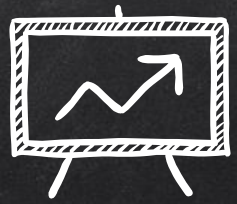


Rodando modelos do Tensorflow no Android

HELLO!

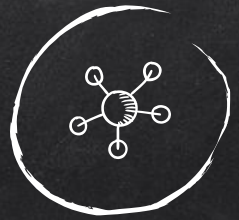
Ricardo Vitor Costa Neto.

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

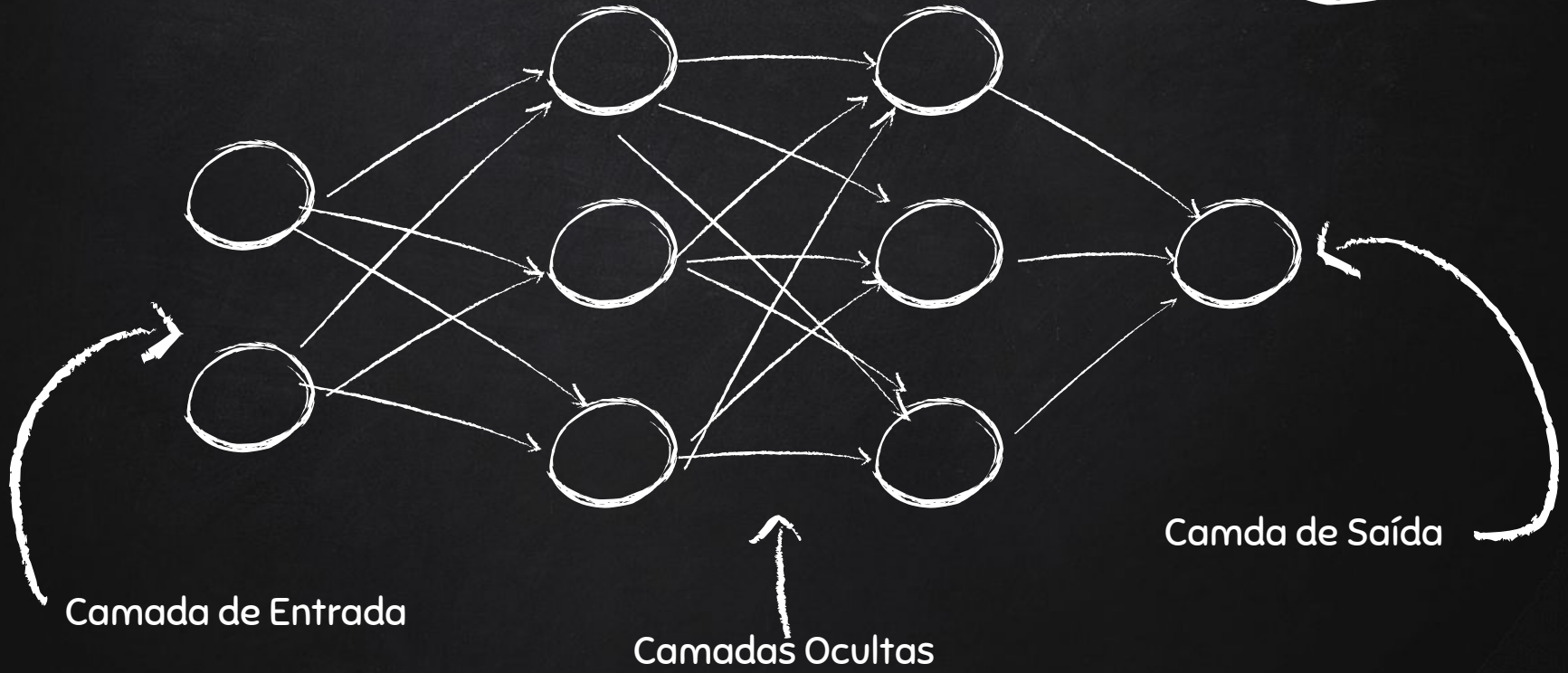
TensorFlow é uma biblioteca de código aberto para aprendizado de máquina aplicável a uma ampla variedade de tarefas.

Redes Neurais

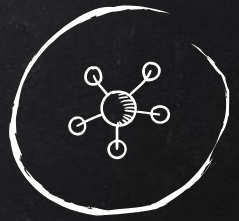
Redes neurais são sistemas de computação com nós interconectados que funcionam como os neurônios do cérebro humano.

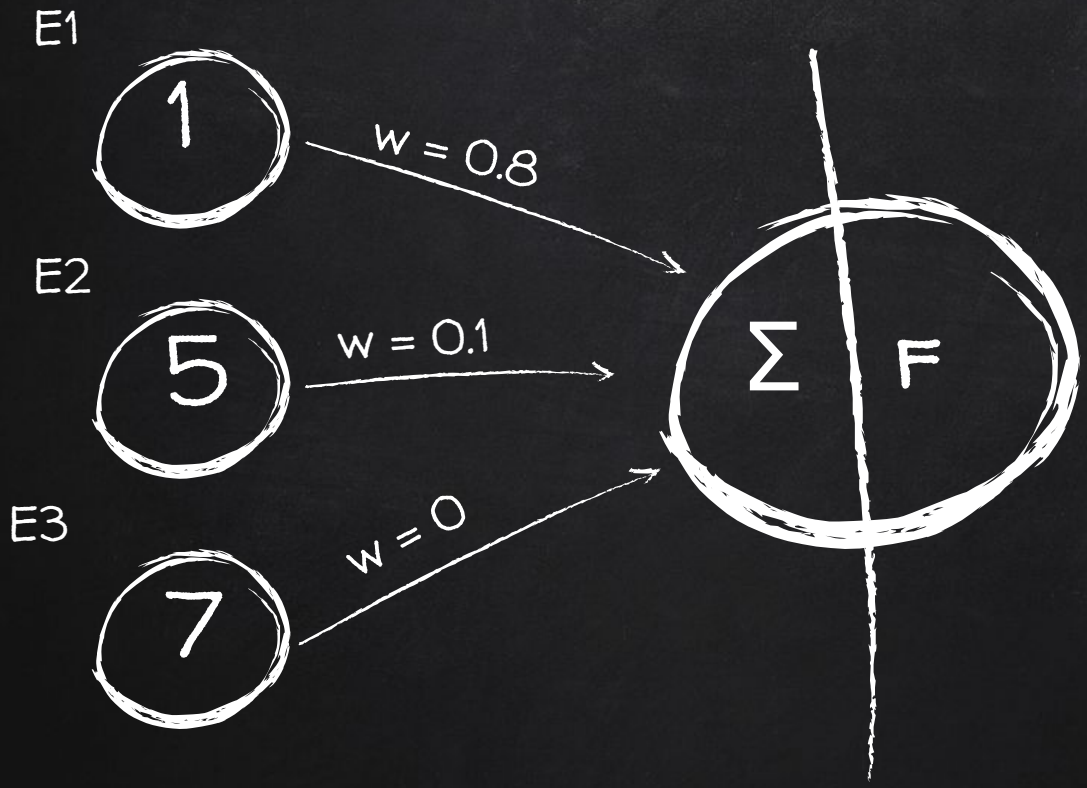
Mais informações sobre o TensorFlow: https://www.tensorflow.org.

Exemplo de Rede Neural



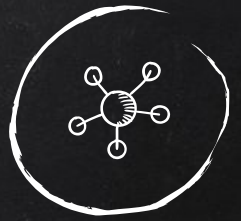
Neurônio Artificial



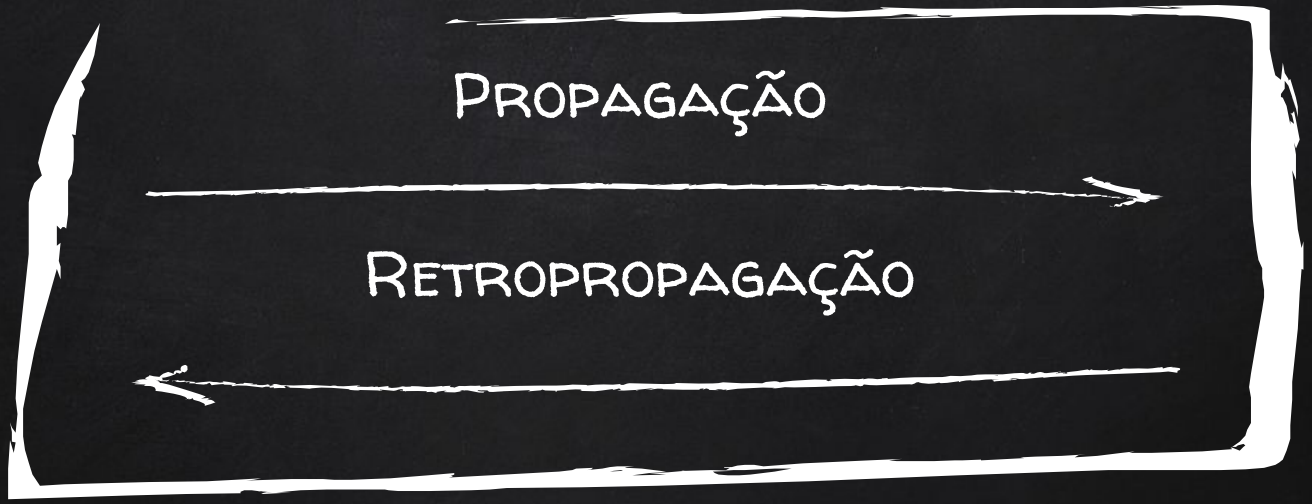


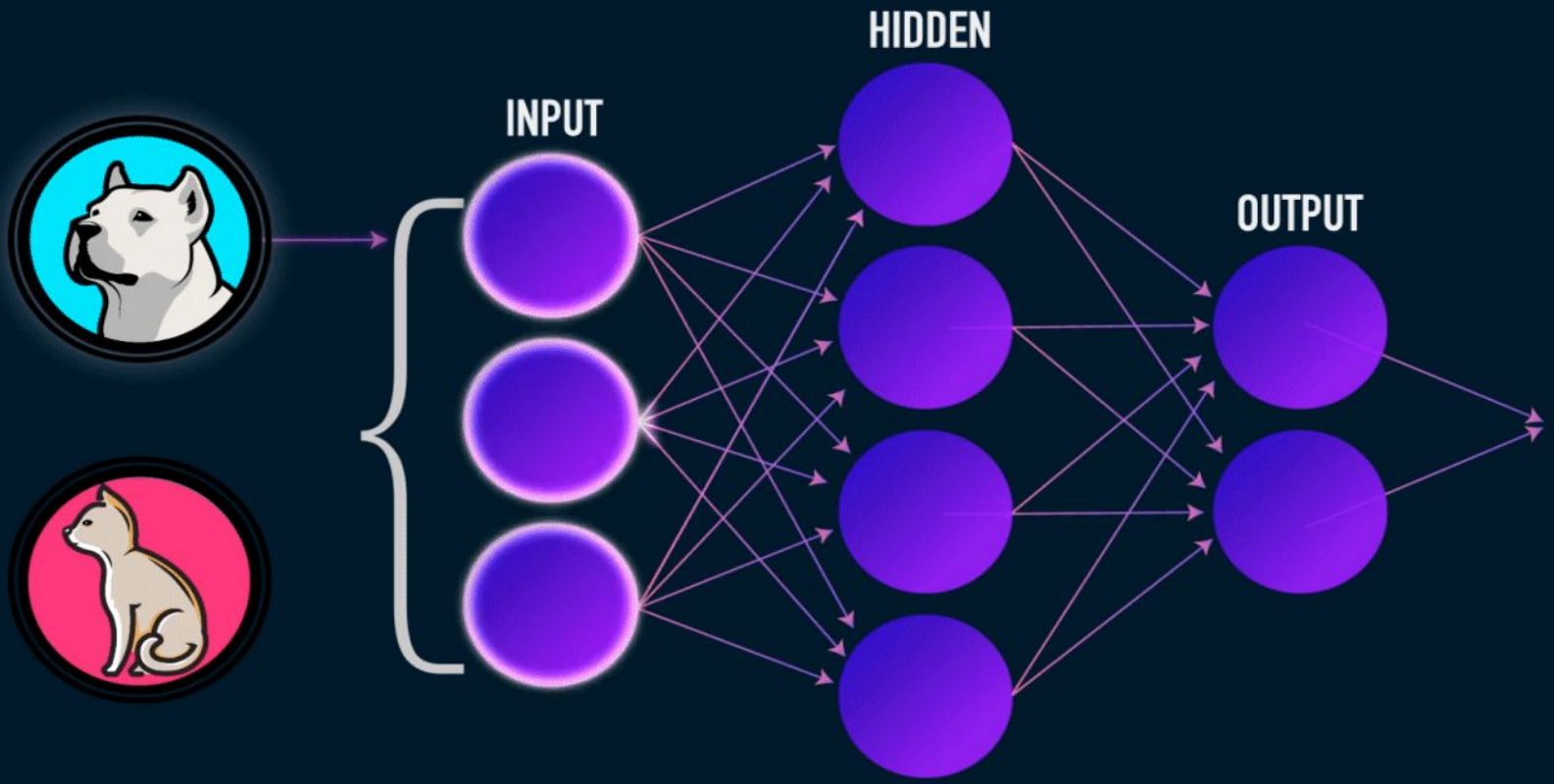
E = Entrada;
w = Pesos das arestas;
Σ = Somatório / Bias;
F = Função de ativação.

Back Propagation

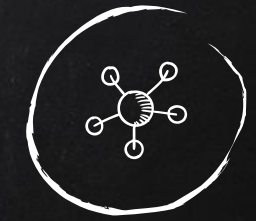


Duas fases:



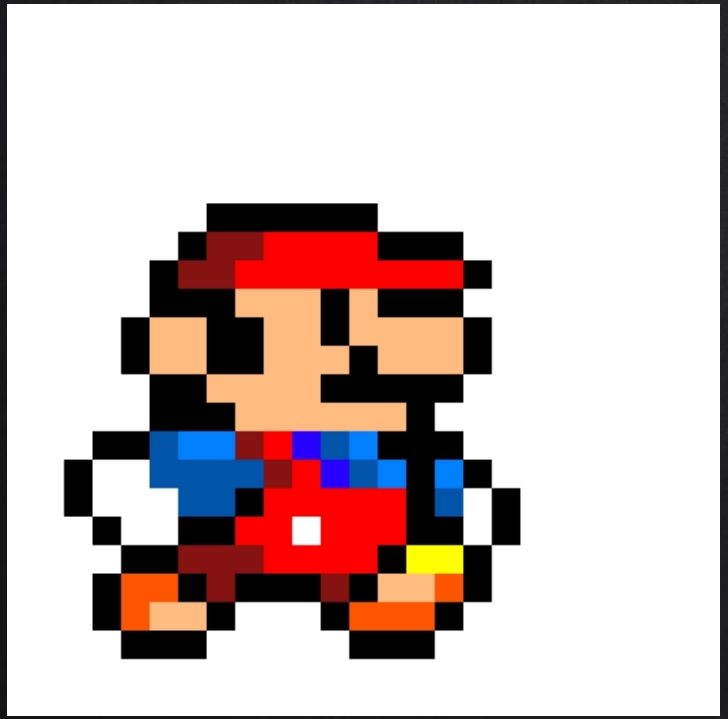

HIDDEN
INPUT
OUTPUT

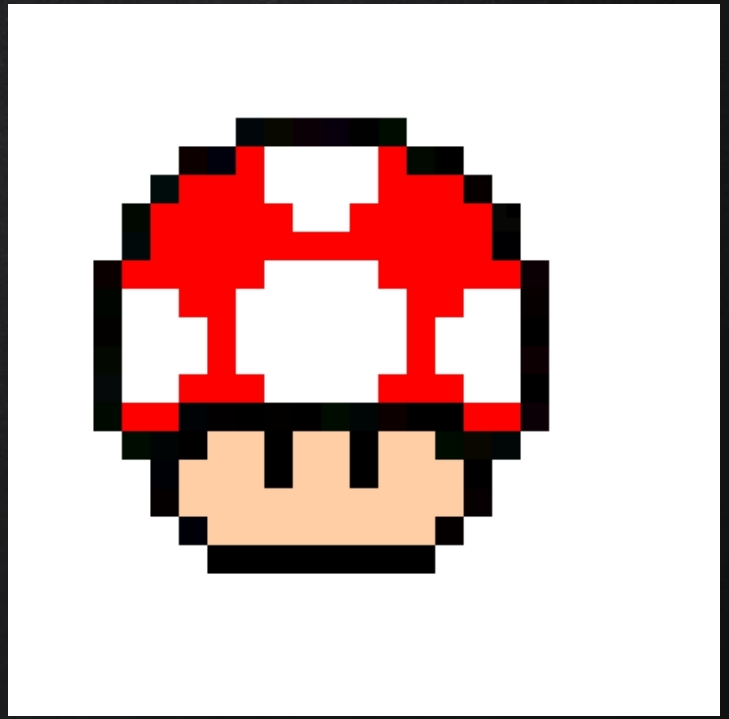
Redes Neurais Convolucionais



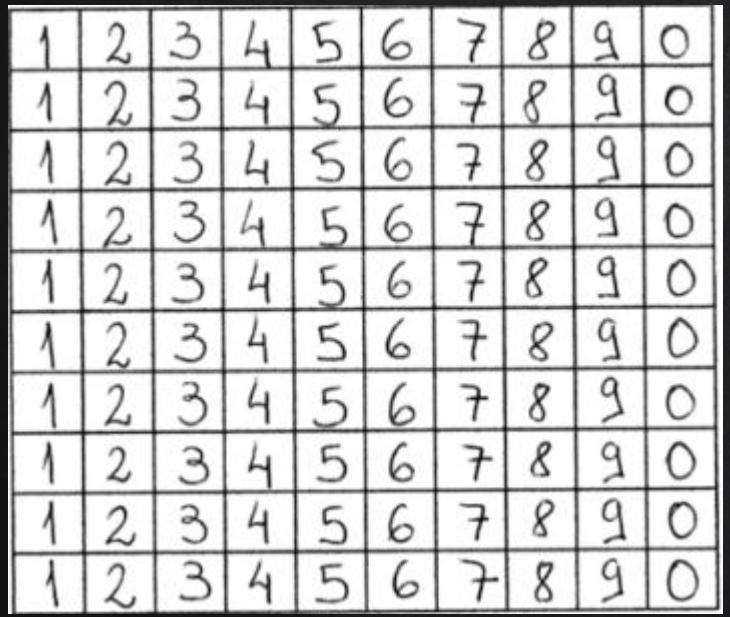
São usadas especificamente para a área de visão computacional. Essa área se preocupa em colocar dentro do computador a capacidade de visão.

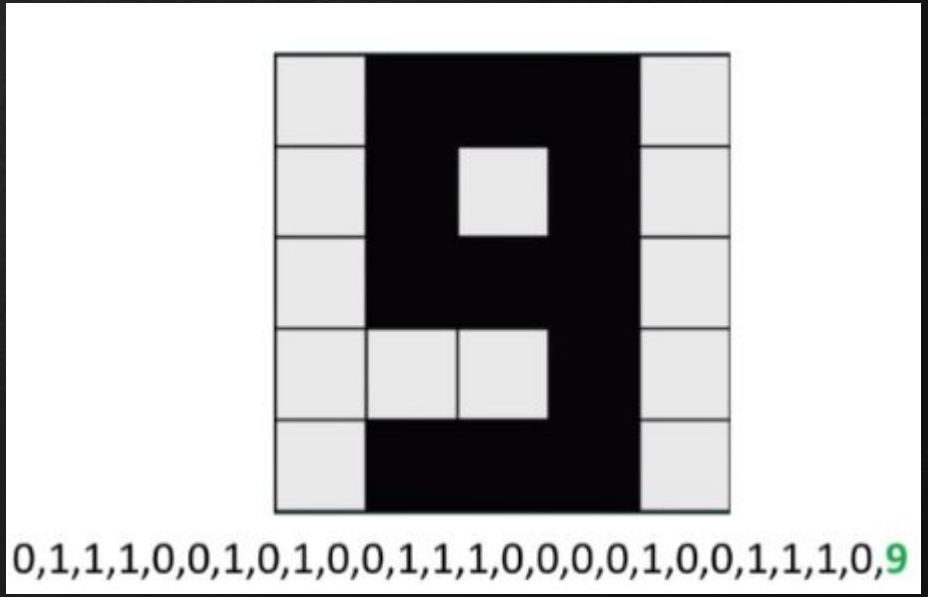
Redes Neurais Convolucionais



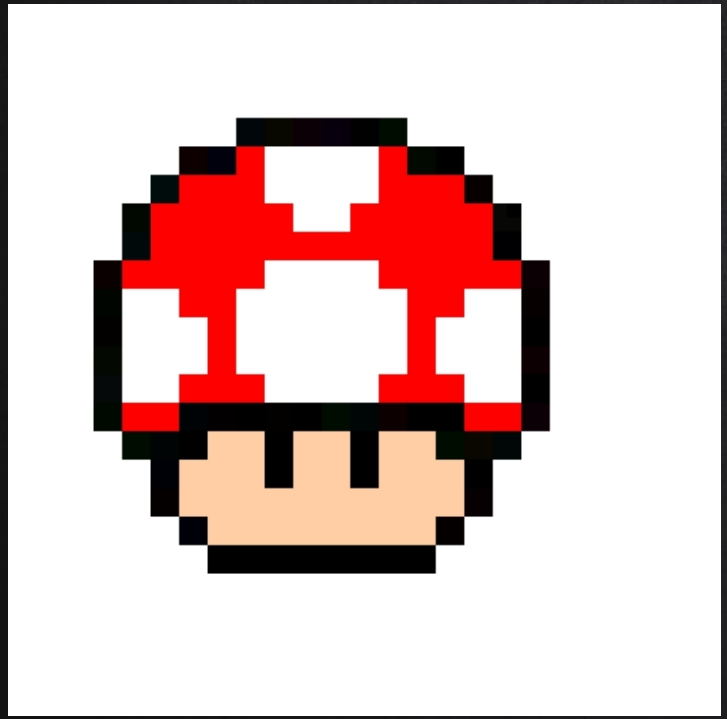


Redes Neurais Convolucionais



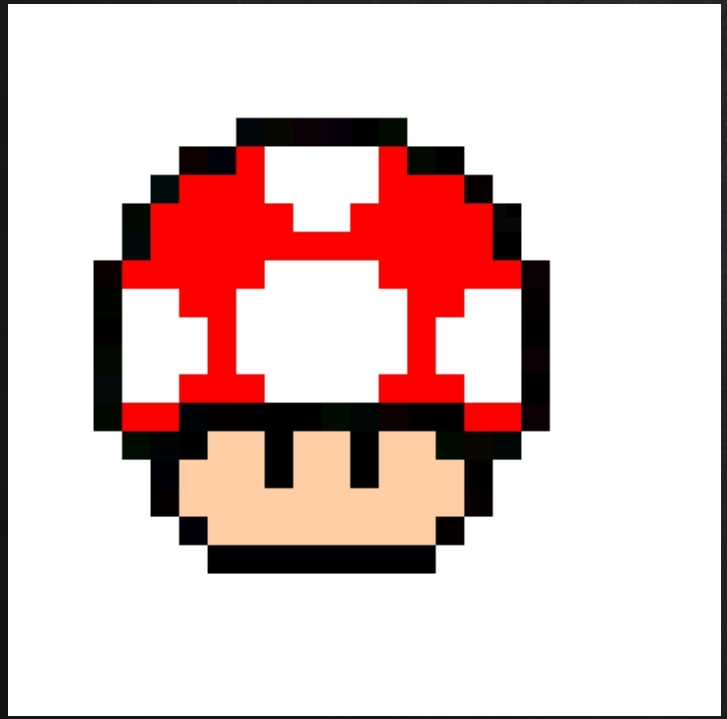


Redes Neurais Convolucionais



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

Redes Neurais Convolucionais



16 x 16 = 256 x 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, automaticamente, e utiliza apenas as partes relevantes da imagem (as melhores características).

Ela utiliza a rede neural tradicional (densa), porém existe um pré-processamento feito na imagem antes dela entrar na rede densa.

Implementação

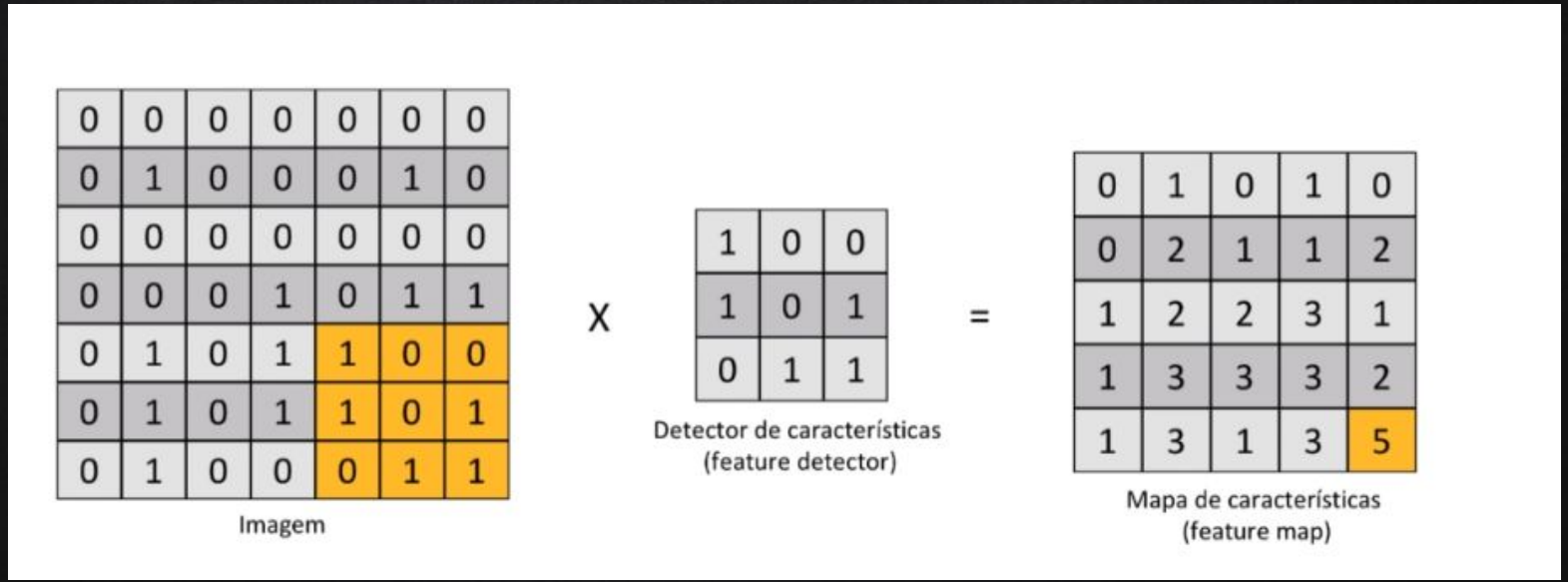
Implementação

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

Implementação

Operador de Convolução → Pooling → FLattening → Rede Neural Densa

Operador de convolução

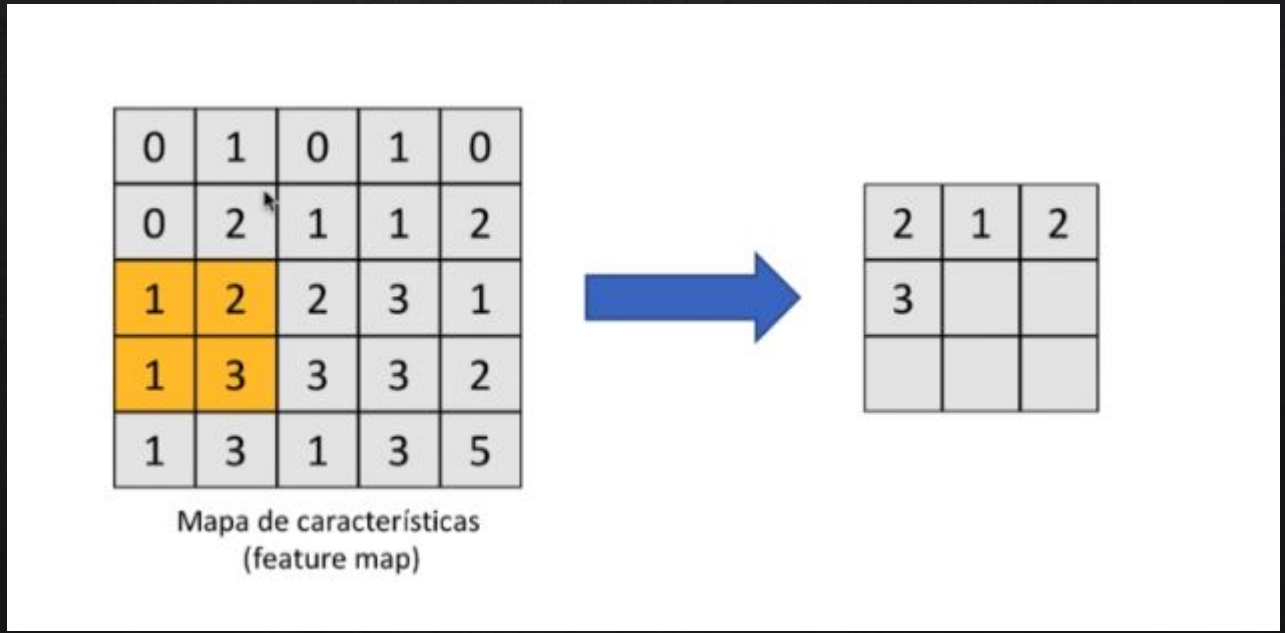


Mais sobre operador de convolução: https://en.wikipedia.org/wiki/Kernel_(image_processing)

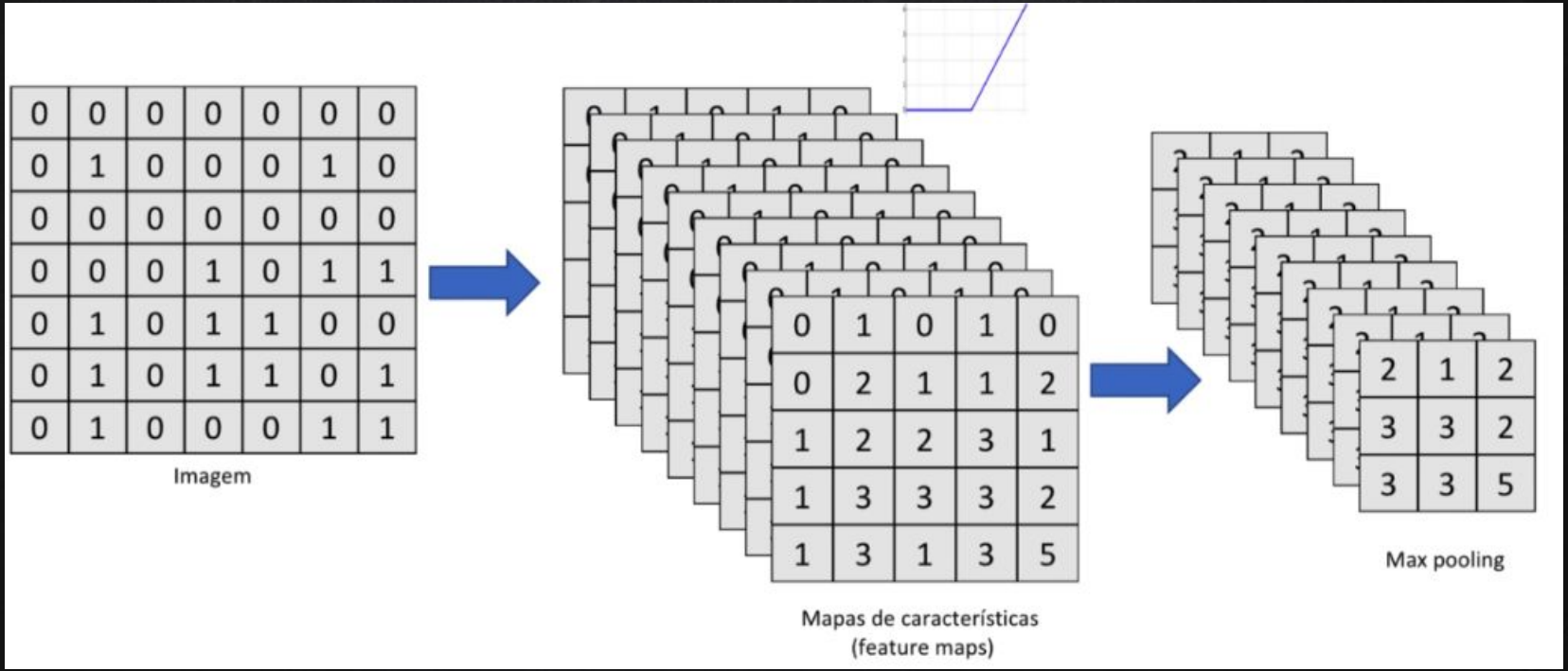
Exemplo prático: http://setosa.io/ev/image-kernels/

Pooling

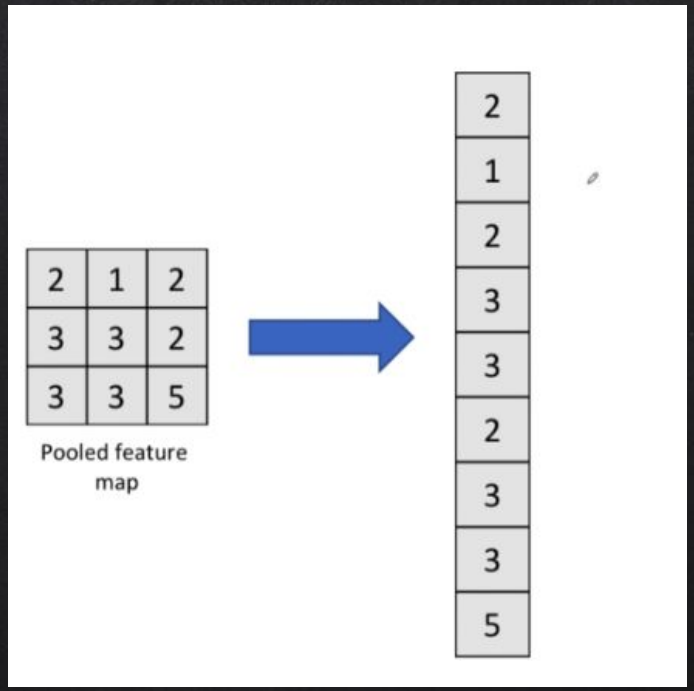
Reduz overfitting e Ruídos desnecessários.



Pooling


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
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
Mapas de características
(feature maps)
2 1 2
3 3 2
3 3 5
Max pooling

Flattening



Implementação

Exemplo prático: https://scs.ryerson.ca/~aharley/vis/conv/

Criando e exportando um modelo: https://github.com/Ricardovcn

Utilizando o modelo em um App android: https://github.com/Ricardovcn

THANKS!

Any questions?

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