

FUNDAMENTOS DA INTELIGÊNCIA ARTIFICIAL

Aula 03 - ADALINE

Prof. Rafael G. Mantovani



Roteiro

- 1** Introdução
- 2** ADALINE
- 3** ADALINE x Perceptron
- 4** Algoritmos de Treinamento ADALINE
- 5** Exemplo / Exercício
- 6** Síntese / Próximas Aulas
- 7** Referências

Roteiro

- 1 Introdução**
- 2 ADALINE**
- 3 ADALINE x Perceptron**
- 4 Algoritmos de Treinamento ADALINE**
- 5 Exemplo / Exercício**
- 6 Síntese / Próximas Aulas**
- 7 Referências**

Introdução

- **ADALINE** (Widrow & Hoff, 1960):
 - **ADA**ptive **LINE**ar neuron/element
 - aplicações voltadas a filtros lineares
 - algoritmo treinamento
 - magnitude e sinal do gradiente do erro
 - direção e valor do ajuste dos pesos W
 - Regra Delta → *Backpropagation* (MLP)

Roteiro

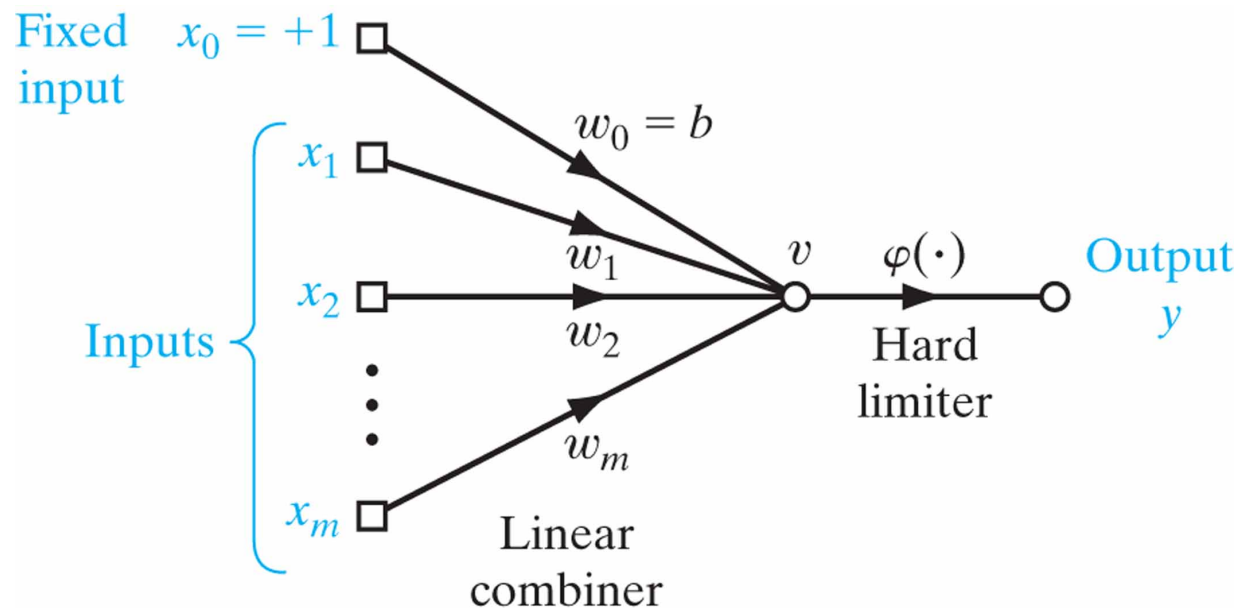
- 1 Introdução
- 2 **ADALINE**
- 3 ADALINE x Perceptron
- 4 Algoritmos de Treinamento ADALINE
- 5 Exemplo / Exercício
- 6 Síntese / Próximas Aulas
- 7 Referências

ADALINE

- ADALINE
 - rede de uma camada
 - neurônio de McCulloch-Pitts
 - bias
 - pesos sinápticos
 - **ativação linear**

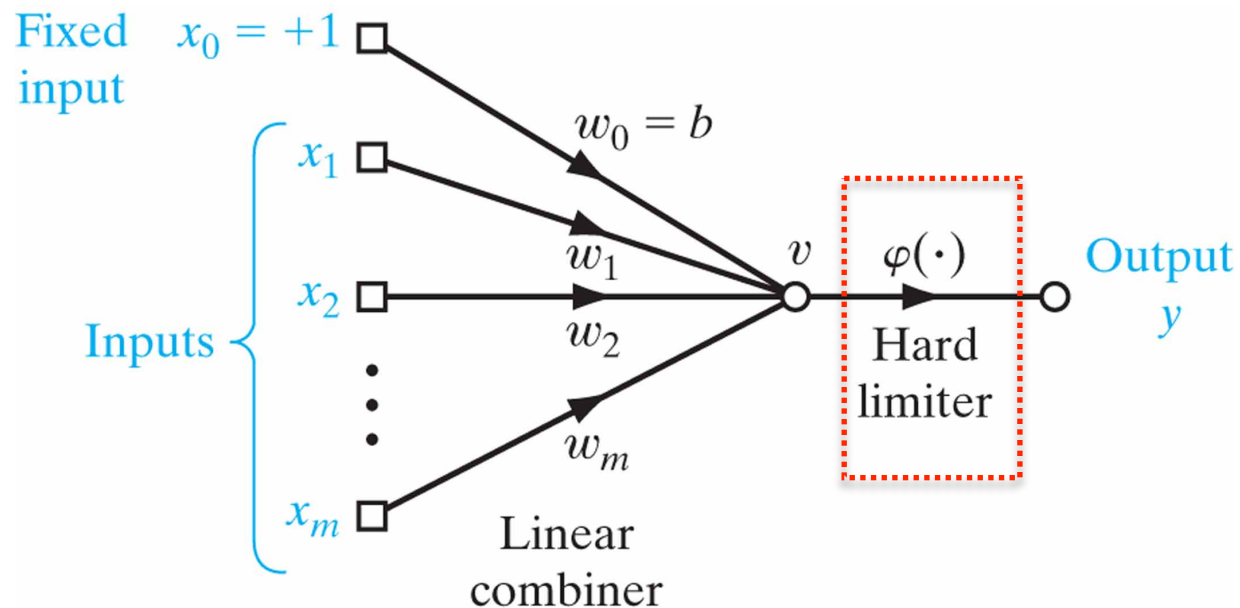
ADALINE

- ADALINE → Neurônio de McCulloch-Pitts



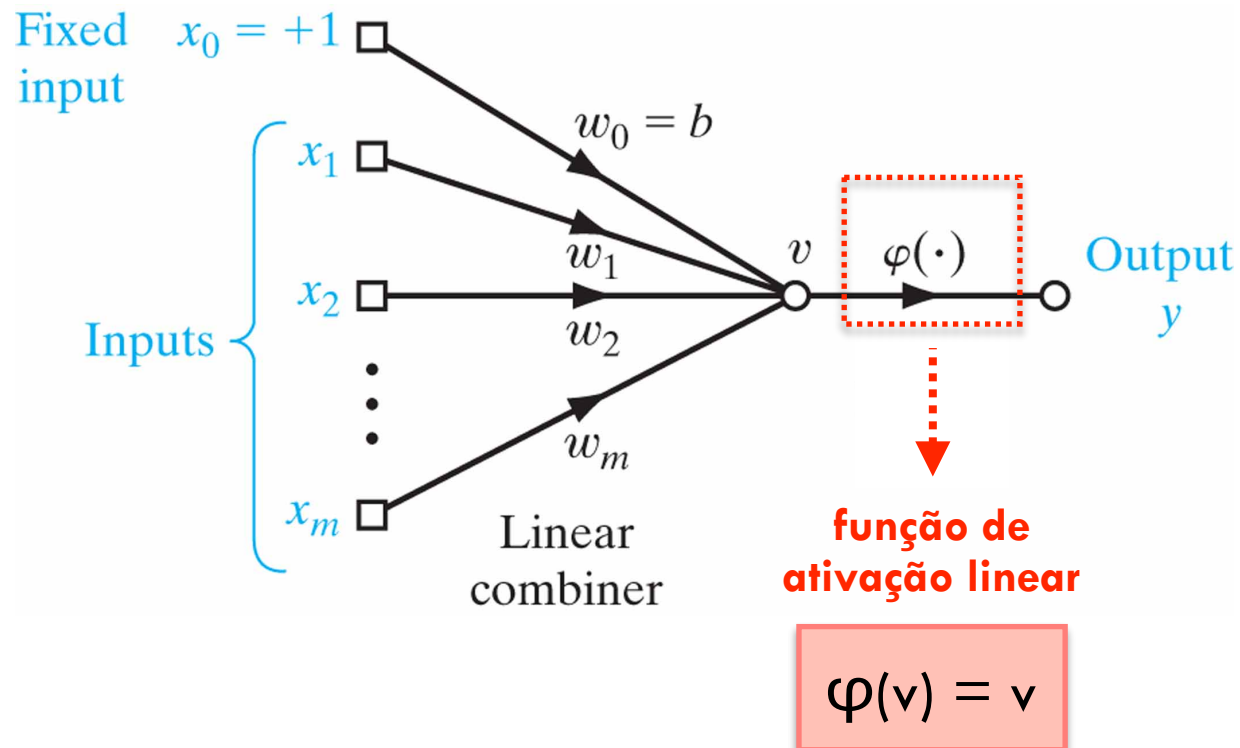
ADALINE

- ADALINE → Neurônio de McCulloch-Pitts



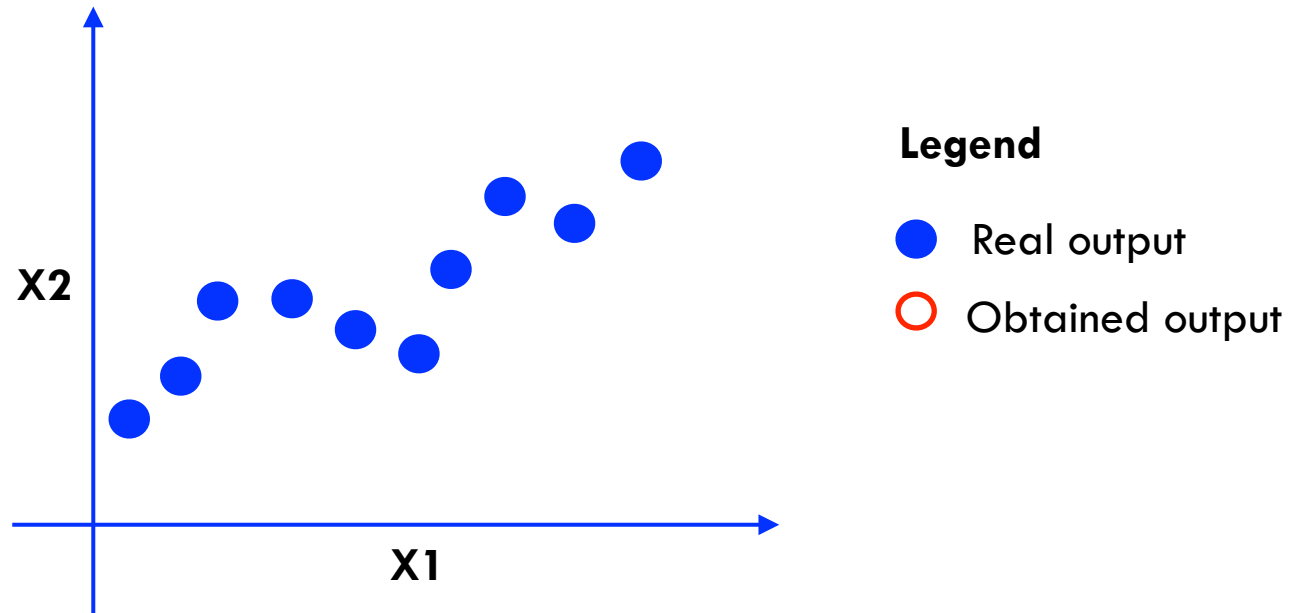
ADALINE

- ADALINE → Neurônio de McCulloch-Pitts



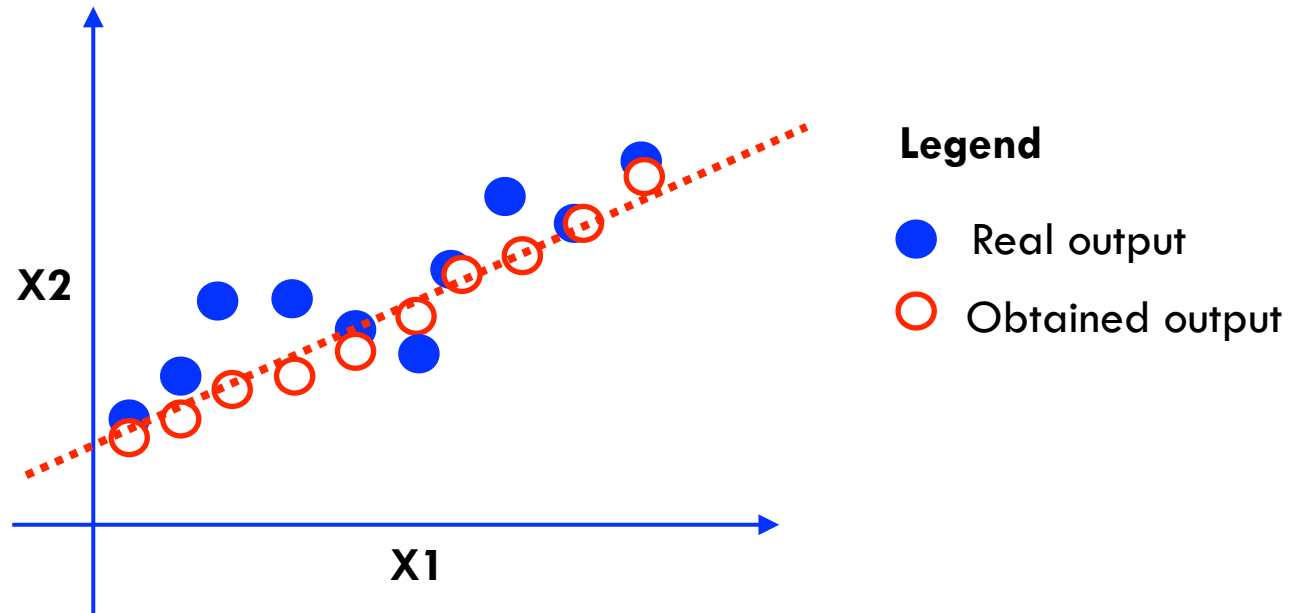
ADALINE

- **Objetivo:** Aproximação de funções (regressão)



ADALINE

- **Objetivo:** Aproximação de funções (regressão)



Adaline

- Pode ser adaptado para classificação
- **Função de ativação:**
 - treinamento (linear):
 - $\varphi(v) = v$
 - teste/predições (degrau):
 - $\varphi(v) = +1$, se $v \geq 0$
= -1 , caso contrário

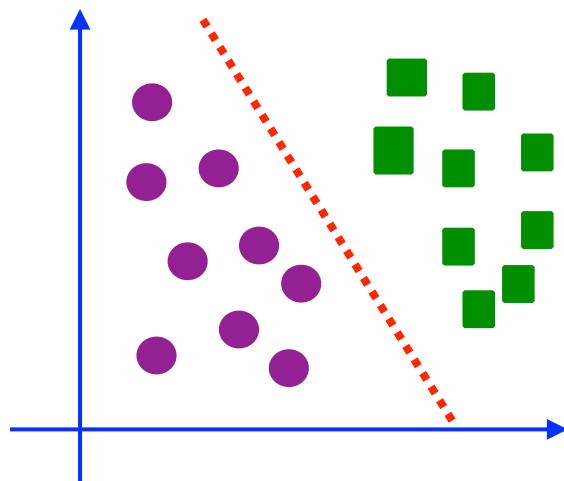
Roteiro

- 1 Introdução
- 2 ADALINE
- 3 ADALINE x Perceptron
- 4 Algoritmos de Treinamento ADALINE
- 5 Exemplo / Exercício
- 6 Síntese / Próximas Aulas
- 7 Referências

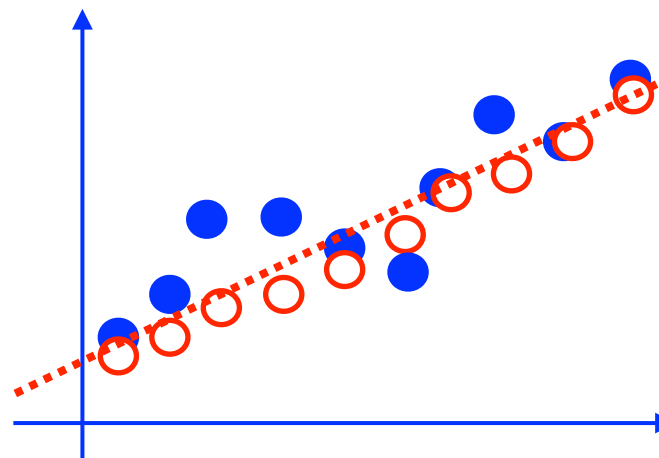
Diferenças

□ Objetivo:

- Perceptron é um separador linear (**classificação**)
- Adaline é um aproximador linear de funções (**regressão**)



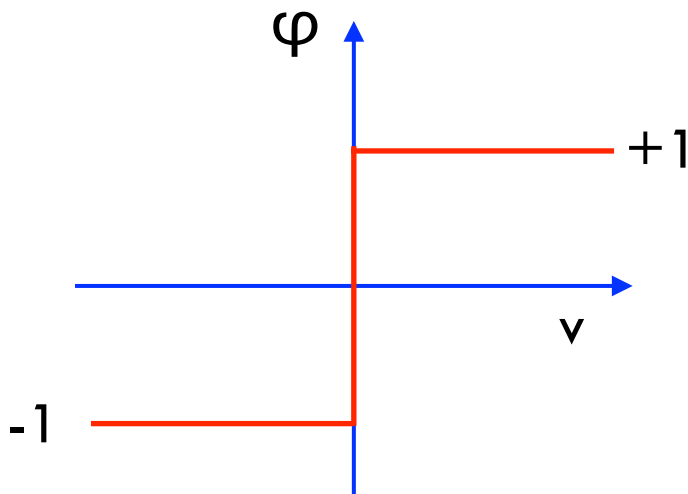
(a) Perceptron



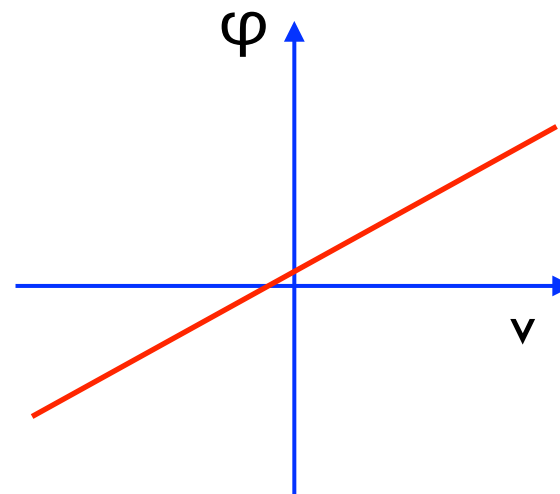
(b) ADALINE

Diferenças

- **Função de Ativação:**
 - Perceptron → função degrau
 - ADALINE → função linear



(a) Perceptron



(b) ADALINE

Diferenças

- **Algoritmo de Treinamento:**

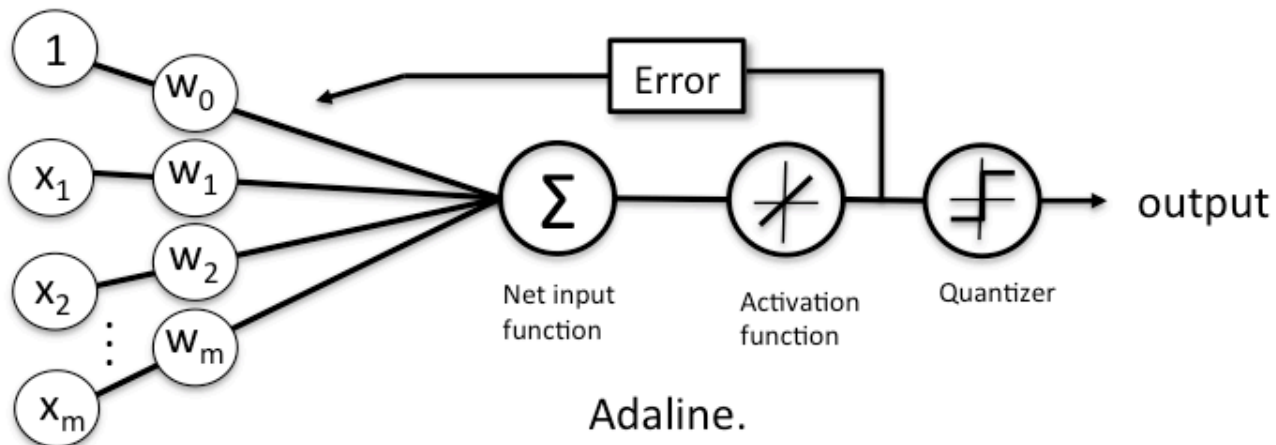
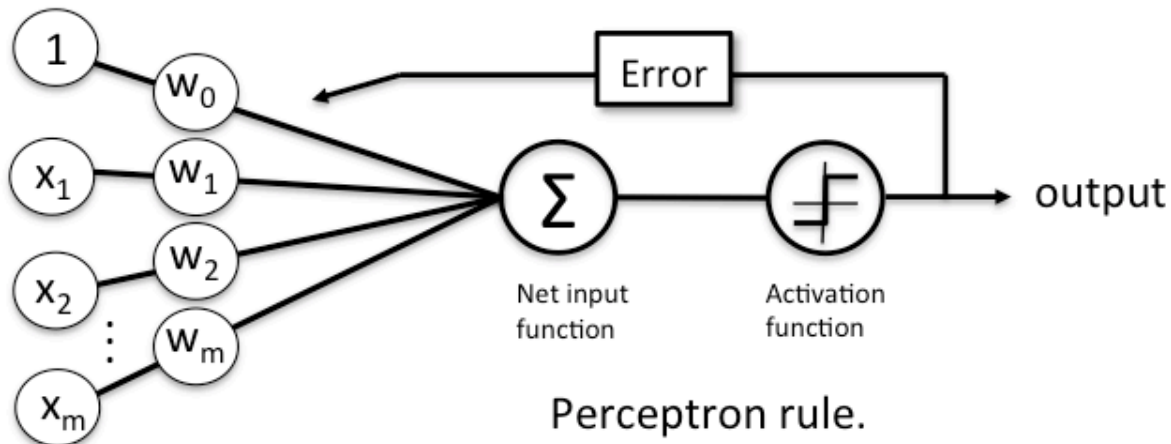
- Perceptron

- atualiza W quando um exemplo está errado

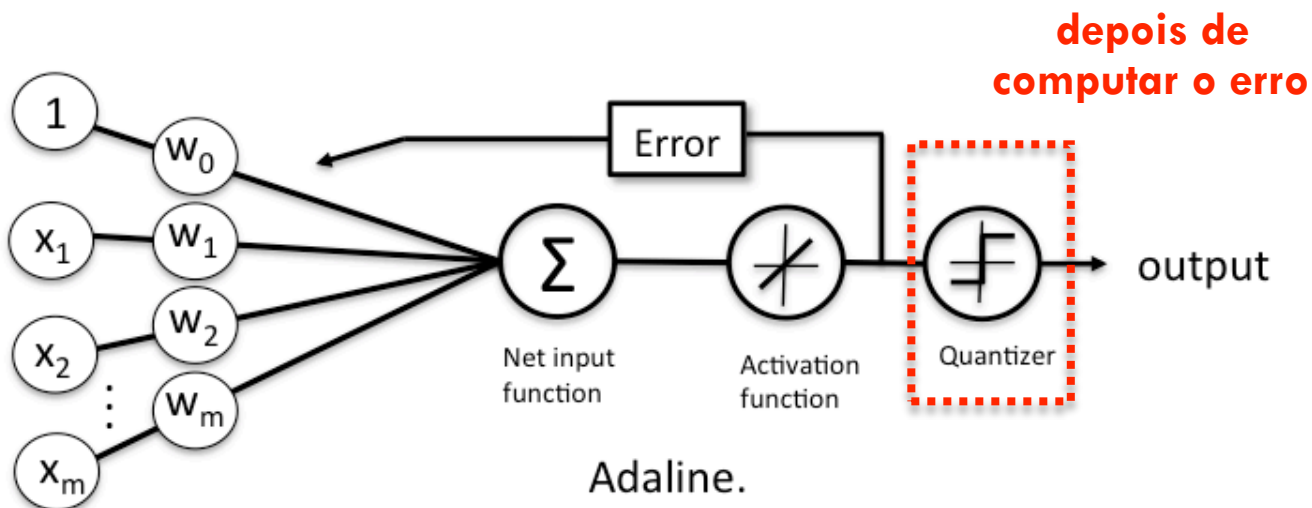
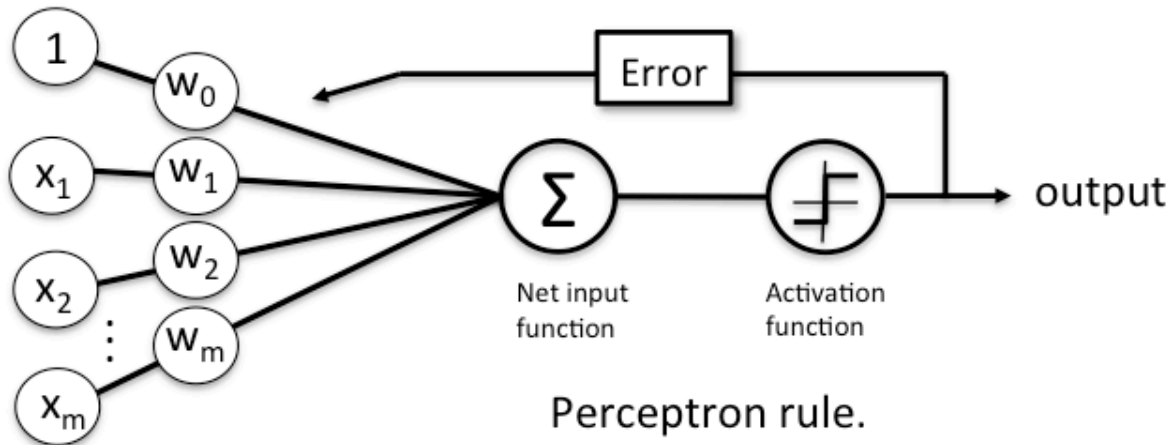
- ADALINE

- Estocástico: atualiza W sempre (para todos exemplos)
 - Batch: Atualiza W considerando todos os exemplos de uma vez (cálculo matricial)

Diferença Treinamento



Diferença Treinamento



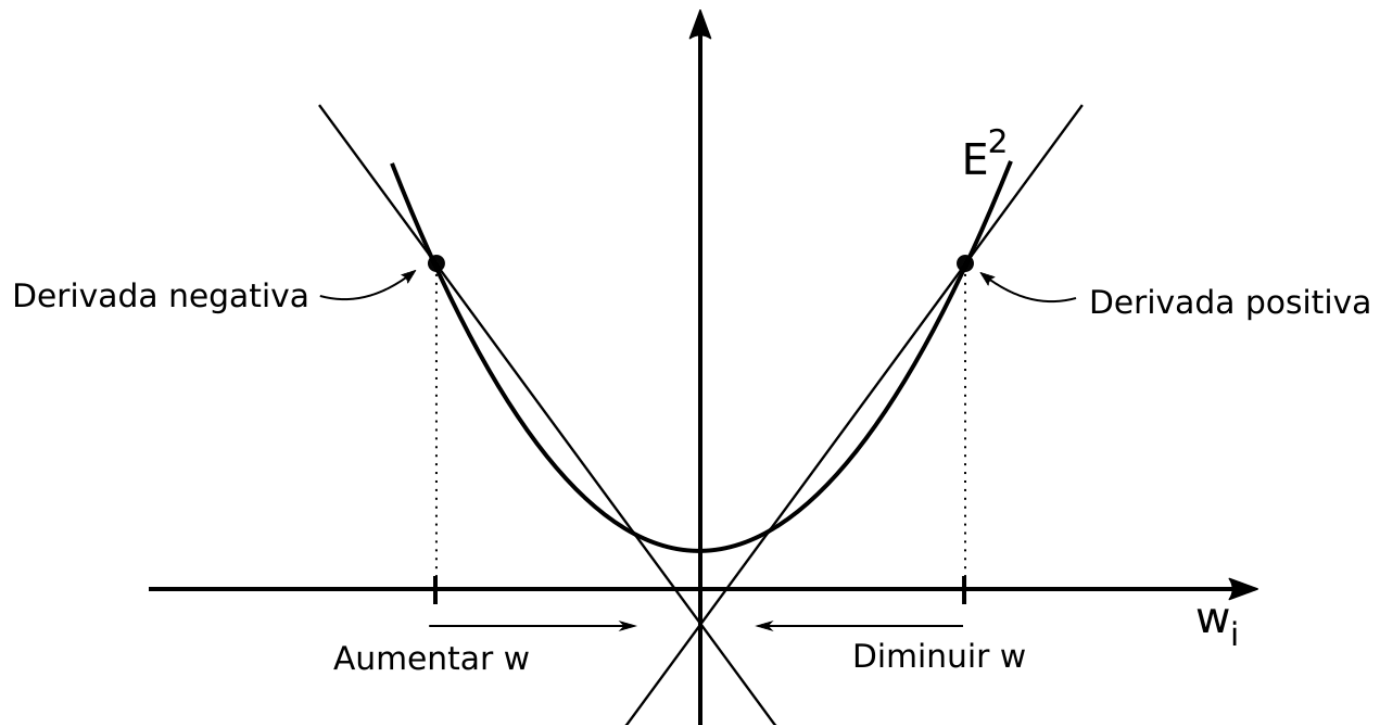
Roteiro

- 1 Introdução
- 2 ADALINE
- 3 ADALINE x Perceptron
- 4 Algoritmo de Treinamento ADALINE
- 5 Exemplo / Exercício
- 6 Síntese / Próximas Aulas
- 7 Referências

Ajuste Sináptico

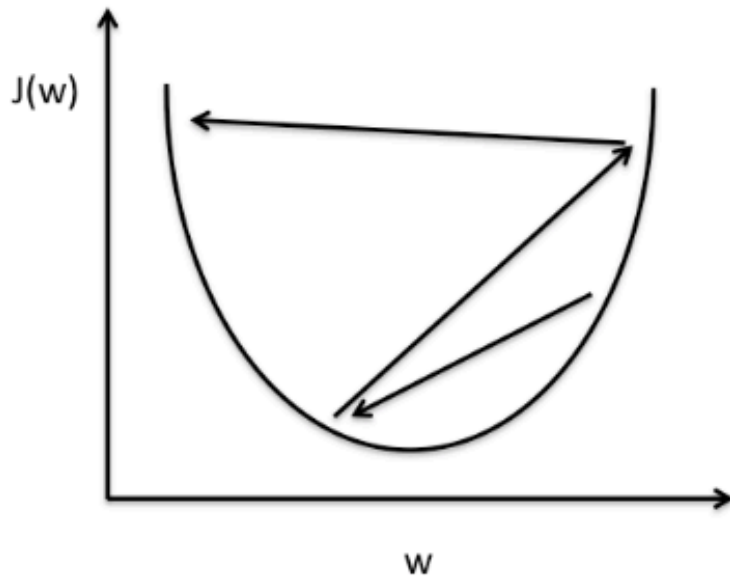
□ Gradiente Descendente

■ $w(n+1) \leftarrow w(n) + \boldsymbol{\eta} (d(n) - y(n)) x(n)$

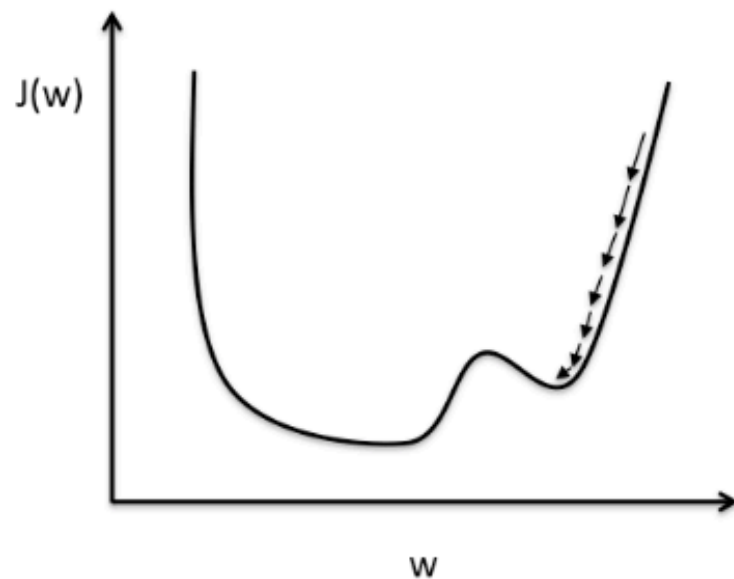


Ajuste Sináptico

- Extremamente sensível ao valor de η



(a) taxa de aprendizado alta :
Gradiente se perde



(b) taxa de aprendizado pequena:
Demora muitos para convergir
Poder cair em mínimos locais

Ajuste Sináptico

- Solução (convergência) → normalização dos dados

- $$\mathbf{X}_{N,\text{std}} = (\mathbf{X}_N - \mu_N) / \sigma_N$$

- \mathbf{X}_N : coluna / atributo do dataset
- μ_N : média dos valores do atributo N
- σ_N : desvio padrão dos valores do atributo N

Algoritmo de Treinamento

- Embora similar ao Algoritmo de atualização dos pesos sinápticos do Perceptron, apresenta duas diferenças:
 1. a saída obtida pela rede é um **número real**, não uma classe
 2. O ajuste sináptico é calculado baseado **em todas as amostras** de treinamento
 - A. Algoritmo Online: Gradiente Descendente Estocástico (SGD)
 - B. Algoritmo Batch: Gradiente Descendente (GD)

Algoritmo de Treinamento (Online)

- **Passo 1.** Inicializar \mathbf{W} com valores aleatórios pequenos
- **Passo 2.** Especificar a taxa de aprendizagem η
- **Passo 3.** Repita
 - 3.1 $\text{epoca} \leftarrow \text{epoca} + 1$
 - 3.2 Para todas as amostras de treinamento $\mathbf{x}(n)$
 - 3.2.1 $v \leftarrow \mathbf{w}^T(n) \mathbf{x}(n)$
 - 3.2.2 $y \leftarrow v$ //ativação linear
 - 3.2.3 Atualiza os pesos sinápticos
 - $\mathbf{w} \leftarrow \mathbf{w} + \eta (d(n) - y(n)) \mathbf{x}(n)$
- // Repetir até que todas que um número de épocas tenha sido executado ou tolerância de erro obtida

Algoritmo de Treinamento (Batch)

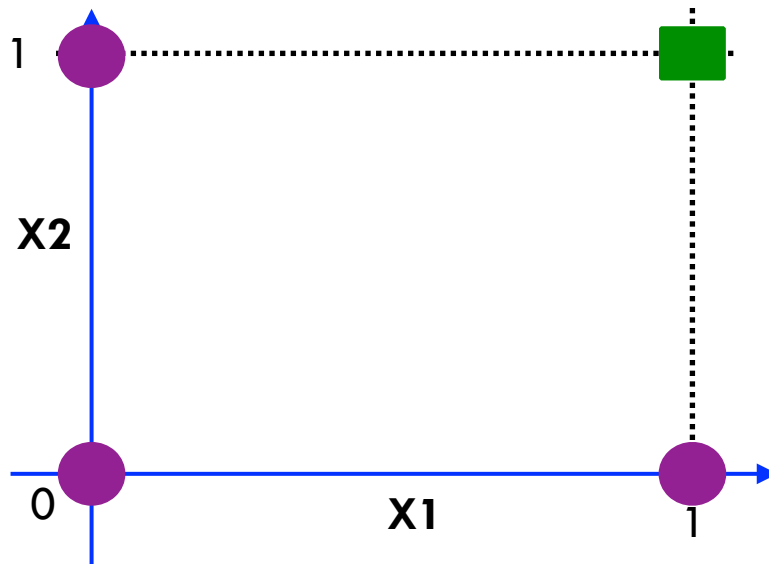
- **Passo 1.** Inicializar \mathbf{W} com valores aleatórios pequenos
- **Passo 2.** Especificar a taxa de aprendizagem η
- **Passo 3.** Repita
 - 3.1 $\text{epoca} \leftarrow \text{epoca} + 1$
 - 3.2 $\mathbf{V} \leftarrow \mathbf{W}^T \mathbf{X}$
 - 3.3 $\mathbf{Y} \leftarrow \mathbf{X}$ //ativação linear
 - 3.4 Atualiza os pesos sinápticos
 - $\mathbf{W} \leftarrow \mathbf{W} + \eta (\mathbf{D} - \mathbf{Y}) \mathbf{X}$
- // Repetir até que todas que um número de épocas tenha sido executado ou tolerância de erro obtida

Roteiro

- 1 Introdução
- 2 ADALINE
- 3 ADALINE x Perceptron
- 4 Algoritmo de Treinamento ADALINE
- 5 Exemplo / Exercício
- 6 Síntese / Próximas Aulas
- 7 Referências

Exemplo/Exercício

- Treinar ADALINE para reconhecer o problema lógico AND, usando o Gradiente Descendente Estocástico. Dados:
 - $w_0 = w_1 = w_2 = 0.5$
 - $\text{bias} = +1$
 - $\eta = 0.1$



x_1	x_2	D
0	0	0
0	1	0
1	0	0
1	1	1

Roteiro

- 1 Introdução
- 2 ADALINE
- 3 ADALINE x Perceptron
- 4 Algoritmos de Treinamento ADALINE
- 5 Exemplo / Exercício
- 6 Síntese / Próximas Aulas
- 7 Referências

Síntese/Revisão

- Adaline
 - um neurônio de McCulloch Pitts
 - bias
 - função de ativação **linear**
- Gradiente Descendente
- Algoritmo de Aprendizado do Adaline
- Exemplo

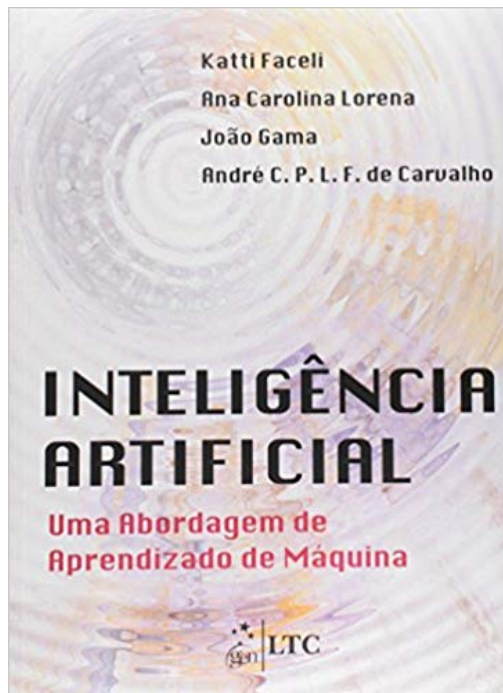
Próxima Aula

- ADALINE x Perceptron Simples
 - implementação dos algoritmos (R/Python)
 - prática: AT01

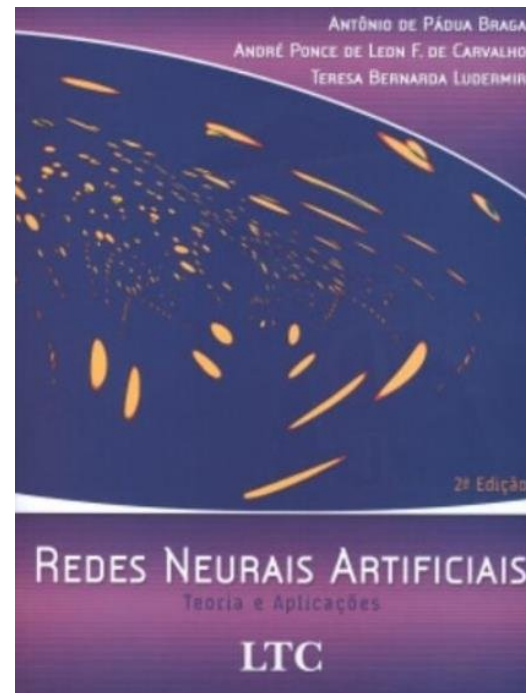
Roteiro

- 1 Introdução
- 2 ADALINE
- 3 ADALINE x Perceptron
- 4 Algoritmos de Treinamento ADALINE
- 5 Exemplo / Exercício
- 6 Síntese / Próximas Aulas
- 7 Referências

Literatura Sugerida

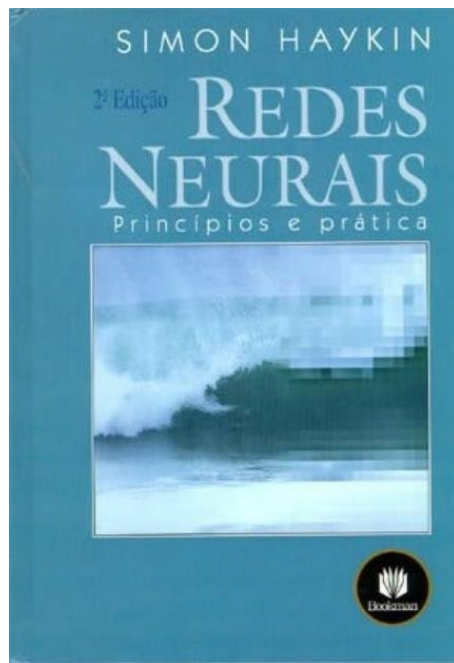


[Faceli et al, 2011]

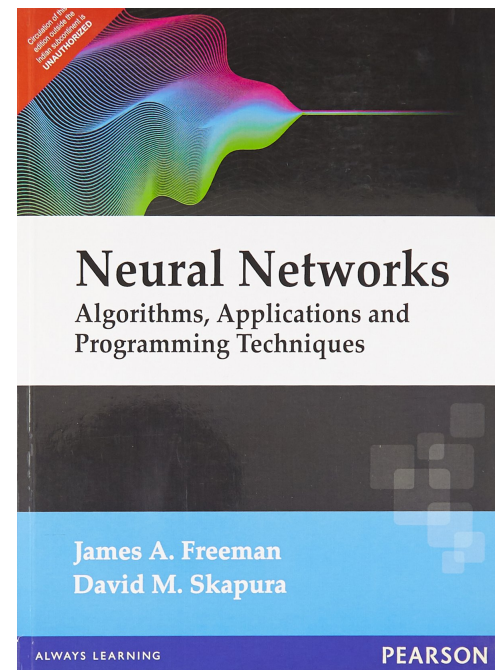


[Braga et al, 2007]

Literatura Sugerida



(Haykin, 1999)



(Freeman & Skapura, 1991)

Referências Adicionais

- F. Rosenblatt. The perceptron, a perceiving and recognizing automaton Project Para. Cornell Aeronautical Laboratory, 1957.
- B. Widrow et al. Adaptive "Adaline" neuron using chemical "memistors". Number Technical Report 1553-2. Stanford Electron. Labs., Stanford, CA, October 1960.

Perguntas?

Prof. Rafael G. Mantovani

rgmantovani@gmail.com