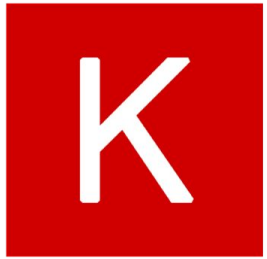


Keras

+

TensorFlow Lite

Construindo modelos para dispositivos móveis

Jeziel Lago

THE
DEVELOPER'S
CONFERENCE

Jeziel Lago

- Mobile Developer
- Entusiasta de Machine Learning
- Mestrado em Computação Aplicada
  - *Classificação de imagens histopatológicas com deep learning*
- Computer Vision Expert Nanodegree

jeziellago

Por quê abordar esse tema?











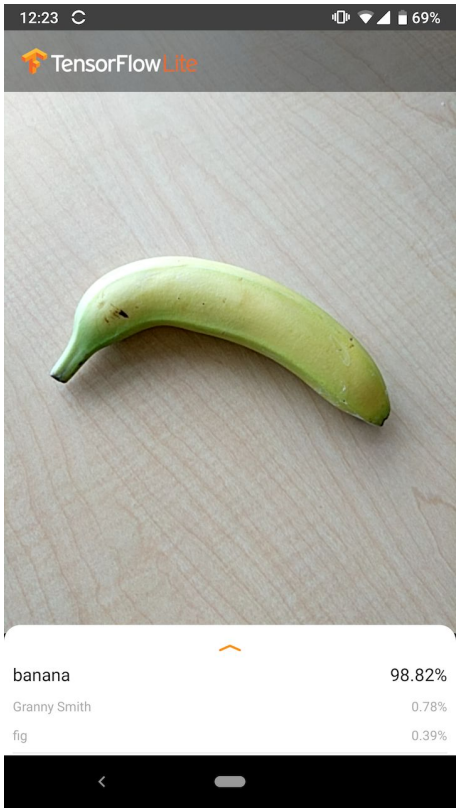


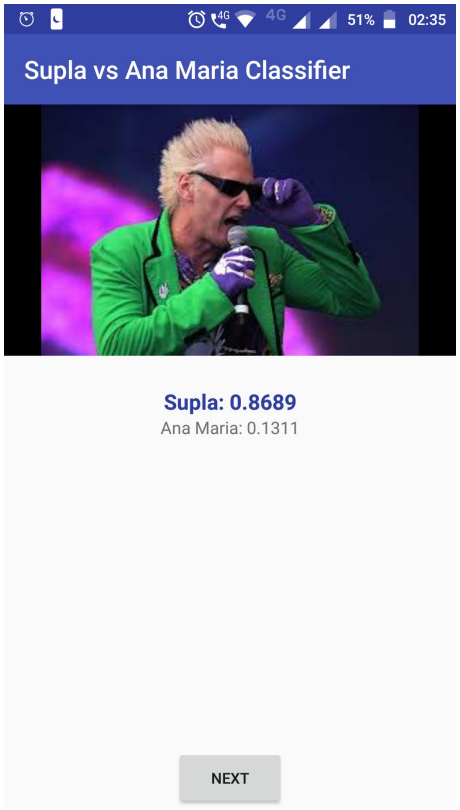



ifood

O que já temos hoje on-device?

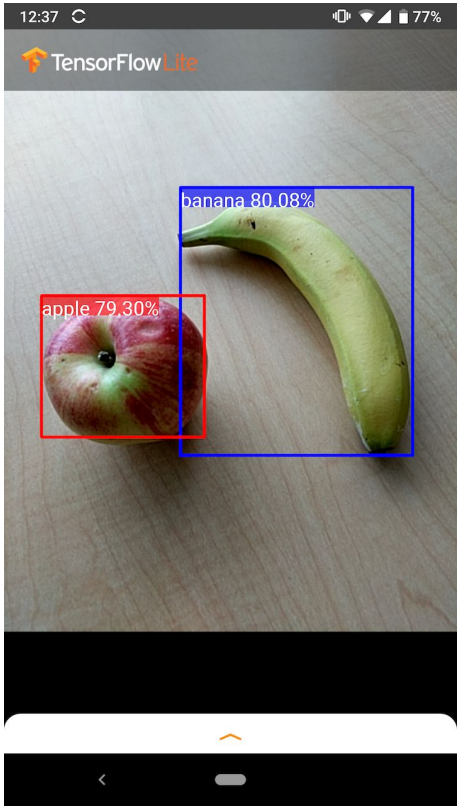
Classificação de Imagens

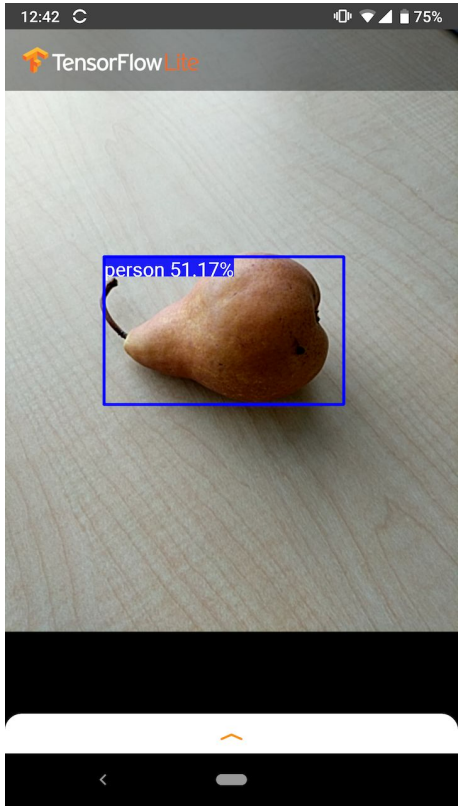




THE DEVELOPER'S CONFERENCE

Detecção de Objetos



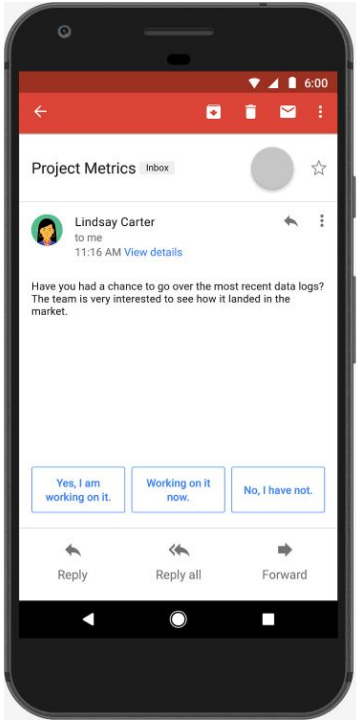


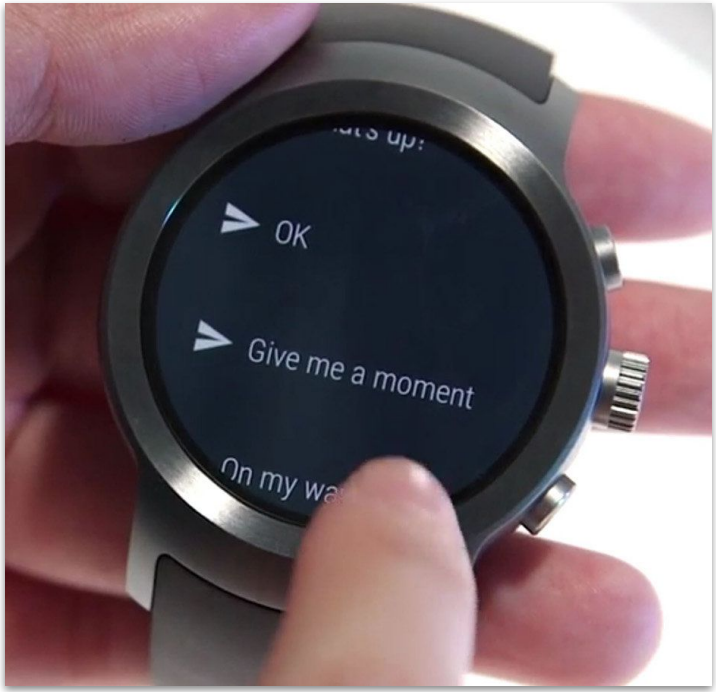
THE DEVELOPER'S CONFERENCE

Segmentação

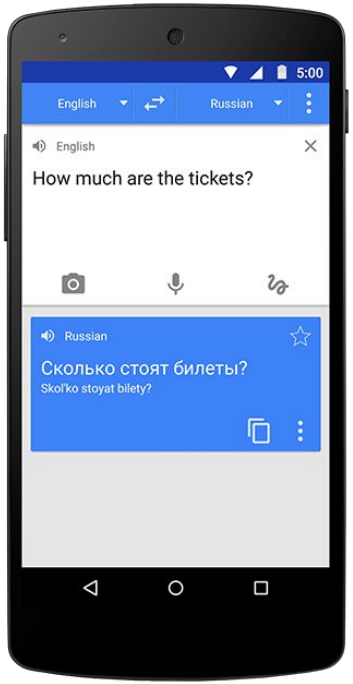
input image

Sugestão de Respostas





Text-to-Speech / Speech-to-Text



Vantagens

Vantagens ML on-device

- **Baixa latência**: Não precisa de um servidor
- **Privacidade**: Os dados não saem do device
- **Funciona offline**: Não precisa de conexão com a internet
- **Baixo consumo de energia**: Conexões de rede consomem mais energia

Desafios

Desafios ML on-device

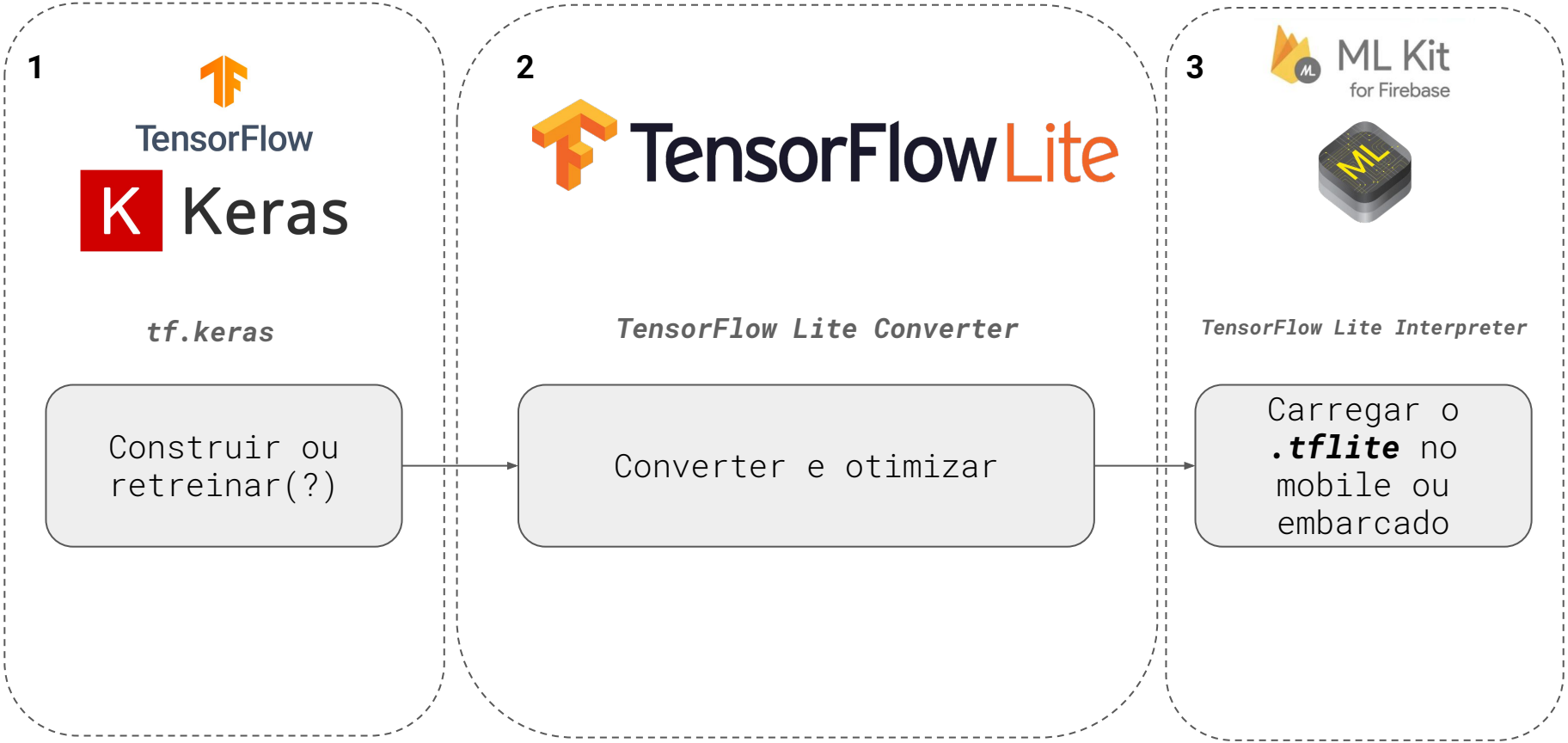
- Memória limitada
- CPU limitada
- Armazenamento limitado
- Microcontroladores (Ops limitadas)

IMPORTANTE!

Atualmente, existem maneiras (não triviais) de treinar o modelo no device, mas por conta das limitações citadas, as soluções atuais realizam apenas operações de inferência.

Construindo um modelo para rodar on-device

“Pipeline”


1
TensorFlow
Keras
tf.keras
Construir ou retreinar(?)
2
TensorFlow Lite
TensorFlow Lite Converter
Converter e otimizar
3
ML Kit for Firebase
TensorFlow Lite Interpreter
Carregar o .tflite no mobile ou embarcado

Antes de construir um modelo, reflita:

- **Não reinvente a roda**!
- **Avalie** o quão **específico** é o seu problema
- **Conexão** com a **internet** disponível no device?*
- Em quais devices pretende utilizar o modelo?

Converter o modelo para *.tflite*

```
# salva o modelo `keras`

model.save('trained_model.h5')

...

from tensorflow.lite import TFLiteConverter, Optimize

# cria um converter a partir do modelo keras salvo anteriormente

converter = TFLiteConverter.from_keras_model_file('trained_model.h5')

tflite_model = converter.convert()

open('trained_model.tflite', "wb").write(tflite_model)
```

Otimizando o modelo

- **Quantization**: modelos ~4x menores
  - Reduz o tamanho dos floats dos pesos e ativações para 8-bit.
  - Pequena perda de acurácia
  - Reduz os custos de acesso à memória para leitura e armazenamento

https://www.tensorflow.org/lite/performance/model_optimization

Otimizando o modelo

Model	Top-1 Accuracy (Original)	Top-1 Accuracy (Post Training Quantized)	Top-1 Accuracy (Quantization Aware Training)	Latency (Original) (ms)	Latency (Post Training Quantized) (ms)	Latency (Quantization Aware Training) (ms)	Size (Original) (MB)	Size (Optimized) (MB)
Mobilenet-v1-1-224	0.709	0.657	0.70	124	112	64	16.9	4.3
Mobilenet-v2-1-224	0.719	0.637	0.709	89	98	54	14	3.6
Inception_v3	0.78	0.772	0.775	1130	845	543	95.7	23.9
Resnet_v2_101	0.770	0.768	N/A	3973	2868	N/A	178.3	44.9

Converter & Otimizar o modelo: *8-bit precision*

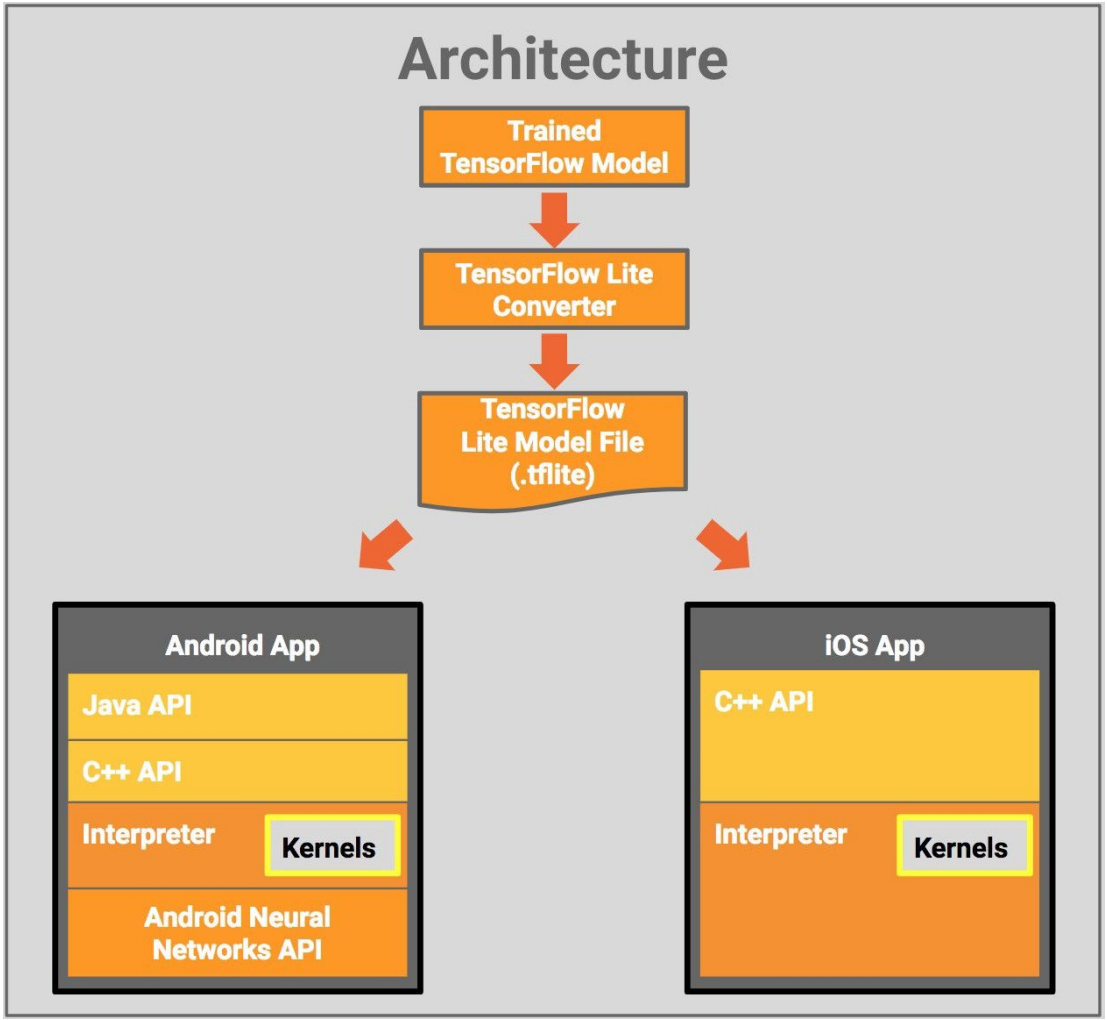
```
from tensorflow.lite import TFLiteConverter, Optimize

converter = TFLiteConverter.from_keras_model_file('trained_model.h5')

converter.optimizations = [Optimize.OPTIMIZE_FOR_SIZE]

tflite_quant_model = converter.convert()

open('trained_model.tflite', "wb").write(tflite_quant_model)
```

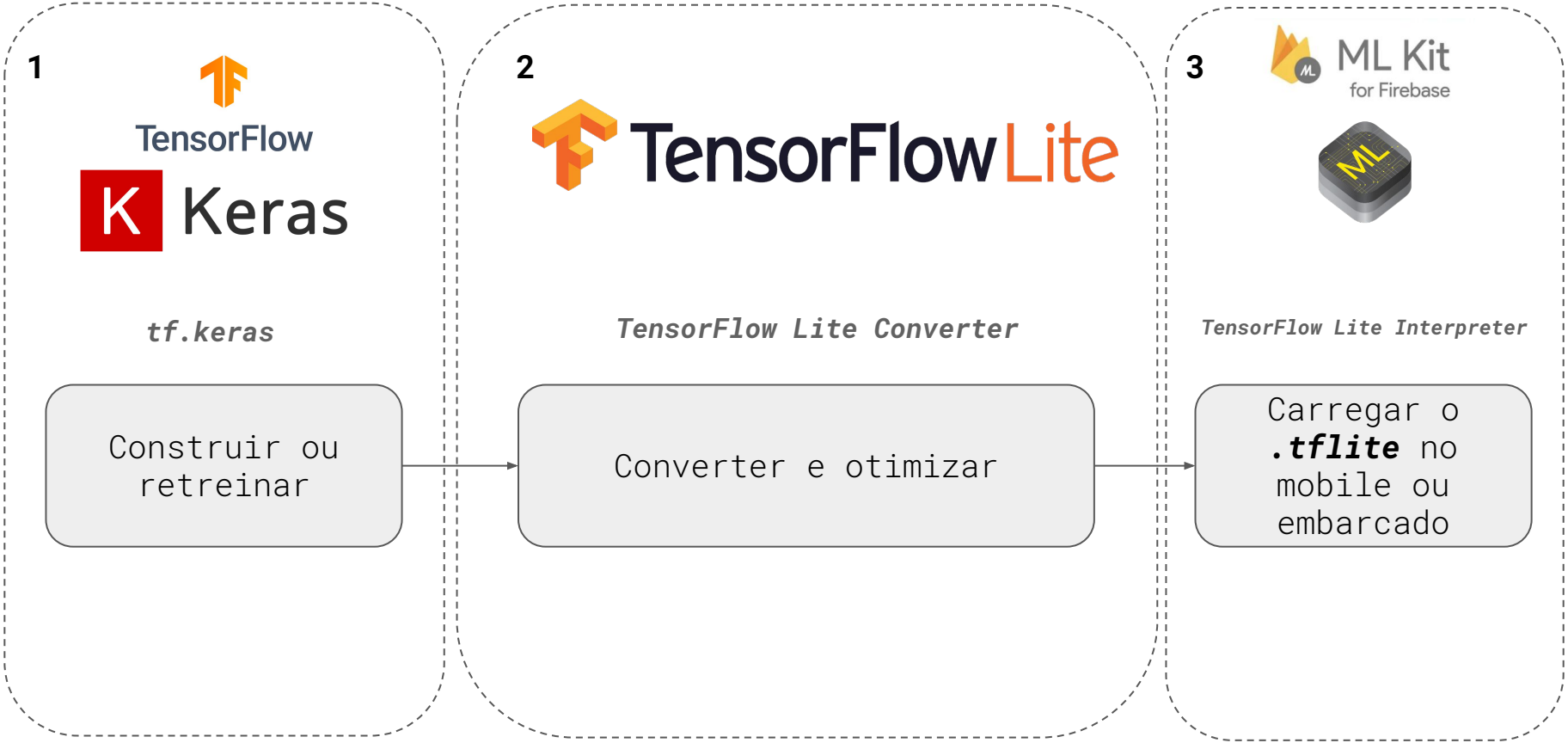
Converter & Otimizar o modelo: *FULL*

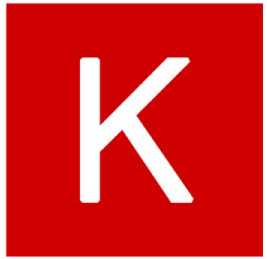
```
from tensorflow.lite import TFLiteConverter, Optimize

converter = TFLiteConverter.from_keras_model_file('trained_model.h5')

converter.optimizations = [Optimize.DEFAULT]

converter.representative_dataset = representative_dataset_gen

tflite_quant_model = converter.convert()

open('trained_model.tflite', "wb").write(tflite_quant_model)
```

Converter & Otimizar o modelo: *only output Int*

```
from tensorflow.lite import TFLiteConverter, Optimize, OpSet

converter = TFLiteConverter.from_keras_model_file('trained_model.h5')

converter.target_spec.supported_ops = [OpSet.TFLITE_BUILTINS_INT8]

tflite_quant_model = converter.convert()

open('trained_model.tflite', "wb").write(tflite_quant_model)
```


Architecture
Trained TensorFlow Model
TensorFlow Lite Converter
TensorFlow Lite Model File (.tflite)
Android App
Java API
C++ API
Interpreter
Kernels
Android Neural Networks API
iOS App
C++ API
Interpreter
Kernels


1
TensorFlow
Keras
tf.keras
Construir ou retreinar
2
TensorFlow Lite
TensorFlow Lite Converter
Converter e otimizar
3
ML Kit for Firebase
TensorFlow Lite Interpreter
Carregar o .tflite no mobile ou embarcado



Keras

+

TensorFlow Lite

Construindo modelos para dispositivos móveis

Jeziel Lago

THE
DEVELOPER'S
CONFERENCE

Links

https://www.tensorflow.org/lite

https://blog.usejournal.com/training-a-tensorflow-image-classification-model-and-integrating-it-into-ios-apps-148fe513f6e

https://heartbeat.fritz.ai/machine-learning-on-mobile-devices-3-steps-for-deploying-it-in-your-apps-48a0a24364a8

https://github.com/tensorflow/tensorflow/blob/master/tensorflow/lite/tutorials/post_training_quant.ipynb