FUNDAMENTOS DA INTELIGÊNCIA ARTIFICIAL

Aula 05 - *Backpropagation*

(*Multilayer Perceptrons* - MLPs)

Prof. Rafael G. Mantovani

Londrina - PR, Brasil
Outubro, 2018

Universidade Estadual de Londrina - UEL
Departamento de Computação - DC

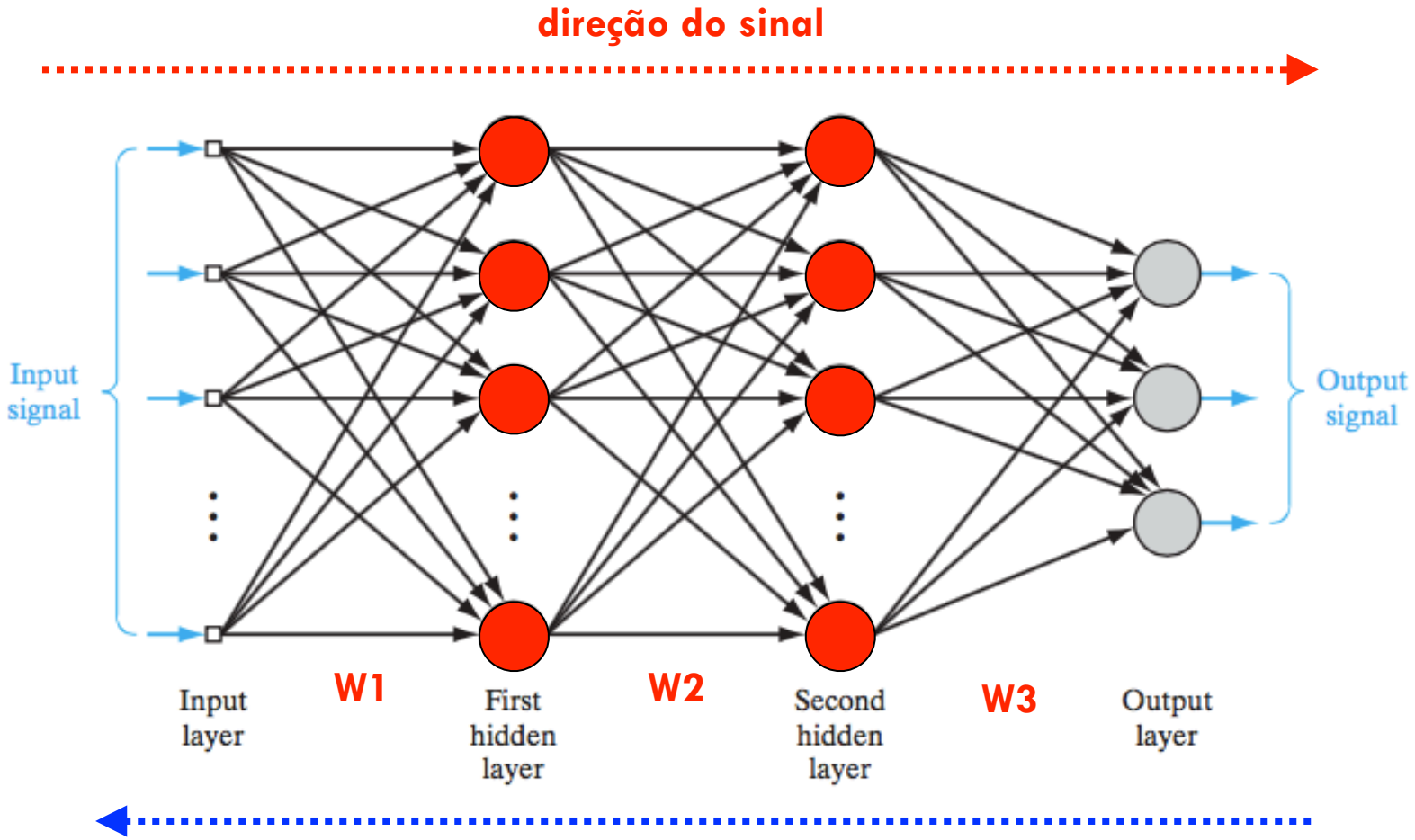
Roteiro

1. Introdução
2. *Backpropagation*
3. Algoritmo Resumido
4. Exemplo / Exercício
5. Síntese / Próximas Aulas
6. Referências

Roteiro

1. **Introdução**
2. ***Backpropagation***
3. **Algoritmo Resumido**
4. **Exemplo / Exercício**
5. **Síntese / Próximas Aulas**
6. **Referências**

Introdução


direção do sinal
Input signal
Output signal
Input layer
W1
First hidden layer
W2
Second hidden layer
W3
Output layer
direção do erro

Introdução

- Sinais de função e sinais de erro

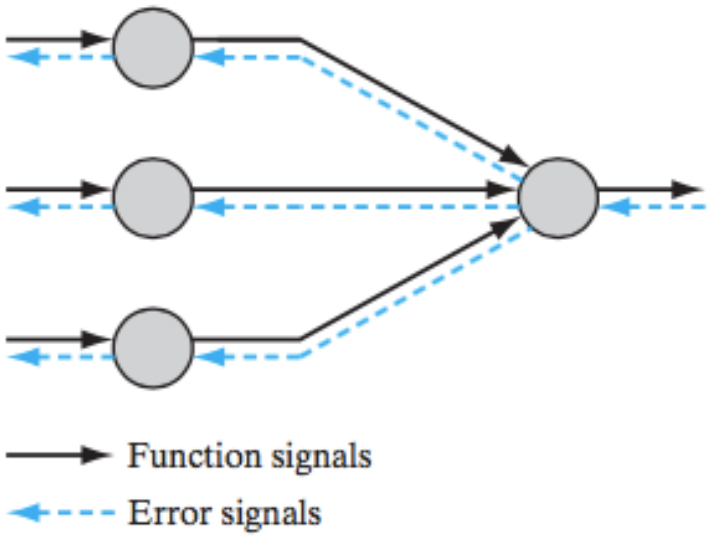
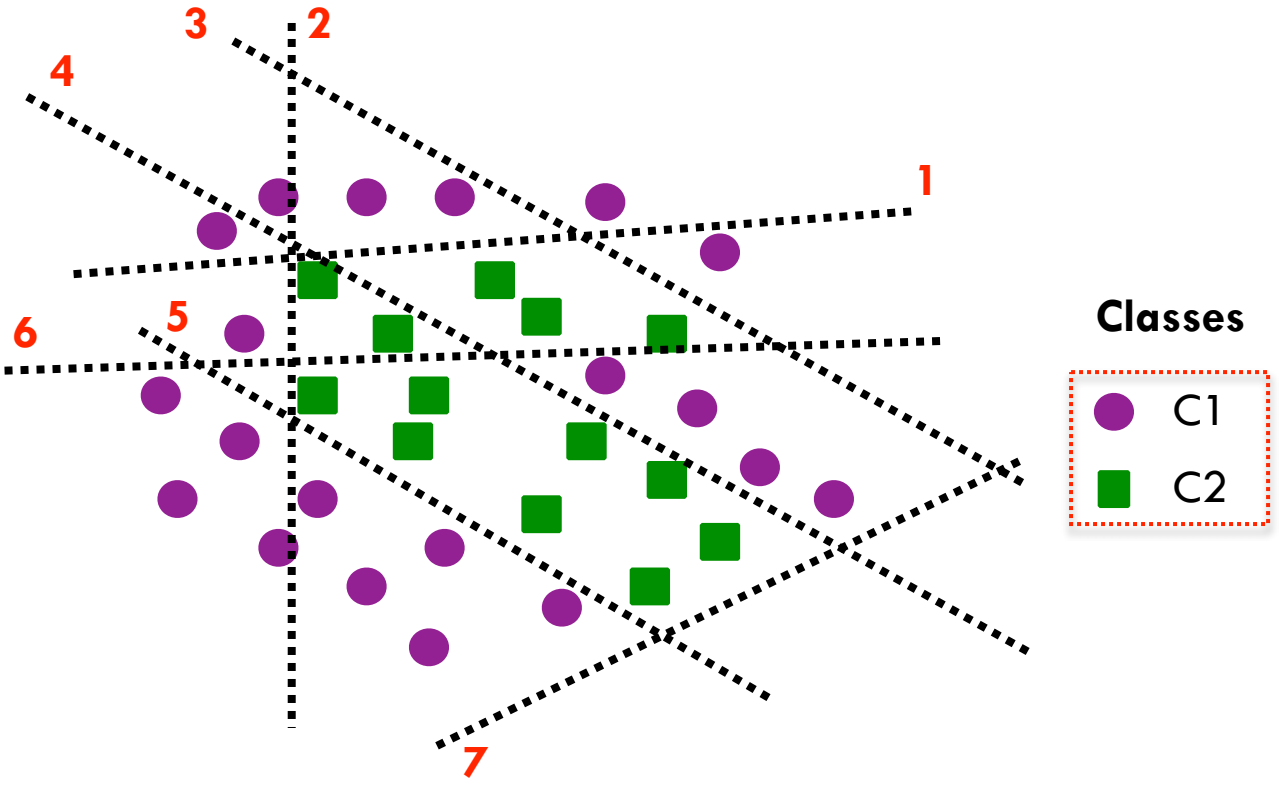


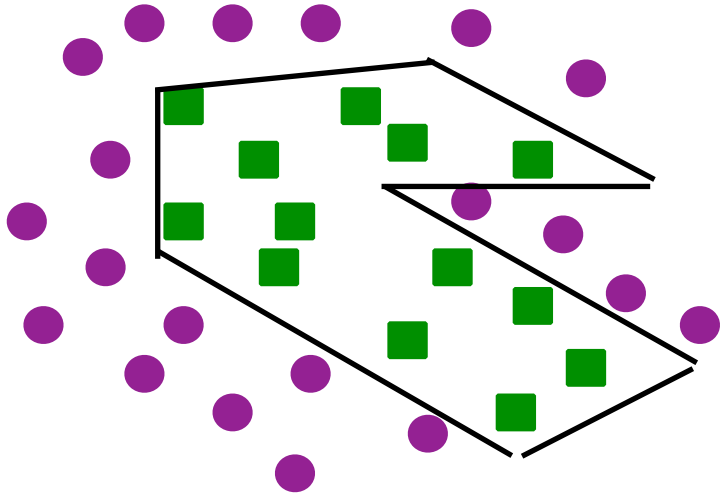
FIGURE 4.2 Illustration of the directions of two basic signal flows in a multilayer perceptron: forward propagation of function signals and back propagation of error signals.

Introdução



7 hiperplanos = 7 neurônios

Introdução

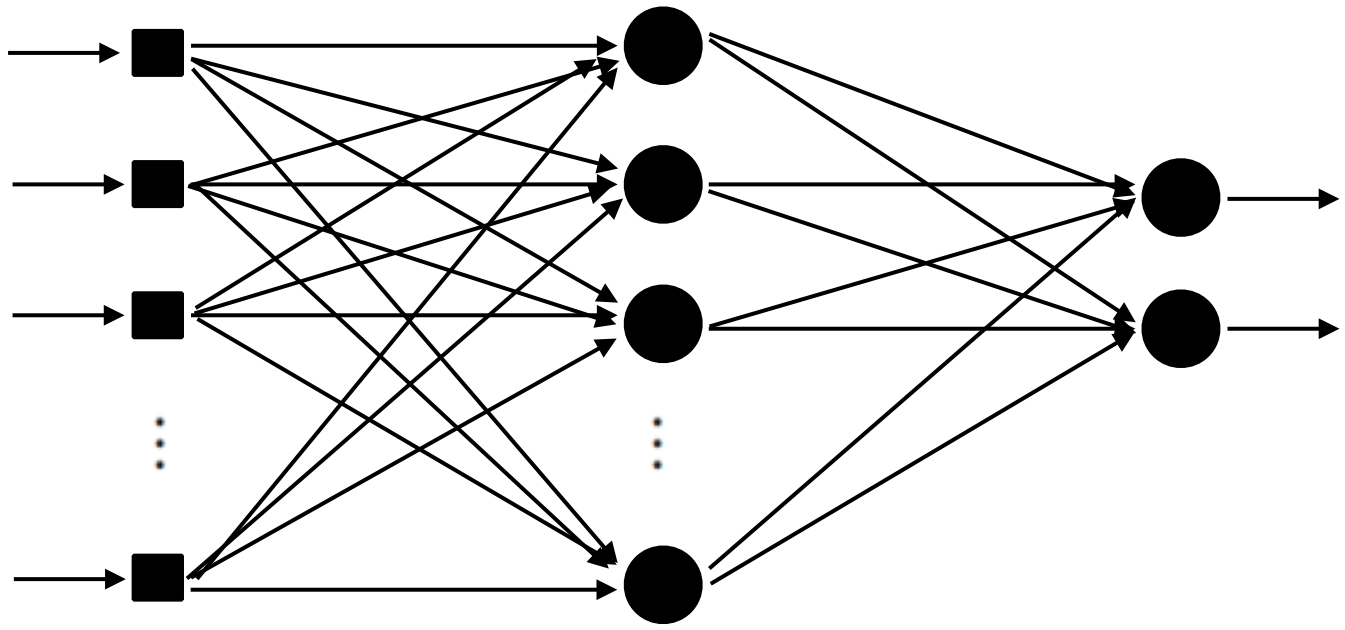


7 hiperplanos = 1 região convexa

Roteiro

1. Introdução
2. *Backpropagation*
3. Algoritmo Resumido
4. Exemplo / Exercício
5. Síntese / Próximas Aulas
6. Referências

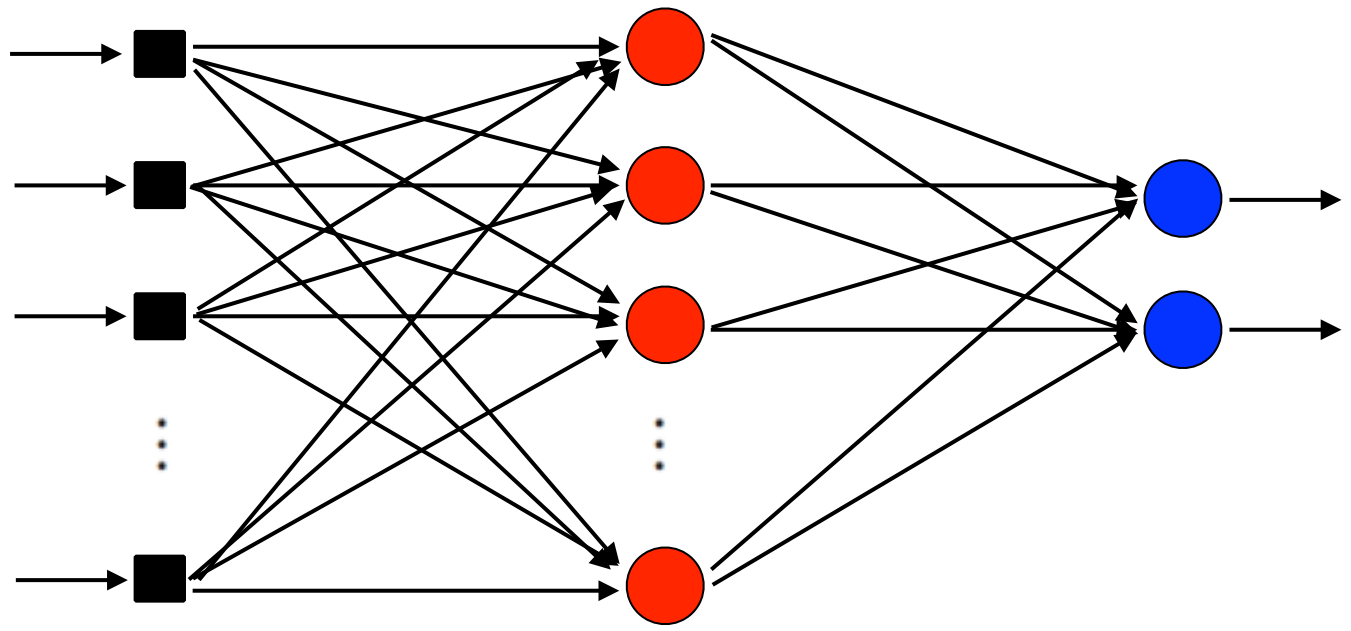
Backpropagation



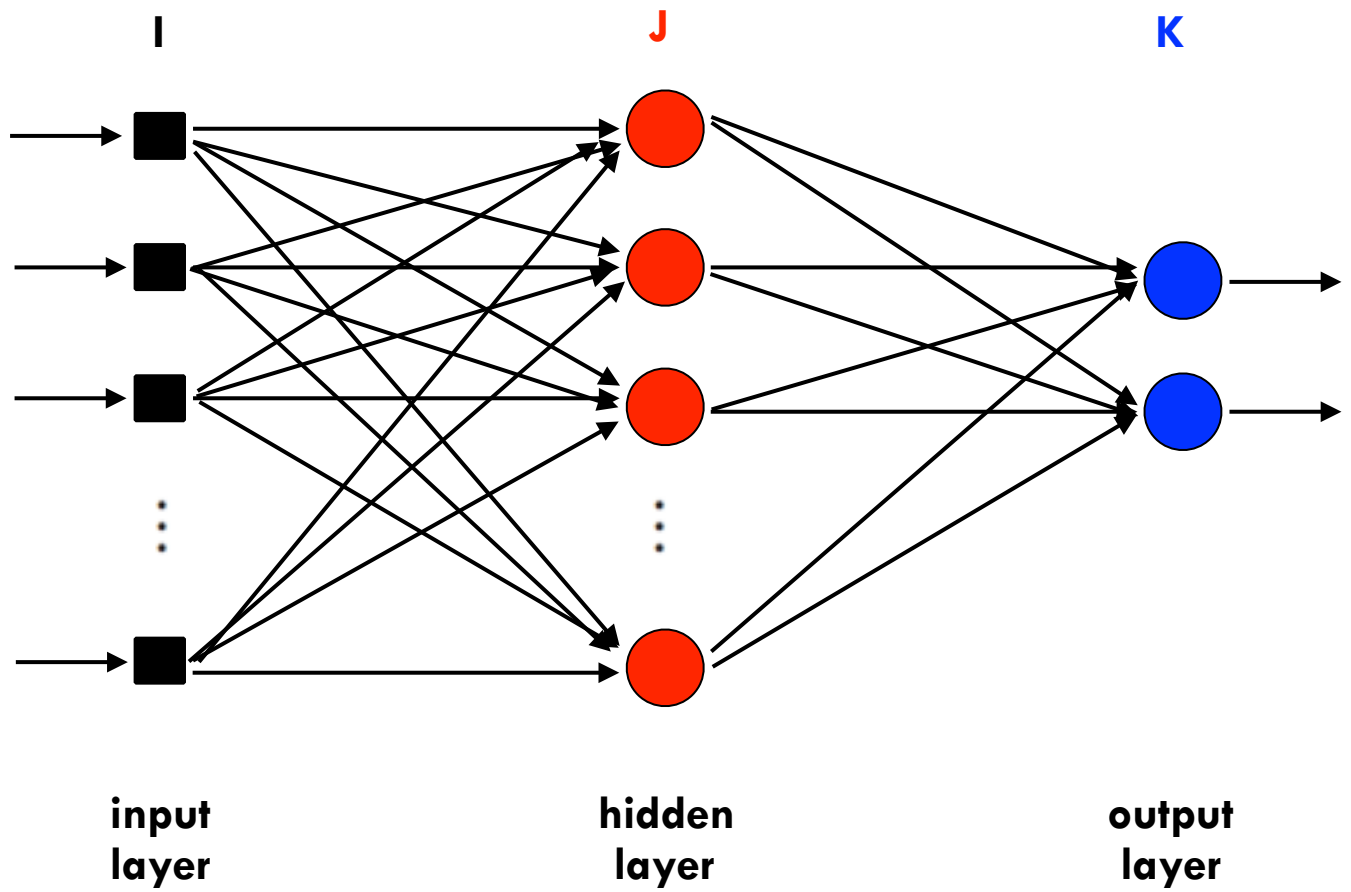
Backpropagation

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 -) e 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

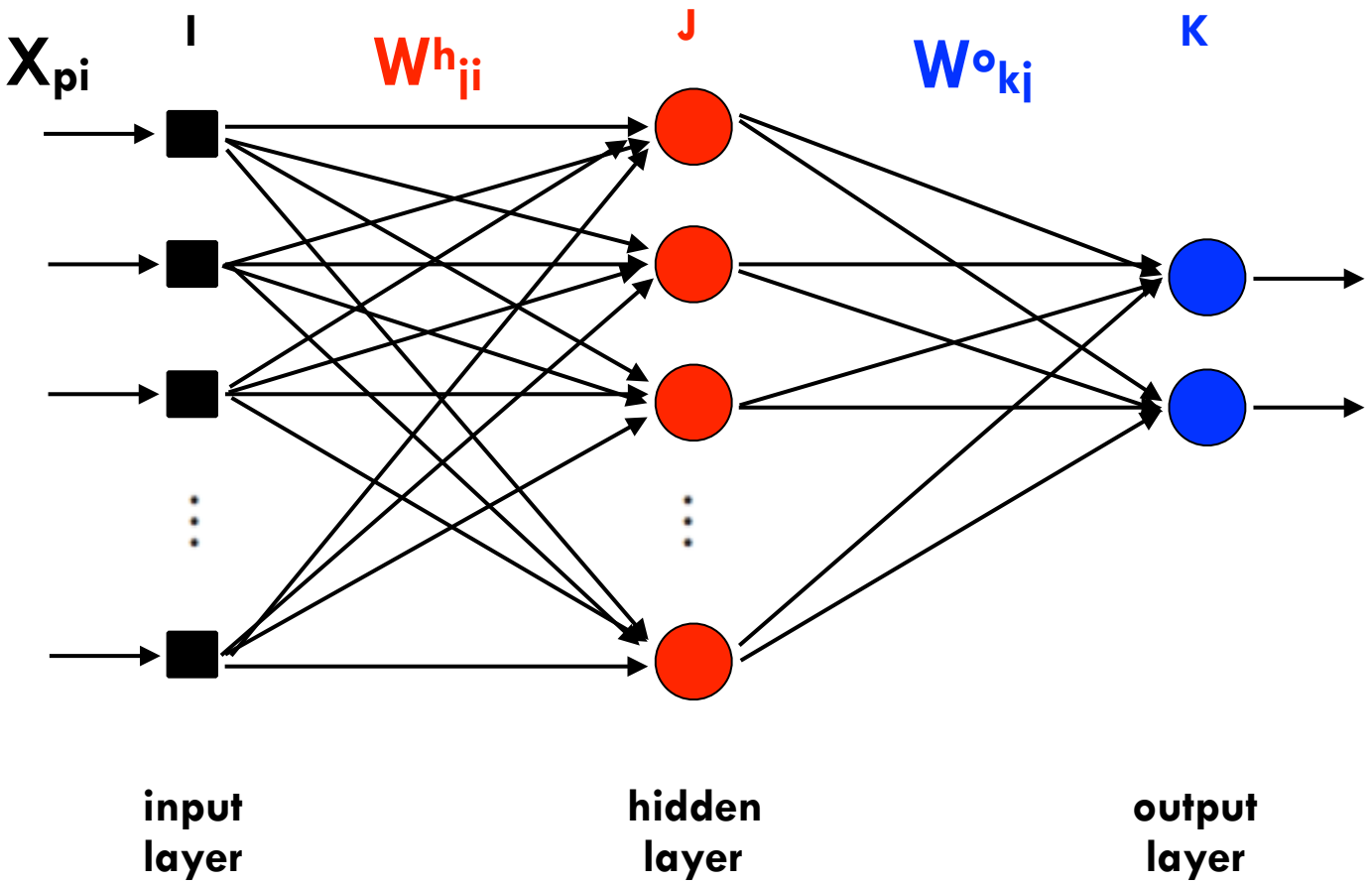
Backpropagation



Backpropagation



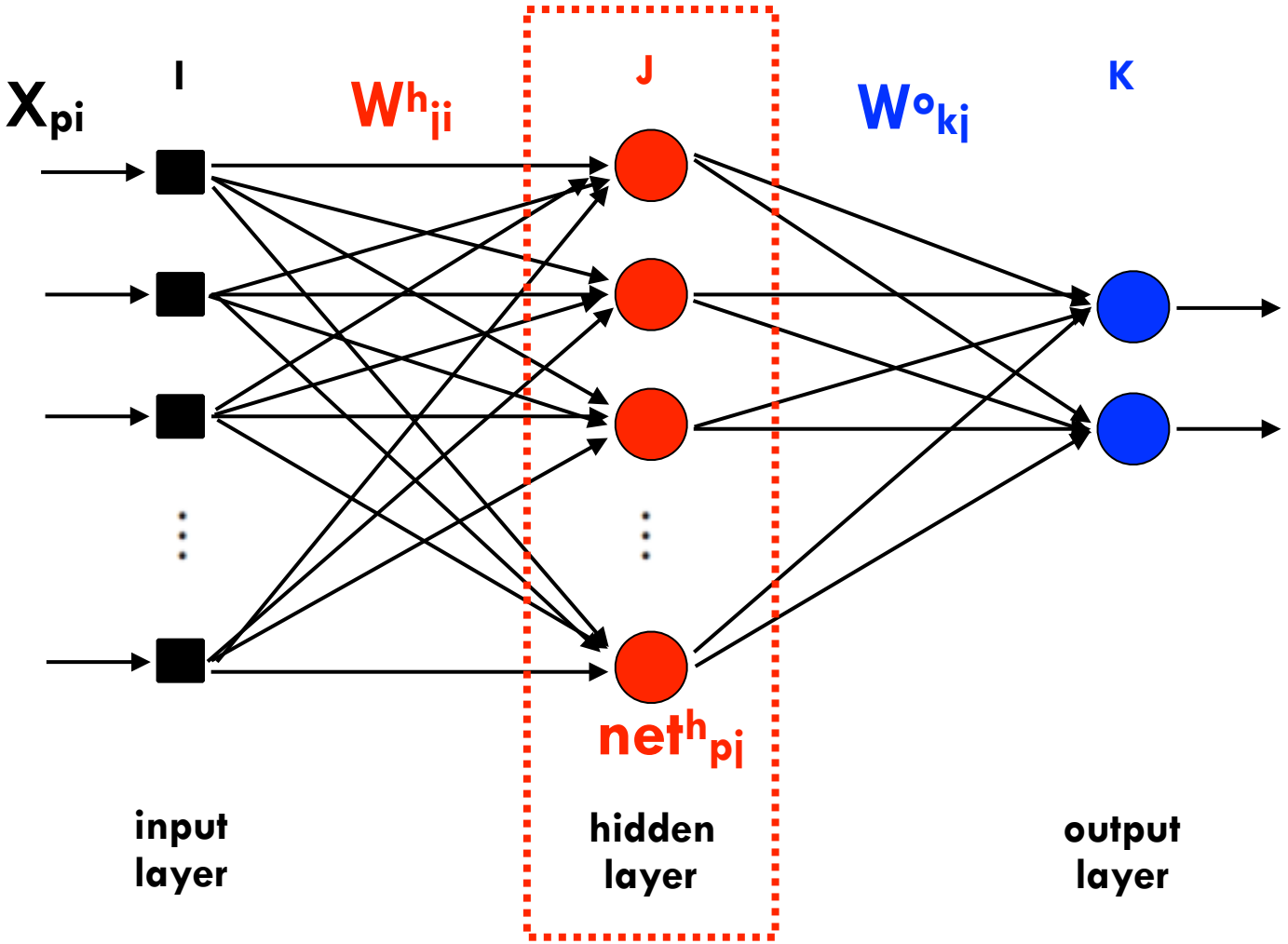
Backpropagation



Backpropagation

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 -) e 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

Backpropagation



Backpropagation

- Sinal no neurônio de índice J na camada escondida:

$$\mathbf{net}_{pj}^{h} = \sum_{i=1}^{N} w_{ji}^{h} x_{pi} + \theta_{j}^{h}$$

N → ?

w^{h}_{ji} → ?

x_{pi} → ?

θ^{h}_{j} → ?

Backpropagation

- Sinal no neurônio de índice J na camada escondida:

$$\mathbf{net}_{pj}^{h} = \sum_{i=1}^{N} w_{ji}^{h} x_{pi} + \theta_{j}^{h}$$

N → número de neurônios na camada de entrada

w^{h}_{ji} → peso da conexão com o neurônio de entrada i

x_{pi} → padrão inserido na entrada da rede

θ^{h}_{j} → bias do neurônio

Backpropagation

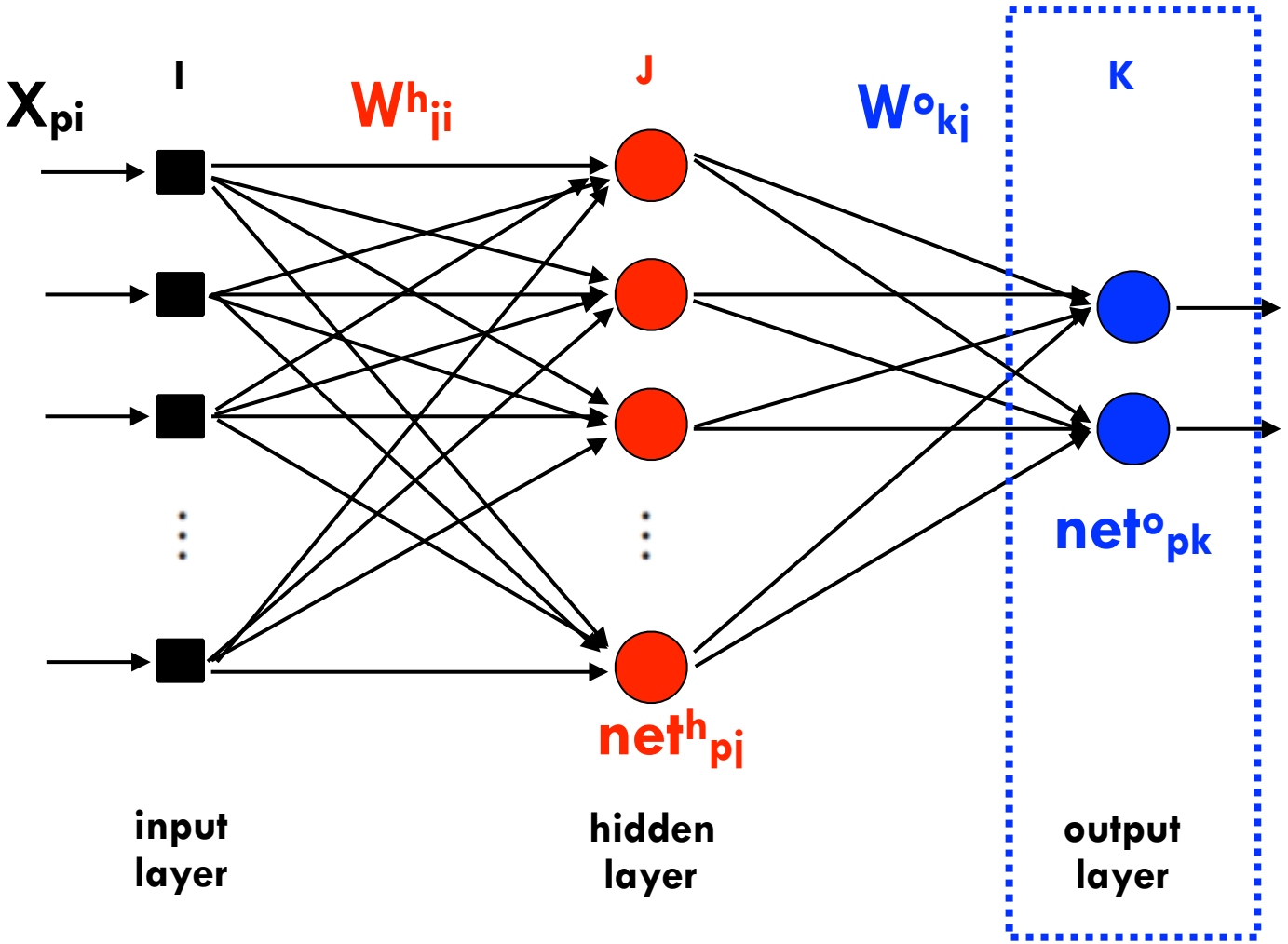
- Sinal no neurônio de índice J na camada escondida:

$$\mathbf{net}_{pj}^{h} = \sum_{i=1}^{N} w_{ji}^{h} x_{pi} + \theta_{j}^{h}$$

- Ativação desse neurônio é igual a:

$$i_{pj} = f^{h}_{j}(net^{h}_{pj})$$

Backpropagation



Backpropagation

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

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

L →

w^{o}_{pj} →

i_{pj} →

θ^{o}_{k} →

Backpropagation

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

$$\mathbf{net}_{pk}^{o} = \sum_{j=1}^{L} w_{kj}^{o} i_{pj} + \theta_{k}^{o}$$

L → número de neurônios na camada de saída

w^{o}_{pj} → peso da conexão com o neurônio j da camada escondida

i_{pj} → valor de ativação do neurônio j da camada escondida

θ^{o}_{k} → bias do neurônio

Backpropagation

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

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

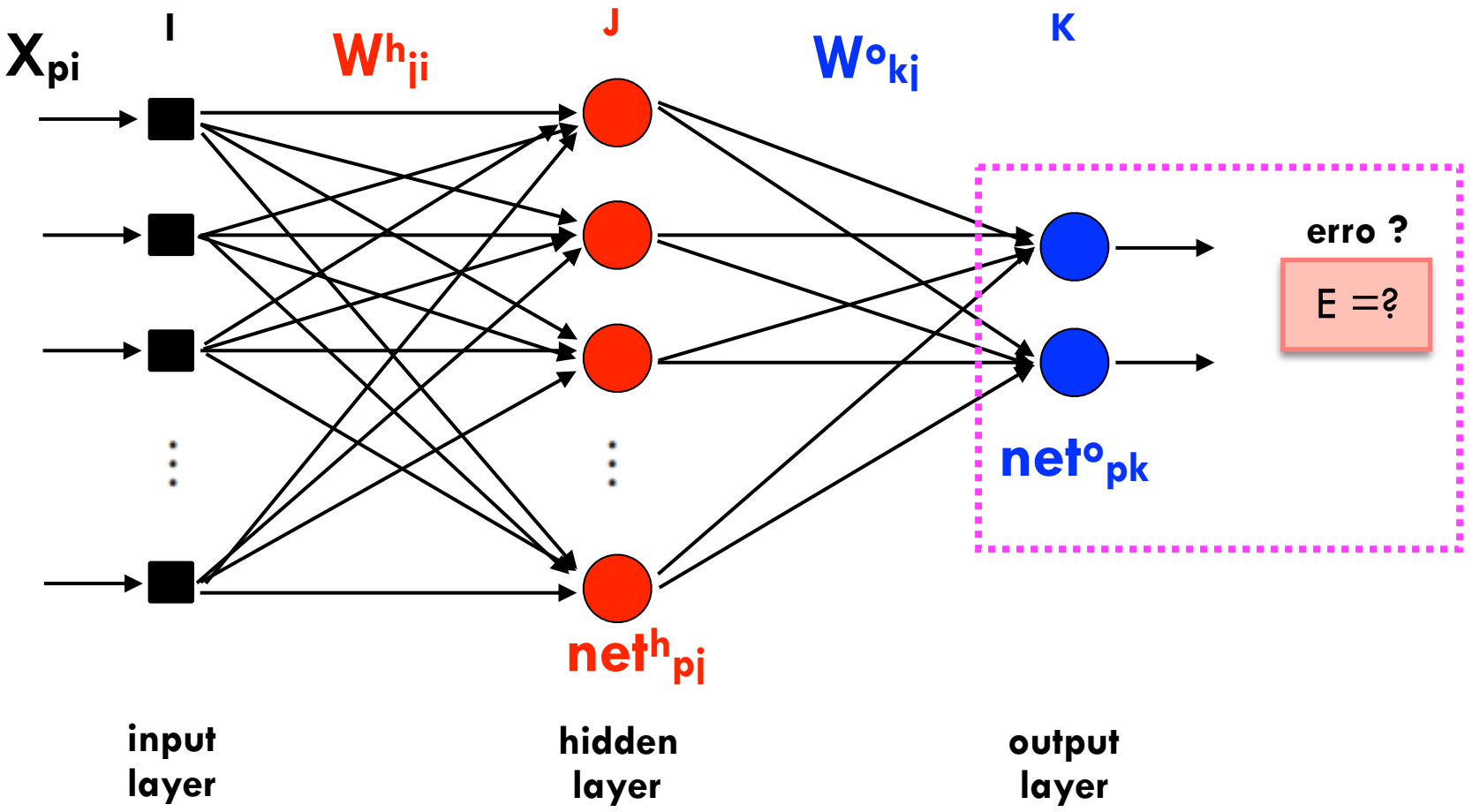
- Saída desse neurônio é igual a:

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

Backpropagation

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 -) e 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

Backpropagation



Backpropagation

- O erro de um único exemplo em uma única unidade de saída é definido por:

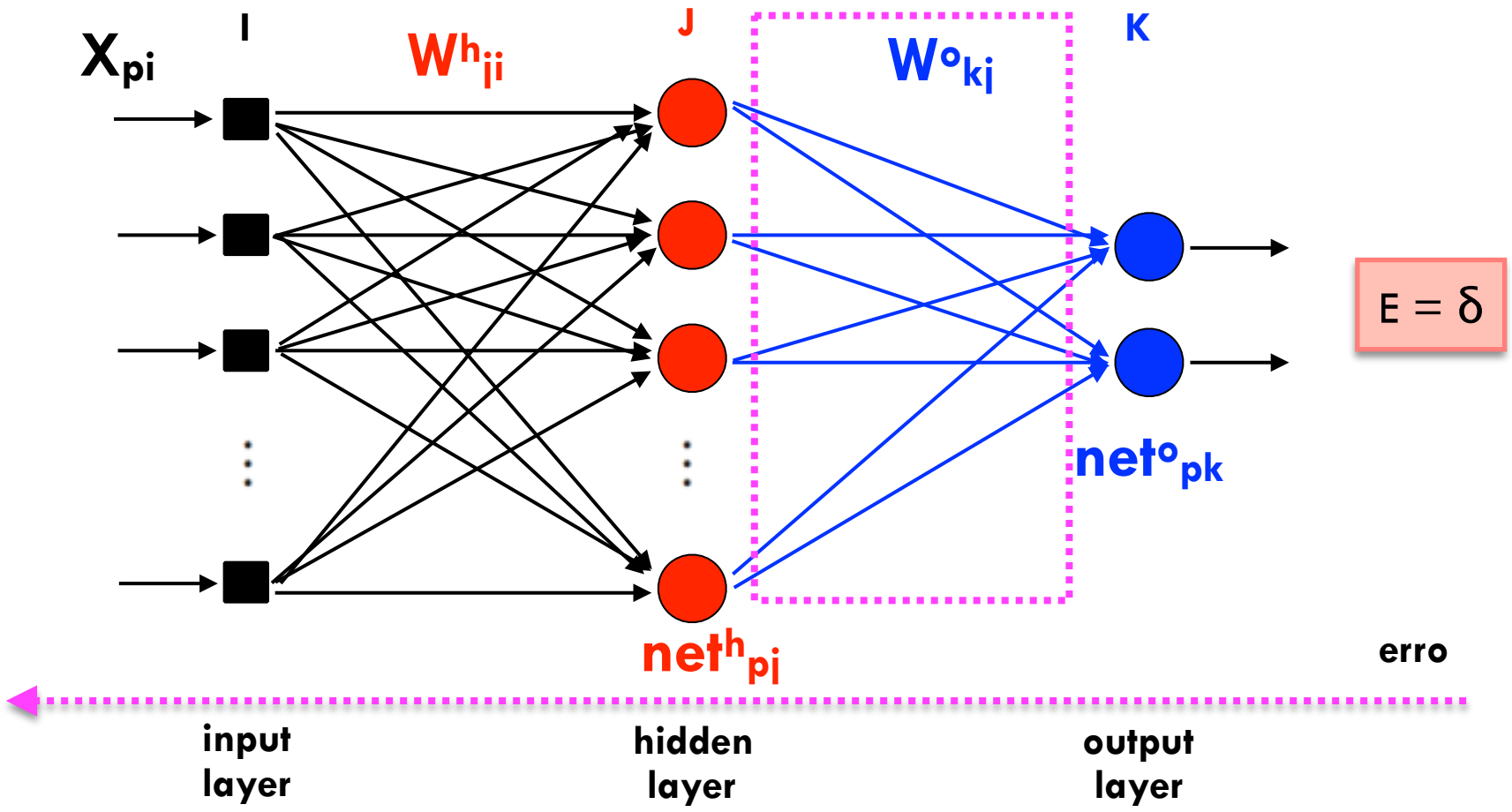
$$\delta_{pk} = (y_{pk} - o_{pk})$$

- onde:
  - p refere-se ao exemplo "p" do conjunto de treinamento
  - y_{pk}: é a saída desejada
  - o_{pk}: é a saída obtida

Backpropagation

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 -) e 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

Backpropagation



Backpropagation

- Os pesos da camada de saída ($W°_{kj}$) são atualizados de acordo com:

- termos de erro da camada de saída:

$$\delta°_{pk} = (y_{pk} - o_{pk}) * f°'_k (net°_{pk})$$

- ajuste de pesos da camada de saída

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

Backpropagation

- Deixando em uma única fórmula:

$$W^{o}_{kj}(t+1) = W^{o}_{kj}(t) + \eta\,(y_{pk} - o_{pk}) * f^{o\prime}_{k}(net^{o}_{pk}) * i_{pj}$$

y_{pk} : é a saída esperada para o padrão p

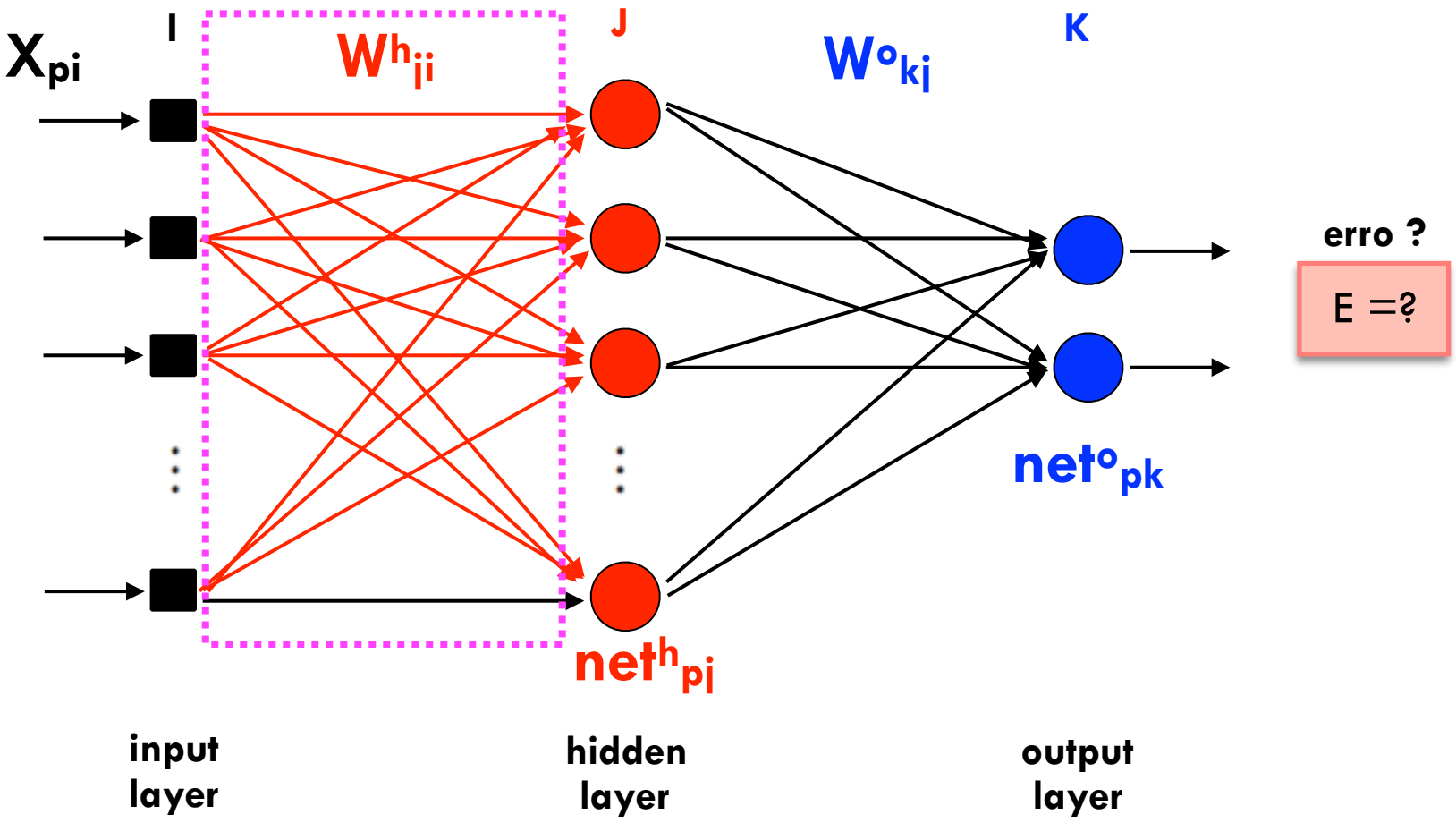
o_{pk} : é a saída obtida para o padrão p

i_{pj} : é a saída de ativação da unidade j da cama oculta

$f^{o\prime}_{k}$: é a derivada da função de ativação

η : é uma taxa de aprendizado, positiva e menor que 1

Backpropagation



Backpropagation

- Os pesos da camada de saída (W^h_{ji}) são atualizados de acordo com:

- termos de erro da camada oculta:

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

- ajuste de pesos da camada oculta

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

Backpropagation

- Por fim, calcula-se o erro total da rede para todos os exemplos de entrada (M):

$$Ep = 0.5 * \sum^{M}_{k} (y_{pk} - o_{pk})^2$$

Roteiro

1. Introdução
2. *Backpropagation*
3. **Algoritmo Resumido**
4. Exemplo / Exercício
5. Síntese / Próximas Aulas
6. Referências

Algoritmo resumido

1. Aplicar o valor de entrada Xp na rede

2. Calcular os valores de entrada na camada oculta (net^h_{pj})

$$\mathbf{net}^h_{pj} = \sum_{i=1}^{N} w^h_{ji} x_{pi} + \theta^h_j$$

3. Calcular as saídas da camada oculta (i_{pj})

$$i_{pj} = f^h_j(net^h_{pj})$$

Algoritmo resumido

4. Avance para a camada de saída. Calcular os valores de entrada para cada unidade de saída (net^{o}_{pk})

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

5. Calcular as saídas da camada de saída (o_{pk})

$$o_{pk} = f^{ok}(net^{h}_{pk})$$

Algoritmo resumido

6. Calcular os termos de erro para as unidades de saída:

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

7. Calcular os termos de erro para as unidades ocultas:

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

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

Algoritmo resumido

8. Atualizar os pesos da camada de saída

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

9. Atualizar os pesos da camada oculta

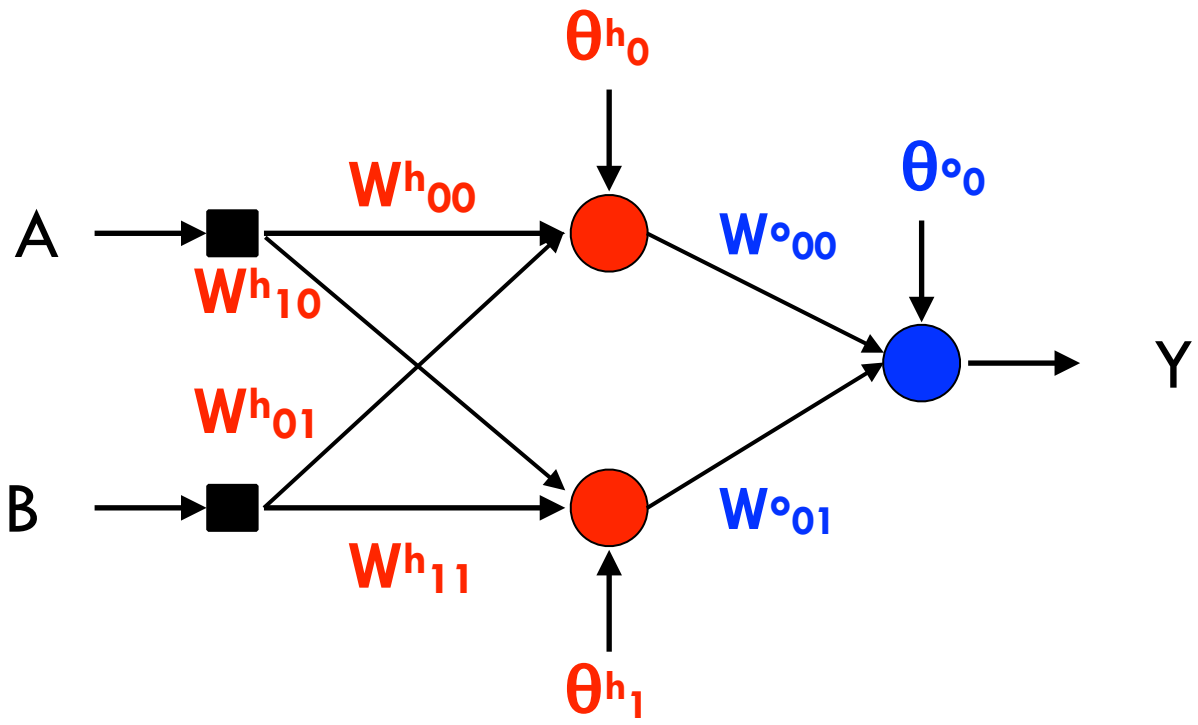
$$W^{h}_{ji}(t+1) = W^{h}_{ji}(t) + \eta * \delta^{h}_{pj} * x_{pi}$$

10. Calcular o erro total da época
    a. Indica o qüão bem a rende está aprendendo.
    b. quando for menor que um limiar, parar

Roteiro

1. Introdução
2. *Backpropagation*
3. Algoritmo Resumido
4. **Exemplo / Exercício**
5. Síntese / Próximas Aulas
6. Referências

Exemplo



XOR dataset

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

Exemplo

epoch	$\theta_h 0$	$\theta_h 1$	$\theta_o 0$	$W_h 00$	$W_h 10$	$W_h 01$	$W_h 11$	$W_o 00$	$W_o 01$
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

1. Introdução
2. *Backpropagation*
3. Algoritmo Resumido
4. Exemplo / Exercício
5. Síntese / Próximas Aulas
6. Referências

Próxima Aula

- MLPs
  - combinação de hiperplanos / regiões convexas
  - Backpropagation
    - forward → propaga sinal
    - backward → propaga o erro
  - exemplo (XOR)

Próxima Aula

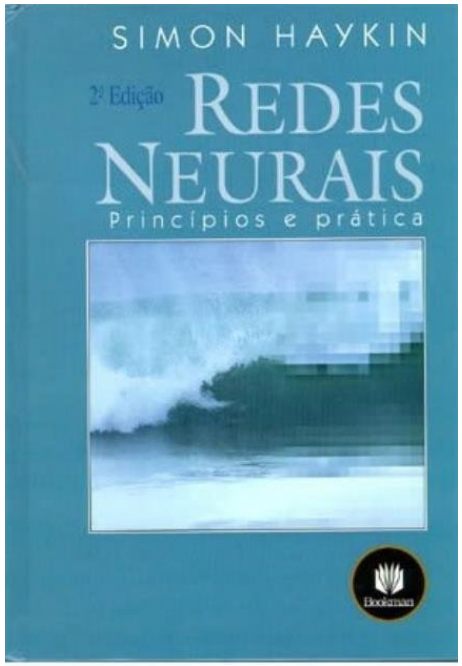
- Implementação:
  - MLP
  - *Backpropagation*

- Redes RBF

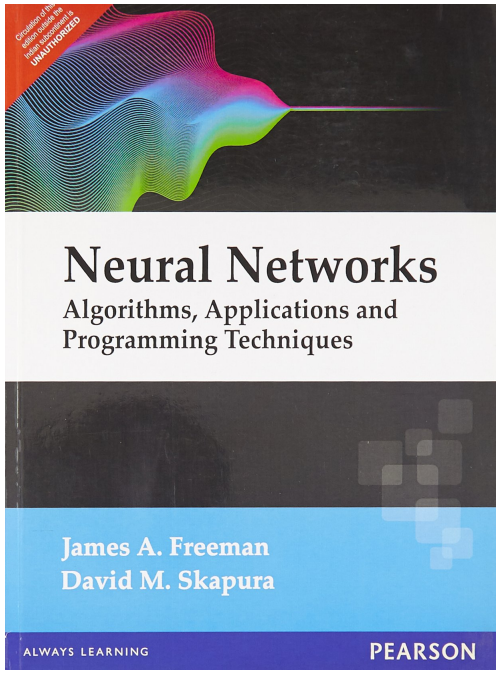
Roteiro

1. Introdução
2. *Backpropagation*
3. Algoritmo Resumido
4. Exemplo / Exercício
5. Síntese / Próximas Aulas
6. Referências

Literatura Sugerida



(Haykin, 1999)



(Freeman & Skapura, 1991)

Perguntas?

Prof. Rafael G. Mantovani

rgmantovani@uel.br

Exercício

epoch	$\theta_h 0$	$\theta_h 1$	$\theta_o 0$	$W_h 00$	$W_h 10$	$W_h 01$	$W_h 11$	$W_o 00$	$W_o 01$
0	0.05	0.06	0.07	0.2	0.15	0.35	0.18	0.10	0.12
1									
2									

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