

Escola Superior de Tecnologia  
Universidade do Estado do Amazonas

4 de novembro de 2018

# *Machine Learning*

*Hands on com Python*

Samsung Ocean – Novembro/2018

**Elloá B. Guedes da Costa**

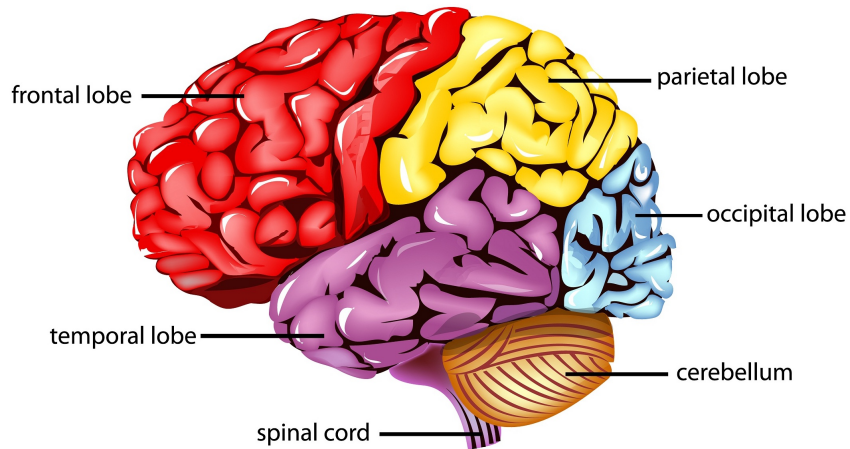
[www.elloaguedes.com](http://www.elloaguedes.com)  
[ebgcosta@uea.edu.br](mailto:ebgcosta@uea.edu.br)

# *Machine Learning – Hands On com Python*

## Outline

- 1 Cérebro Humano

## Cérebro Humano



## Cérebro Humano

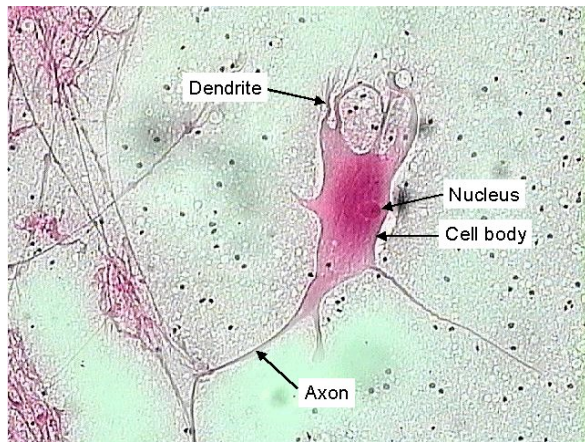
- 4 grandes áreas
- Pesa em torno de 1,5 kg
- Poderoso e complexo
  
- Lida com ruído, dados inconsistentes, etc
- Processa dados de alta ordem dimensional muito rapidamente
- Robustez: pouca perda de performance

## Neurônios Biológicos

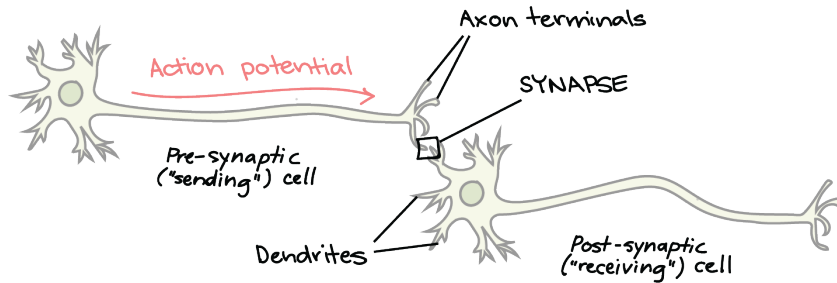
- Composto de elementos estruturalmente **simples** e **fáceis** de entender
- **Neurônios**: Células nervosas
- Cerca de 100 bilhões de neurônios ( $10^{11}$  neurônios)

## *Machine Learning – Hands On com Python*

# Neurônios Biológicos



## Neurônios Biológicos – Sinapse



# *Machine Learning – Hands On com Python*

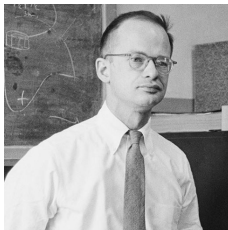
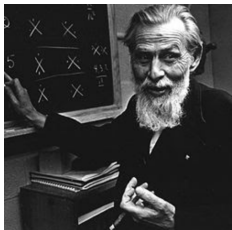
## Outline

### 2 Neurônios Artificiais



## *Machine Learning – Hands On com Python*

# Neurônio Artificial



- Modelo McCulloch & Pitts, 1943
- McCulloch: Psiquiatra e Neuroanatomista
- Pitts: Matemático
- Simplificação do que se conhecia sobre o neurônio biológico

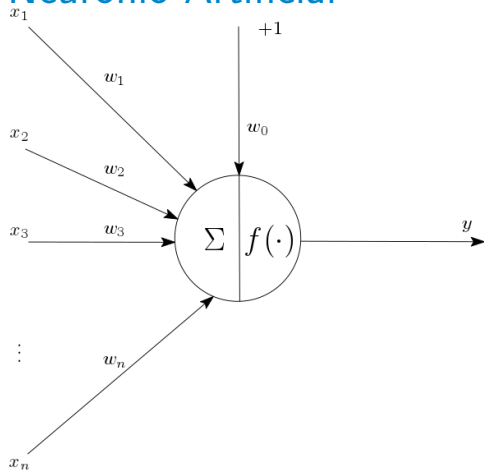
## Neurônio Artificial

- Modelo McCulloch e Pitts (MCP): 1943
- $n$  terminais de entrada  $(x_1, x_2, \dots, x_n)$
- Saída:  $y$
- Pesos acoplados:  $w_1, w_2, \dots, w_n$  (positivos ou negativos)
- Efeito de uma sinapse:  $x_i \cdot w_i$
- Peso: “grau” da contribuição
- Disparo: soma ponderada ultrapassa limiar de excitação (*threshold*)

## Neurônio Artificial

- Melhorias posteriores levaram à inclusão de um viés (*bias*)
- Rosenblatt, 1957
- Perceptron de Rosenblatt
- Resolução de problemas linearmente separáveis

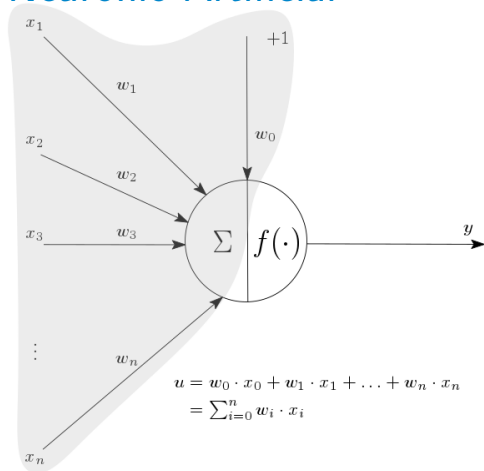
## Neurônio Artificial



## Neurônio Artificial

- $n$  terminais de entrada  $(x_0, x_1, \dots, x_n)$
- Pesos associados:  $w_1, w_2, \dots, w_n$  (positivos ou negativos)
- Efeito de uma sinapse:  $x_i \cdot w_i$
- Peso: “grau” da contribuição

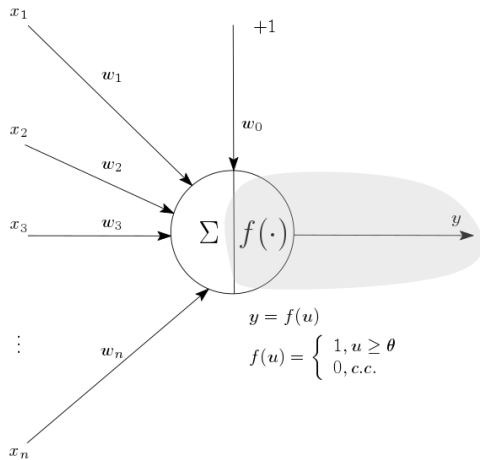
## Neurônio Artificial



## Neurônio Artificial

- Produção da soma ponderada das entradas sujeitas aos pesos
- $u = \sum_{i=0}^n x_i \cdot w_i$
- Aplicação da **função de ativação**
- Avalia se a soma ultrapassa certo limiar ( $\theta$ )
- Saída: 1 (houve disparo), 0 (não houve disparo)

## Neurônio Artificial

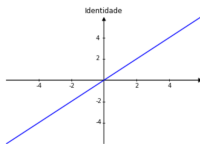




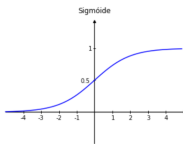
# Neurônio Artificial – Função de Ativação

- Responsável por determinar o limiar de disparo
- Diferentes tipos de função de ativação

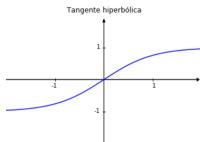
(a) Função identidade.



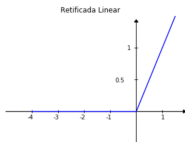
(b) Função sigmóide.



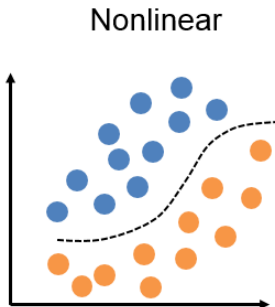
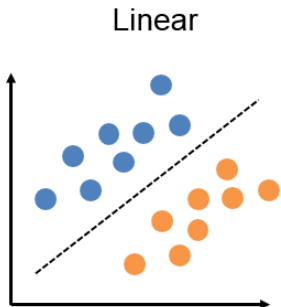
(c) Função tangente hiperbólica.



(d) Função retificada linear.



## Neurônio Artificial – Limitação



# *Machine Learning – Hands On com Python*

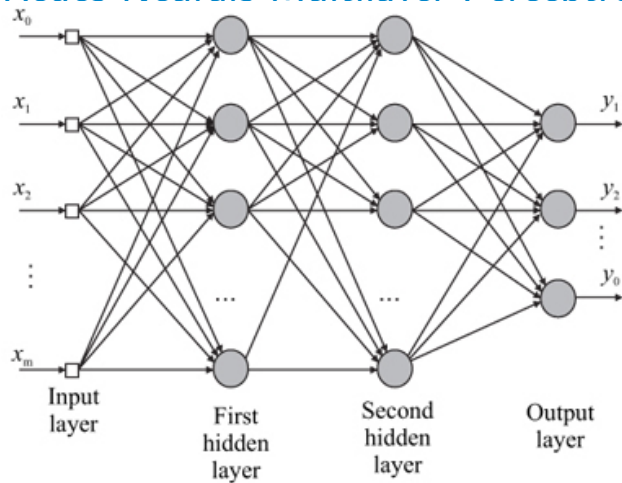
## Outline

### 3 Redes Neurais Artificiais

## Redes Neurais Multilayer Perceptron (MLP)

- **Redes MLP:** redes neurais de múltiplas camadas compostas por neurônios com funções de ativação sigmoidais nas camadas intermediárias
- 1986: Rumelhart, Hinton, e McClelland
- Cybenko 1989: Uma rede MLP com uma camada intermediária pode aproximar qualquer função contínua
- Duas camadas intermediárias permitem a aproximação de qualquer função

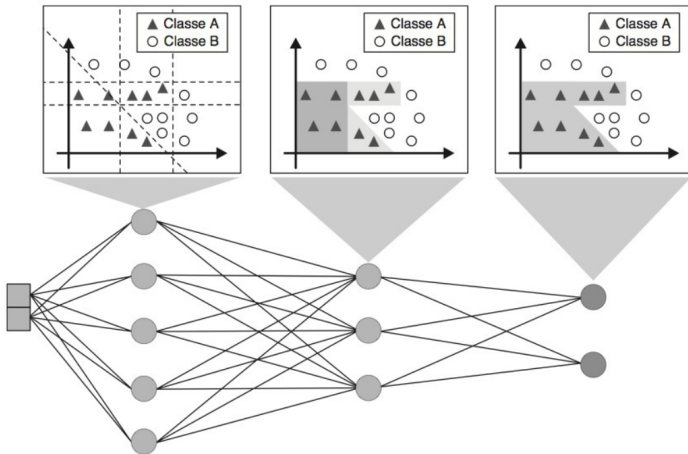
## Redes Neurais Multilayer Perceptron (MLP)



## Redes Neurais Multilayer Perceptron (MLP)

- Fluxo da informação: camada de entrada  $\rightarrow \dots \rightarrow$  camada de saída
- Algoritmo de aprendizado supervisionado: *Backpropagation*
- Resolução de problemas não-linearmente separáveis
- Camadas intermediárias: transformações sucessivas que tornam a entrada linearmente separável
- Problema torna-se tratável para camada de saída

## Redes Neurais Multilayer Perceptron (MLP)



## Redes Neurais Multilayer Perceptron (MLP)

- São vários os **desafios** do projeto de Redes Neurais
- Determinação da arquitetura
  - Número de camadas escondidas
  - Quantidade de neurônios nas camadas escondidas
  - Escolha da função de ativação
- Taxa de aprendizado
- Otimizador para o cálculo do gradiente descendente no *Backpropagation*
- Diversos outros



*Machine Learning – Hands On com Python*

## Outline

- 4 Reconhecendo Dígitos Manuscritos com Redes Neurais Artificiais

## Reconhecendo Dígitos Manuscritos



## Reconhecendo Dígitos Manuscritos

- Base de dados **MNIST**
- *Modified National Institute of Standards and Technology database*
- Oriundos de alunos de ensino médio e de profissionais do Censo dos Estados Unidos
- Reconhecimento automático do CEP em cartas (LeCun, 1989)
- 70.000 dígitos manuscritos
- Previamente particionada (85/15):
  - 60.000 exemplos de treinamento
  - 10.000 exemplos de teste

## *Machine Learning – Hands On com Python*

# Reconhecendo Dígitos Manuscritos

- Base de dados **MNIST**
- Imagens com  $28 \times 28$  pixels em dois níveis (preto e branco)
- **Objetivo:** Tarefa de classificação multiclasse
- **Nosso desafio:** Propor redes neurais artificiais para este problema

Escola Superior de Tecnologia  
Universidade do Estado do Amazonas

4 de novembro de 2018

# *Machine Learning*

*Hands on com Python*

Samsung Ocean – Novembro/2018

**Elloá B. Guedes da Costa**

[www.elloaguedes.com](http://www.elloaguedes.com)  
[ebgcosta@uea.edu.br](mailto:ebgcosta@uea.edu.br)