

# Redes Neurais Artificiais *for Dummies*: Praticando um pouco de L<sup>A</sup>T<sub>E</sub>X no CI/UFPB

Gustavo Oliveira, *DCC/UFPB* e Rafael Magalhães, *DCX/UFPB*

**Abstract**—Este artigo é um ensaio minimalista que tem o propósito de servir como texto-base para o mini-curso: *Conhecendo o L<sup>A</sup>T<sub>E</sub>X: usos, dicas e práticas*. Alguns tópicos sobre redes neurais artificiais são apresentados de maneira genérica. Com diversos elementos, mostramos como a linguagem L<sup>A</sup>T<sub>E</sub>X é poderosa para manipular não apenas textos simples, mas também equações, figuras, tabelas, referências cruzadas e citações. Conclui-se que a ferramenta é muito eficaz para a produção de praticamente qualquer tipo de texto acadêmico.

**Index Terms**—Redes neurais artificiais, funções de ativação, Python, L<sup>A</sup>T<sub>E</sub>X.

## 1 INTRODUÇÃO

O Cérebro humano contém em torno de  $10^{11}$  neurônios. Cada um deles processa informações e se conecta com outros milhares de neurônios continuamente ou de modo paralelo. A estrutura individual de suas conexões e o comportamento conjunto destes nós naturais formam a base para o estudo das *redes neurais artificiais* - RNAs. A Figura 1 contém uma ilustração simplificada para um neurônio biológico, bem como seu modelo associado. Este modelo de neurônio artificial foi proposto por McCulloch e Pitts [1] em 1943 e ficou conhecido como MCP.

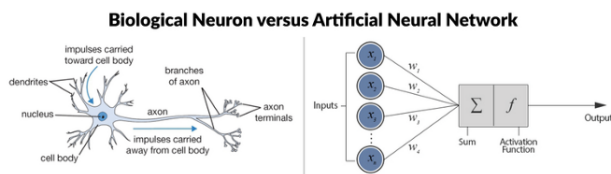


Fig. 1: Esquema de neurônio biológico e modelo MCP de McCulloch e Pitts.

### 1.1 Aprendizagem: o perceptron

RNAs possuem a capacidade de aprender por exemplos e tomar decisões sobre aquilo que aprendem. Para tanto, necessitam de um conjunto de procedimentos bem definidos que adaptam seus parâmetros, ou seja, um **algoritmo de aprendizagem**. Dessa maneira, define-se que

aprendizagem é o processo pelo qual os parâmetros de uma rede neural são ajustados através de uma forma continuada de estímulo pelo ambiente no qual a rede está operando, sendo o tipo específico de aprendizagem realizada definido pela maneira particular como ocorrem os ajustes realizados nos parâmetros.

Desde a década de 1940, muito se aperfeiçoou na compreensão das redes artificiais. Em 1958, Frank Rosenblatt introduziu um novo modelo para o neurônio biológico formado por duas unidades básicas: os *nós MCP* e uma regra de aprendizado.

- G.Oliveira é professor do Departamento de Computação Científica do Centro de Informática da Universidade Federal da Paraíba.  
E-mail: gustavo.oliveira@ci.ufpb.br
- G. Oliveira e R. Magalhães agradecem a sua atenção.

Manuscript received July 25, 2019; revised August 26, 2019.

Este modelo foi denominado *perceptron*, o qual é utilizado até hoje. Então, de meados da década de 1970 para cá, grande progresso foi atingido com aprendizagem de máquina. Vejamos, por exemplo, a linha do tempo mostrada na Figura 2.

## 2 MÉTODOS

### 2.1 Limiar de excitação

Um neurônio biológico dispara quando a soma dos impulsos que ele recebe ultrapassa seu limiar de excitação (*threshold*)  $\theta$ . Matematicamente, podemos modelar um neurônio  $k$  pelo seguinte par de equações:

$$v_k = b_k + \sum_{j=1}^m w_{kj} x_j \quad (1)$$

e

$$y_k = \phi(v_k), \quad (2)$$

em que

- $x_1, x_2, \dots, x_m$  são **sinais de entrada**;
- $w_{k1}, w_{k2}, \dots, w_{km}$  são **pesos sinápticos**;
- $u_k$  é o **adicionador** ou **combinador linear**;
- $b_k$  é o **bias**<sup>1</sup>;
- $v_k$  é o **potencial de ativação**;
- $\phi$  é a **função de ativação**;
- $y_k$  é a **saída**.

Quando  $v_k > \theta$ , o neurônio é ativado e produz uma saída. No caso do nó MCP,  $y_k \in \{0, 1\}$ .

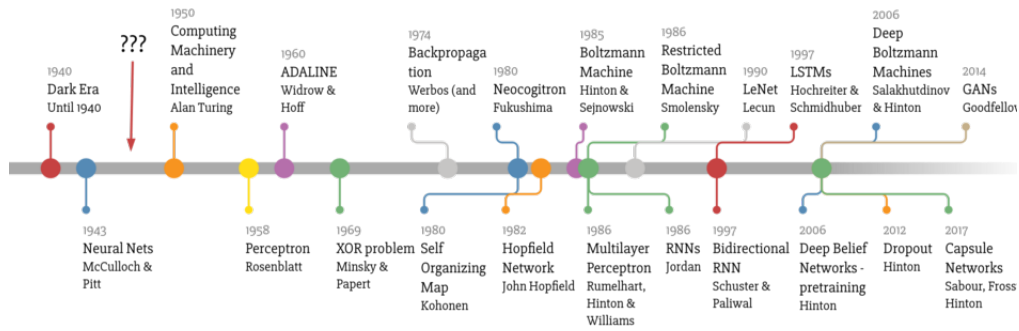
### 2.2 Funções de ativação

Uma função de ativação define a saída do nó. A mais básica é a Heaviside, também conhecida como *função de limiar*, definida na Eq. (3).

$$\phi(v_k) = \begin{cases} 1, & \text{se } v_k \geq 0 \\ 0, & \text{se } v_k < 0. \end{cases} \quad (3)$$

Outras funções de ativação são também utilizadas. A seguir, veremos mais dois exemplos.

1. O *bias* é um parâmetro externo de entrada  $x_0$  e peso  $w_{k0} = b_k$  que tem o efeito de aumentar (se positivo) ou diminuir (se negativo) a entrada líquida da função de ativação.

Fig. 2: Linha do tempo: *deep learning*.

### 2.2.1 Função Relu

A função Relu (*rectified linear unit*) é definida como

$$\phi(v_k) = \max\{0, v_k\}.$$

### 2.2.2 Função sigmoide

A função sigmoide é definida como

$$\phi(v_k) = \frac{1}{1 + \exp(-av_k)},$$

onde  $a$  é um parâmetro de suavização.

## 3 RESULTADOS

### 3.1 Códigos

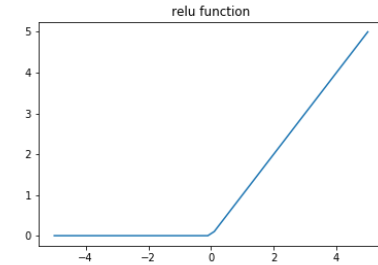
Podemos incluir um bloco de código que mostra como gerar gráficos em Python para as funções de ativação anteriores.

```
1 from matplotlib.pyplot \
2 import figure, plot, title, savefig
3 from numpy import linspace
4 from math import exp
5
6 def relu(x):
7     return max(0.0, x)
8
9 def sigmoid(x, a):
10    return 1.0 / (1.0 + exp(-a*x))
11
12 # plot
13 x = [v for v in linspace(-5, 5, 50)]
14 r = [relu(v) for v in x]
15 s05 = [sigmoid(v, 0.5) for v in x]
16 s1 = [sigmoid(v, 1) for v in x]
17 s2 = [sigmoid(v, 1.5) for v in x]
18
19 figure(figsize=(6, 4))
20 plot(x, r)
21 title('relu function')
22 savefig('../figs/relu.png')
23 figure(figsize=(6, 4))
24 plot(x, s05, 'r', x, s1, 'g', x, s2, 'b')
25 title('sigmoid function')
26 savefig('../figs/sigmoid.png')
```

Listing 1: Código Python.

### 3.2 Plotagens

O código incluído na Subseção 3.1 gera os gráficos mostrados na Figura 3.



(a) Relu.

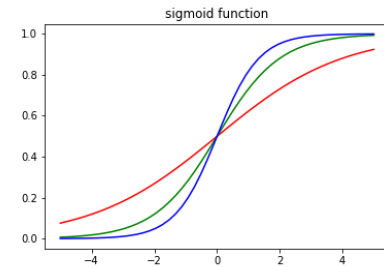
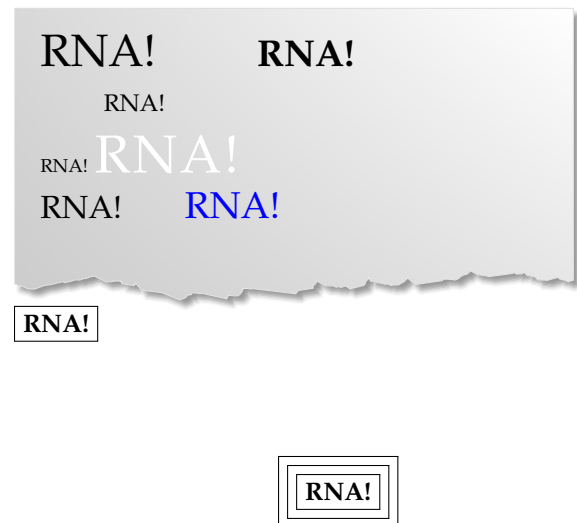
(b) Sig. —  $a = 0.5$  —  $a = 1$  —  $a = 1.5$ .

Fig. 3: Exemplos de funções de ativação.



## 4 CONCLUSÃO

Neste texto, fizemos uma breve apresentação sobre o conceito de redes neurais artificiais. Introduzimos na seção 1. Incluímos uma metodologia na 2 e resultados na seção 3. Estudos têm avançado

para muitos campos. Para encerrar, incluímos um resumo sobre tipos de arquiteturas de RNAs na Tabela 1.

no. de camadas	conexão nodal	conectividade
única	feedforward (ou acíclica)	fracamente conectada
múltiplas	feedback (ou cíclica)	completamente conectada

TABLE 1: Tabela-resumo dos tipos de arquitetura de RNAs.

Saiba mais sobre RNAs em [2], [3] e [4].

## APPENDIX A

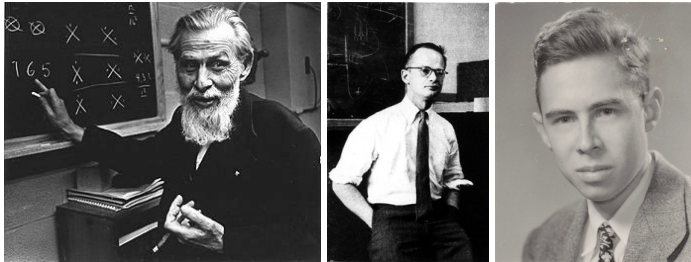
### ALGUMAS EQUAÇÕES PARA PRATICAR A ESCRITA

Vide arquivo `mc-desafio.pdf`...

## APPENDIX B

### PERSONALIDADES NO ESTUDO DE RNAs

- 1) Warren S. McCulloch (★ 1868 - † 1969)
- 2) Walter Pitts (★ 1923 - † 1969)
- 3) Frank Rosenblatt (★ 1928 - † 1971)



## ACKNOWLEDGMENTS

G.O. agradece a oportunidade de ministrar este mini-curso.

## REFERENCES

- [1] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.
- [2] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.
- [3] A. de Pádua Braga, A. C. P. de Leon Ferreira, and T. B. Ludermir, *Redes neurais artificiais: teoria e aplicações*. LTC Editora Rio de Janeiro, Brazil, 2007.
- [4] Z. L. Kovács, *Redes neurais artificiais*. Editora Livraria da Física, 2002.

**Gustavo Oliveira** Prof. do DCC/CI/UFPB e pesquisador do LaMEP.  
Visite-nos em: [lamep.ci.ufpb.br](http://lamep.ci.ufpb.br).

**Rafael Magalhães** Prof. do DCX/CCAE/UFPB e pesquisador no LaMEP/UFPB.