

# Inteligência Artificial

## Redes Neurais

Marlon Henry Schweigert

Semestre 2018-1

Primeiro neurônio artificial criado em 1943 utilizando circuitos elétricos [7].

Snark foi o primeiro neuro computador (1953) a processar utilizando redes neurais com ajuste de pesos. Porém, nunca processou uma função interessante [7].

1956 no Dartmouth College criou o neurônio Perceptron, programável [7].

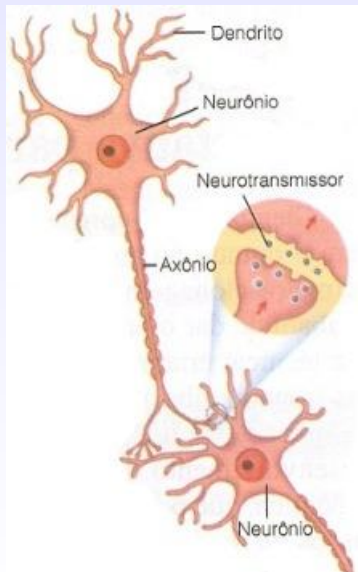
# Motivação



Redes Neurais tem o embasamento teórico de processamento do sistema nervoso biológico.

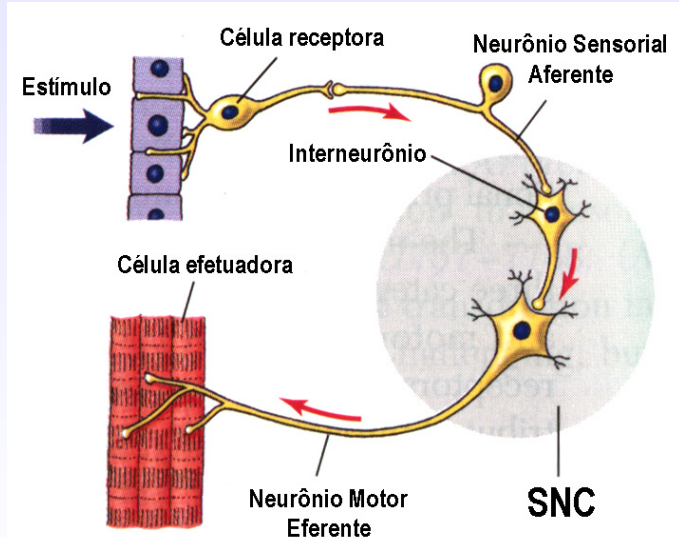
Células especificadas em conduzir energia podendo alterar as características dos impulsos conduzidos[4].

# Neurônio

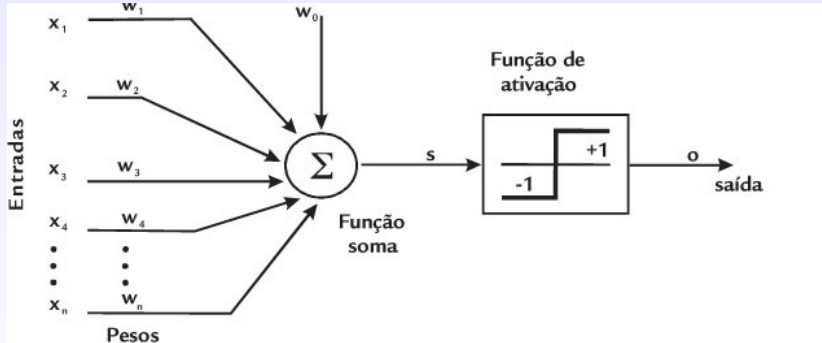




# Neurônio



# Modelo Matemático



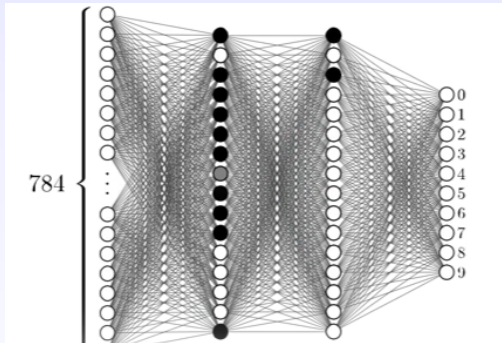
Neurônio Perceptron[5].

- Entradas  $w_n$ : Entrada de dados do neurônio, seja um vetor de dados ou outros neurônios (Rede multicamada).
- Entrada estável (bias): Constante de entrada para ter um valor padrão em caso de entrada nula. No neurônio Perceptron torna-se padrão a constante  $-1$ .
- Função soma: Função que agregará todas as entradas em um valor único  $s$ .
- Função de ativação: Função que tratará a entrada  $u$  a fim de retornar uma resposta para outro neurônio ou retorno ao sistema.

- Função Degrau:  $g(u) = \begin{cases} 1 & \text{se } (u \geq 0) \\ 0 & \text{se } (u < 0) \end{cases}$
- Função Bipolar:  $g(u) = \begin{cases} 1 & \text{se } (u > 0) \\ 0 & \text{se } (u = 0) \\ -1 & \text{se } (u < 0) \end{cases}$  ou  $\begin{cases} 1 & \text{se } (u \geq 0) \\ -1 & \text{se } (u < 0) \end{cases}$

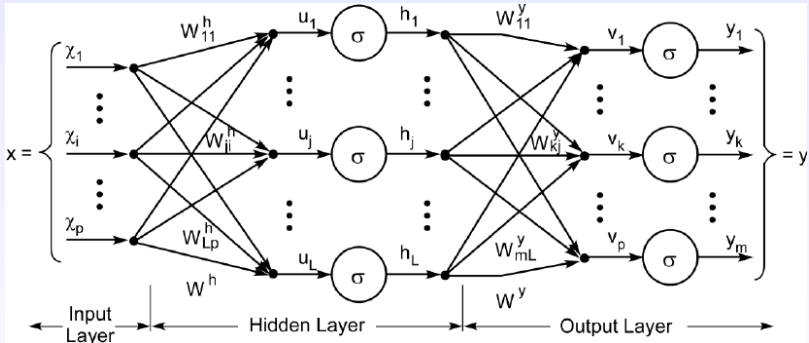
- Função Logística:  $g(u) = \frac{1}{1+e^{-bu}}$ , onde  $b$  é constante
- Função Tangente Hiperbólico:  $g(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$





Modelo para reconhecimento de caracteres numéricas[1].

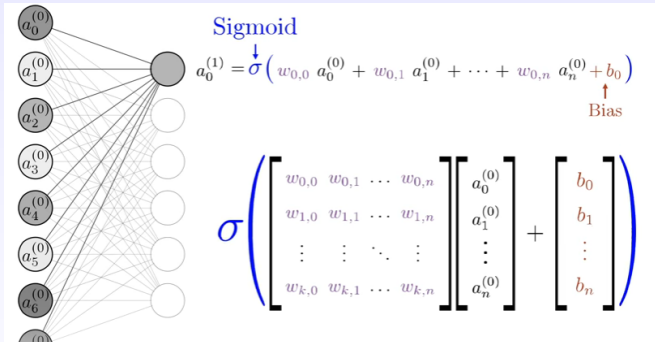
# Rede Neural de múltiplas camadas



Múltiplas camadas em uma rede neural[2].



# Modelagem Matricial



Produzindo sinapses com matrizes[1].

# Como treinar?

Entradas de 28 por 28 pixels, em escala de cinza:  
 $28 \times 28(\text{pesos}) + 28(\text{bias}) = 812$  variáveis por camada.  
Como encontrar o valor de cada peso? [3]

- 1 Regressão Linear
- 2 Regressões Polinomiais
- 3 Simulated Annealing
- 4 Algoritmos Genéticos / Culturais
- 5 Backpropagation

Como encontrar o erro?

Como ajustar os erros das camadas anteriores utilizando este erro?

# Propagando correções aos pesos

$Err_n = V_e - V_r$ ,  $V_e \rightarrow$  Vetor Esperado,  $V_r \rightarrow$  Vetor Real

Erro de cada nó:  $Err_h = \frac{w_n}{\sum w_i} * \sum Err_{h+1}$  [6]

O erro é utilizado para atualizar o peso da sinapse com relação a um valor aleatório (opcional) e o Learning rate.

# Por onde começar?

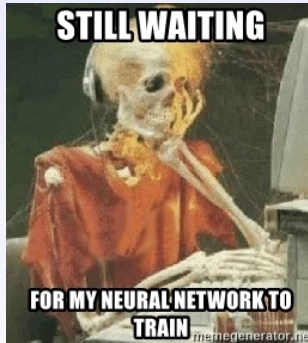
Implementação manual? Talvez

Frameworks? Tensorflow, Karas, etc

Aplicações? Data Mining, Reconstrução de imagens (HyperGAN),

Visão computacional, etc.

# Por onde começar?








GPU: Processo de encontrar os pesos de uma rede neural é extremamente custoso.



# Onde utilizar?

- 1 IA para jogos.
- 2 Veículos autônomos.
- 3 Reconhecimento de padrões.
- 4 Classificação e recomendação.
- 5 Reconstrução de dados.
- 6 Avaliação de qualidade.
- 7 Próteses cervo-nervosas.
- 8 Gerar imagens de pokémon ou gatinhos
- 9 Muito mais!

- 1 [https://github.com/Schweigert/Pokemon\\_GAN](https://github.com/Schweigert/Pokemon_GAN) (MUITA GPU)
- 2 <https://playground.tensorflow.org>



-  But what \*is\* a Neural Network? | Chapter 1, deep learning - YouTube, Jun 2018. [Online; accessed 16. Jun. 2018].
-  Figure 3. The structure of a multilayer perceptron neural network., Jun 2018. [Online; accessed 18. Jun. 2018].
-  How neural networks are trained, May 2018. [Online; accessed 18. Jun. 2018].
-  F. H. Netter. *Netter Atlas De Anatomia Humana*. Elsevier Editora Ltda., 2015.
-  Germano Mendes Rosa and José Aurélio Medeiros da Luz. Mix grinding simulation by artificial neural network. *Rem: Revista Escola de Minas*, 65(2):247–256, Jun 2012.

-  [Shiffman](#). 10.14: Neural Networks: Backpropagation Part 1 - The Nature of Code, Jun 2018. [Online; accessed 19. Jun. 2018].
-  [N. Yadav, A. Yadav, and M. Kumar](#). *An Introduction to Neural Network Methods for Differential Equations*. Springer Netherlands, 2015.