

# Redes Neurais



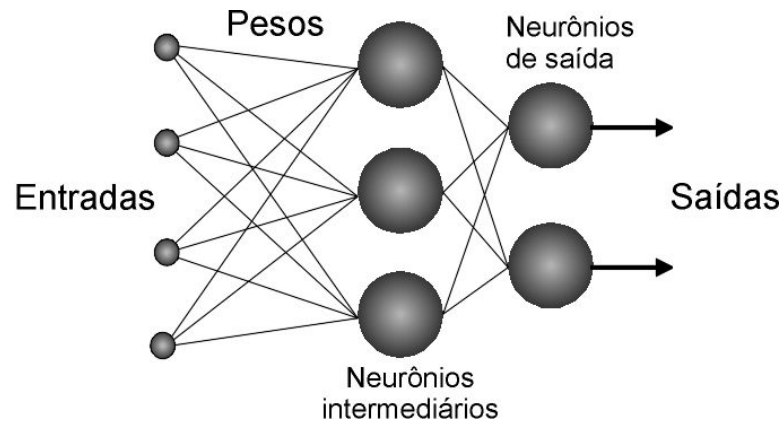
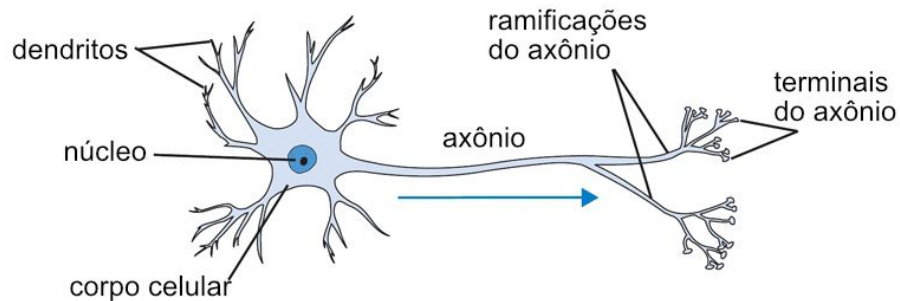
Implementação de Perceptron Simples com Python e  
bibliotecas padrão

Felipe Corrêa

# Redes Neurais

O que são:

- Trabalho proposto por McCulloch e Pitts em 1943;
- Baseada em um neurônio biológico;



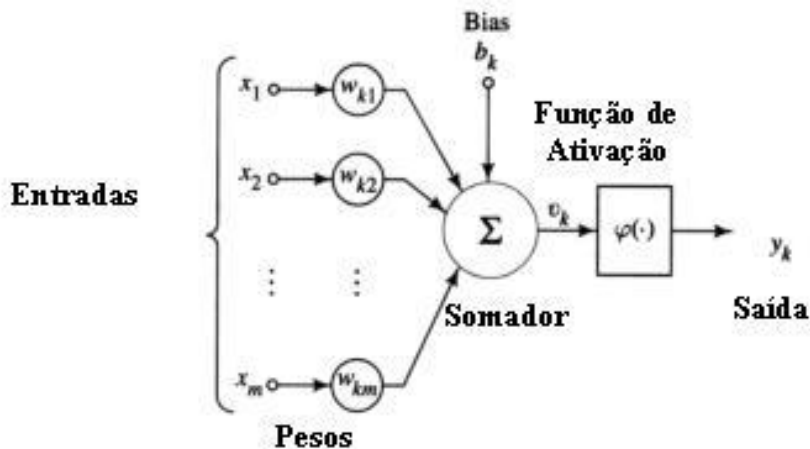
# Redes Neurais

## Definições:

- Lippmann (1997): Redes Neurais Artificiais são sistemas que podem adquirir, armazenar e utilizar conhecimentos experimentais que podem alcançar boa performance, devido à sua densa interconexão entre os nós da rede;
- Conhecidas também por modelos connexionistas, modelos de processamento paralelo distribuído e sistemas neuromórficos;

# Perceptron Simples

- Proposto por Frank Rosenblatt (1958);
- Forma mais simples de uma rede neural, pois possui uma única camada;
- Resolvem apenas problemas linearmente separáveis;
- Utilizada, em geral, para classificação de padrões;



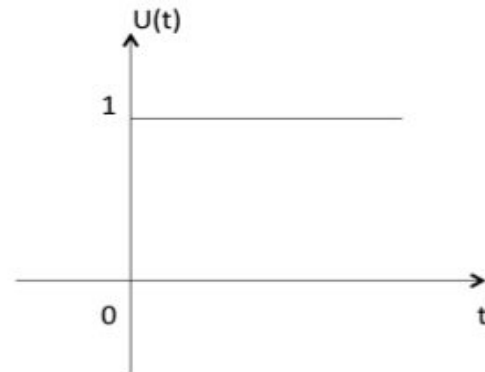
# Modelo Matemático

- $x_i$  = entradas da rede;
- $w_i$  = peso sináptico associado à entrada;
- $\theta$  = limiar de ativação (bias);
- $u$  = potencial de ativação;
- $g(u)$  = função de ativação;
- $y$  = saída da rede;

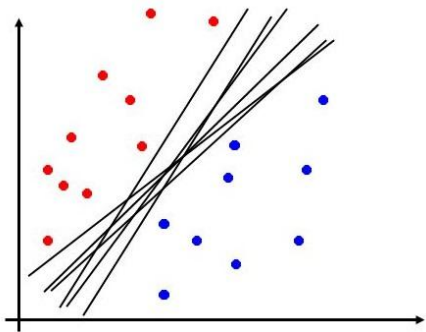
$$\begin{cases} u = \sum_{i=1}^N x_i w_i - \theta \\ y = g(u) \end{cases}$$

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

Função Degrau



# Algumas definições



- O Perceptron Simples atua, traçando retas entre as classes, até conseguir valores de peso que ‘separem’ todos os pontos;

- O bias ( $\theta$ ) serve para aumentar os graus de liberdade, permitindo uma melhor adaptação por parte da rede neural ao conhecimento à ela fornecido;
- Taxa de aprendizagem, um valor entre 0 e 1, diz o quão rápido a rede converge;

# 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

# 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



# Referências e dicas de leitura

- 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://github.com/cuekoo/Binary-classification-dataset>