

FUNDAMENTOS DA INTELIGÊNCIA ARTIFICIAL

Aula 04 - Perceptron Multicamadas (*Multilayer Perceptrons - MLPs*)

Prof. Rafael G. Mantovani



Roteiro

- 1** Introdução
- 2** Multilayer Perceptron
- 3** Exemplo
- 4** Formalização / Treinamento
- 5** Função de Ativação
- 6** Síntese / Próximas Aulas
- 7** Referências

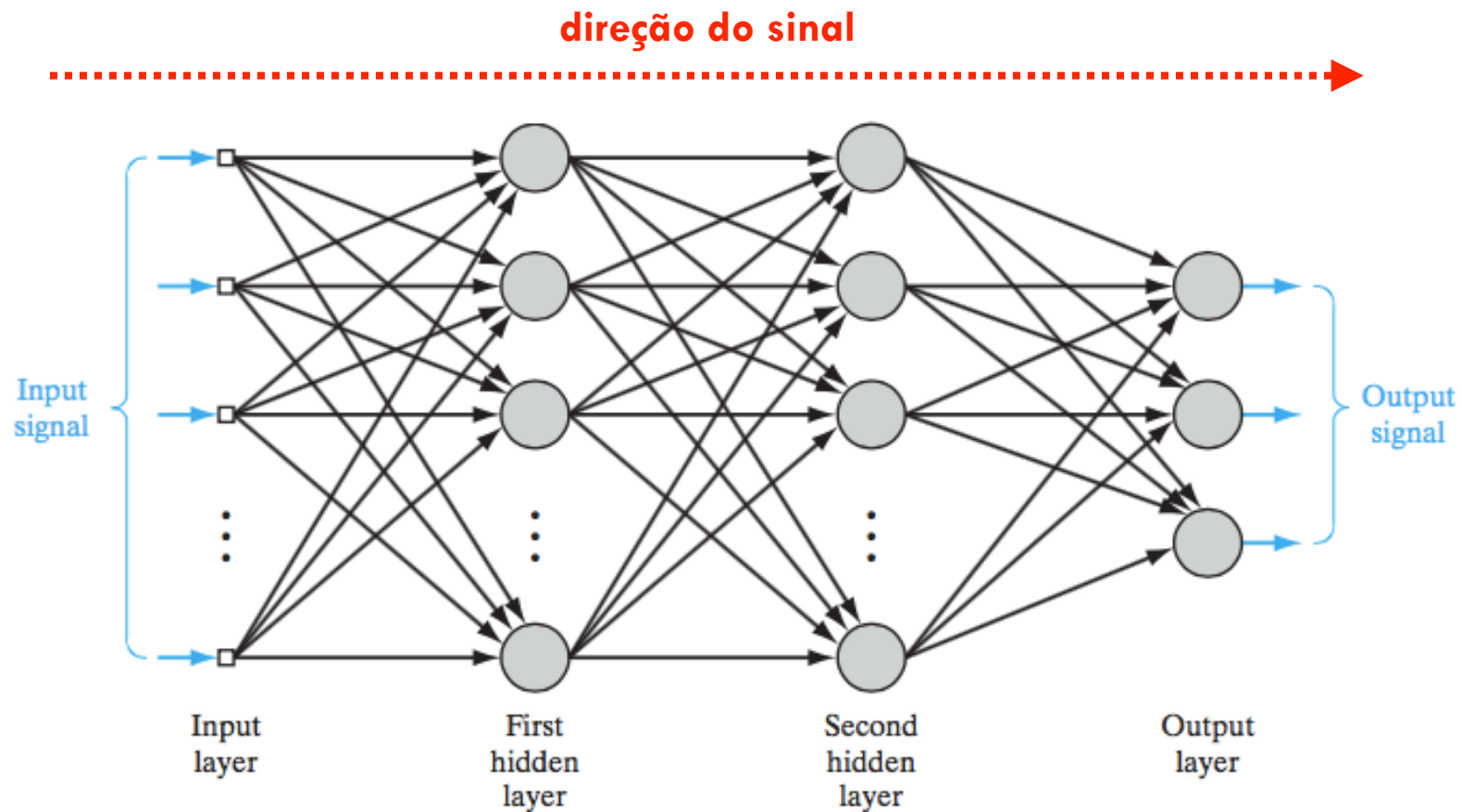
Roteiro

- 1 Introdução**
- 2 Multilayer Perceptron**
- 3 Exemplo**
- 4 Formalização / Treinamento**
- 5 Função de Ativação**
- 6 Síntese / Próximas Aulas**
- 7 Referências**

Introdução

- **Multilayer Perceptron:**
 - Supera as limitações práticas do Perceptron
 - neurônio função de ativação → não-linear e diferenciável
 - contém uma ou mais camadas escondidas
 - a rede possui alto grau de conectividade

Introdução



Introdução

□ Deficiências:

- análise teórica é difícil → há muitas conexões e funções não lineares
- muitos neurônios → difícil de visualizar o processo de aprendizado
- Aprendizado → difícil: há um espaço muito maior de funções. Há mais representações dos padrões de entrada

Roteiro

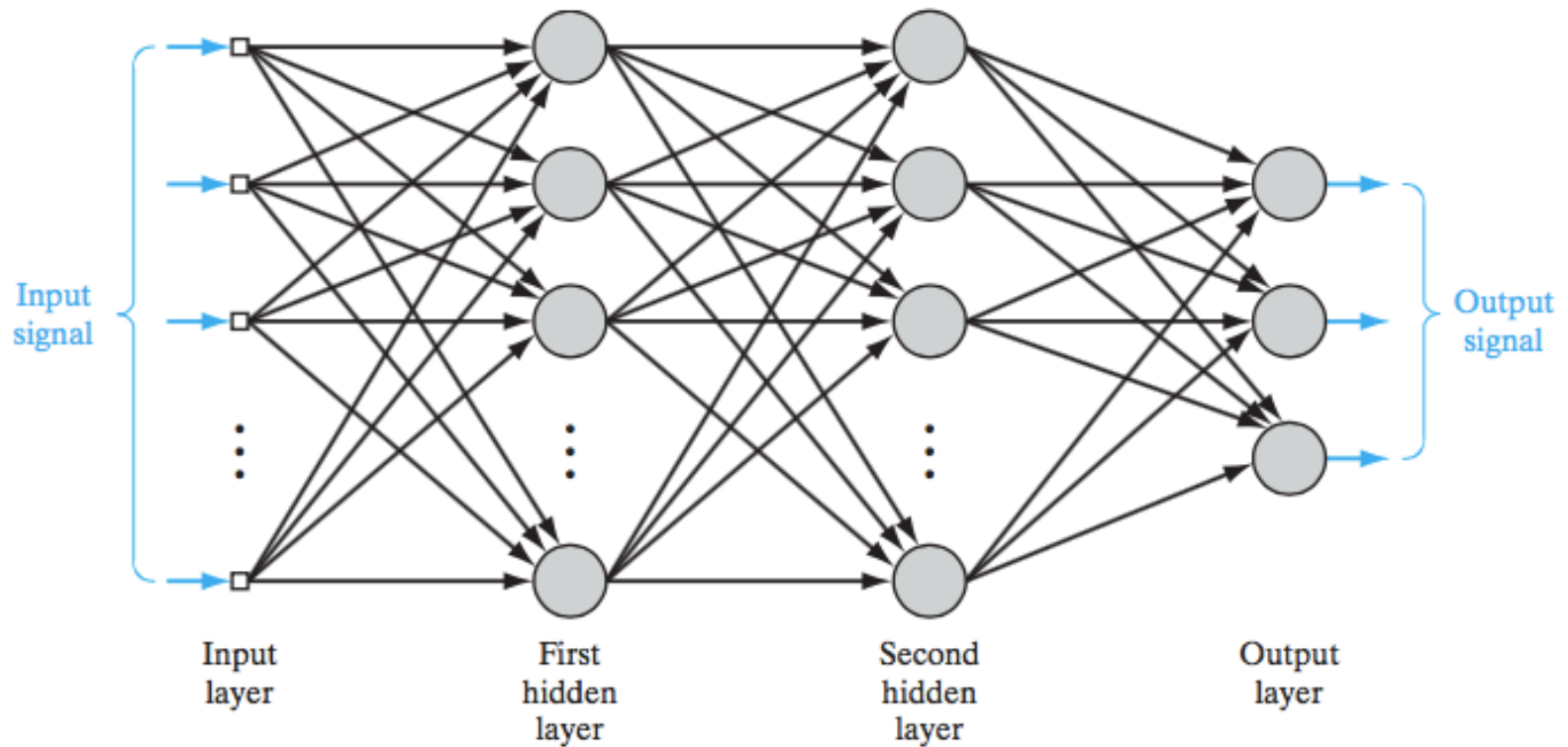
- 1 Introdução
- 2 Multilayer Perceptron
- 3 Exemplo
- 4 Formalização / Treinamento
- 5 Função de Ativação
- 6 Síntese / Próximas Aulas
- 7 Referências

Multilayer Perceptron

- Como aprender?

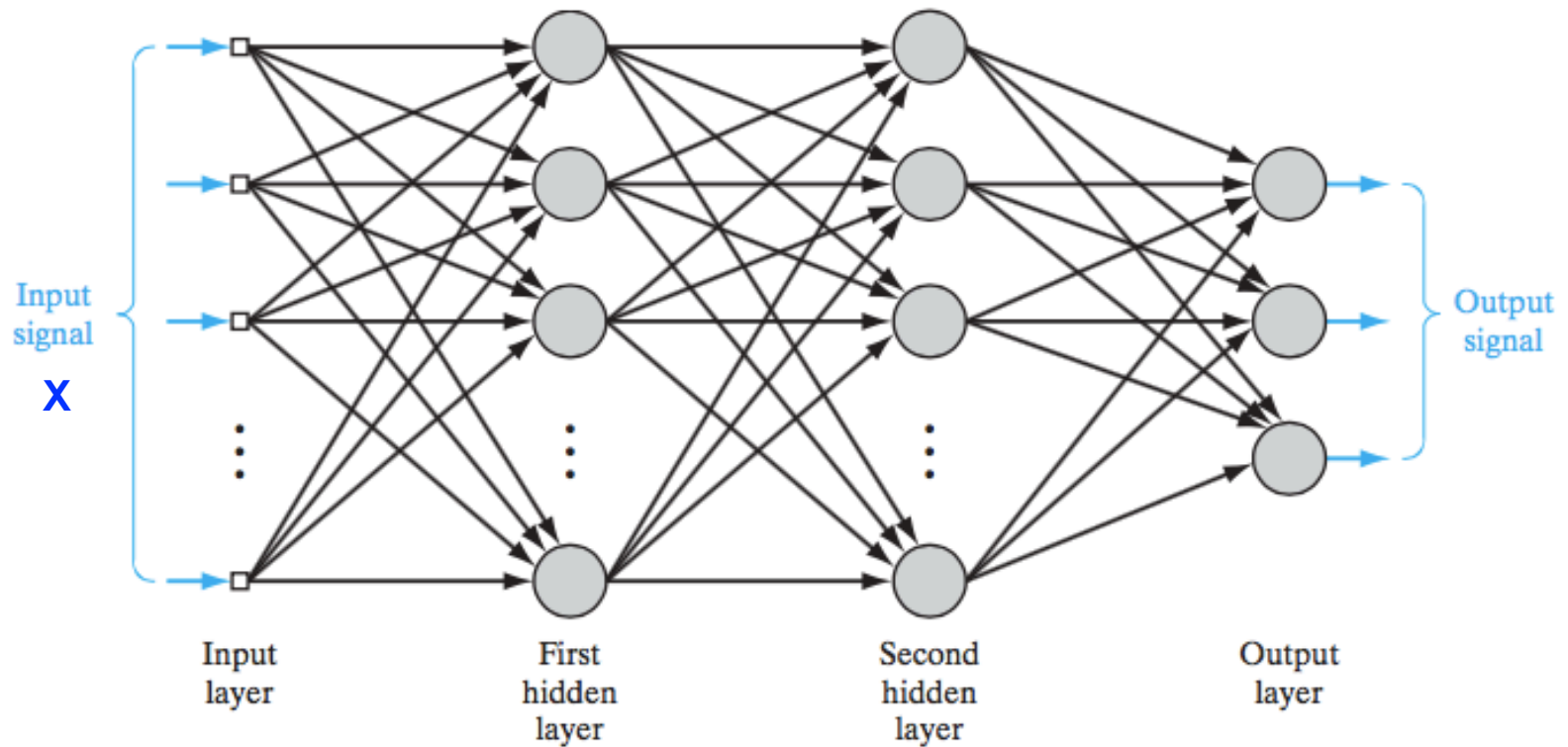
Multilayer Perceptron

- **Backpropagation**



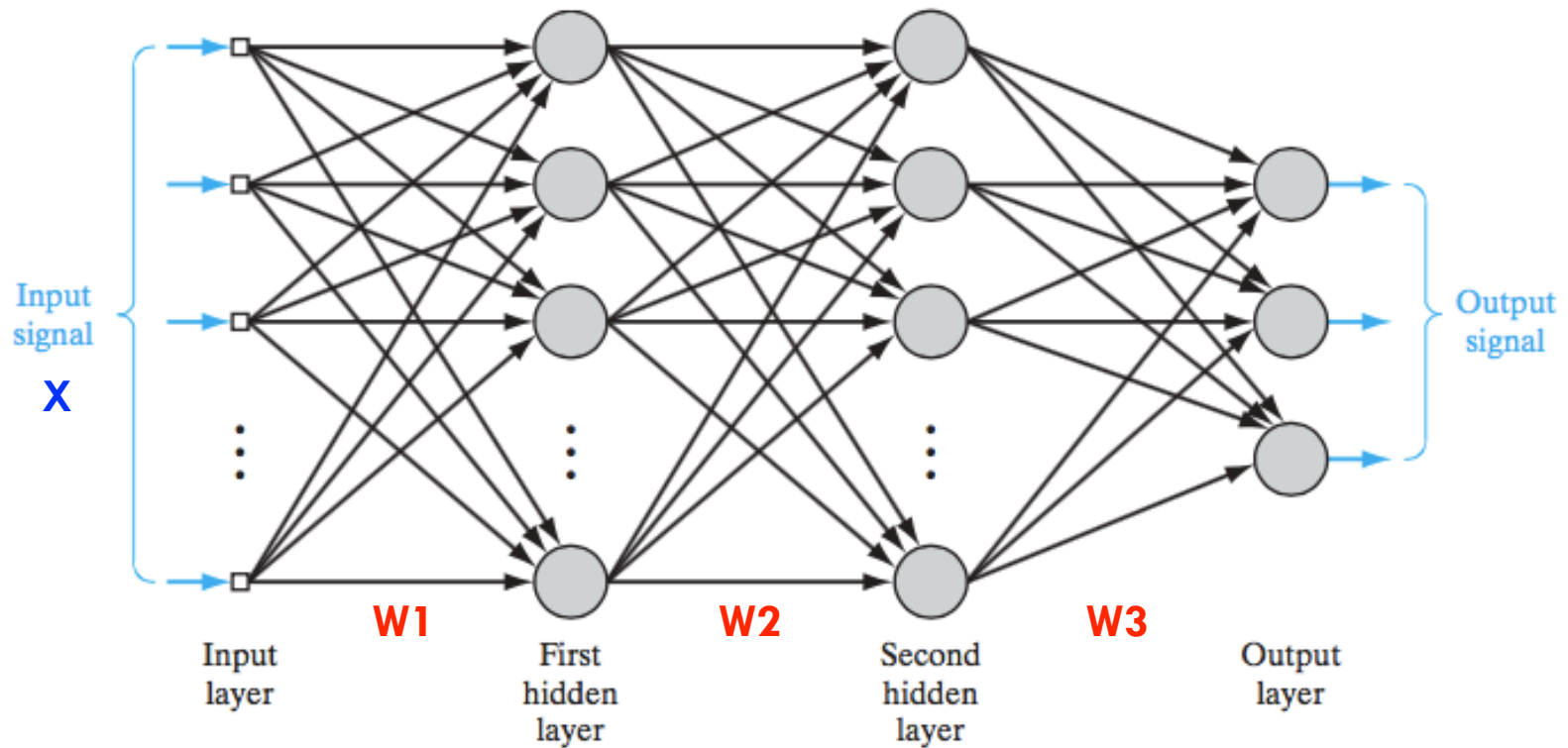
Multilayer Perceptron

- **Backpropagation**



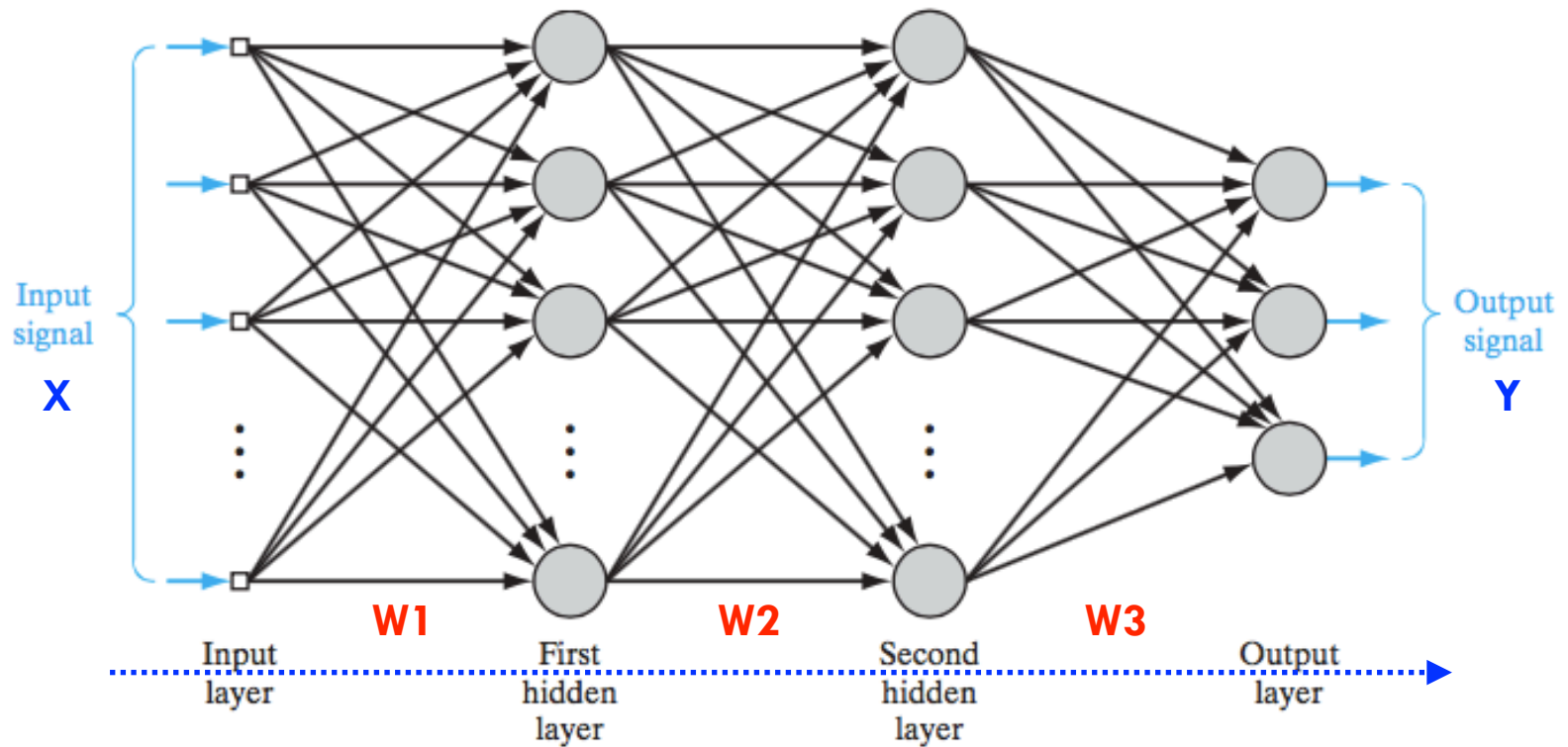
Multilayer Perceptron

- **Backpropagation**



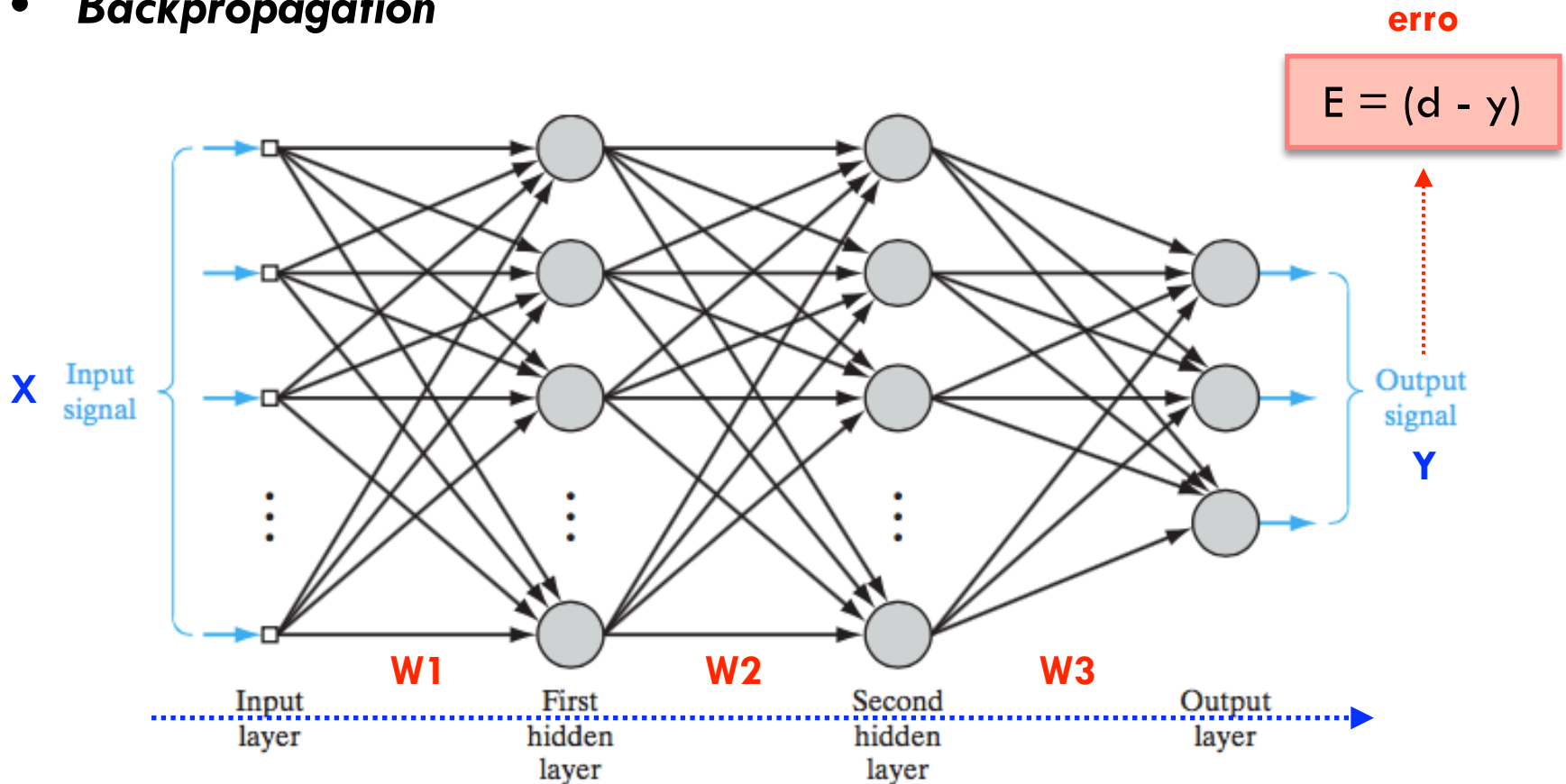
Multilayer Perceptron

- Backpropagation**



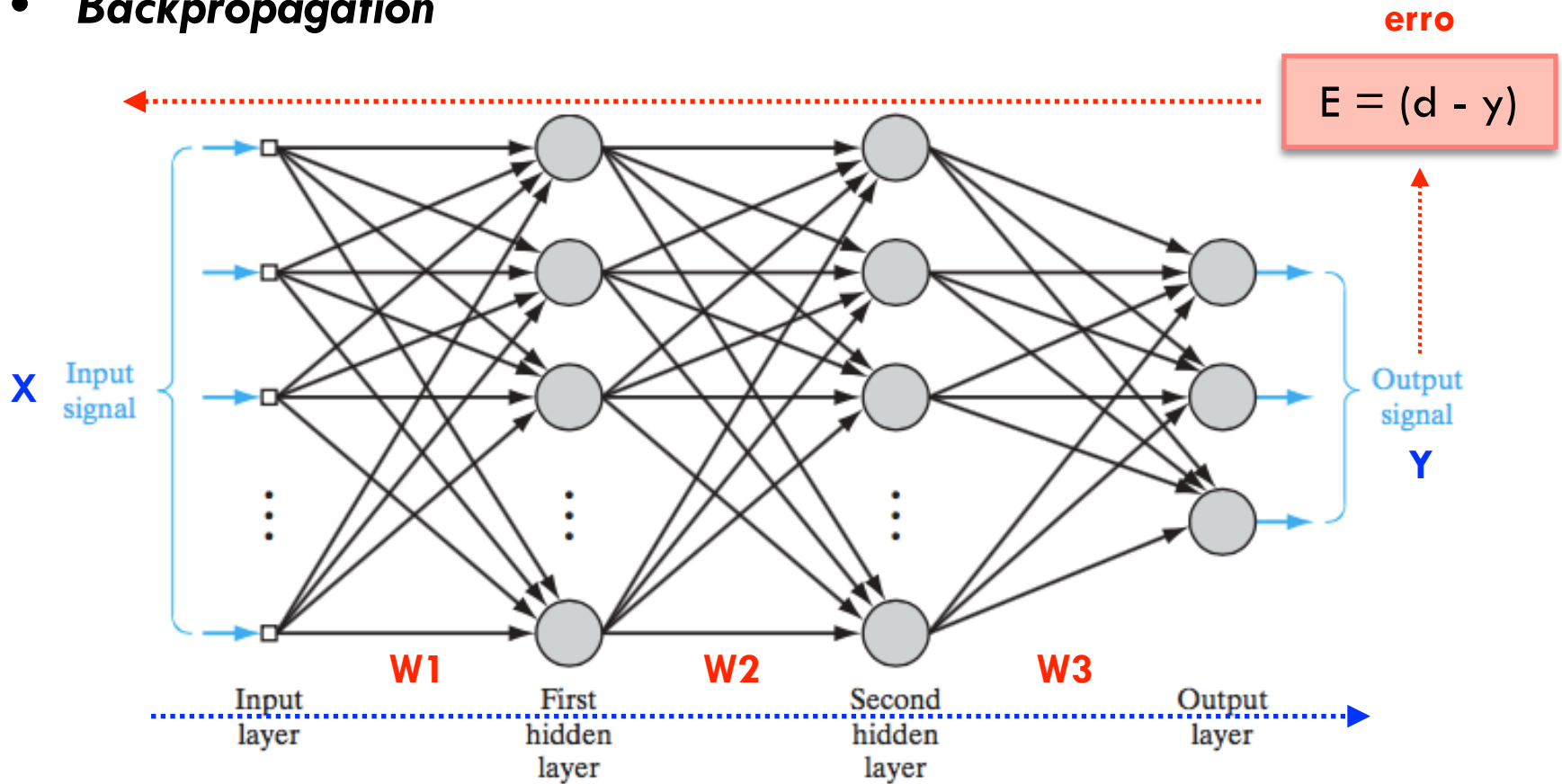
Multilayer Perceptron

- Backpropagation**



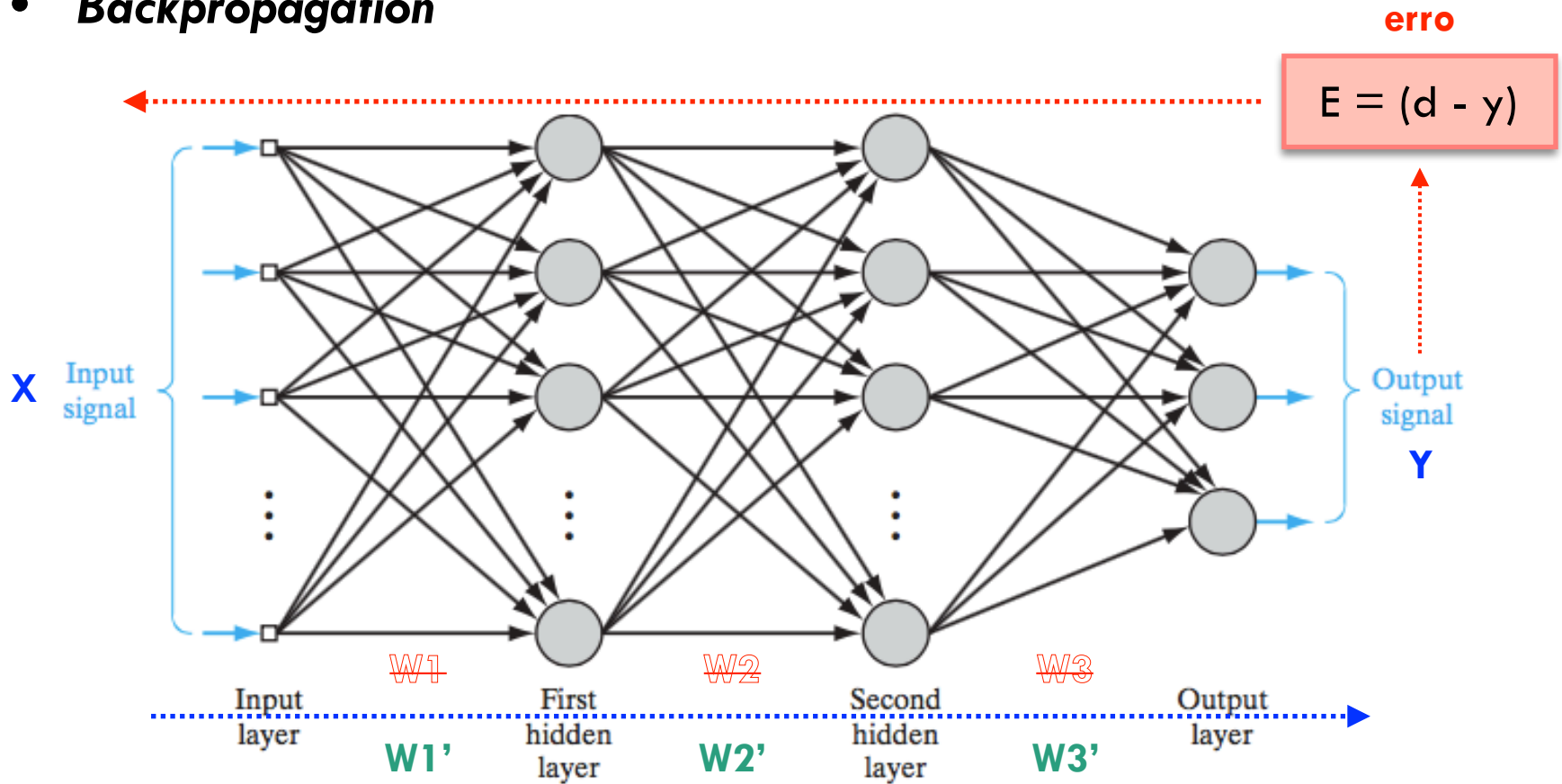
Multilayer Perceptron

- Backpropagation**



Multilayer Perceptron

- Backpropagation**



Multilayer Perceptron

□ Fases:

- Forward: propagação do sinal

Entrada → Camadas Ocultas → Saída

- Backward: sinal de erro é retropropagado

Entrada ← Camadas Ocultas ← Saída

Ajustes sinápticos

Multilayer Perceptron

- Sinais de função e sinais de erro

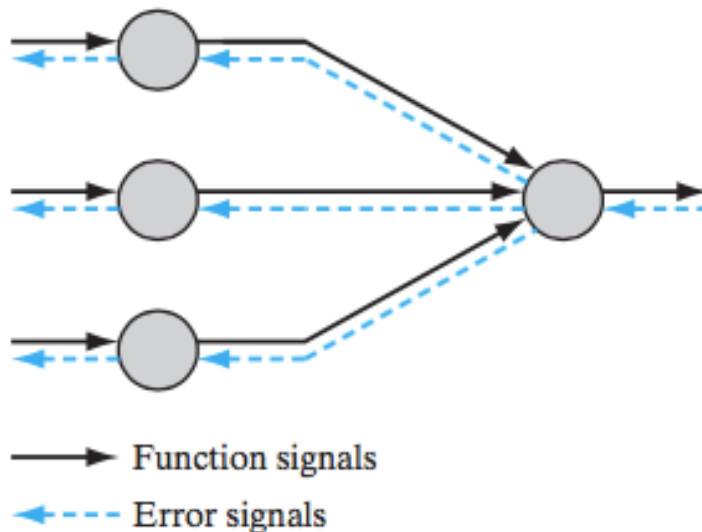
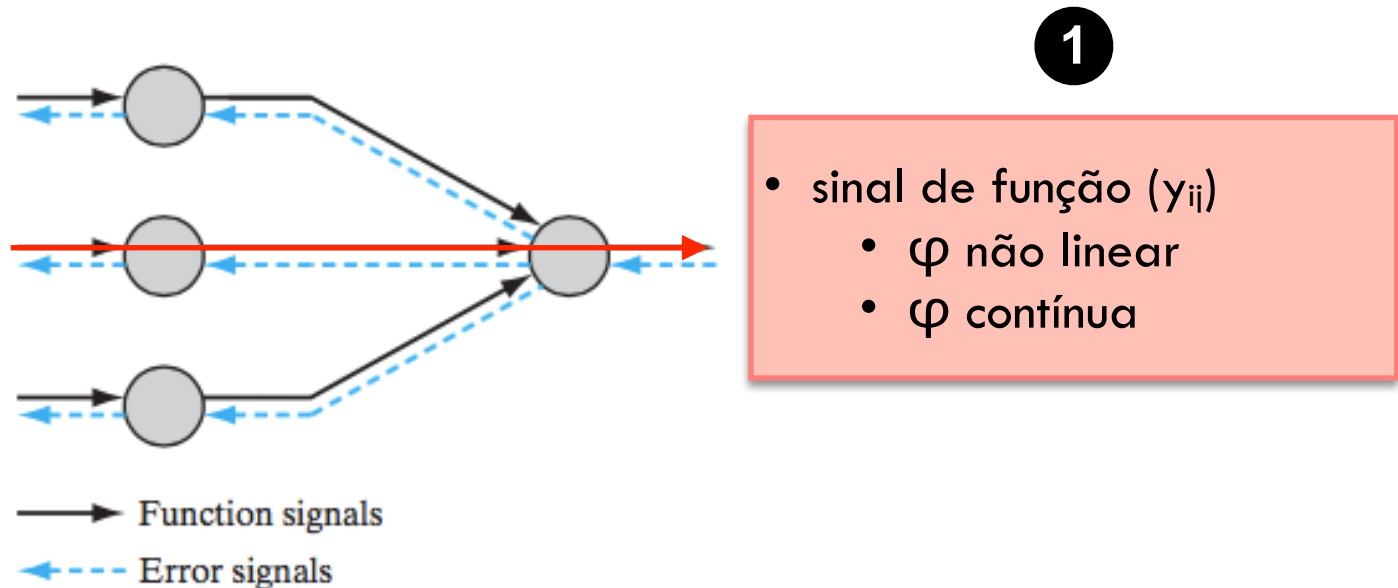


FIGURE 4.2 Illustration of the directions of two basic signal flows in a multilayer perceptron: forward propagation of function signals and back propagation of error signals.

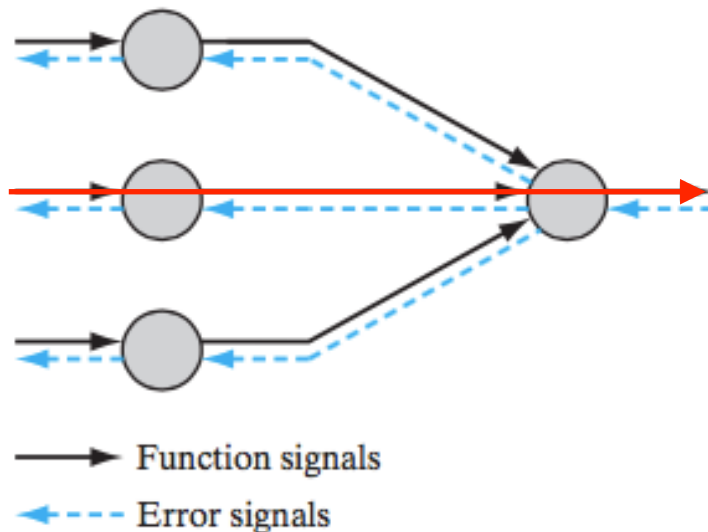
Multilayer Perceptron

- Neurônios da camada oculta, executam dois cálculos:



Multilayer Perceptron

- Neurônios da camada oculta, executam dois cálculos:

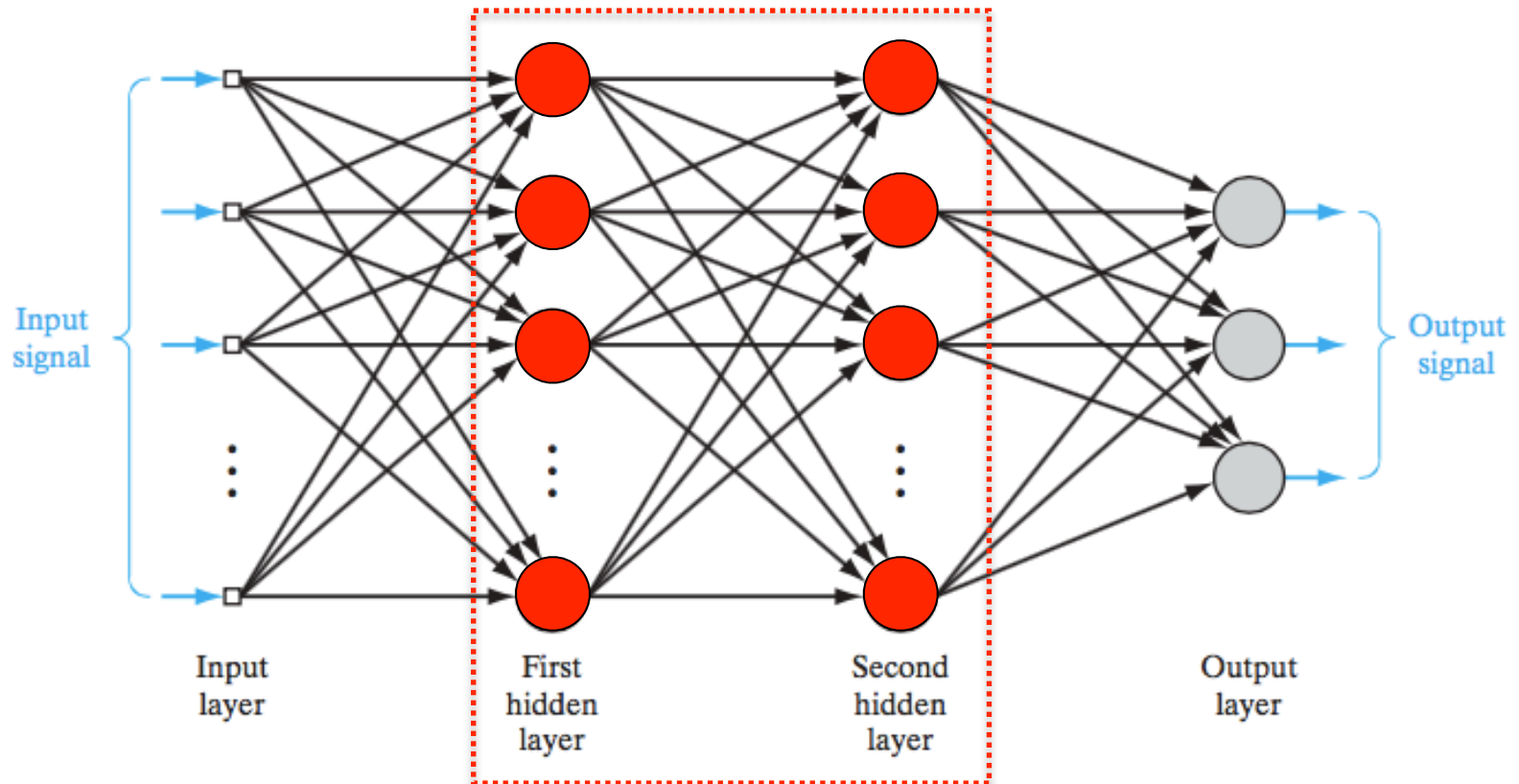


2

- estimativa do vetor gradiente (δ_{ij})
 - gradiente da superfície de erro
 - retropropagação

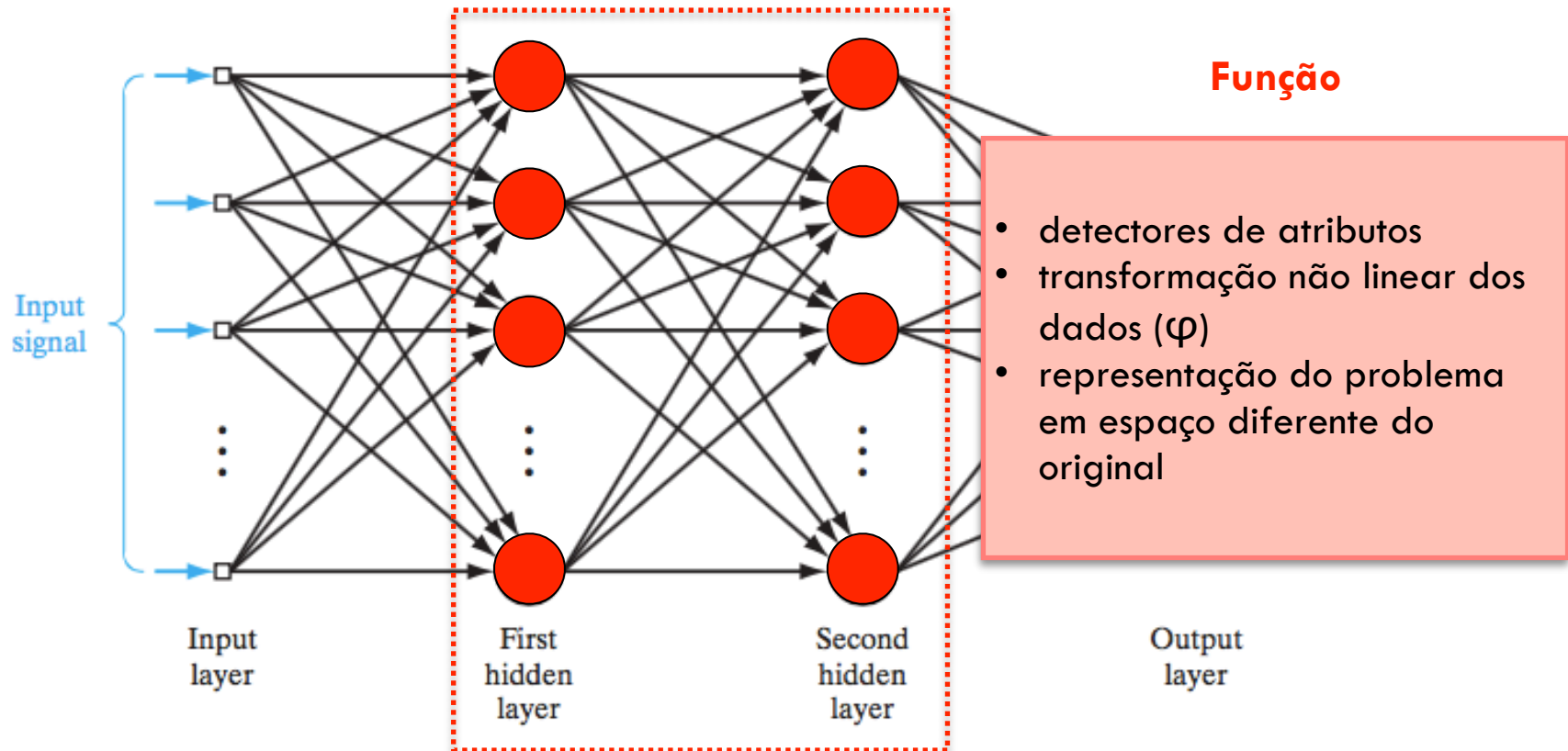
Multilayer Perceptron

- Neurônios das camadas ocultas



Multilayer Perceptron

- Neurônios das camadas ocultas



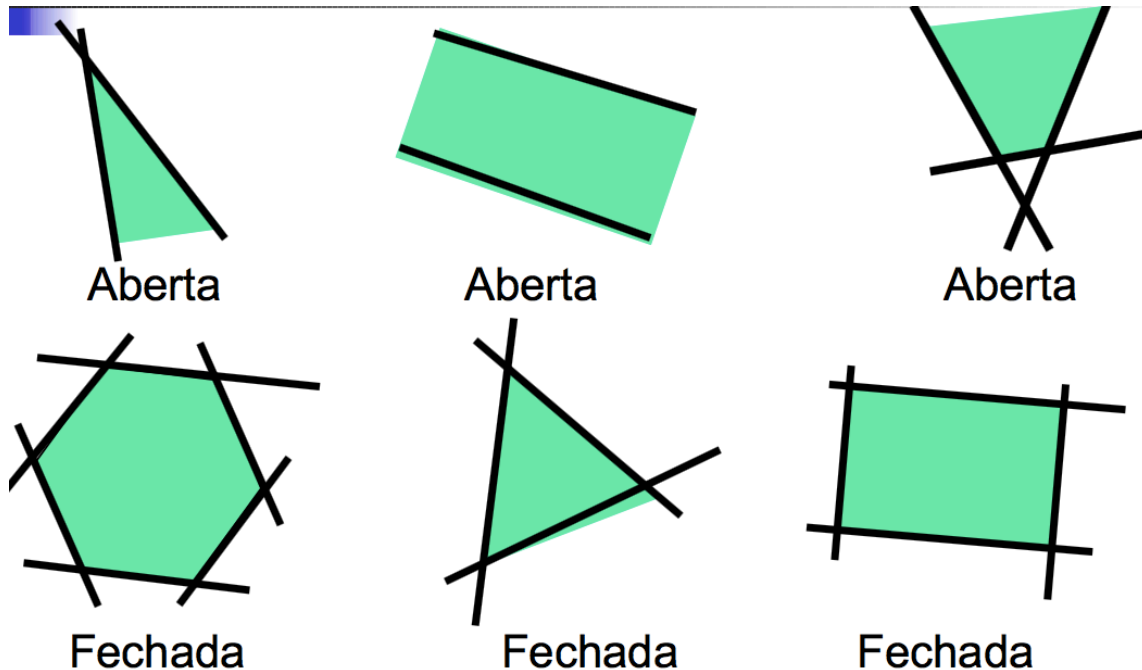
Multilayer Perceptron

□ **Camadas intermediárias:**

- Primeira: linhas retas no espaço de decisão
- Segunda: combina as linhas da camada anterior em regiões convexas
- Terceira: combina as figuras convexas produzindo formatos abstratos

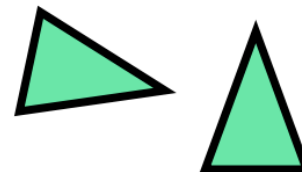
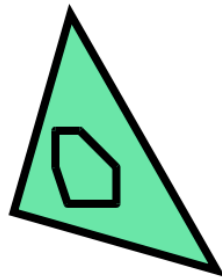
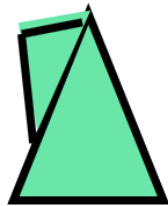
Regiões Convexas

□ Combinações de hiperplanos



Figuras Convexas

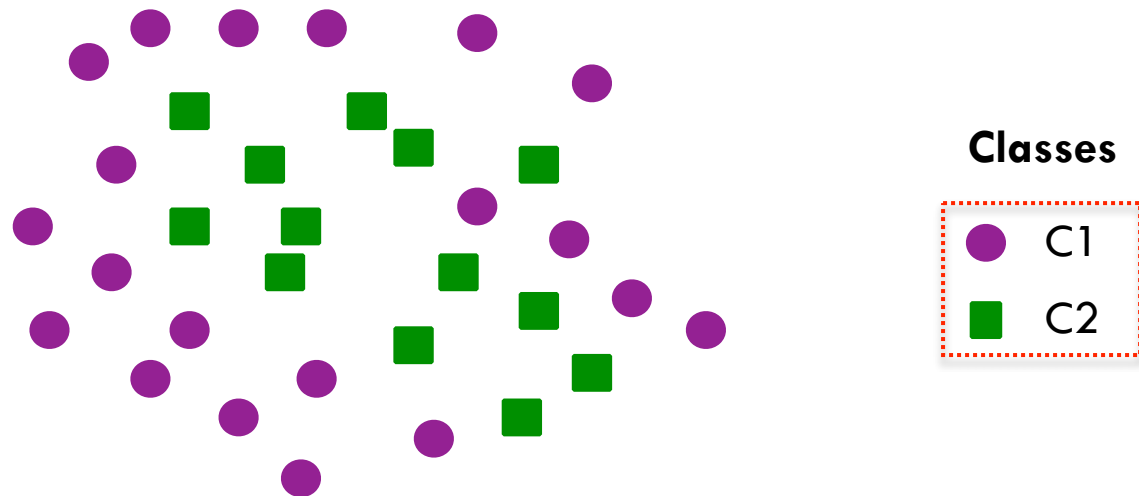
- **Combinações de regiões convexas**



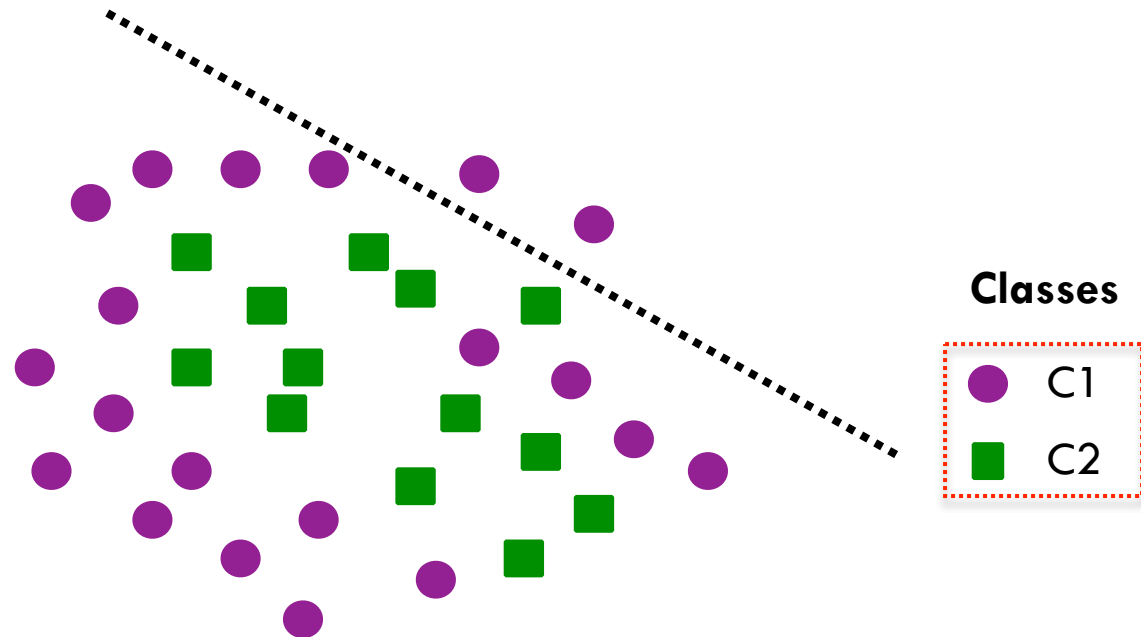
Roteiro

- 1 Introdução
- 2 Multilayer Perceptron
- 3 Exemplo
- 4 Formalização / Treinamento
- 5 Função de Ativação
- 6 Síntese / Próximas Aulas
- 7 Referências

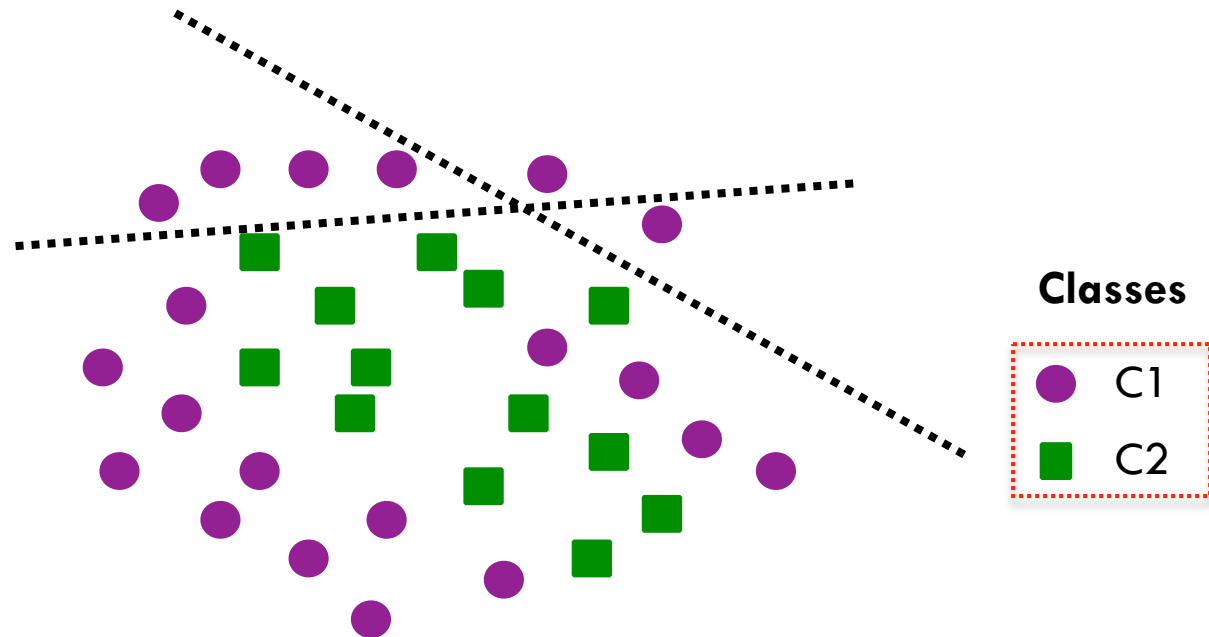
Exemplo



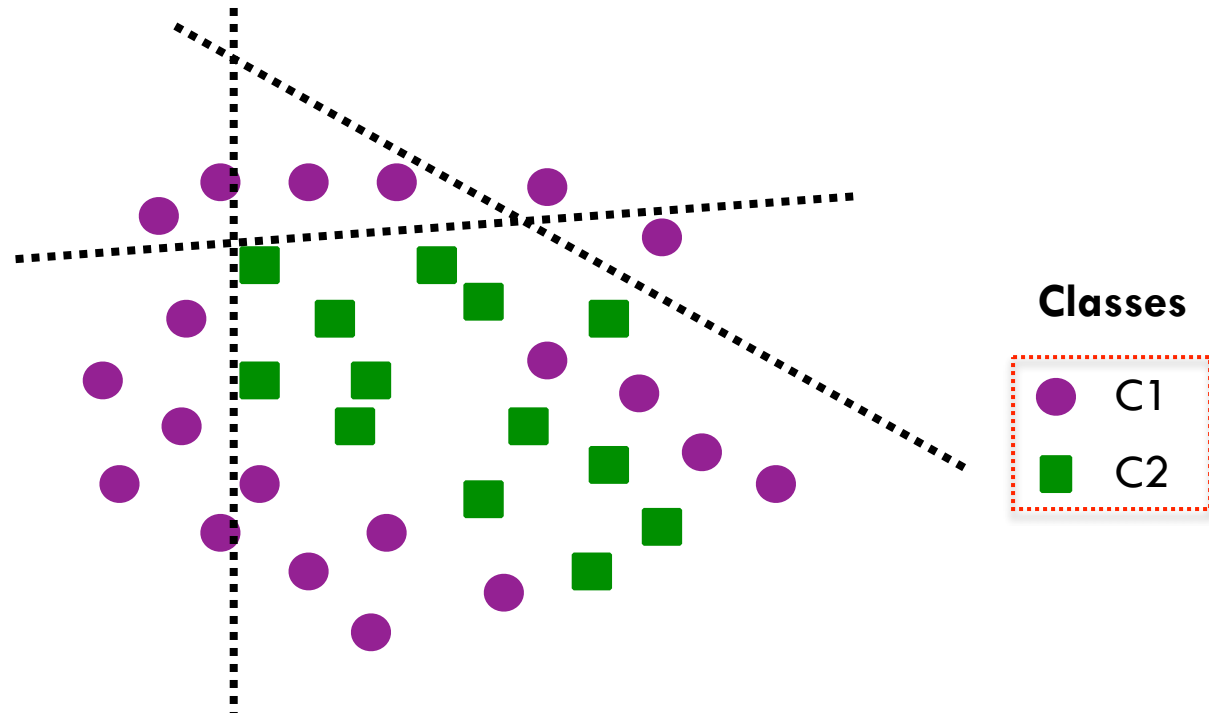
Exemplo



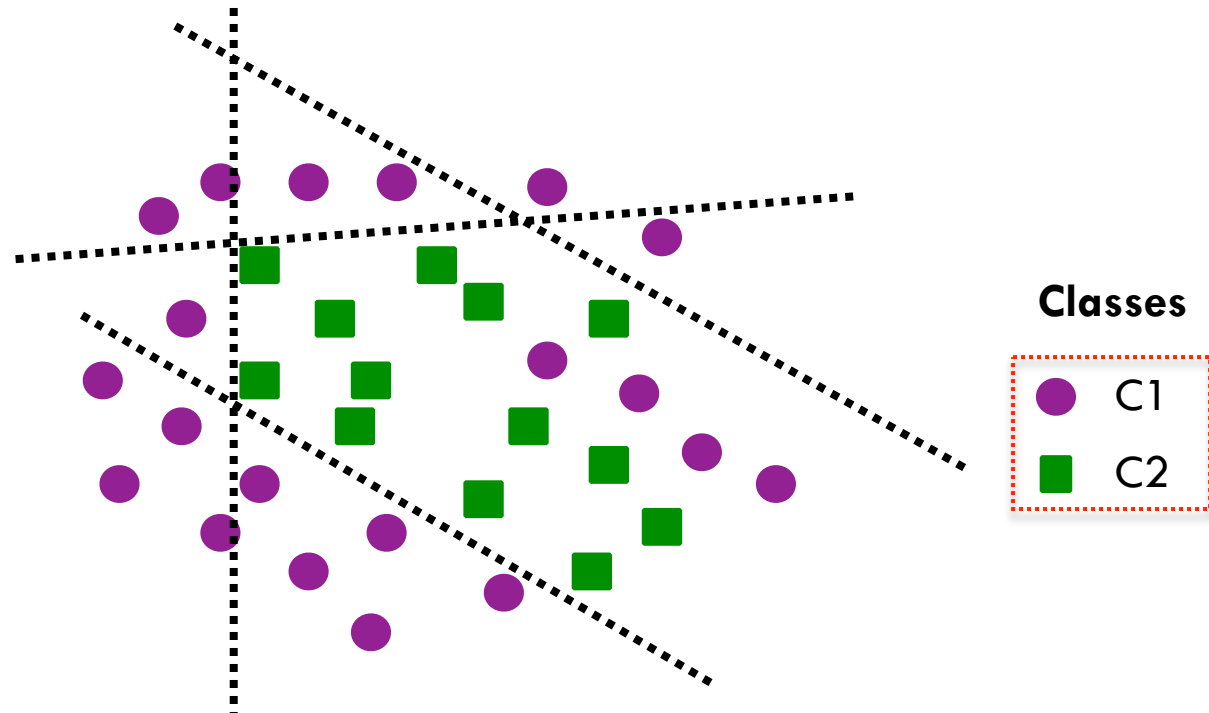
Exemplo



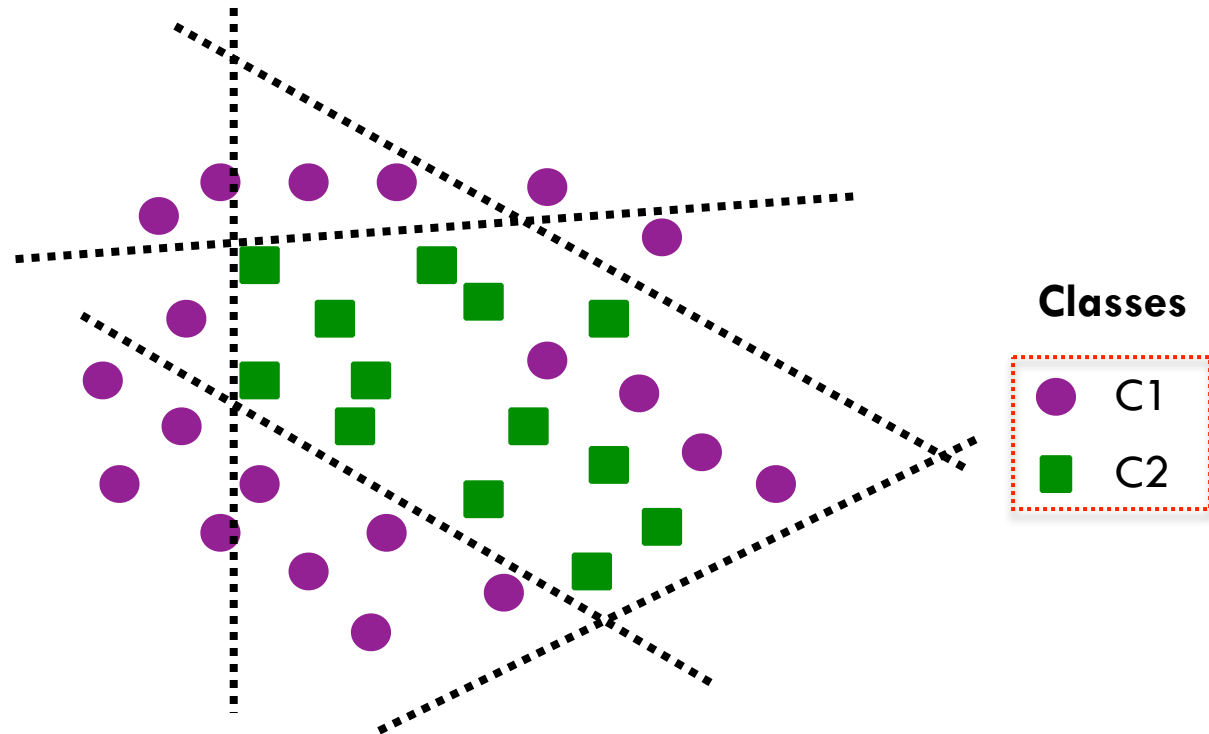
Exemplo



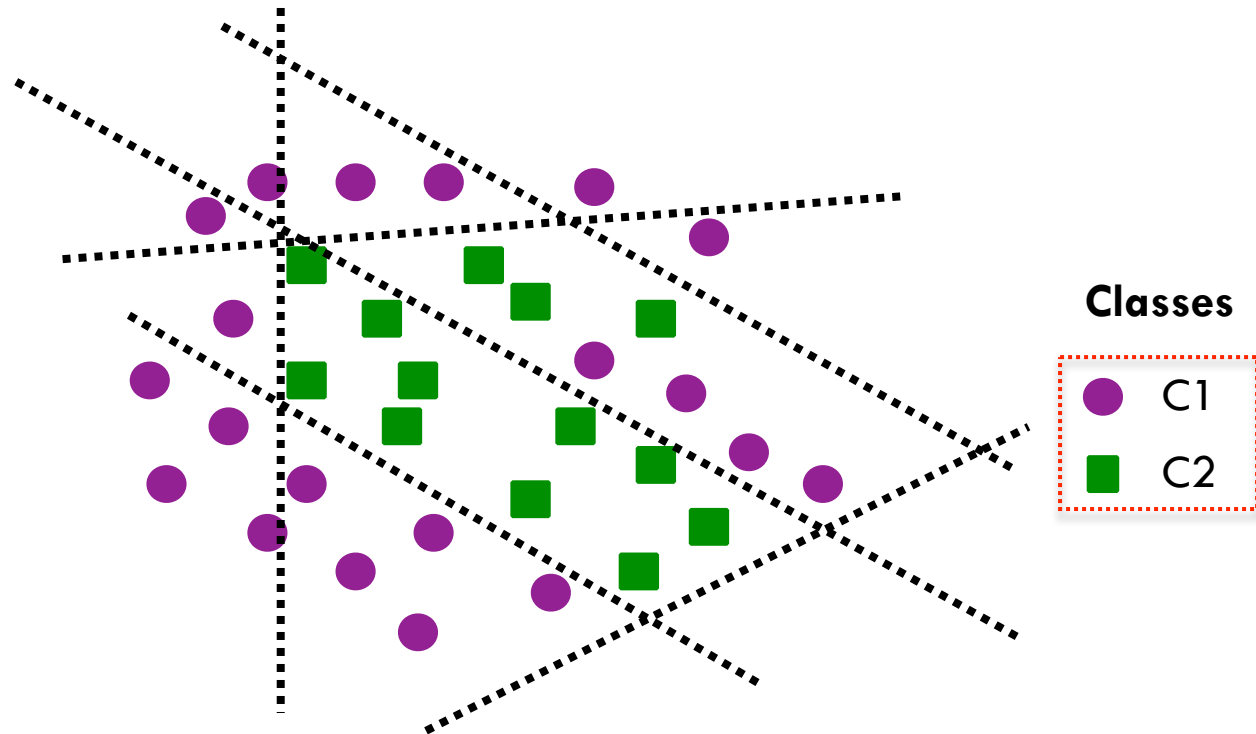
Exemplo



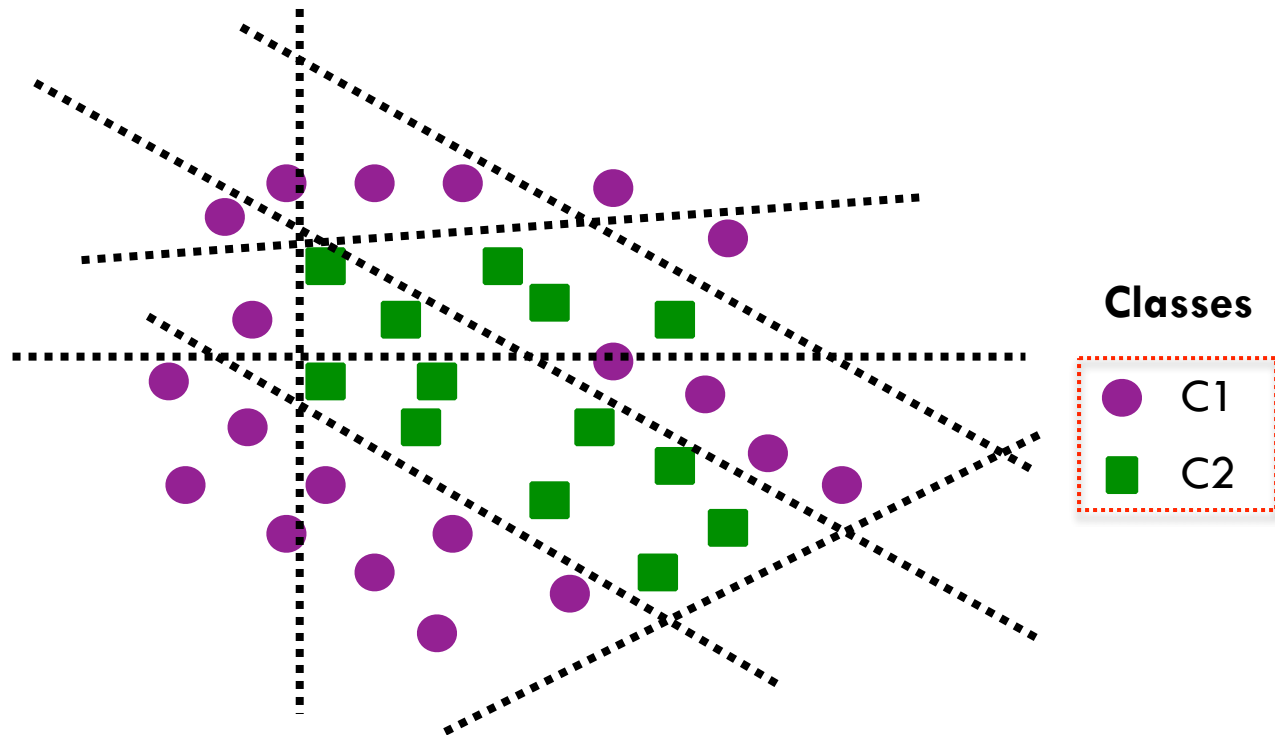
Exemplo



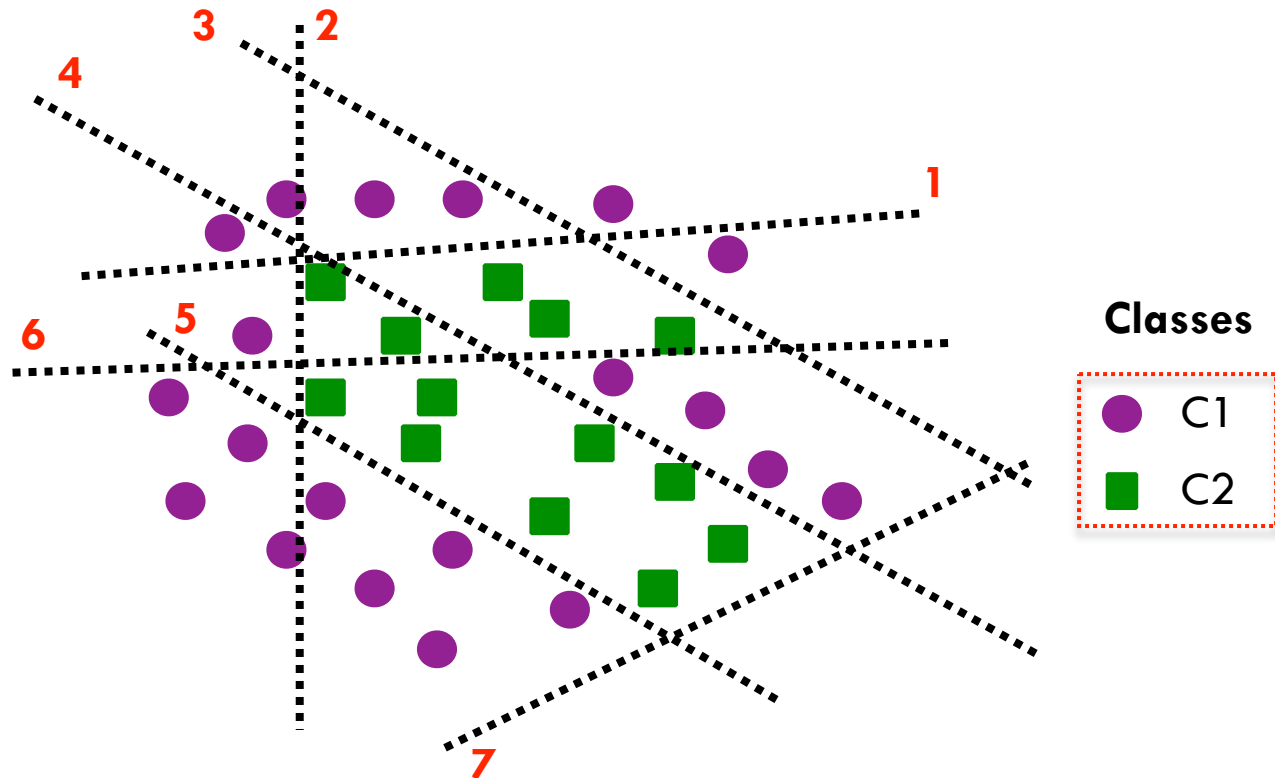
Exemplo



Exemplo

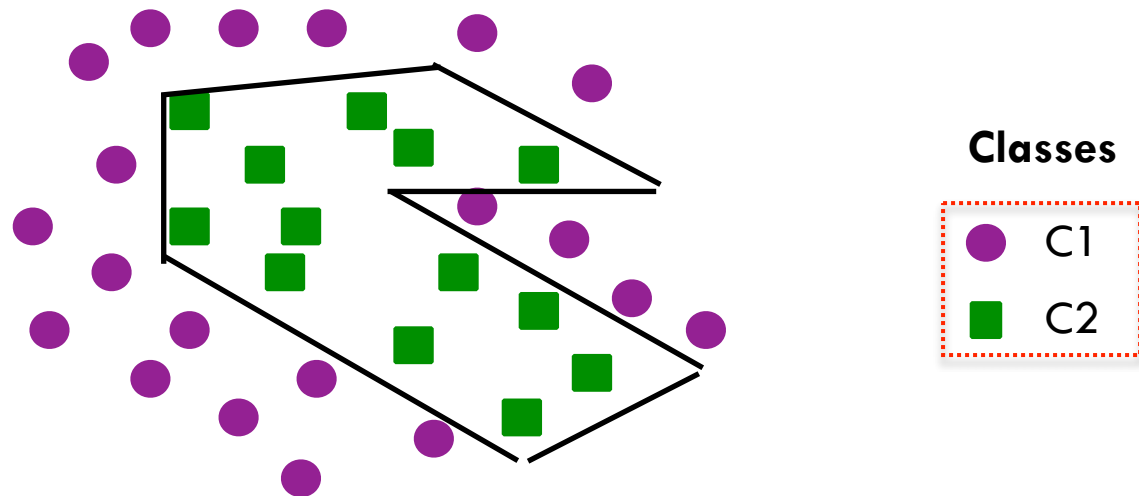


Exemplo



7 hiperplanos = 7 neurônios

Exemplo



7 hiperplanos = 1 região convexa

Roteiro

- 1 Introdução
- 2 Multilayer Perceptron
- 3 Exemplo
- 4 Formalização / Treinamento
- 5 Função de Ativação
- 6 Síntese / Próximas Aulas
- 7 Referências

Formalização

- **MLP:**

- $\tau = [x(n), d(n)] \rightarrow$ exemplo de treinamento
- $y_i(n) \rightarrow$ sinal produzido na saída do neurônio i na camada de saída, estimulado por $x(n)$, aplicado na camada de entrada
- sinal do erro produzido pelo neurônio i é:
 - $e_i(n) = d_i(n) - y_i(n)$

Formalização

- O erro instantâneo produzido no neurônio j é dado por:

$$\varepsilon_j(n) = \frac{1}{2} e_j^2(n)$$

- Somando os erros de todos os neurônios da camada de saída:

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

- C é o conjunto de neurônios de saída. Se houverem N exemplos de treinamento, o erro médio sobre todos os exemplos (risco empírico) é dado por:

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

Treinamento

- **Modo de treinamento batch:**
 - apresenta todos os exemplos → ajuste dos pesos
 - 1 ajuste p uma época completa
- Curva de aprendizado
 - ϵ_{av} x épocas
 - época, os exemplos são embaralhados/reordenados
- Vários experimentos, iniciando W com valores diferentes
 - Média do desempenho

Treinamento

- **Modo de treinamento batch:**
 - estimativa mais precisa do vetor de gradientes
 - derivada da função de custo ϵ_{av} em relação a W
 - mínimo local
 - Paralelização do processo de aprendizado
- Mais difícil de detectar mudanças pequenas nos dados
- Se há exemplos redundantes, não consegue identificar (pois ajusta os pesos para todos os exemplos)

Treinamento

- **Modo de treinamento online:**
 - ajuste de W após a apresentação de cada exemplo
- A busca no espaço de pesos multidimensional torna-se estocástica
 - método estocástico
- Menos suscetível a ficar preso em mínimos locais
- Quando há redundância, tira vantagem ao ajustar os pesos
- Detecta melhor pequenas mudanças nos dados de treinamento
- Simples de implementar / Bons resultados em problemas difíceis

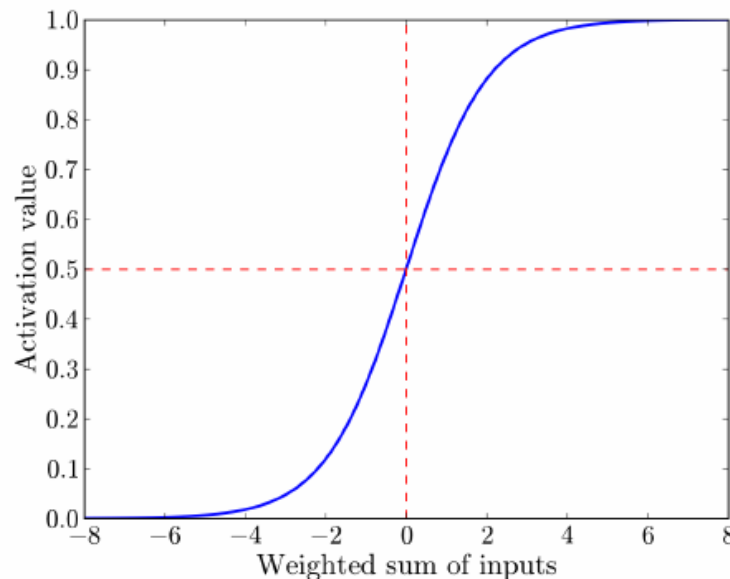
Roteiro

- 1 Introdução
- 2 Multilayer Perceptron
- 3 Exemplo
- 4 Formalização / Treinamento
- 5 Função de Ativação
- 6 Síntese / Próximas Aulas
- 7 Referências

Funções de Ativação

- **φ deve ser diferenciável:**
 - funções sigmoidal / logística

$$\varphi_j(v_j(n)) = \frac{1}{1 + \exp(-av_j(n))}, \quad a > 0$$

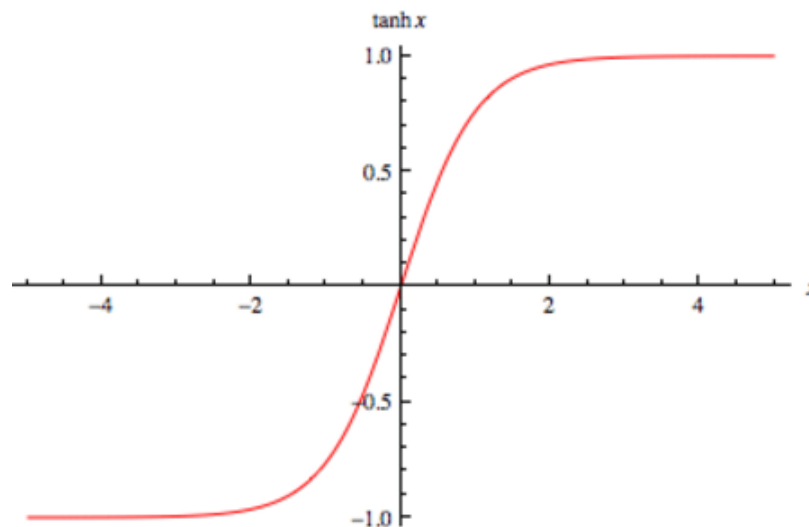


Funções de Ativação

- função tangente hiperbólica:

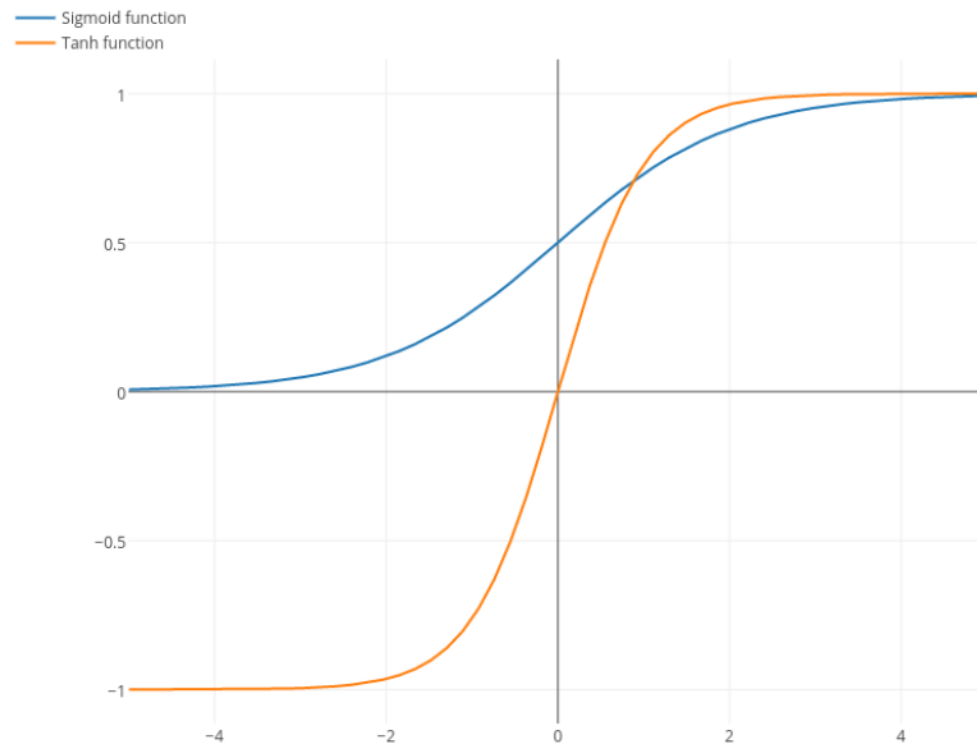
$$\varphi_j(v_j(n)) = a \tanh(bv_j(n))$$

- a e b são constantes positivas
- amplitude do sinal de saída: $-a \leq y_i \leq +a$



Funções de Ativação

□ tahn x sigmoid



Roteiro

- 1 Introdução
- 2 Multilayer Perceptron
- 3 Exemplo
- 4 Formalização / Treinamento
- 5 Função de Ativação
- 6 Síntese / Próximas Aulas
- 7 Referências

Síntese/Revisão

- MLP
 - perceptron multicamadas
 - neurônio j - sinais de ativação, sinais de erro
 - função de ativação **diferenciável**
- Gradiente Descendente
- Exemplo

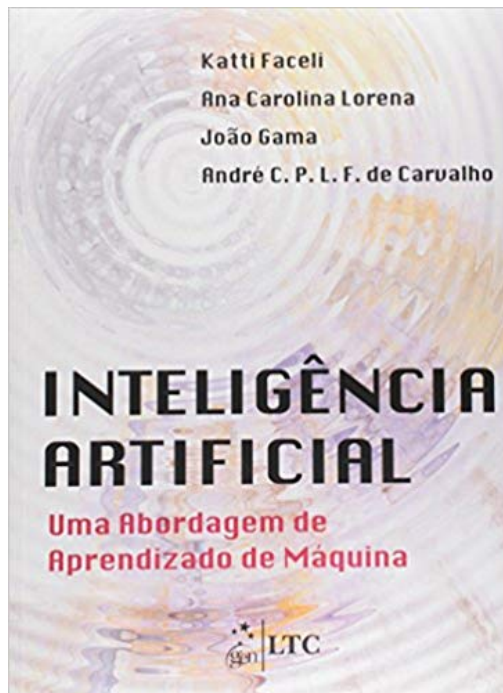
Próxima Aula

- Algoritmo Backpropagation
 - formalização para ajuste sináptico
 - exemplos

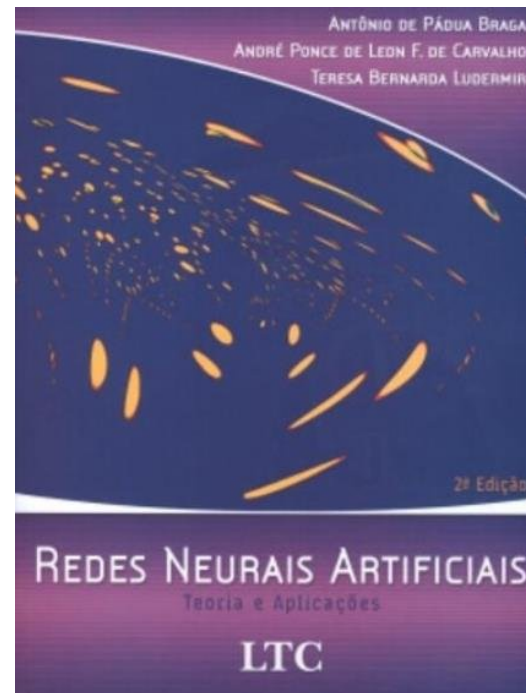
Roteiro

- 1 Introdução
- 2 Multilayer Perceptron
- 3 Exemplo
- 4 Formalização / Treinamento
- 5 Função de Ativação
- 6 Síntese / Próximas Aulas
- 7 Referências

Literatura Sugerida

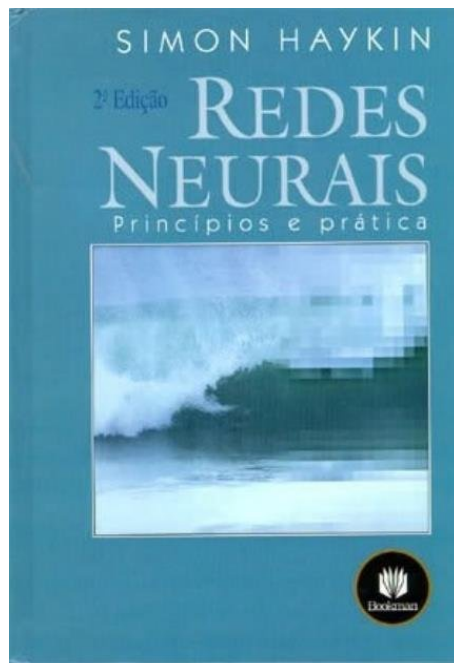


[Faceli et al, 2011]

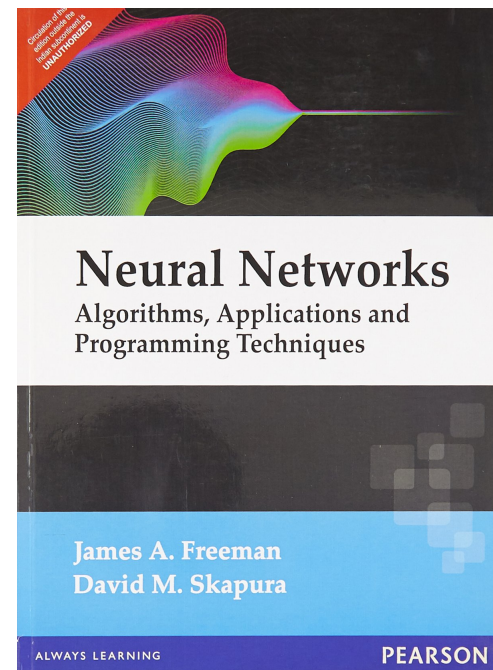


[Braga et al, 2007]

Literatura Sugerida



(Haykin, 1999)



(Freeman & Skapura, 1991)

Perguntas?

Prof. Rafael G. Mantovani

rgmantovani@gmail.com