Make you IoT Smarter with Tensorflow Lite ...

... to Design the Future of Vertical Farming

@alexis0duque
Who am I?

Alexis DUQUE
Director of Research & Development

- @alexis0duque
- alexisduque
- alexisd@rtone.fr
- alexisduque.me
- https://goo.gl/oNUWu6
Outline & What You Will Learn

- Indoor Vertical Farming
- Why Intelligence at the Edge?
- Introduction to Tensorflow Lite. How to Use It?
- Setup your laptop & RPI
- Build and train a small model to predict lettuce weight
- Convert and deploy it on a RPI
- Run predictions on IoT devices
- Benchmarks
- Further Work
Indoor Vertical Farming
System Architecture

avg $\text{CO}_2$ (double, in ppm)

avg dissolved oxygen (double, in ppm)

avg electrical conductivity (double, in $\mu$S/cm)

avg RedOx potential (double, in mV)

avg PPFD (Photosynthetic Photon Flux Density, double, in $\mu$mol/m$^2$/s)

avg water pH (double, in -)

average humidity (double, in %)

average temperature (double, in °C)
Irrigation System

- HPA
- Nebulization
- NFT
- Ebb & Flow

https://www.hydroponic-urban-gardening.com/hydroponics-guide/various-hydroponics-systems
System Architecture

Sensors

Motherboard

OpenCV

TensorFlow

TensorFlowLite
Computer Vision
System Architecture

Cloud

Motherboard

OpenCV

TensorFlowLite
System Architecture

Cloud

R&D

jupyter

T

Sensors

Motherboard

OpenCV

TensorFlow

Lite

FOSDEM #IoT #Tensorflow #AI

@alexis0duque
Why Machine Learning?

Is the system working as expected?
When my lettuce will be ready to be eaten?
What should I do to make vegetables looks and tastes better?
What should I do to make it grow faster?

Flavor-cyber-agriculture: Optimization of plant metabolites in an open-source control environment through surrogate modeling

Why Edge Computing?

- Infrastructure and cloud cost
- Scalability
- Must work in the field, without internet
- Network Latency
- Privacy
TensorFlow

Open source library created by Google
Platform for Machine Learning
2.0 (released on October 2019)

Create, train, debug and use various machine learning model (neural network but not only!)
Keras, Lite, Tensorboard, Tensorflow Probability
Tensorflow Lite (TF-Lite)

Converter + Interpreter

Tensorflow vs TF-Lite?

• smaller model size
• faster inference
• mobile, embedded, MCU

but

• no training
• model is frozen => no re-training
• no transfer learning

#FOSDEM #IoT #Tensorflow #AI
Tensorflow Lite (TF-Lite)

Optimization

- Pruning
- Post Training Quantization

Delegate to offload execution

- GPU, TPU, DSP
ML Workflow with Tensorflow Lite

Import your dataset

Work on data: preprocessing, normalization, features selection

Build your model with Tensorflow

Train your model

Export and convert to .tflite
ML Workflow with Tensorflow Lite

Load your model (or grab one in [github.com/tensorflow/models](https://github.com/tensorflow/models))
Preprocess input data
Allocate Memory
Run inference
Interpret output
Setup on your Laptop

$ python3 --version
$ pip3 --version
$ virtualenv --version

$ virtualenv --system-site-packages -p python3 ./venv
$ source ./venv/bin/activate
$ pip install --upgrade pip
$ pip install --upgrade tensorflow=2.0
$ pip install numpy pandas jupyter jupyterlab notebook matplotlib
Setup on your RPI

Tensorflow Lite Interpreter
1. Using pip and official TF release (not always up to date)
2. Cross compile Tensorflow for ARMv7 on your laptop
3. Build Bazel and Tensorflow on your RPI (> 24h)
4. Using pip and a community built .whl package

https://github.com/PINTO0309/Tensorflowlite-bin
Setup on your RPI

Tensorflow Lite Interpreter

$ sudo apt install swig libjpeg-dev zlib1g-dev python3-dev python3-numpy unzip
$ wget https://github.com/PINTO0309/TensorflowLite-bin/raw/master/2.0.0/tflite_runtime-2.0.0-cp37-cp37m-linux_armv7l.whl
$ pip install --upgrade tflite_runtime-2.0.0-cp37-cp37m-linux_armv7l.whl
Demo

**Part 1** - Build, Train and Convert a simple Neural Network model to predict lettuce weight

**Part 2** - Deploy your tflite model on RPI and run inference
Tensorflow Lite Limitations

Reinforcement Learning
Transfer Learning
Recurrent Neural Network (RNN) like LSTM
Operation Compatibility

https://www.tensorflow.org/lite/guide/ops_compatibility
https://github.com/tensorflow/tensorflow/blob/master/tensorflow/lite/experimental/examples/lstm/g3doc/README.md
Further Work

Training at the Edge
Transfer Learning
Federated Learning
Summary

Build, Train, Optimize, Convert on laptop
Deploy, Infer on device
Some operations are not supported
Quantization does not affect accuracy
Inference on IoT and microcontrollers is feasible
Regression, anomalies detection, objects recognition, smart reply, etc.
Thanks!

frama.link/rtone-iot-tflite

frama.link/rtone-jobs

#FOSDEM #IoT #Tensorflow #AI
References

https://medium.com/tensorflow/how-to-get-started-with-machine-learning-on-arduino-7daf95b4157
https://www.tensorflow.org/lite
https://www.tensorflow.org
https://coral.withgoogle.com/
https://arxiv.org/abs/1902.01046
https://medium.com/tensorflow/tensorflow-model-optimization-toolkit-pruning-api-42cac9157a6a
TFLite on Microcontrollers

Inference on Cortex-M microcontroller

Only some operations are supported

Enough for hotword, gesture and speech recognition

Arduino Nano 33

#FOSDEM #IoT #Tensorflow #AI
TFLite on Microcontrollers

C++ API

```cpp
#include <TensorFlowLite.h>
// This is your tflite model
#include "lg_weight_model.h"
#include "tensorflow/lite/experimental/micro/kernels/all_ops_resolver.h"
#include "tensorflow/lite/experimental/micro/micro_interpreter.h"
#include "tensorflow/lite/schema/schema_generated.h"

TfLiteTensor *input = nullptr;
TfLiteTensor *output = nullptr;
```
TFLite on Microcontrollers

// Finding the min value for your model may require tests!
constexpr int kTensorArenaSize = 2 * 1024;
uint8_t tensor_arena[kTensorArenaSize];

// Load your model.
model = tflite::GetModel(g_weight_regression_model_data);
// This pulls in all the operation implementations we need.
static tflite::ops::micro::AllOpsResolver resolver;

// Build an interpreter to run the model with.
static tflite::MicroInterpreter static_interpreter(
    model, resolver, tensor_arena, kTensorArenaSize, error_reporter);
interpreter = &static_interpreter;
TFLite on Microcontrollers

// Allocate memory for the model's tensors.
TfLiteStatus allocate_status = interpreter->AllocateTensors();
// Obtain pointers to the model's input and output tensors.
input = interpreter->input(0);
output = interpreter->output(0);
// Feed the interpreter with the input value
float x_val = random(0, 10);
inpu->data.f[0] = x_val;

// Run Inference
TfLiteStatus invoke_status = interpreter->Invoke();
// Get inference result
float y_val = output->data.f[0];