



# Neurônios Perceptron e Adaline

Disciplina: Inteligência Artificial

“Prova Didática do EDITAL Nº121/2023, nos termos do Despacho nº 155/2023- RUNESP de 08/05/2023”

Setembro de 2023

**Candidato:** Danilo Roberto Pereira



## ■ **Resumo - Breve Histórico**

### ■ **Neurônio**

- Perceptron
- Adaline
- O que é a variável de viés? Qual sua importância?
- Treinamento em lote (*batch*)

### ■ **Limitações dos neurônios**

- Como contornar as limitações?

### ■ **Redes Neurais Multicamadas**

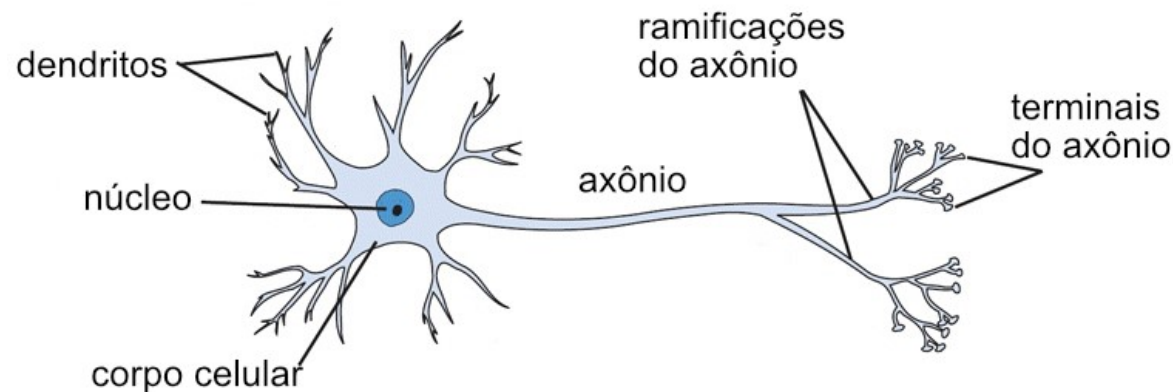
- Introdução
- Leitura sugerida- Próxima aula

### ■ **Exercício**

### ■ **Bibliografia**

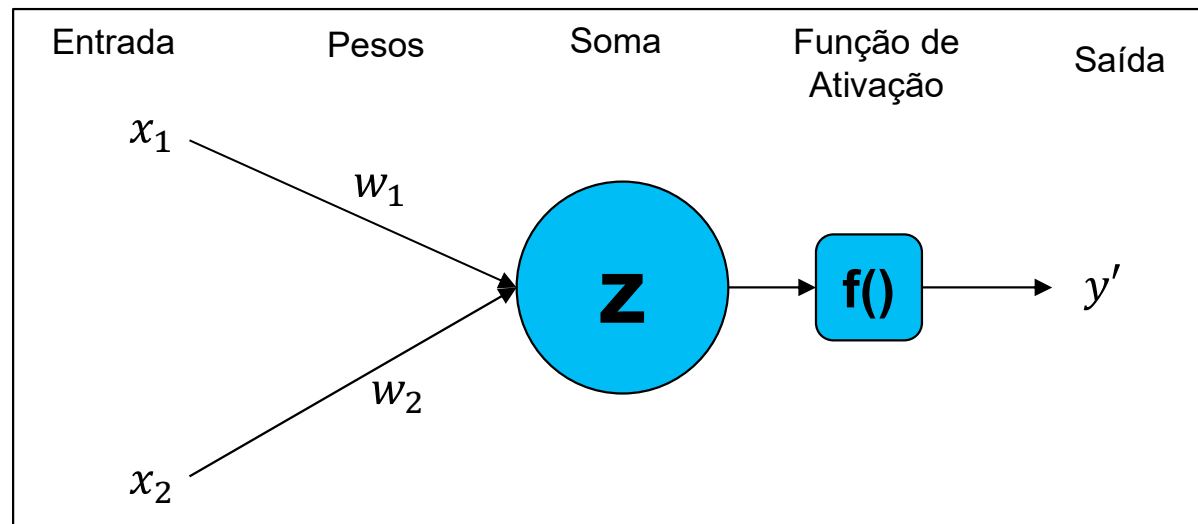
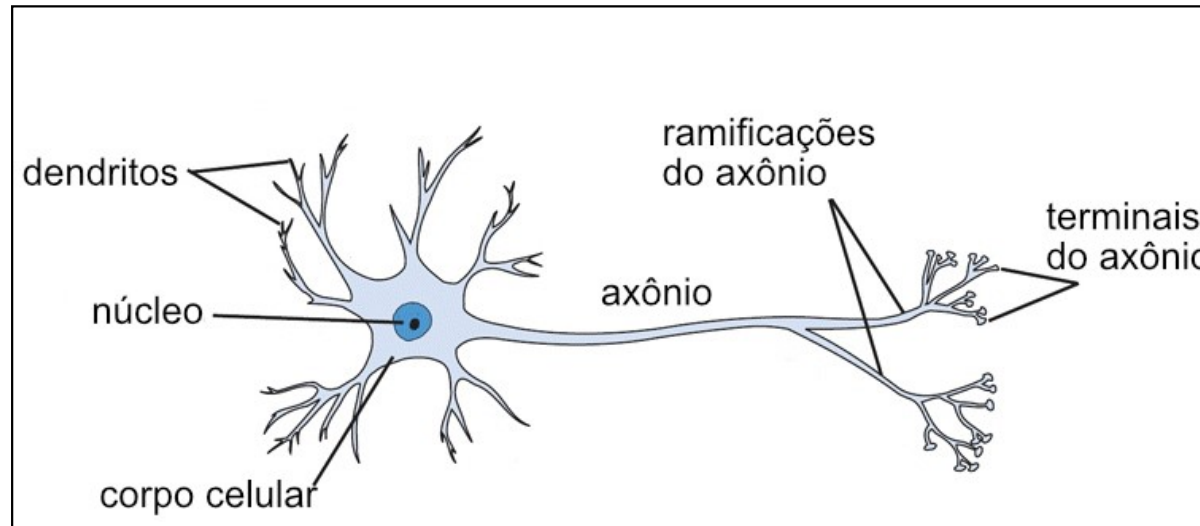


- **O sistema nervoso de mamíferos é formado por um conjunto extremamente complexo de células, os neurônios.**
  - Exerce papel essencial na determinação do funcionamento e comportamento do corpo humano e do raciocínio



- **Redes Neurais Artificiais são modelos matemático baseados na estrutura biológica de um neurônio.**

# Breve Histórico





- **Resumo - Breve Histórico**

- **Neurônio**

- Perceptron
- Adaline
- O que é a variável de viés? Qual sua importância?
- Treinamento em lote (*batch*)

- **Limitações dos neurônios**

- Como contornar as limitações?

- **Redes Neurais Multicamadas**

- Introdução
- Leitura sugerida- Próxima aula

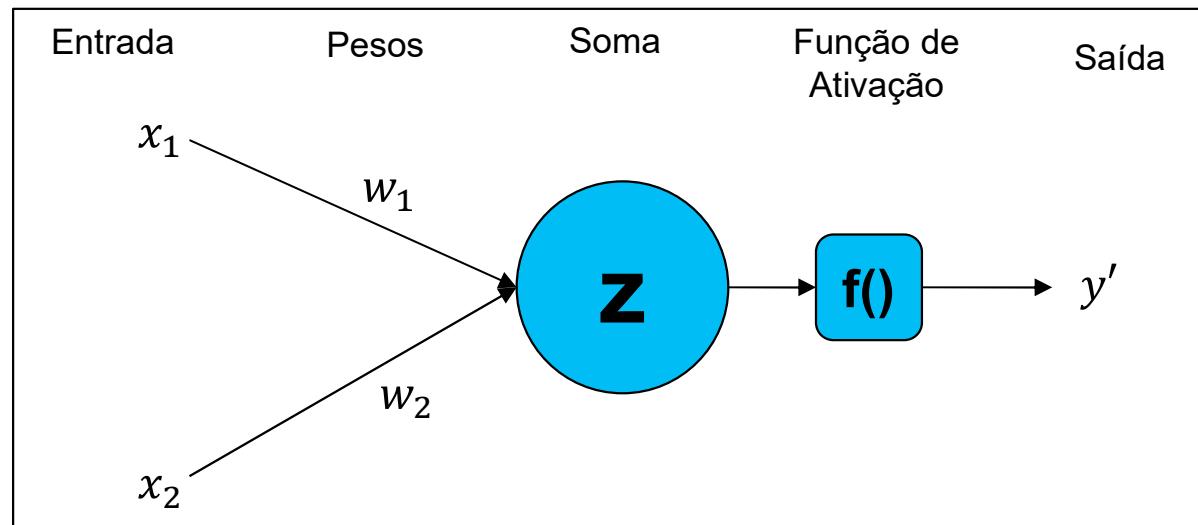
- **Exercício**

- **Bibliografia**

# Neurônio Perceptron - Introdução



- Foi proposto em 1958 por Frank Rosenblatt



$$Z = x_1 w_1 + x_2 w_2$$

$$f(z) = f(x_1 w_1 + x_2 w_2)$$

$$y' = f(z) = \begin{cases} 1 & \text{se } z \geq \theta \\ 0 & \text{se } z < \theta \end{cases}$$

$$erro = y - y'$$



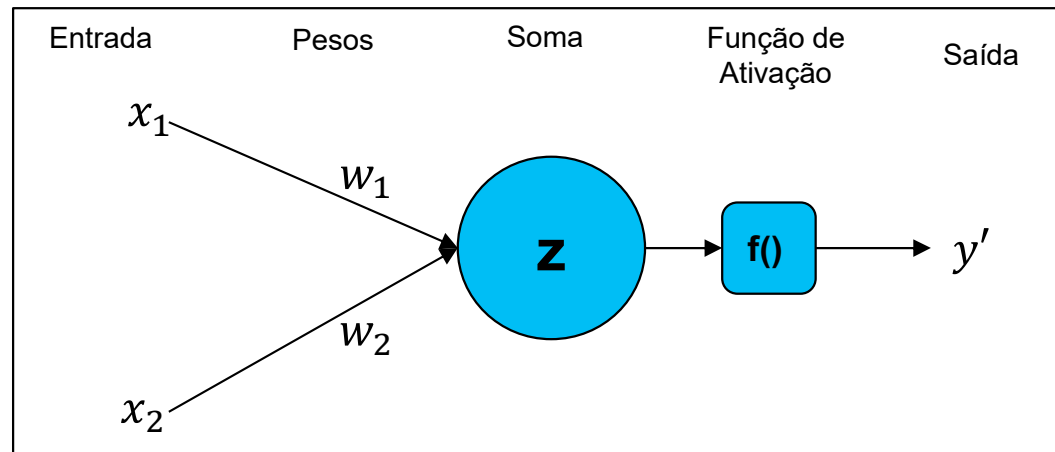
## ■ Processo de Aprendizado:

- Inicialização dos Pesos: Os pesos são inicializados com valores aleatórios ou zero.
- Cálculo da Soma Ponderada: A soma ponderada  $z$  das entradas e pesos é calculada.
- Função de Ativação: A soma ponderada  $z$  é passada por uma função de ativação  $f$ 
  - $y' = f(z)$

## ■ Treinamento usando a Regra de Aprendizado do Perceptron:

- Ajustar os pesos para que o Perceptron classifique corretamente os dados de treino.
- A regra de aprendizado atualiza os pesos com base no erro de classificação.
- Atualização dos Pesos (Fórmula):
  - $w_i = w_i + \eta (y - y')x_i$ 
    - onde  $w_i$  é o peso da entrada  $x_i$ ,  $\eta$  é a taxa de aprendizado,  $y$  é o rótulo verdadeiro e  $y'$  é a saída do Perceptron.

# Neurônio Perceptron - Introdução



$$w_i = w_i + \eta (y - y')x_i$$

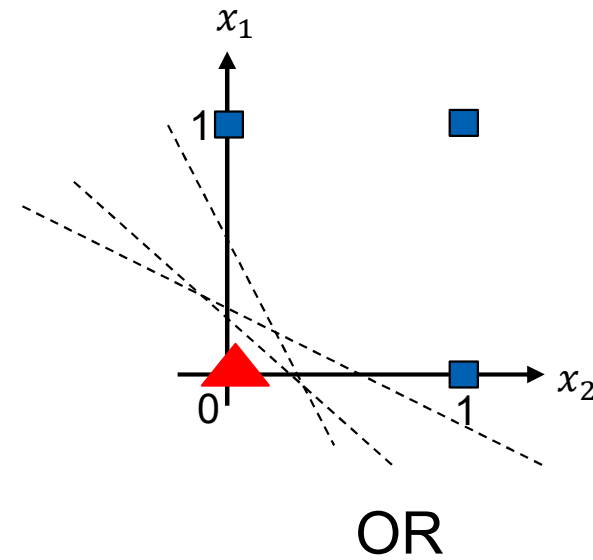
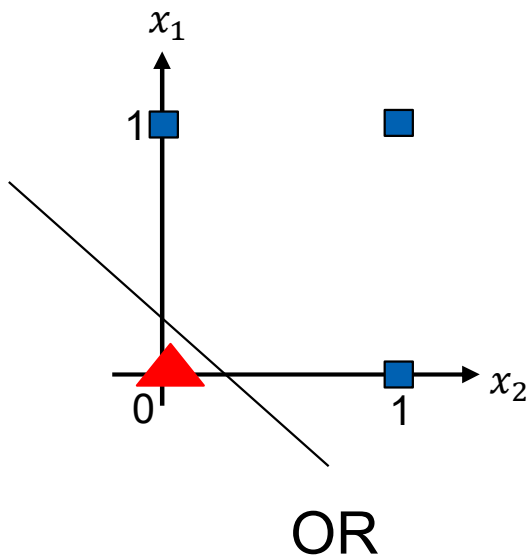
Amostra	x1	x2	y
0	1	0	1
1	0	0	0
2	0	1	1
3	1	1	1

$\theta$	w1	w2	Learning Rate
0.5	0.2	0.3	0.25

Amostra	x1	x2	y	z	y' = f(z)	Erro (y-y')	w1'	w2'
0	1	0	1	0.2	0	1	0.45	0.3
1	0	0	0	0	0	0	0.45	0.3
2	0	1	1	0.3	0	1	0.45	0.55
3	1	1	1	1	1	0	0.45	0.55



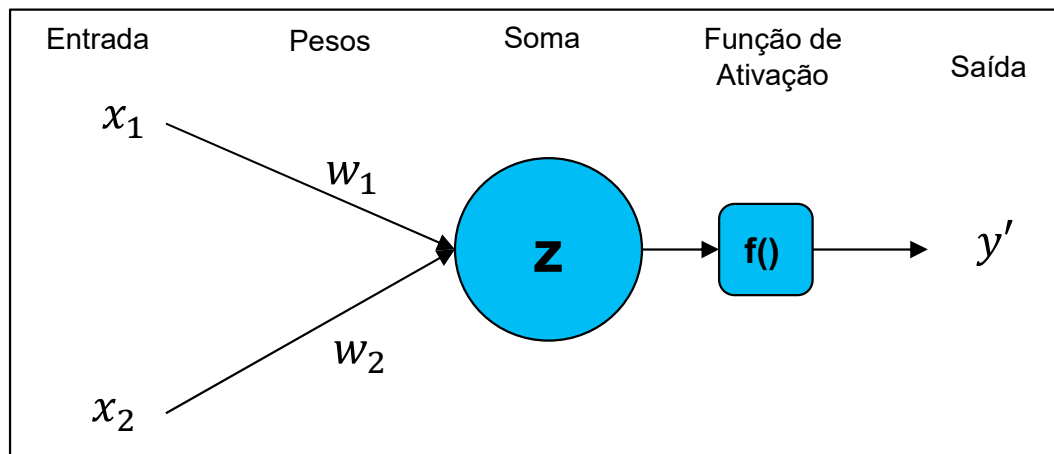
# Perceptron – Exemplo Prático



# Neurônio Adaline - Introdução



- O Neurônio do tipo Adaline (*Adaptive Linear Element*) foi proposto por Widrow e Hoff em 1960.



$$erro = \frac{(y - y')^2}{2}$$

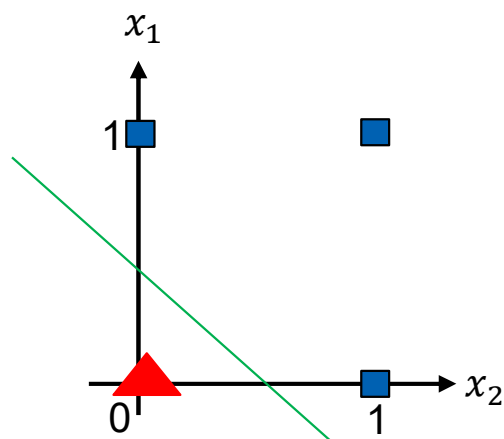
$$w_i = w_i + \eta (y - z)x_i$$

Amostra	x1	x2	y
0	1	0	1
1	0	0	0
2	0	1	1
3	1	1	1

$\theta$	w1	w2	Learning Rate
0.5	0.2	0.3	0.25

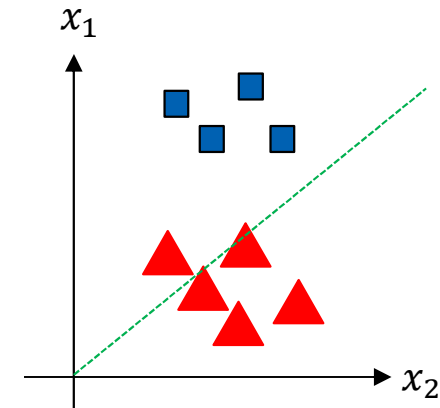
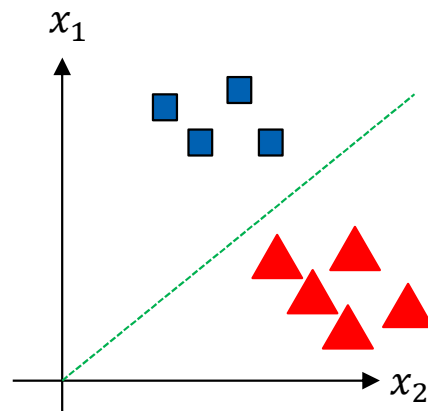
Amostra	x1	x2	y	z	y' = f(z)	Erro [ (y-z)^2 / 2 ]	dErro [y-z]	w1'	w2'
0	1	0	1	0.2	0	0.32	0.8	0.4	0.3
1	0	0	0	0	0	0	0	0.4	0.3
2	0	1	1	0.3	0	0.245	0.7	0.4	0.475
3	1	1	1	0.875	1	0.0078125	0.125	0.43125	0.50625

# Adaline – Exemplo Prático



OR

# O que é a variável de viés?



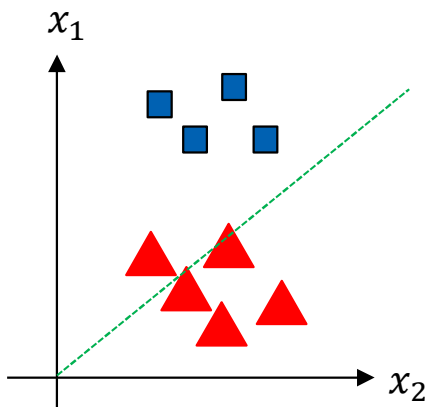
$$z = x_1 w_1 + x_2 w_2$$

$$f(z) = f(x_1 w_1 + x_2 w_2)$$

# O que é a variável de viés?

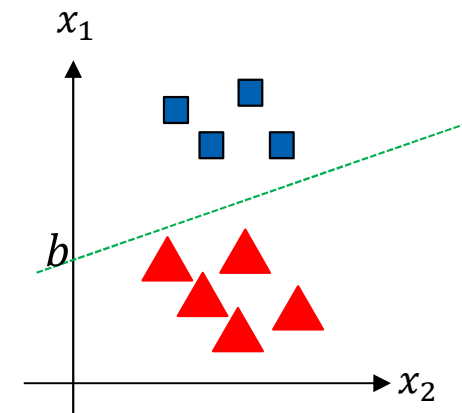


- O viés é um valor adicional adicionado à soma ponderada das entradas antes de ser passado pela função de ativação. Isso permite que o neurônio tenha um deslocamento (ou *bias*) na ativação



$$z = x_1 w_1 + x_2 w_2$$

$$f(z) = f(x_1 w_1 + x_2 w_2)$$



$$z = x_1 w_1 + x_2 w_2 + b$$

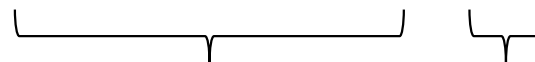
$$f(z) = f(x_1 w_1 + x_2 w_2 + b)$$

# O que é a variável de viés?



- O viés é um valor adicional adicionado à soma ponderada das entradas antes de ser passado pela função de ativação. Isso permite que o neurônio tenha um deslocamento (ou *bias*) na ativação

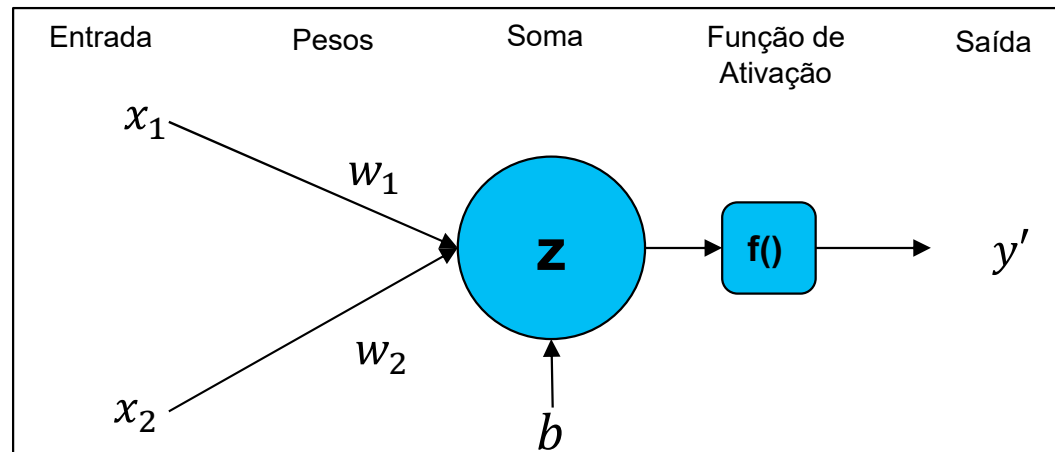
$$f(z) = f(x_1w_1 + x_2w_2 + b)$$



Coeficiente angular

Coeficiente linear

# Neurônio Adaline com variável de viés



$$w_i = w_i + \eta (y - z)x_i$$

$$b = b + \eta (y - z)$$

Amostra	x1	x2	y
0	1	0	1
1	0	0	0
2	0	1	1
3	1	1	1

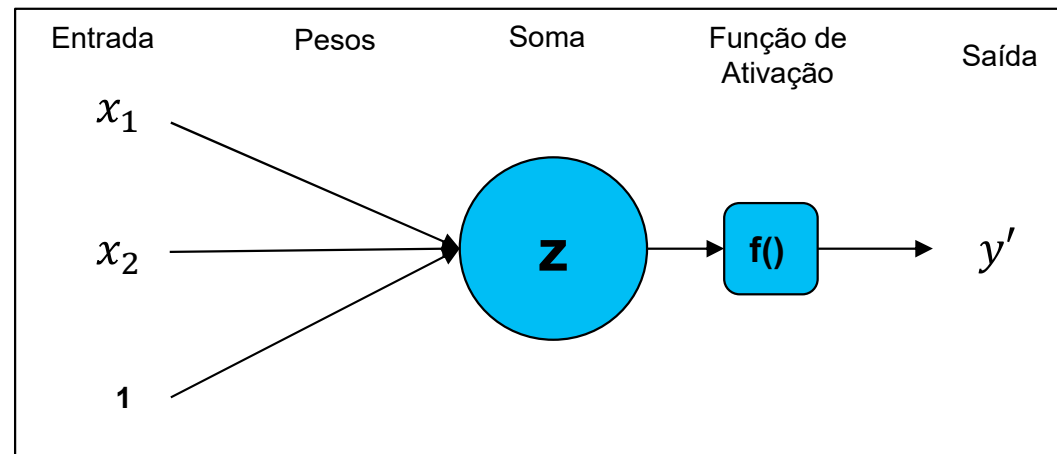
$\theta$	w1	w2	b	Learning Rate
0.5	0.2	0.3	1	0.5

x1	x2	y	z	y' = f(z)	Erro [ (y-z)^2 / 2 ]	w1'	w2'	b'
1	0	1	0.7	1	0.045	0.35	0.3	1.15
0	0	0	0	1	0.66125	0.35	0.3	0.575
0	1	1	0.875	1	0.0078125	0.35	0.3625	0.6375
1	1	1	1.35	1	0.06125	0.175	0.1875	0.4625

# Neurônio Adaline com variável de viés



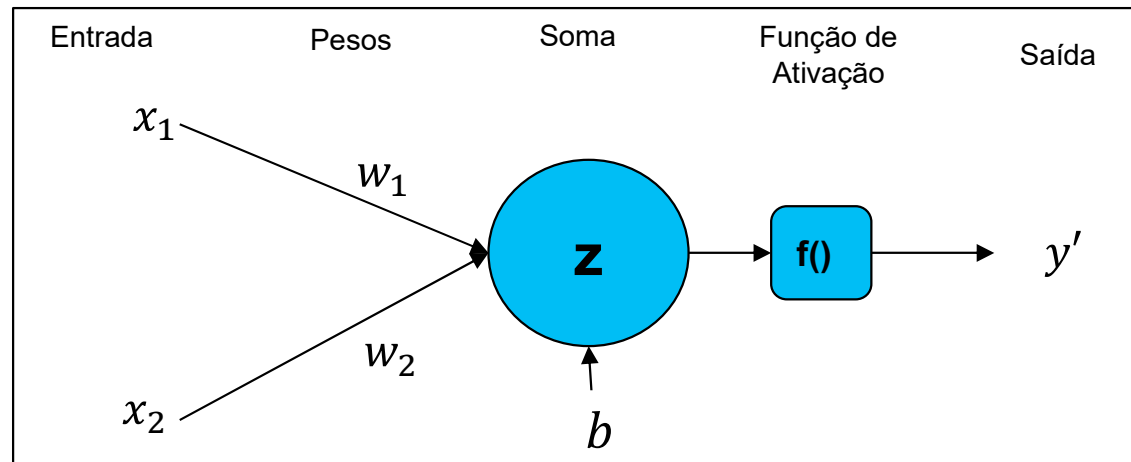
- É possível tratar o viés como uma entrada adicional de valor constante 1.



$$z = x_1w_1 + x_2w_2 + 1w_b$$



# Neurônio Adaline – Treinamento em lote (*batch*)



$$w_i = w_i + \Delta w_i$$

$$\Delta w_i = \eta \frac{1}{N} \sum_{j=1}^N (y - s^j) x_{ij}$$

$$b = b + \Delta b$$

$$\Delta b = \eta \frac{1}{N} \sum_{j=1}^N (y - s^j)$$

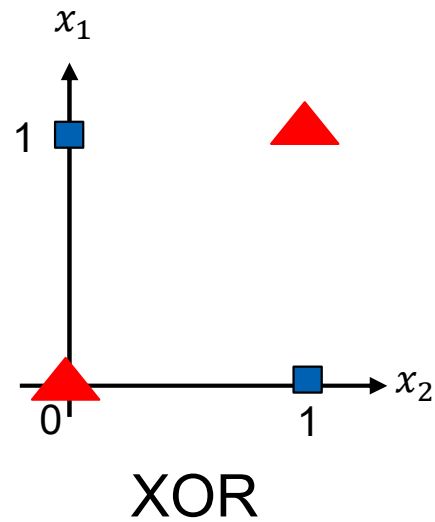
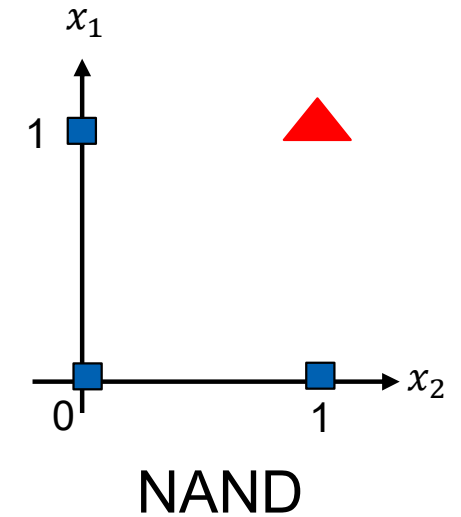
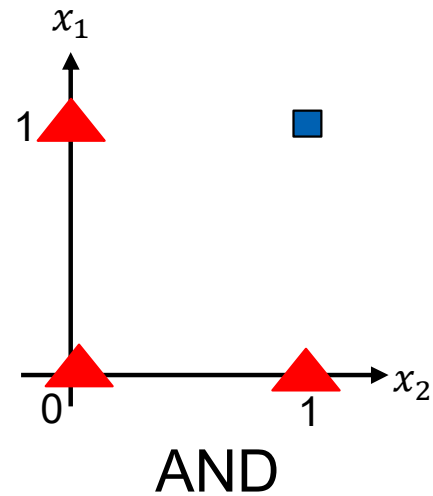
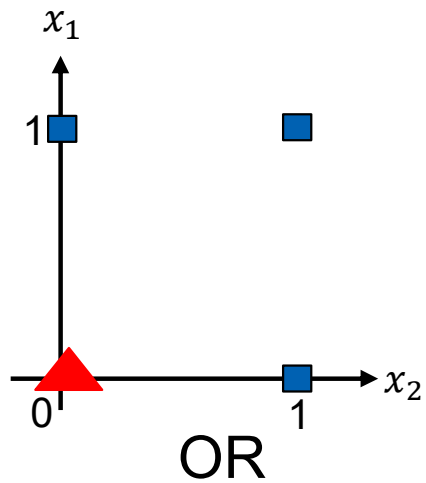


- **Resumo - Breve Histórico**
- **Neurônio**
  - Perceptron
  - Adaline
  - O que é a variável de viés? Qual sua importância?
  - Treinamento em lote (*batch*)

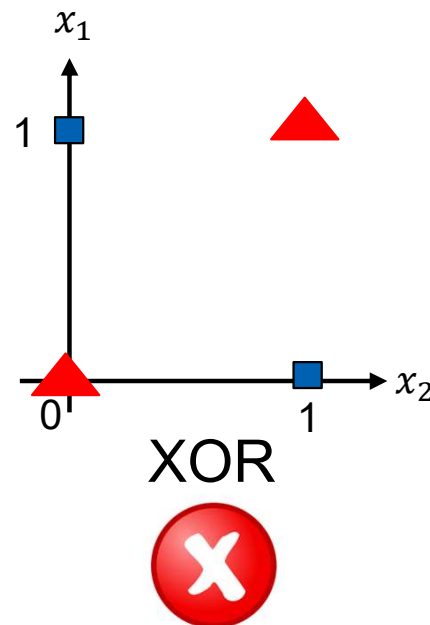
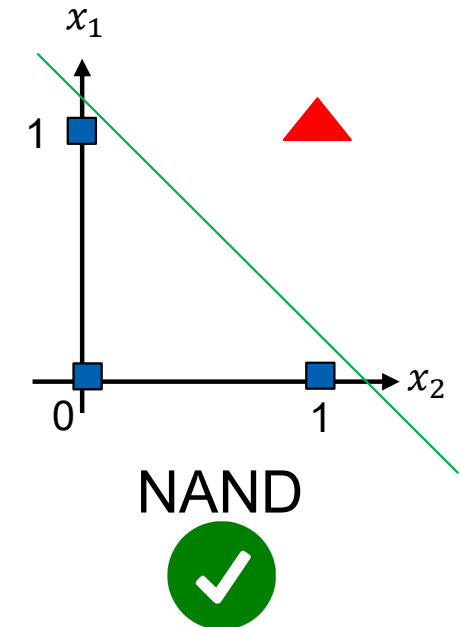
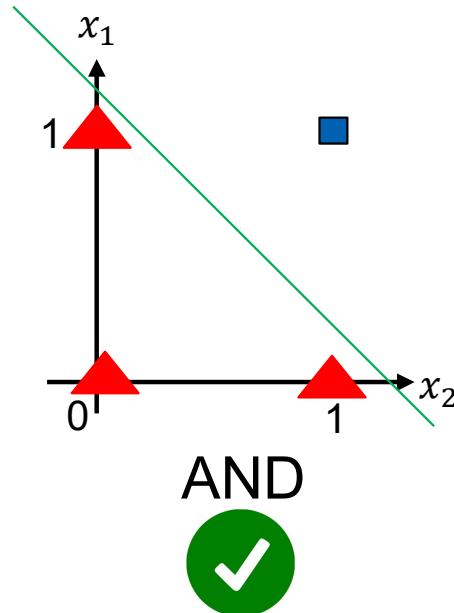
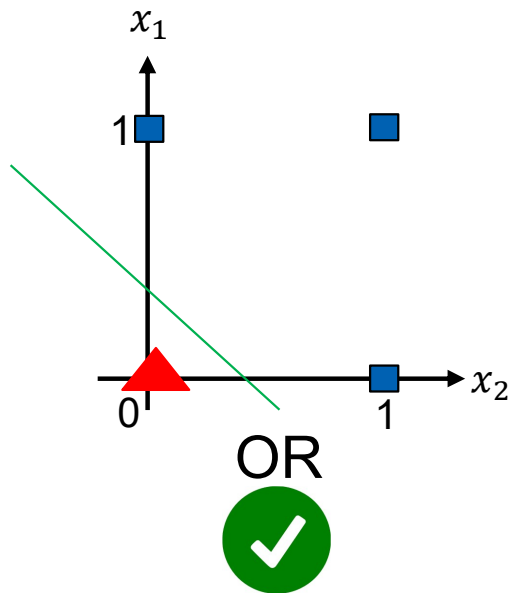
- **Limitações dos neurônios**
  - Como contornar as limitações?

- **Redes Neurais Multicamadas**
  - Introdução
  - Leitura sugerida- Próxima aula
- **Exercício**
- **Bibliografia**

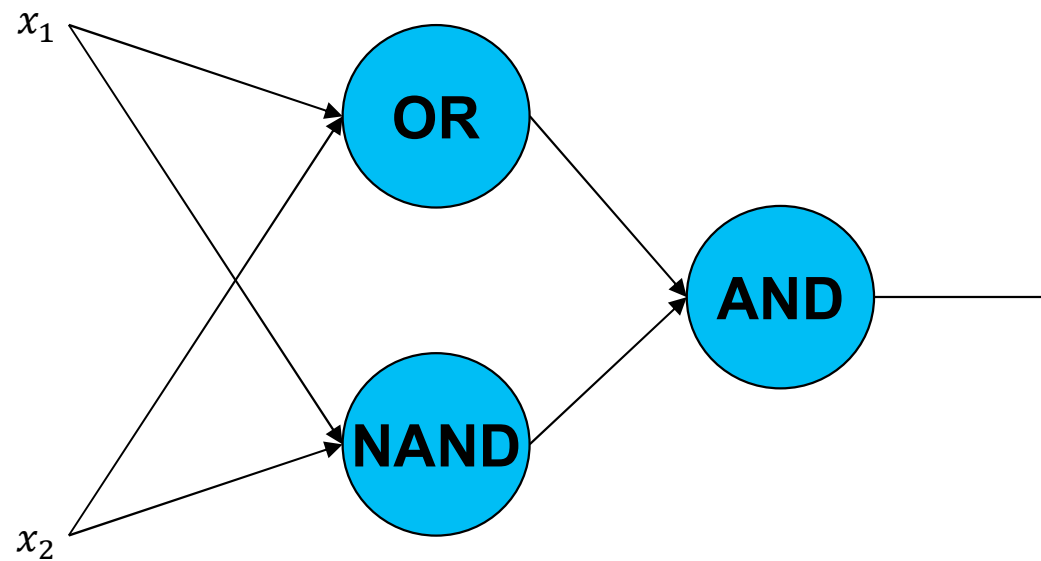
# Limitações dos neurônios



# Limitações dos neurônios

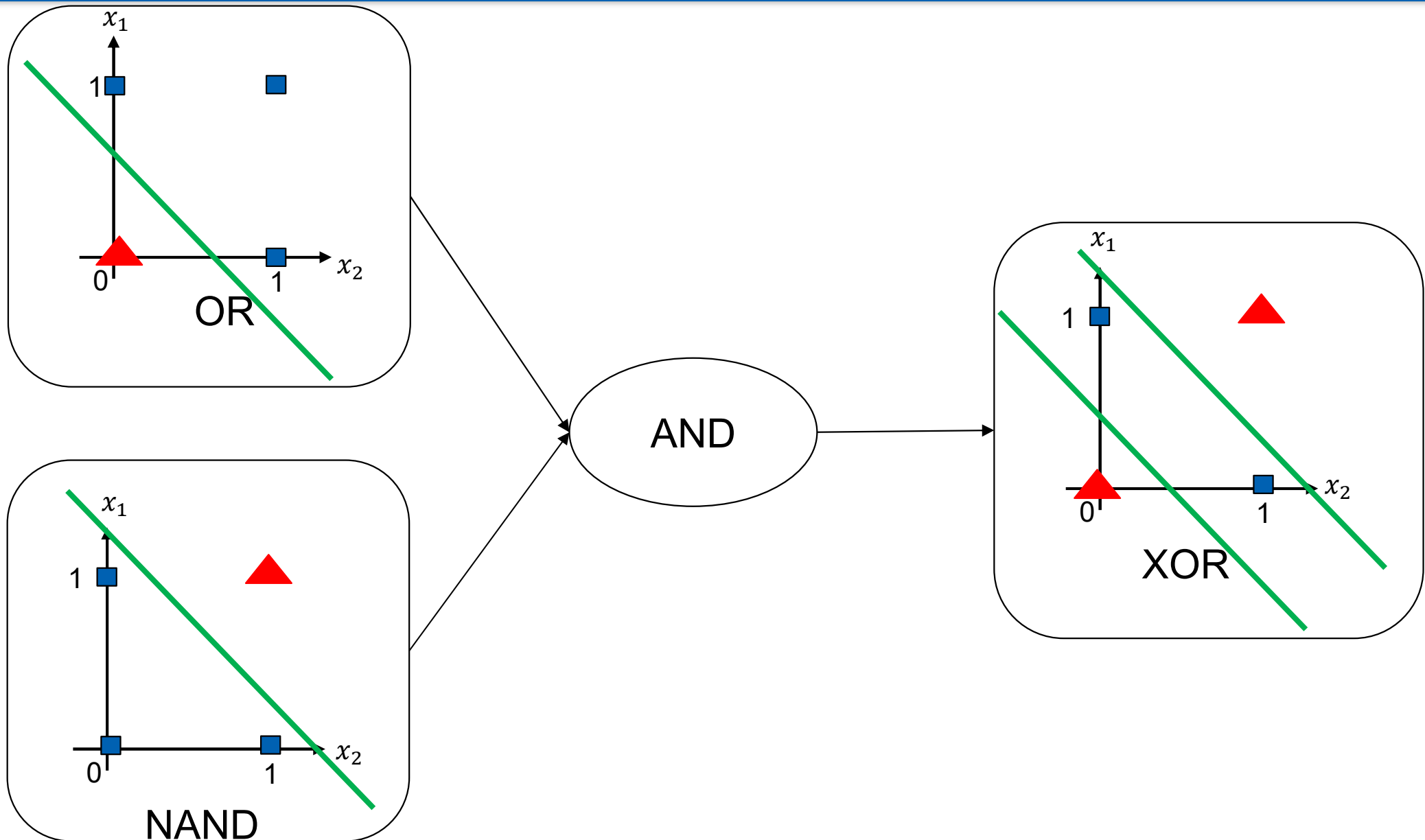


# Limitações dos neurônios - Como contornar as limitações?



# Como contornar as limitações?

## Abstração

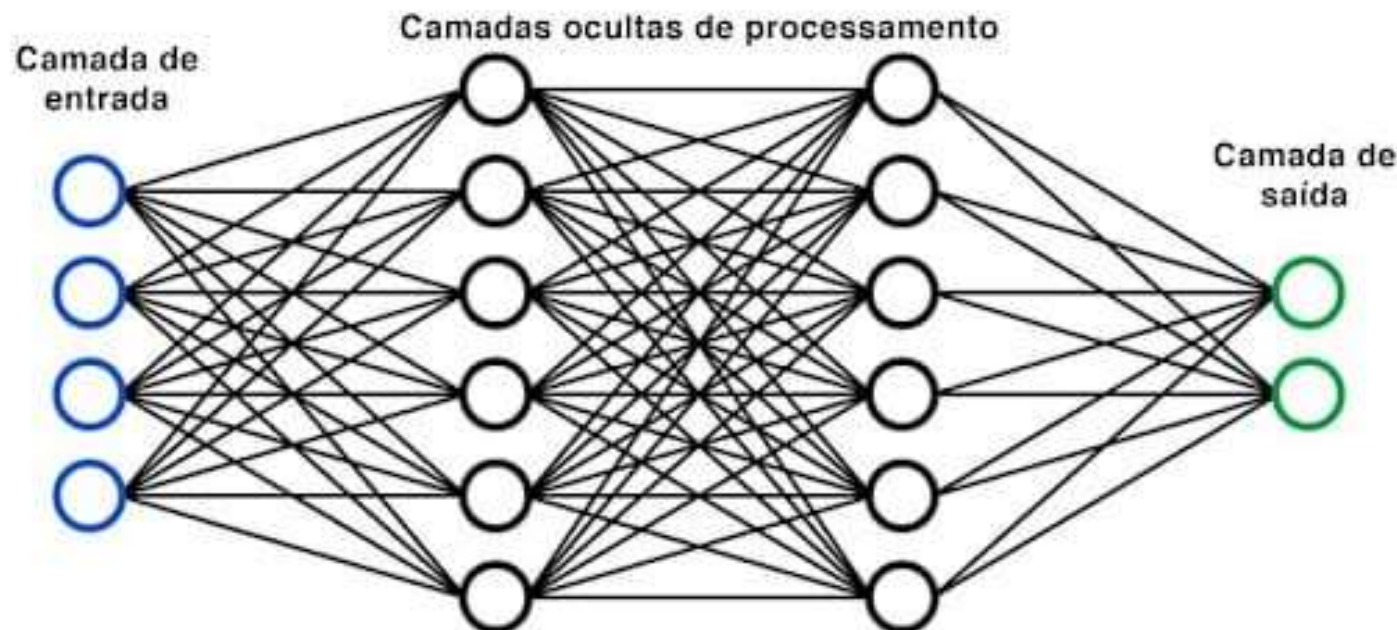




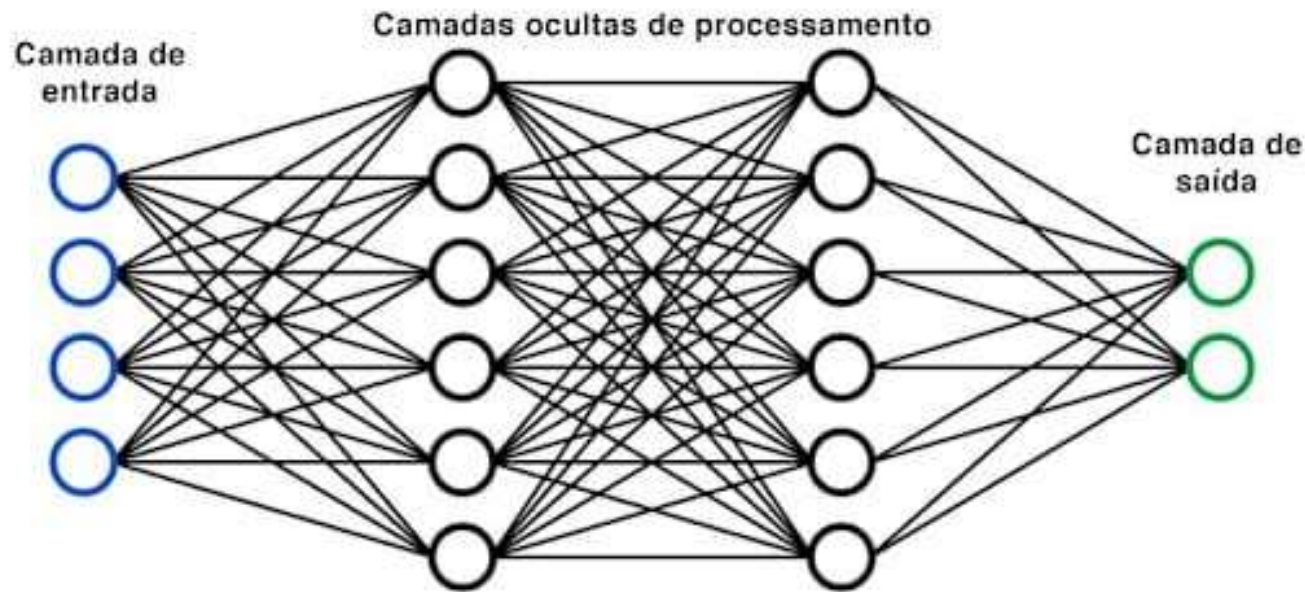
- **Resumo - Breve Histórico**
- **Neurônio**
  - Perceptron
  - Adaline
  - O que é a variável de viés? Qual sua importância?
  - Treinamento em lote (*batch*)
- **Limitações dos neurônios**
  - Como contornar as limitações?
- **Redes Neurais Multicamadas**
  - Introdução
  - Leitura sugerida- Próxima aula
- **Exercício**
- **Bibliografia**



- Não linearidades são inerentes à maioria das situações e problemas reais.
- Não linearidades são incorporadas através:
  - Da composição de sucessivas camadas de neurônios (vide exemplo do XOR).
  - De funções de ativação não lineares.







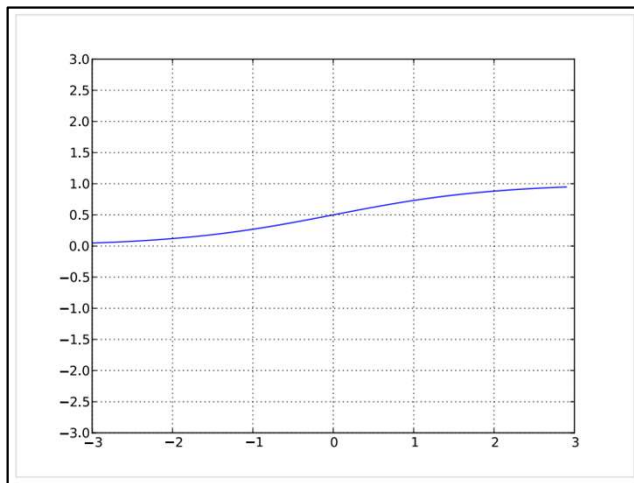
- **Saída somente à última camada.**
  - Erro computado na camada de saída
- **Não há saídas desejadas para camadas intermediárias.**
  - Como calcular ou estimar o erro das camadas intermediárias?

# Função de ativação



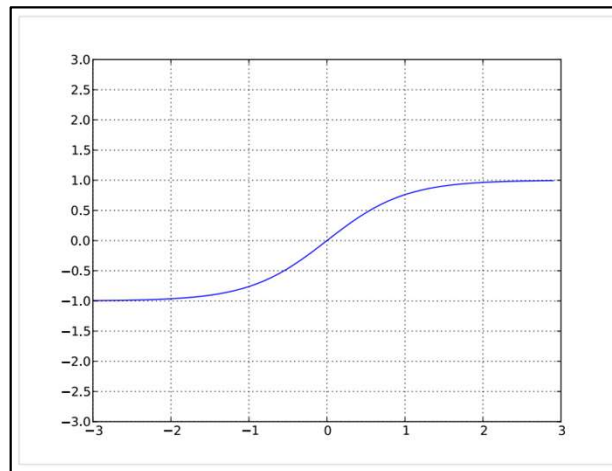
## Sigmoid

$$g(x) = \frac{1}{1 + e^{-x}}$$



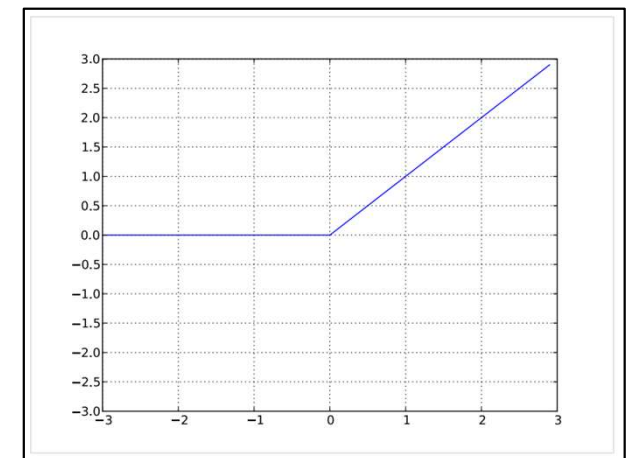
## Tangente Hiperbólica

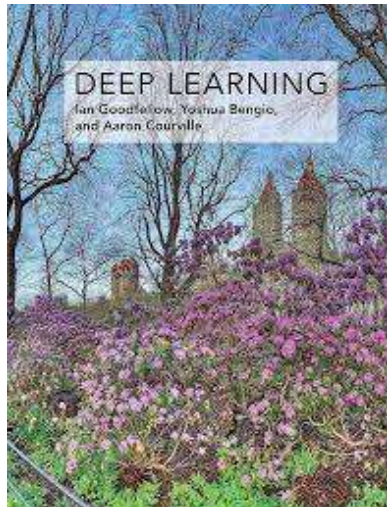
$$g(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



## ReLU

$$g(x) = \text{relu}(x) = \max(0, x)$$



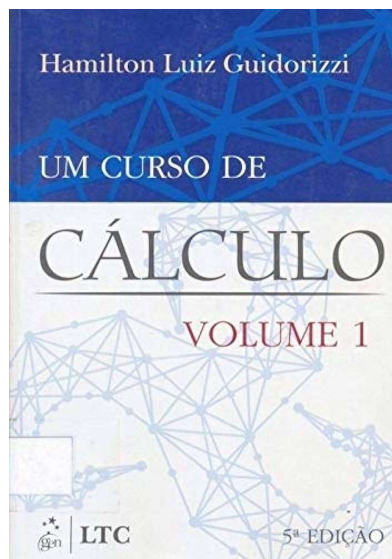


GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. Deep Learning. MIT Press, 2016.

## **Capítulo 6: Deep Feedforward Networks**

Disponível no endereço

<https://www.deeplearningbook.org/>



GUIDORIZZI, Hamilton Luiz. Um curso de cálculo, vol. 1, 5.ed. Rio de Janeiro : LTC, 2013

## **Capítulo 7: Derivadas**

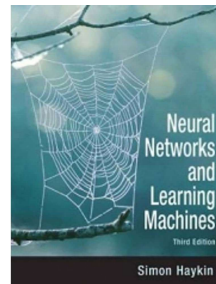


- **Resumo - Breve Histórico**
- **Neurônio**
  - Perceptron
  - Adaline
  - O que é a variável de viés? Qual sua importância?
  - Treinamento em lote (*batch*)
- **Limitações dos neurônios**
  - Como contornar as limitações?
- **Redes Neurais Multicamadas**
  - Introdução
  - Leitura sugerida- Próxima aula
- **Exercício**
- **Bibliografia**

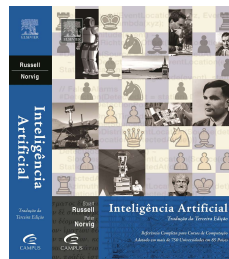


- Implementar em linguagem Python duas classe Neurônio (Perceptron e Adaline) baseado no *template* fornecido no endereço <https://github.com/danillorobertopereira/UNESP-DiscInteligenciaArtificial-Neuronio>
  - Grupo de 2 alunos
  - Utilizar as 3 bases disponíveis no *github*, separando-as em treino e teste (70/30)
  - Para cada umas das bases gerar relatório contendo o erro do treinamento por época
    - Utilizar 3 *learning rates* diferentes (Sugestão:0.0001, 0.01, 0.1).
    - Comparar a convergência e acurácia dos neurônios Perceptron e Adaline
      1. Tabelas com o erro ao longo das épocas
      2. Gráficos de convergência dos conjuntos de treino e testes
        - 🕒 eixo X: épocas, eixo Y: erro do neurônio
  - Data entrega: 21/09/2023
- Critérios de avaliação:
  - Corretude do código: 4.0 pontos
  - Organização do código: 2.0 pontos
  - Apresentação dos relatórios: 4.0 pontos

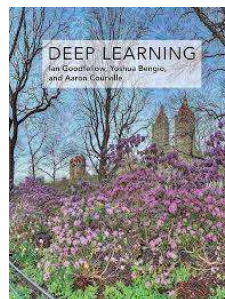
# Referências Bibliográficas



HAYKIN, Simon S. Neural networks and learning. machines. 3. ed. New York: Prentice Hall, 2009



RUSSELL, Stuart J Stuart Jonathan; NORVIG, Peter; WAZLAWICK, Raul Sidnei; SOUZA, Vandenberg D. de. Inteligência artificial. Rio de Janeiro: Elsevier: Campus, 2004.



GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. Deep Learning. MIT Press, 2016