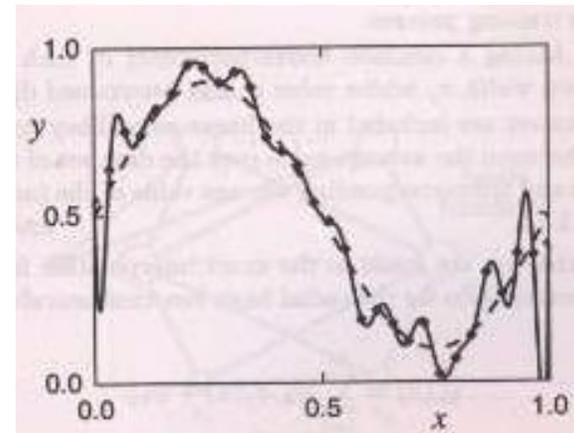
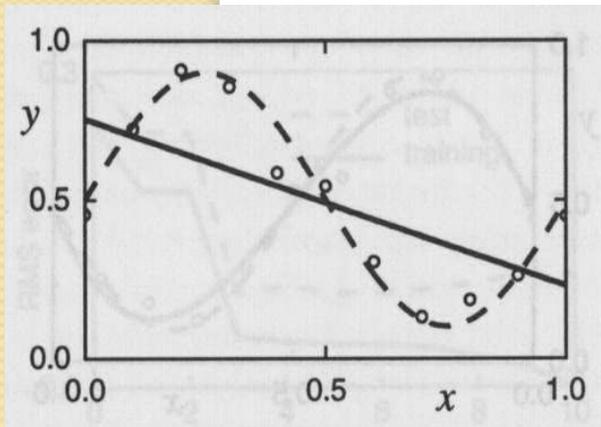
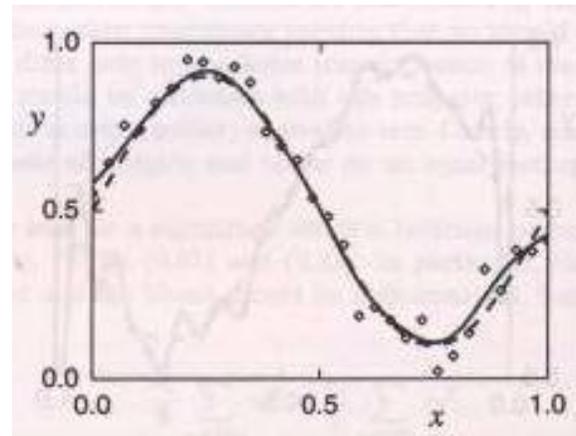


Complexidade Funcional do MLP x Número de Camadas

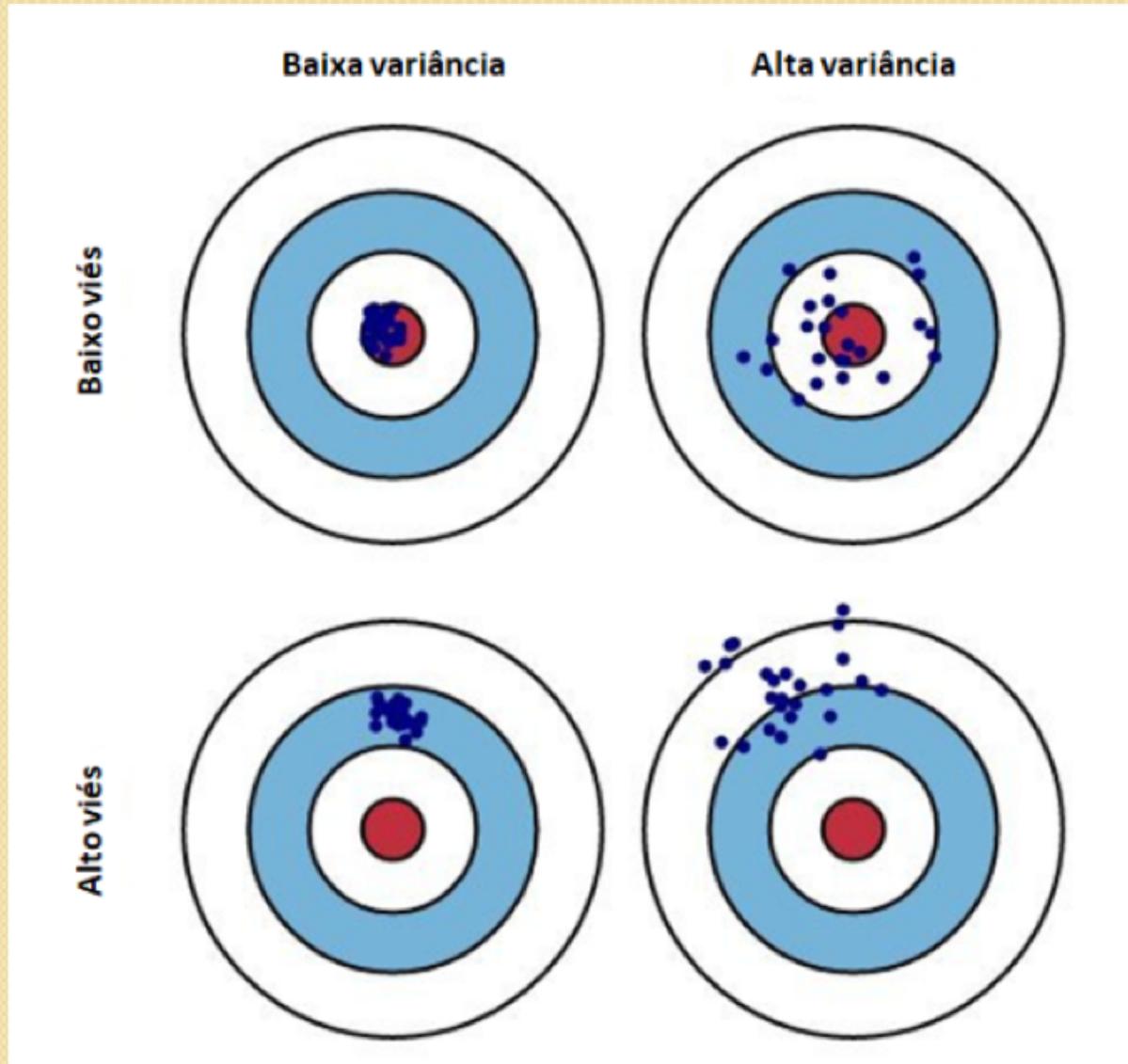


(after Lippmann, IEEE ASSP April 1987)

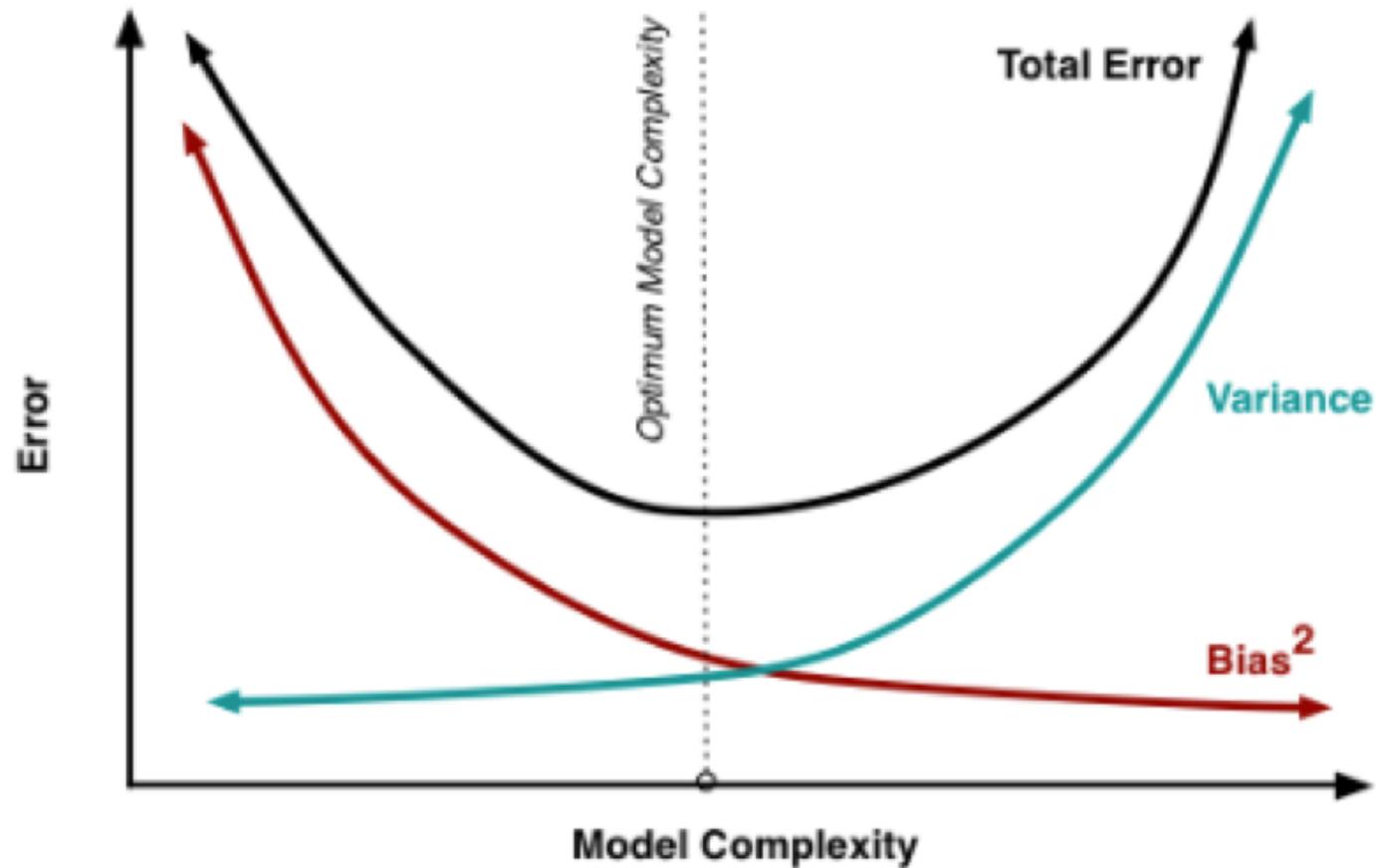
Complexidade Funcional versus Over-fitting



O dilema viés-variância (bias-variance)



O dilema viés-variância (bias-variance)



Treinamento com Validação Cruzada

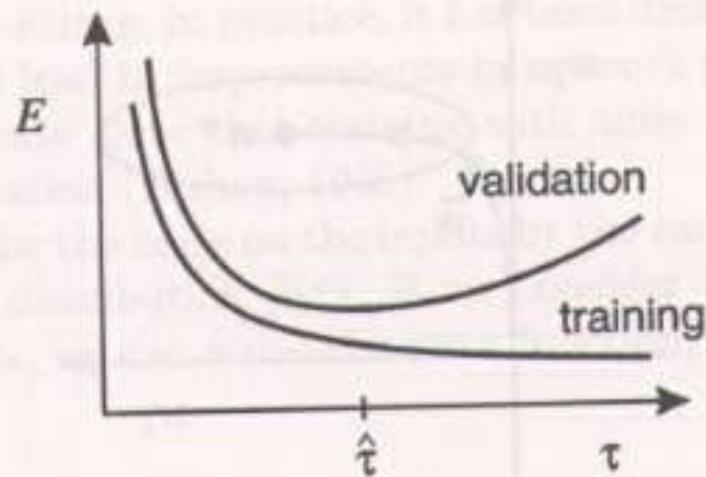
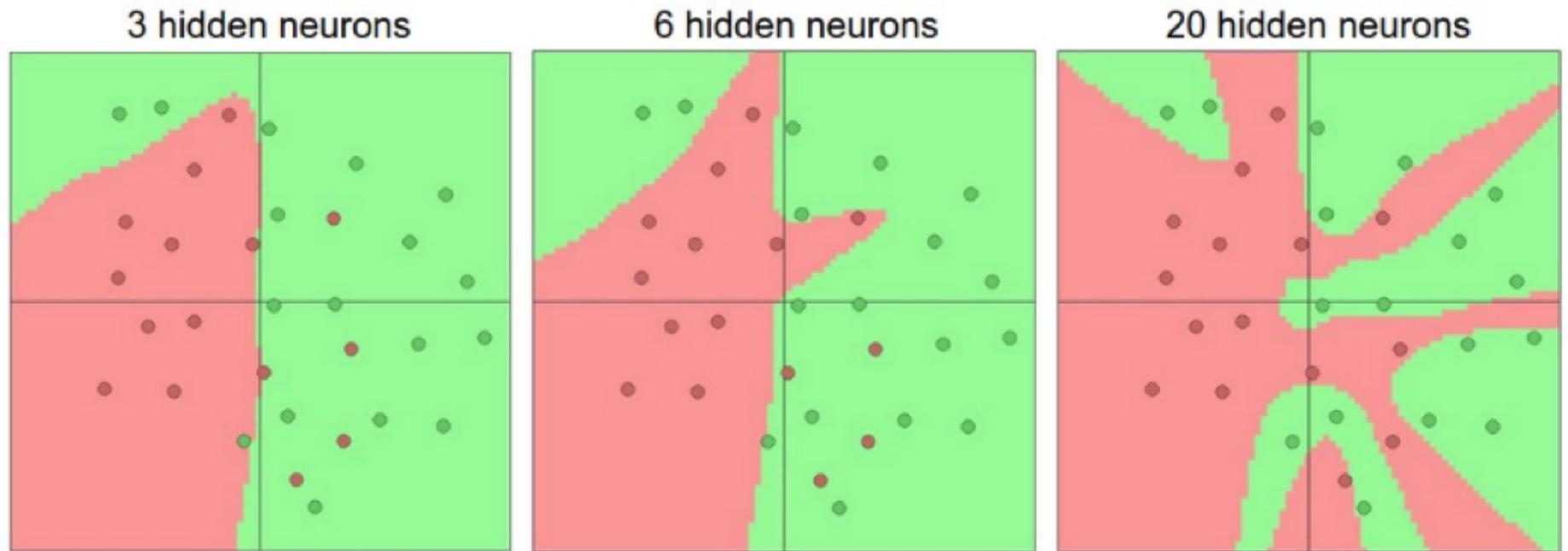


Figure 9.7. A schematic illustration of the behaviour of training and validation set errors during a typical training session, as a function of the iteration step τ . The goal of achieving the best generalization performance suggests that training should be stopped at the point $\hat{\tau}$ corresponding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

Ainda Sobre Overfitting



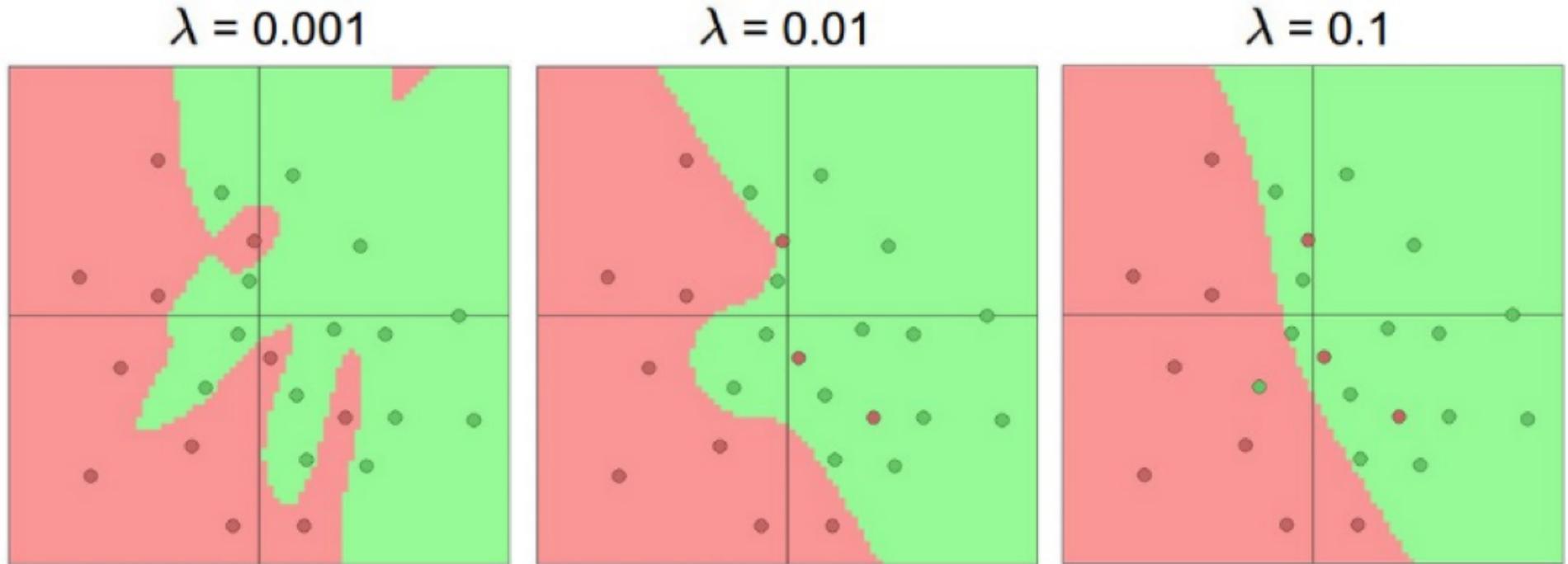
Larger Neural Networks can represent more complicated functions. The data are shown as circles colored by their class, and the decision regions by a trained neural network are shown underneath. You can play with these examples in this [ConvNetsJS demo](#).

Regularização (Regularization)

$$C = \frac{1}{2n} \sum_x \|y - a^L\|^2 + \frac{\lambda}{2n} \sum_w w^2.$$

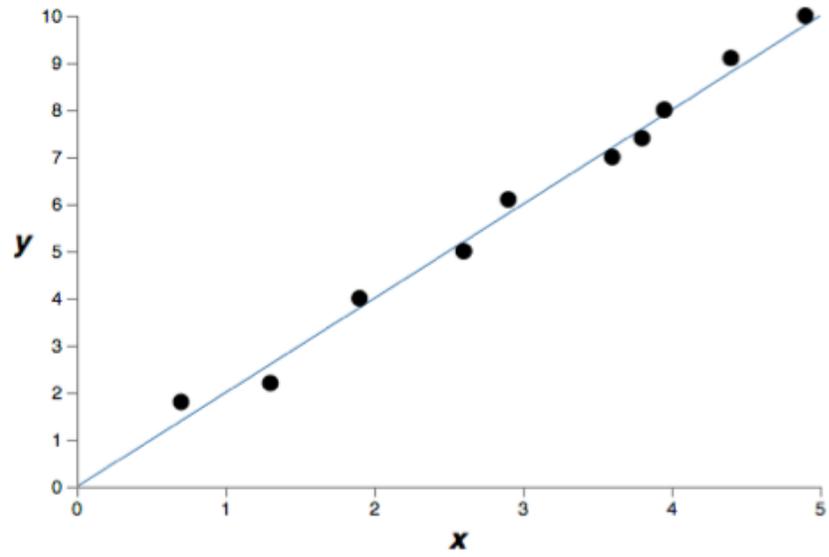
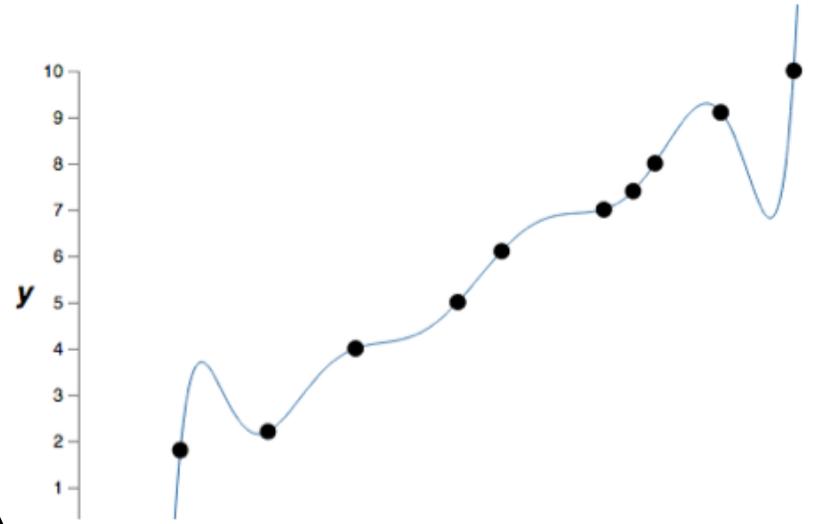
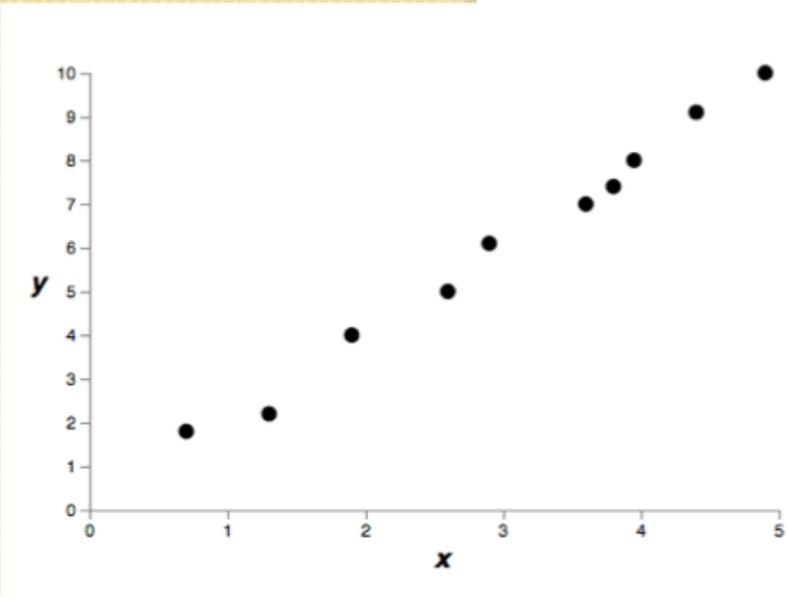
Compromisso entre redução do erro e pesos pequenos

Efeito do Parâmetro λ

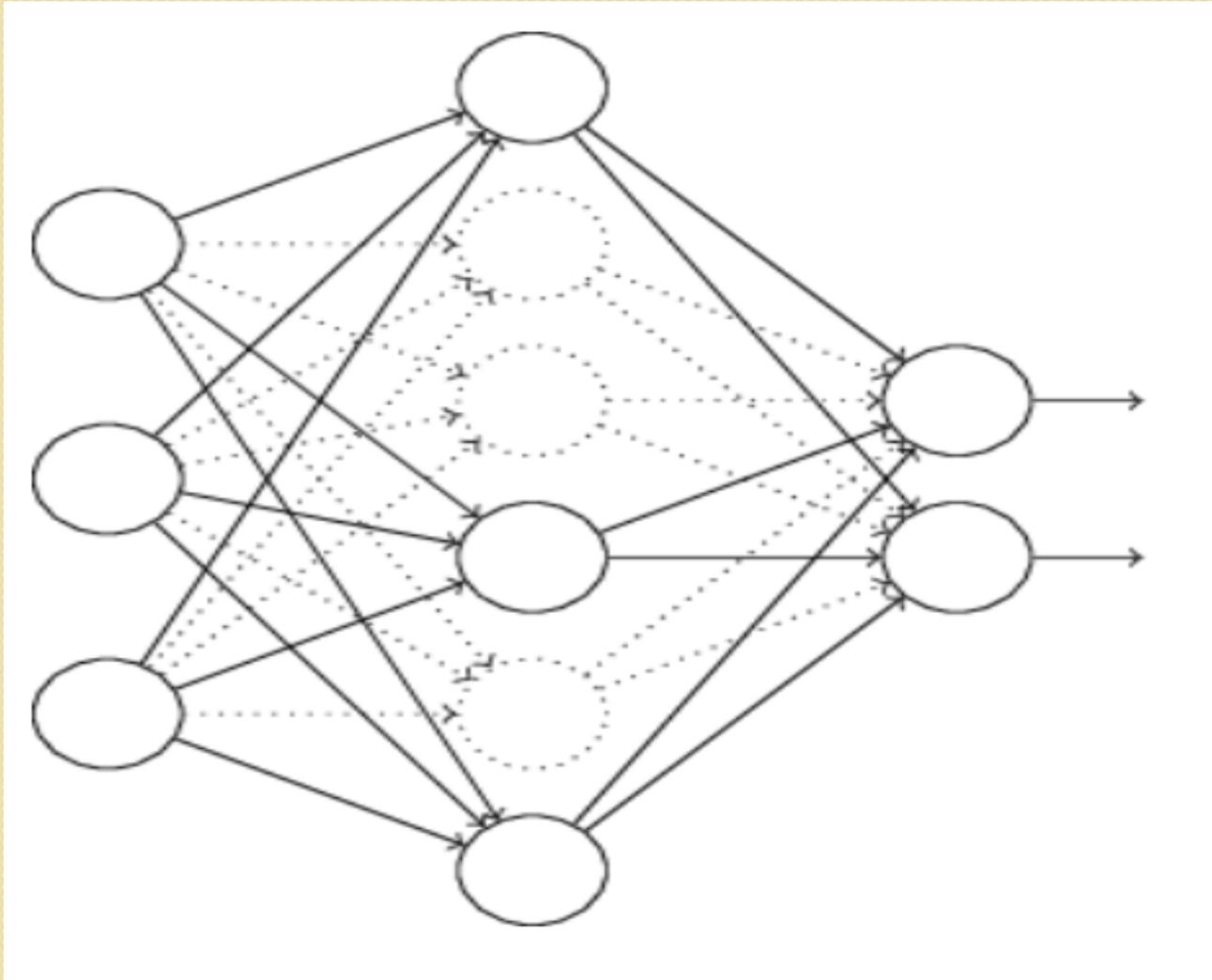


The effects of regularization strength: Each neural network above has 20 hidden neurons, but changing the regularization strength makes its final decision regions smoother with a higher regularization. You can play with these examples in this [ConvNetsJS demo](#).

Efeitos da Regularização



Regularization: Dropout



Demo...

<http://playground.tensorflow.org/>



Aplicações do MLP



Análise de mercado



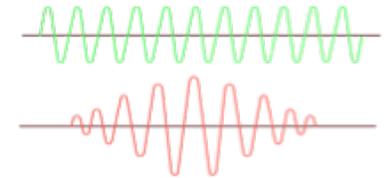
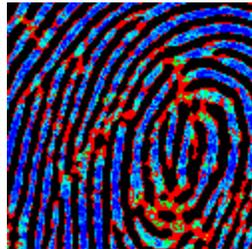
Proc. voz



Data mining



Análise de crédito

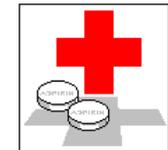


Proc. sinais



Previsão séries

Luciana de Galen Maciel



Diagnose médica



Det. fraudes



Rec. odores



Interfaces

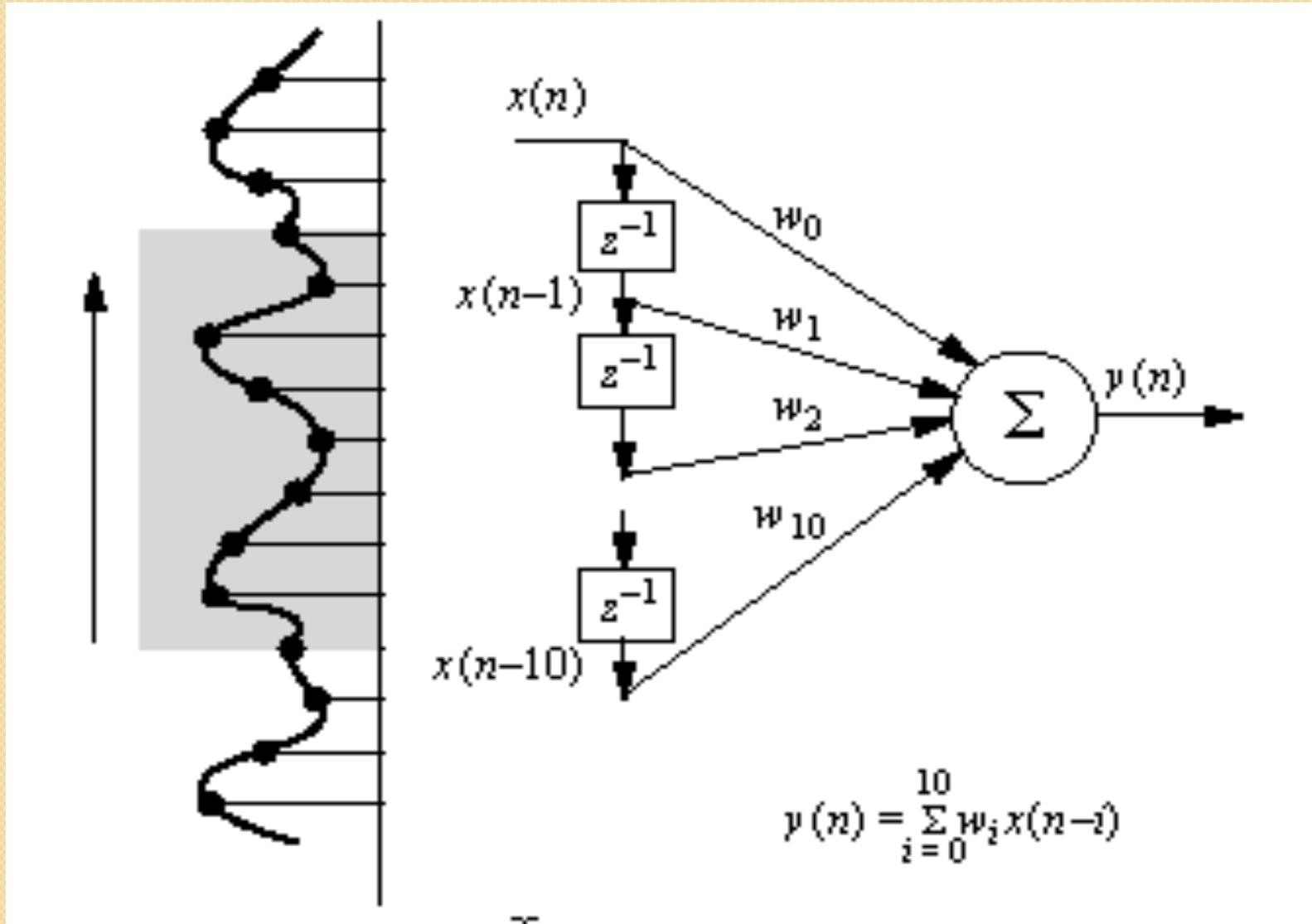
Exemplo: Previsão (Forecasting)

- Dado um conjunto de n valores de uma variável $(y(t_1), y(t_2), \dots, y(t_n))$ em uma sequência de tempo t_1, t_2, \dots, t_n ,
- Prever o valor $y(t_{n+1})$ num futuro t_{n+1}

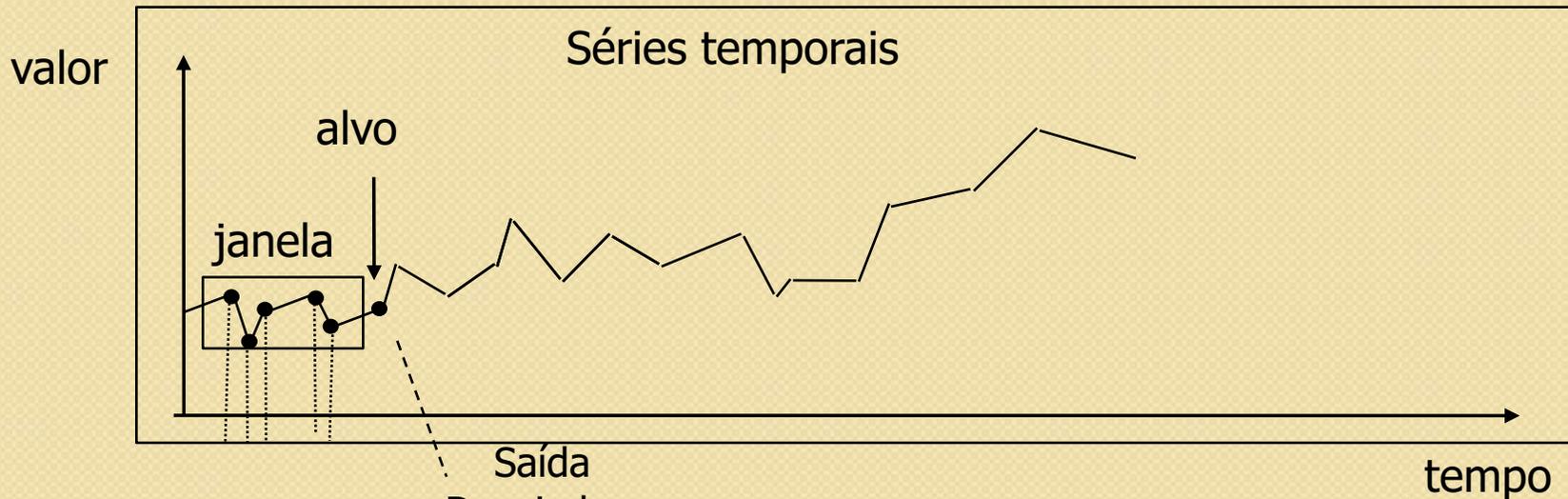
Previsão – Questões Relevantes

- Definição da janela de entrada
- Definição do horizonte de previsão
- Definição de outras variáveis explicativas

Previsão com uma Rede MLP



Previsão de Séries Temporais



Entradas da rede =
n valores passados

Ex: 5 valores passados

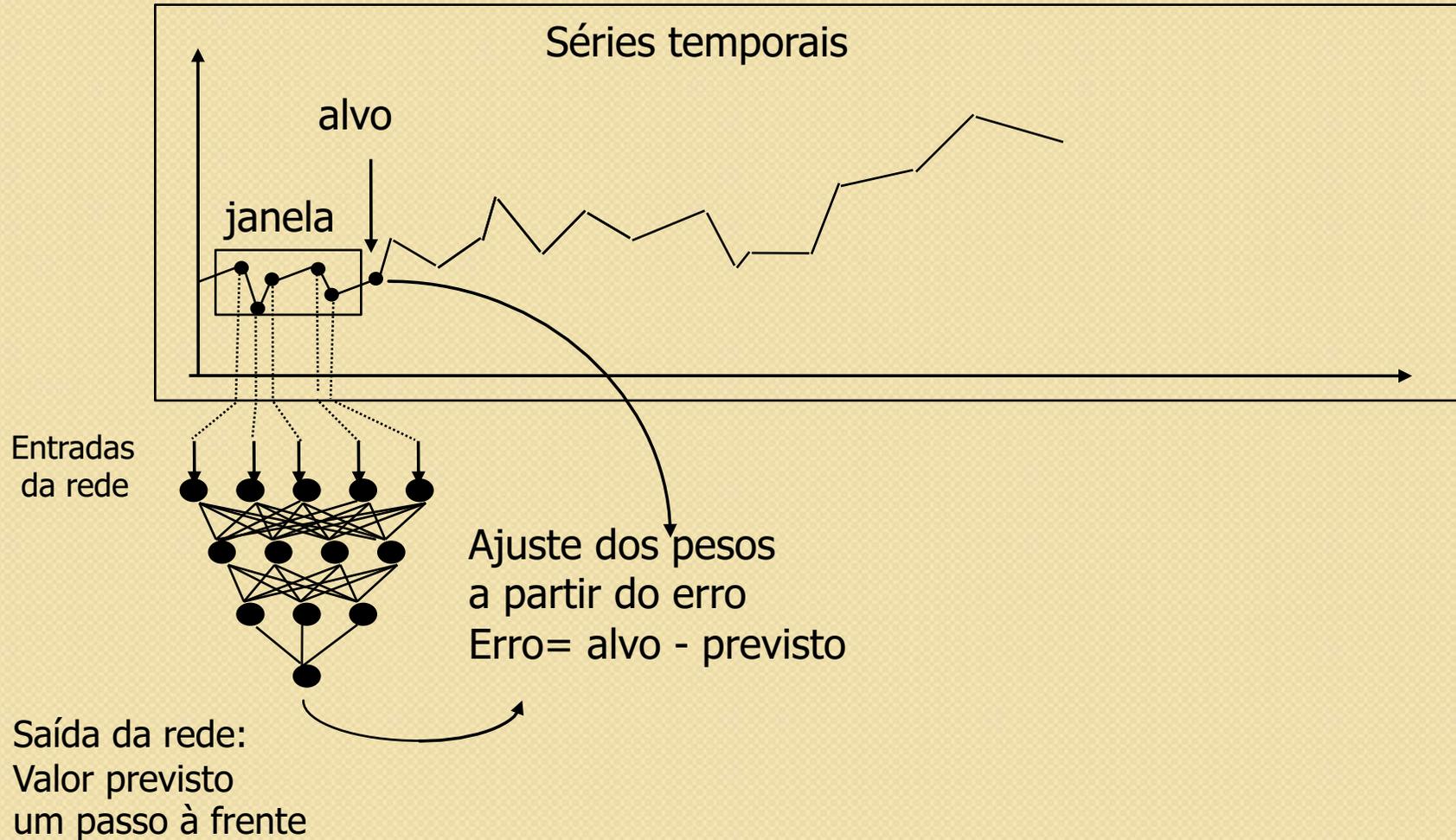
Desejada =
valor da série
k passos à frente

Ex: valor um passo à frente

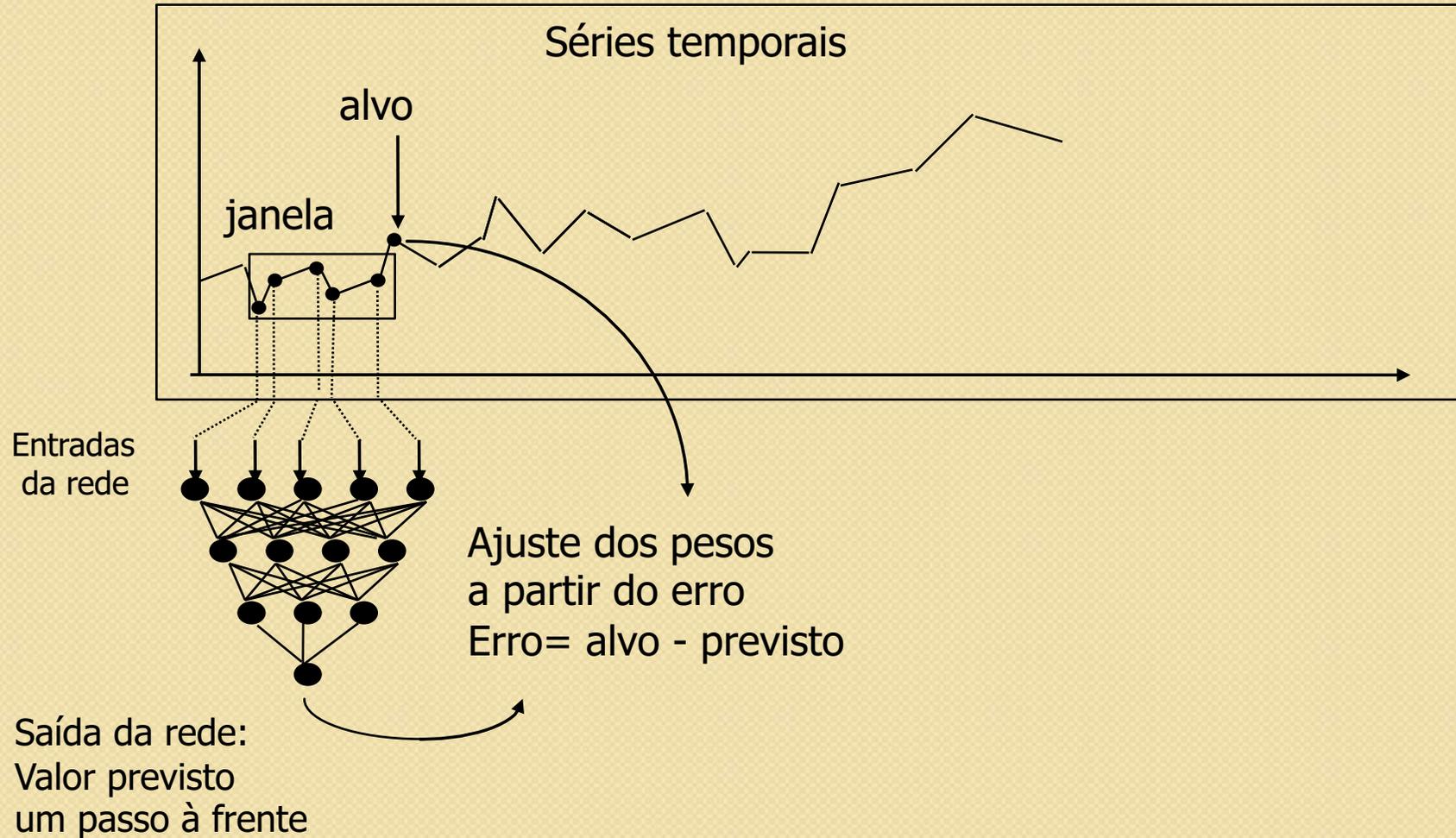
Definição da janela de entrada

Definição da janela de saída

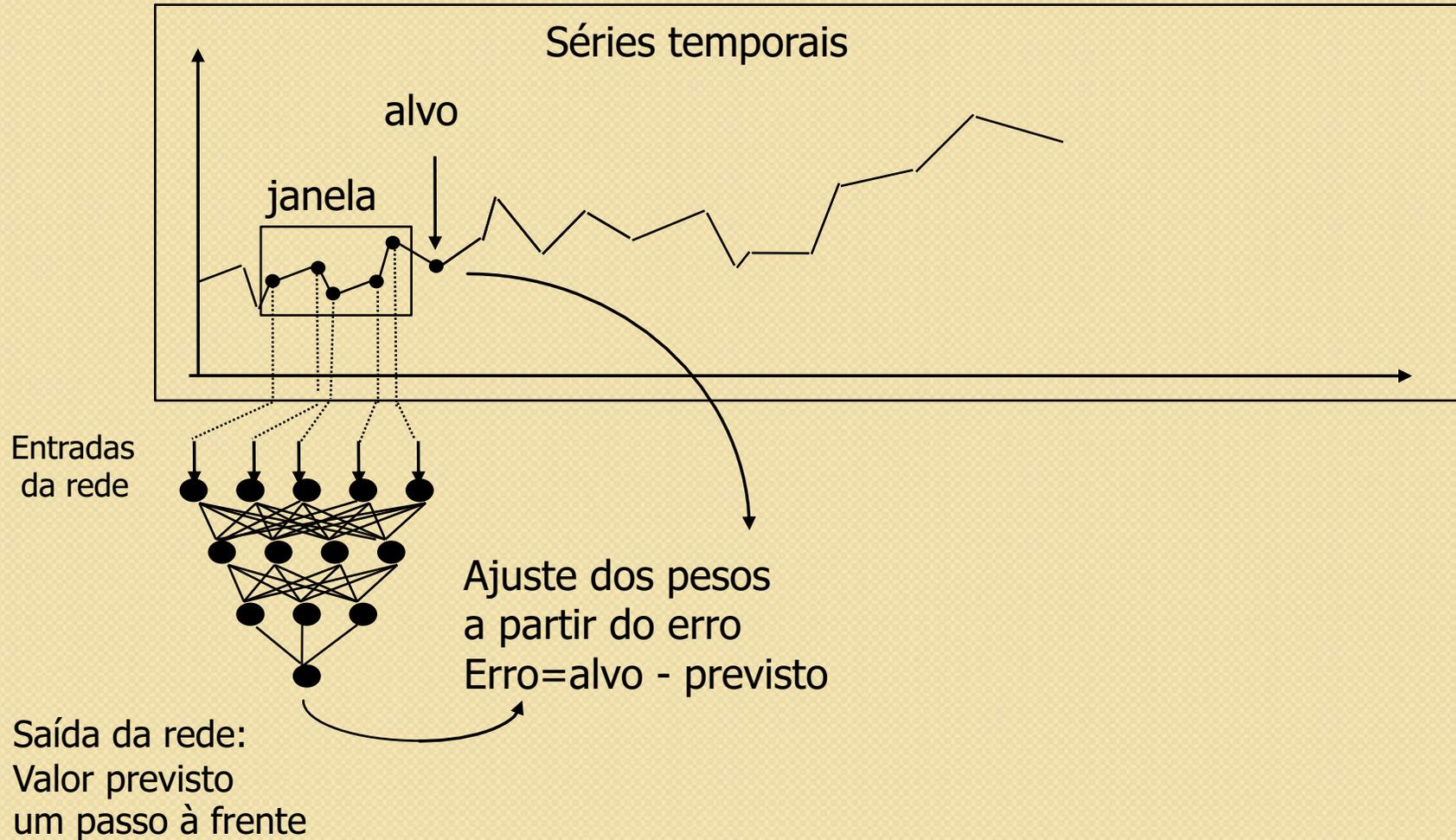
Exemplo: previsão utilizando apenas a série histórica como entrada



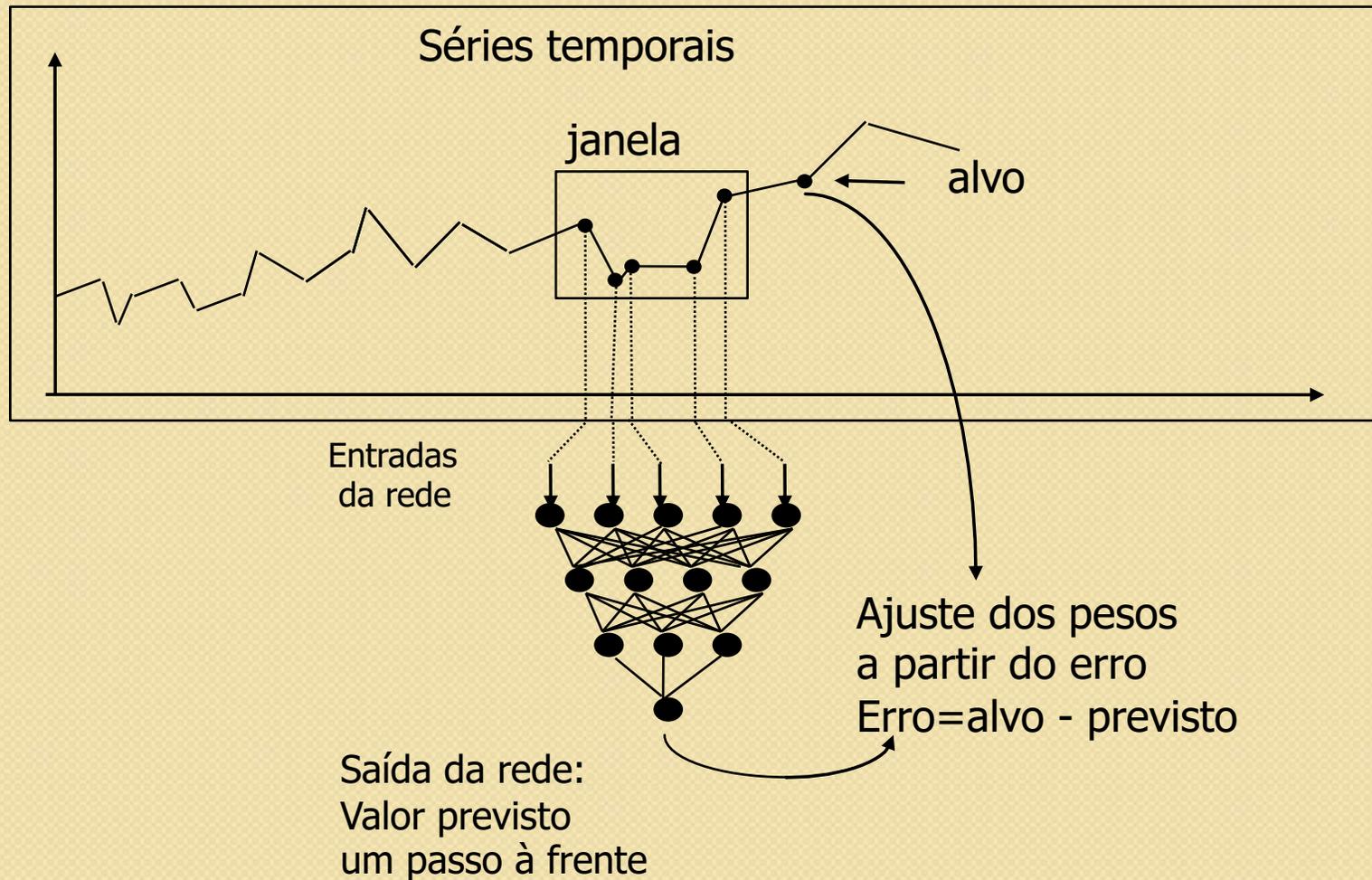
Exemplo: previsão utilizando apenas a série histórica como entrada



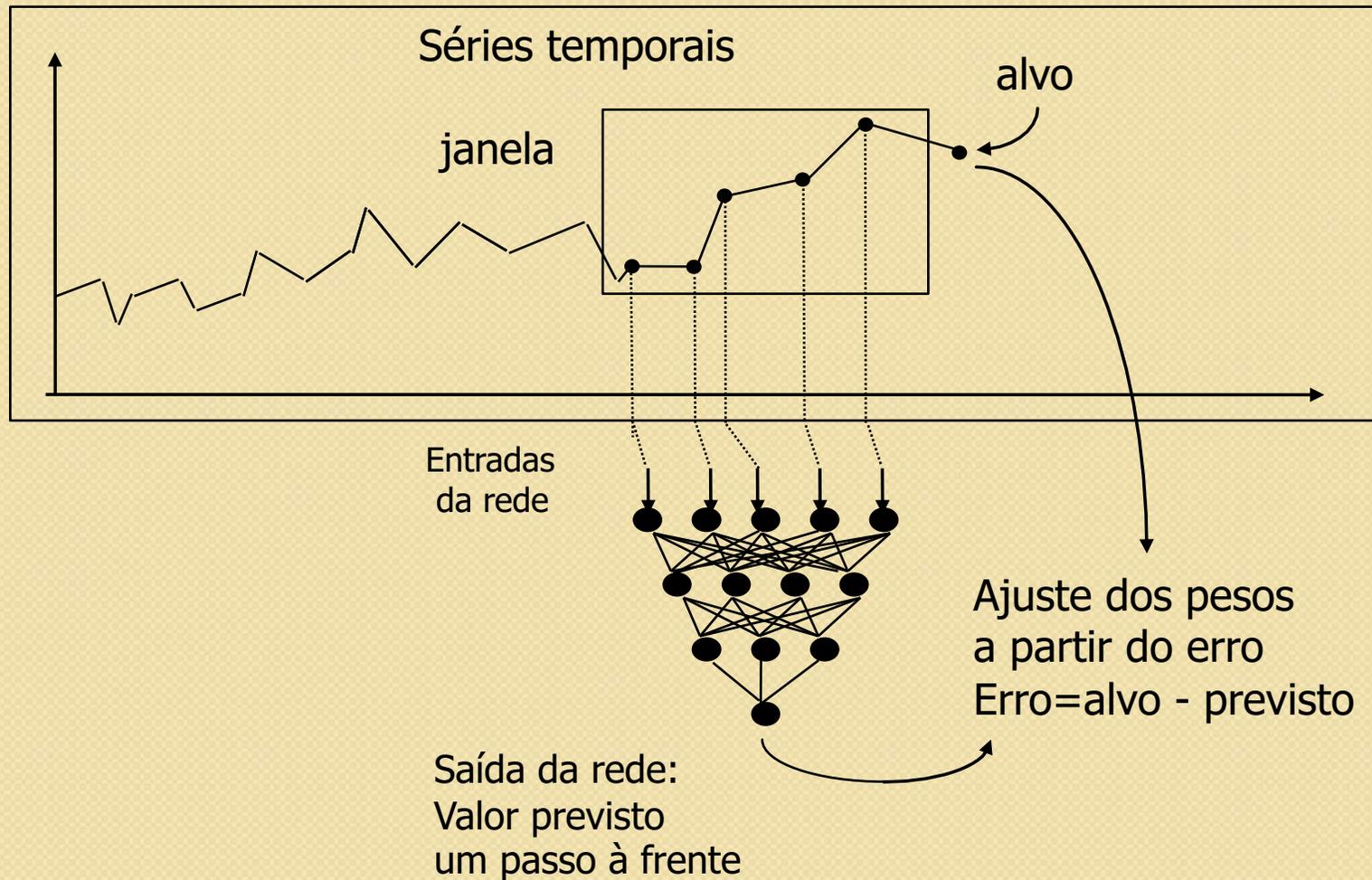
Exemplo: previsão utilizando apenas a série histórica como entrada



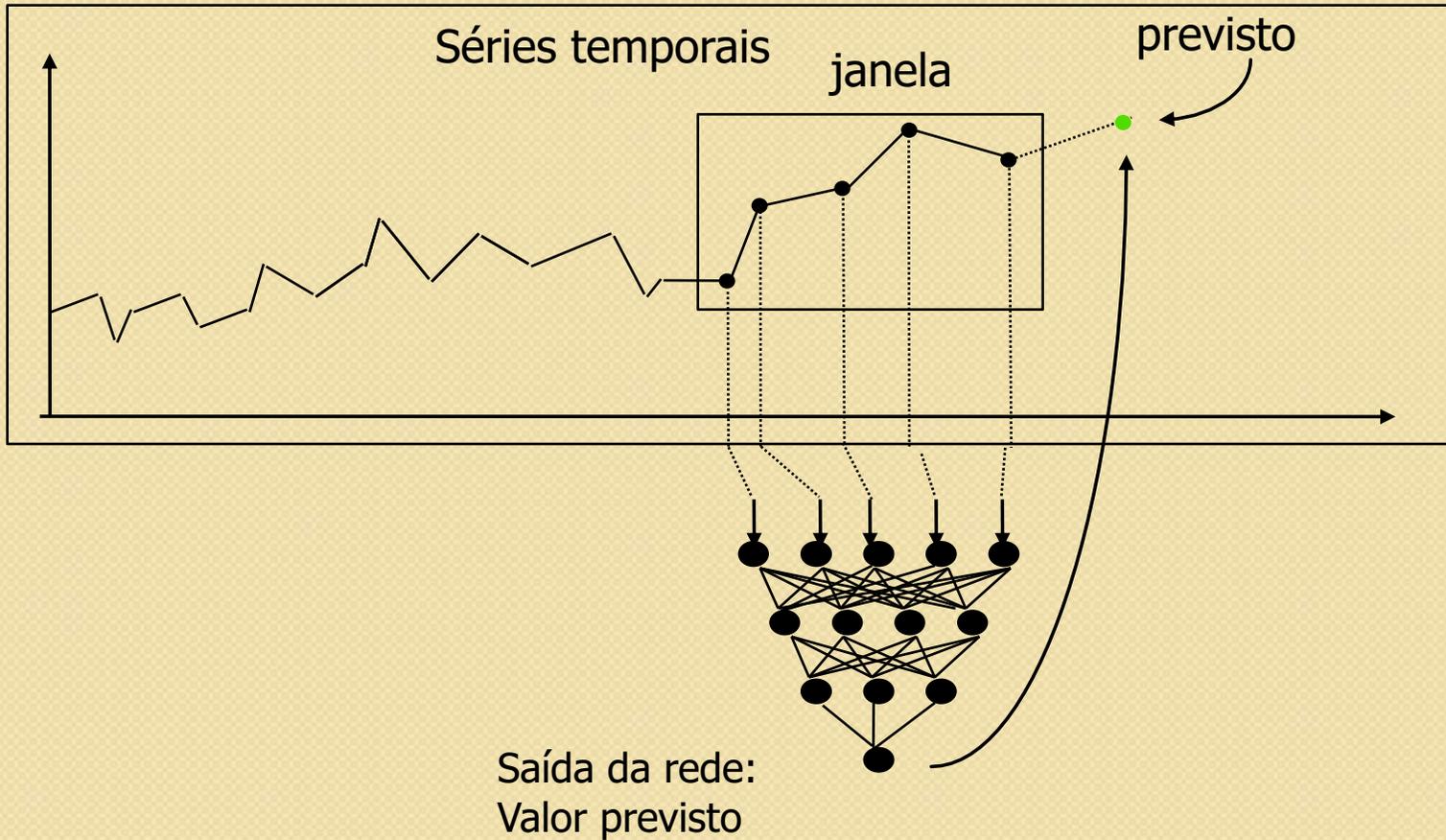
Exemplo: previsão utilizando apenas a série histórica como entrada



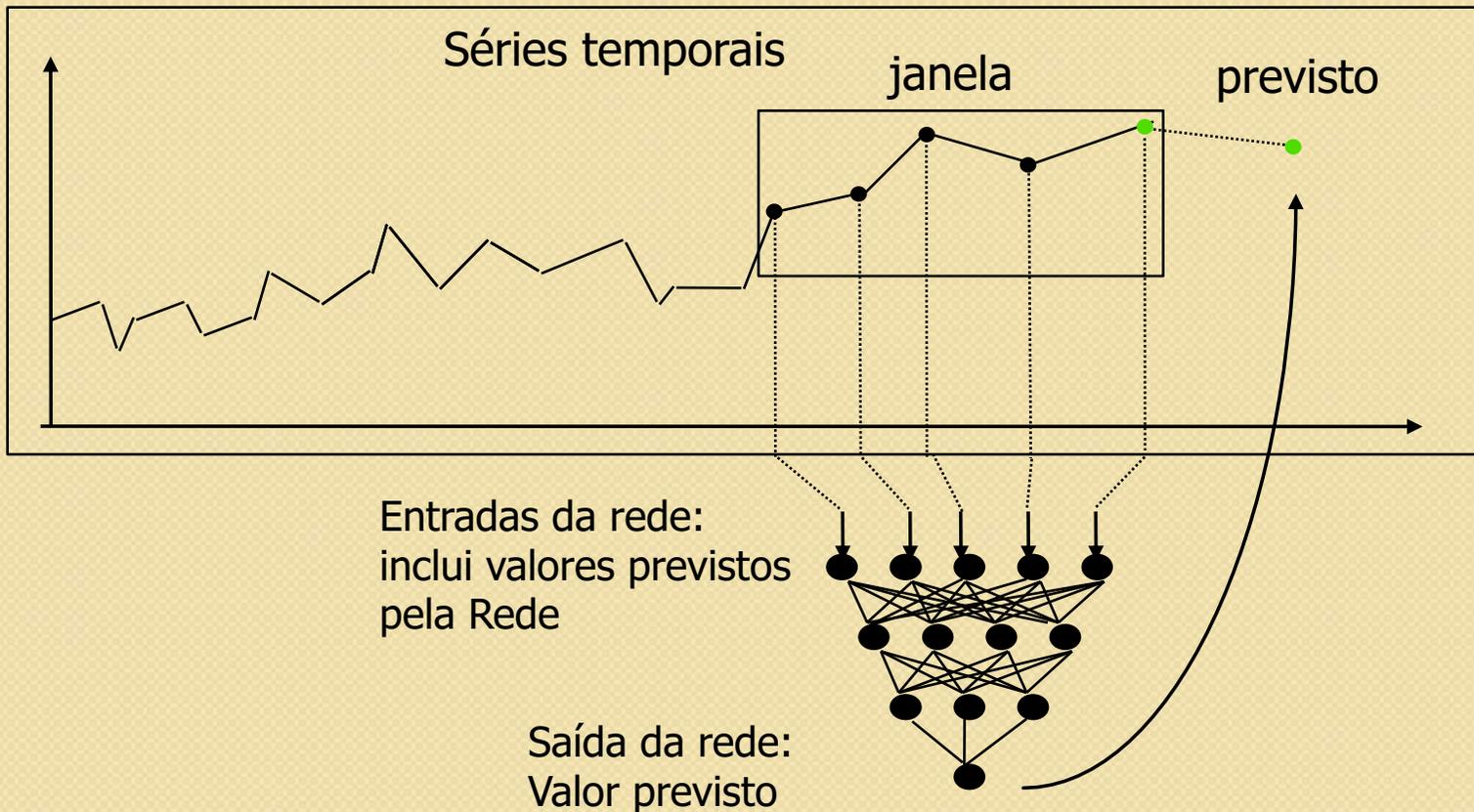
Exemplo: previsão utilizando apenas a série histórica como entrada



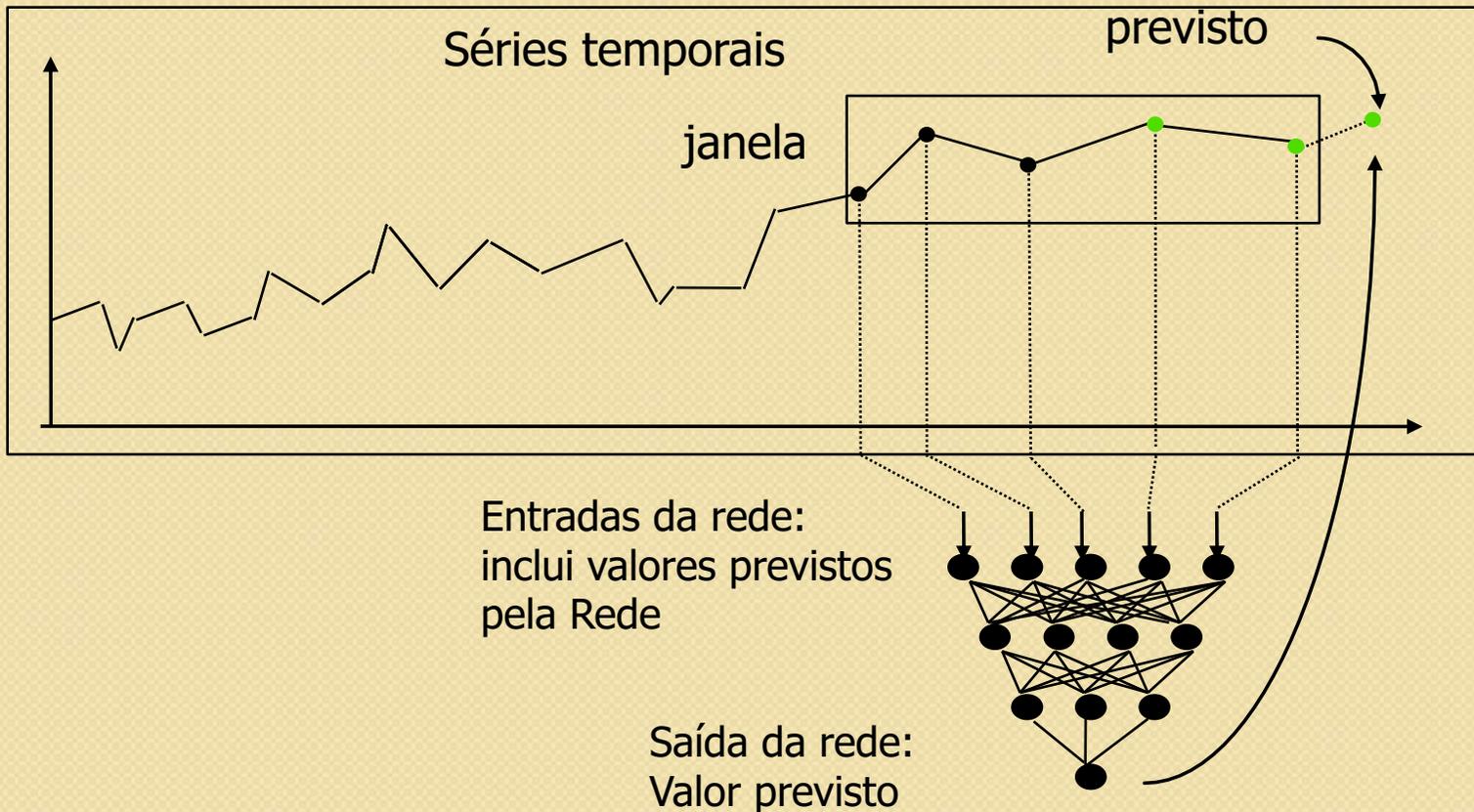
Exemplo: previsão utilizando apenas a série histórica como entrada



Exemplo: previsão utilizando apenas a série histórica como entrada

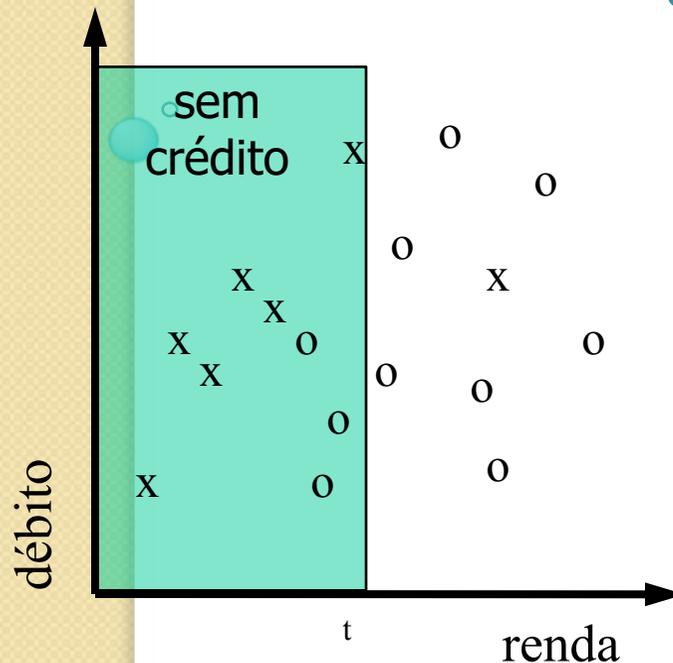


Exemplo: previsão utilizando apenas a série histórica como entrada



Complexidade Funcional (I)

Análise de crédito

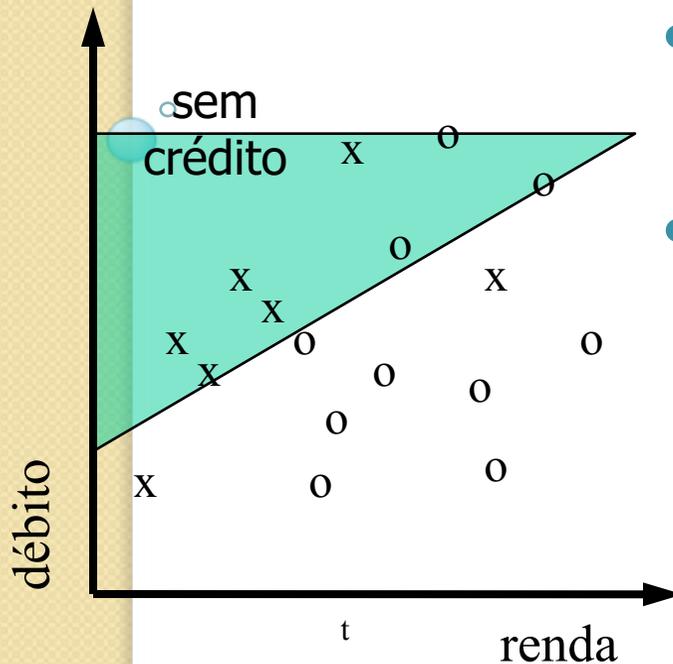


x: exemplo recusado
o: exemplo aceito

- 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

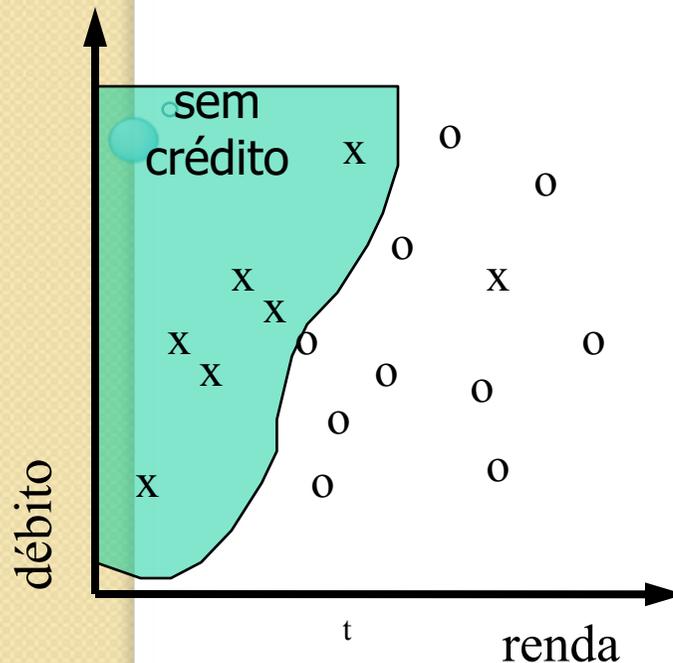


x: exemplo recusado
o: exemplo aceito

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

Complexidade Funcional (III)

Análise de crédito

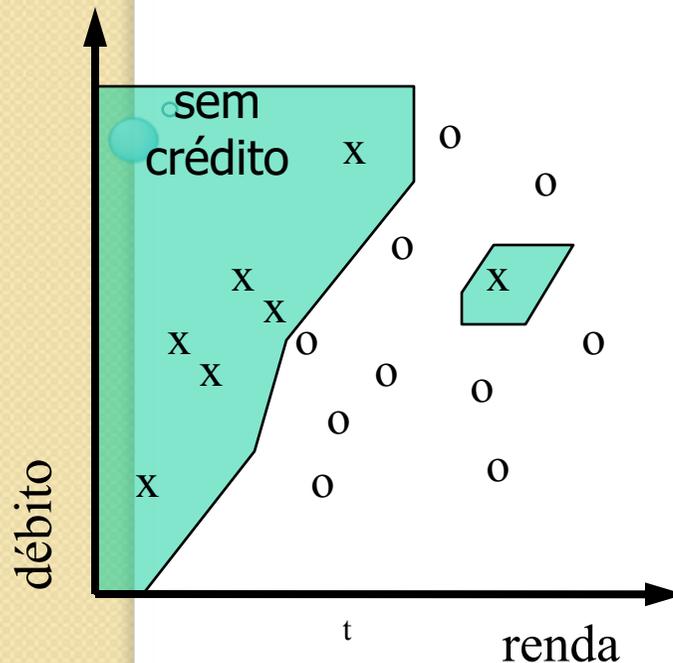


x: exemplo recusado
o: exemplo aceito

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

Complexidade Funcional (IV)

Análise de crédito

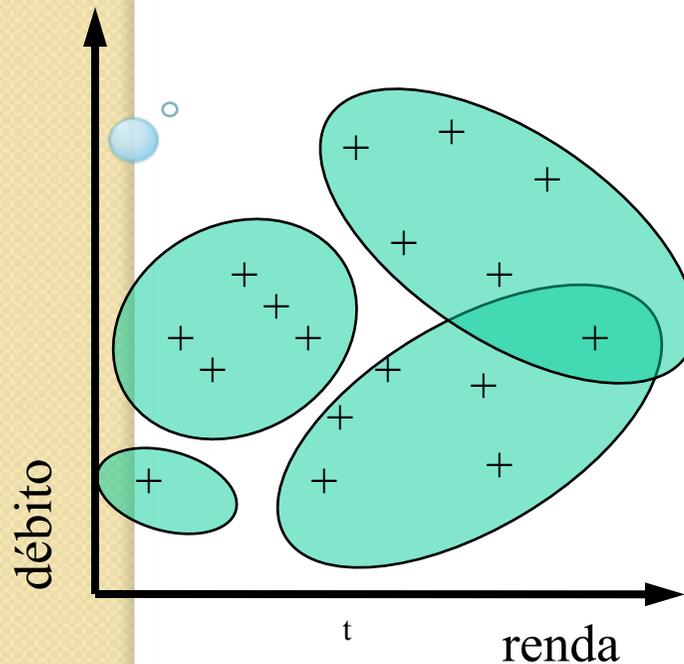


x: exemplo recusado
o: exemplo aceito

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

Complexidade Funcional (V)

Análise de crédito



+: exemplo

- Agrupamento
- Exemplo:
 - vector quantization;
 - ART (Adaptive Resonance Theory)