

Deep Learning

**Kauan Alves
Carol**

10/2018

Deep Learning & Machine Learning

- Não são comparáveis
- DL é uma categoria de ML

Deep Learning & Machine Learning

ARTIFICIAL INTELLIGENCE

Programs with the ability to learn and reason like humans

MACHINE LEARNING

Algorithms with the ability to learn without being explicitly programmed

DEEP LEARNING

Subset of machine learning in which artificial neural networks adapt and learn from vast amounts of data

Interesse ao longo do tempo

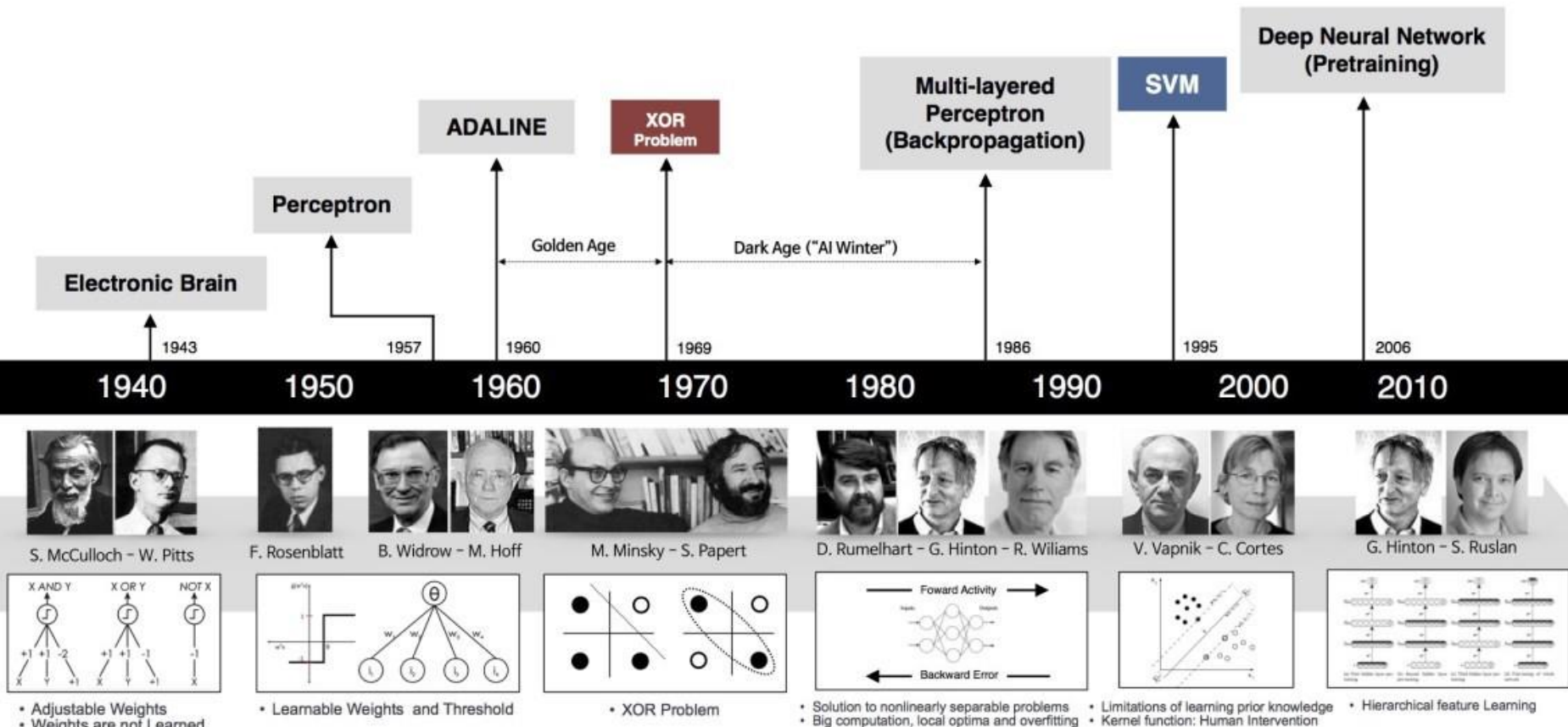
Google Trends

● Deep Learning ● Machine Learning ● Artificial Intelligence



Todo o mundo. Nos últimos 5 anos. Pesquisa Google na Web.

Historia do DL

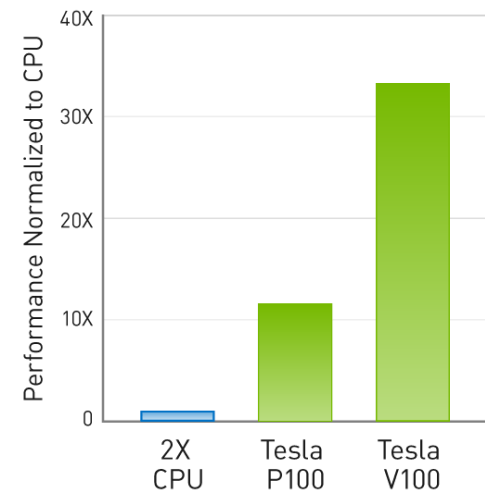


Historia do DL

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



30x Higher Throughput than CPU
Server on Deep Learning Inference

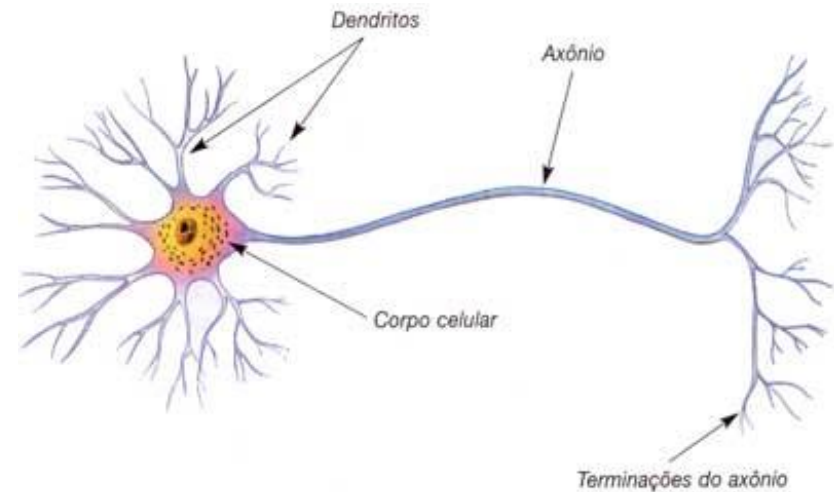
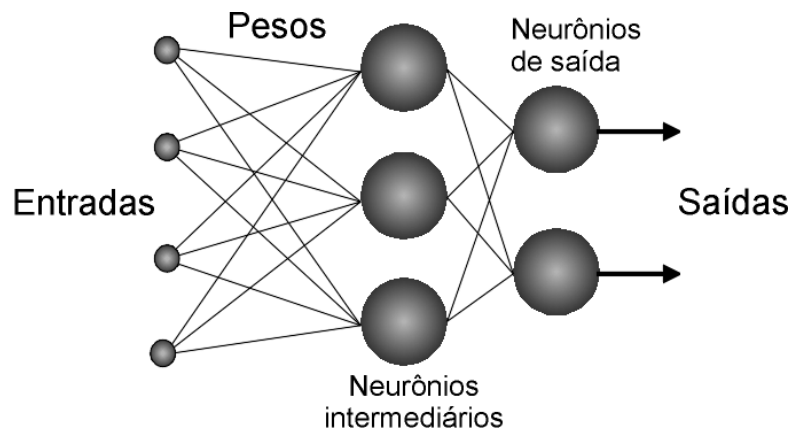


Workload: ResNet-50 | CPU: 2X Xeon E5-2660 v4, 2GHz | GPU: add 1X Tesla P100 or V100 at 150W | V100 measured on pre-production hardware.

Redes Neurais

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

Redes Neurais



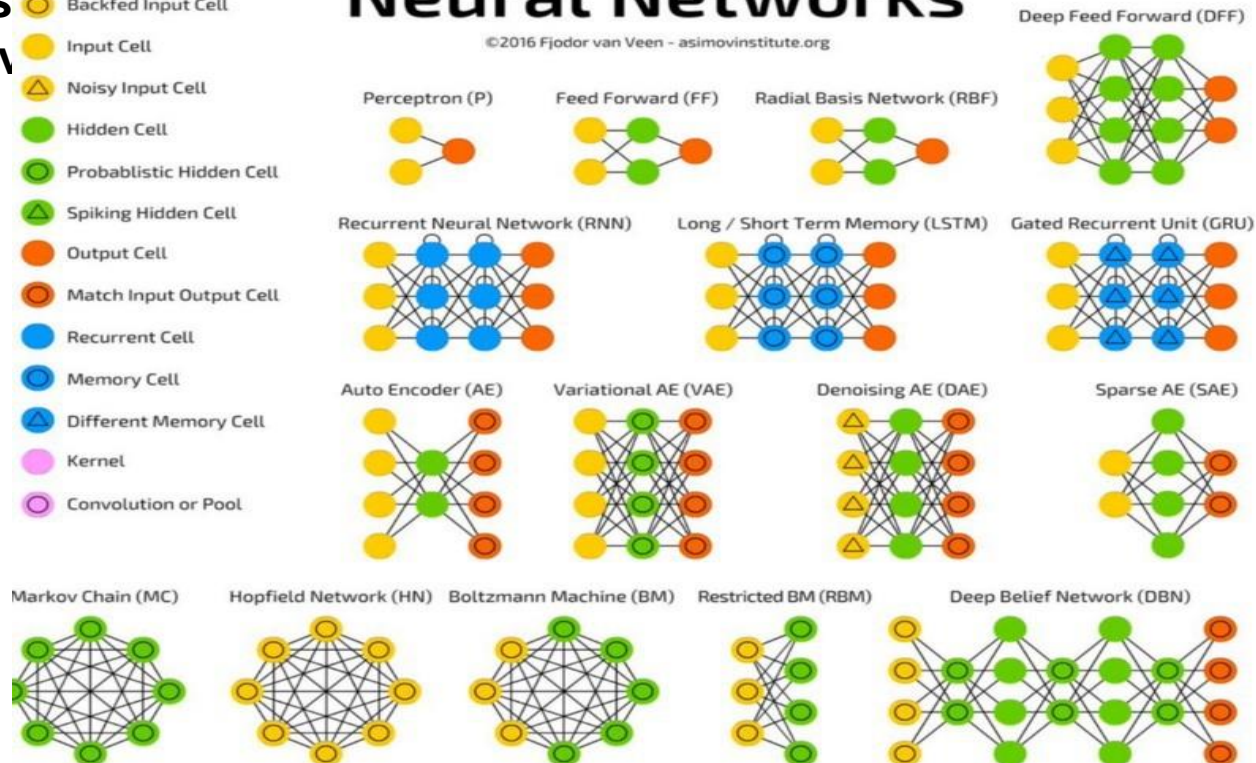
Redes Neurais

- Divers
- Resol



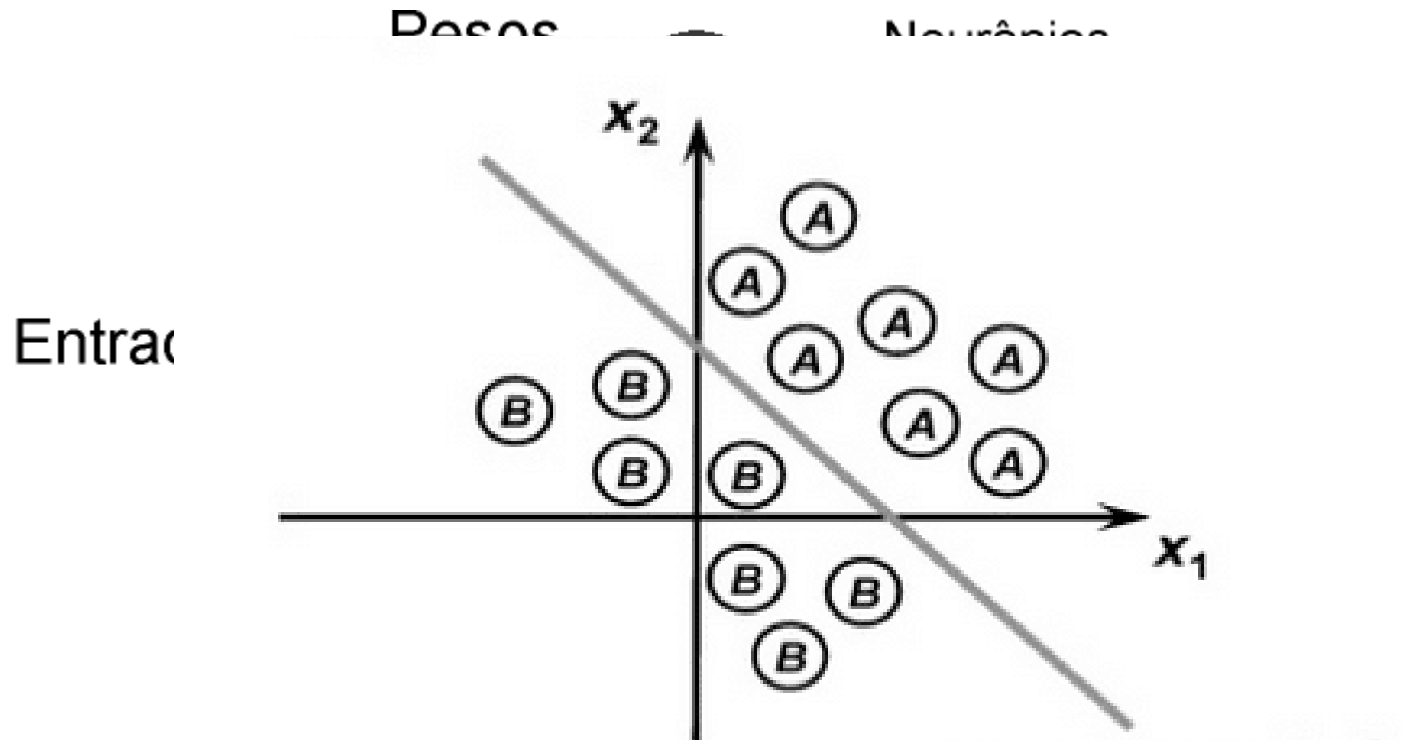
A mostly complete chart of Neural Networks

©2016 Fjodor van Veen - asimovinstitute.org

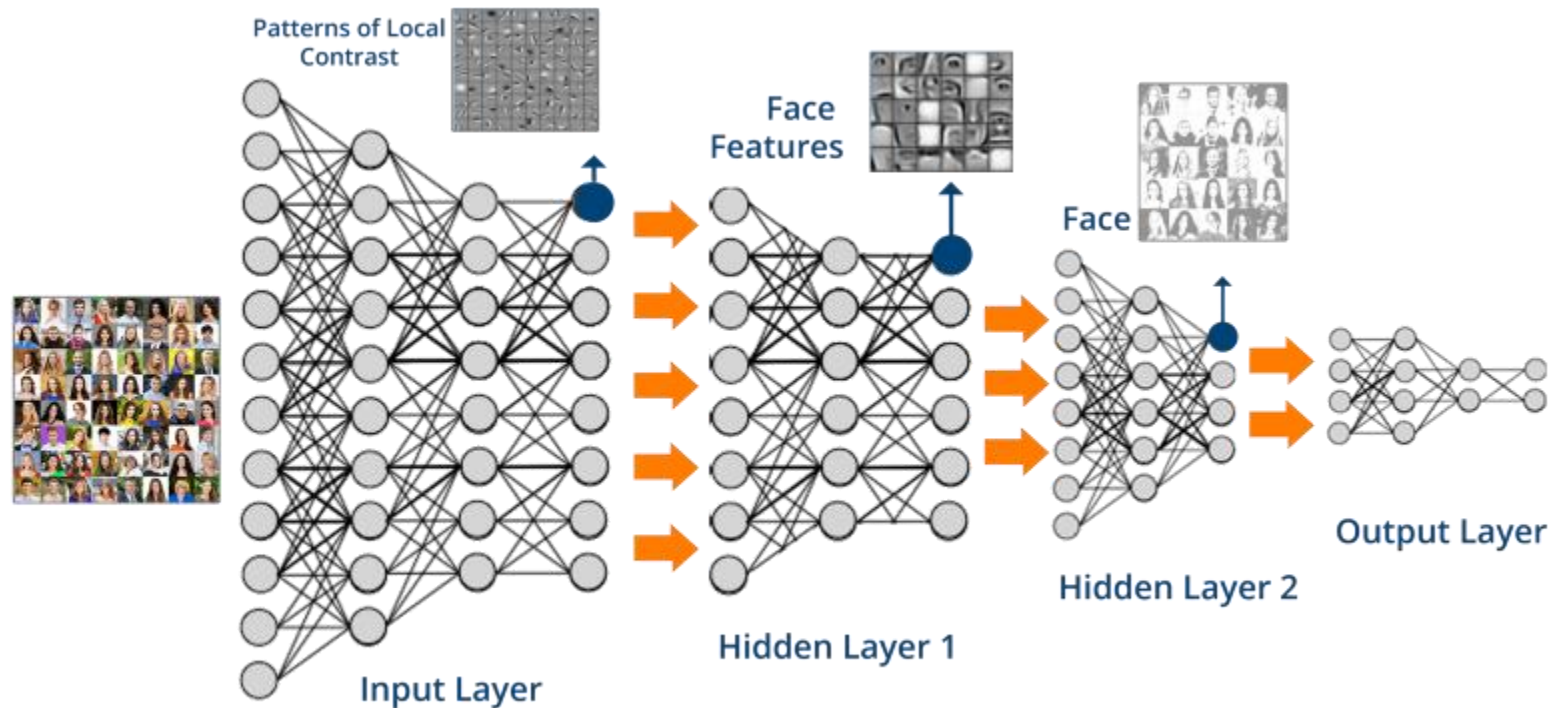


Perceptron

Ilustrando o funcionamento de um Perceptron simples
(Apenas uma camada)



CNN – Redes Neurais Convolucionais



CÓDIGO


O Dataset utilizado

| | | | |
|---|--------|--------|--------|
| 1 | 1.0000 | 2.2541 | 2.0142 |
| 2 | 1.0000 | 3.0051 | 3.5071 |
| 3 | 1.0000 | 2.7009 | 2.3684 |
| 4 | 1.0000 | 3.3393 | 3.2794 |
| 5 | 1.0000 | 3.5760 | 4.2409 |
| 6 | 1.0000 | 4.4244 | 4.4939 |

(1 ou 0) – Significa a marca da cerveja

Significa a quantidade de cevada

Significa a quantidade de álcool



CÓDIGO

Não se assustem se não entenderem nada!

Eu também não entendo

```
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

Não se assustem se não entenderem nada!

Eu também não entendo

```
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

Agora até que dá pra entender.

```
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
```

Lê o Dataset utilizando a biblioteca Pandas

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

Cria o Perceptron e os conjuntos de teste e treino

```
# 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)
```

Executa o treinamento

```
# Treinamento
```

```
classificador = perceptron.fit(x_train, y_train)
```

Valida os dados e a acurácia

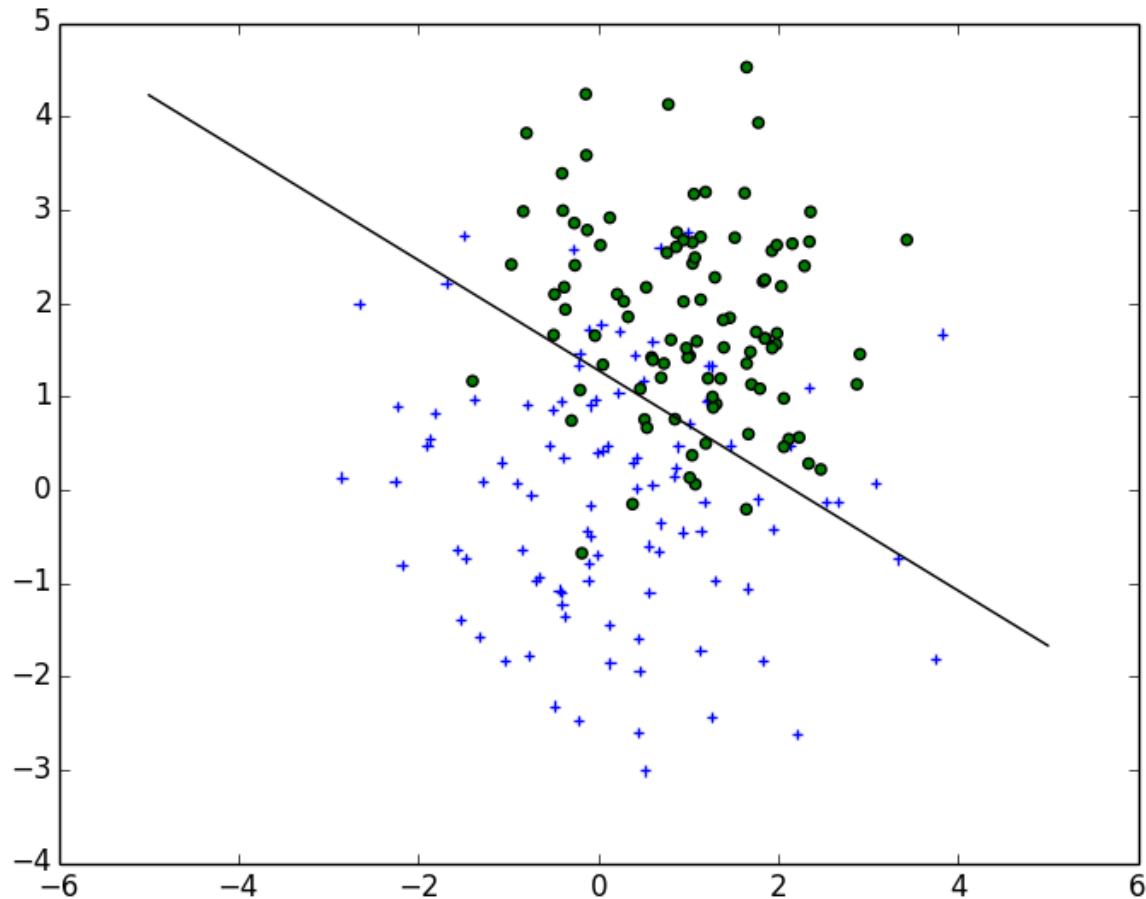
```
# Validação
```

```
y_predict = classificador.predict(x_test)
```

```
# Acurácia
```

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

Código



Obrigado!

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

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