SICO7O
SISTEMAS INTELIGENTES 2

Aula 05 - *Backpropagation*

Prof. Rafael G. **Mantovani**

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Apucarana - PR, Brasil

Universidade Tecnológica Federal do Paraná (UTFPR)
Engenharia de Computação

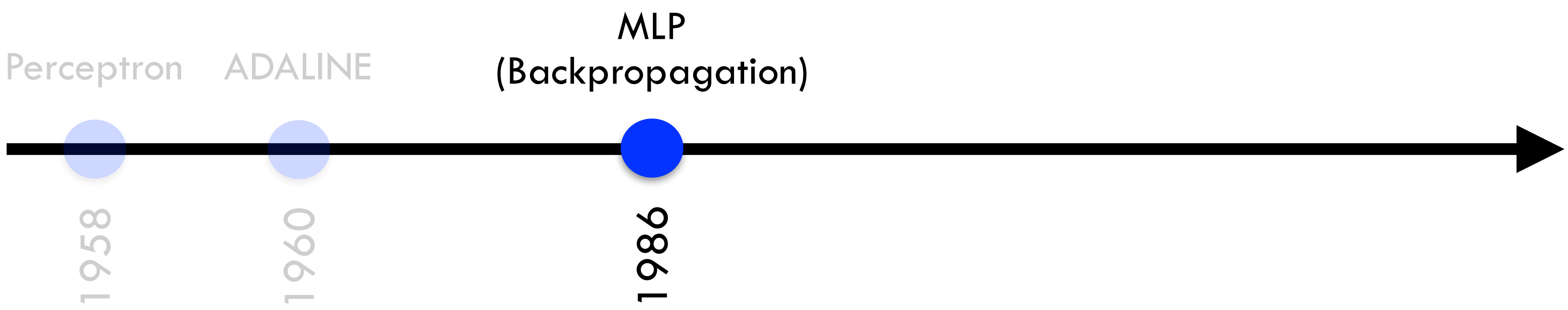
Roteiro

1. Introdução
2. Backpropagation
3. Exemplo
4. Exercício
5. Referências

Roteiro

1. Introdução
2. Backpropagation
3. Exemplo
4. Exercício
5. Referências

Introdução

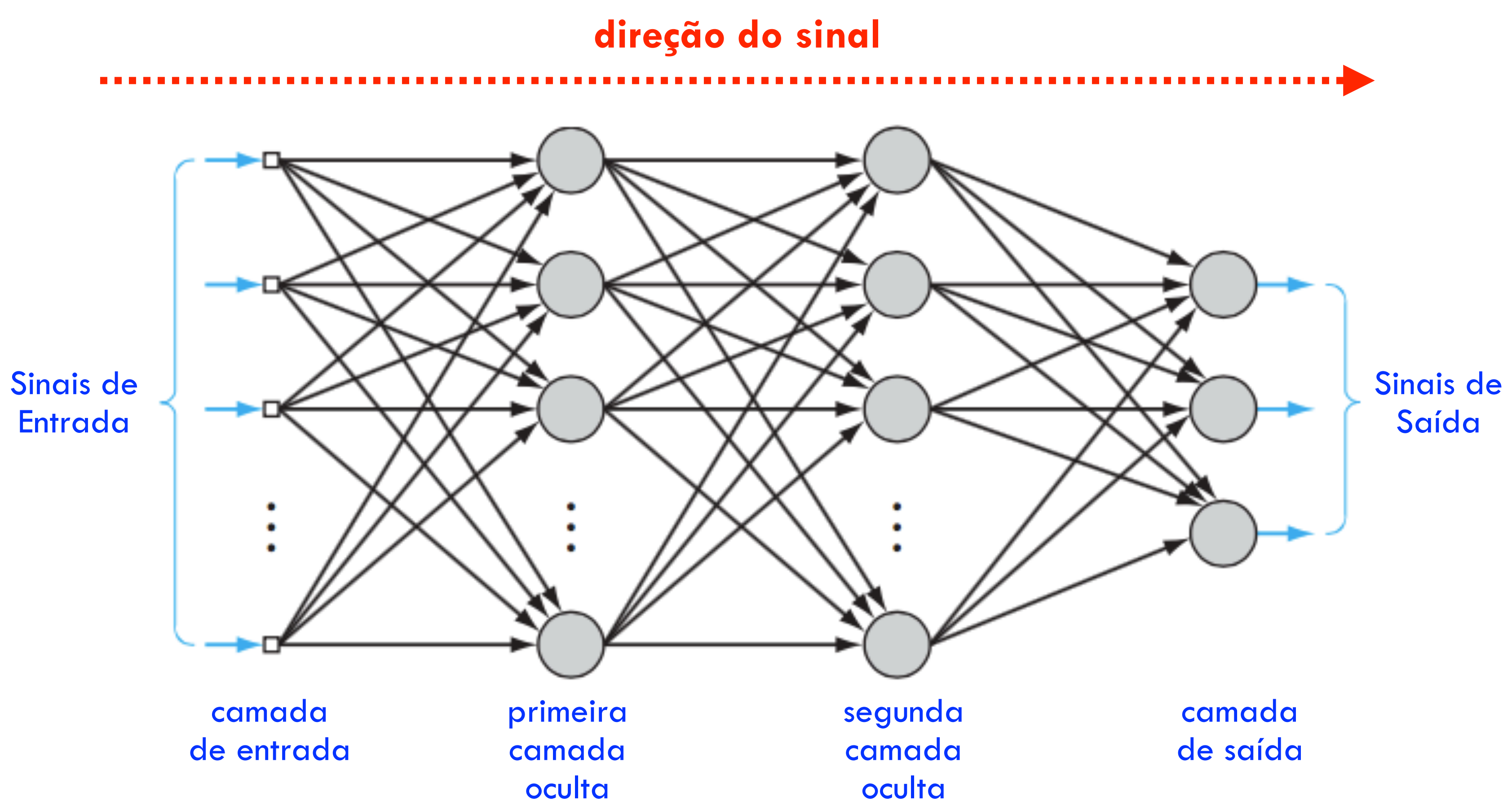


Introdução

- **Multilayer Perceptron:**

  - neurônios possuem uma função de ativação **não-linear e diferenciável**
  - contém uma ou mais camadas escondidas
  - a rede possui alto grau de conectividade

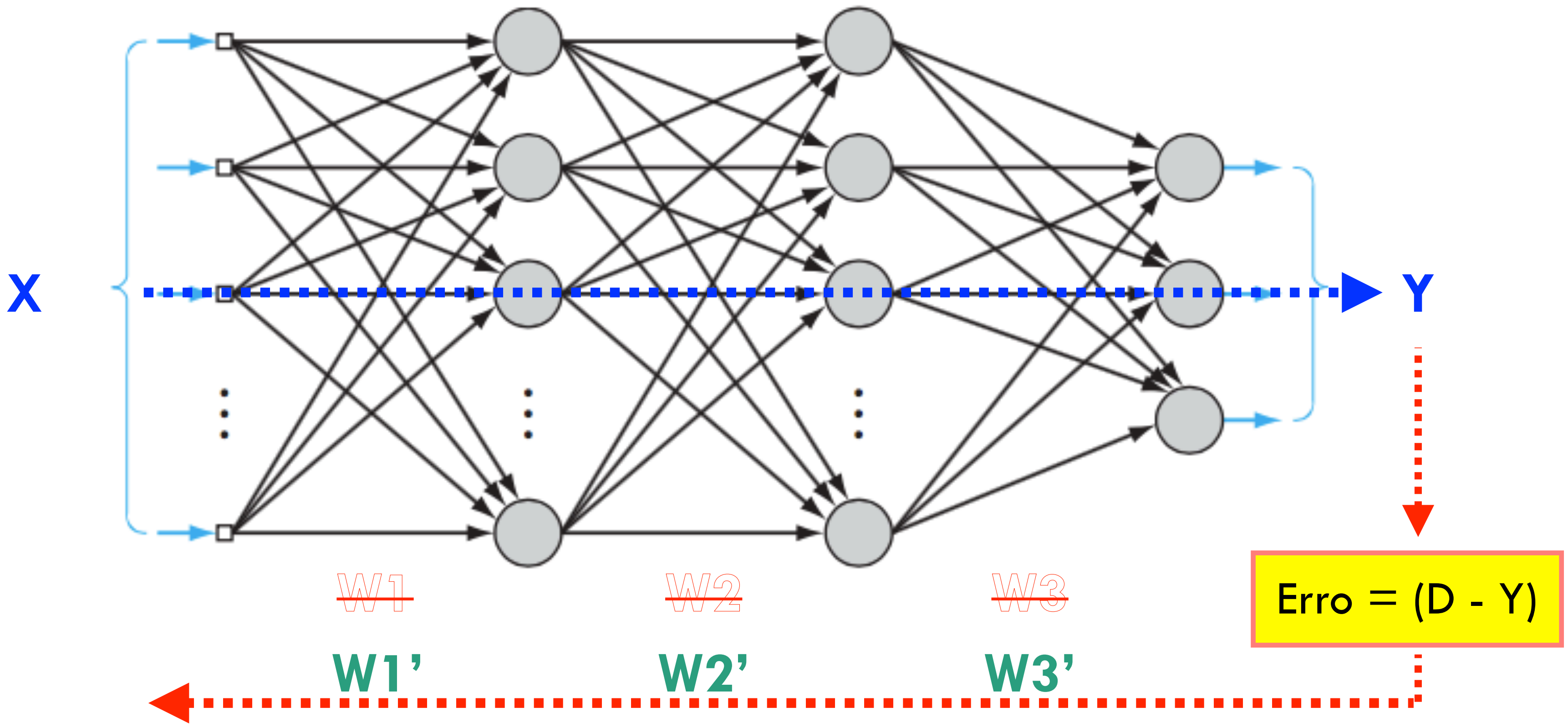
Introdução



direção do sinal
Sinais de Entrada
Sinais de Saída
camada de entrada
primeira camada oculta
segunda camada oculta
camada de saída

Introdução

- ***Backpropagation***



Introdução

- Se houverem **N** exemplos de treinamento, o erro médio sobre todos os exemplos (risco empírico) é dado por:

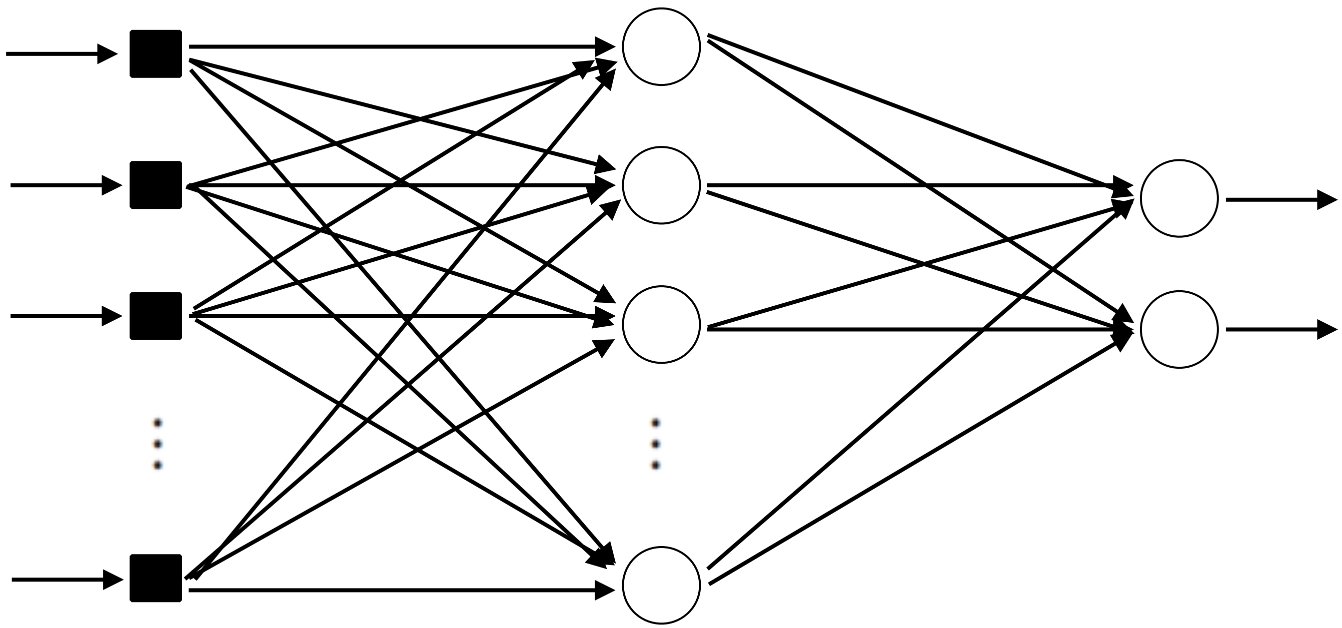
$$\varepsilon_{avg}(N) = \frac{1}{N}\sum_{n=1}^{N}\varepsilon(n) = \frac{1}{2N}\sum_{n=1}^{N}\sum_{j\in C} e_j^2(n)$$

Erro quadrático médio da época considera todos os neurônios da camada de saída **C** e todos os exemplos do conjunto de treinamento (**N**)

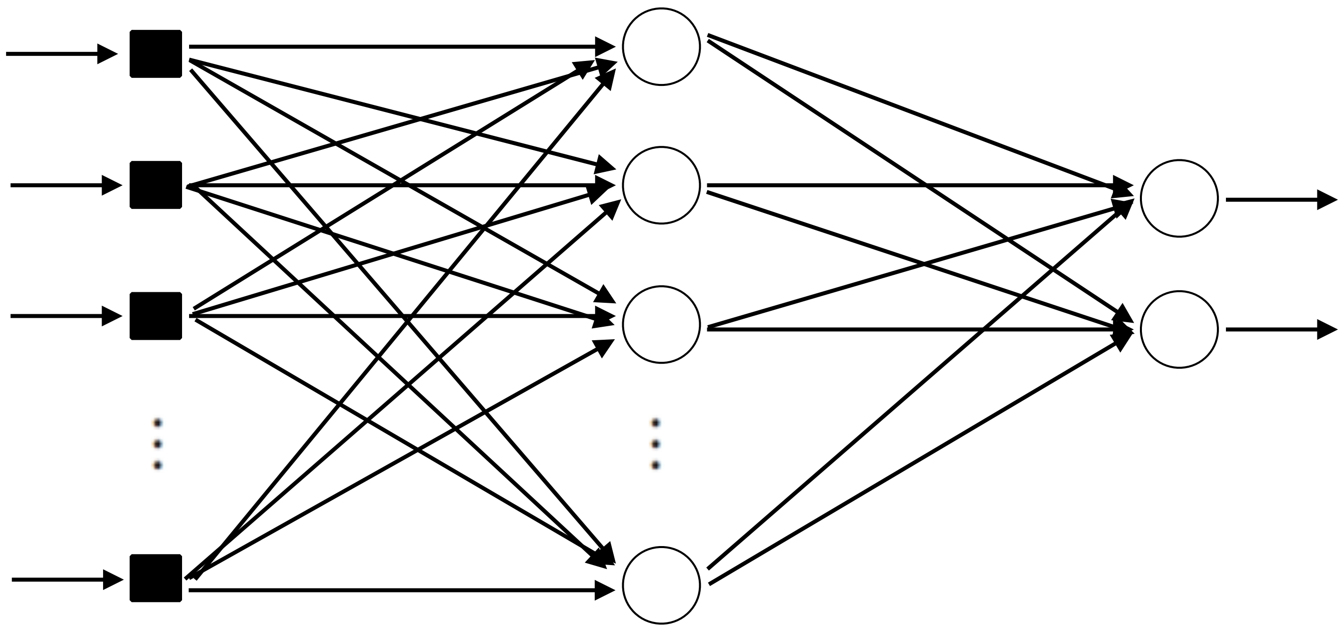
Roteiro

1. Introdução
2. **Backpropagation**
3. Exemplo
4. Exercício
5. Referências

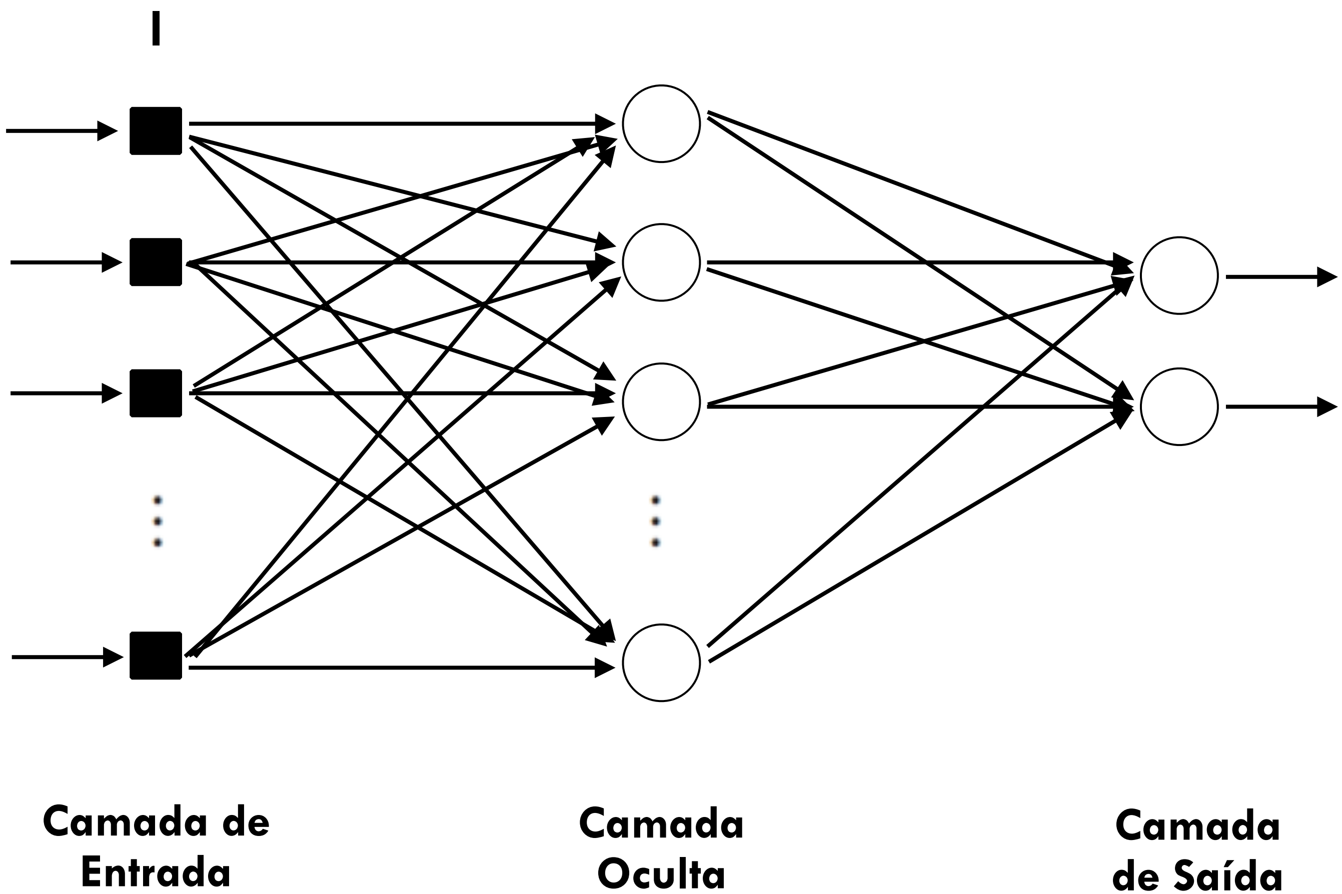
Backpropagation



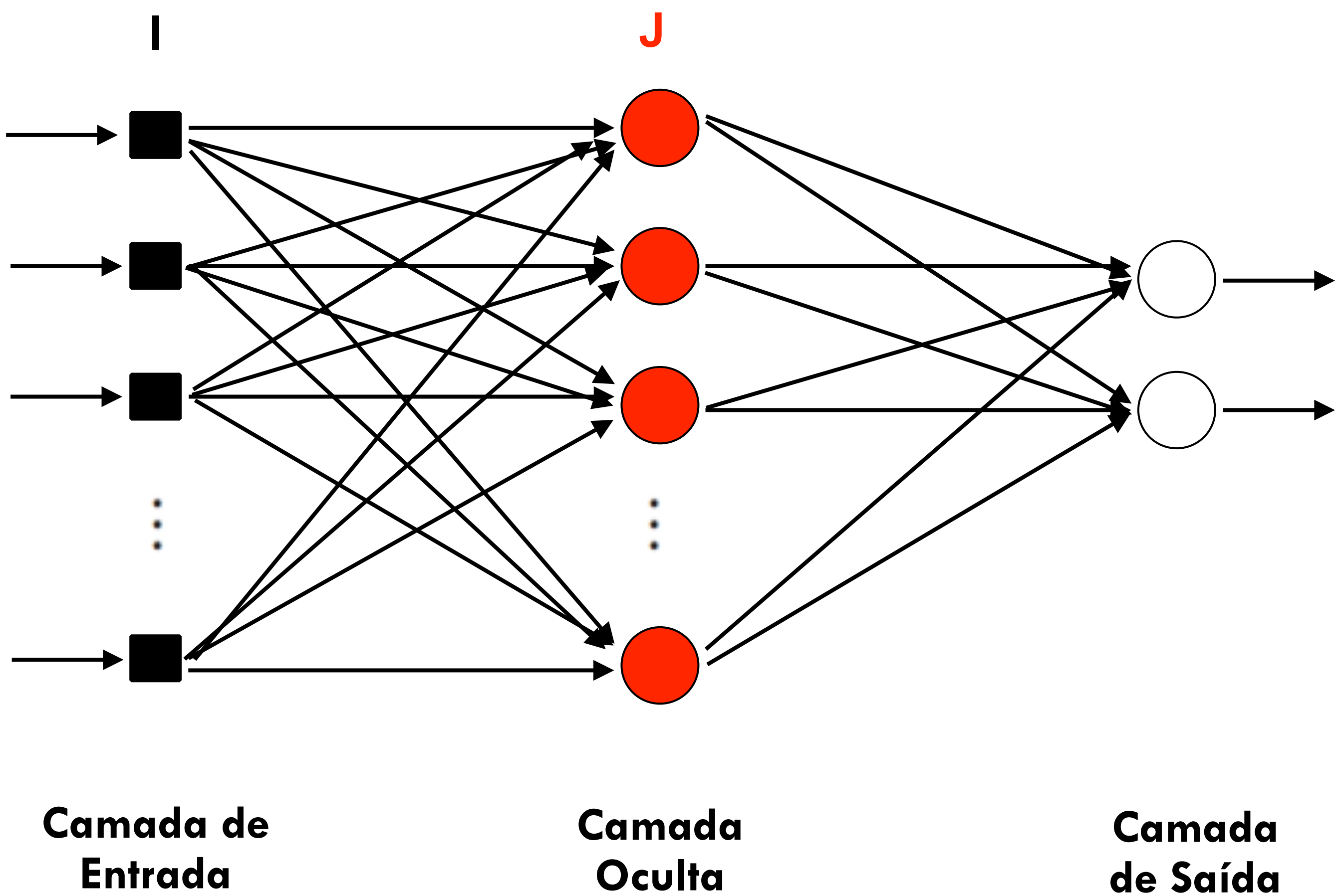
Backpropagation



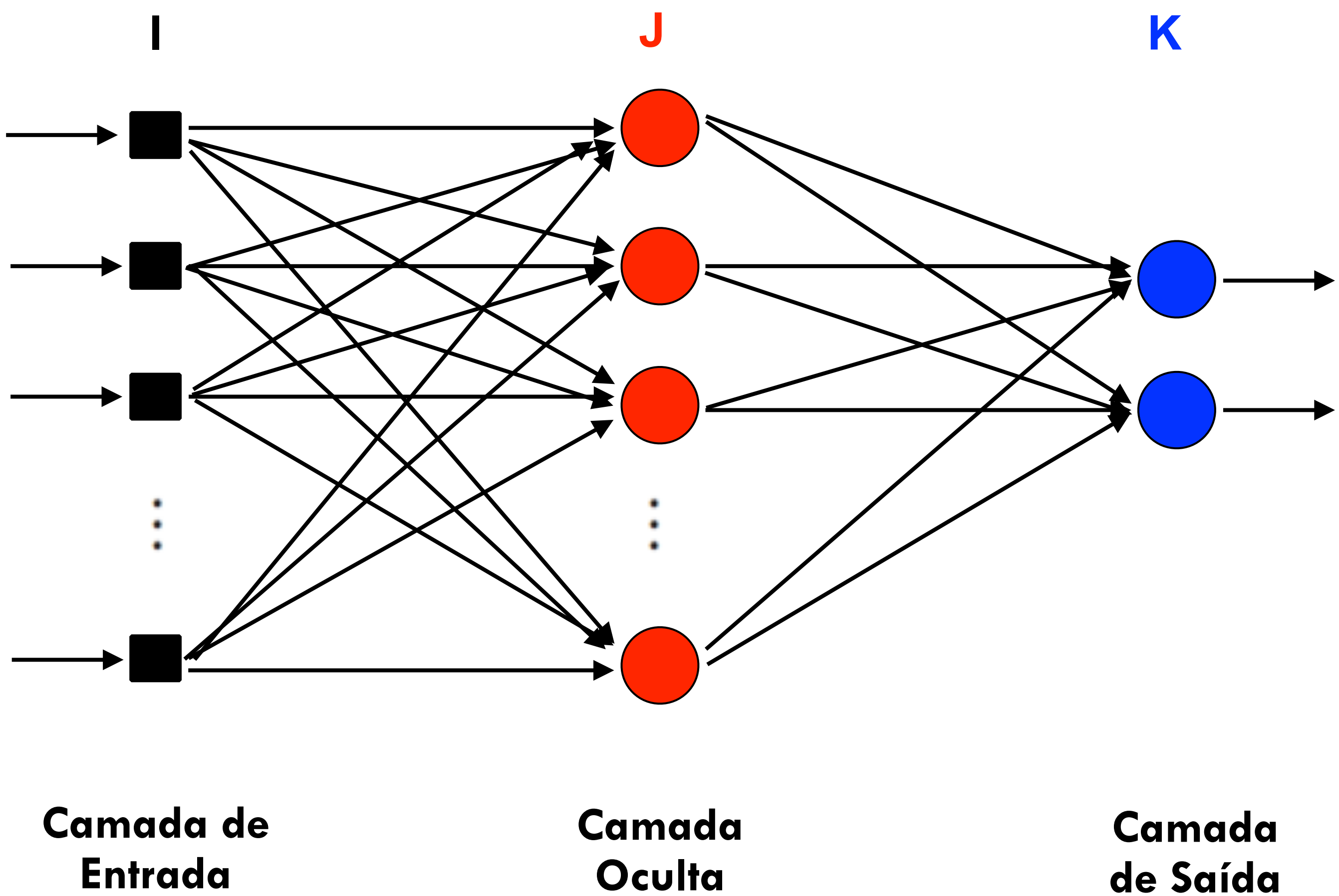
Backpropagation



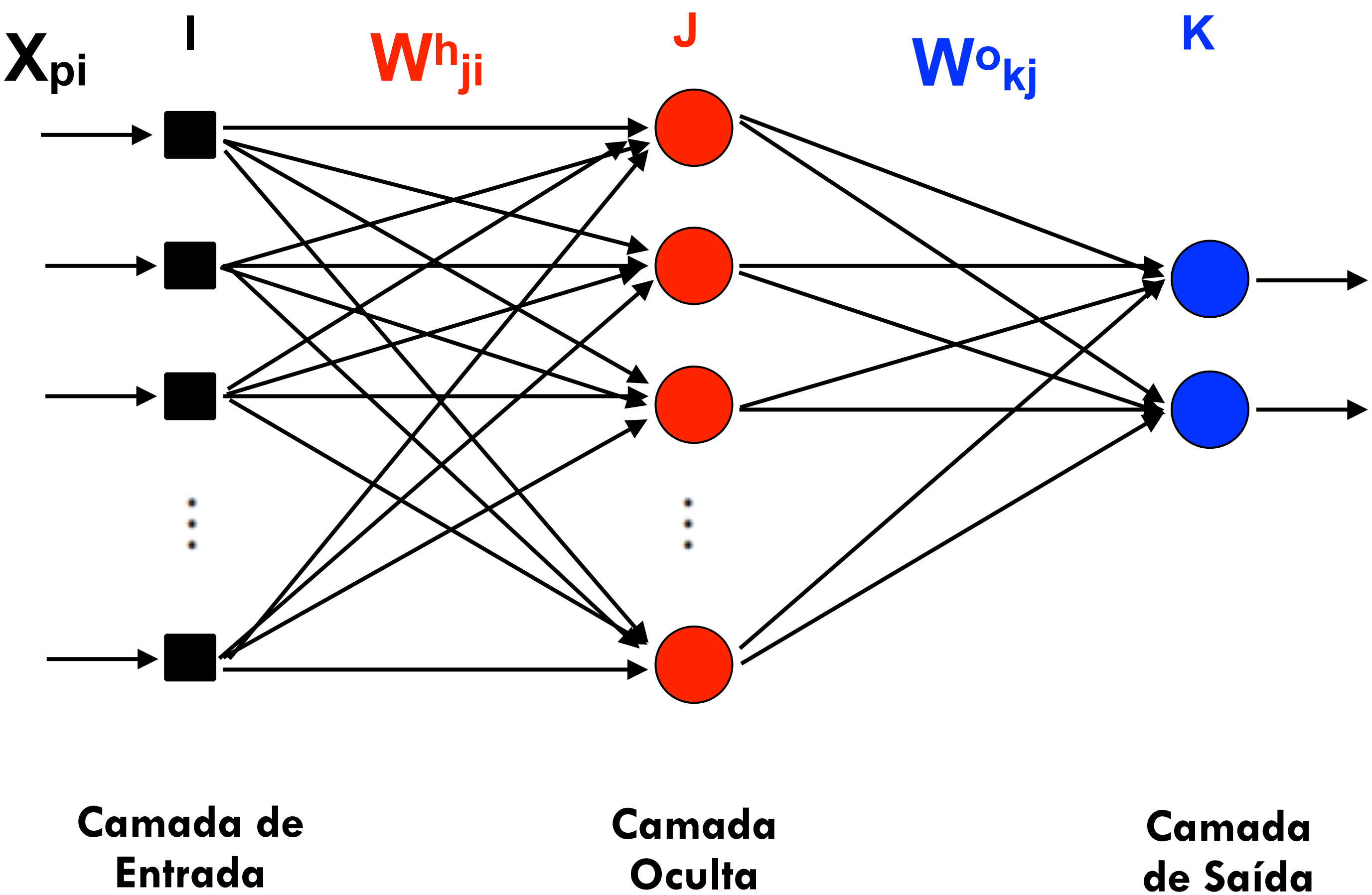
Backpropagation



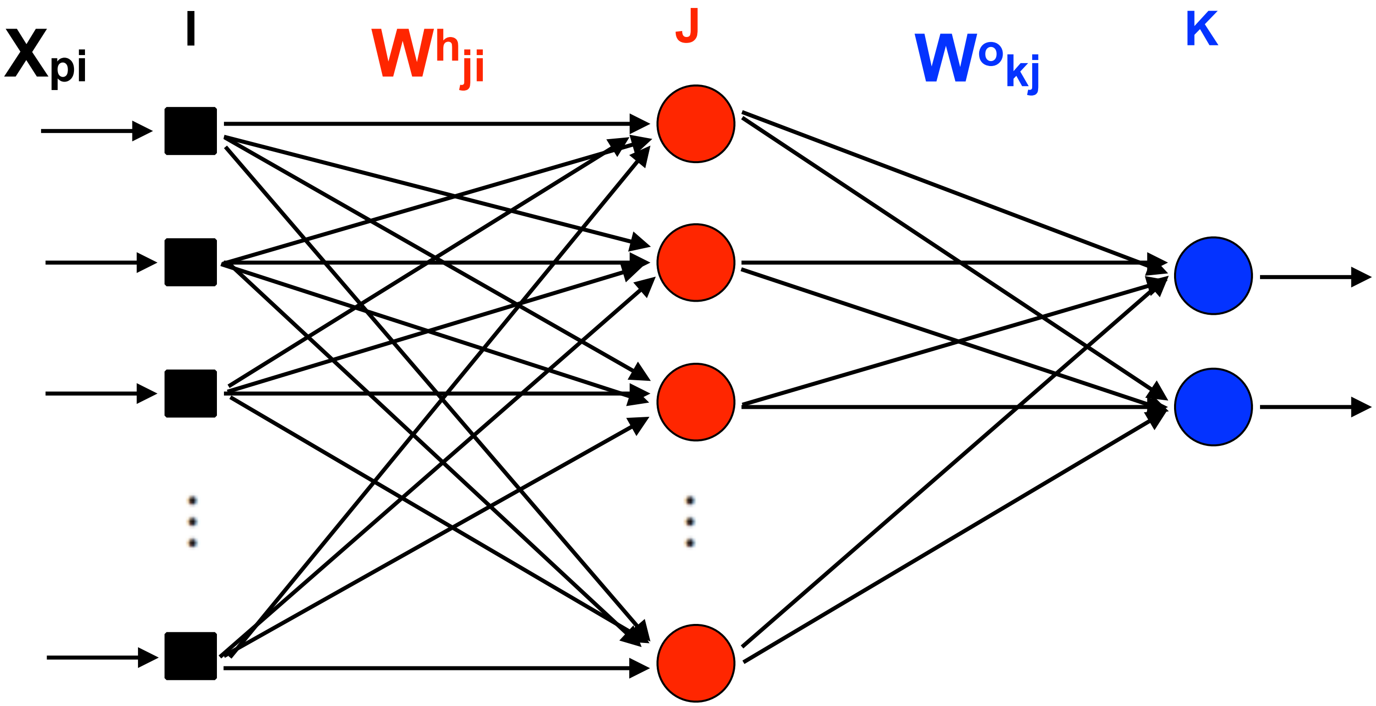
Backpropagation



Backpropagation



Backpropagation

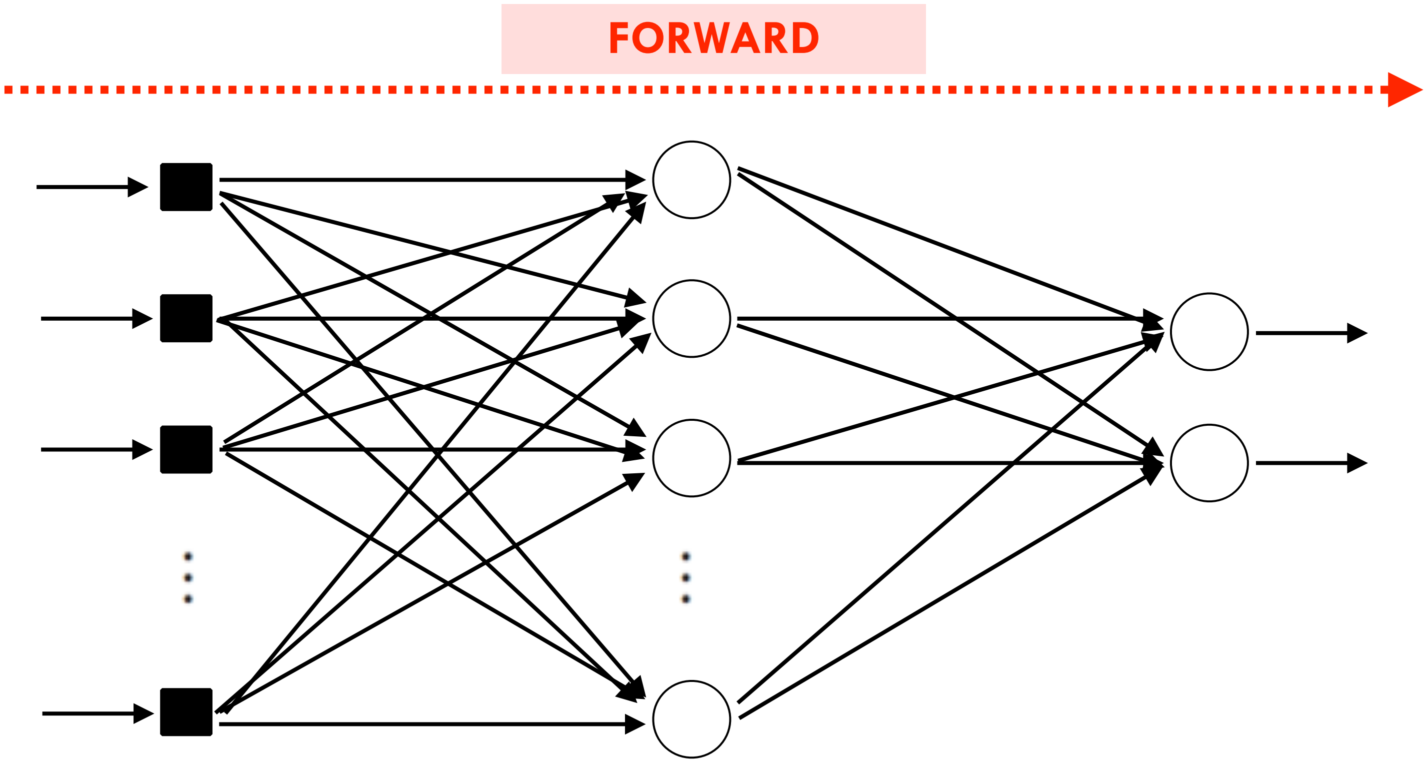


Os pesos sinápticos são matrizes:

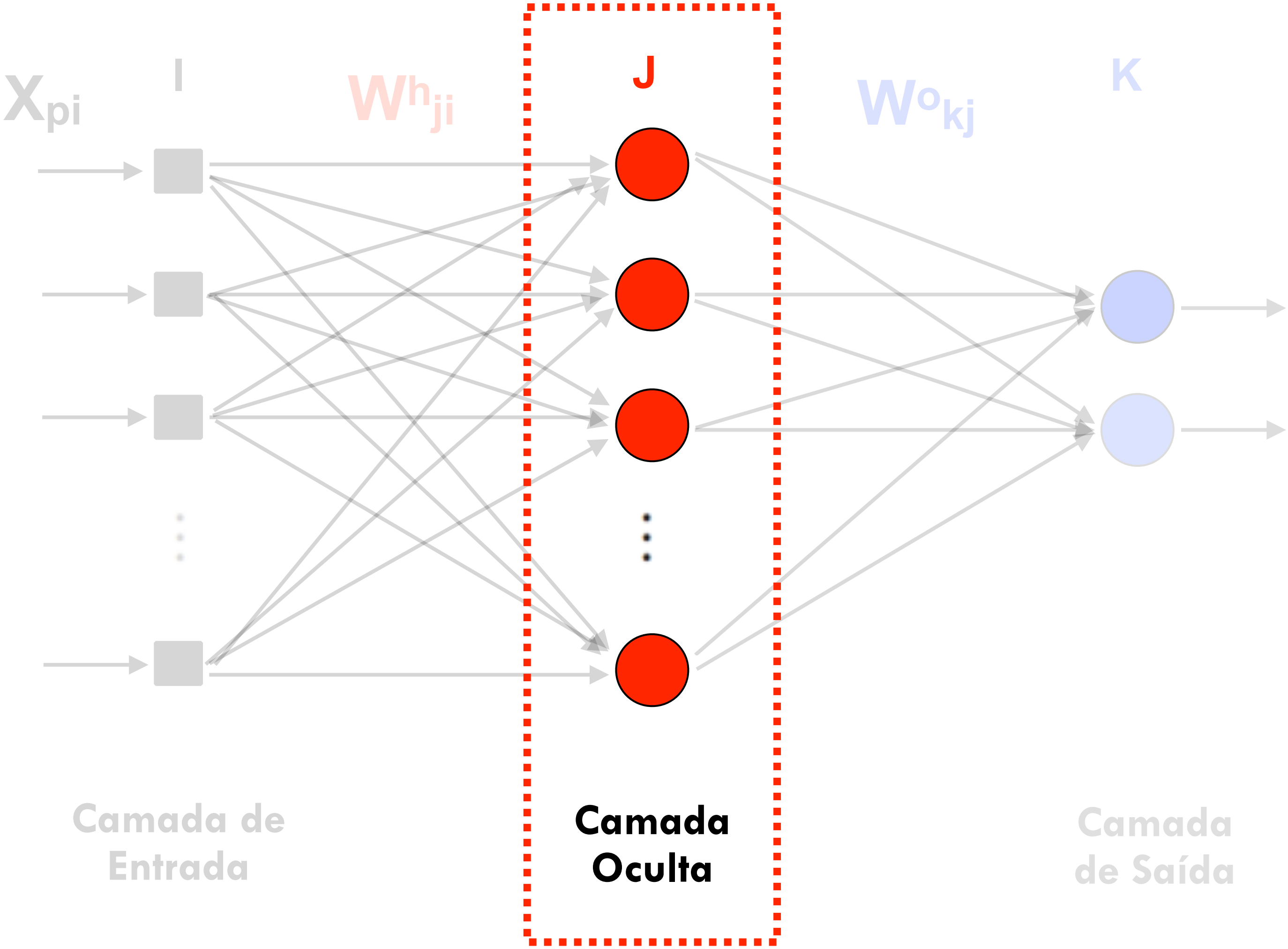
W^h - conecta camada **oculta** e a camada de **entrada**

W^o - conecta a camada de **saída** e a camada **oculta**

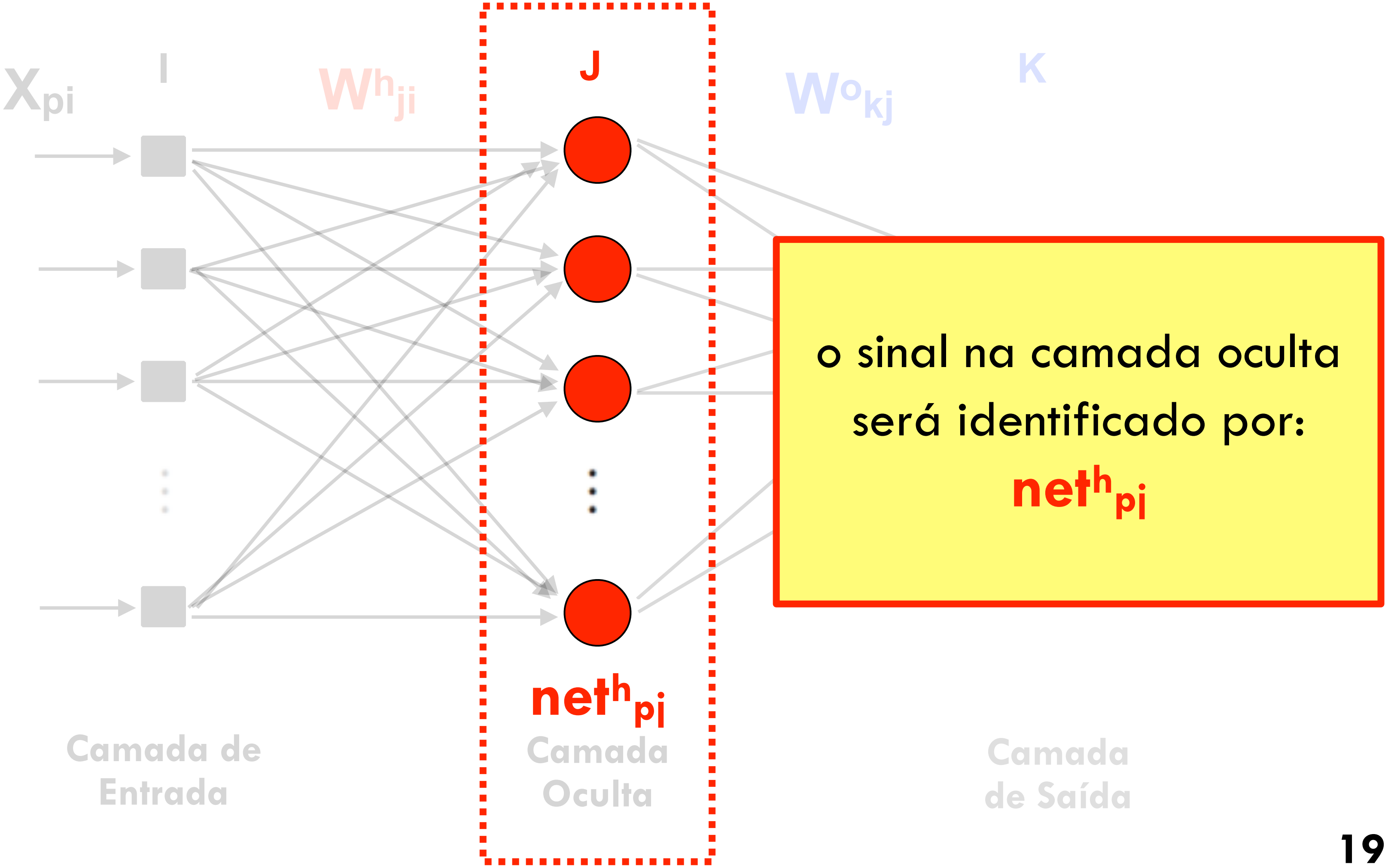
Backpropagation



Backpropagation



Backpropagation



Backpropagation

1 Sinal no neurônio de índice J na camada escondida:

$$net_{pj}^{h} = \sum_{i=1}^{N} W_{ji}^{h}\, x_{pi} + \theta_{j}^{h}$$

Backpropagation

1 Sinal no neurônio de índice J na camada escondida:

$$net_{pj}^{h} = \sum_{i=1}^{N} W_{ji}^{h} \, x_{pi} + \theta_{j}^{h}$$

N : número de neurônios na camada de entrada

Backpropagation

1. Sinal no neurônio de índice J na camada escondida:

$$net_{pj}^{h} = \sum_{i=1}^{N} W_{ji}^{h}\, x_{pi} + \theta_j^{h}$$

N : número de neurônios na camada de entrada

x_{pi} : exemplo de entrada fornecido para a rede

Backpropagation

1 Sinal no neurônio de índice J na camada escondida:

$$net_{pj}^{h} = \sum_{i=1}^{N} W_{ji}^{h} x_{pi} + \theta_{j}^{h}$$

N : número de neurônios na camada de entrada

x_{pi} : exemplo de entrada fornecido para a rede

W_{ji}^{h} : pesos sinápticos que conectam os valores de entrada i ao neurônio j

Backpropagation

1 Sinal no neurônio de índice J na camada escondida:

$$net_{pj}^{h} = \sum_{i=1}^{N} W_{ji}^{h} x_{pi} + \theta_j^h$$

N : número de neurônios na camada de entrada

x_{pi} : exemplo de entrada fornecido para a rede

W_{ji}^{h} : pesos sinápticos que conectam os valores de entrada i ao neurônio J

θ_j^h : bias do neurônio J

Backpropagation

1 Sinal no neurônio de índice J na camada escondida:

$$net_{pj}^{h} = \sum_{i=1}^{N} W_{ji}^{h} \, x_{pi} + \theta_{j}^{h}$$

Backpropagation

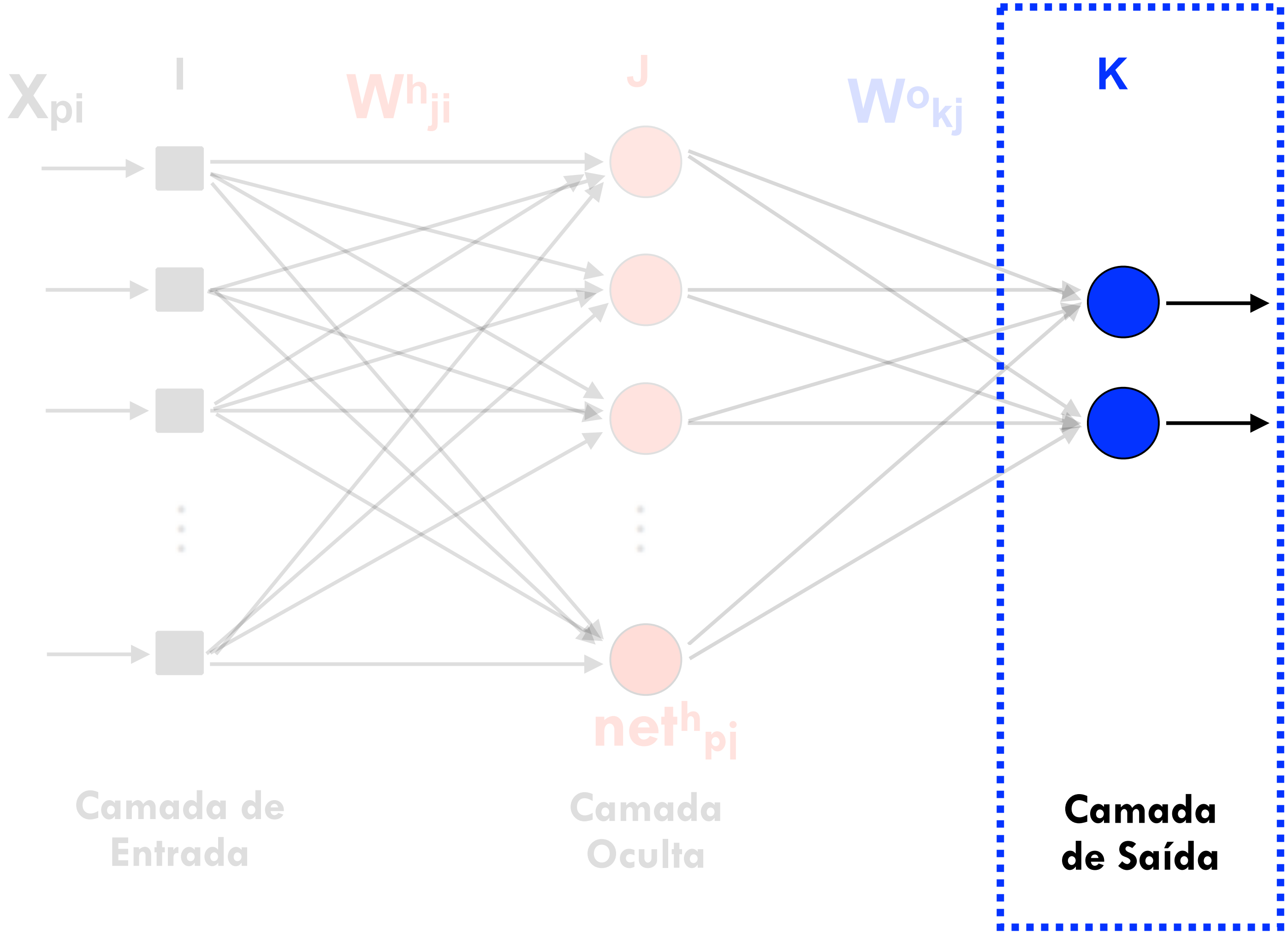
1 Sinal no neurônio de índice J na camada escondida:

$$net_{pj}^{h} = \sum_{i=1}^{N} W_{ji}^{h}\, x_{pi} + \theta_{j}^{h}$$

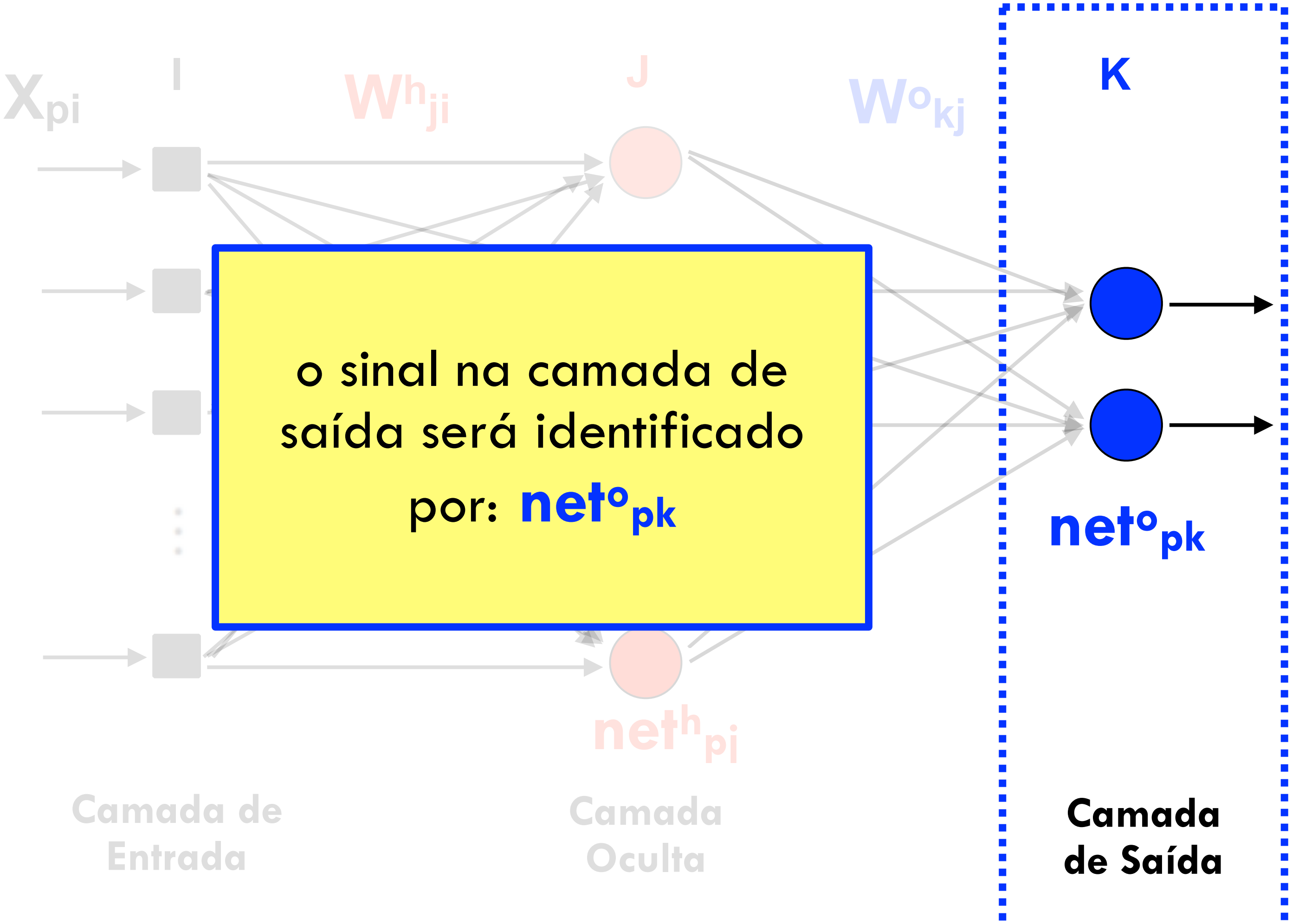
2 Ativação desse neurônio é igual a:

$$i_{pj} = f_{j}^{h}(net_{pj}^{h})$$

Backpropagation



Backpropagation



Backpropagation

3 Sinal no neurônio de índice K na camada de saída:

Backpropagation

3 Sinal no neurônio de índice K na camada de saída:

$$net_{pk}^{o} = \sum_{j=1}^{L} W_{kj}^{o}\, i_{pj} + \theta_{k}^{o}$$

Backpropagation

3 Sinal no neurônio de índice K na camada de saída:

$$net_{pk}^{o} = \sum_{j=1}^{L} W_{kj}^{o}\, i_{pj} + \theta_{k}^{o}$$

L : número de neurônios na camada oculta

i_{pj} : valor da ativação do neurônio i na camada oculta

W_{kj}^{o} : pesos sinápticos que conectam o neurônio J com a camada escondida

θ_{k}^{o} : bias do neurônio k

Backpropagation

3 Sinal no neurônio de índice K na camada de saída:

$$net_{pk}^{o} = \sum_{j=1}^{L} W_{kj}^{o}\, i_{pj} + \theta_{k}^{o}$$

Backpropagation

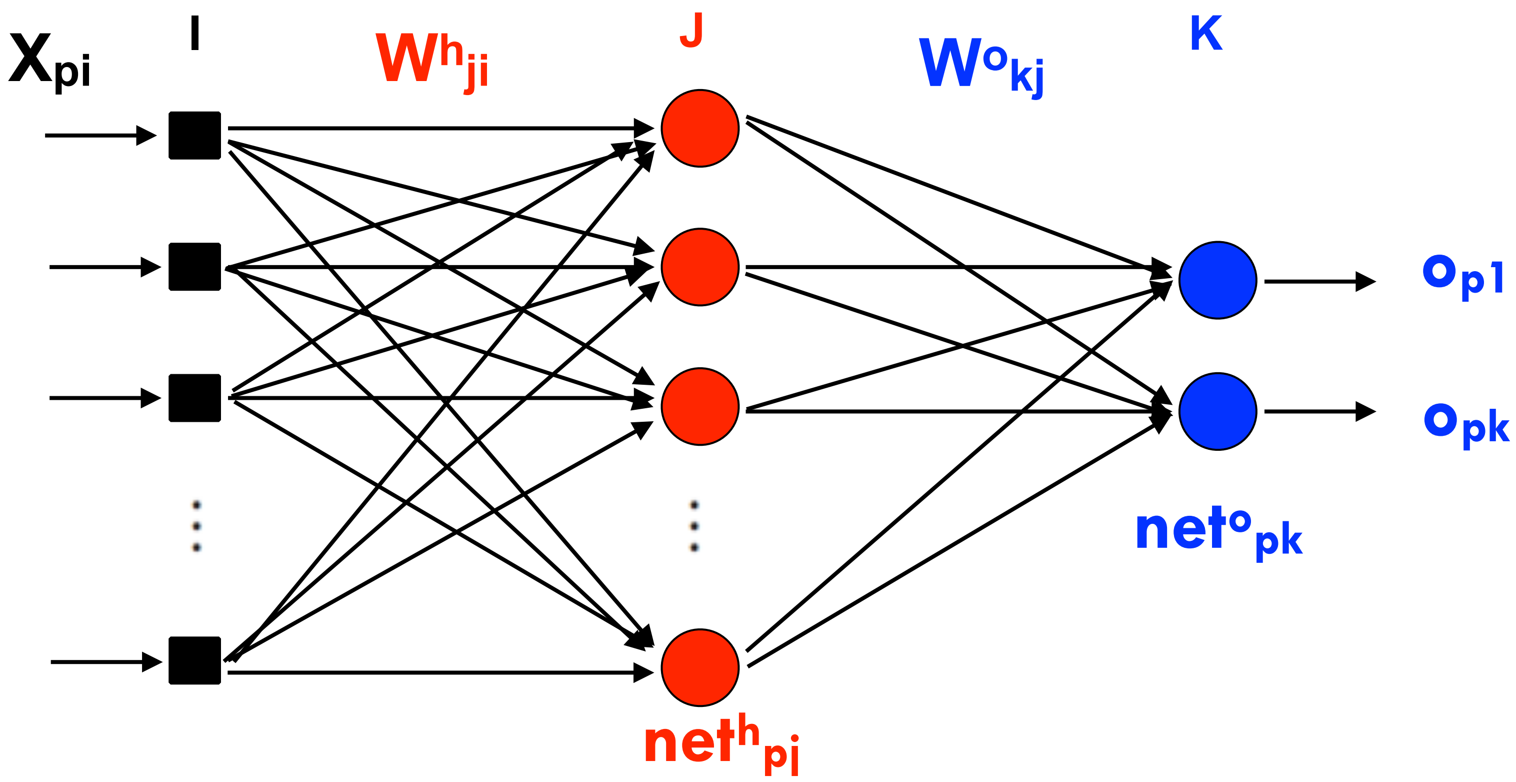
3 Sinal no neurônio de índice K na camada de saída:

$$net^{o}_{pk} = \sum_{j=1}^{L} W^{o}_{kj}\, i_{pj} + \theta^{o}_{k}$$

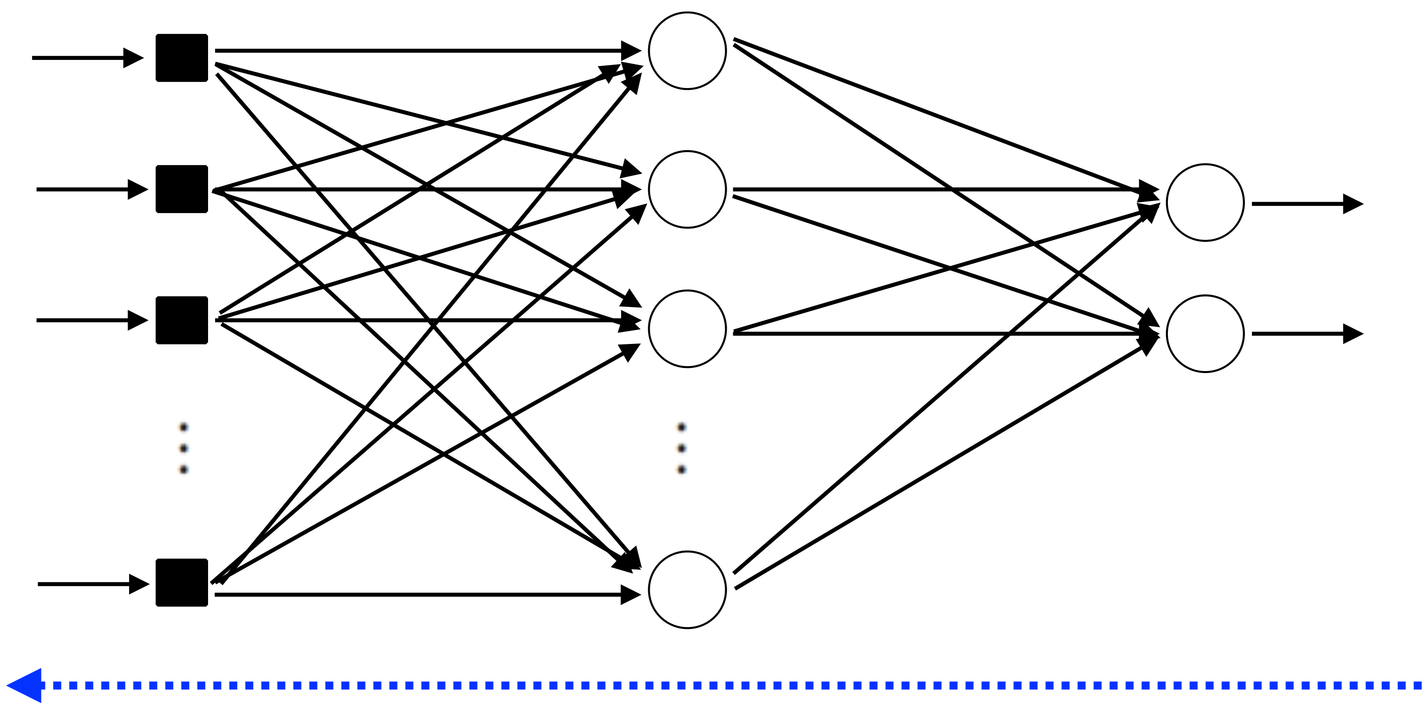
4 Saída desse neurônio é igual a:

$$o_{pk} = f^{o}_{k}(net^{o}_{pk})$$

Backpropagation

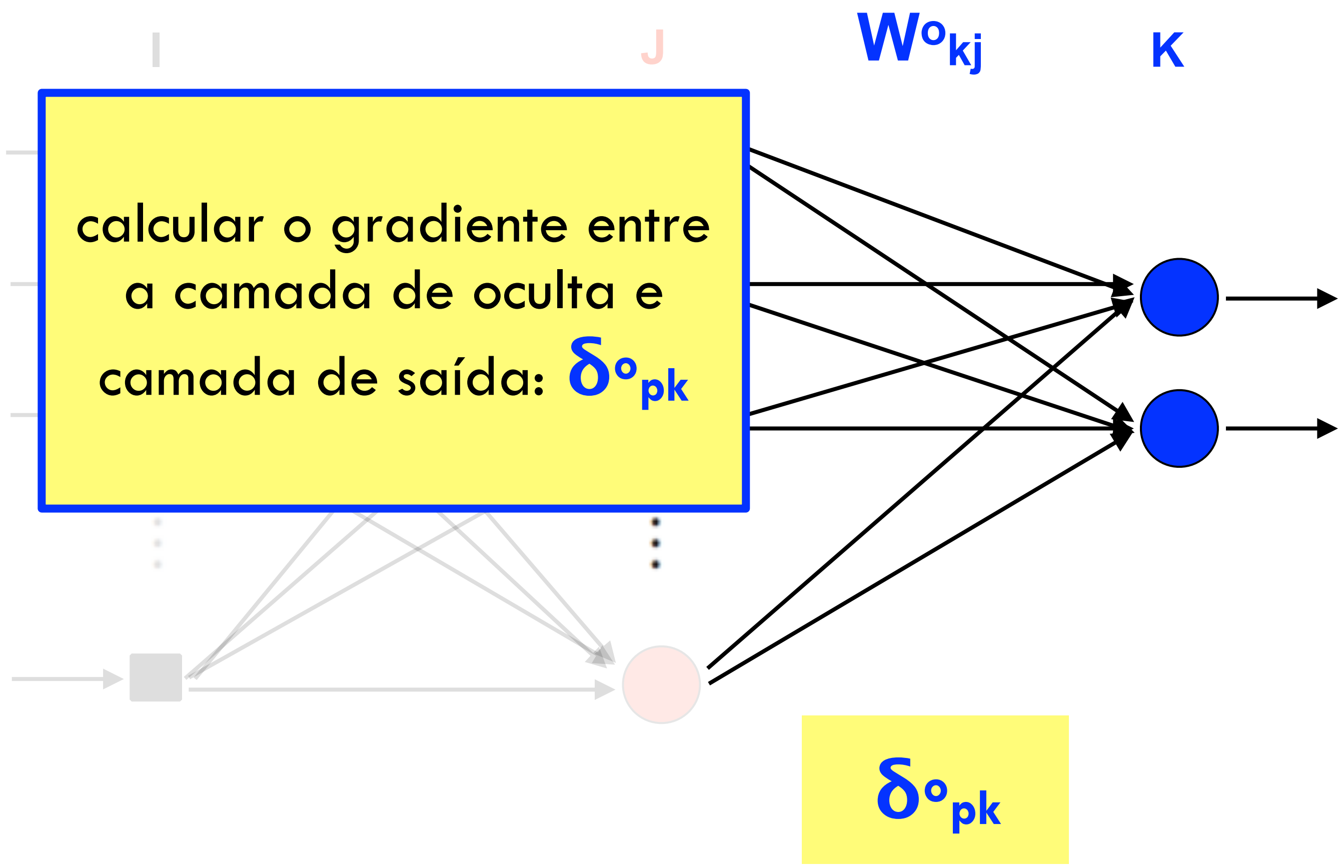


Backpropagation



BACKPROPAGATION

Backpropagation



Backpropagation

5 Calcular os termos de erro para as unidades de saída:

$$\delta^{o}_{pk} = (y_{pk} - o_{pk}) * f^{o\prime}_{k}(net^{o}_{pk})$$

Backpropagation

5 Calcular os termos de erro para as unidades de saída:

$$\delta^{o}_{pk} = (y_{pk} - o_{pk}) * f^{o\prime}_{k}(net^{o}_{pk})$$

y_{pk} : saída esperada no neurônio k

Backpropagation

5 Calcular os termos de erro para as unidades de saída:

$$\delta^{o}_{pk} = (y_{pk} - o_{pk}) * f^{o\prime}_{k}(net^{o}_{pk})$$

y_{pk} : saída esperada no neurônio k

o_{pk} : saída obtida no neurônio k

Backpropagation

5 Calcular os termos de erro para as unidades de saída:

$$\delta^o_{pk} = (y_{pk} - o_{pk}) * f^{o\prime}_k (net^o_{pk})$$

y_{pk} : saída esperada no neurônio k

o_{pk} : saída obtida no neurônio k

net^o_{pk} : sinal obtido no neurônio k

Backpropagation

5 Calcular os termos de erro para as unidades de saída:

$$\delta^{o}_{pk} = (y_{pk} - o_{pk}) * f^{o\prime}_{k}(net^{o}_{pk})$$

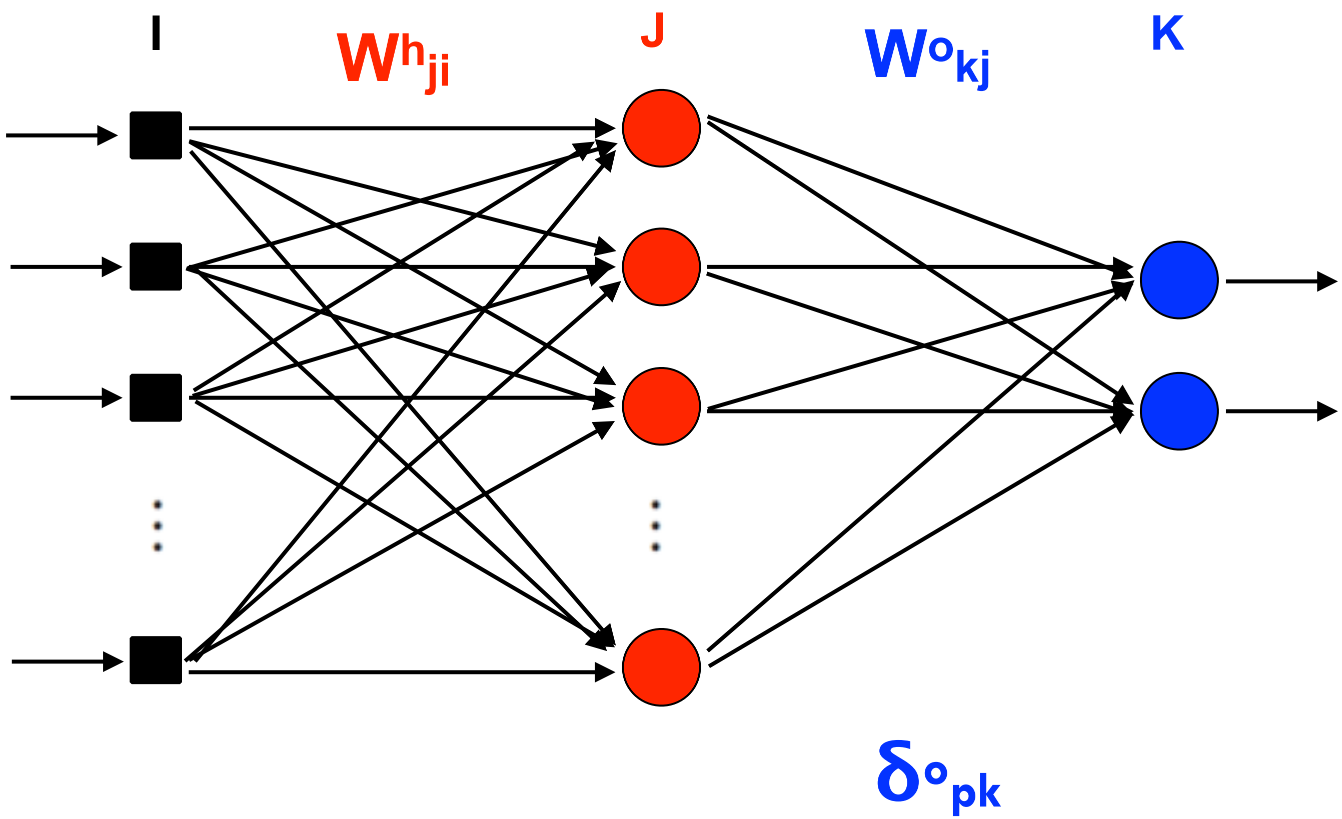
y_{pk} : saída esperada no neurônio k

o_{pk} : saída obtida no neurônio k

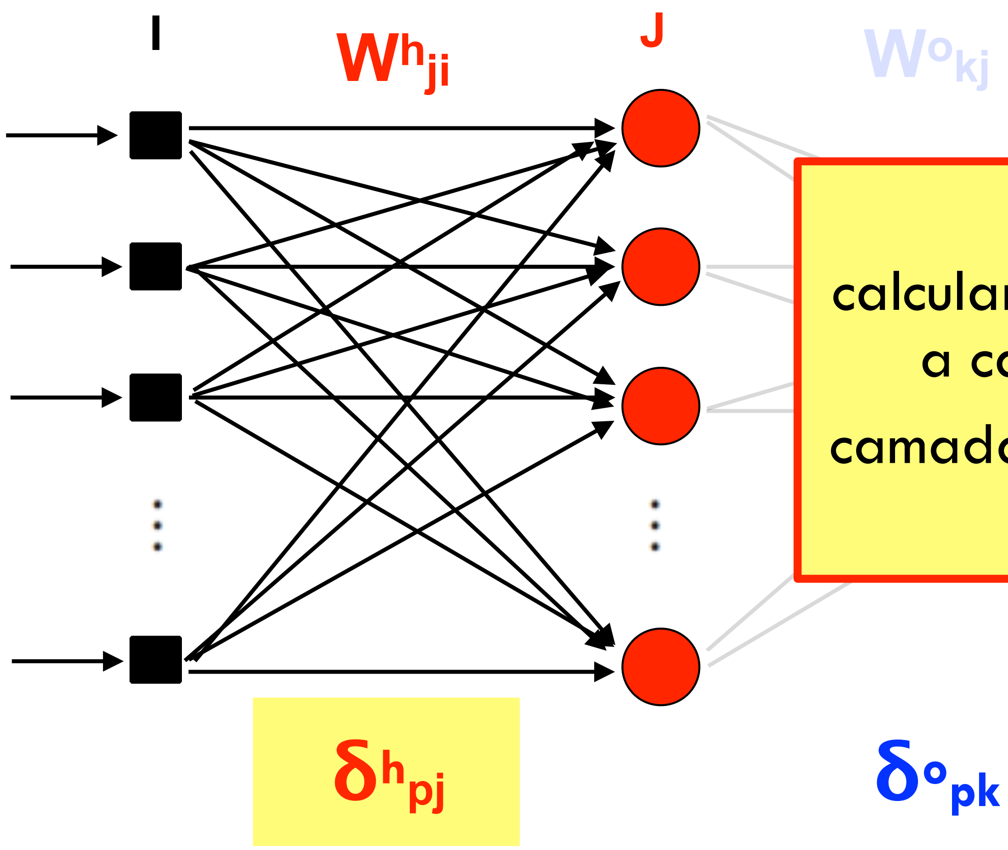
net^{o}_{pk} : sinal obtido no neurônio k

f'^{o}_{k} : derivada da função de ativação do neurônio k

Backpropagation



Backpropagation



calcular o gradiente entre a camada oculta e camada de entrada: δ^h_{pj}

Backpropagation

6 Calcular os termos de erro para as unidades ocultas:

$$\delta^{h}_{pj} = f^{h\prime}_{j}(net^{h}_{pj}) \sum_{k} (\delta^{o}_{pk} * W^{o}_{kj})$$

Backpropagation

6 Calcular os termos de erro para as unidades ocultas:

$$\delta^h_{pj} = f^{h\prime}_j(net^h_{pj}) \sum_k (\delta^o_{pk} * W^o_{kj})$$

δ^o_{pk}

W^o_{kj}

net^h_{pj}

$f_j^{\prime h}$

Backpropagation

6 Calcular os termos de erro para as unidades ocultas:

$$\delta^{h}_{pj} = f^{h\prime}_{j}(net^{h}_{pj}) \sum_{k} (\delta^{o}_{pk} * W^{o}_{kj})$$

δ^{o}_{pk} : termo de erro (gradiente) para o neurônio k da camada de saída

W^{o}_{kj} : pesos sinápticos que conectam o neurônio j da camada oculta a todos os neurônios k da camada de saída

net^{h}_{pj} : sinal obtido no neurônio j

$f_j^{\prime h}$: derivada da função de ativação do neurônio j

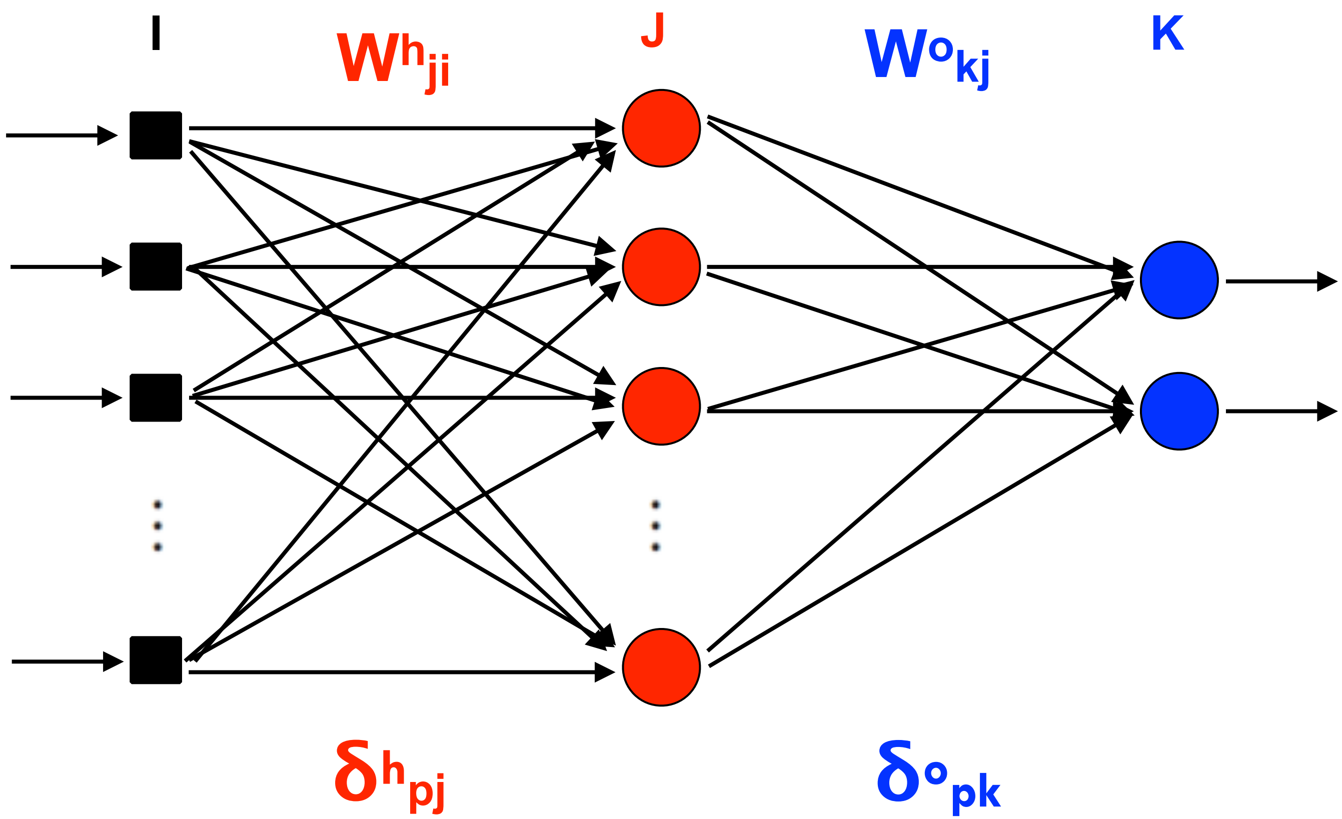
Backpropagation

6 Calcular os termos de erro para as unidades ocultas:

$$\delta^{h}_{pj} = f^{h\prime}_{j}(net^{h}_{pj}) \sum_{k} (\delta^{o}_{pk} * W^{o}_{kj})$$

Obs: o erro das unidades ocultas é calculado **ANTES** do ajuste de pesos da camada de saída

Backpropagation



Backpropagation

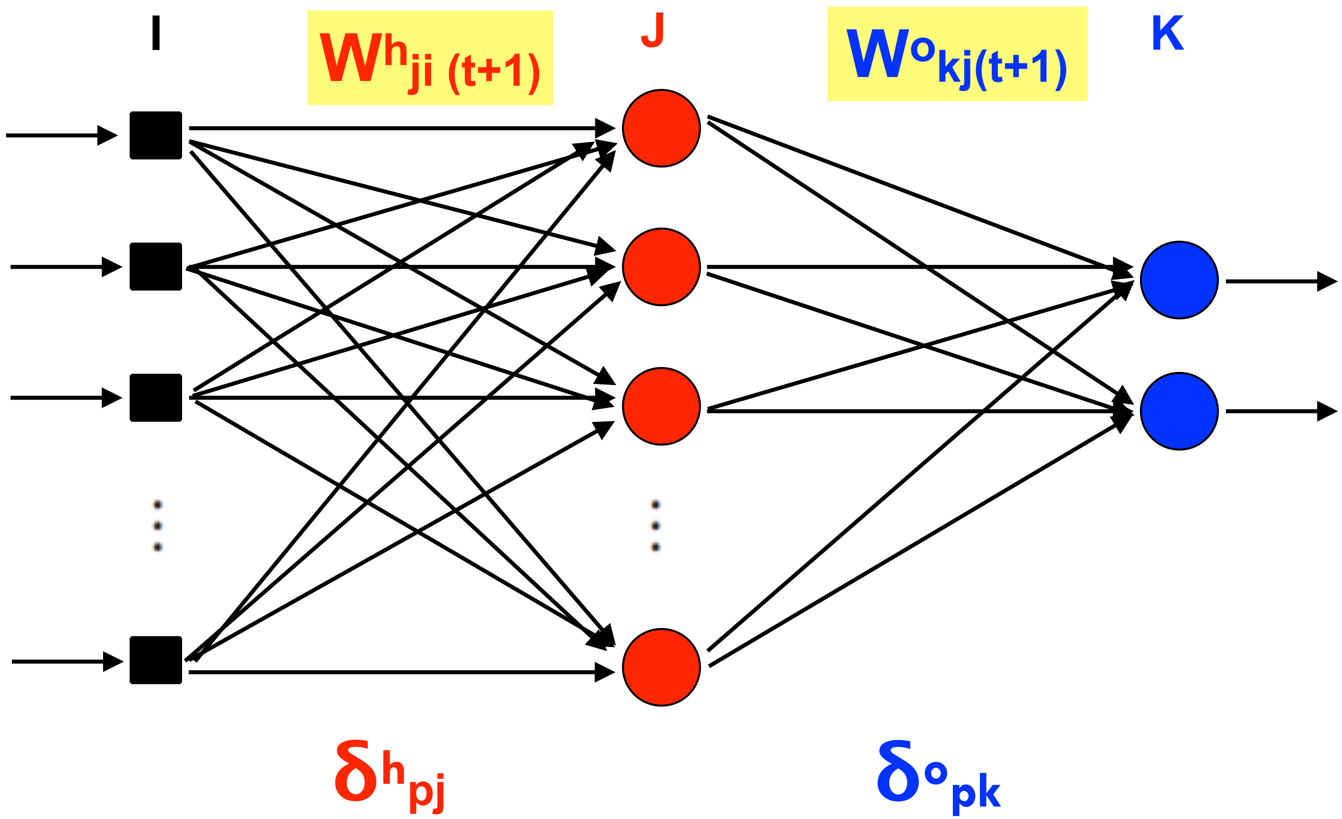
7 Atualizar os pesos da camada de saída

$$W^{o}_{kj}(t+1) = W^{o}_{kj}(t) + \eta * \delta^{o}_{pk} * i_{pj}$$

8 Atualizar os pesos da camada oculta

$$W^{h}_{ji}(t+1) = W^{h}_{ji}(t) + \eta * \delta^{h}_{pj} * x_{pi}$$

Backpropagation



Backpropagation

9 Calcular o erro total da **época**

- Indica o quão bem a rende está aprendendo.
- quando for menor que um limiar (***threshold***), parar

Algoritmo resumido

***Multilayer Perceptron* (MLP):**

Inicio algoritmo

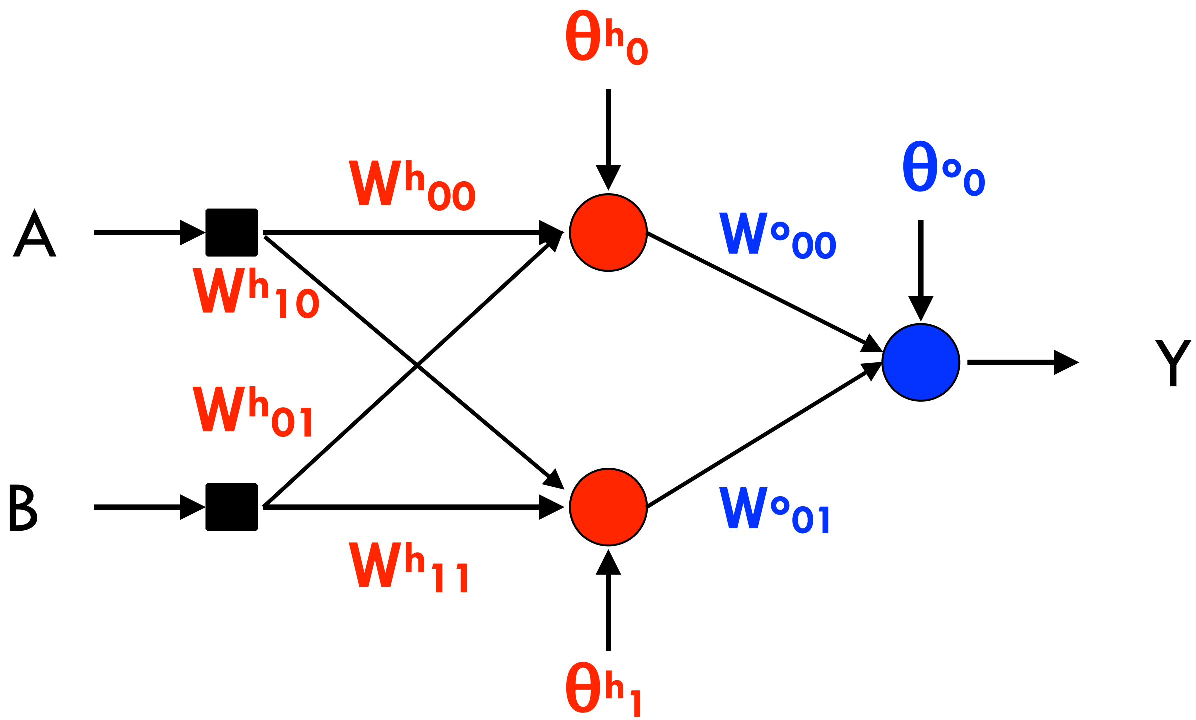
1. Aplicar um **vetor de entrada** para a rede (**X**) e calcular os valores de saída
2. Comparar as **saídas atuais** com as **saídas desejadas** e obter uma medida de **erro**
3. Determinar em qual **direção** (+ ou -) se deve modificar os pesos para minimizar o erro
4. Determinar a **quantidade** para se modificar cada peso
5. Aplicar as **correções aos pesos**
6. Repetir de 1 a 5 com todos os exemplos de treinamento, até que uma margem de erro de treinamento seja atingida

Fim algoritmo

Roteiro

1. Introdução
2. Backpropagation
3. **Exemplo**
4. Exercício
5. Referências

Exemplo



XOR dataset

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Exemplo

época	$\theta^h 0$	$\theta^h 1$	$\theta^o 0$	$W^h 00$	$W^h 10$	$W^h 01$	$W^h 11$	$W^o 00$	$W^o 01$
0	0.05	0.06	0.07	0.2	0.15	0.35	0.18	0.10	0.12
1									
2									

- $\eta = 0.2$
- $f(net) = 1/(1+ exp^{-net})$
- $f'(net) = net\ (1 - net)$

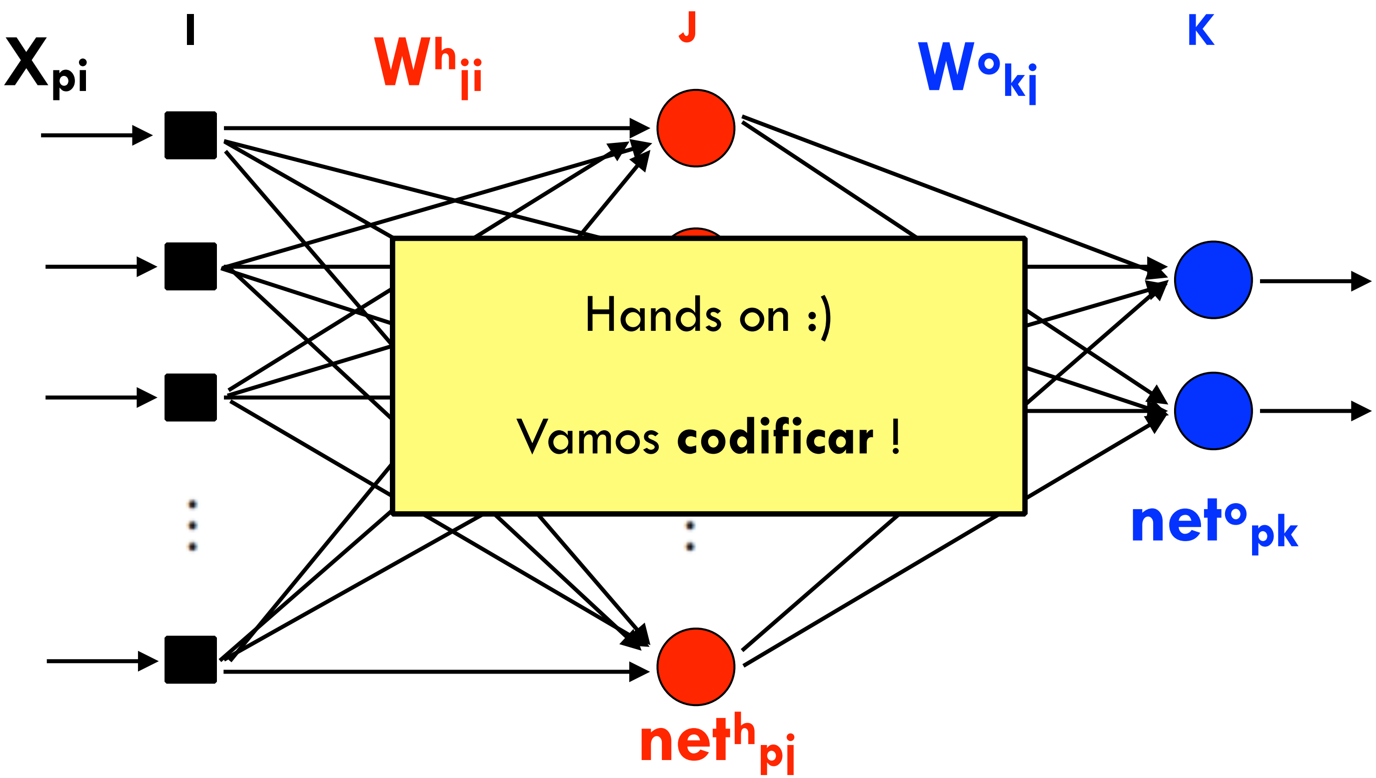
Roteiro

Exercício

época	$\theta^{h}0$	$\theta^{h}1$	$\theta^{o}0$	$W^{h}00$	$W^{h}10$	$W^{h}01$	$W^{h}11$	$W^{o}00$	$W^{o}01$
0	0.05	0.06	0.07	0.2	0.15	0.35	0.18	0.10	0.12
1									
2									

- X = XOR dataset
- $\eta = 0.2$
- **$f(net) = net^3 + 0.5$**
- **$f'(net) = 3*net^2$**

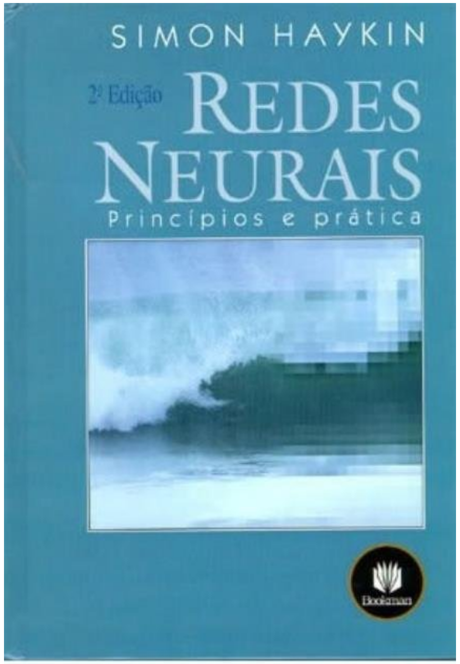
Exercício



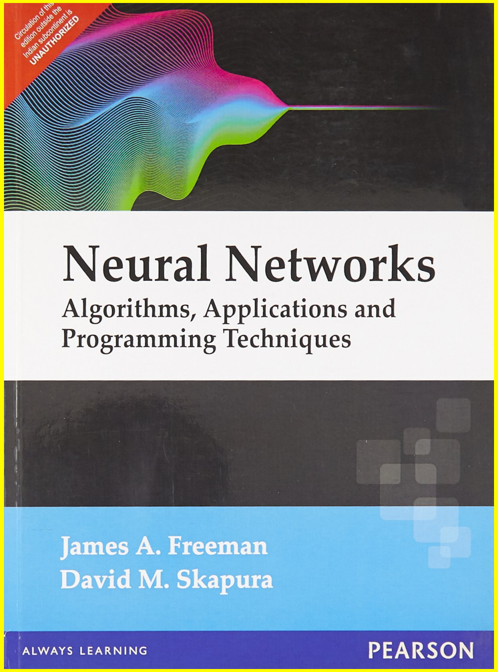
Roteiro

1. Introdução
2. Backpropagation
3. Exemplo
4. Exercício
5. Referências

Literatura Sugerida

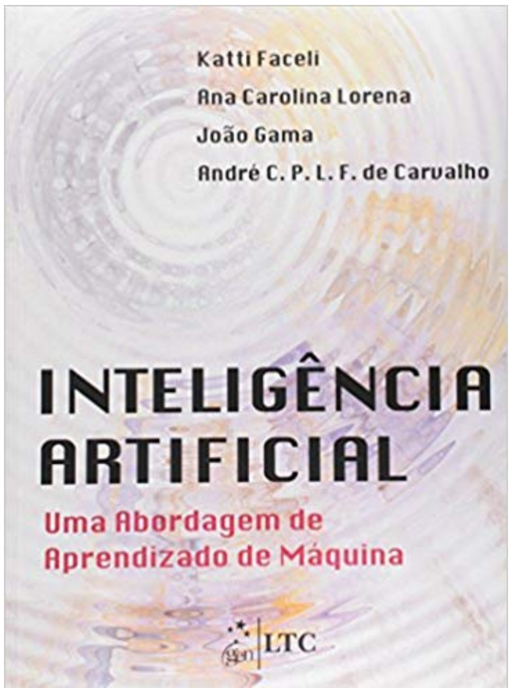


(Haykin, 1999)

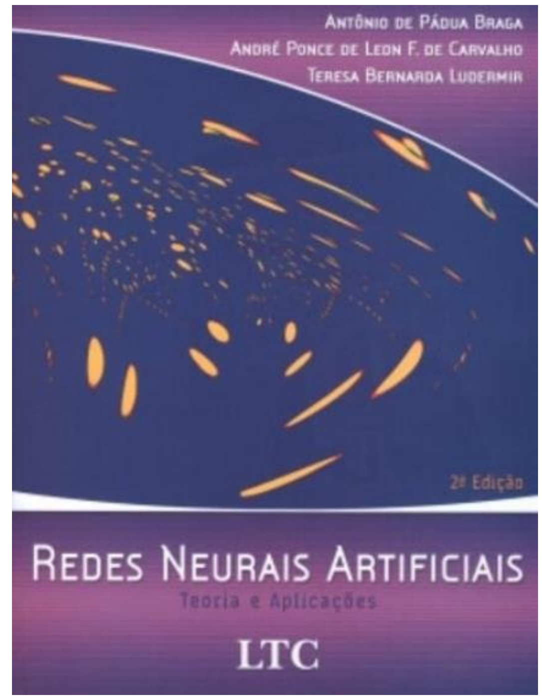


(Freeman & Skapura, 1991)

Literatura Sugerida



[Faceli et al, 2011]



[Braga et al, 2007]

Perguntas?

Prof. Rafael G. Mantovani

rgmantovani@gmail.com