Introdução ao TensorFlow para Redes Neurais

Lucas Leite

Apresentação

Lucas Gonçalves de Moura Leite

Doutorando em Ciência da Computação Computação (UFC 2017-)

Mestre em Ciência da Computação (UFC 2014-2016)

2018 – Atual

UniFametro – Professor Assistente

Instituto Atlântico – Cientista de Dados

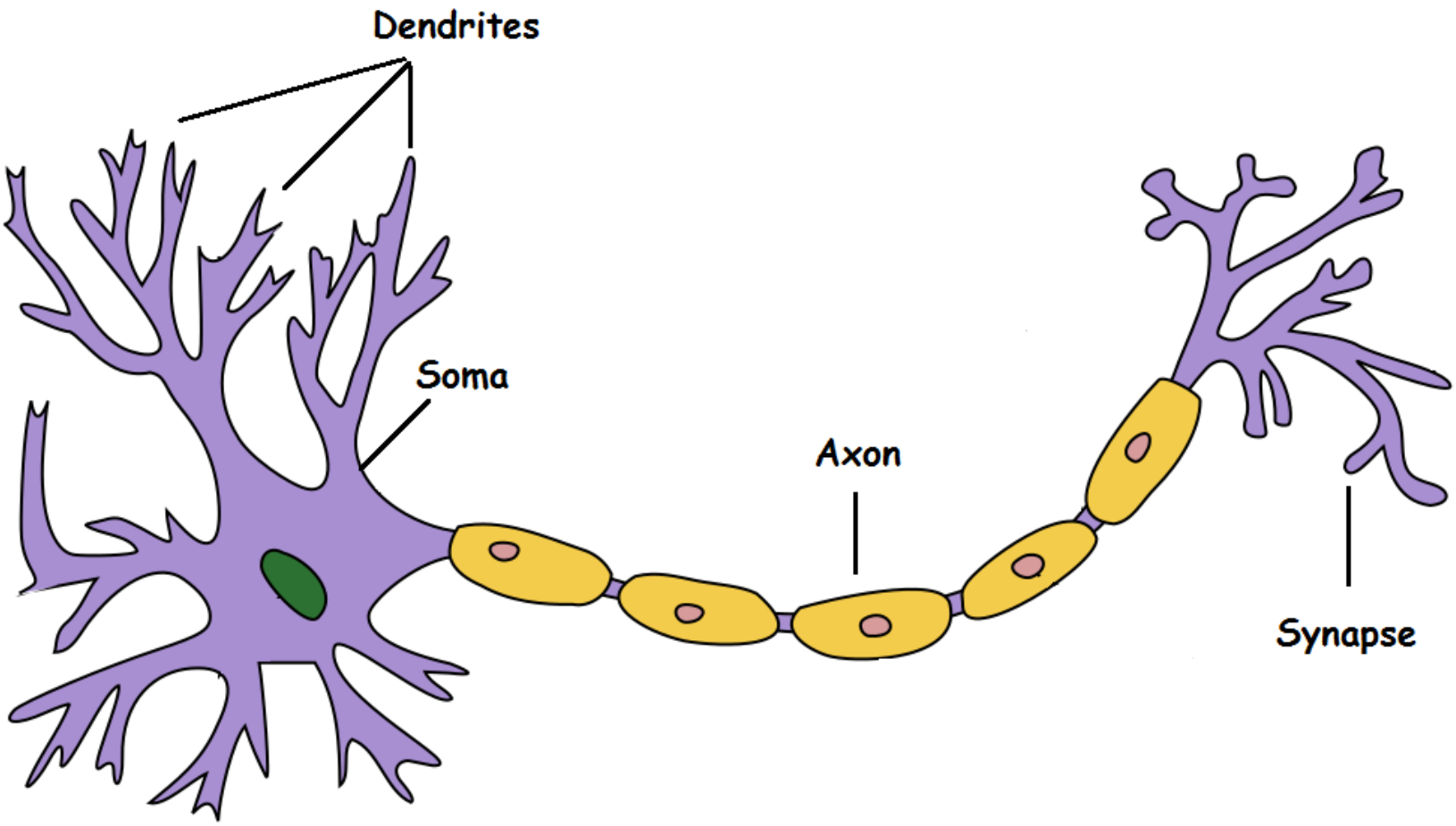
2016 – 2018

Laboratório de Sistemas e Banco de Dados (UFC) – Pesquisador em Aprendizado de Máquina

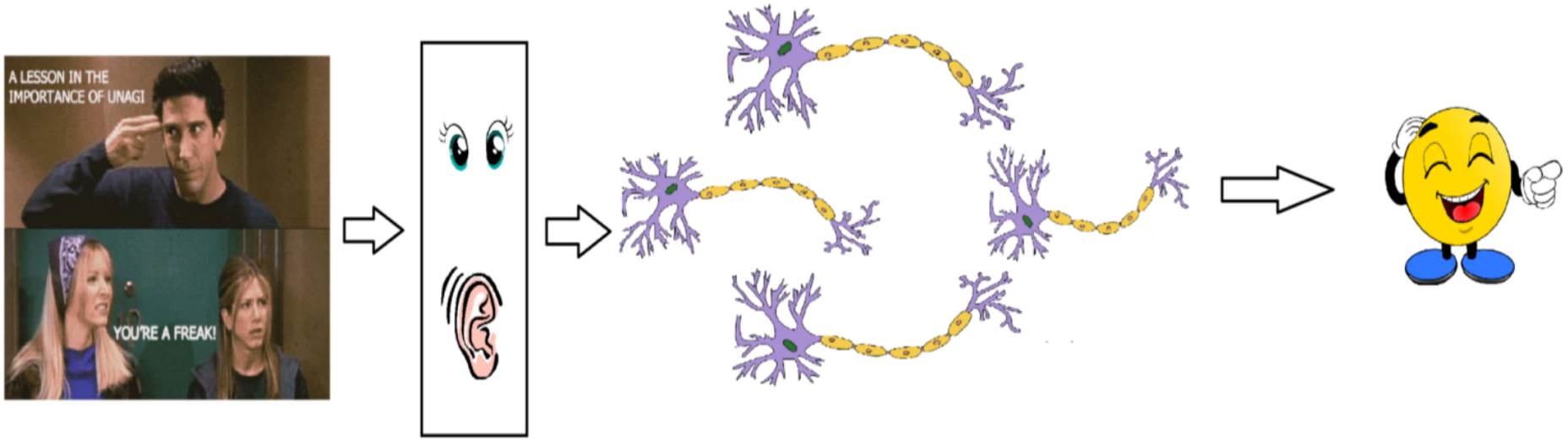
Introdução às Redes Neurais

- Funcionamento inspirado no neurônio biológico
- Neurônio Artificial de McCulloch-Pitts (1943)

Neurônio

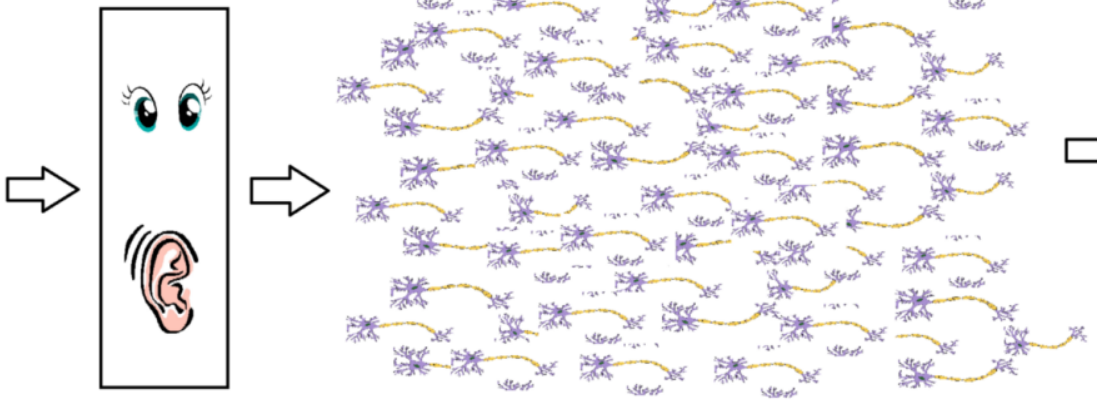


Funcionamento dos Neurônios

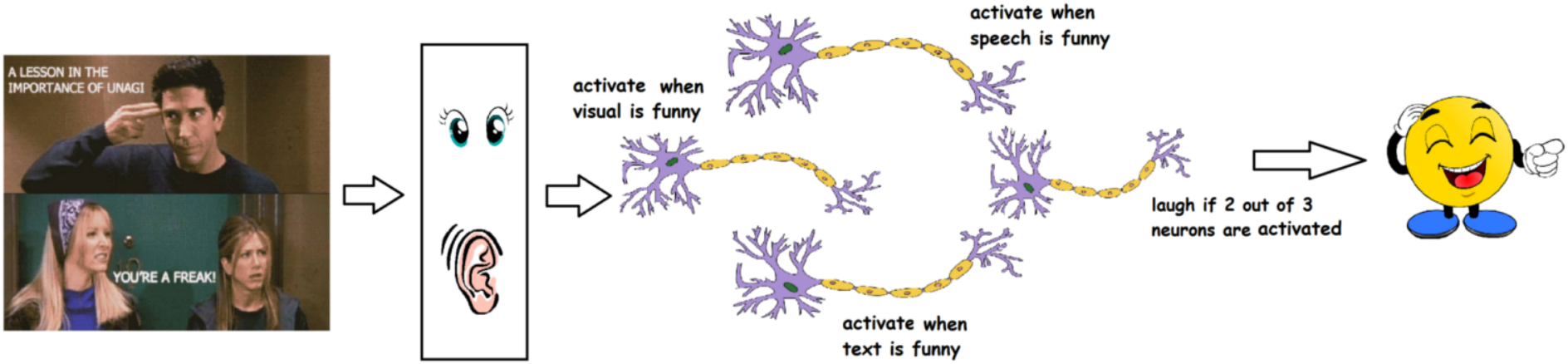


Funcionamento dos Neurônios

10^{11} neurônios (100 bilhões)

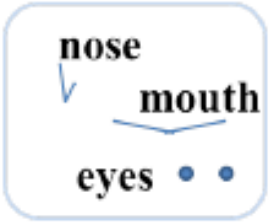


Funcionamento dos Neurônios



Funcionamento dos Neurônios

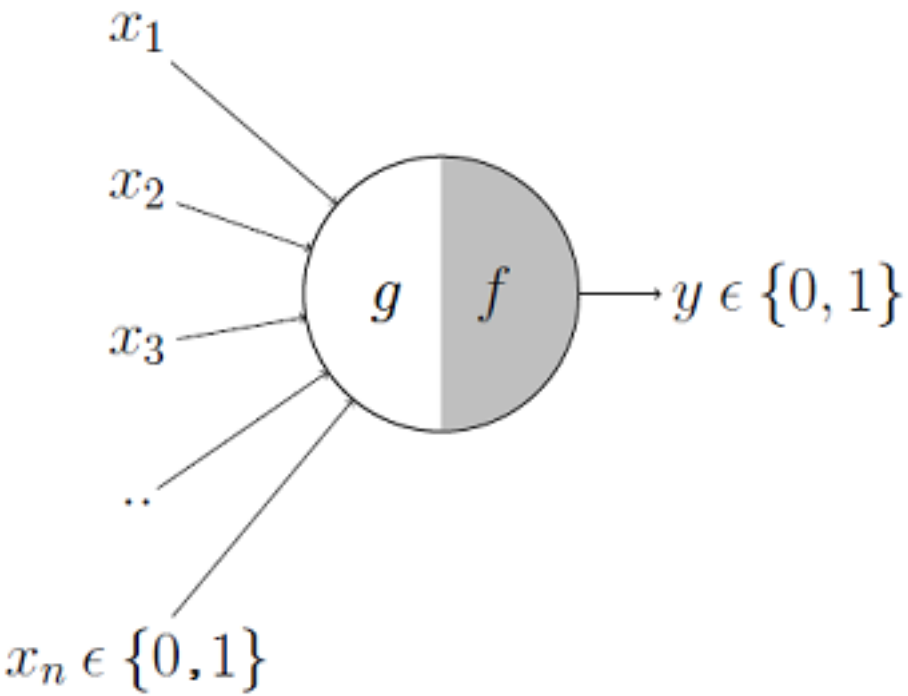
Layer 1: detect edges & corners



Layer 2: form feature groups

Layer 3: detect high level objects, faces, etc.

Modelo de McCulloch-Pitts



Perceptron

McCulloch Pitts Neuron
(assuming no inhibitory inputs)

$$y = 1 \quad if \sum_{i=0}^{n} x_i \geq 0$$

$$= 0 \quad if \sum_{i=0}^{n} x_i < 0$$

Perceptron

$$y = 1 \quad if \sum_{i=0}^{n} w_i * x_i \geq 0$$

$$= 0 \quad if \sum_{i=0}^{n} w_i * x_i < 0$$

Predição de um Perceptron

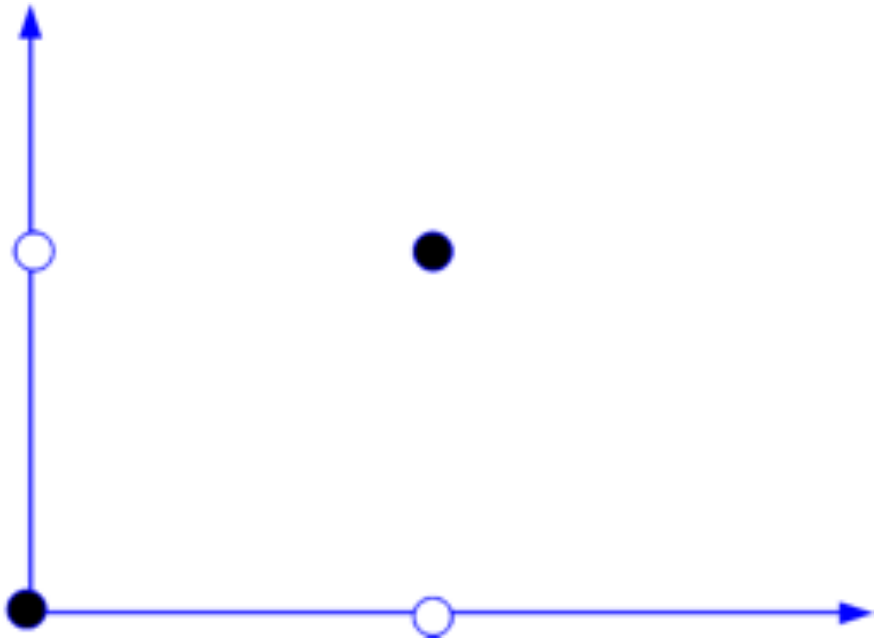
- x_1 - É um jogo do Brasileirão?
- x_2 - É um amistoso?
- x_3 - Estarei em casa?
- x_4 - É jogo do Ceará?

$$g(x_1, x_2, x_3, \ldots, x_n) = g(\mathbf{x}) = \sum_{i=1}^{n} x_i$$

$$\begin{aligned} y = f(g(\mathbf{x})) &= 1 \quad if \quad g(\mathbf{x}) \geq \theta \\ &= 0 \quad if \quad g(\mathbf{x}) < \theta \end{aligned}$$

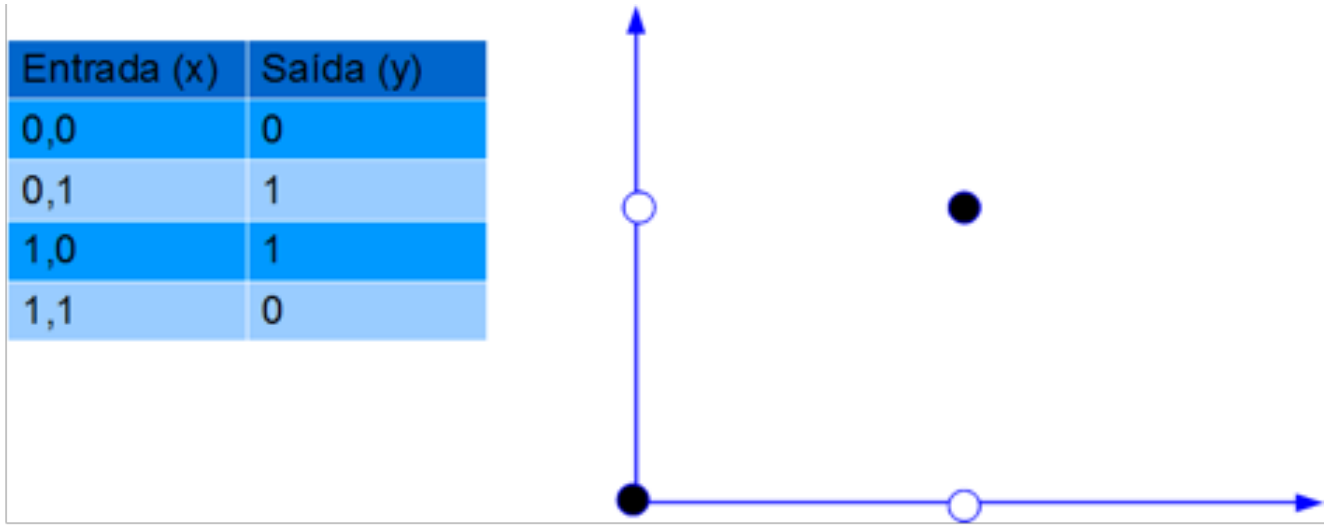
Impacto dos neurônios

Entrada (x)	Saída (y)
0,0	0
0,1	1
1,0	1
1,1	0



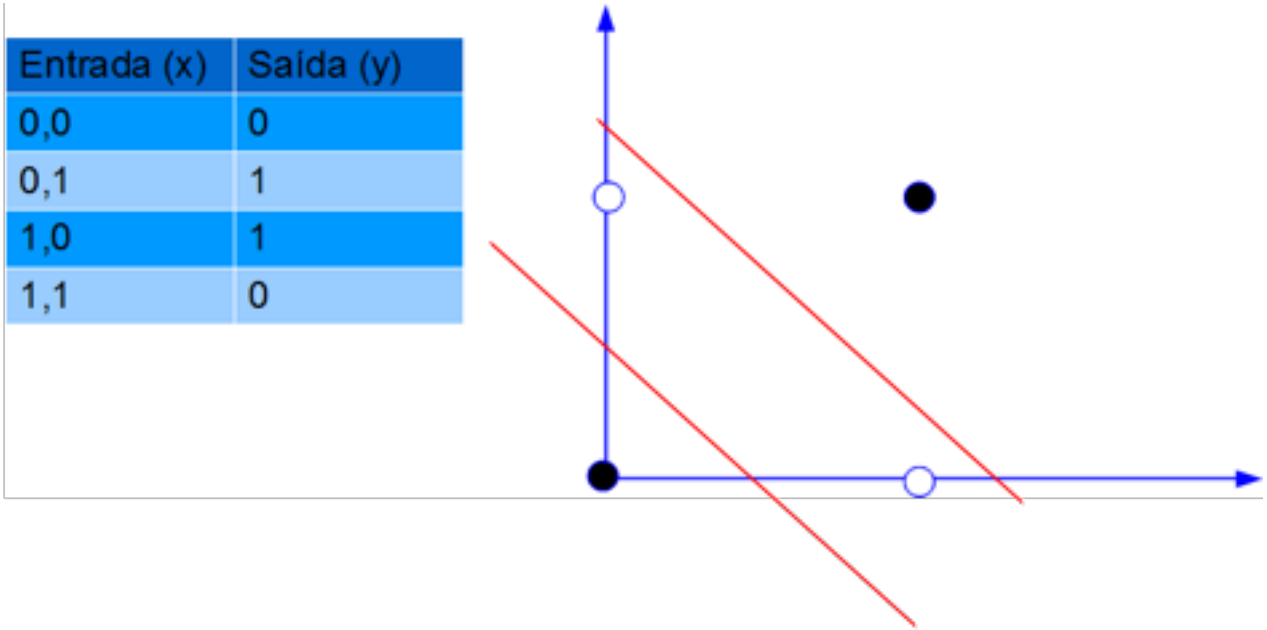
Impacto dos neurônios

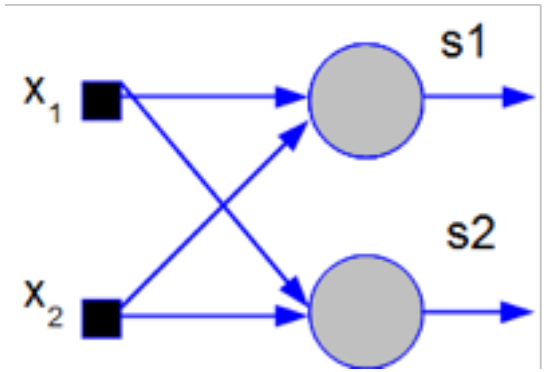
Entrada (x)	Saída (y)
0,0	0
0,1	1
1,0	1
1,1	0



Impacto dos neurônios

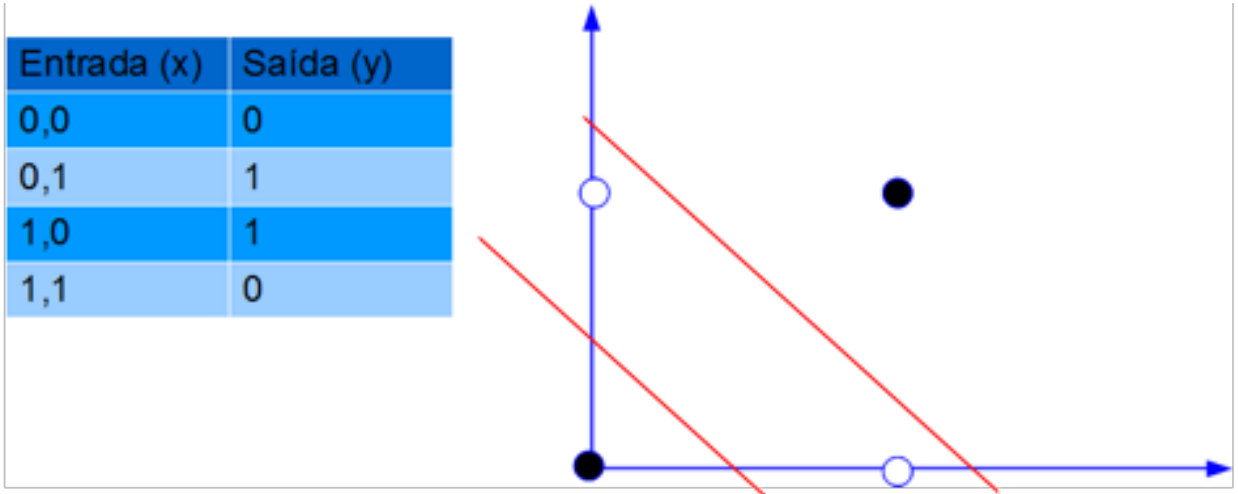
Entrada (x)	Saída (y)
0,0	0
0,1	1
1,0	1
1,1	0

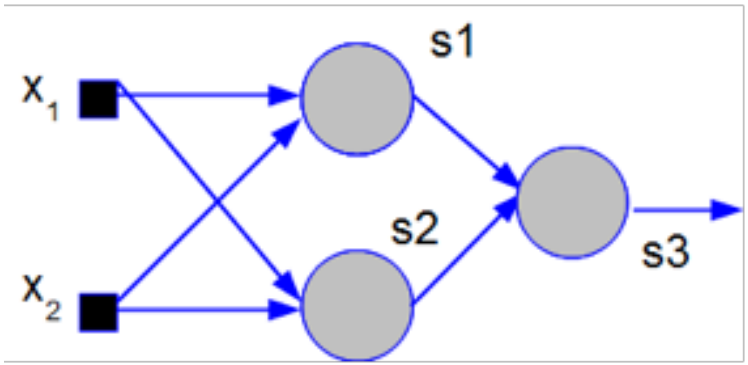




Impacto dos neurônios

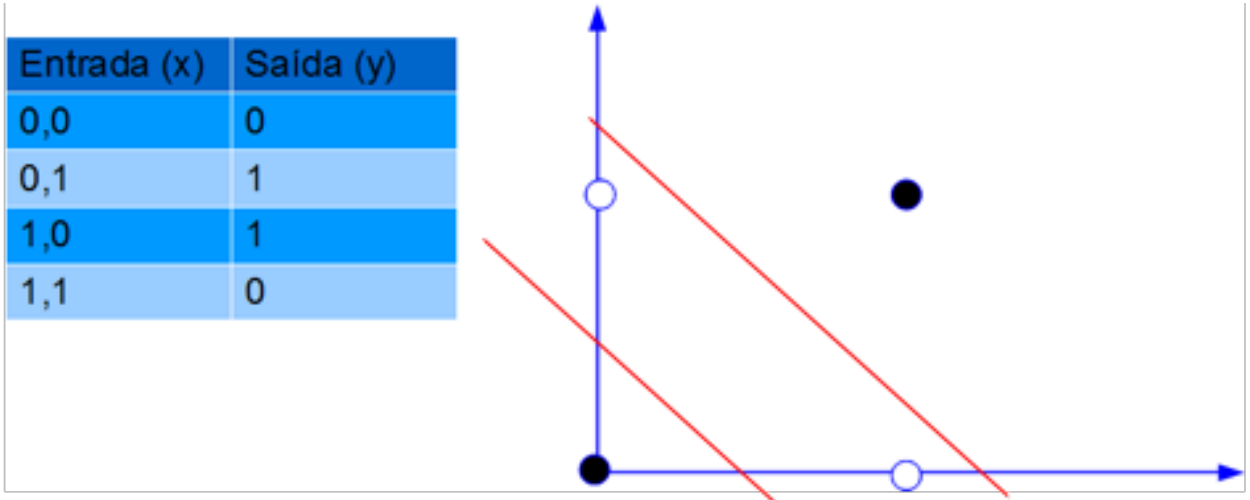
Entrada (x)	Saída (y)
0,0	0
0,1	1
1,0	1
1,1	0

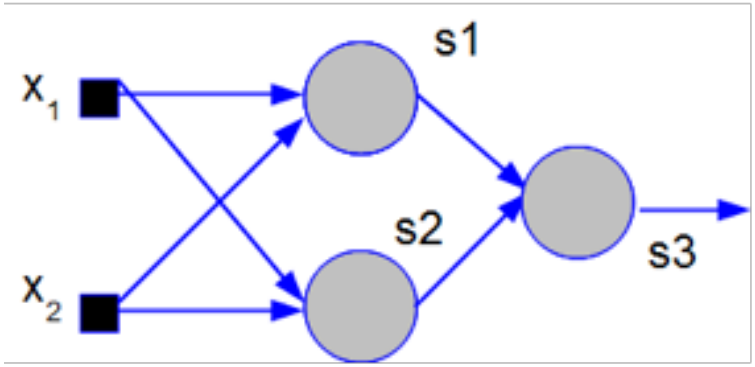


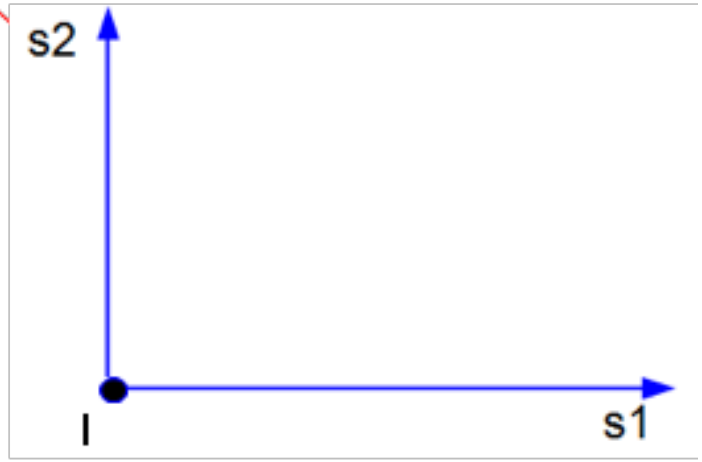


Impacto dos neurônios

Entrada (x)	Saída (y)
0,0	0
0,1	1
1,0	1
1,1	0

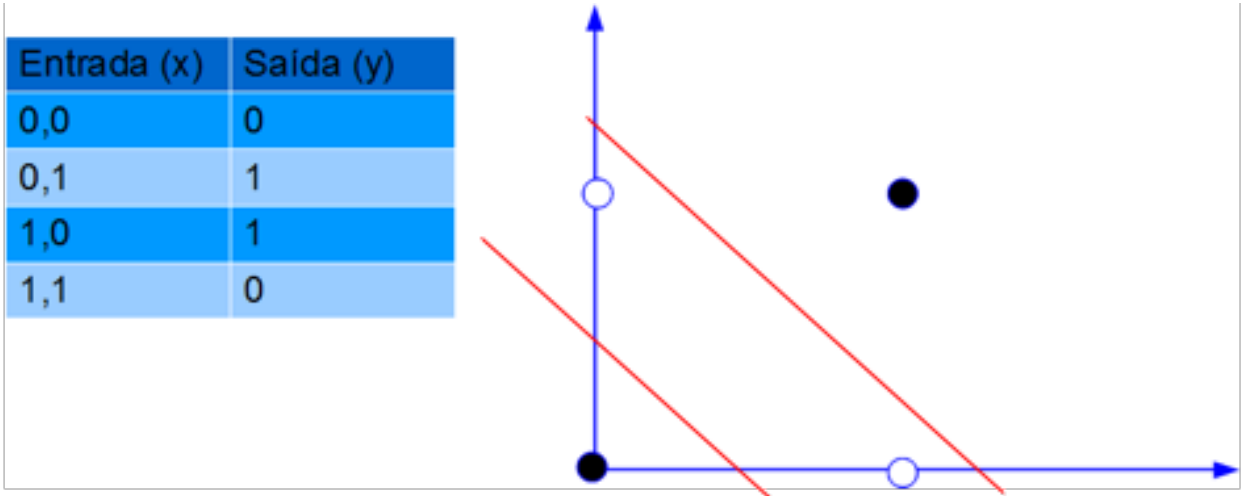


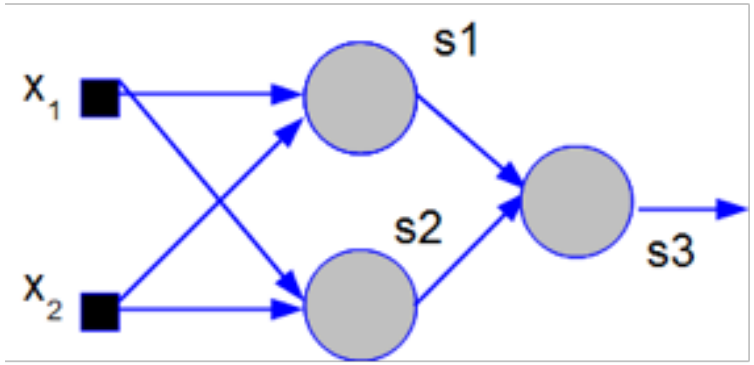


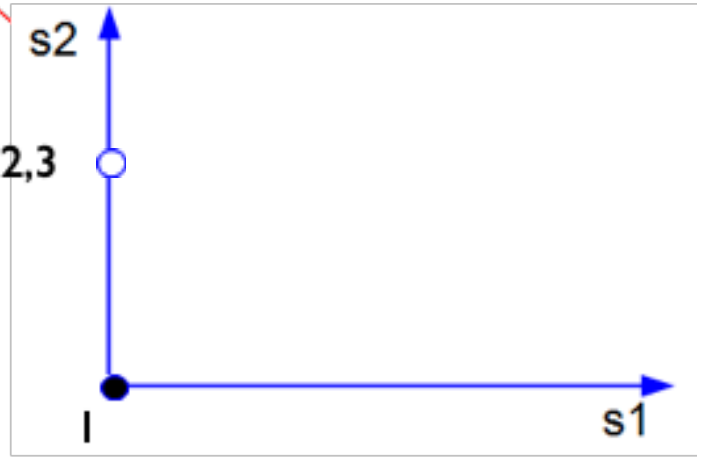


Impacto dos neurônios

Entrada (x)	Saída (y)
0,0	0
0,1	1
1,0	1
1,1	0

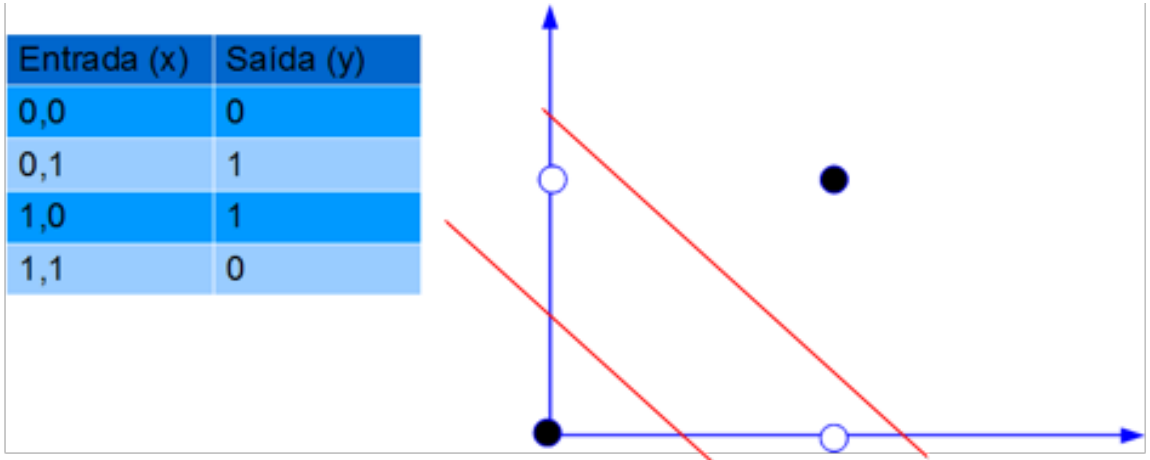


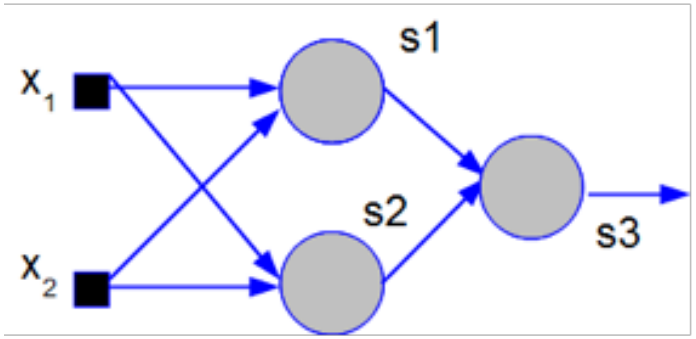


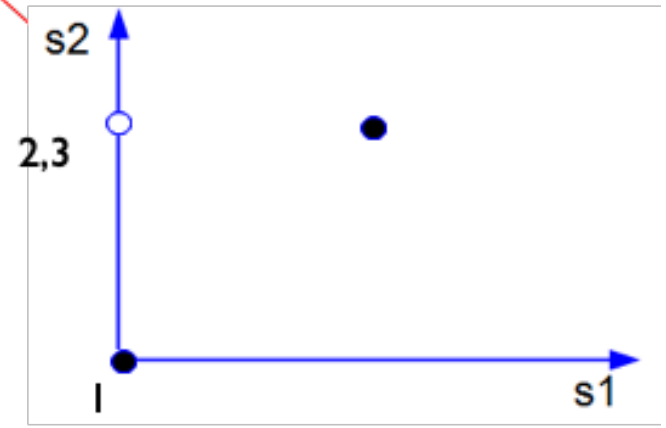


Impacto dos neurônios

Entrada (x)	Saída (y)
0,0	0
0,1	1
1,0	1
1,1	0

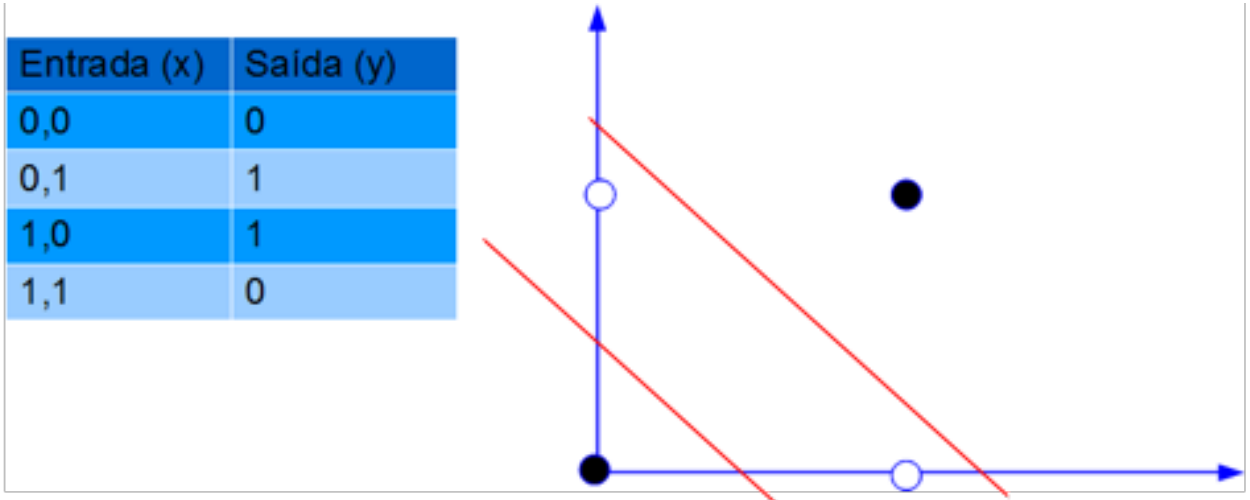


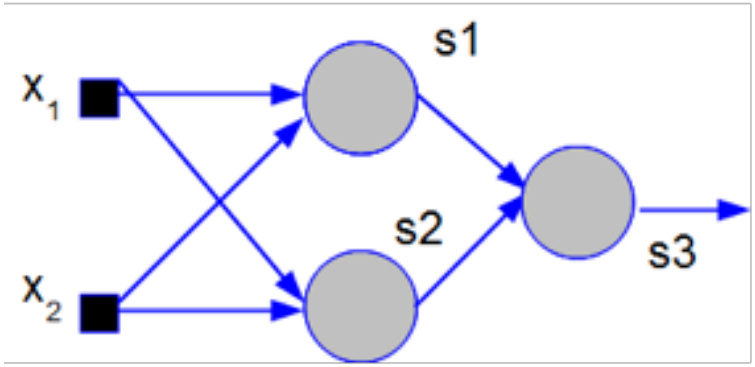


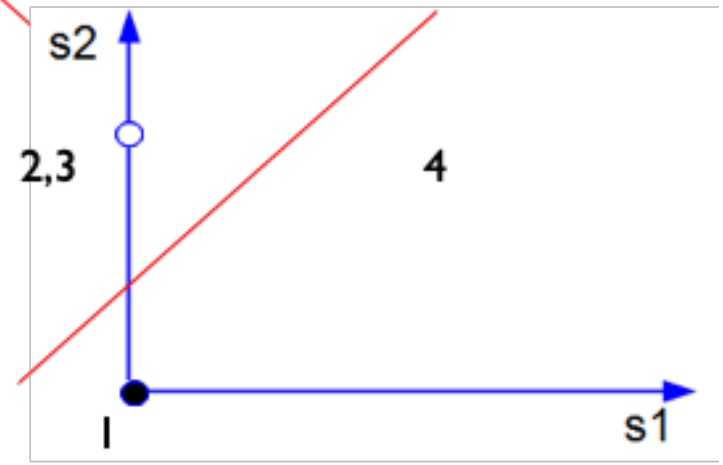


Impacto dos neurônios

Entrada (x)	Saída (y)
0,0	0
0,1	1
1,0	1
1,1	0



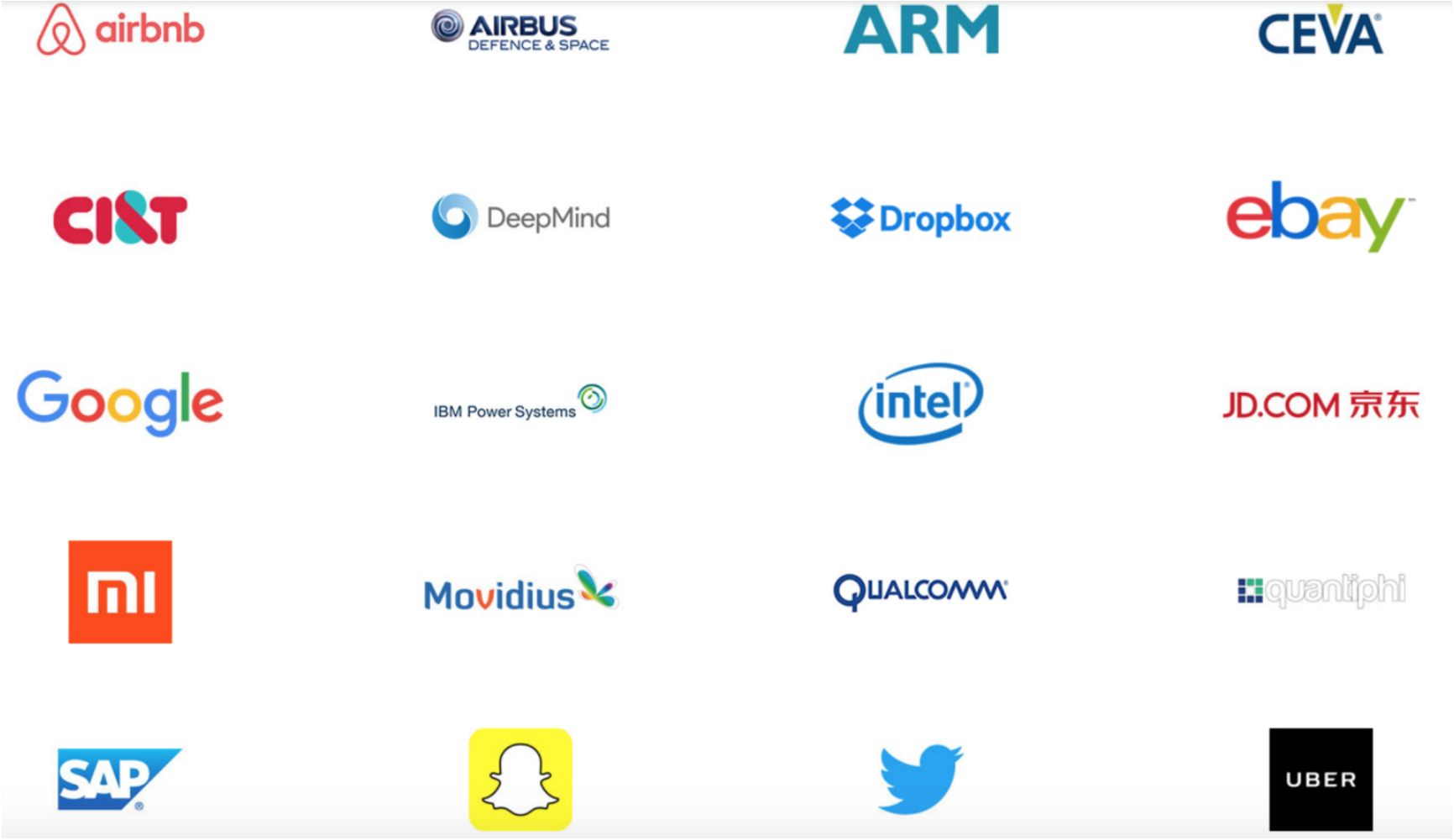




TensorFlow

- Google Brain
- Biblioteca para computação numérica
- CPUs, GPUs e dispositivos móveis
- Desenvolvido para Aprendizado de Máquina e Deep Learning (e não somente isso)

TensorFlow


airbnb
AIRBUS
DEFENCE & SPACE
ARM
CEVA
CI&T
DeepMind
Dropbox
ebay
Google
IBM Power Systems
intel
JD.COM 京东
Movidius
QUALCOMM
quantiphi
SAP
UBER

Redes Convolucionais

- Imagens são consideradas matrizes
- Alta dimensionalidade
- Grande número de pesos

Convolução

INPUT IMAGE

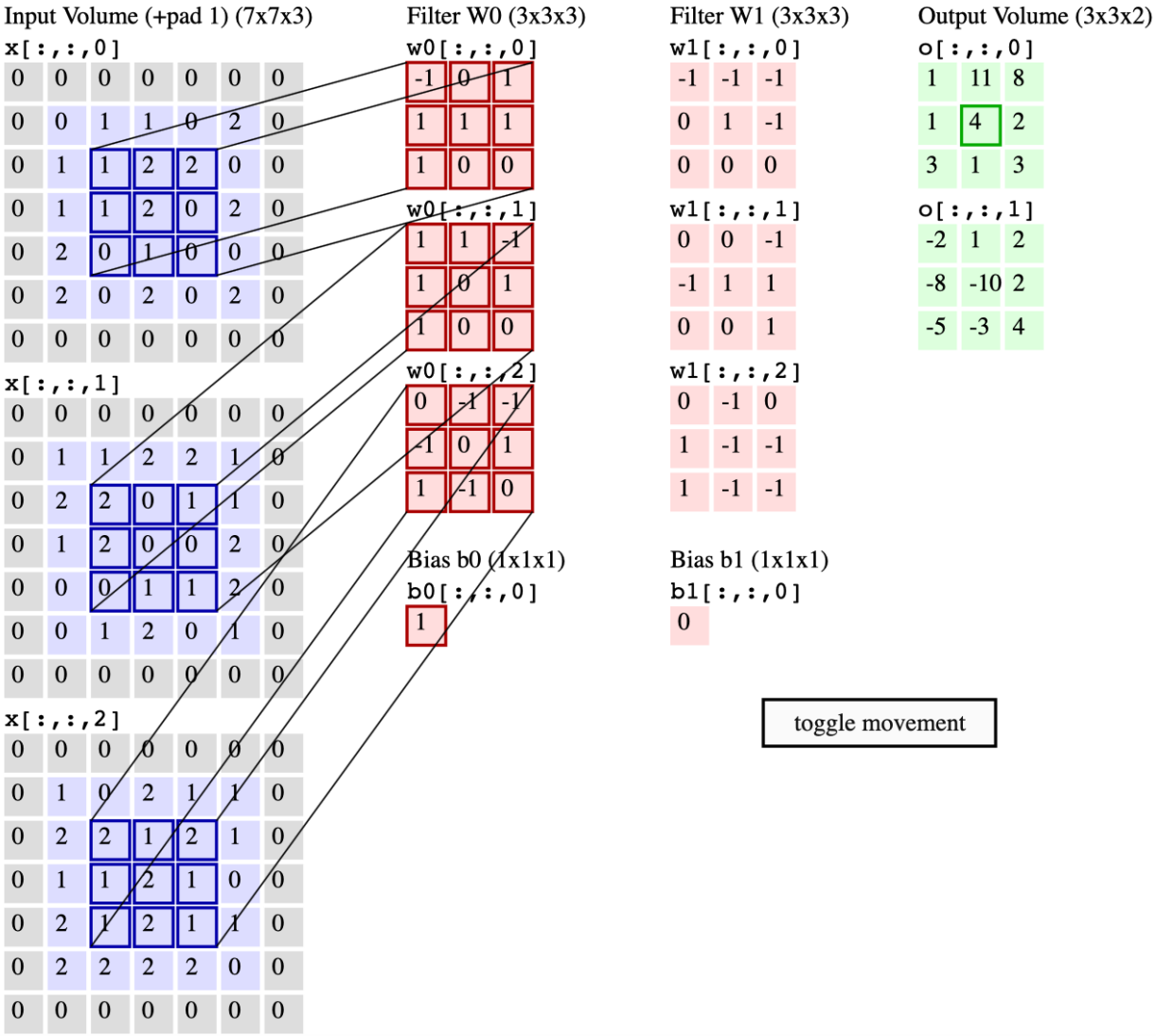
18	54	51	239	244	188
55	121	75	78	95	88
35	24	204	113	109	221
3	154	104	235	25	130
15	253	225	159	78	233
68	85	180	214	245	0

WEIGHT

1	0	1
0	1	0
1	0	1

429

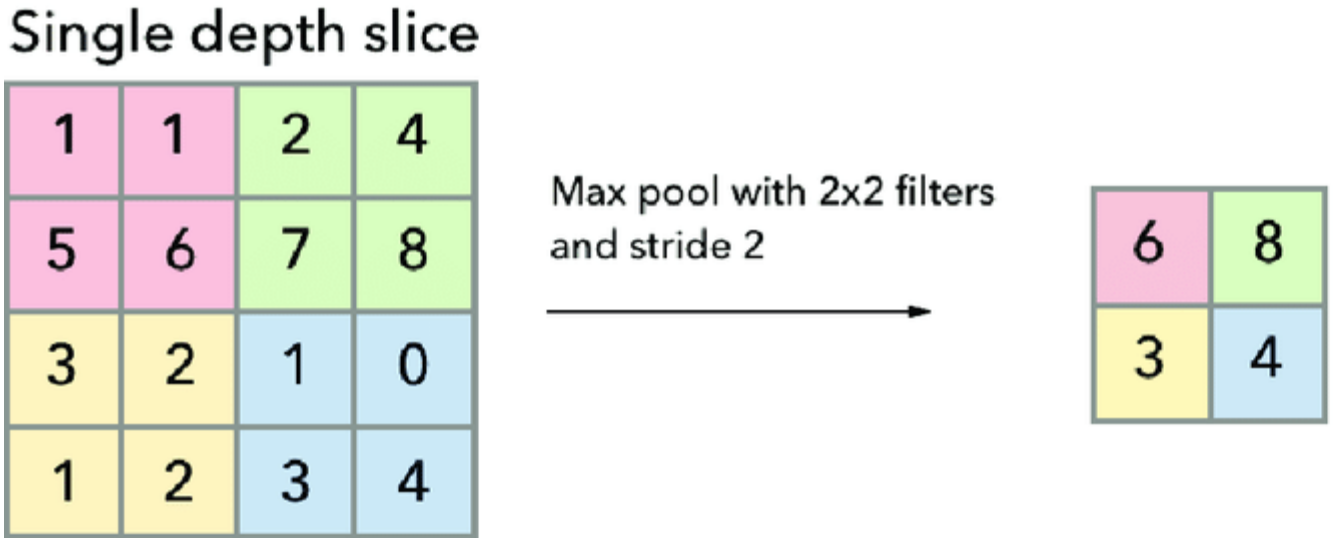
Convolução


Input Volume (+pad 1) (7x7x3)
x[:,:,0]
0 0 0 0 0 0 0
0 0 1 1 0 2 0
0 1 1 2 2 0 0
0 1 1 2 0 2 0
0 2 0 1 0 0 0
0 2 0 2 0 2 0
0 0 0 0 0 0 0
x[:,:,1]
0 0 0 0 0 0 0
0 1 1 2 2 1 0
0 2 2 0 1 1 0
0 1 2 0 0 2 0
0 0 0 1 1 2 0
0 0 1 2 0 1 0
0 0 0 0 0 0 0
x[:,:,2]
0 0 0 0 0 0 0
0 1 0 2 1 1 0
0 2 2 1 2 1 0
0 1 1 2 1 0 0
0 2 1 2 1 1 0
0 2 2 2 2 0 0
0 0 0 0 0 0 0
Filter W0 (3x3x3)
w0[:,:,0]
-1 0 1
1 1 1
1 0 0
w0[:,:,1]
1 1 -1
1 0 1
1 0 0
w0[:,:,2]
0 -1 -1
-1 0 1
1 -1 0
Bias b0 (1x1x1)
b0[:,:,0]
1
Filter W1 (3x3x3)
w1[:,:,0]
-1 -1 -1
0 1 -1
0 0 0
w1[:,:,1]
0 0 -1
-1 1 1
0 0 1
w1[:,:,2]
0 -1 0
1 -1 -1
1 -1 -1
Bias b1 (1x1x1)
b1[:,:,0]
0
Output Volume (3x3x2)
o[:,:,0]
1 11 8
1 4 2
3 1 3
o[:,:,1]
-2 1 2
-8 -10 2
-5 -3 4
toggle movement

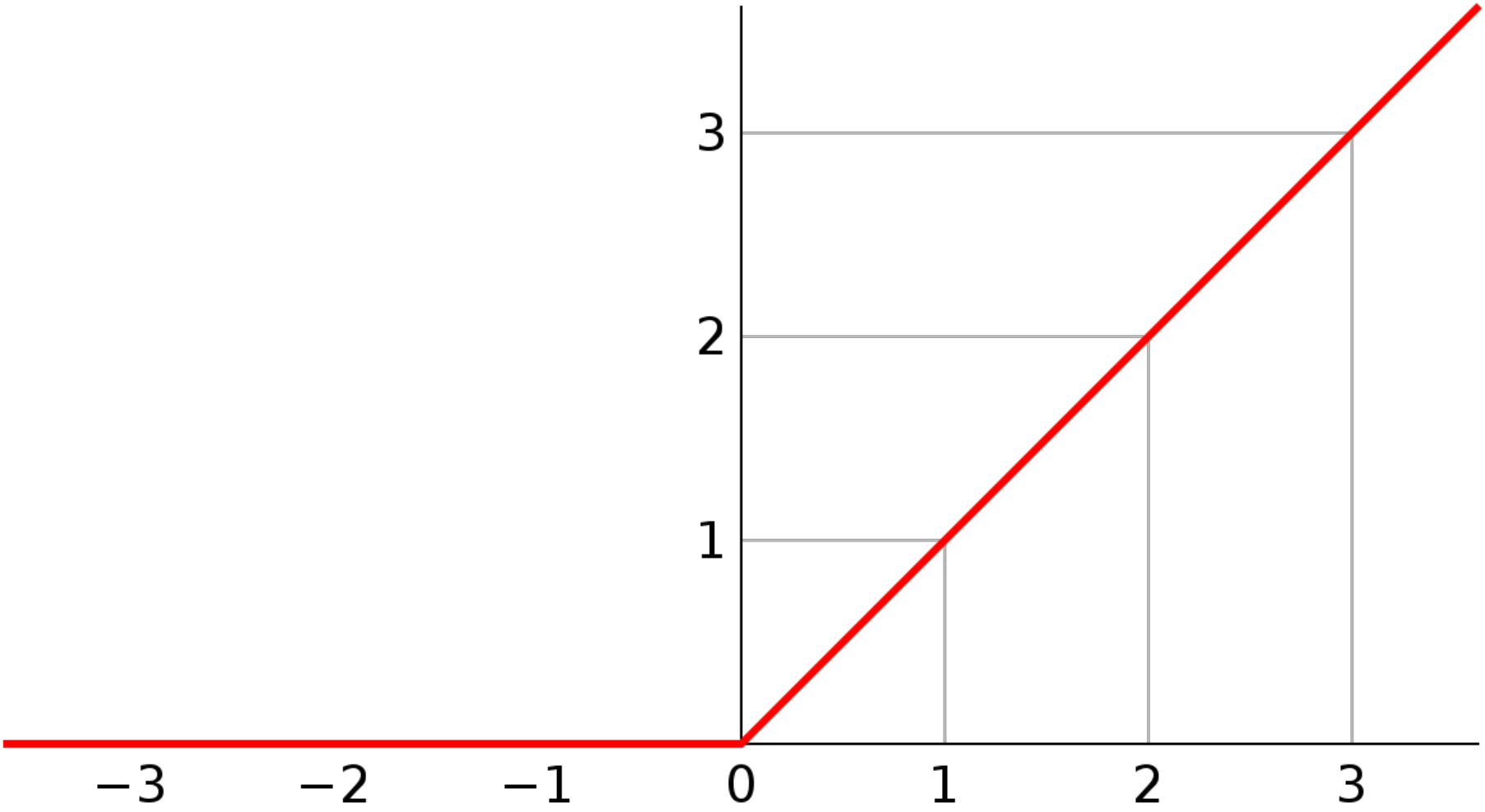
Filtros

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

MaxPooling


Single depth slice
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
Max pool with 2x2 filters
and stride 2
6 8
3 4

Rectified Linear Unit - ReLU



Rectified Linear Unit - ReLU

Rectified Linear Unit - ReLU

Rede Neural Convolucional - CNN