Treinamento de Machine Learning e Deep Learning

Do Básico ao Avançado

Salomão Machado Mafalda[1]

[1]Universidade Federal do Acre
PAVIC

2023

Agenda

Perceptron

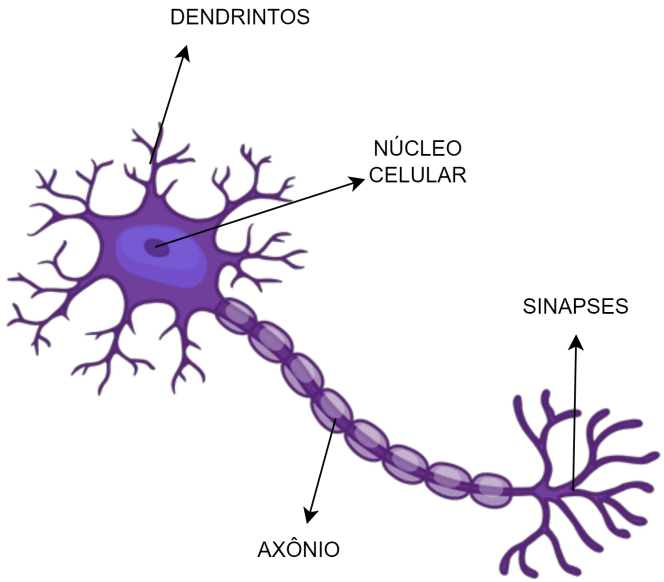


Figure: Neurônio humano

Perceptron

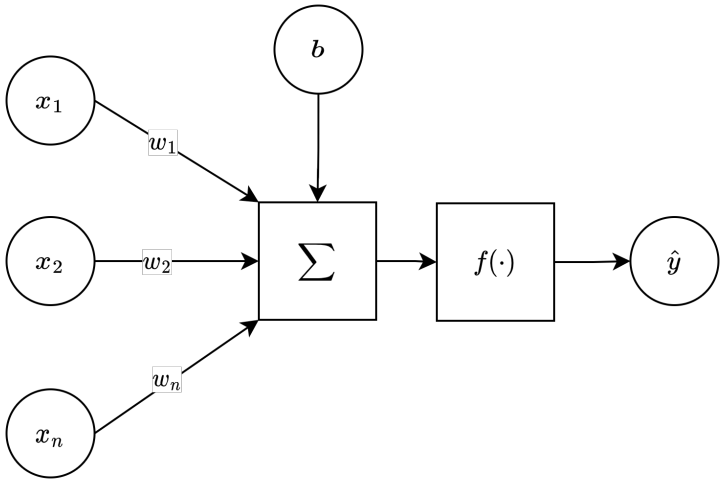
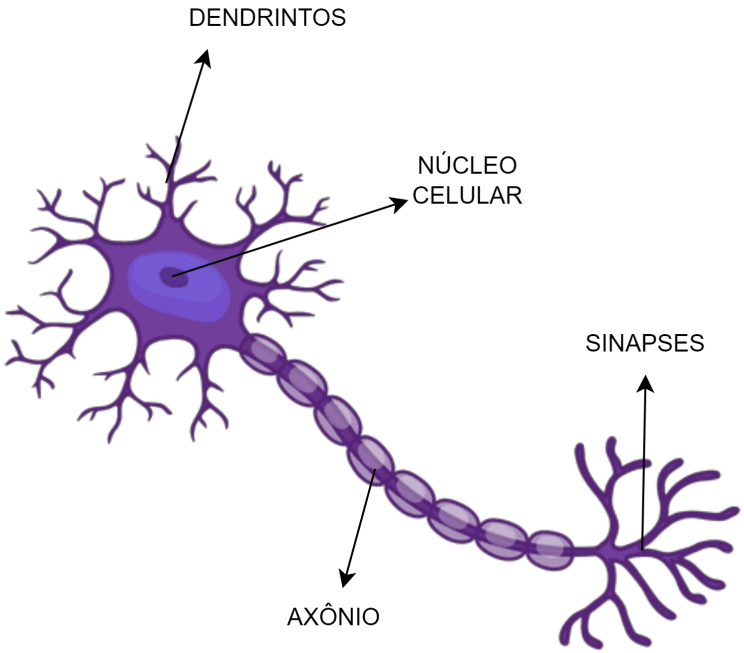
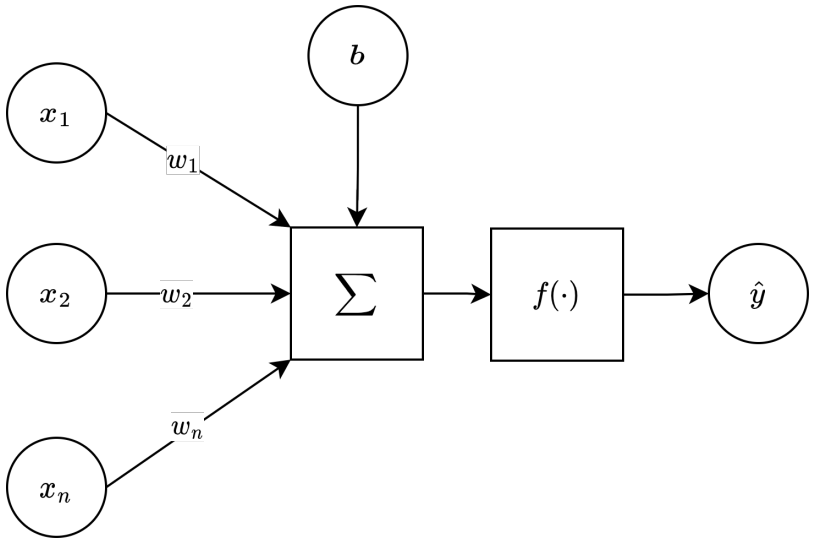


Figure: Neurônio Artificial

Perceptron



(a) Neurônio Humano



(b) Neurônio Artificial

Perceptron

- Modelo mais básico de NN
- Um neurônio
- N entradas, Uma saída ŷ

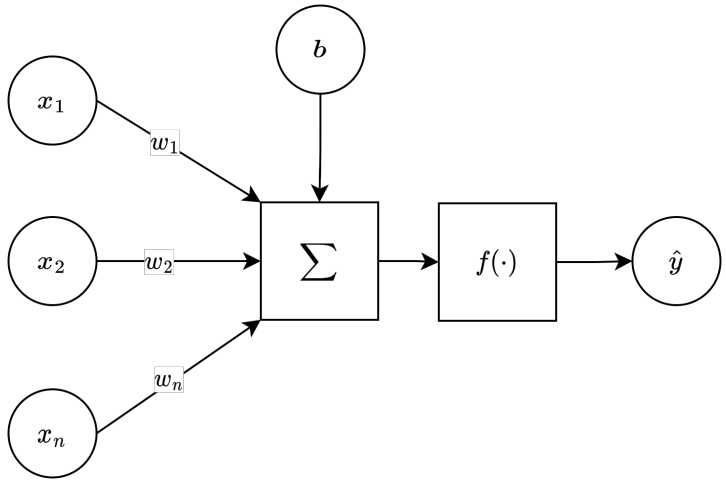
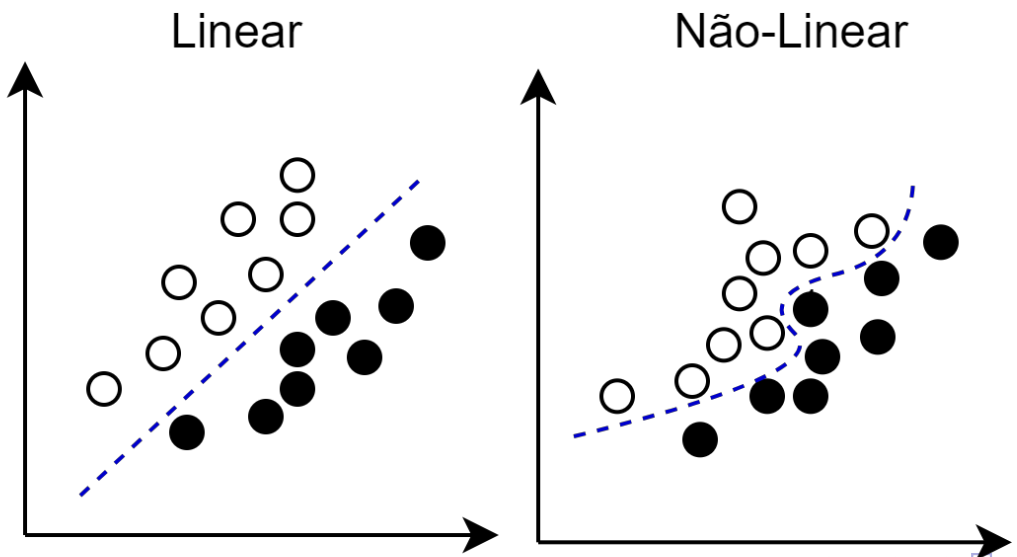


Figure: Neurônio Artificial

$$\hat{y} = f(\sum_i w_i x_i + b)$$

Perceptron

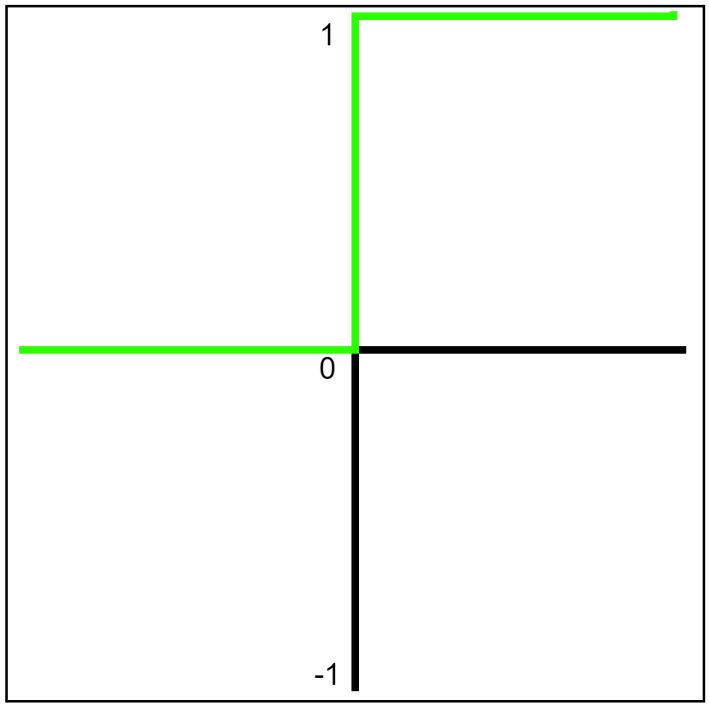
- Modelo mais básico de NN
- Um neurônio
- N entradas, Uma saída $\hat{y}$
- Classificador binário linear
- Pode ser usado para Regressão
- Perceptron Rule
- Aprendizado Online
  - Atualiza os pesos por amostra



Perceptron

Função de ativação do perceptron

- 0 se for negativo
- 1 se maior ou igual a 0



Perceptron

Perceptron Rule

O perceptron atualiza seus pesos utilizando a perceptron rule, não com o gradiente

Atualização dos pesos:

$$w_i = w_i + \lambda * (y_i - \hat{y}_i) * x_i$$

Atualização do *bias*:

$$b_i = b_i + \lambda * (y_i - \hat{y}_i)$$

Observação importante

"Quando a diferença yi - ŷi for 0 então não ocorrerá a atualização dos pesos"

Perceptron

Ponto de partida diferente

Com diferentes pontos de partida, o algoritmo encontra quase a mesma solução, embora com diferentes taxas de convergência.

- Caso 01
- Caso 02

Learning Rate - Taxa de aprendizagem

- LR = 0.01 a velocidade de convergência é muito lenta. Quando o cálculo se torna complicado, uma taxa de aprendizado muito baixa afetará a velocidade do algoritmo, mesmo nunca atingindo o destino.
- LR = 0.5, o algoritmo se aproxima do alvo muito rapidamente após várias iterações. No entanto, o algoritmo falha ao convergir porque o salto é muito grande, fazendo com que ele fique parado no destino.

- Caso 01
- Caso 02

Perceptron Rule

O perceptron atualiza seus pesos utilizando a perceptron rule, não com o gradiente

Atualização dos pesos:

$$w_i = w_i + \lambda * (y_i - \hat{y}_i) * x_i$$

Atualização do *bias*:

$$b_i = b_i + \lambda * (y_i - \hat{y}_i)$$

Highlighting text

$$W = np.random.randn(5, 10) \tag{1}$$
$$X = np.random.randn(3, 10) \tag{2}$$
$$Y = X.dot(W^T) \tag{3}$$
$$f(x) \tag{4}$$

In this slide, some important text will be highlighted because it's important. Please, don't abuse it.

Remark

Sample text

Important theorem

Sample text in red box

Examples

Sample text in green box. The title of the block is "Examples"