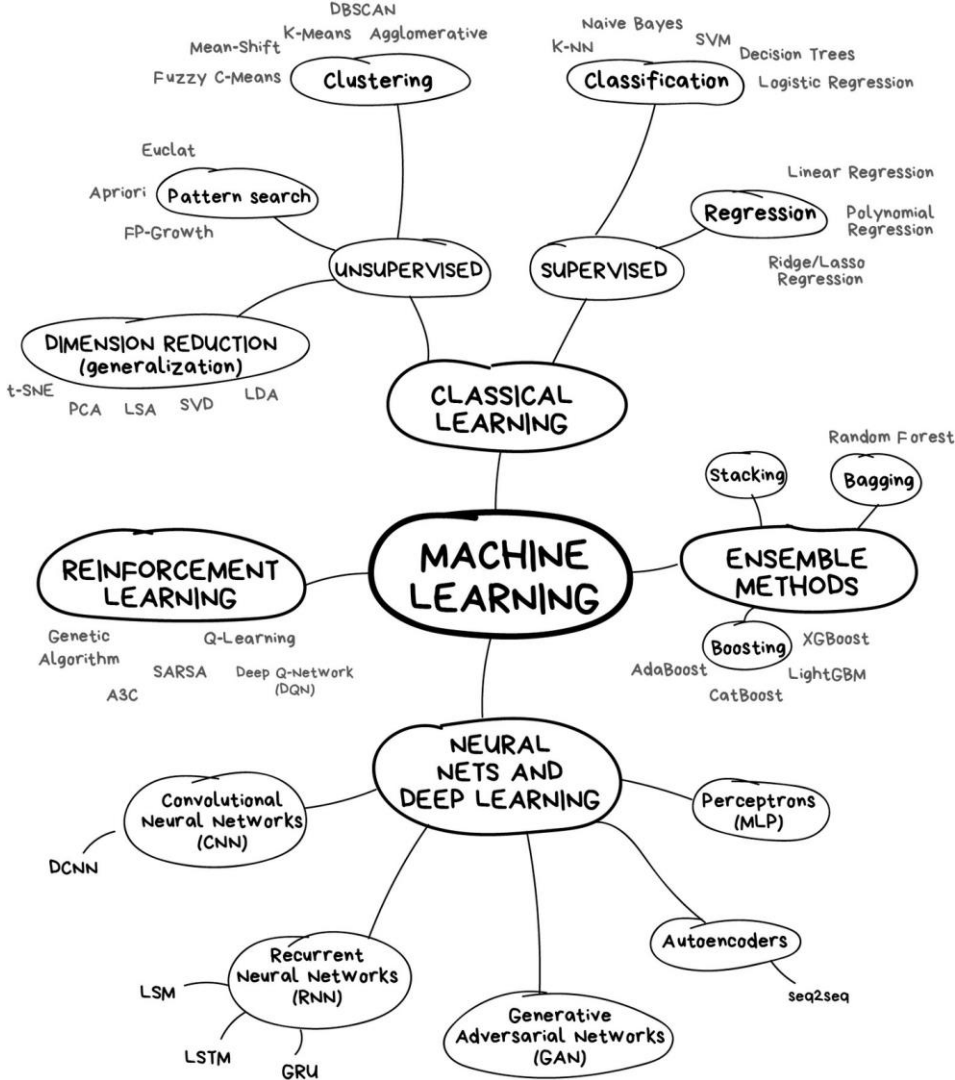
Introdução a Machine Learning e Deep Learning

Guilherme Campos



GDG Cuiabá

Comunidades de desenvolvimento

http://devmt.herokuapp.com/

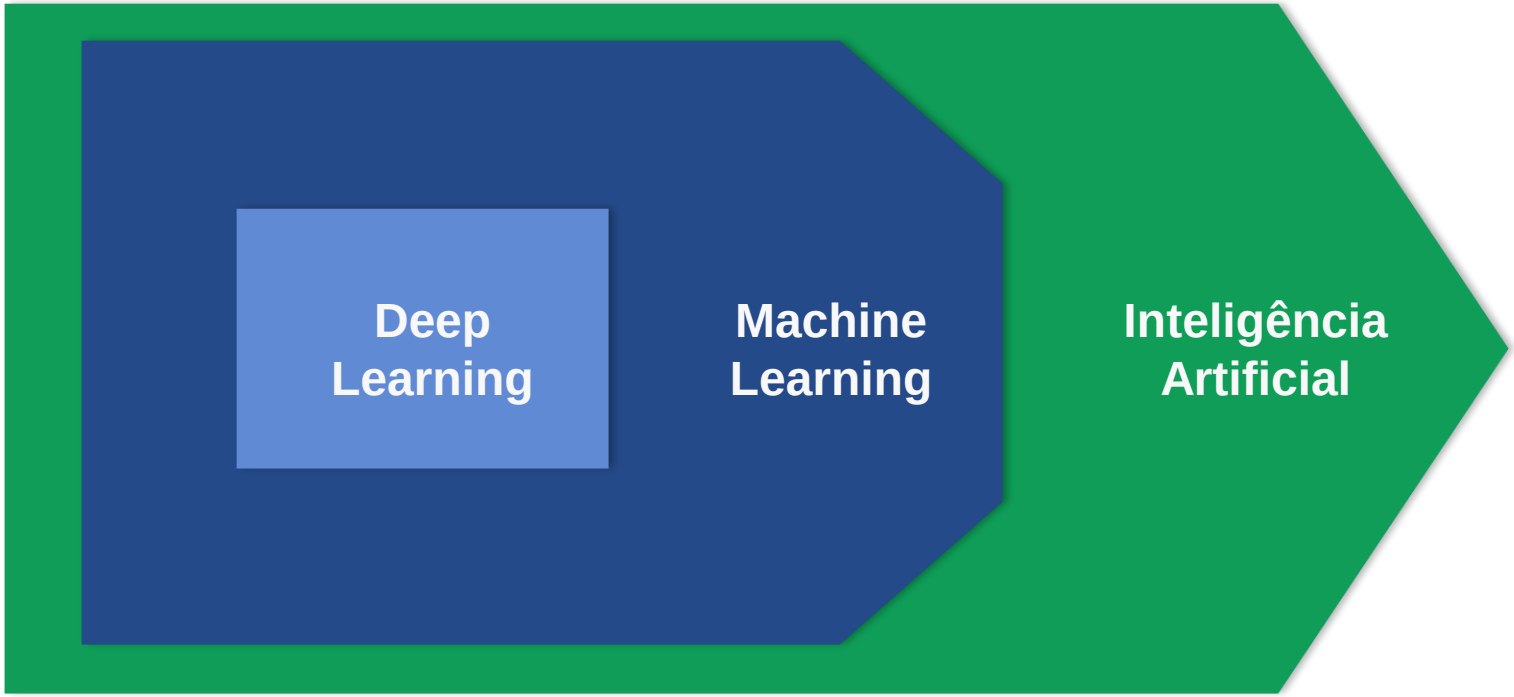
Agenda

Conceitos

- Machine Learning
- Tipos de aprendizado
- Redes Neurais
- Deep Learning

Conceitos

Machine Learning

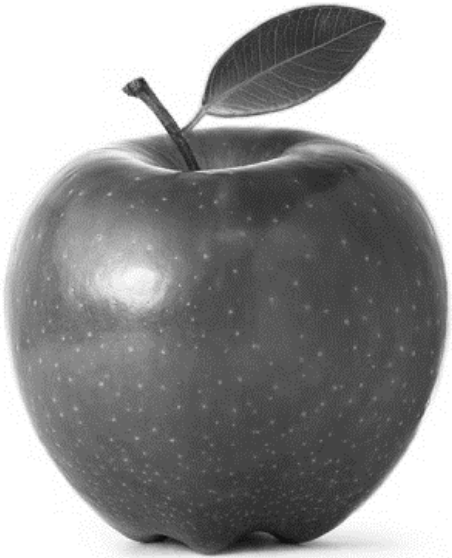


Exemplo

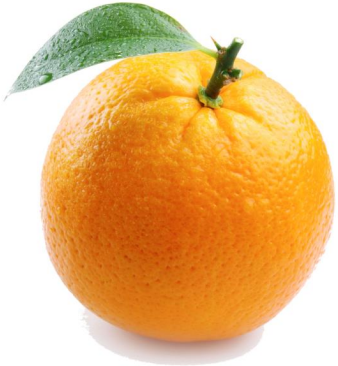
- Como classificar uma imagem como Maçã ou Laranja?

Exemplo

Exemplo

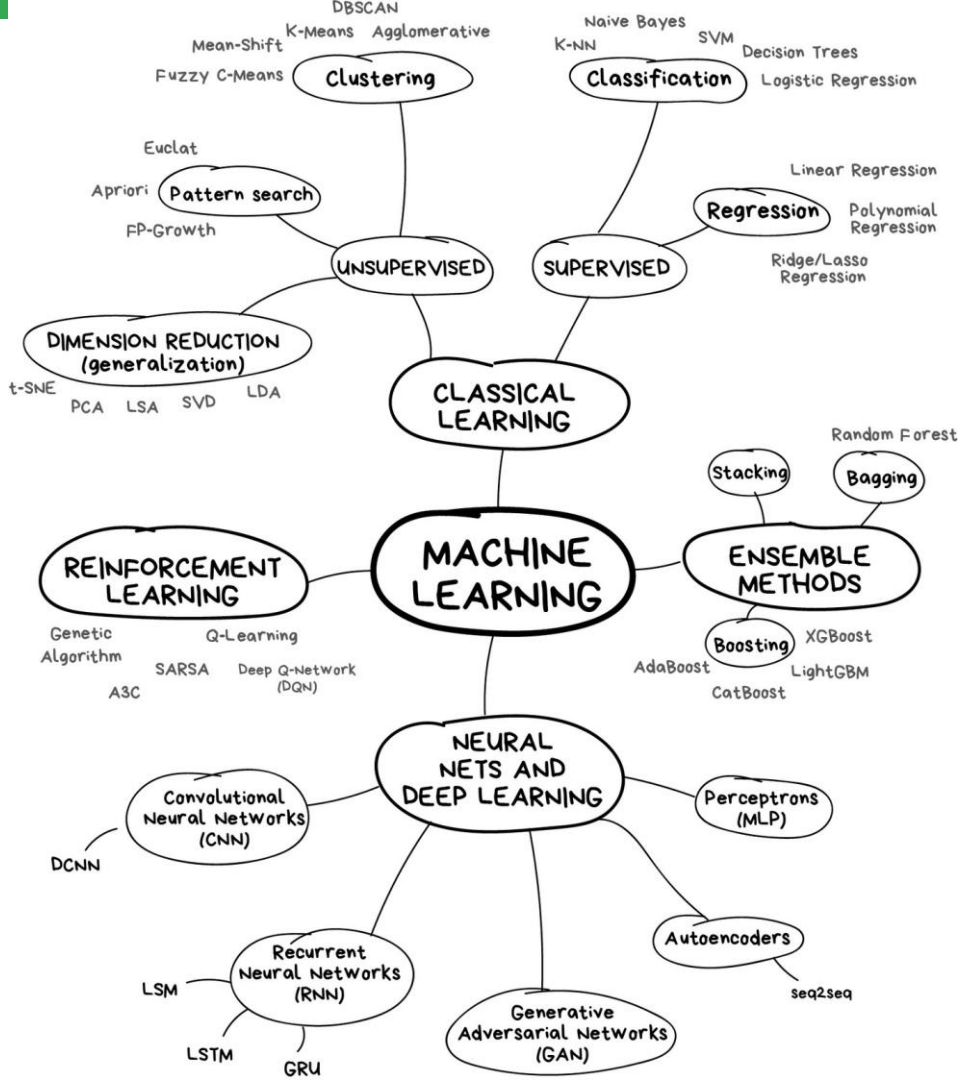


Exemplo



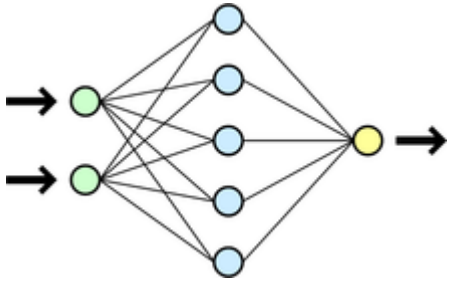
Novo problema? Comece tudo de novo!

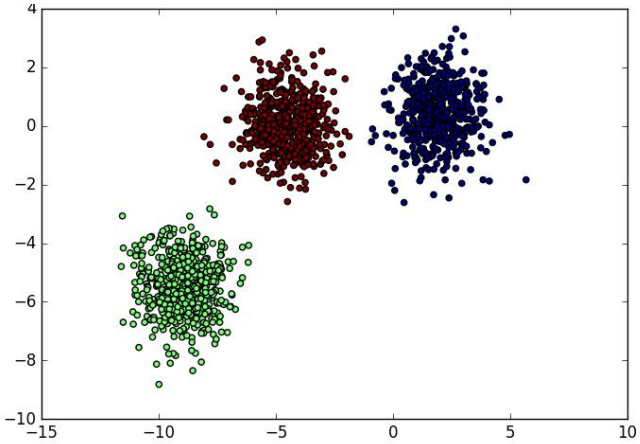
Machine Learning


MACHINE LEARNING
CLASSICAL LEARNING
Clustering
DBSCAN
K-Means
Agglomerative
Mean-Shift
Fuzzy C-Means
Classification
Naive Bayes
K-NN
SVM
Decision Trees
Logistic Regression
UNSUPERVISED
Pattern search
Euclat
Apriori
FP-Growth
SUPERVISED
Regression
Linear Regression
Polynomial Regression
Ridge/Lasso Regression
DIMENSION REDUCTION (generalization)
t-SNE
PCA
LSA
SVD
LDA
REINFORCEMENT LEARNING
Genetic Algorithm
Q-Learning
SARSA
Deep Q-Network (DQN)
A3C
ENSEMBLE METHODS
Stacking
Bagging
Random Forest
Boosting
XGBoost
AdaBoost
LightGBM
CatBoost
NEURAL NETS AND DEEP LEARNING
Convolutional Neural Networks (CNN)
DCNN
Perceptrons (MLP)
Recurrent Neural Networks (RNN)
LSM
LSTM
GRU
Autoencoders
seq2seq
Generative Adversarial Networks (GAN)

Machine Learning – Aprendizagem de Máquina

- Aprendizagem Supervisionado
- Aprendizagem Semi-supervisionada
- Aprendizagem Não-supervisionado
- Aprendizagem por Reforço



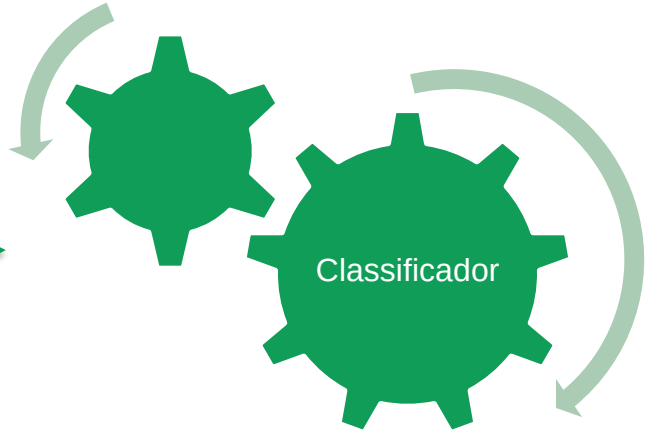


Aprendizagem supervisionada

Caracteristicas	Rotulo
1.jpg	Maçã
2.jpg	Laranja
...	...

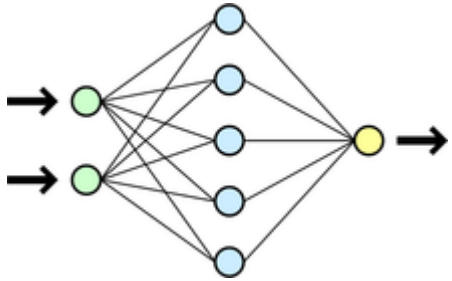
Conjunto de dados

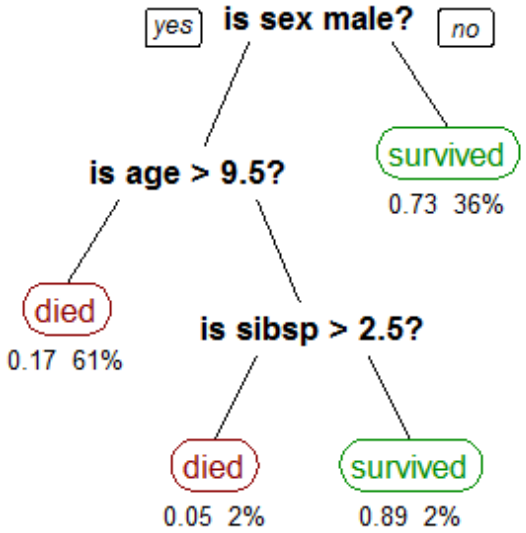
Buscar um padrão

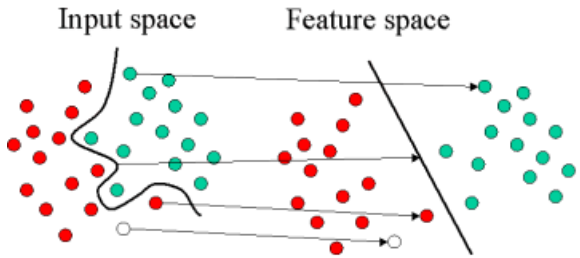


Aprendizagem supervisionada

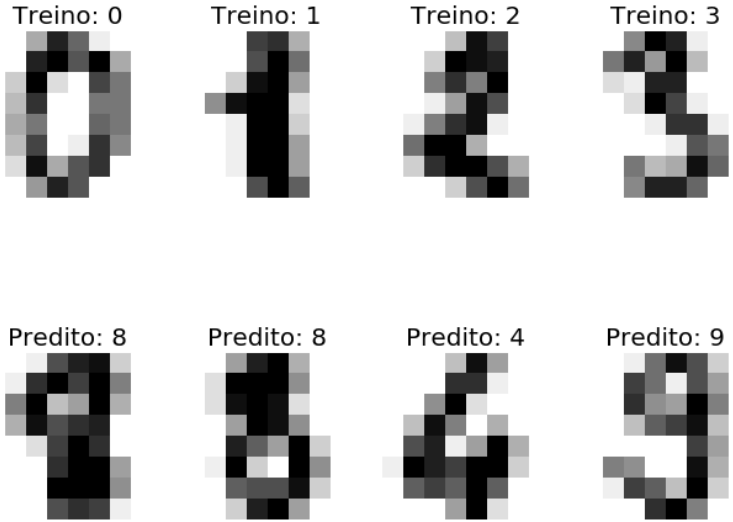
- Árvores de decisão
- Regressão Linear/Logística
- Support Vector Machines
- Redes Neurais
- K- Vizinhos mais próximos







```
from sklearn import datasets, svm, metrics
import matplotlib.pyplot as plt

# Carrega o dataset
digits = datasets.load_digits()

# transforma a imagem em vetor
n_samples = len(digits.images)
data = digits.images.reshape((n_samples, -1))

x_train = data[:n_samples / 2]
y_train = digits.target[:n_samples / 2]
# cria um classificador SVM
classifier = svm.SVC(gamma=0.001)

# realiza o ajuste dos dados
classifier.fit(x_train, y_train)

esperado = digits.target[n_samples / 2:]
predito = classifier.predict(data[n_samples / 2:])
```



Aprendizagem não supervisionada

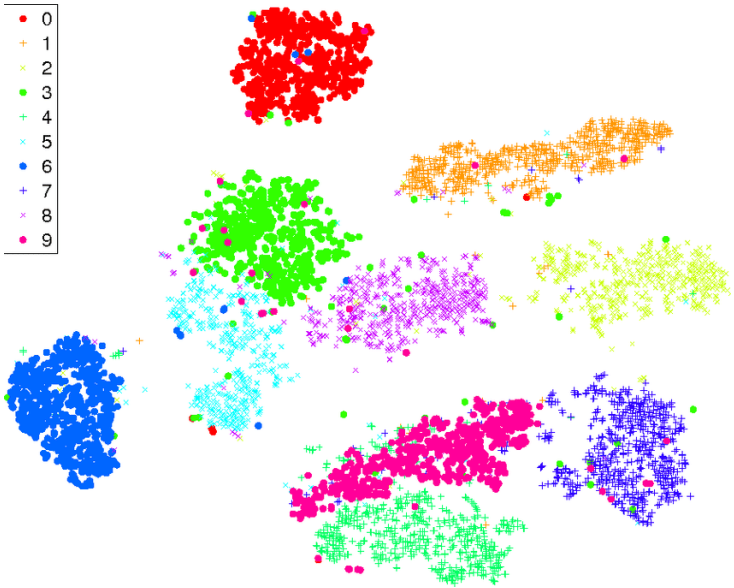
Caracteristicas	Rotulo
1.jpg	?
2.jpg	?
...	???

Conjunto de dados

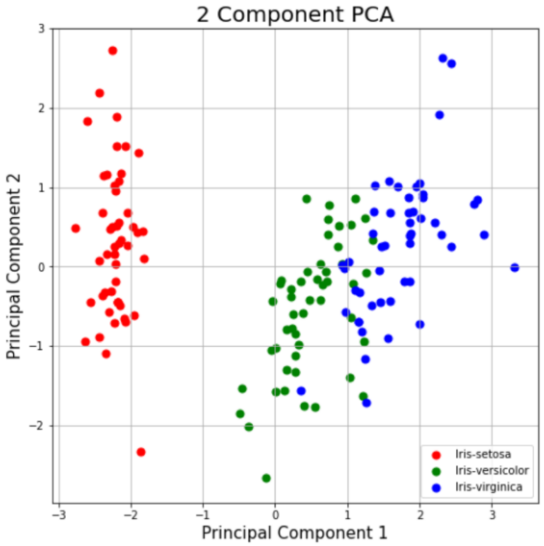
Buscar um padrão

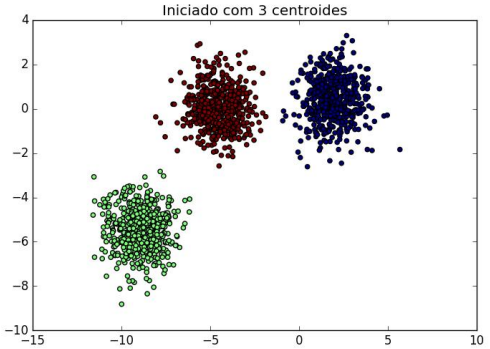
Aprendizagem não supervisionada

- DBSCAN
- K-means
- t-SNE
- PCA

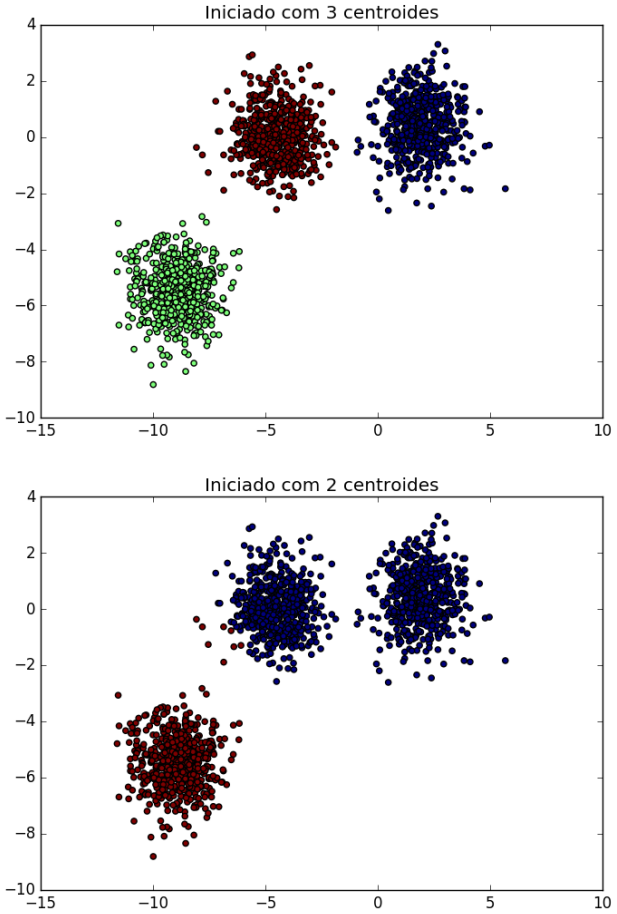


(a) Visualization by t-SNE.





```python
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs

n_samples = 1500
random_state = 170
X, y = make_blobs(n_samples=n_samples, random_state=random_state)

y_pred = KMeans(n_clusters=3, random_state=random_state).fit_predict(X)
plt.subplot(211)
plt.scatter(X[:, 0], X[:, 1], c=y_pred)
plt.title("Iniciado com 3 centroides")

y_pred = KMeans(n_clusters=2, random_state=random_state).fit_predict(X)
plt.subplot(212)
plt.scatter(X[:, 0], X[:, 1], c=y_pred)
plt.title("Iniciado com 2 centroides")

plt.show()
```

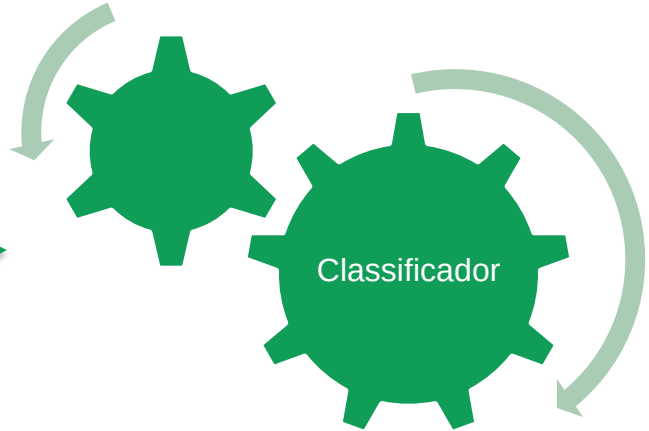


Aprendizagem semi-supervisionada

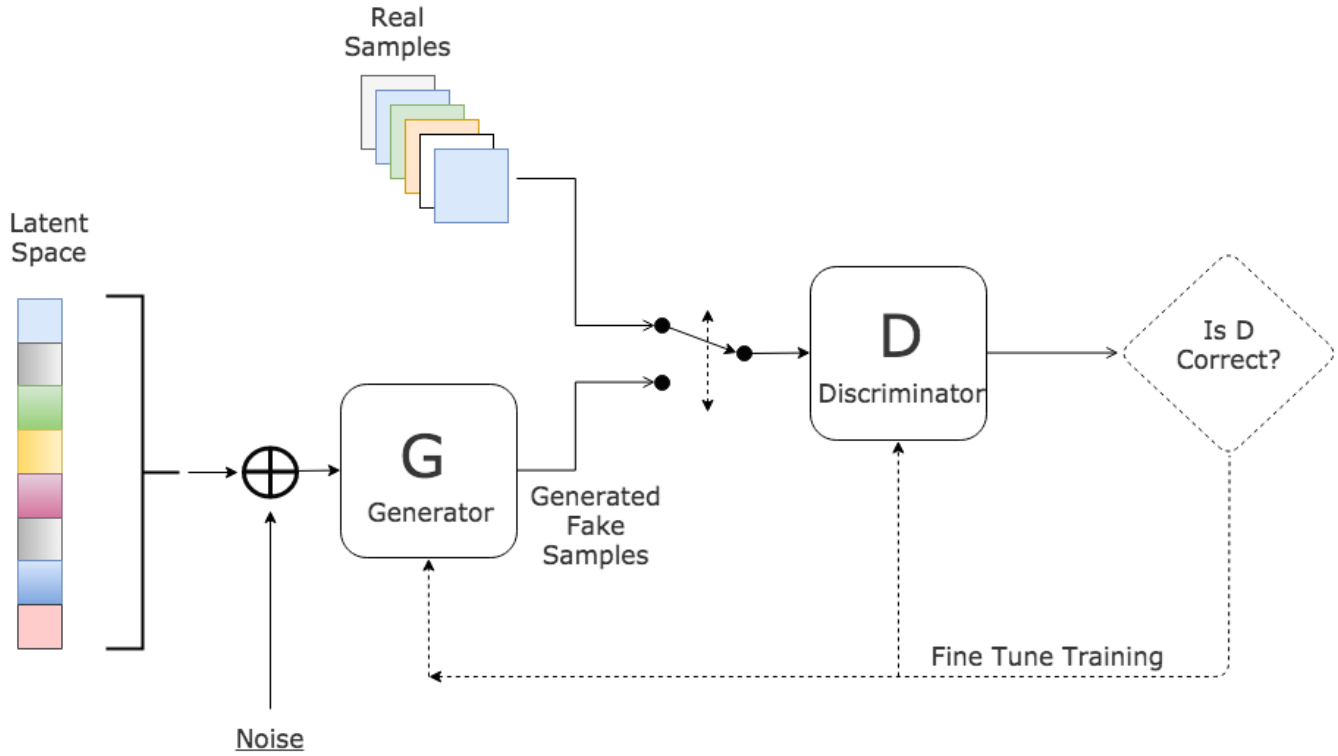
Caracteristicas	Rotulo
1.jpg	Maçã
2.jpg	???
3.jpg	Laranja
4.jpg	???
...	...

Conjunto de dados

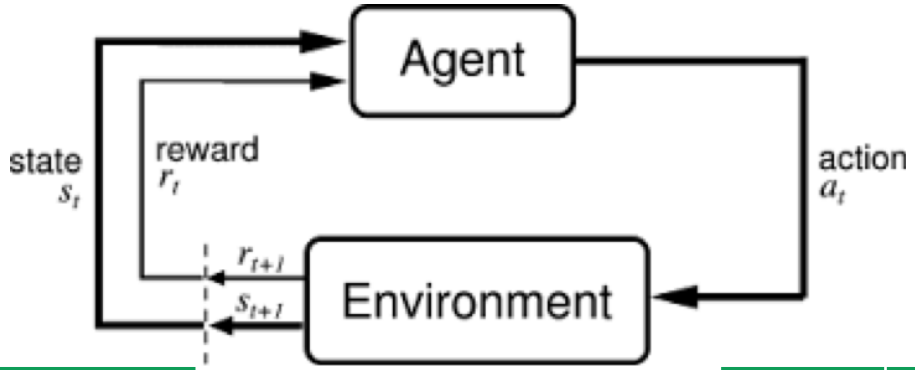
Buscar um padrão



Generative Adversarial Network



Aprendizagem por reforço



Estado	Ação	Recompensa
???	?	?
???	?	?
...	???	???

Conjunto de estados

Realiza uma ação

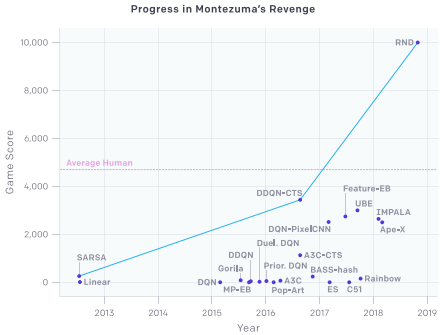
Estado	Ação	Recompensa
1	UP	0
???	?	?
...	???	???

Conjunto de estados

Aprendizagem por reforço

Better than human-level control of classic Atari games through Deep Reinforcement Learning.

OpenAI



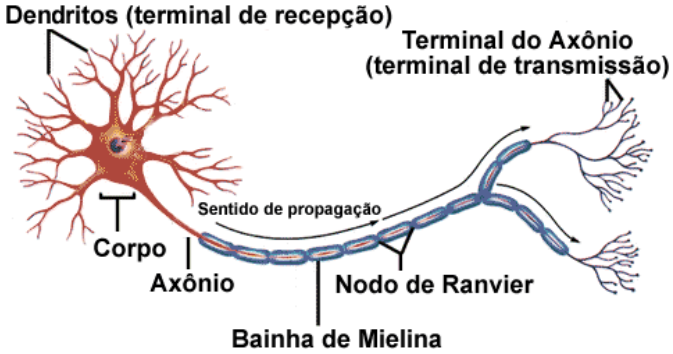
DeepMind

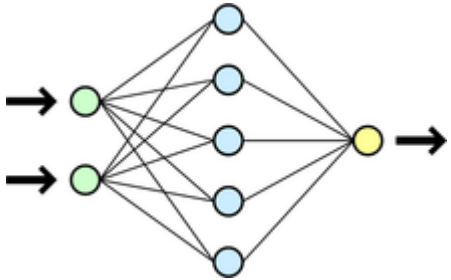
GDG Cuiabá

Redes Neurais

Redes Neurais Artificiais

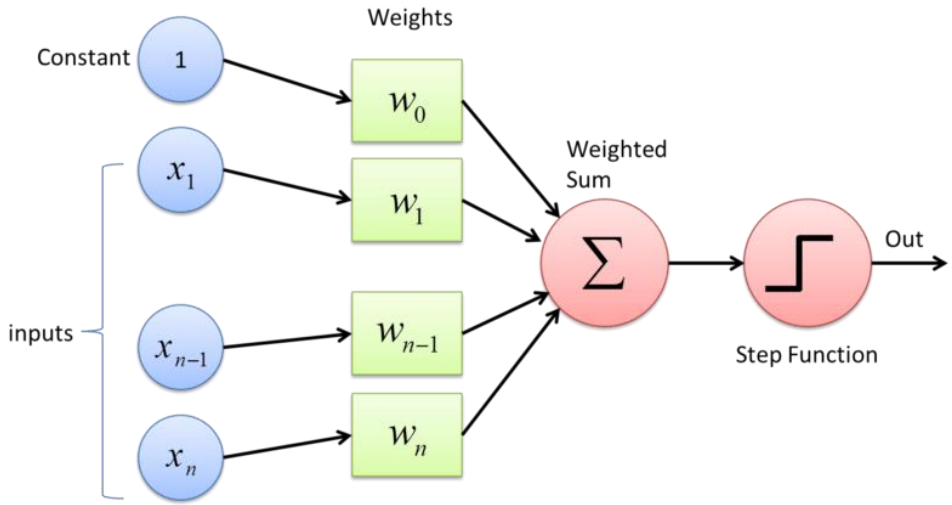
- Inicialmente foram inspiradas no sistema nervoso.
- Uma rede de unidades ou nós interconectados .
- Os nós são chamados de neurônios artificiais.



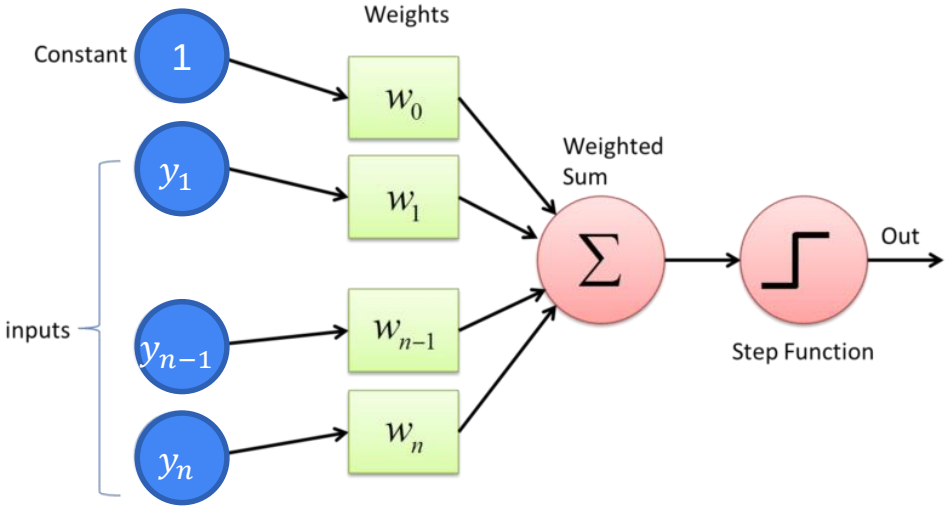


Perceptron

- Modelo mais simples de rede neural,
- Possui apenas uma camada e um único nó.
- Esse modelo foi proposto em 1958.
- É um classificador binário.



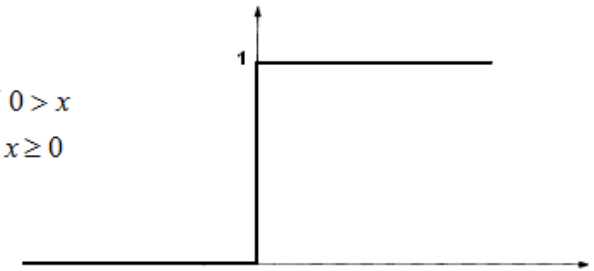
Perceptron



$$x = \sum_{i=1}^{n} w_i \, y_i + \theta$$

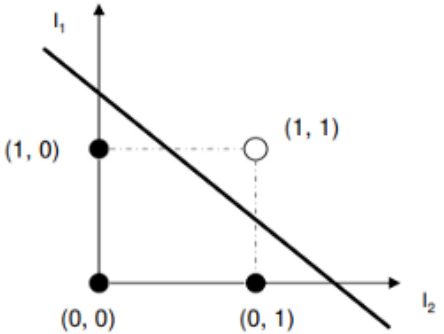
Unit step (threshold)

$$f(x) = \begin{cases} 0 \text{ if } 0 > x \\ 1 \text{ if } x \geq 0 \end{cases}$$

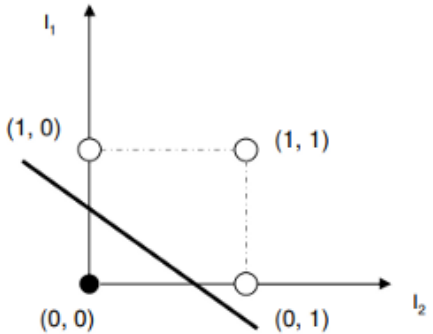


Perceptron - Exemplo

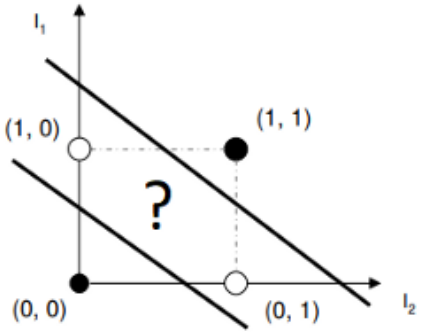
AND		
I_1	I_2	out
0	0	0
0	1	0
1	0	0
1	1	1



OR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	1



XOR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	0

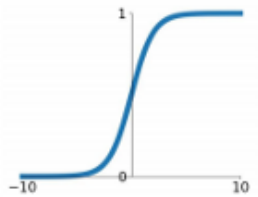


Multilayer Perceptron (MLP)

- Perceptron com mais de uma camada.
- Aprendizado por backpropagation.
- Uma rede de perceptrons.
- Diferentes funções de ativação.

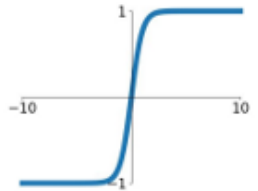
Sigmoid

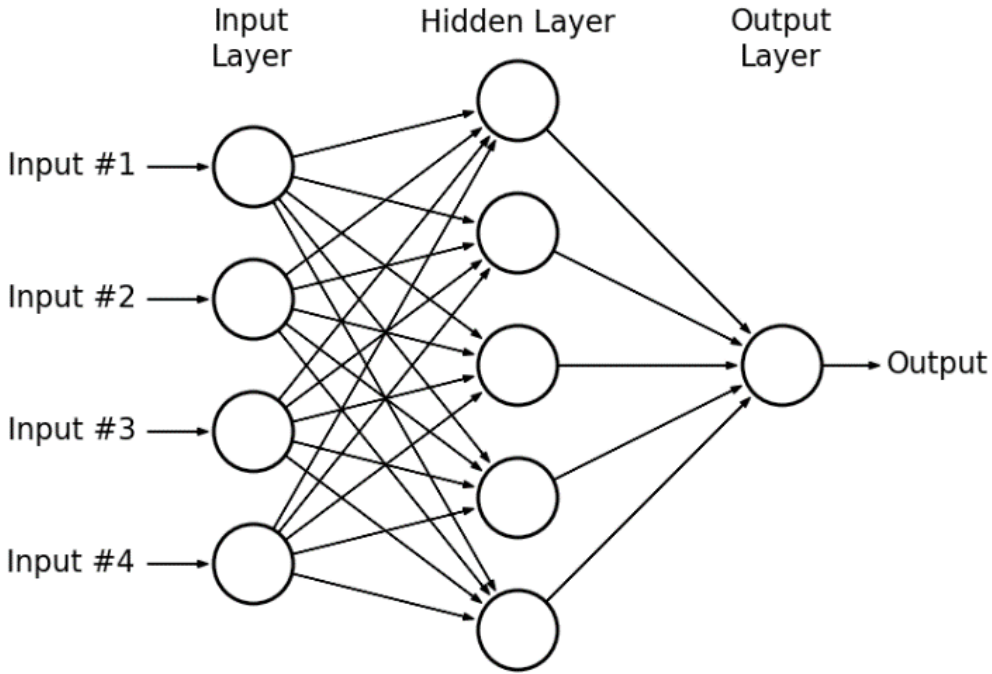
$$\sigma(x) = \frac{1}{1+e^{-x}}$$



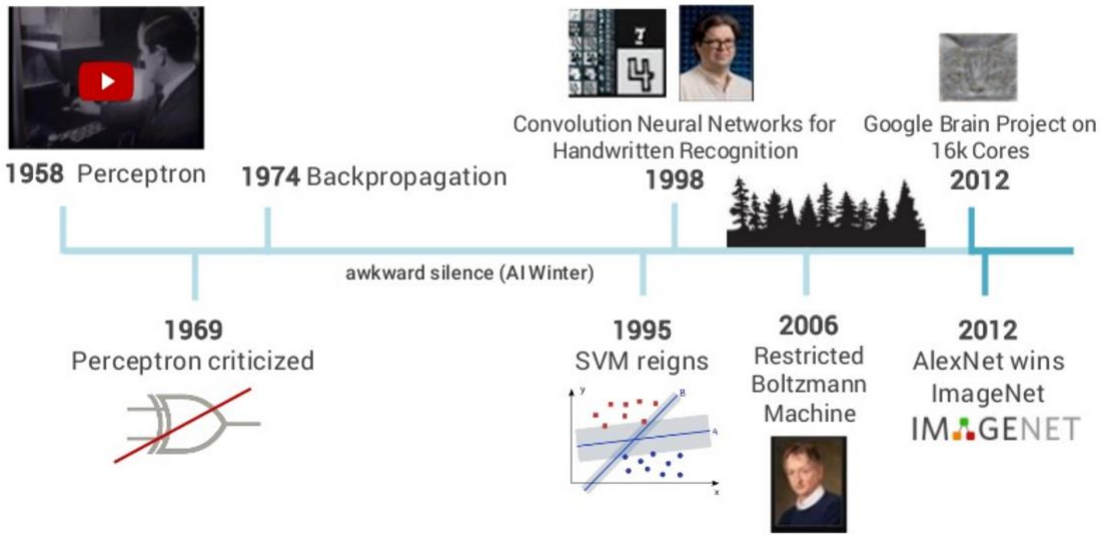
tanh

$$\tanh(x)$$

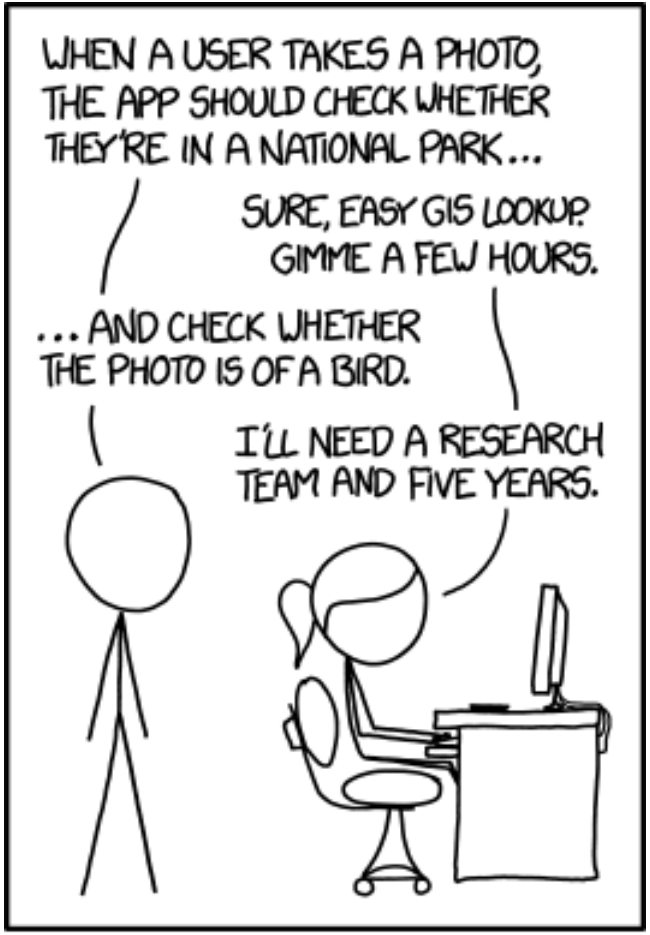




Curto histórico



Deep Learning

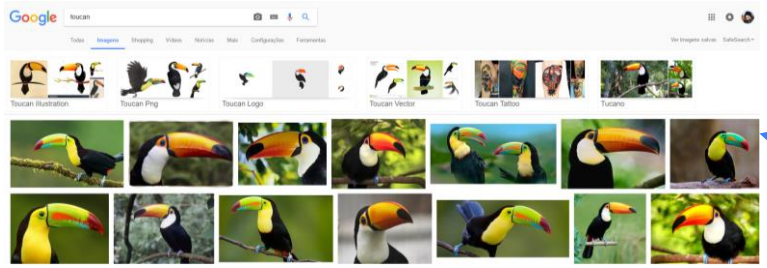

WHEN A USER TAKES A PHOTO, THE APP SHOULD CHECK WHETHER THEY'RE IN A NATIONAL PARK...
SURE, EASY GIS LOOKUP. GIMME A FEW HOURS.
...AND CHECK WHETHER THE PHOTO IS OF A BIRD.
I'LL NEED A RESEARCH TEAM AND FIVE YEARS.
IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

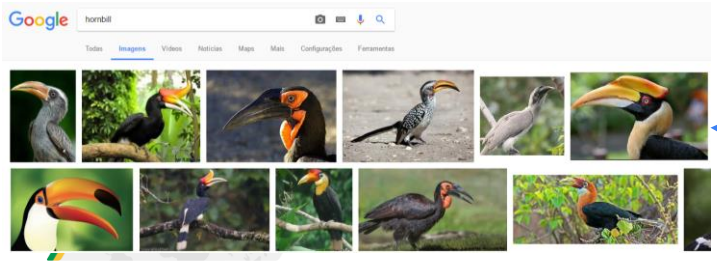
O você vê.

O que o computador “vê”.

50	68	78	95	...
67	21	23	42	
71	59	58	31	
47	19	29	39	
...				

pixels

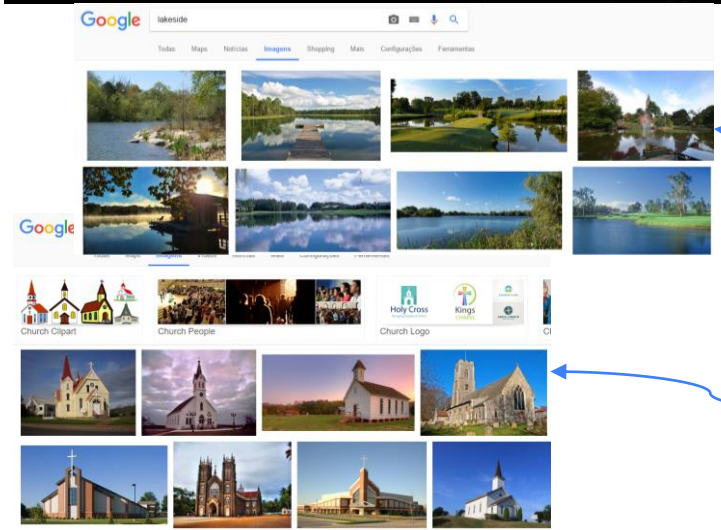





```python
import numpy as np
import keras
from keras.preprocessing import image
from keras.applications.inception_v3 import preprocess_input,
decode_predictions

model = keras.applications.InceptionV3(weights='imagenet')

img_path = 'aracari_castanho.jpg'
img = image.load_img(img_path, target_size=(299,299))
x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)
x = preprocess_input(x)

preds = model.predict(x)
# decode
print('Predicted:', decode_predictions(preds, top=3)[0])
```

Predicted: [('n01843383', 'toucan', 0.71278381), ('n01829413', 'hornbill', 0.16843531), ('n04146614', 'school_bus', 0.01657751)]

Toucan	71,27%
Hornbill	16,84%
School Bus	1,65%

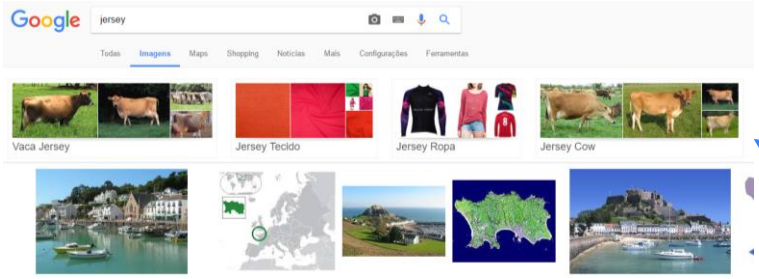



```python
import numpy as np
import keras
from keras.preprocessing import image
from keras.applications.inception_v3 import preprocess_input,
decode_predictions

model = keras.applications.InceptionV3(weights='imagenet')

img_path = 'arvore.jpg'
img = image.load_img(img_path, target_size=(299, 299))
x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)
x = preprocess_input(x)

preds = model.predict(x)
# decode
print('Predicted:', decode_predictions(preds, top=3)[0])
```

Predicted: [('n09332890', 'lakeside', 0.21762265), ('n03028079', 'church', 0.080397919), ('n09468604', 'valley', 0.078168809)]

Lakeside	21,76%
Church	8,03%
Valley	7,81%

```
import numpy as np
import keras
from keras.preprocessing import image
from keras.applications.inception_v3 import preprocess_input,
decode_predictions

model = keras.applications.InceptionV3(weights='imagenet')

img_path = 'alvaro.jpg'
img = image.load_img(img_path, target_size=(299, 299))
x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)
x = preprocess_input(x)

preds = model.predict(x)
# decode
print('Predicted:', decode_predictions(preds, top=3)[0])
```

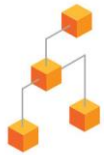


Predicted: [('n03595614', 'jersey', 0.51262307), ('n04584207', 'wig', 0.025850503), ('n03250847', 'drumstick', 0.021243958)]

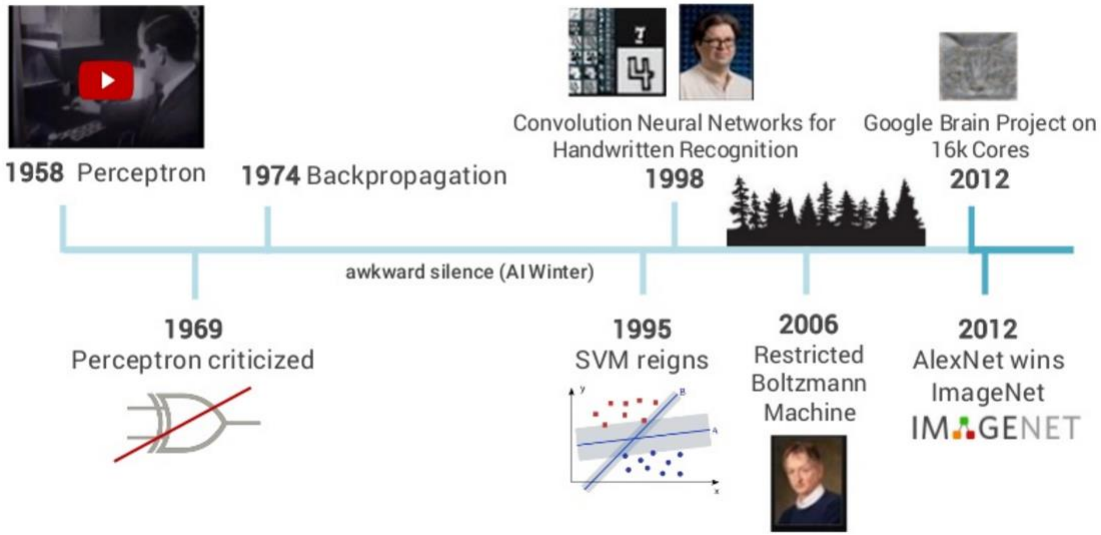
Jersey	51,26%
Wig	2,58%
Drumstick	2,12%

O que é deep learning?

- Aprendizagem de representação dos dados.
- Utiliza algoritmos hierárquicos com varias camadas.
- Se alimentado com muitos dados consegue generalizar o problema.
- Método de como as redes neurais aprendem.



Curto histórico

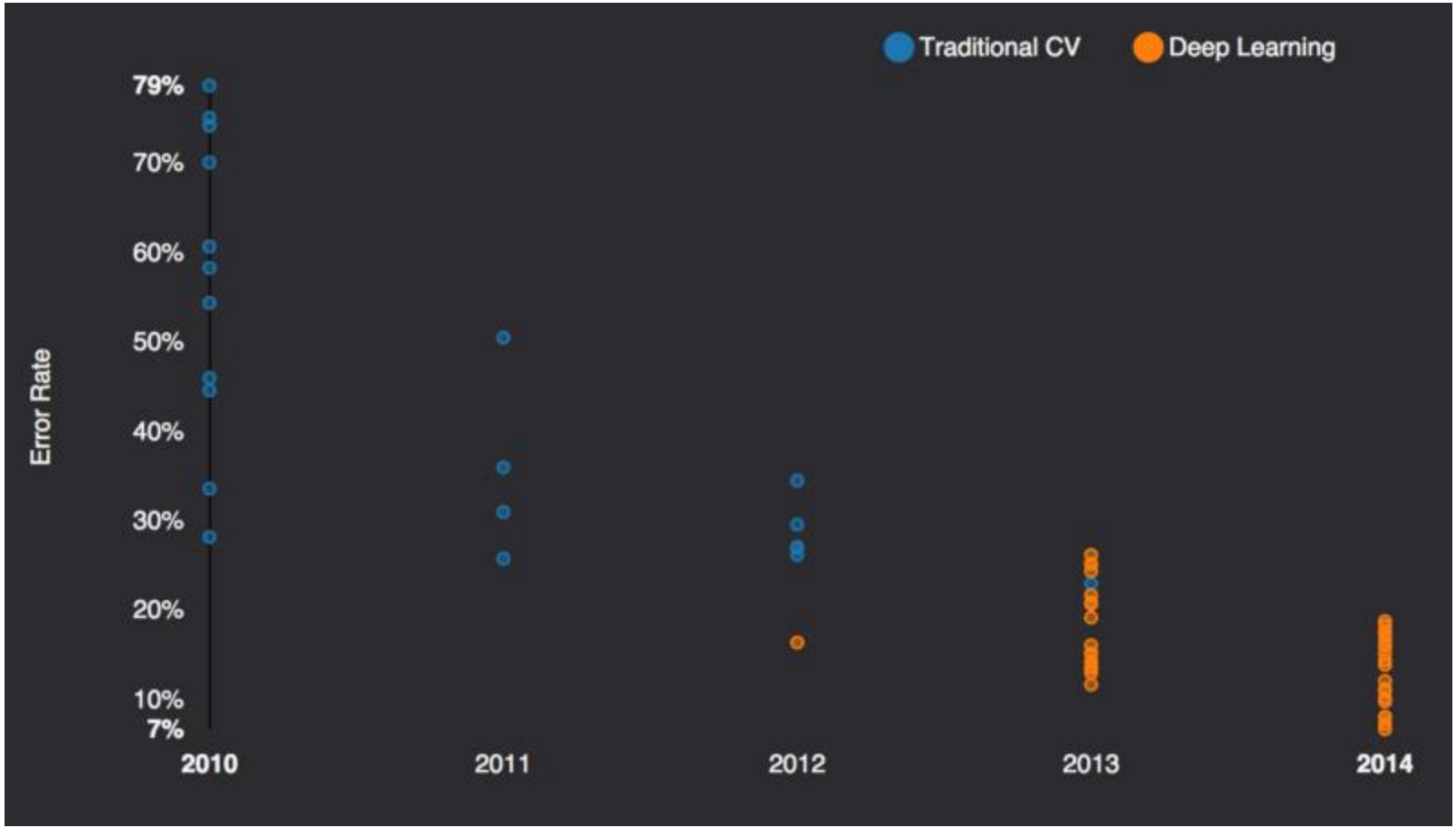


ImageNet Challenge

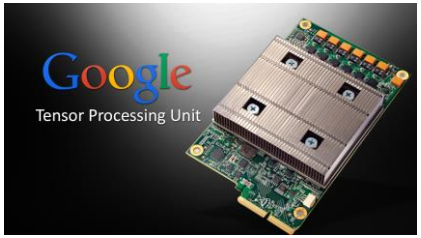
IMAGENET

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.

4


Traditional CV
Deep Learning
Error Rate
79%
70%
60%
50%
40%
30%
20%
10%
7%
2010
2011
2012
2013
2014

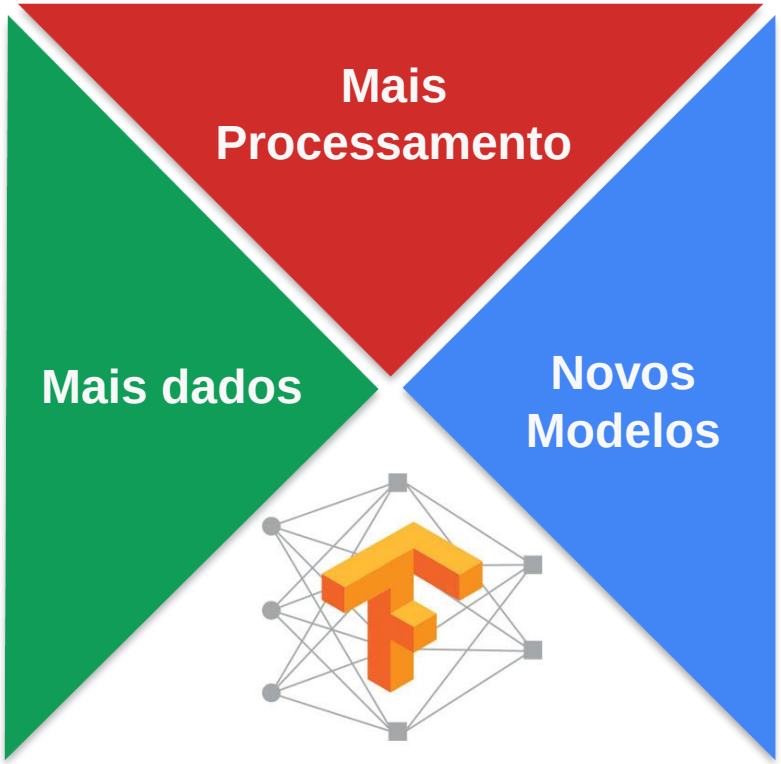
Deep Learning

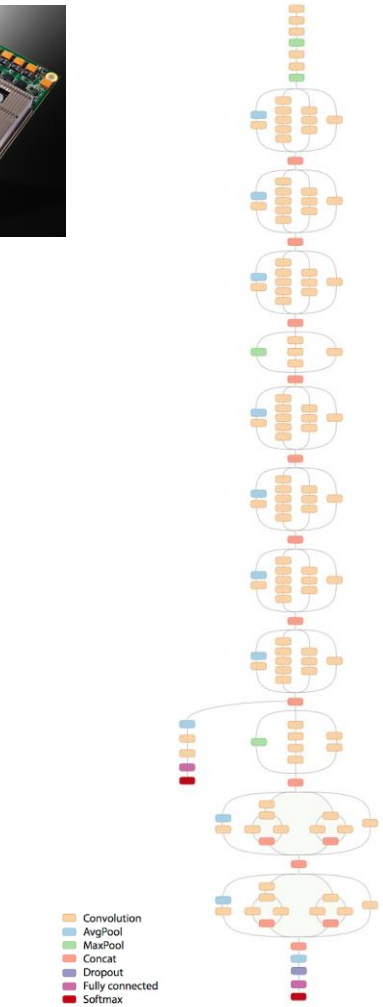


MNIST

IMAGENET

YouTube-8M

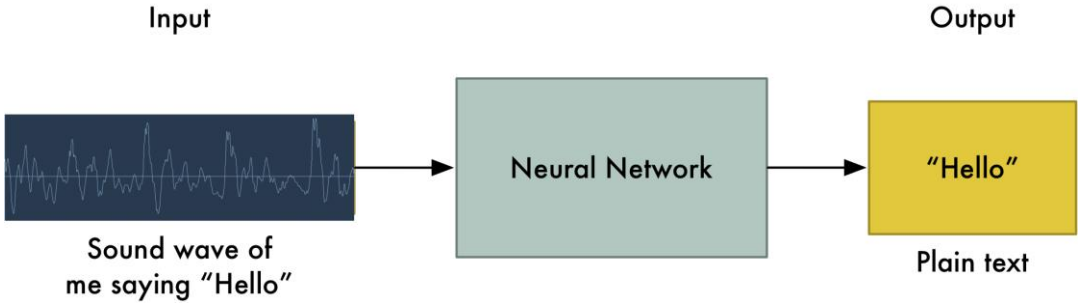




GDG Cuiabá

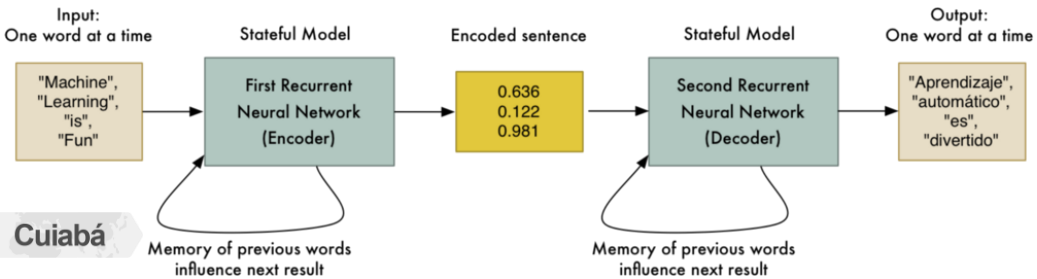
Deep Learning - Aplicações

- Reconhecimento de voz



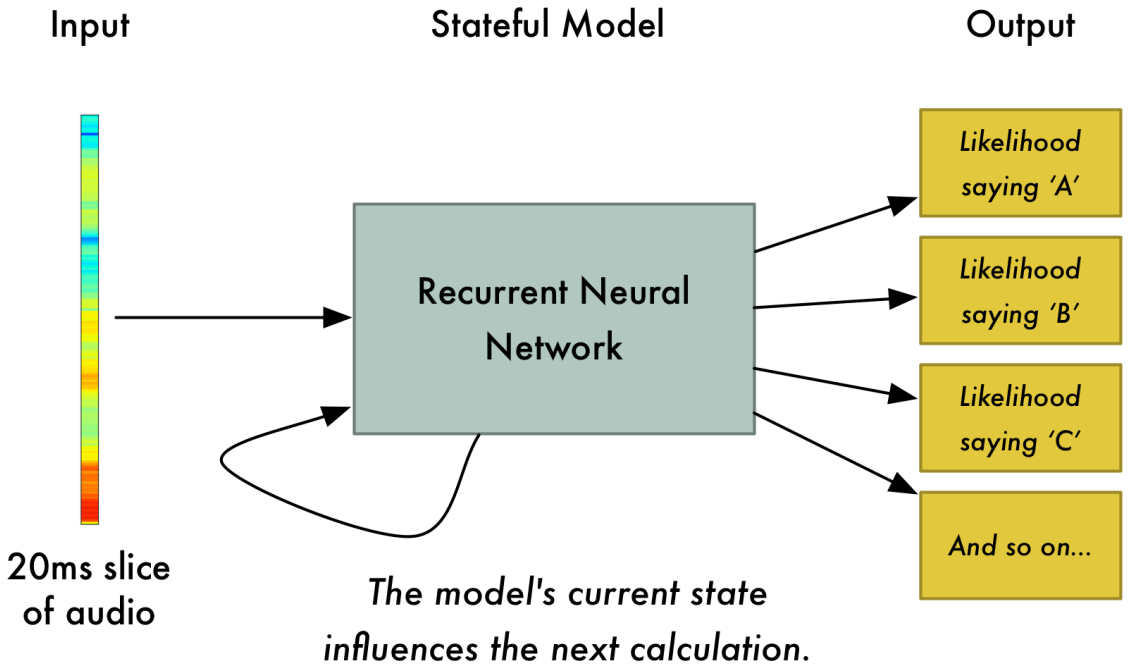
- Visão computacional

- Processamento de linguagem natural

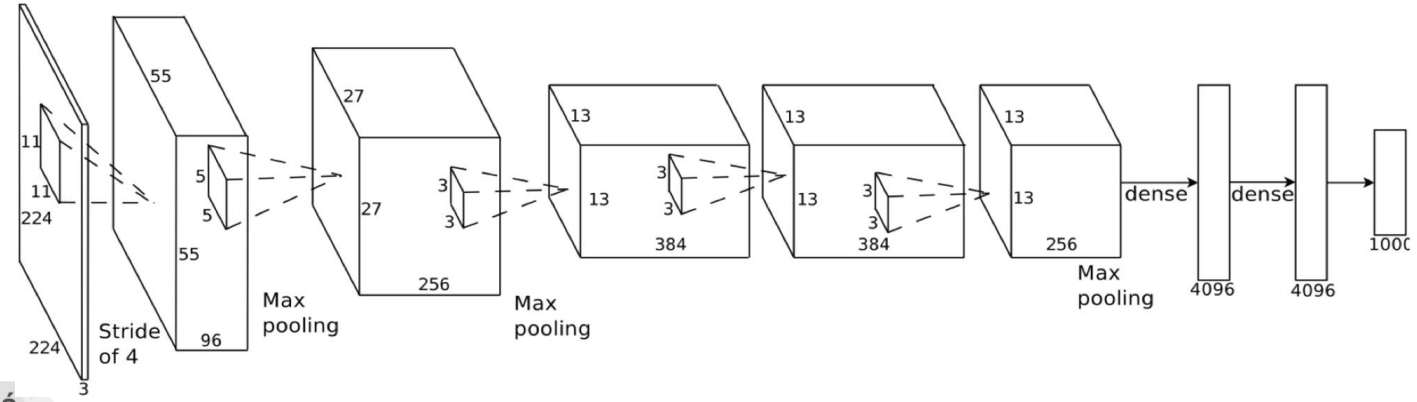


GDG Cuiabá

Reconhecimento de voz

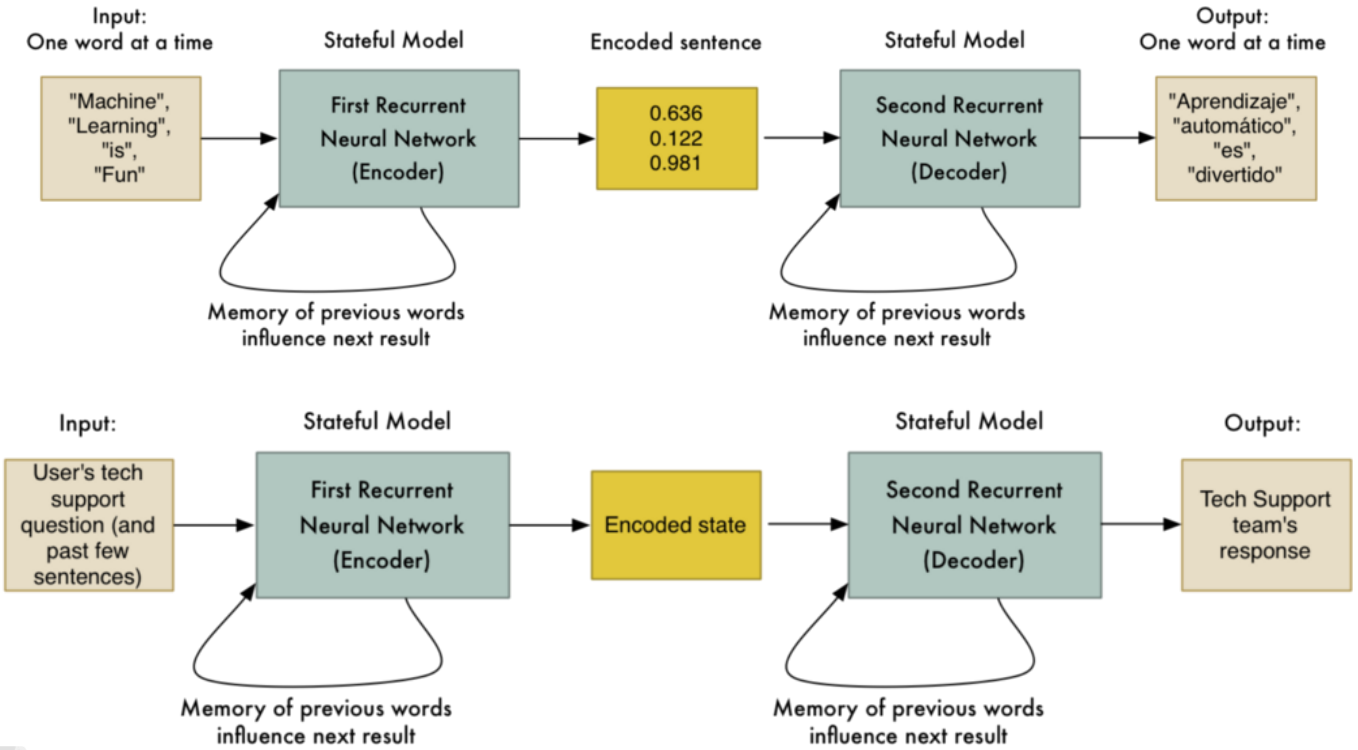


Visão Computacional

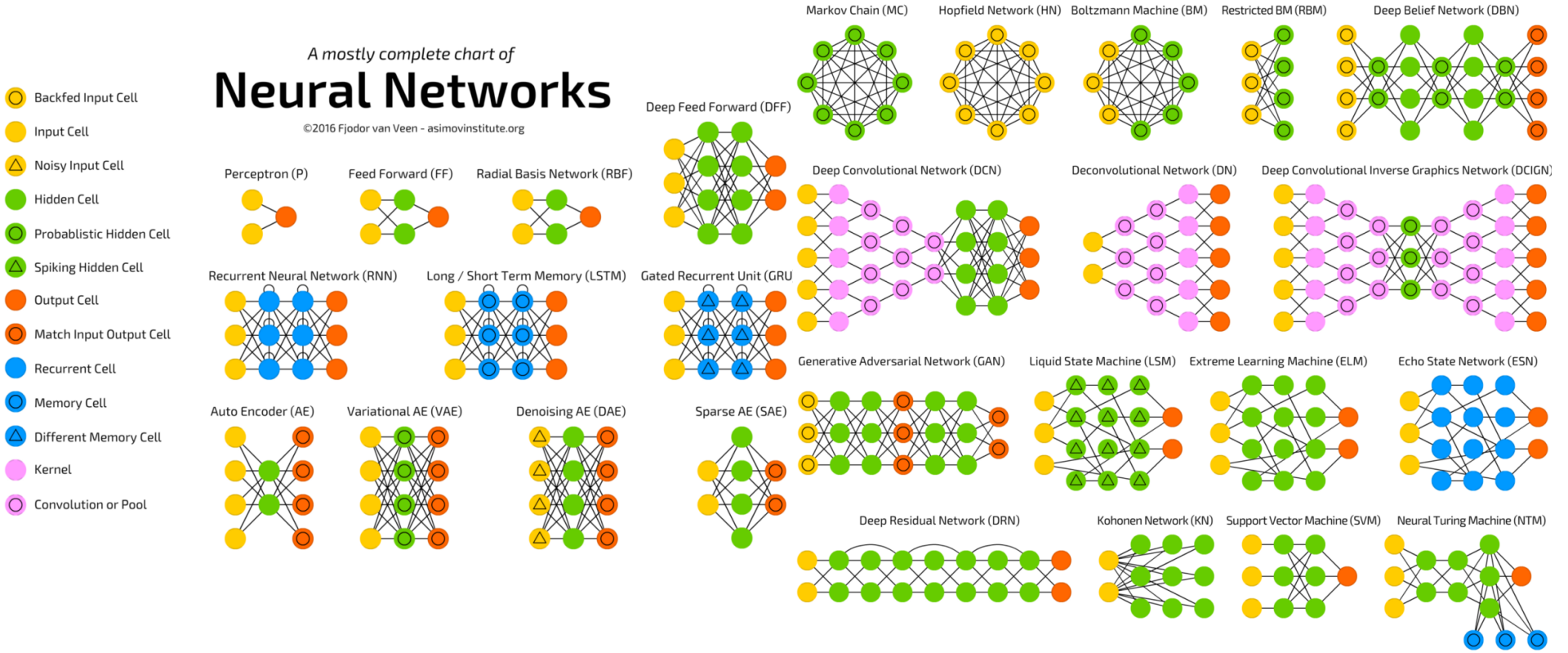


GDG Cuiabá

Processamento de linguagem natural

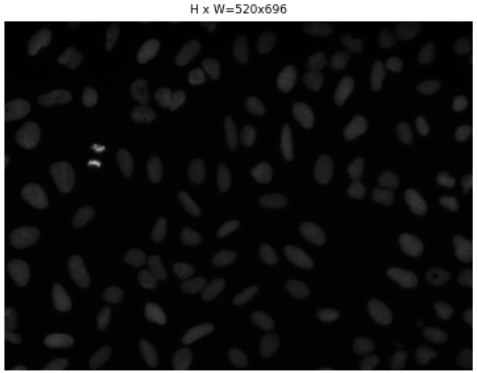


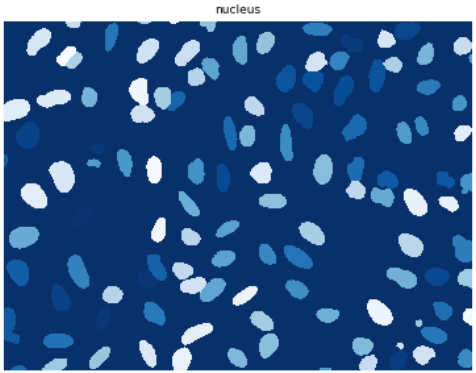
Model Zoo

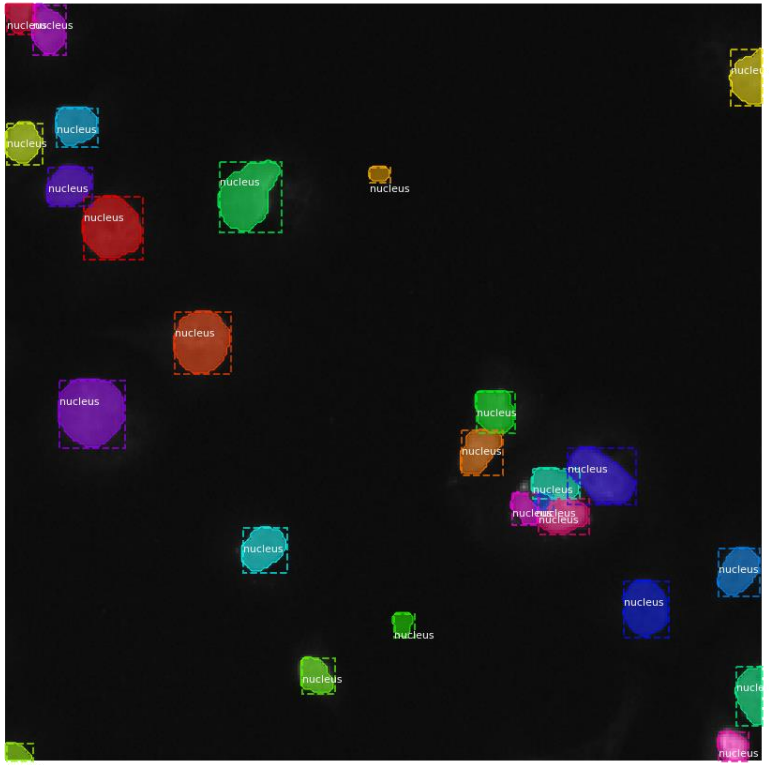

Backfed Input Cell
Input Cell
Noisy Input Cell
Hidden Cell
Probabilistic Hidden Cell
Spiking Hidden Cell
Output Cell
Match Input Output Cell
Recurrent Cell
Memory Cell
Different Memory Cell
Kernel
Convolution or Pool
A mostly complete chart of
Neural Networks
©2016 Fjodor van Veen - asimovinstitute.org
Perceptron (P)
Feed Forward (FF)
Radial Basis Network (RBF)
Deep Feed Forward (DFF)
Recurrent Neural Network (RNN)
Long / Short Term Memory (LSTM)
Gated Recurrent Unit (GRU)
Auto Encoder (AE)
Variational AE (VAE)
Denoising AE (DAE)
Sparse AE (SAE)
Markov Chain (MC)
Hopfield Network (HN)
Boltzmann Machine (BM)
Restricted BM (RBM)
Deep Belief Network (DBN)
Deep Convolutional Network (DCN)
Deconvolutional Network (DN)
Deep Convolutional Inverse Graphics Network (DCIGN)
Generative Adversarial Network (GAN)
Liquid State Machine (LSM)
Extreme Learning Machine (ELM)
Echo State Network (ESN)
Deep Residual Network (DRN)
Kohonen Network (KN)
Support Vector Machine (SVM)
Neural Turing Machine (NTM)

Utilização - kaggle

- Competição
  - https://www.kaggle.com/c/data-science-bowl-2018

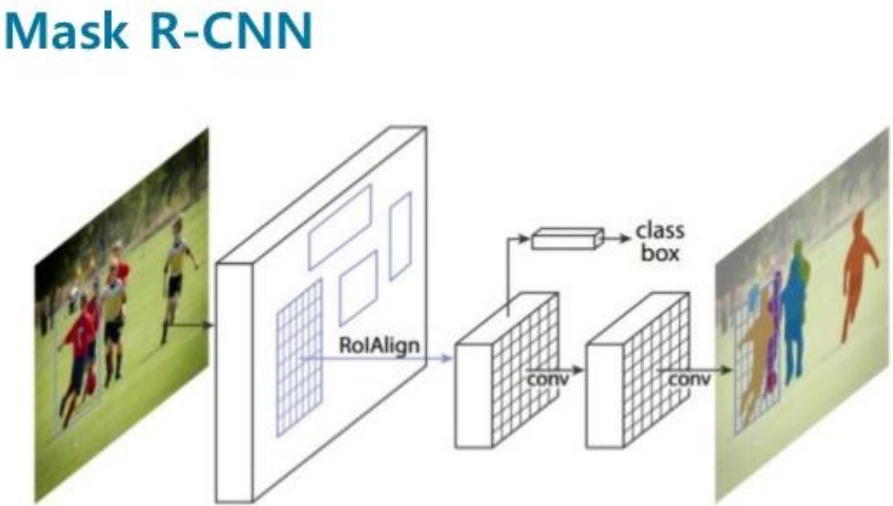


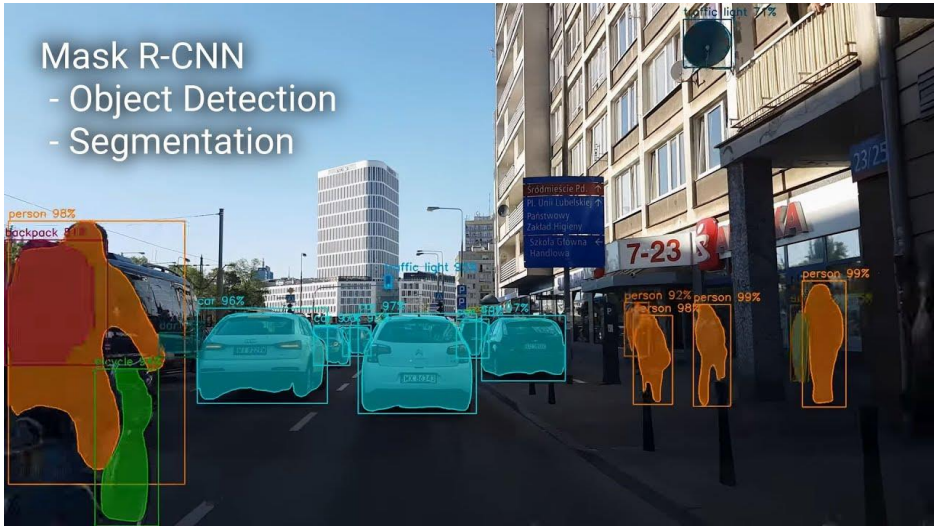




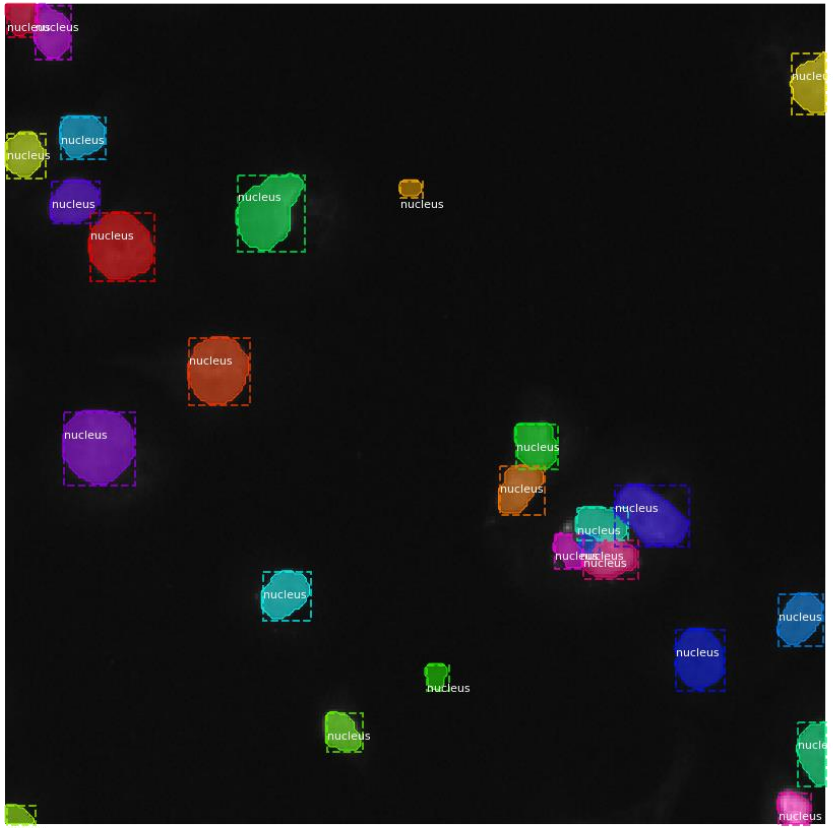
Utilização - kaggle

- Modelo utilizado – Mask R-CNN





Utilização - kaggle



Perguntas?