Análise e Desenvolvimento de Sistemas

Fatec Indaiatuba
Dr. Archimedes Lammoglia

Centro Paula Souza

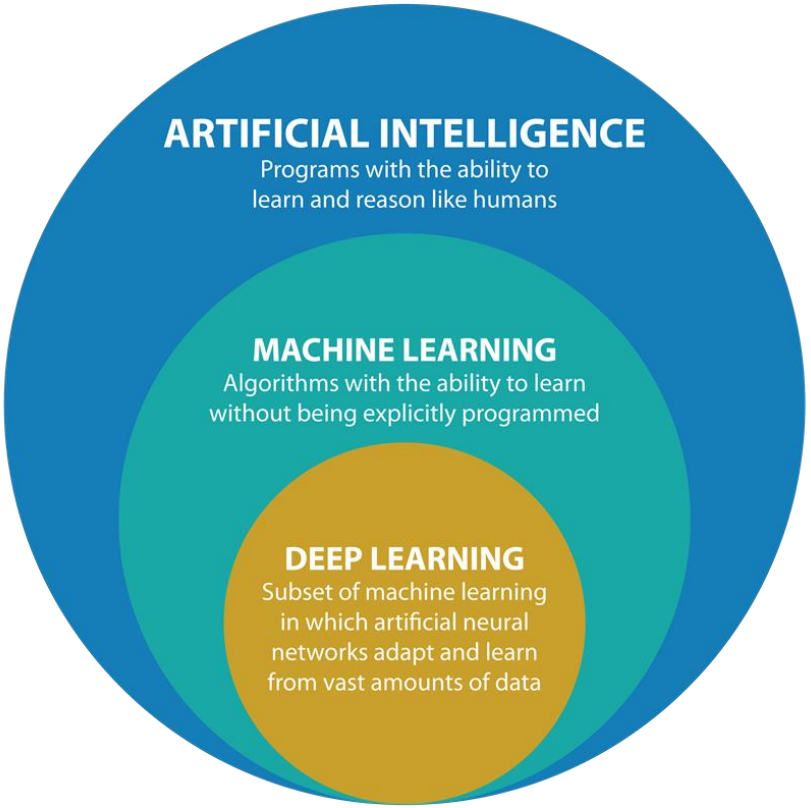
GOVERNO DO ESTADO SÃO PAULO
Secretaria de Desenvolvimento Econômico, Ciência, Tecnologia e Inovação

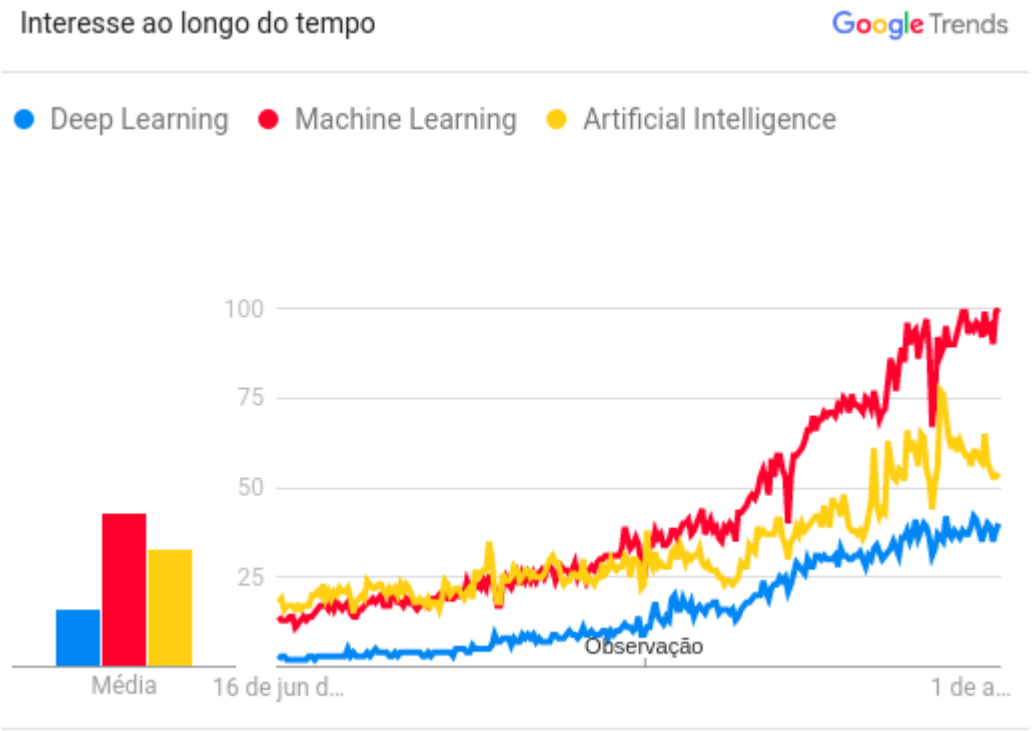
Deep Learning

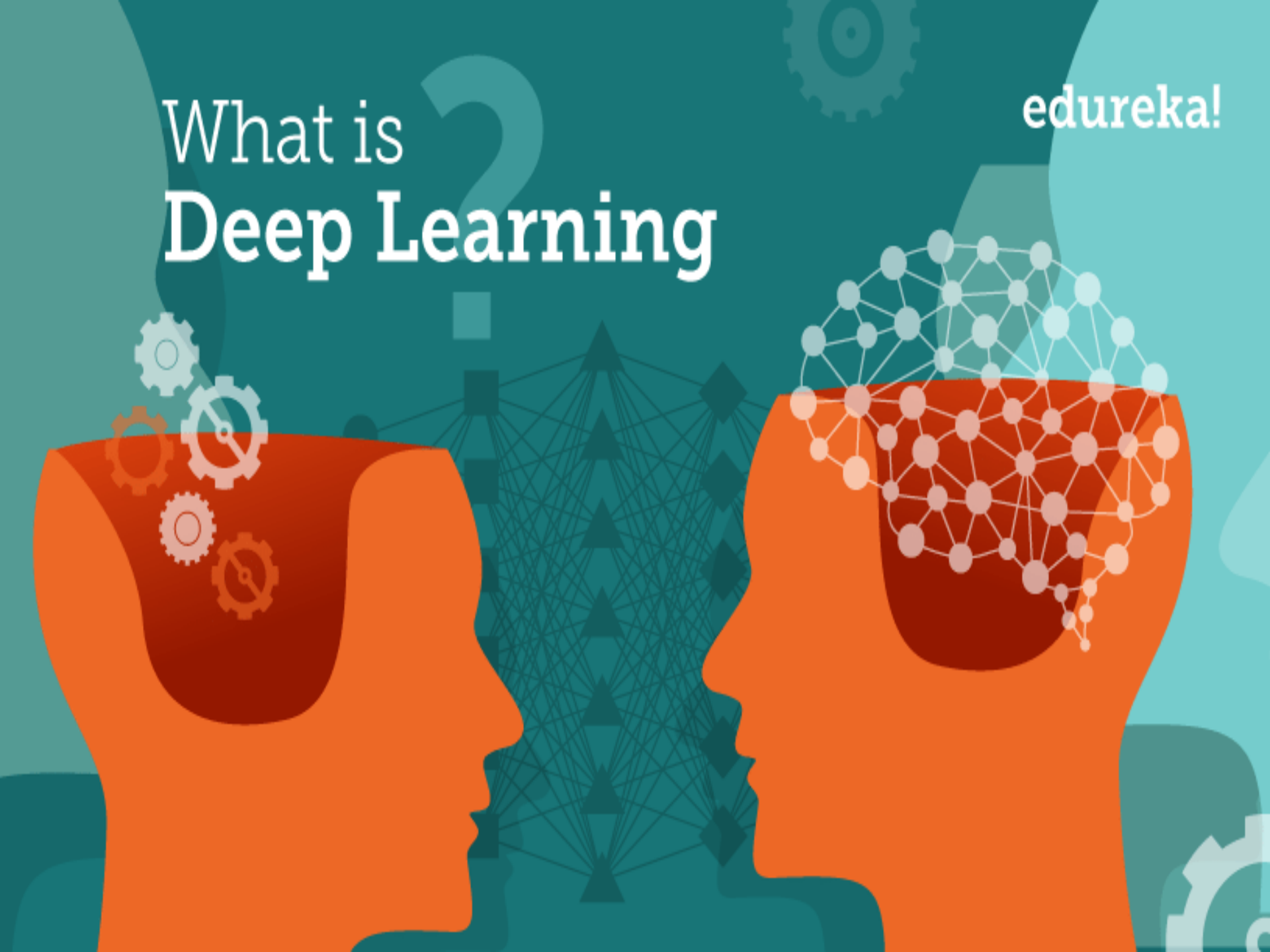
Kauan Alves
Carolina Maldonado

10/2018

Deep Learning & Machine Learning

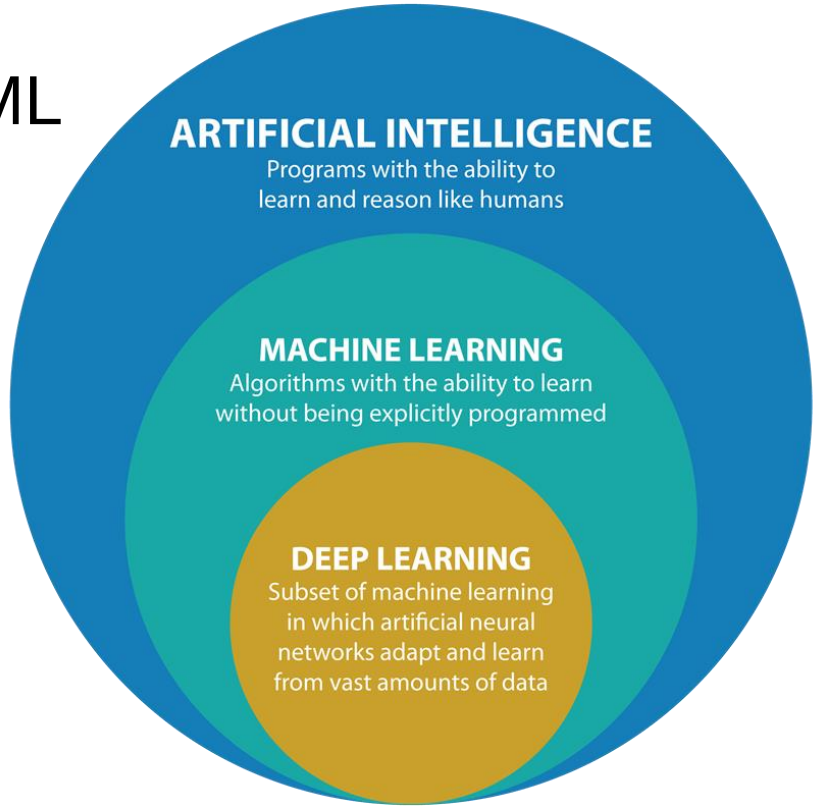




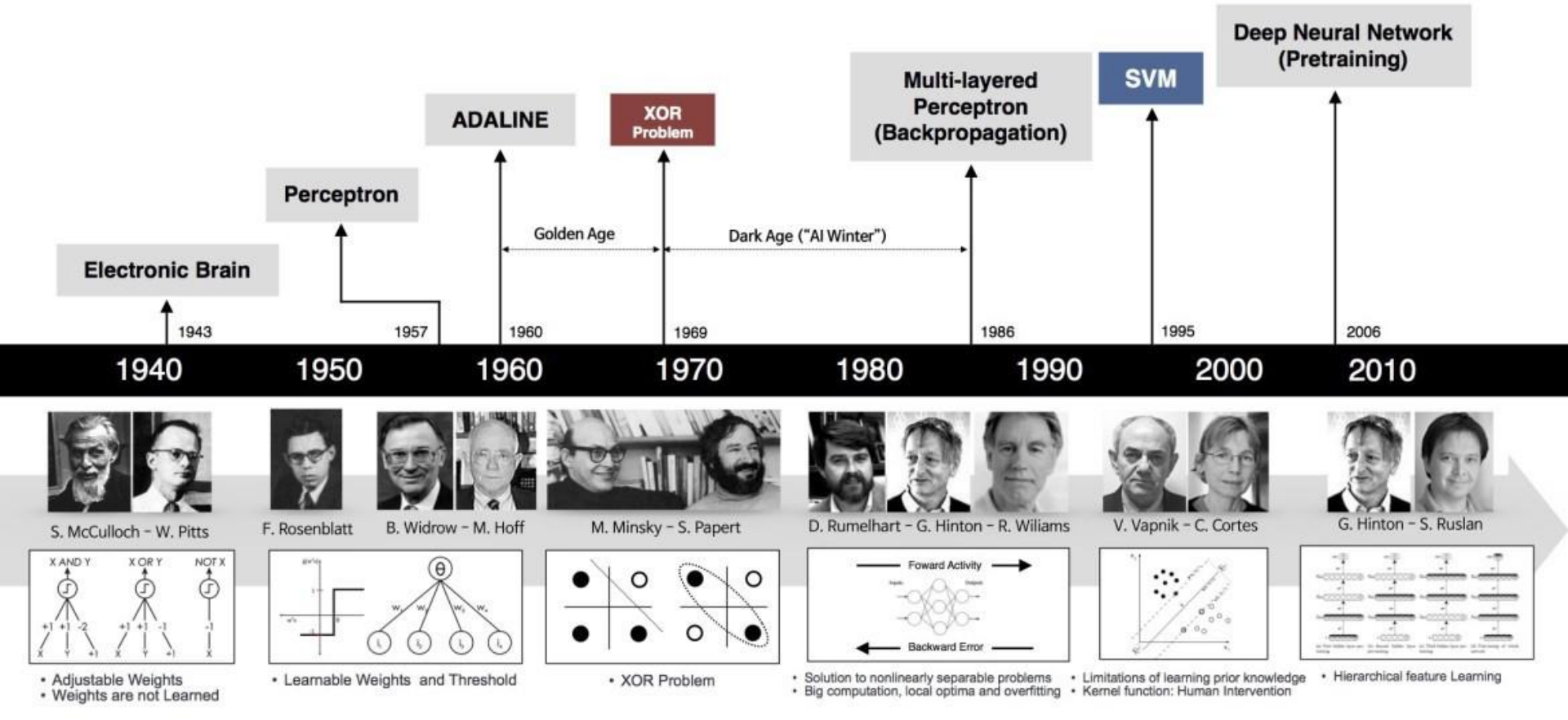

What is
Deep Learning
edureka!

Deep Learning & Machine Learning

- Não são comparáveis
- DL é uma categoria de ML

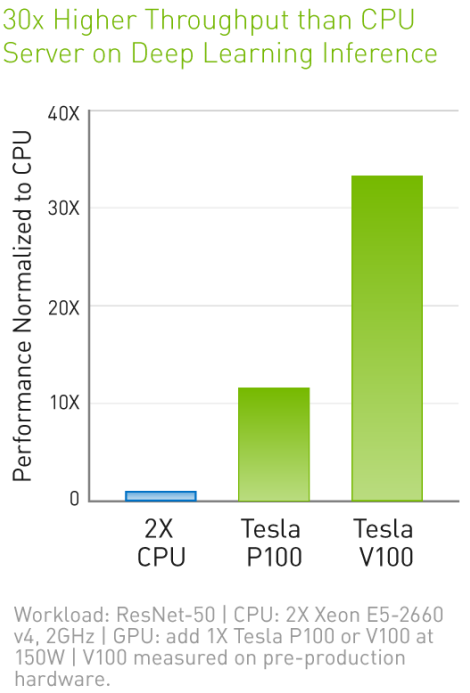


Historia do DL



Historia do DL

Se o Deep Learning existe desde 1940, porque só ficou popular nos dias atuais?

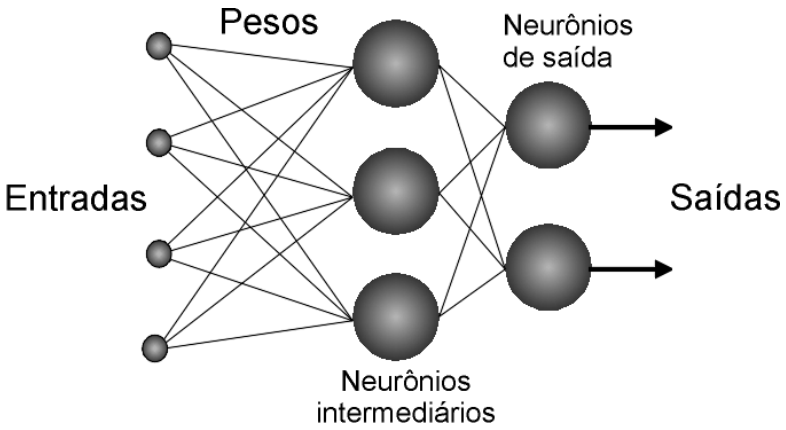


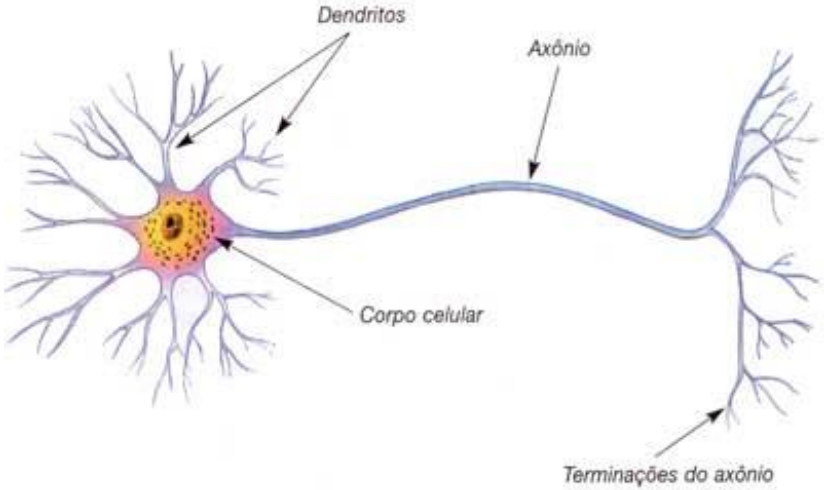
Redes Neurais

- **O que são?**
- **Do que se alimentam?**
- **Onde vivem?**

Hoje, no Globo Reporter!

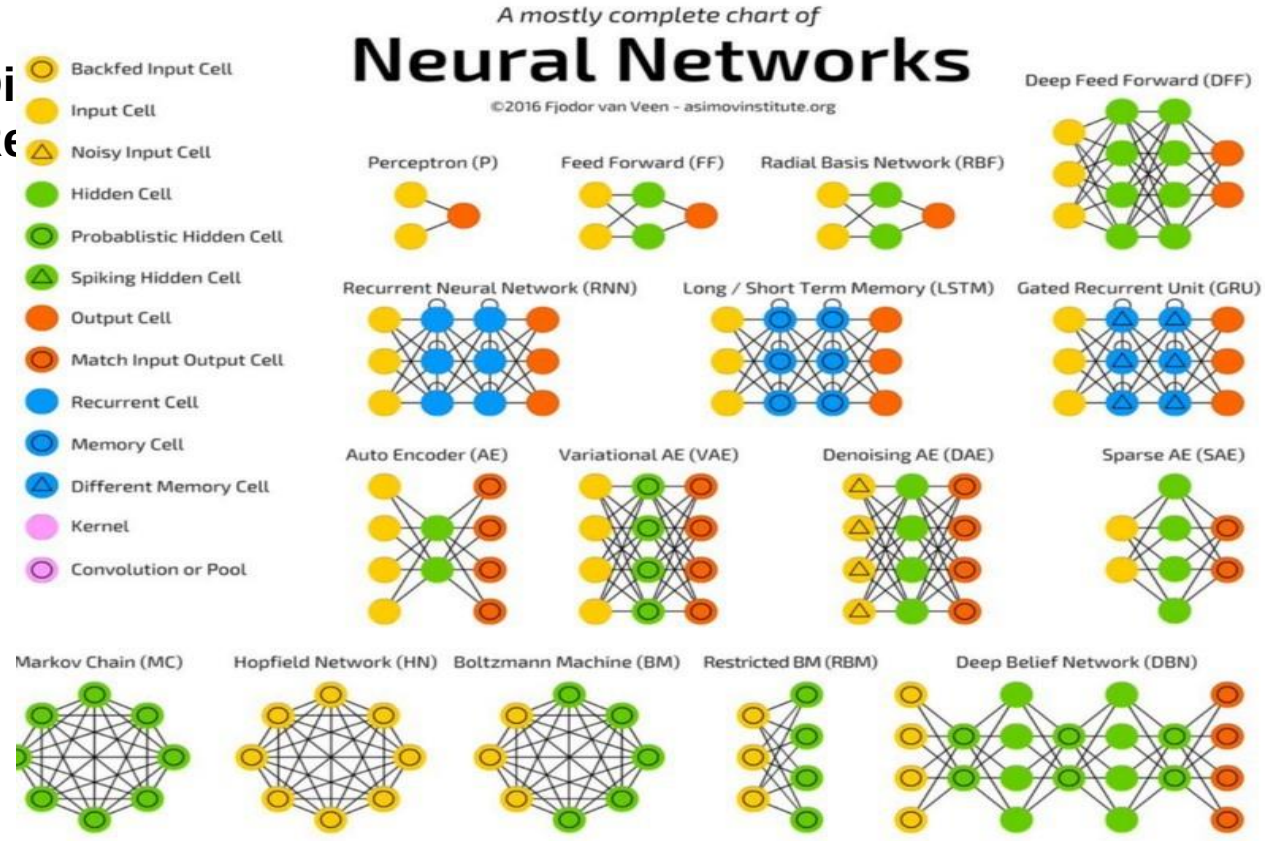
Redes Neurais





Redes Neurais

- Di
- Re

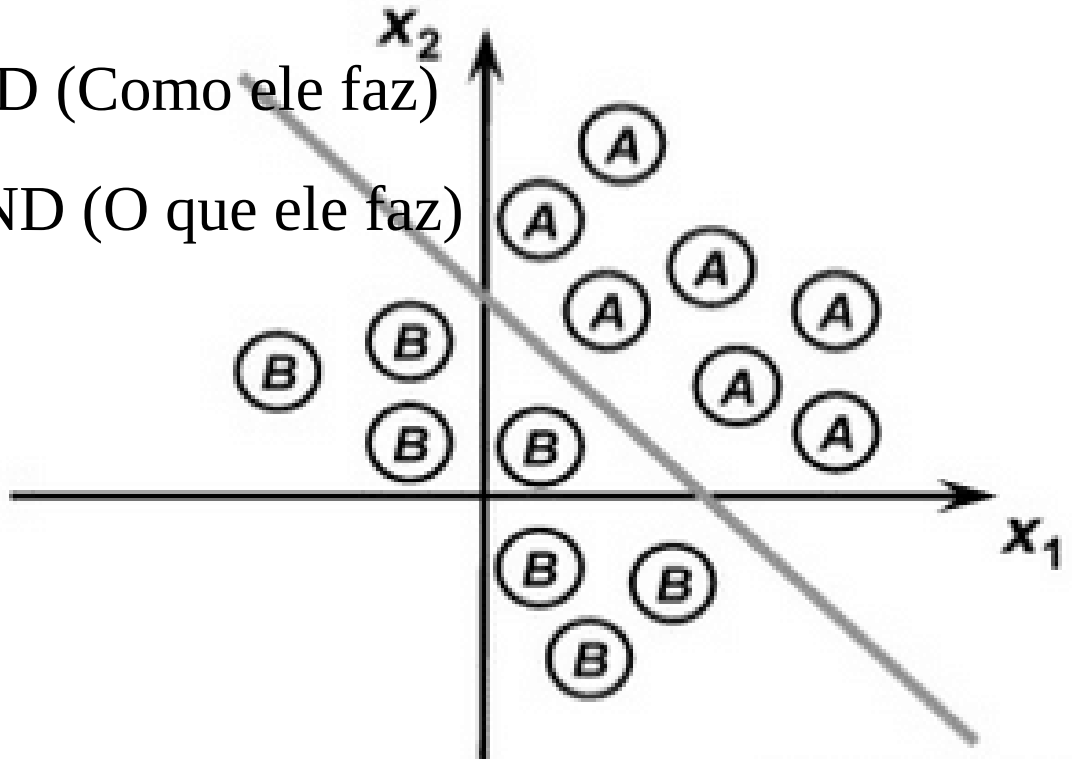


Perceptron

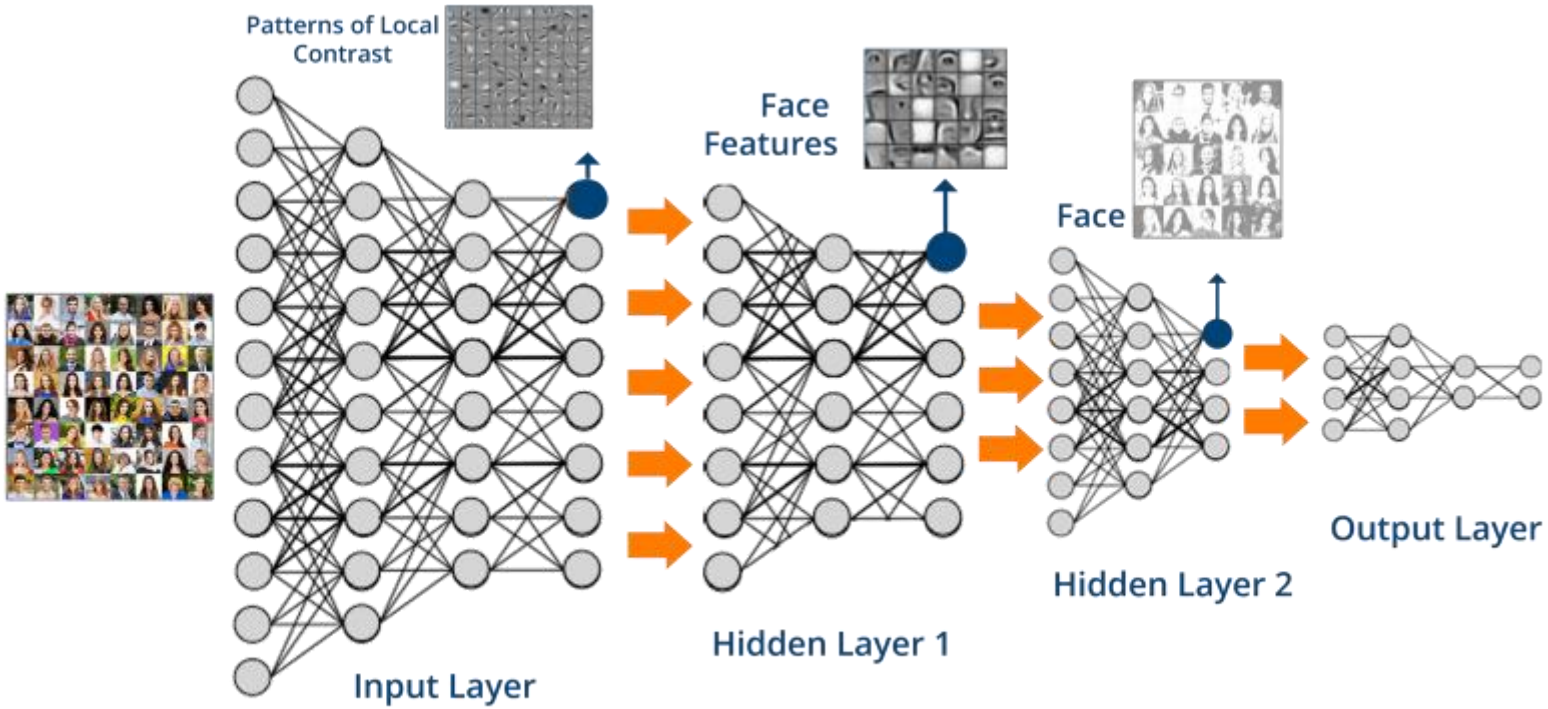
Ilustrando o funcionamento de um Perceptron simples (Apena

BACK-END (Como ele faz)

FRONT-END (O que ele faz)



CNN – Redes Neurais Convulacionais



CÓDIGO

O Dataset utilizado

(1 ou -1) – Significa a marca da cerveja

Significa a quantidade de cevada

Significa a quantidade de álcool

1	1.0000	2.2541	2.0142
2	1.0000	3.0630	3.5071
3	1.0000	2.7079	2.3684
4	1.0000	3.3393	3.2794
5	1.0000	3.5760	4.2409
6	1.0000	4.4244	4.4939

CÓDIGO

```python
def treino_teste_split(dataset, porcentagem):
    """
    Separa e monta o conjunto principal
    em dois para teste e treino.
    """
    percent = porcentagem*len(dataset) // 100
    data_treino = random.sample(dataset, percent)
    data_teste = [data for data in dataset if data not in data_treino]

    def montar(dataset):
        x, y = [], []
        for data in dataset:
            x.append(data[1:3])
            y.append(data[0])
        return x, y

    x_train, y_train = montar(data_treino)
    x_test, y_test = montar(data_teste)
    return x_train, y_train, x_test, y_test
```

Define uma porcentagem de dados para treinar a rede

Função que monta os conjuntos de teste e treino

Utiliza a função para montar os conjuntos e retorna os valores.

CÓDIGO

```
def perceptron_fit(x, d):
    """ Executa o treinamento da rede """
    epoca = 0
    w = [random.random() for i in range(3)]
    print(w)
    while True:
        erro = False
        for i in range(len(x)):
            u = sum([w[0]*-1, w[1]*x[i][0], w[2]*x[i][1]])
            y = sinal(u)
            if y != d[i]:
                w[0] = ajuste(w[0], -1, d[i], y)
                w[1] = ajuste(w[1], x[i][0], d[i], y)
                w[2] = ajuste(w[2], x[i][1], d[i], y)
                erro = True
        epoca += 1
        if erro is False or epoca == 1000:
            break
    print(epoca)
    return w
```

Inicia uma época = 0 (para traçar o fim do treinamento), e um peso (w) para ajustar a rede

Inicia um loop e executa um treinamento, onde o U é a fórmula do perceptron

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

Verifica se não houve erro (significa que a rede finalizou o treinamento) ou se chegou ao fim das épocas e finaliza o treino

CÓDIGO

```python
from sklearn import linear_model, model_selection, metrics
import pandas as pd

# Carrega os elementos do dataset
dataset = pd.read_csv('data1.csv')
x = dataset.iloc[:, 1:].values
y = dataset.iloc[:, 0].values

# Cria um objeto Perceptron
perceptron = linear_model.Perceptron()

# Split em conjunto de treino e teste
x_train, x_test, y_train, y_test = model_selection.train_test_split(x, y, test_size=0.2, random_state=0)

# Treinamento
classificador = perceptron.fit(x_train, y_train)

# Validação
y_predict = classificador.predict(x_test)

# Acurácia
print(metrics.accuracy_score(y_test, y_predict))
```

Lê o Dataset utilizando a biblioteca Pandas

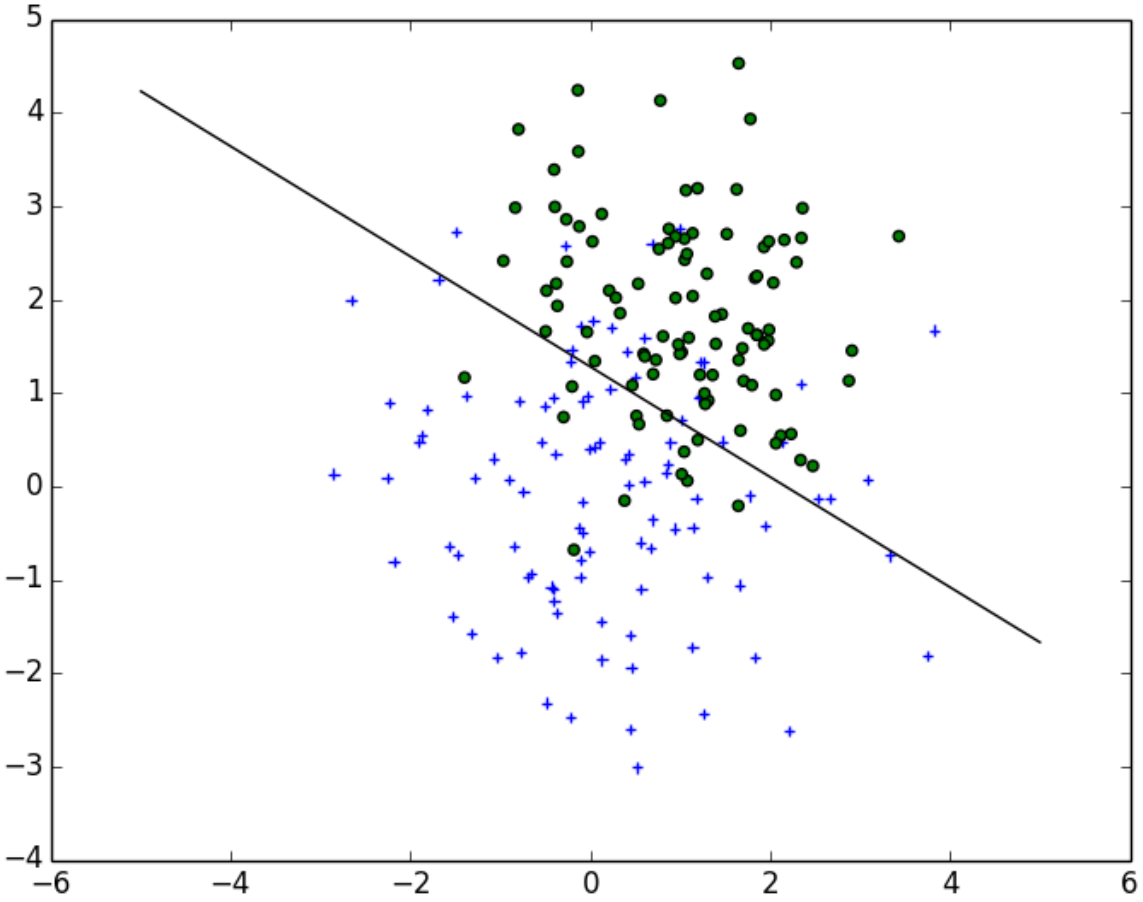
Cria o Perceptron

Separa conjuntos de teste e treino

Executa o treinamento

Valida os dados e a acurácia

Código



Obrigado!

Código: https://github.com/dunossauro/live-de-python/blob/master/codigo/Live35/perceptron_padrao.py

Lives: https://www.youtube.com/user/mendesesduardo

Video: https://www.youtube.com/watch?v=aircAruvnKk