



RODANDO MODELOS DO TENSORFLOW NO ANDROID



HELLO!

Ricardo Vitor Costa Neto.

Github: <https://github.com/Ricardovcn>

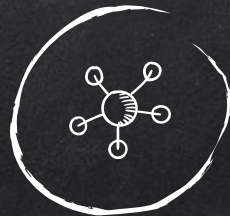
PASSO A PASSO



- ✗ Entendendo mais sobre Redes Neurais.
- ✗ Treinando seu próprio modelo de Rede Neural..
- ✗ Exportando seu modelo modelo.
- ✗ Usando o modelo pré-treinando no android.



INTRODUÇÃO



TensorFlow

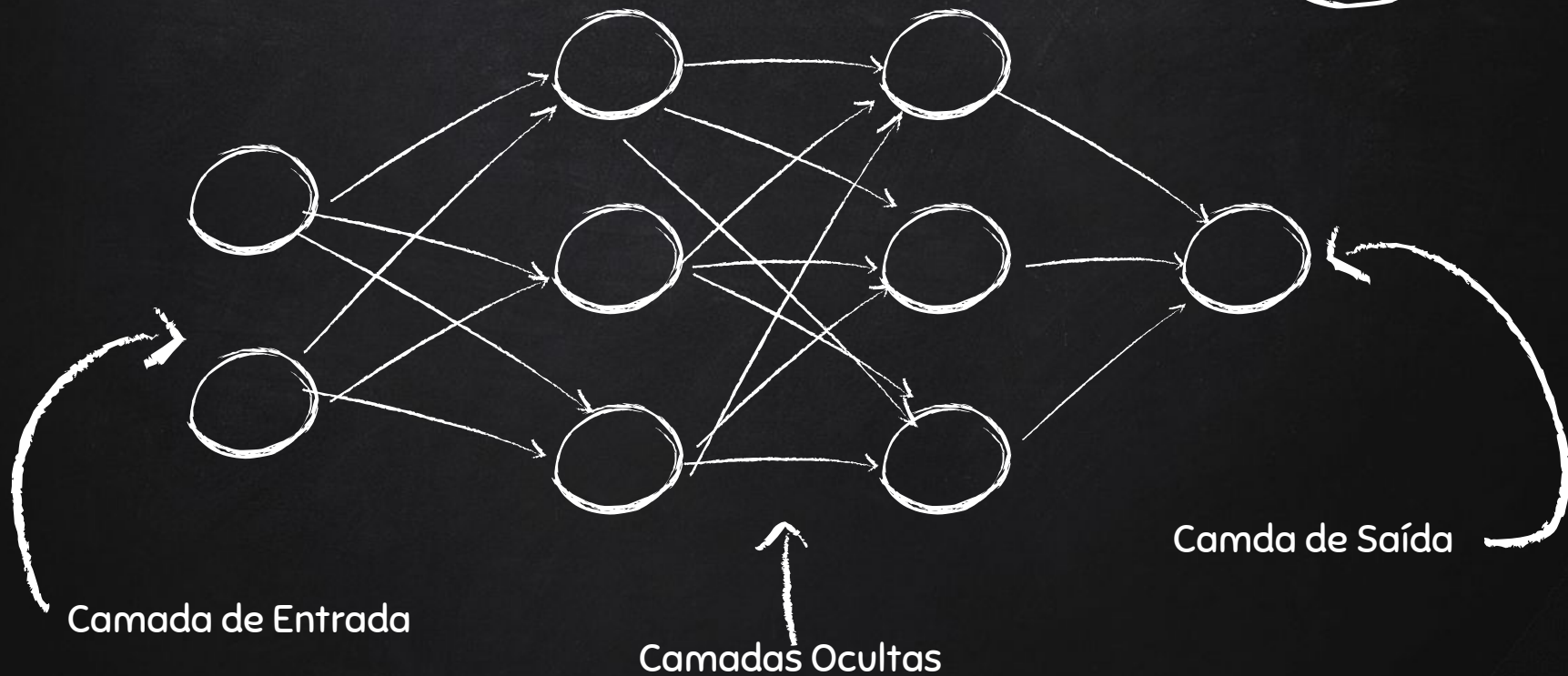
TensorFlow é uma biblioteca de código aberto para aprendizado de máquina aplicável a uma ampla variedade de tarefas.

Redes Neurais

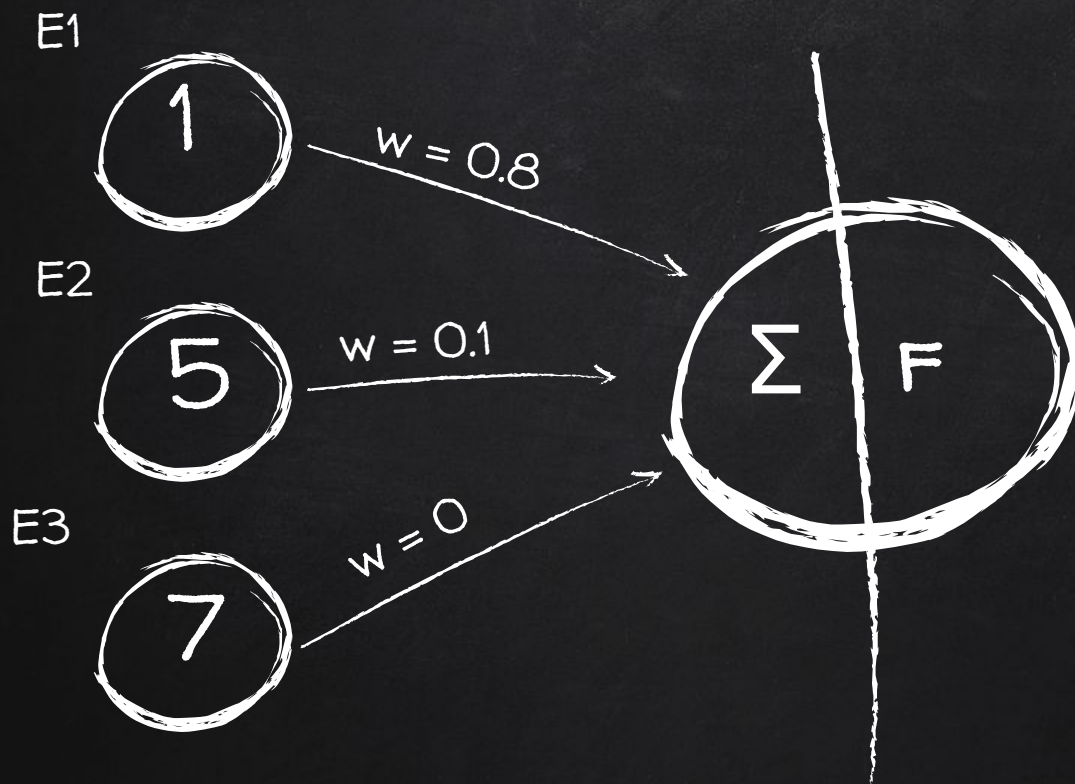
Redes neurais são sistemas de computação com nós interconectados que funcionam como os neurônios do cérebro humano.

Mais informações sobre o TensorFlow: <https://www.tensorflow.org>.

EXEMPLO DE REDE NEURAL

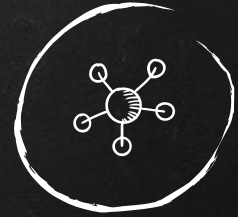


NEURÔNIO ARTIFICIAL

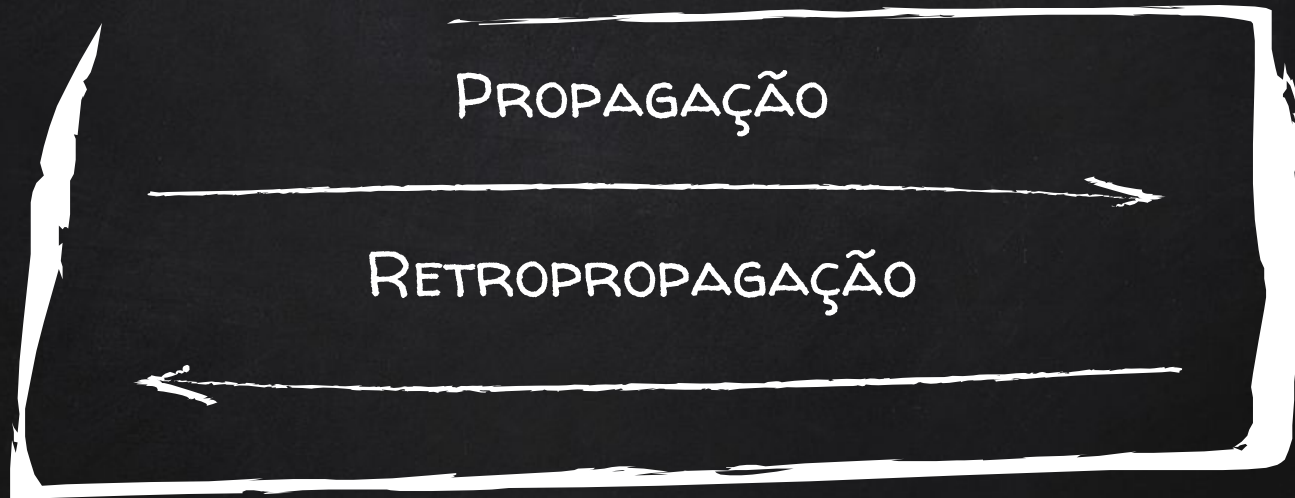


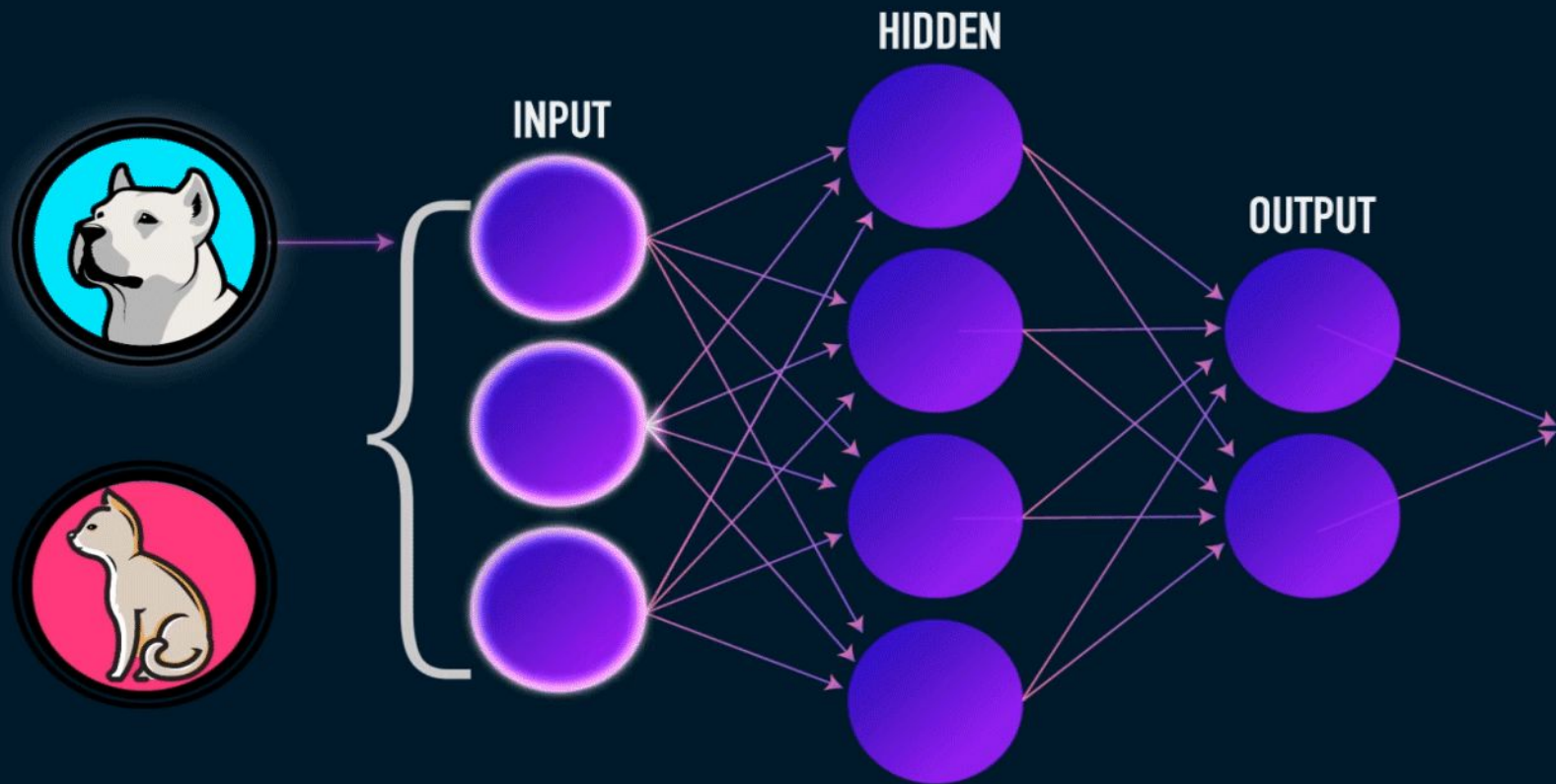
E = ENTRADA;
w = PESOS DAS ARESTAS;
 Σ = SOMATÓRIO / BIAS;
F = FUNÇÃO DE ATIVAÇÃO.

BACK PROPAGATION

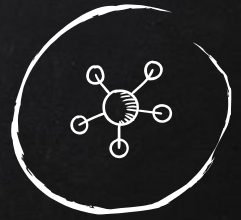


DUAS FASES:



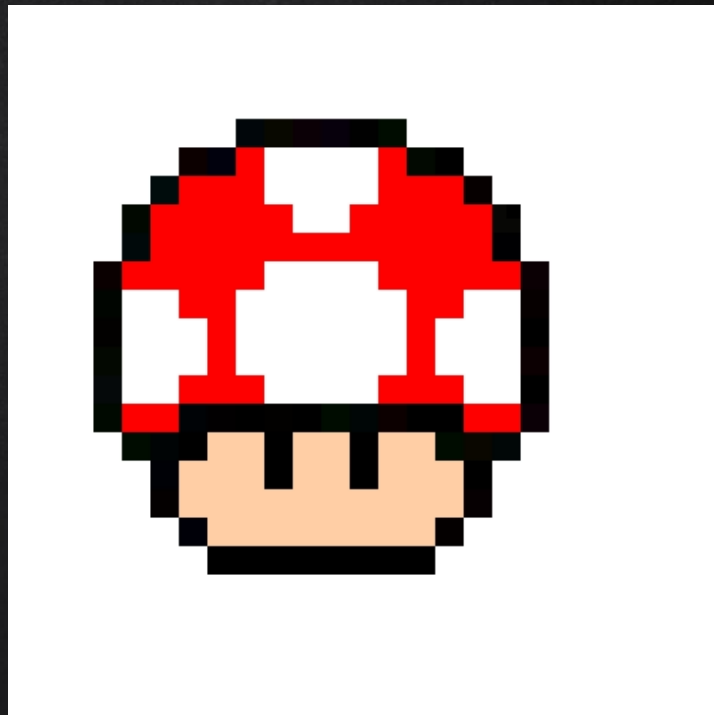
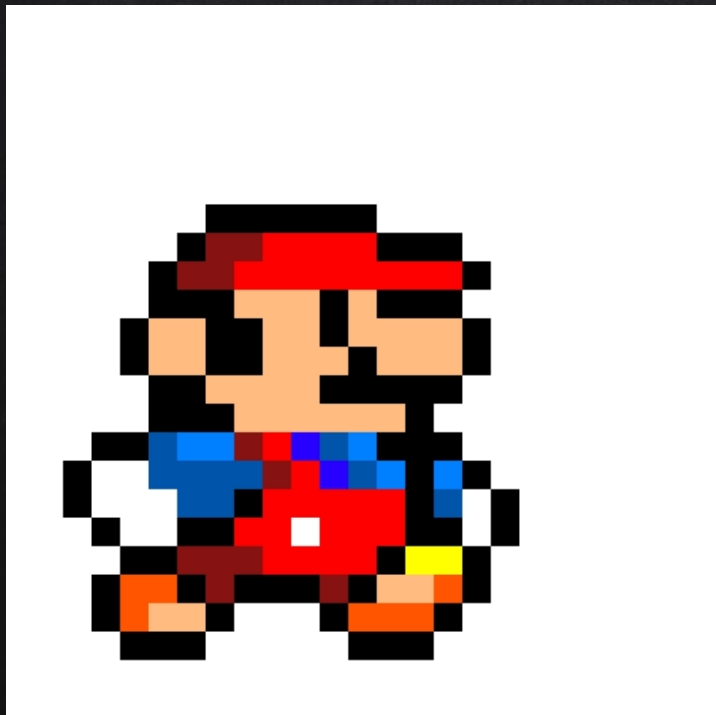


REDES NEURAIS CONVOLUCIONAIS



São usadas especificamente para a área de visão computacional. Essa área se preocupa em colocar dentro do computador a capacidade de visão.

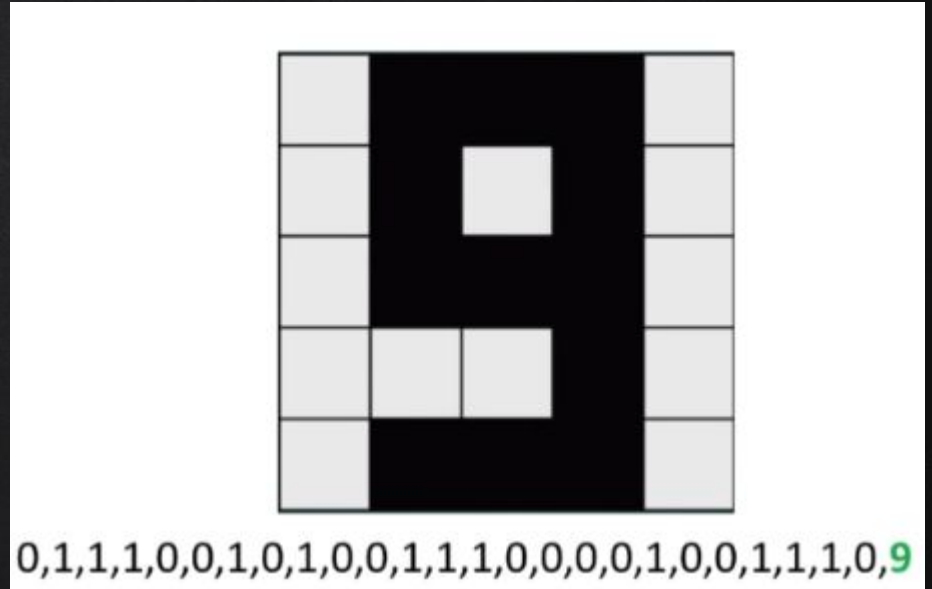
REDES NEURAIS CONVOLUCIONAIS



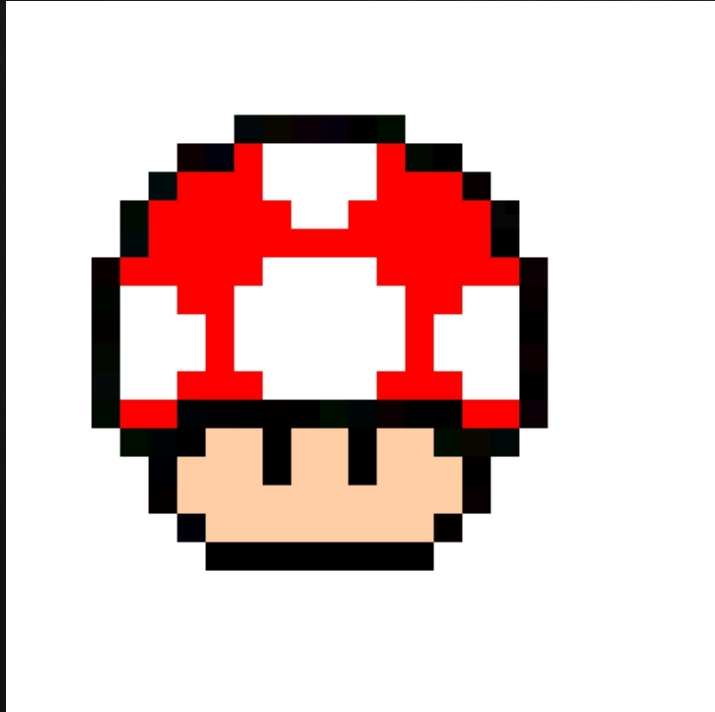
REDES NEURAIS CONVOLUCIONAIS



1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0

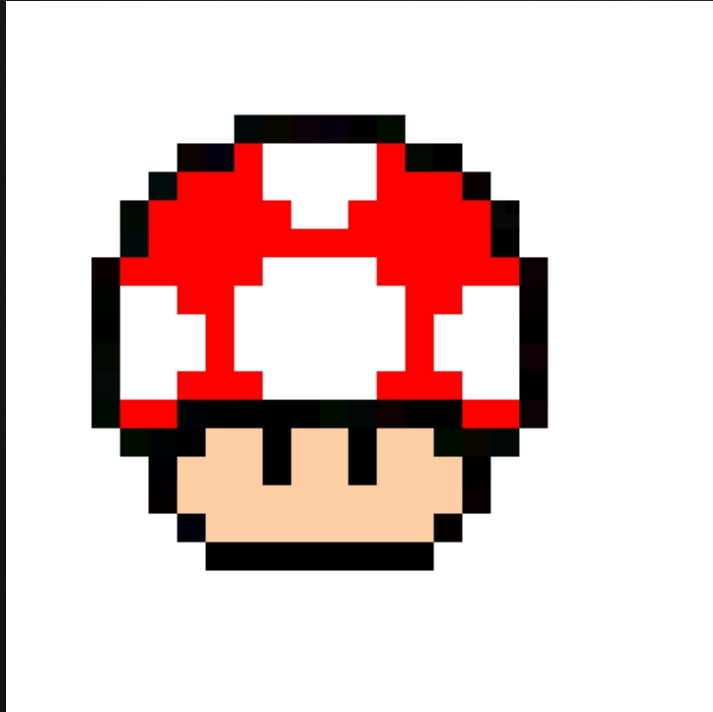


REDES NEURAIS CONVOLUCIONAIS



A imagem tem 16 pixels de altura e 16 pixel de largura.

REDES NEURAIS CONVOLUCIONAIS



$$16 \times 16 = 256 \times 3 = 768$$

Para representar as cores,
usando RGB são necessárias
3 entradas para cada pixel.

REDES NEURAIS CONVOLUCIONAIS



Para que a rede não fique lenta, a solução foi não usar todas as entradas ou todos os pixels. Uma rede neural convolucional seleciona e utiliza apenas as partes relevantes da imagem (as melhores características).

2.

IMPLEMENTAÇÃO

IMPLEMENTAÇÃO



- ✕ Operador de convolução.
- ✕ Pooling.
- ✕ Flattering.
- ✕ Rede Neural Densa.



IMPLEMENTAÇÃO

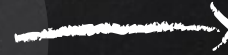
Operador
de
Convolução



Pooling



Flattening



Rede
Neural
Densa



OPERADOR DE CONVOLUÇÃO

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

Imagem

X

1	0	0
1	0	1
0	1	1

Detector de características
(feature detector)

=

0	1	0	1	0
0	2	1	1	2
1	2	2	3	1
1	3	3	3	2
1	3	1	3	5

Mapa de características
(feature map)

Mais sobre operador de convolução: [https://en.wikipedia.org/wiki/Kernel_\(image_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing))

Exemplo prático: <http://setosa.io/ev/image-kernels/>

POOLING



Reduz overfitting e Ruídos desnecessários.

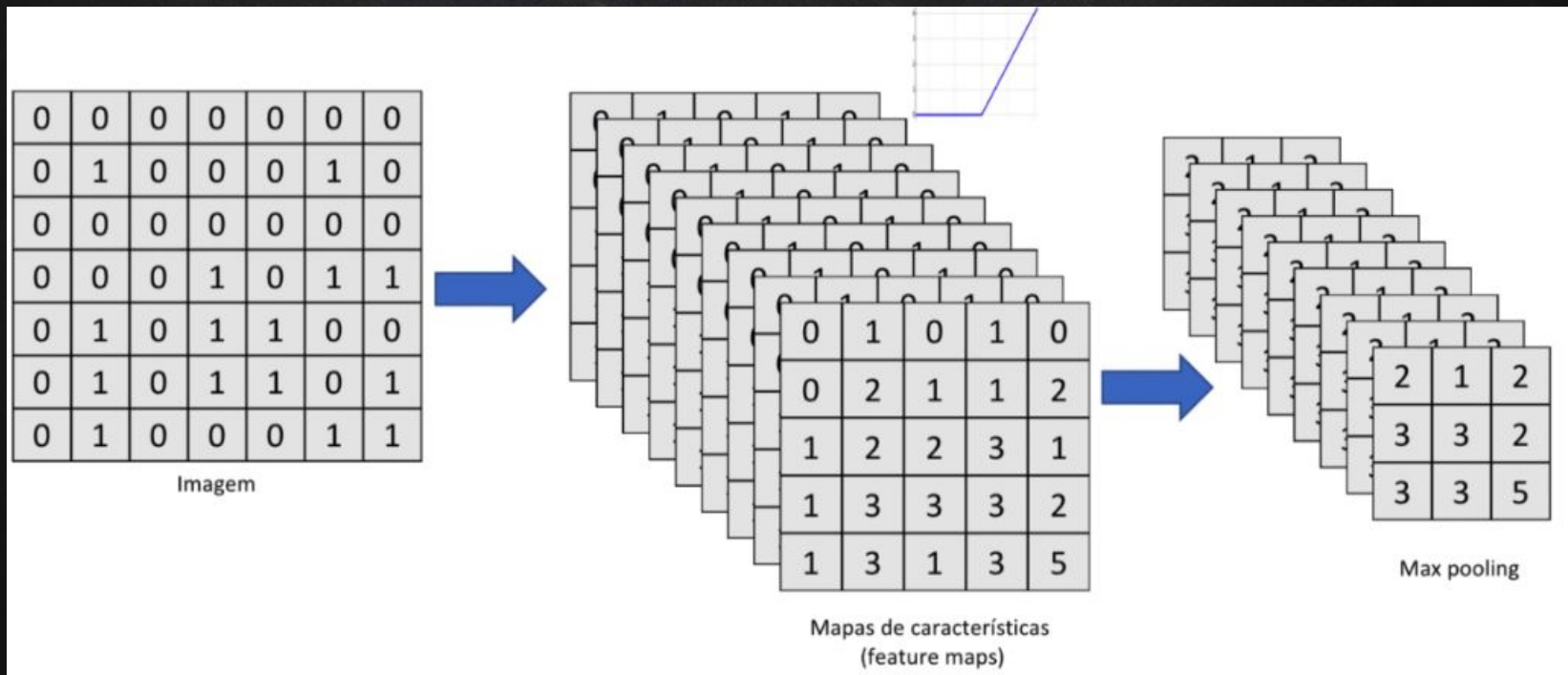
0	1	0	1	0
0	2	1	1	2
1	2	2	3	1
1	3	3	3	2
1	3	1	3	5

Mapa de características
(feature map)

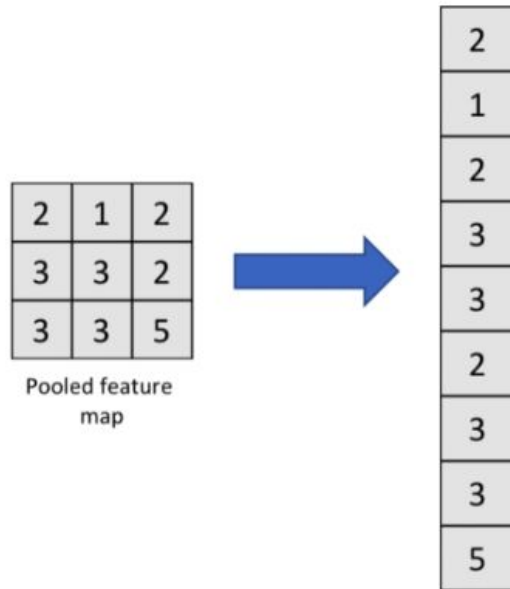


2	1	2
3		

POOLING



FLATTENING





IMPLEMENTAÇÃO

Criando e exportando um modelo:

https://github.com/Ricardovcn/minicurso-tensorflow-android/blob/master/criando_e_exportando_o_modelo.ipynb

Utilizando o modelo em um App android:

https://github.com/Ricardovcn/minicurso-tensorflow-android/tree/master/mnista_android

Exemplo prático:

<https://github.com/MindorksOpenSource/AndroidTensorFlowMachineLearningExample>



FONTES

<https://www.youtube.com/watch?v=kFWKdLOxykE>

https://github.com/llSourcell/A_Guide_to_Running_Tensorflow_Models_on_Android

<https://www.udemy.com/course/deep-learning-com-python-az-curso-completo/>



THANKS!

Any questions?

You can find me at
@Ricardovcn
ricardovitorcn@gmail.com