

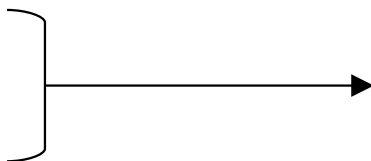
Entendendo e aplicando Inteligência Artificial - Parte II

Adriano Mendes

Inteligência Artificial

Algoritmos

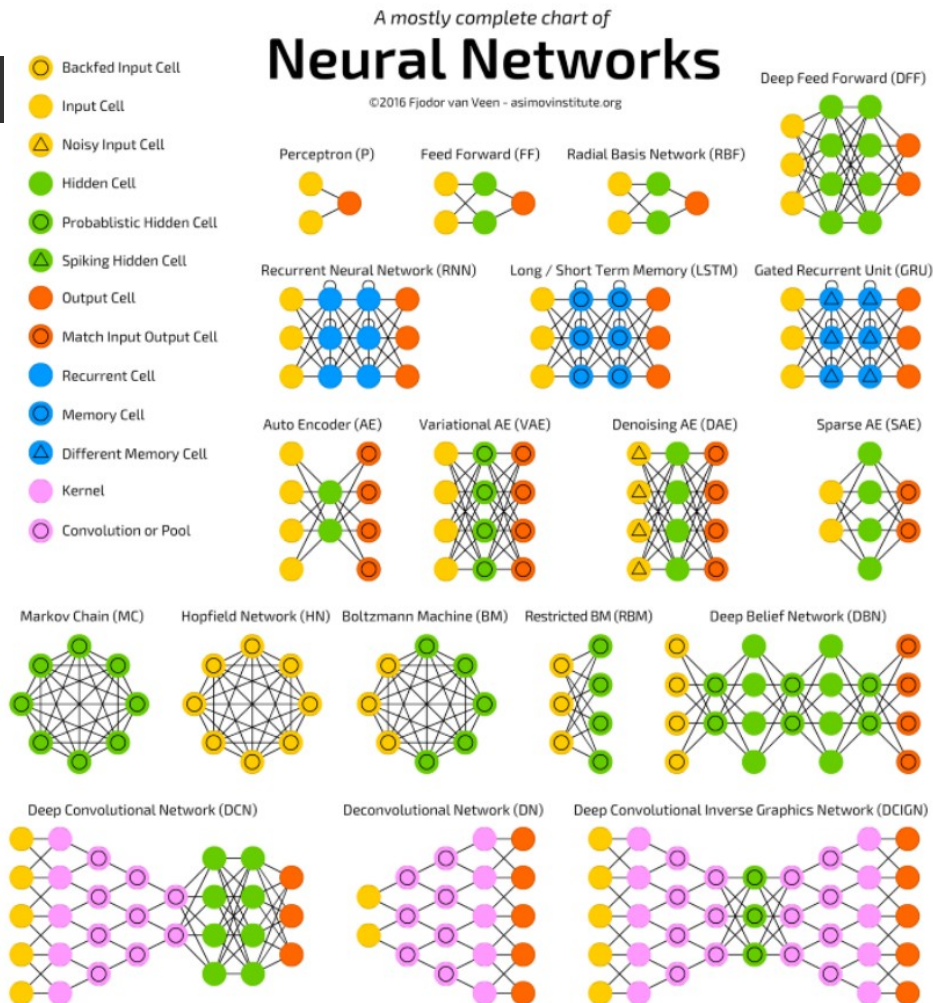
- Regressores
- SVM
- D-Trees
- Redes Neurais & Deep learning



Nota:

Confira mais detalhes em:

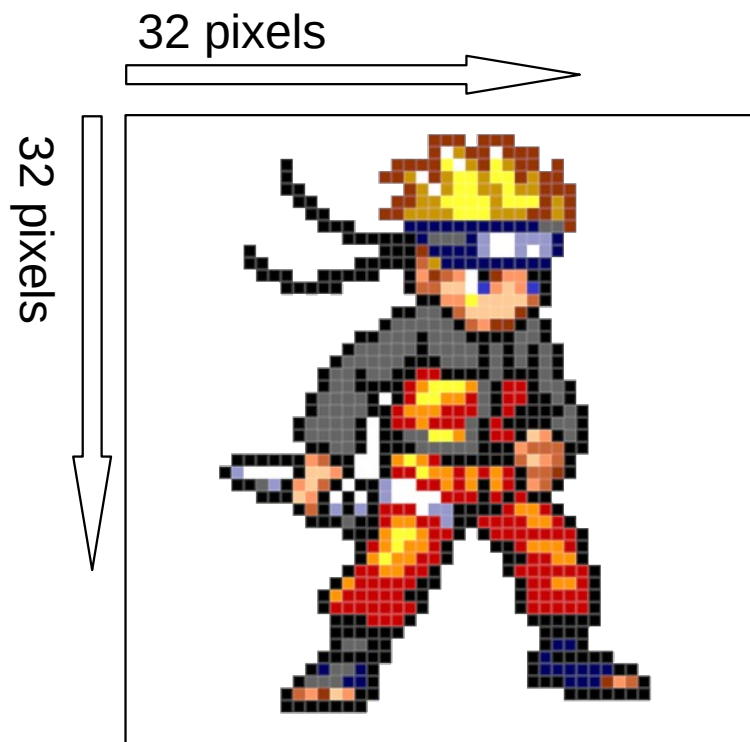
- <https://www.asimovinstitute.org/neural-network-zoo/>
- <http://deeplearningbook.com.br/>
- <https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464>
- <https://www.digitalvidya.com/blog/types-of-neural-networks/>



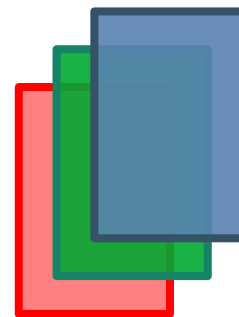
Rede Neural Convolucional (CNN)

- Convolutional Neural Network (CNN), são um tipo especializado de redes neurais
- Capacidade de extração de características (convolução)
- Amplamente usadas em problemas envolvendo imagens (Visão Computacional)
- Principais problemas
 - Classificação de imagens
 - Reconhecimento de objetos
 - Reconhecimento de pedestres

Imagens e pixels



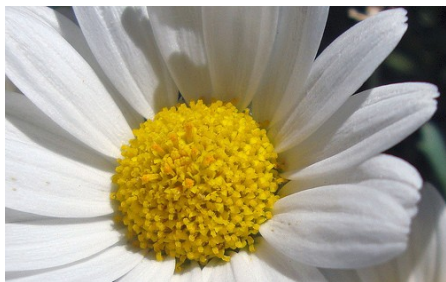
RGB



$$32 \times 32 \times 3 = 3.072 \text{ pixels}$$

Características

Margarida



Dente de leão



Rosa



Características

- Cores
- Numero de pétalas
- Tamanho das pétalas
- Tamanho do núcleo

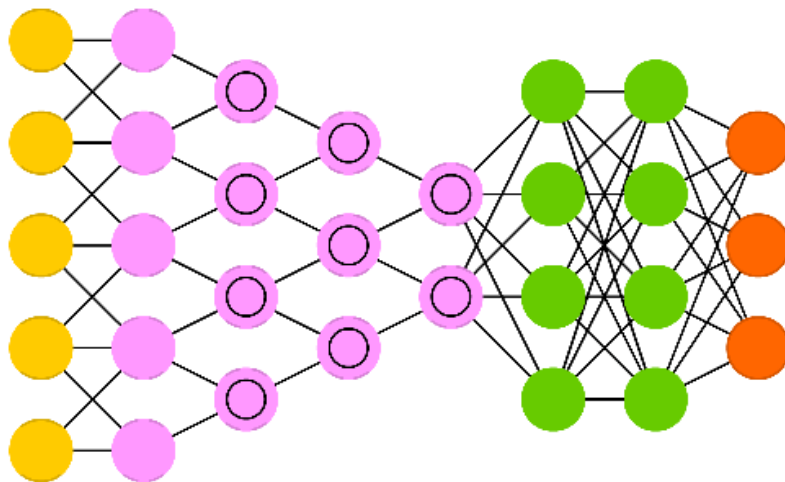
Cores	Num. Pétalas	Tam. Pétalas	Tam. Núcleo	CLASSE
0	25	5	20	Margarida
1	30	0,5	0,3	Dente de leão
1	15	20	5	Rosa
0	23	6	19	Margarida
1	27	0,6	0,4	Dente de leão
1	16	19	6	Rosa

Nota:

- Dataset disponível em: <https://www.kaggle.com/alxmamaev/flowers-recognition>

Rede Neural Densa & Convolucionais

$32 \times 32 \times 3 = 3.072$ pixels



- Não usam todos os pixels
- Usam uma rede neural tradicional, mas nas camadas iniciais transforma os dados
- Descobre as características mais relevantes



Redes Neurais Convolucionais (CNN)

- Quais são as melhores características?
- Em um problema de reconhecimento de faces
 - Localização do nariz
 - Distância dos olhos
 - Localização da boca
- Como diferenciar uma face humana de um animal?
- CNN consegue “descobrir” estas características



Redes Neurais Convolucionais (CNN)

Etapas executadas

- Passo 1: Convolução
- Passo 2: Pooling
- Passo 3: Flattening
- Passo 4: Rede Neural Densa

CNN – Passo 1 : Convolução

- Operador de convolução
 - Convolução é uma operação matemática aplicada a funções, e a ideia é entender como uma função afeta a outra ([detalhes](#))
 - No caso de redes neurais, a imagem e o kernel são as funções, e o resultado da operação representa a matriz de características
 - Sobre os kernels
 - [https://en.wikipedia.org/wiki/Kernel_\(image_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing))
 - Exemplo on-line
 - <http://setosa.io/ev/image-kernels/>

CNN – Passo 1 : Convolução

1	1	1	1	1	1	0
0	1	0	1	0	1	0
0	1	0	1	1	1	1
1	1	1	1	1	1	0
1	1	1	1	1	0	1
1	0	0	1	0	0	0
1	1	1	0	1	0	1

Imagem

X

1	0	1
1	1	0
1	0	1

Detector de características
(Filtro ou Kernel)

=

3	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Mata de
Características

$$(1 * 1) + (1 * 0) + (1 * 1) + (0 * 1) + (1 * 1) + (0 * 0) + (0 * 1) + (1 * 0) + (0 * 1) = 3$$

CNN – Passo 1 : Convolução

1	1	1	1	1	1	0
0	1	0	1	0	1	0
0	1	0	1	1	1	1
1	1	1	1	1	1	0
1	1	1	1	1	0	1
1	0	0	1	0	0	0
1	1	1	0	1	0	1

Imagem

X

1	0	1
1	1	0
1	0	1

Detector de características
(Filtro ou Kernel)

=

3	5	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Mata de
Características

$$(1 * 1) + (1 * 0) + (1 * 1) + (1 * 1) + (0 * 1) + (1 * 0) + (1 * 1) + (0 * 0) + (1 * 1) = 5$$

CNN – Passo 1 : Convolução

1	1	1	1	1	1	0
0	1	0	1	0	1	0
0	1	0	1	1	1	1
1	1	1	1	1	1	0
1	1	1	1	1	0	1
1	0	0	1	0	0	0
1	1	1	0	1	0	1

Imagem

X

1	0	1
1	1	0
1	0	1

Detector de características
(Filtro ou Kernel)

=

3	5	4	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Mata de
Características

$$(1 * 1) + (1 * 0) + (1 * 1) + (0 * 1) + (1 * 1) + (0 * 0) + (0 * 1) + (1 * 0) + (1 * 1) = 4$$

CNN – Passo 1 : Convolução

1	1	1	1	1	1	0
0	1	0	1	0	1	0
0	1	0	1	1	1	1
1	1	1	1	1	1	0
1	1	1	1	1	0	1
1	0	0	1	0	0	0
1	1	1	0	1	0	1

Imagem

X

1	0	1
1	1	0
1	0	1

Detector de características
(Filtro ou Kernel)

=

3	5	4	5	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Mata de
Características

$$(1 * 1) + (1 * 0) + (1 * 1) + (1 * 1) + (0 * 1) + (1 * 0) + (1 * 1) + (1 * 0) + (1 * 1) = 5$$

CNN – Passo 1 : Convolução

1	1	1	1	1	1	0
0	1	0	1	0	1	0
0	1	0	1	1	1	1
1	1	1	1	1	1	0
1	1	1	1	1	0	1
1	0	0	1	0	0	0
1	1	1	0	1	0	1

Imagem

X

1	0	1
1	1	0
1	0	1

Detector de características
(Filtro ou Kernel)

=

3	5	4	5	4
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Mata de
Características

$$(1 * 1) + (1 * 0) + (0 * 1) + (0 * 1) + (1 * 1) + (0 * 0) + (1 * 1) + (1 * 0) + (1 * 1) = 4$$

CNN – Passo 1 : Convolução

1	1	1	1	1	1	0
0	1	0	1	0	1	0
0	1	0	1	1	1	1
1	1	1	1	1	1	0
1	1	1	1	1	0	1
1	0	0	1	0	0	0
1	1	1	0	1	0	1

Imagem

X

1	0	1
1	1	0
1	0	1

Detector de características
(Filtro ou Kernel)

=

3	5	4	5	4
3	5	3	6	3
4	6	5	5	6
5	5	4	5	2
5	3	5	2	4

Mata de
Características

$$(1 * 1) + (0 * 0) + (1 * 1) + (0 * 1) + (0 * 1) + (0 * 0) + (1 * 1) + (0 * 0) + (1 * 1) = 4$$



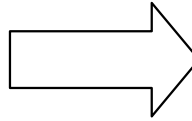
CNN – Passo 1 : Convolução

- Com o uso do filtro, consegue-se reduzir o tamanho dos dados para facilitar o processamento
- Partes da imagem podem se perder, mas a rede “guarda” as características principais
- Após todo o processo, ainda é aplicada a função de ativação “Relu”

CNN – Passo 1 : Convolução

1	1	1	1	1	1	0
0	1	0	1	0	1	0
0	1	0	1	1	1	1
1	1	1	1	1	1	0
1	1	1	1	1	0	1
1	0	0	1	0	0	0
1	1	1	0	1	0	1

Imagem



2	5	4	5	4
2	5	4	5	4
2	5	4	5	4
2	5	4	5	4
2	5	4	5	4
3	5	4	5	4
3	5	3	6	3
4	6	5	5	6
5	5	4	5	2
5	3	5	2	4

Mata de
Características

A convolução é a aplicação de vários kenels a imagem original



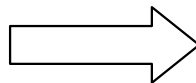
CNN – Passo 2 : Pooling

- A camada de Pooling
 - Tem a função de enfatizar as características principais
 - Assim como na convolução, Pooling é uma operação aplicada em cima dos valores da matriz (imagem)

CNN – Passo 2 : Pooling

1	2	1	2	0
1	2	1	1	1
2	1	1	1	2
2	1	2	0	3
3	1	3	0	3

Feature map



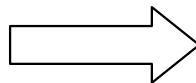
2	0	0
0	0	0
0	0	0

Pooled feature
map

CNN – Passo 2 : Pooling

1	2	1	2	0
1	2	1	1	1
2	1	1	1	2
2	1	2	0	3
3	1	3	0	3

Mata de
Características



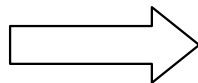
2	2	0
0	0	0
0	0	0

Pooled feature
map

CNN – Passo 2 : Pooling

1	2	1	2	0
1	2	1	1	1
2	1	1	1	2
2	1	2	0	3
3	1	3	0	3

Mata de
Características



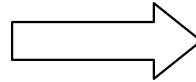
2	2	1
0	0	0
0	0	0

Pooled feature
map

CNN – Passo 2 : Pooling

1	2	1	2	0
1	2	1	1	1
2	1	1	1	2
2	1	2	0	3
3	1	3	0	3

Mata de
Características



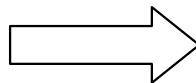
2	2	1
2	0	0
0	0	0

Pooled feature
map

CNN – Passo 2 : Pooling

1	2	1	2	0
1	2	1	1	1
2	1	1	1	2
2	1	2	0	3
3	1	3	0	3

Mata de
Características



2	2	1
2	2	3
3	3	3

Pooled feature
map



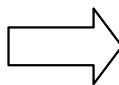
CNN – Passo 2 : Pooling

- Camada de Pooling
 - Diminui a dimensão dos dados
 - Reduz o overfitting e ruídos nos dados
 - Podem ser max pooling, mean pooling, lembrando que sempre queremos maximizar as características principais

CNN – Passo 2 : Pooling

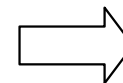
1	1	1	1	1	1	0
0	1	0	1	0	1	0
0	1	0	1	1	1	1
1	1	1	1	1	1	0
1	1	1	1	1	0	1
1	0	0	1	0	0	0
1	1	1	0	1	0	1

Imagem



2	5	4	5	4
3	5	4	5	4
3	5	4	5	4
3	5	4	5	4
3	5	4	5	4
3	5	4	5	4
3	5	4	5	4

Feature map

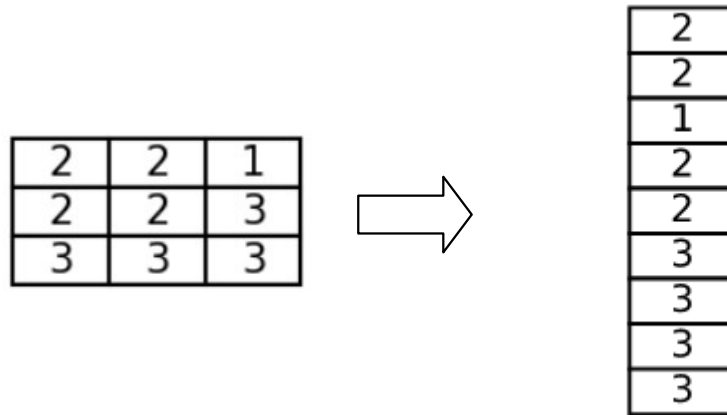


2	2	1
2	2	1
2	2	1
2	2	1
2	2	1
2	2	1
2	2	1

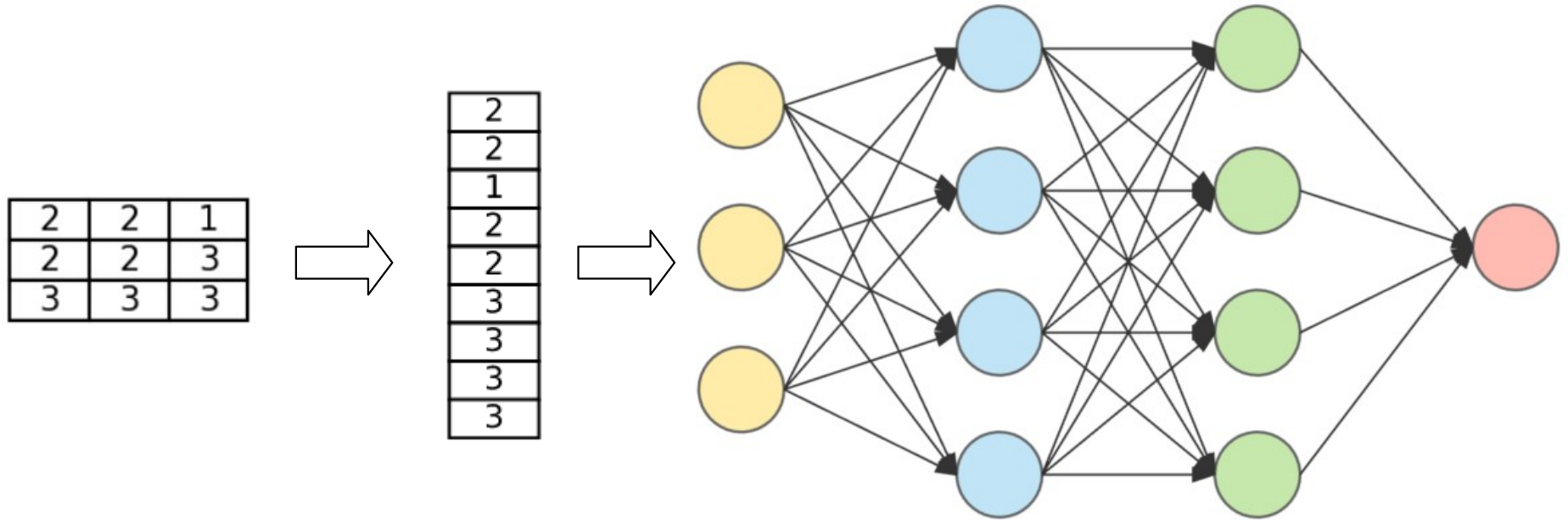
Pooled feature map

CNN – Passo 3 : Flattening

- A camada de Flattening
 - Basicamente aqui os dados em forma matricial são convertidos para serem a entrada da rede neural densa

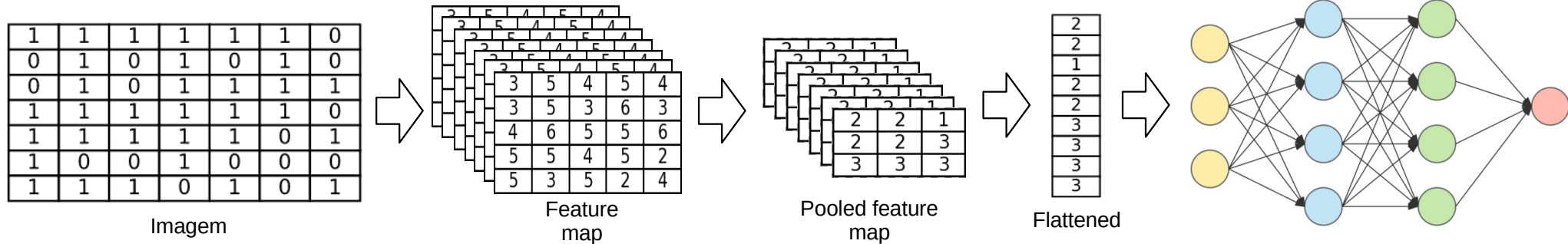


CNN - Passo 4 : Rede Neural Densa



CNN

- Vendo na prática um exemplo
 - Mostrando uma CNN funcionando
<https://scs.ryerson.ca/~aharley/vis/conv/>



Experimento 2

- Ideia
 - Realizar o reconhecimento de roupas utilizando CNN implementada por Tensorflow
- Recursos
 - Linguagem Python¹ via distribuição Anaconda² e frameworks Tensorflow³
 - Usando a base de dados Fashion MNIST⁴
 - Fazendo uso de Jupyter Notebooks usando a IDE do VS Code⁵

(1) <https://www.python.org/>

(2) <https://www.anaconda.com/>

(3) <http://www.Tensorflow.com>

(4) <https://www.openml.org/d/40996>

(5) <https://code.visualstudio.com/> e <https://code.visualstudio.com/docs/python/data-science-tutorial>

Experimento 3

- Ideia
 - Realizar o reconhecimento de objetos utilizando Fast R-CNN implementada por Tensorflow
- Recursos
 - Linguagem Python via distribuição Anaconda e frameworks Tensorflow
 - Fazendo uso de Jupyter Notebooks usando a IDE do VS Code

Referências

SITES

- Python.org
<https://www.python.org/https://www.python.org/>
- OpenML.org
<https://www.openml.org/>
- Data Science Academy
<https://www.datascienceacademy.com.br/pages/home>
- Towards Data Science
<https://towardsdatascience.com/perceptron-learning-algorithm-d5db0deab975>
- IA Experts
<https://iaexpert.academy/>



Referências

Livros

- Katia Faceli, **Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina** Gen, 2011
- Russell Stuart, **Artificial Intelligence, A modern approach**, 3ª edição, Pearson, 2010
- Jake VanderPlas, **Python Data Science Handbook**, O'Reilly, 2016
- Joel Gruss, **Data Science from Scratch**, O'Reilly, 2015