

# **Rede Neural com Python puro? Pode isso Arnaldo?**

---

Felipe Israel Corrêa

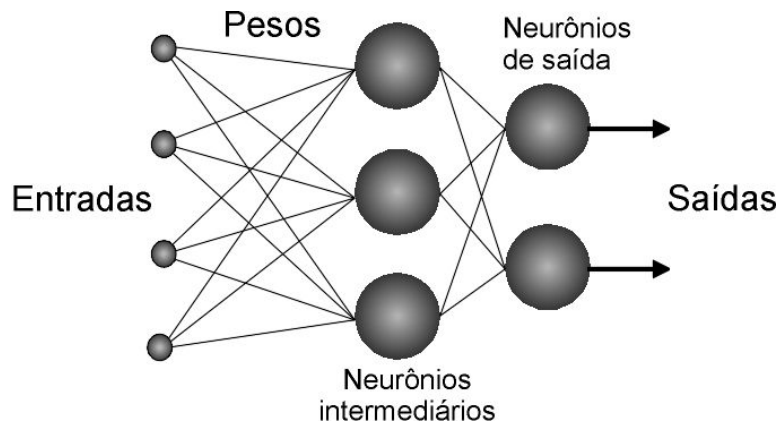
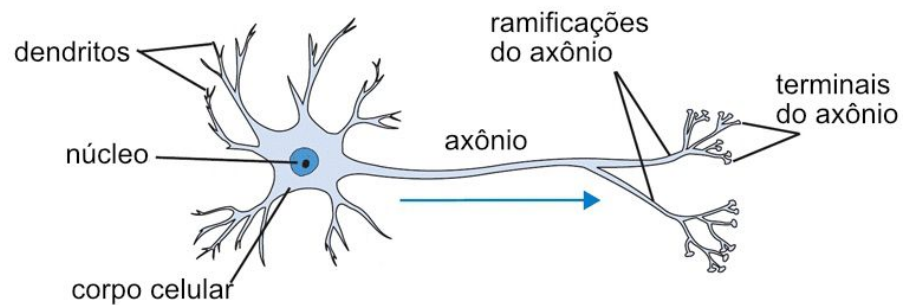
# Quem sou eu

Engenheiro de Computação

@fircorrea

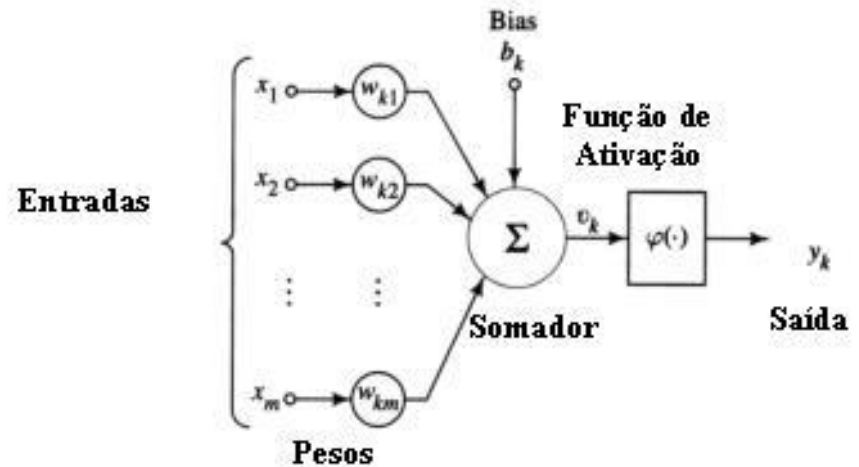
<https://github.com/ficorrea>

# Definição



# Perceptron

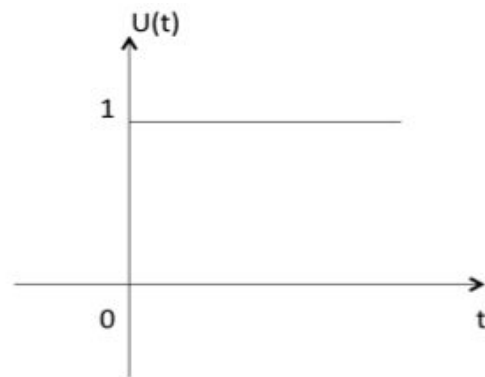
- $x_i$  = entradas da rede;
- $w_{ki}$  = peso sináptico associado à entrada;
- $b_k$  = limiar de ativação (bias);
- $u_k$  = potencial de ativação;
- $g(u)$  = função de ativação;
- $y_k$  = saída da rede;



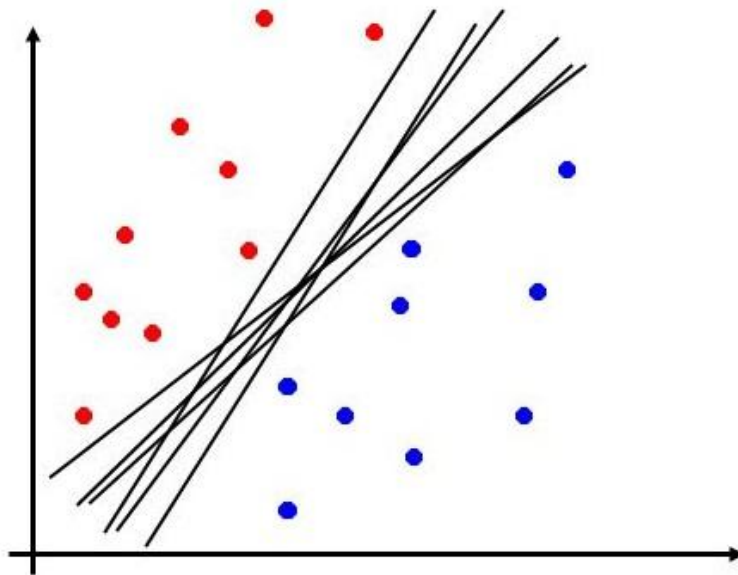
# Perceptron

$$\begin{cases} u = \sum_{i=1}^N x_i w_i - \theta \\ y = g(u) \end{cases}$$

$$u(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases}$$



# Perceptron



# Processo de Aprendizado

- Supervisionado: quando é utilizado um agente externo que indica à rede a resposta desejada para o padrão de entrada;
- Não Supervisionado: quando não existe um agente externo indicando a resposta desejada para os padrões de entrada

# Algoritmo

## Treinamento

Obter o conjunto de amostras de treinamento  $\{x^{(k)}\}$ ;  
Associar o valor desejado  $\{d^{(k)}\}$  para cada amostra;  
Iniciar o vetor de pesos  $\{w\}$  com valores aleatórios pequenos;

Especificar a taxa de aprendizagem  $\{\eta\}$ ;

Iniciar o contador de épocas (época = 0);

Repetir instruções até que o erro inexista:

    Inicializa erro  $\leftarrow$  False;

    Para todas as amostras de treinamento faça:

$$u = w^T \cdot x^{(k)}$$

$$y = g(u)$$

    Se  $y \neq d$  faça:

$$w \leftarrow w + \eta * (d^{(k)} - y) * x^{(k)}$$

        erro  $\leftarrow$  True

    época  $\leftarrow$  época + 1

Até que erro = False

## Operação (Validação)

Obter conjunto de amostras para classificação;

Carregar o vetor de pesos  $\{w\}$ , ajustado no treinamento;

Para cada amostra  $\{x\}$  faça:

$$u = w^T * x$$

$$y = g(u)$$

Verificar saída:

    Se  $y = 0$ ,  $x \in$  à classe A

    Se  $y = 1$ ,  $x \in$  à classe B



# Referências

- Referências:
  - <https://www.monolitonimbus.com.br/perceptron-redes-neurais/>
  - <http://conteudo.icmc.usp.br/pessoas/andre/research/neural/>
  - Slides Inteligência Computacional Prof. Maurílio J. Inácio
- Livros:
  - Data Science do Zero
  - Redes Neurais Artificiais para Engenharias e Ciências Aplicadas
  - Inteligência Artificial – Noções Gerais
- Dataset:
  - <https://archive.ics.uci.edu/ml/datasets/User+Knowledge+Modeling>
- Material:
  - <https://github.com/ficorrea/justpython>