

Neurônio Artificial – Especialização Técnica

♦ 1. Entradas (Inputs)

Cada neurônio recebe sinais de entrada, que podem ser variáveis do problema (ex: pixels, palavras, preços, etc.).

Essas entradas são representadas como um vetor:

$$x = [x_1, x_2, \dots, x_n]$$

♦ 2. Pesos (Weights)

Cada entrada possui um peso w_i que determina sua **importância** no resultado.

O neurônio realiza uma **soma ponderada** das entradas:

ini

CopiarEditar

$$z = (x_1 * w_1) + (x_2 * w_2) + \dots + (x_n * w_n) + b$$

- b é o **bias**, que ajusta a ativação para mais ou menos sensibilidade.
-

♦ 3. Função de Ativação (Activation Function)

Ela decide se o neurônio será ativado (isto é, se envia sinal adiante).

É aplicada sobre o valor z :

ini

CopiarEditar

$$y = f(z)$$

As mais comuns:

Função	Fórmula	Uso comum
ReLU	$f(z) = \max(0, z)$	Visão computacional, CNNs
Sigmoid	$f(z) = 1 / (1 + e^{-z})$	Saídas binárias
Tanh	$f(z) = (e^z - e^{-z}) / (e^z + e^{-z})$	Saídas entre -1 e 1

Softmax $f(z_i) = e^{z_i} / \sum e^{z_i}$

Classificação multi-classe (output)

♦ 4. Propagação e Aprendizado

👉 Forward Propagation

Os dados fluem da entrada para a saída passando por todos os neurônios.

👉 Backpropagation

Erro é calculado na saída, e os pesos são ajustados **de trás para frente** para minimizar o erro.

É aqui que entra a otimização, normalmente com algoritmos como **Gradient Descent**.

♦ 5. Neurônio como Função Matemática

Um neurônio pode ser visto como:

CopiarEditar

$$f(x) = \varphi(W \cdot x + b)$$

Onde:

- $W \cdot x$ é o produto escalar entre pesos e entradas
 - φ é a função de ativação
-



Analogia Biológica

Biológico	Artificial
Dendritos	Entradas (x)
Sinapses	Pesos (w)
Soma	Soma ponderada
Axônio	Saída (y)

Potencial de
ação

Ativação (φ)
