

Tópicos Especiais: Inteligência Artificial

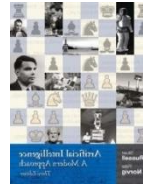
Deep Learning

Bibliografia

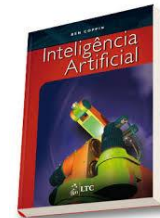
- Inteligência Artificial

- Russell & Norvig

- Site: <http://aima.cs.berkeley.edu>



- Inteligência Artificial, Ben Coppin.



- Online Course: Zero to Deep Learning

Disponível em: [udemy.com/zero-to-deep-learning](https://www.udemy.com/zero-to-deep-learning)

Deep Learning

- Estado da arte
 - Popular a partir de 2016
- Deep Learning = Aprendizagem de máquina (machine learning)
- Aplicação em:
 - Imagens
 - Texto
 - Som
 - Séries históricas
- No estado atual (2017) se equiparam ao ser humano no reconhecimento de imagens.
 - Veja: Facebook DeepFace
Disponível em: pt.slideshare.net/EmanueleSantellani/facebook-deep-face-72711694
 - Artigo: <https://research.fb.com/publications/deepface-closing-the-gap-to-human-level-performance-in-face-verification/>

DeepFace

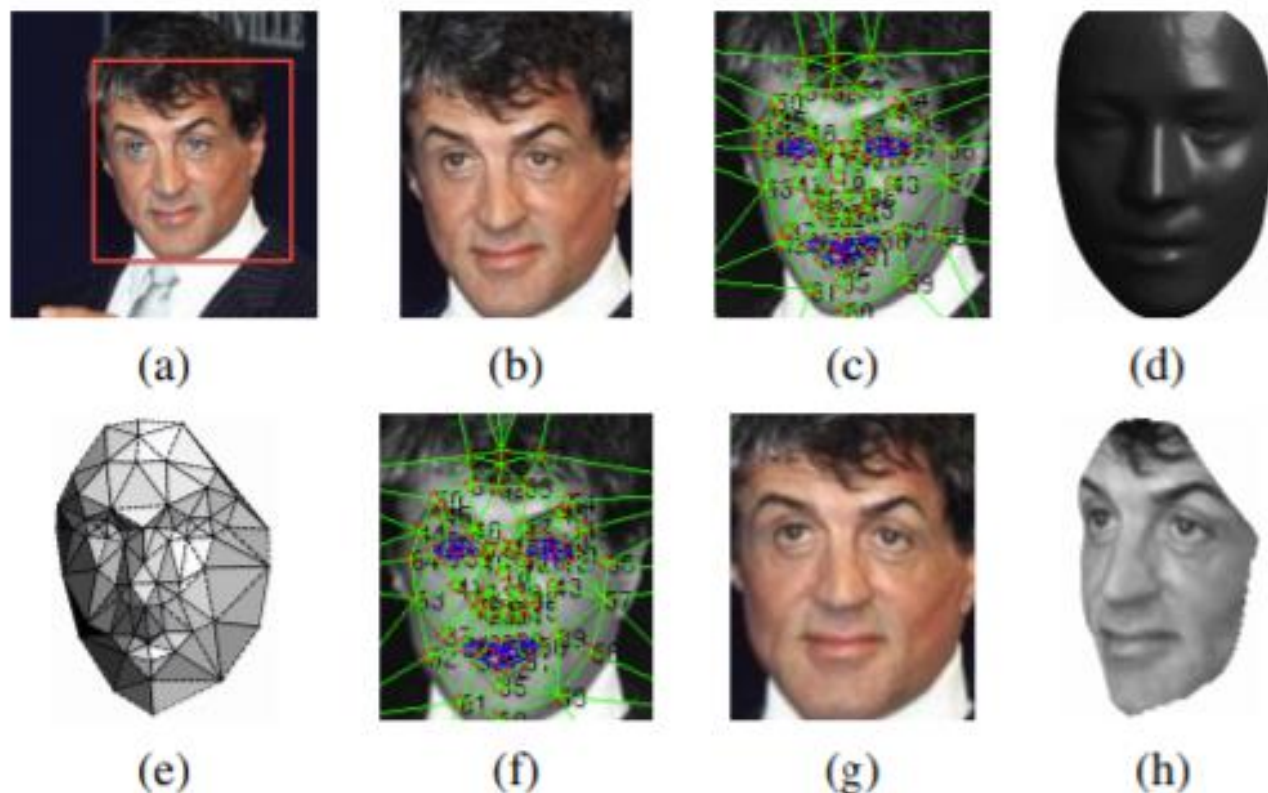


Figure 1. Alignment pipeline. (a) The detected face, with 6 initial fiducial points. (b) The induced 2D-aligned crop. (c) 67 fiducial points on the 2D-aligned crop with their corresponding Delaunay triangulation, we added triangles on the contour to avoid discontinuities. (d) The reference 3D shape transformed to the 2D-aligned crop image-plane. (e) Triangle visibility w.r.t. to the fitted 3D-2D camera; darker triangles are less visible. (f) The 67 fiducial points induced by the 3D model that are used to direct the piece-wise affine warping. (g) The final frontalized crop. (h) A new view generated by the 3D model (not used in this paper).

DeepFace

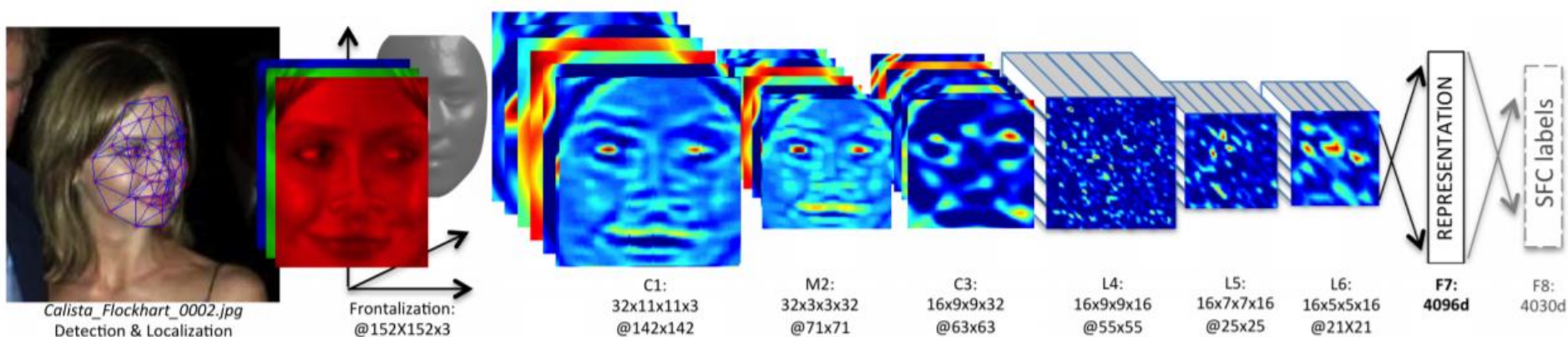


Figure 2. **Outline of the *DeepFace* architecture.** A front-end of a single convolution-pooling-convolution filtering on the rectified input, followed by three locally-connected layers and two fully-connected layers. Colors illustrate feature maps produced at each layer. The net includes more than 120 million parameters, where more than 95% come from the local and fully connected layers.

DeepFace (Estado da Arte)

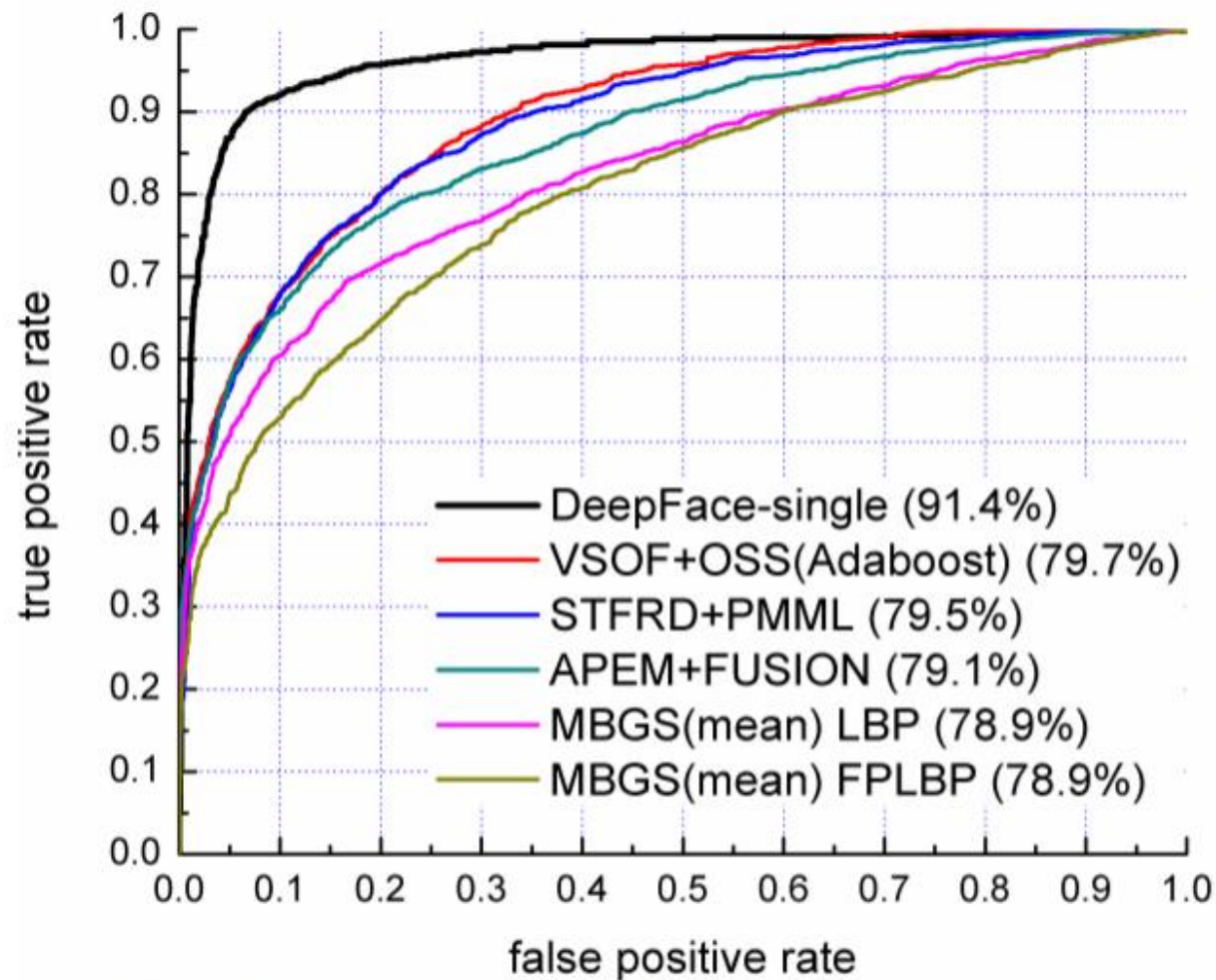


Figure 4. The ROC curves on the *YTF* dataset. Best viewed in color.

Outras aplicações

- Carros Autônomos
- Games
- Google Assistente
- Microsoft Cortana
- Apple Siri
- Aplicações médicas (reconhecimento de câncer de pele ou problemas na retina)
- Indústria
 - O sistema de ar-condicionado dos datacenters do Google é 40% mais econômico graças ao Deep Learning.

Instalação do Anaconda

(Distribuição Python 3.6 com bibliotecas já configuradas)

Download for Your Preferred Platform



Windows



macOS



Linux

<https://anaconda.org>

Anaconda 5.0.1 For Windows Installer

Python 3.6 version *

↓ Download

[64-Bit Graphical Installer \(515 MB\)](#) ?

[32-Bit Graphical Installer \(420 MB\)](#)

Python 2.7 version *

↓ Download

[64-Bit Graphical Installer \(500 MB\)](#) ?

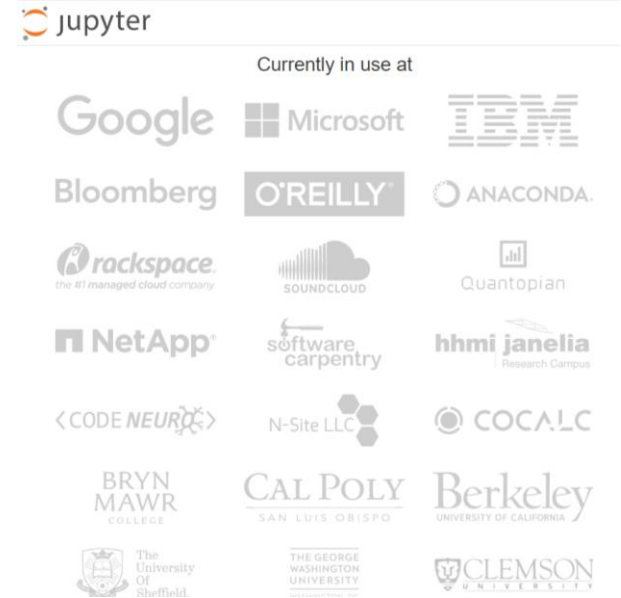
[32-Bit Graphical Installer \(403 MB\)](#)

Bibliotecas utilizadas

- Pip: ferramenta de instalação de pacotes e bibliotecas
- Numpy: framework base para cálculos científicos e matriciais
- Jupyter: execução de código python no browser
- Matplotlib: Plotagem de gráficos
- Scikit: ferramentas para mineração e análise de dados
- Scipy: Funções matemáticas e científicas
- Pandas: framework dados tabulados (Excel)
- Pillow: Python Image Library para Python 3
- Seaborn: visualização de dados estatísticos
- H5py: Para compressão de arquivos binários
- Tensorflow: Deep learning do Google
- Keras: Interface para simplificar o acesso ao Tensorflow

Instalação

- Link para Download
 - <http://jupyter.org>
- TensorFlow
 - <https://www.tensorflow.org/install>
- Código disponível no GitHub com exemplo:
 - Recomendo que façam este curso (nesta disciplina não cobriremos todos os tópicos)!
 - https://github.com/dataweekends/zero_to_deep_learning_udemy
 - `cd zero_to_deep_learning_udemy`
 - `conda env create`
 - `source activate ztdl` (Linux) ou `activate ztdl` (Windows)
 - `jupyter notebook`
 - Browser: `http://localhost:8888`

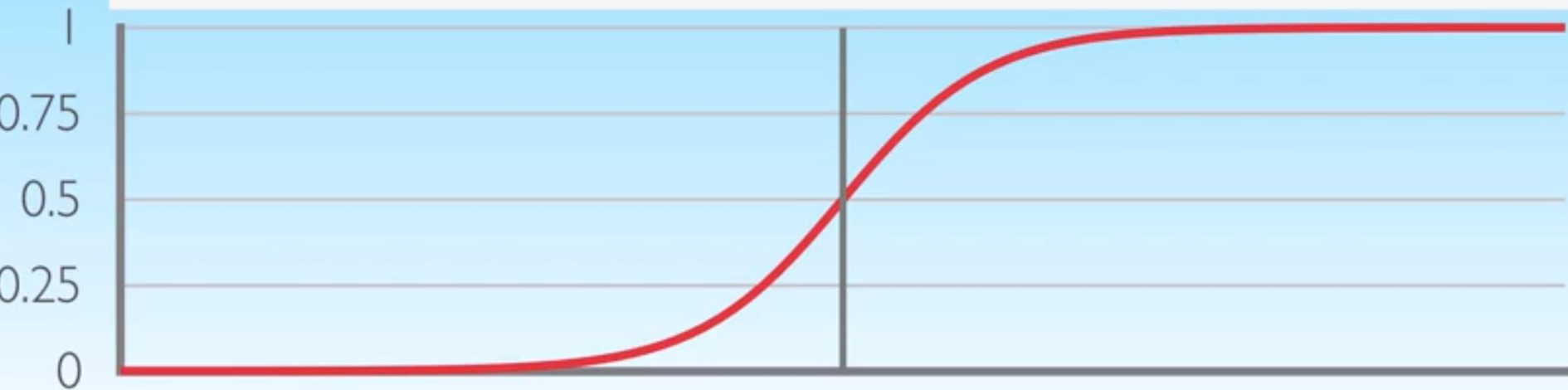


Funções de ativação para Deep Learning

FUNÇÕES DE ATIVAÇÃO

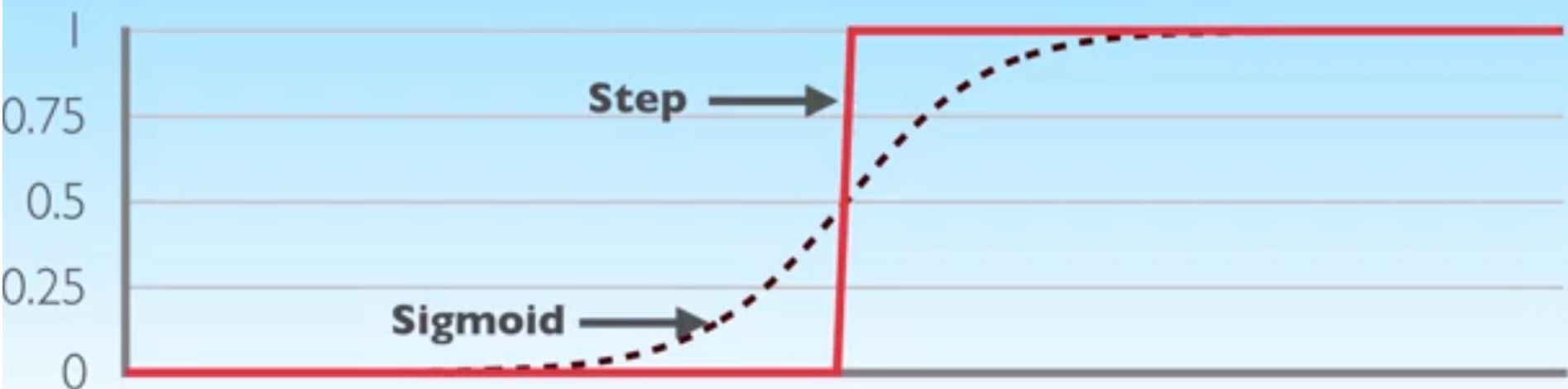
Sigmoid

```
def sigmoid(x):  
    return 1.0 / (1.0 + np.exp(-x))
```

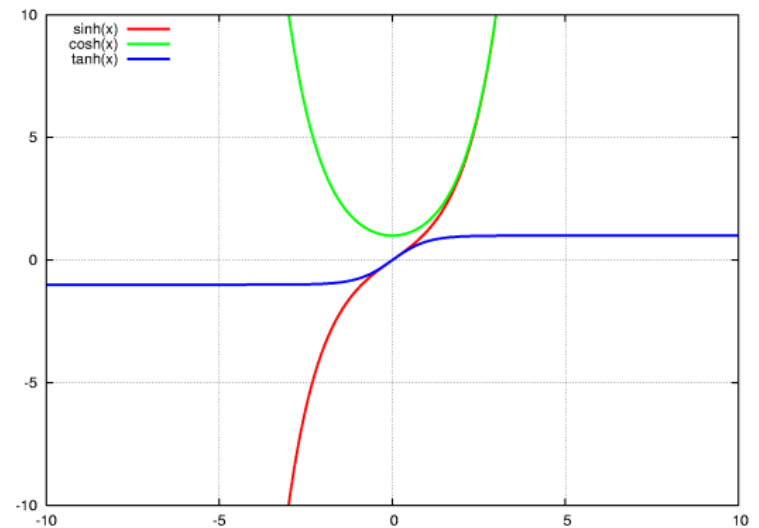
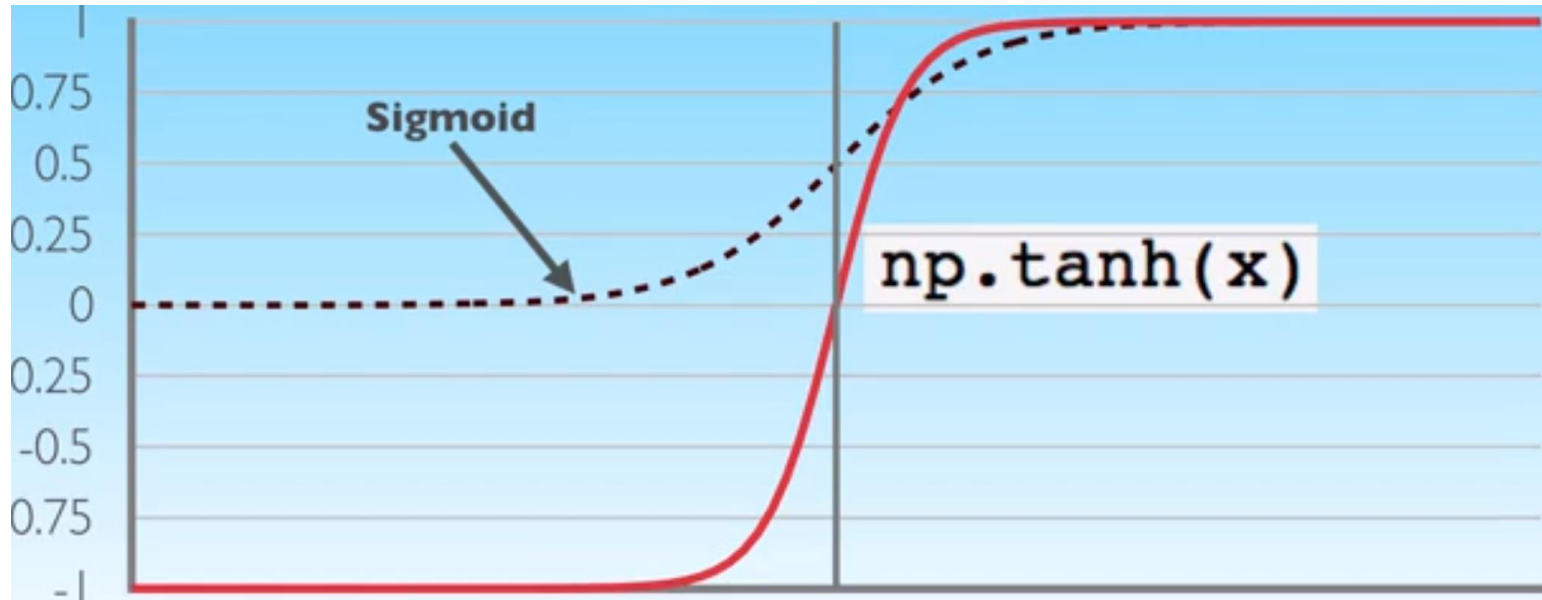


Step Function / Binária

```
def step(x):  
    return x > 0
```



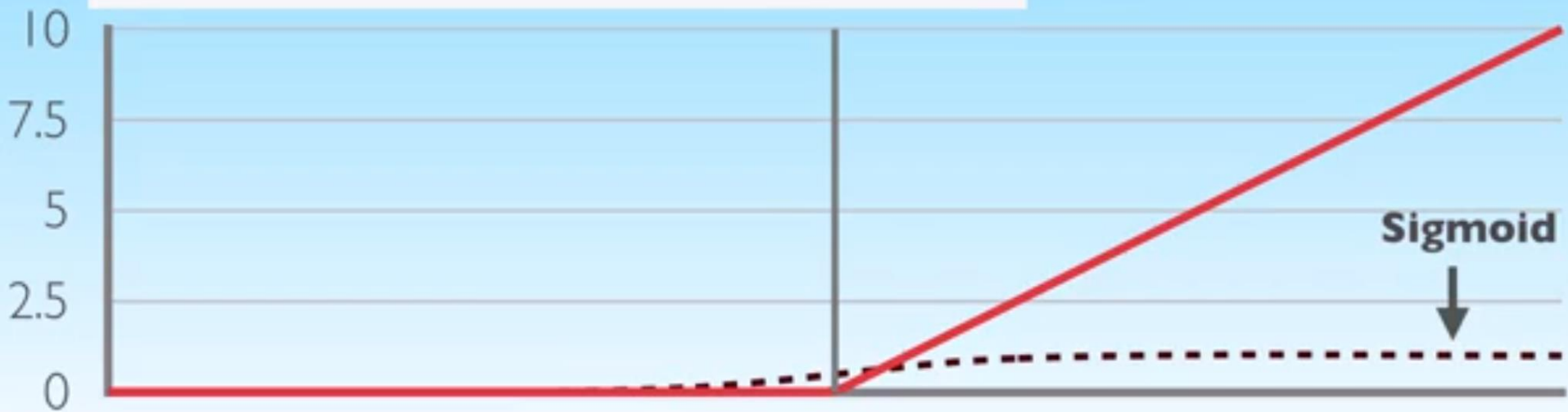
Tangente Hiperbólica / TANH



RELU

```
def relu(x):  
    return x * (x > 0)
```

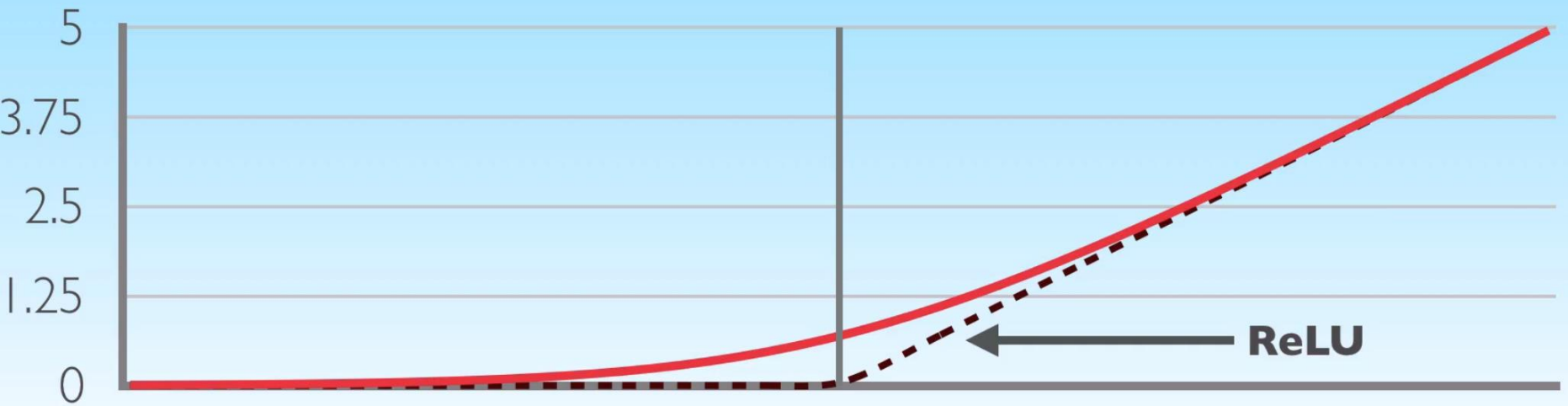
$$y = \max(0, x)$$



SOFTPLUS

```
def softplus(x):  
    return np.log1p(np.exp(x))
```

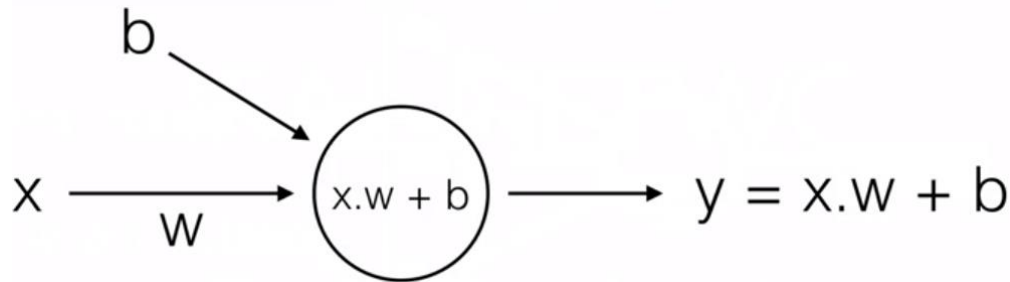
$$y = \log(1 + e^x)$$



Neurônios e topologia de redes neurais para Deep Learning

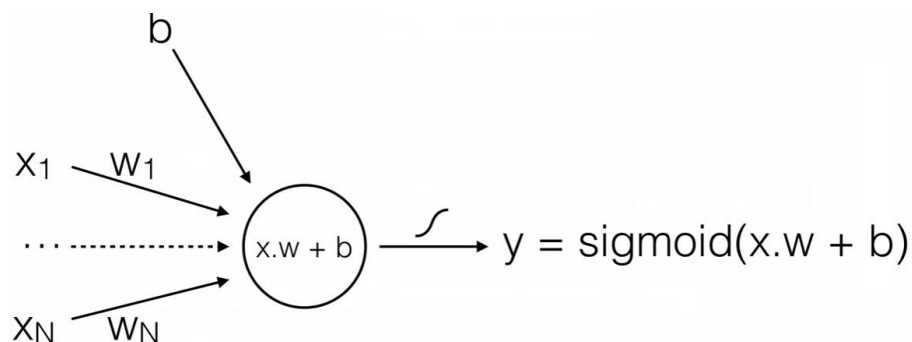
TOPOLOGIA DE NEURAL NETS

Regressão Linear



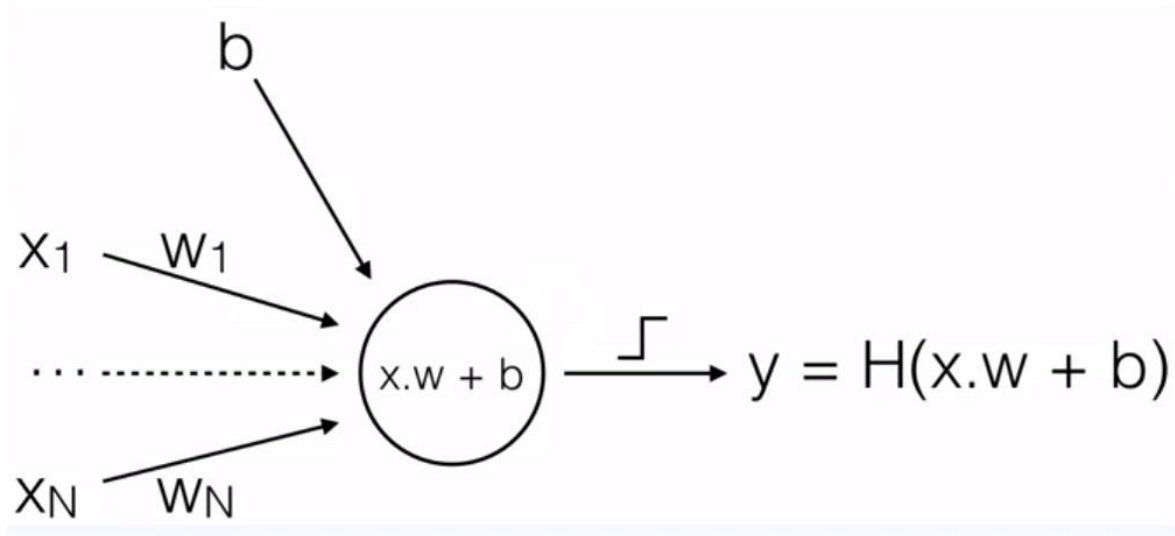
- Em estatística ou econometria, regressão linear é uma equação para se estimar a condicional (valor esperado) de uma variável y , dados os valores de algumas outras variáveis x .
- A regressão, em geral, tem como objetivo tratar de um valor que não se consegue estimar inicialmente.
- A regressão linear é chamada "linear" porque se considera que a relação da resposta às variáveis é uma função linear de alguns parâmetros. Os modelos de regressão que não são uma função linear dos parâmetros se chamam modelos de regressão não-linear.

Regressão logística

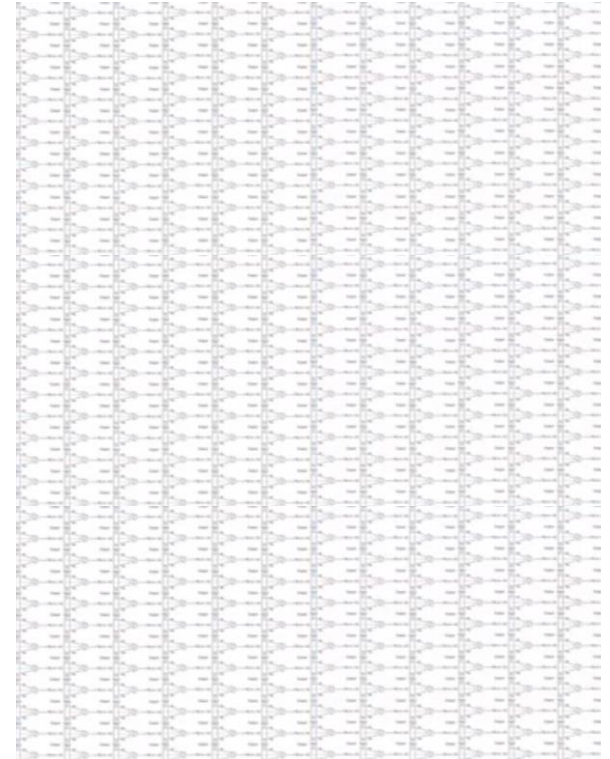
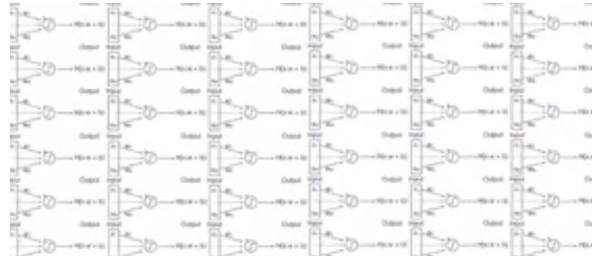
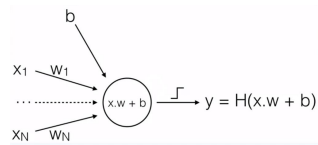


- A **regressão logística** é uma técnica [estatística](#) que tem como objetivo produzir, a partir de um conjunto de observações, um modelo que permita a predição de valores tomados por uma variável categórica, frequentemente binária, a partir de uma série de variáveis explicativas contínuas e/ou binárias.
- Em [medicina](#), permite por exemplo determinar os factores que caracterizam um grupo de indivíduos doentes em relação a indivíduos sãos.
- No domínio dos [seguros](#), permite encontrar fracções da clientela que sejam sensíveis a determinada política securitária em relação a um dado risco particular.
- Em instituições financeiras, pode detectar os grupos de risco para a subscrição de um crédito.
- Em [econometria](#), permite explicar uma variável discreta, como por exemplo as intenções de voto em actos eleitorais.

Perceptron



Deep/Fundo



- Lembre-se: Saída de uma camada é a entrada da outra...

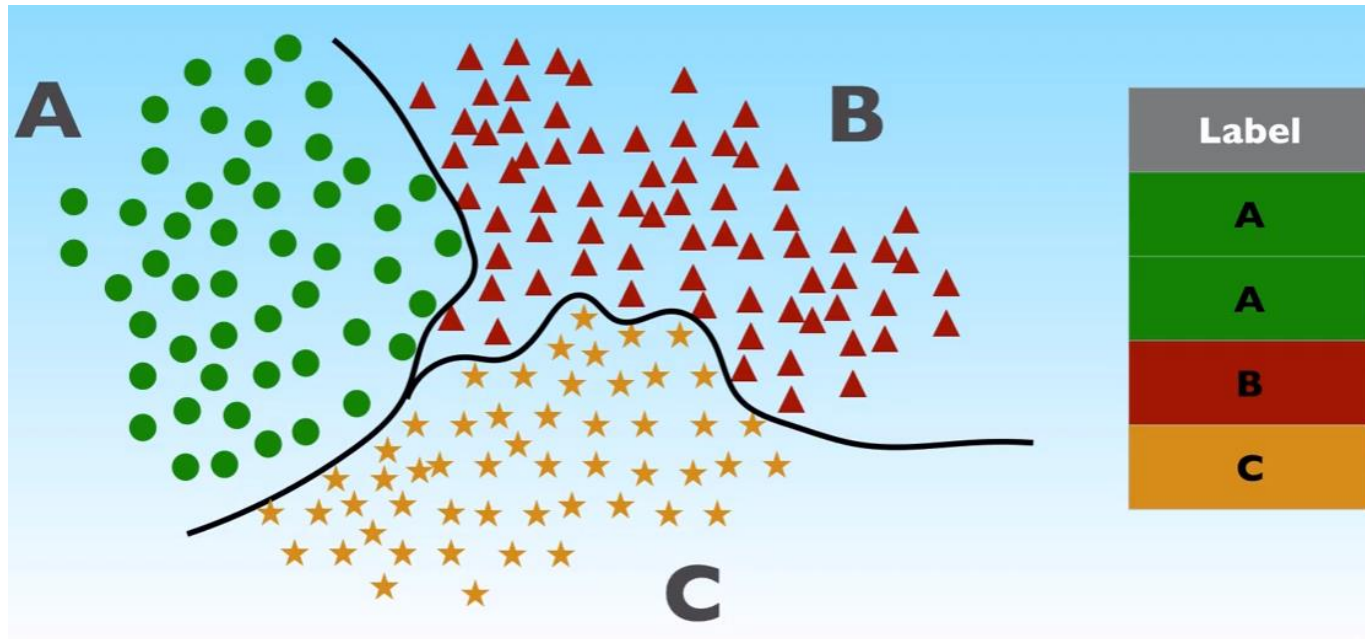
Vamos à prática...

- Deep Learning Intro no Jupyter Notebook
 - Testes
 - Explicação do código

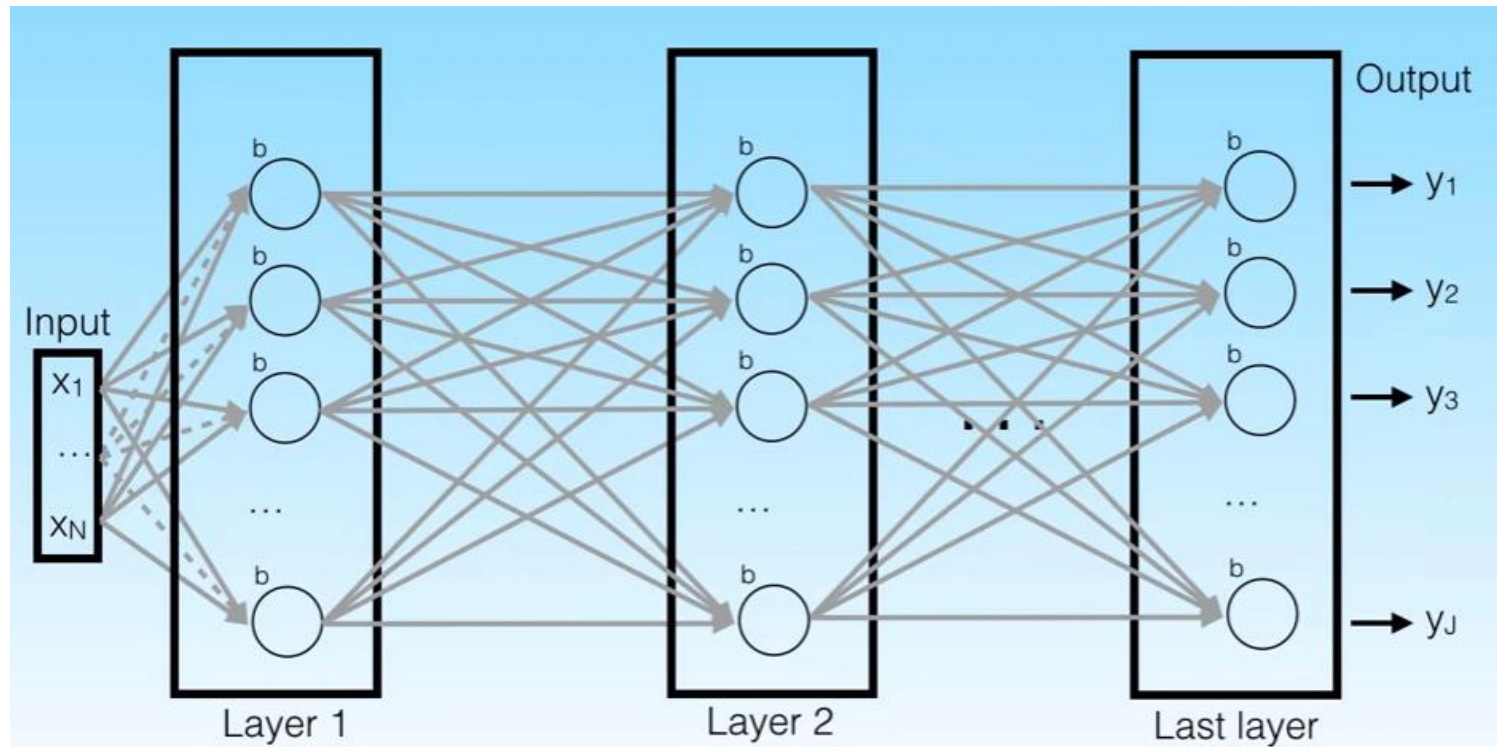
Preparação dos dados do conjunto de treinamento e de testes

PREPARAÇÃO DOS DADOS

O que fazer quando temos mais de duas classes de saída?

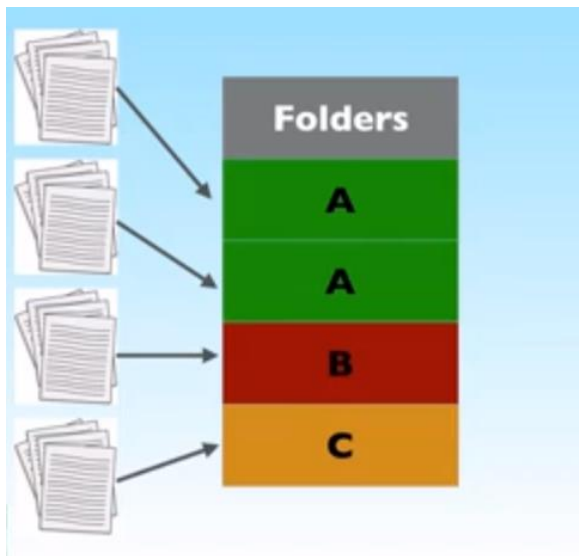


A camada final precisa ter mais neurônios



Os clusters/classes podem ser exclusivos ou não, veja os exemplos:

Exclusivo

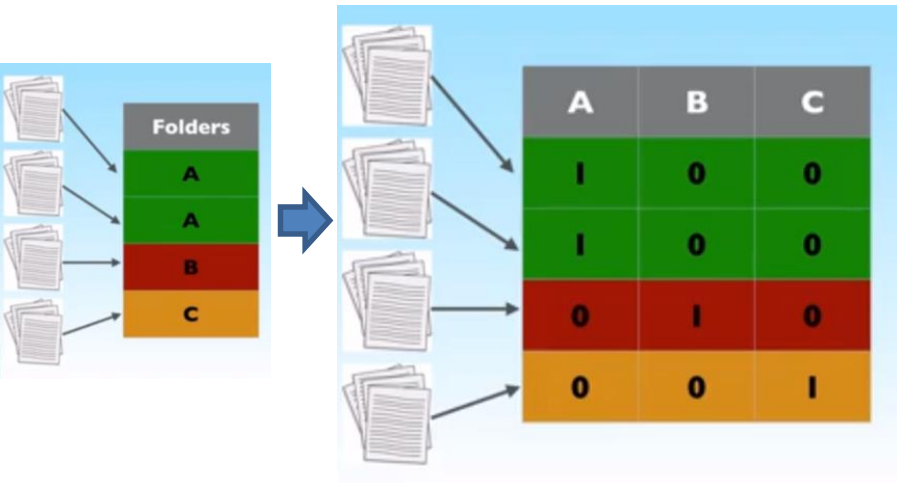


Não exclusivo

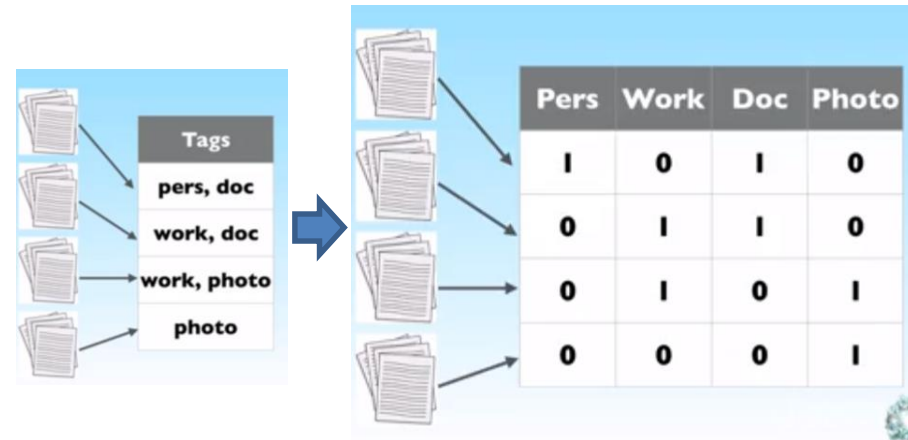


Dados precisam ser preparados

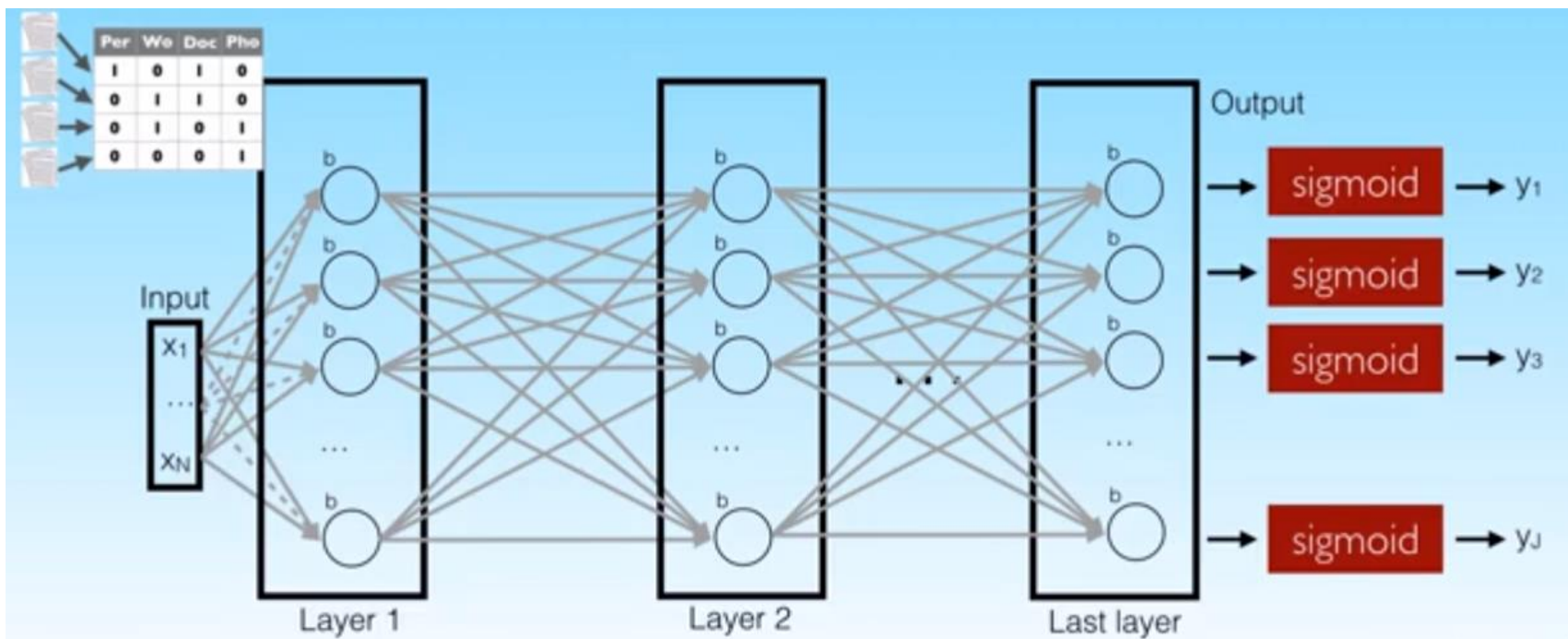
Exclusivo



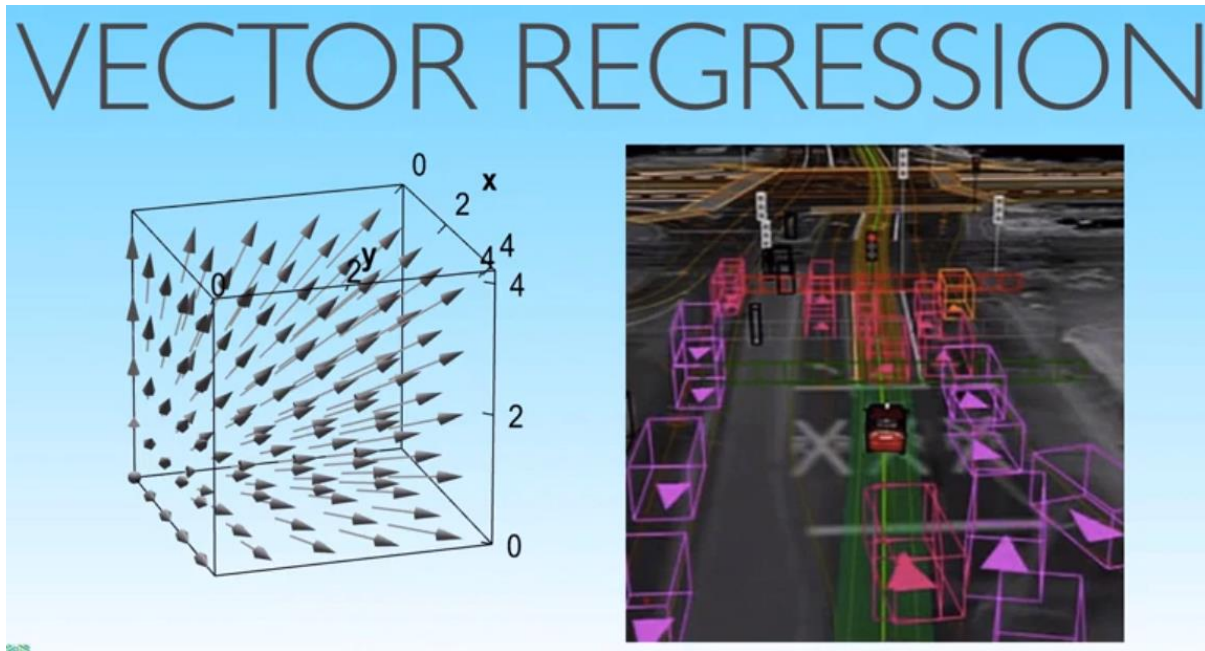
Não exclusivo



Topologia para saídas não exclusivas

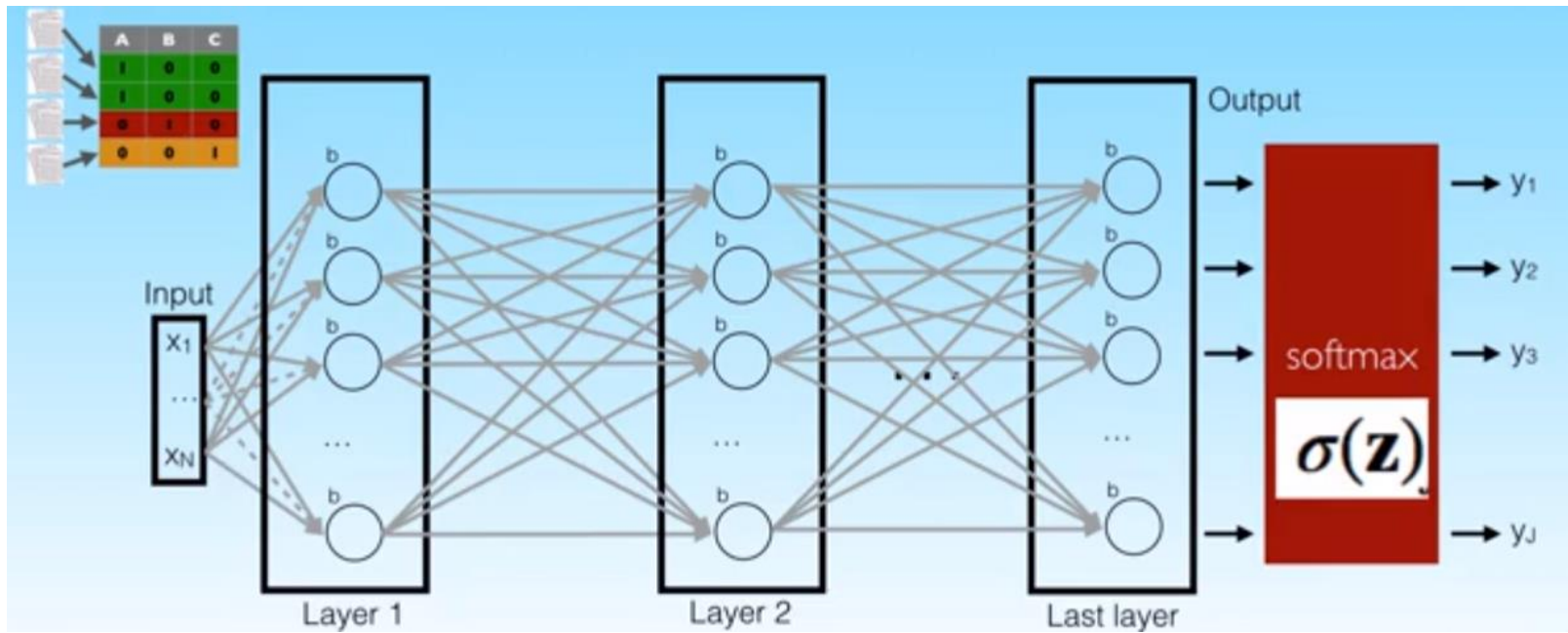


Carros autônomos precisam muitas saídas...



Mas e o caso de uma saída exclusiva?

- Função SOFTMAX
$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \text{ for } j = 1, \dots, K.$$



Resumo

- Regressão de vetores dependem de muitos nós de saída.
- Classificação multi classe utiliza a função SOFTMAX para saídas.
- Classes não exclusivas usam função sigmoid com múltiplos nós/neurônios na saída.
- Exercícios
 - Deep Learning Intro no Jupyter Notebook
 - Vá até “Multiclass classification”
 - Testes
 - Explicação do código