SciTinyML - ICTP workshop
Scientific Use of Machine Learning on Low Power Devices

Setting up the software tools

Prof. Marcelo José Rovai
UNIFEI - Universidade Federal de Itajubá, Brazil
Web: https://github.com/Mjrovai
Who I am

- Brazilian from São Paulo, **Data Science Master’s degree by UDD, Chile**, and MBA by IBMEC (INSPER), Brazil.
- Graduated in 1982 as an **Engineer from UNIFEI** with Specialization from Poli/USP, both in Brazil.
- Worked as a **teacher, engineer, and executive** in several technology companies such as CDT/ETEP, AVIBRAS Aeroespacial, SID Informática, ATT-GIS, NCR, DELL, COMPAQ (HP), and more recently at IGT, where I continue as a Senior Advisor for Latin America.
- **Write about electronics**, publishing my works in sites as MJRoBot.org (Editor/Writer), Hackster.io (#1 Contributor), Instructables.com, and Medium.com (TDS – Towards Data Science).
- **Volunteer Professor** at UNIFEI Engineering Institute: “Machine Learning applied to Embedded Devices” course (IESTI01).
- Active member of the **TinyML4D group**, an initiative to bring TinyML education to developing countries.
What is Tiny Machine Learning (TinyML)?

TinyML

- Fastest-growing field of ML
- Algorithms, hardware, software
- On-device sensor analytics
- Low power consumption
- Always-on ML
- Battery-operated

Source: TinyML4D seminar: "Why The Future of ML is Tiny and Bright" by Vijay Janapa Reddi
What Makes TinyML?
Machine Learning Workflow

Collect Data ➔ Preprocess Data
Machine Learning Workflow
Machine Learning Workflow
Machine Learning Workflow ("What")
Machine Learning Workflow ("Where")

Collect Data -> Preprocess Data -> Design a Model -> Train a Model -> Evaluate Optimize -> Convert Model -> Deploy Model -> Make Inferences
Machine Learning Workflow ("How")
Machine Learning Workflow ("How")

Collect Data ➔ Preprocess Data ➔ Design a Model ➔ Train a Model ➔ Evaluate Model ➔ Optimize ➔ Convert Model ➔ Deploy Model ➔ Make Inferences

- TensorFlow
- TensorFlow Lite
- TensorFlow Lite Micro
Machine Learning Workflow ("How")

1. Collect Data
2. Preprocess Data
3. Design a Model
4. Train a Model
5. Evaluate Optimise
6. Convert Model
7. Deploy Model
8. Make Inferences

Tools:
- TensorFlow
- TensorFlow Lite
- TensorFlow Lite Micro
Collect Data → Preprocess Data → Design a Model → Train a Model → Evaluate Optimize → Convert Model → Deploy Model → Make Inferences
EI Studio - Embedded ML platform

Learn more at http://edgeimpulse.com
Making things smarter

Edge Impulse is the leading development platform for machine learning on edge devices, free for developers and trusted by enterprises.

Trusted by thousands of embedded developers running critical machine learning projects across millions of data samples.
Sign up successful!

Thanks Marcelo Rovai!

You have successfully signed up for Edge Impulse.

Click here to build your first ML model!

Start building embedded machine learning models today.

© 2021 EdgeImpulse Inc. All rights reserved
Welcome to your new Edge Impulse project!

You’re ready to add real intelligence to your edge devices. Let’s set up your project. What type of data are you dealing with?

- **Accelerometer data**
  Analyze movement of your device in real-time to predict machine failure, detect human gestures, or monitor rotating machines.

- **Audio**
  Listen to what’s happening around you to create voice interfaces, listen to keywords, detect audible events, or to hear what’s happening around your device.

- **Images**
  Add sight to your sensors with image classification or object detection - to detect humans and animals, monitor production lines or track objects.

- **Something else**
  Different sensor? No problem! You can collect and import data from any sensor, from environmental sensors to radars - and deploy your trained model back to virtually any device.

I know what I’m doing, hide this wizard!

Deploy
Package the complete impulse up, from signal processing code to trained model, and deploy it on your device. This ensures that the impulse runs with low latency and without requiring a network connection.
Welcome to your new Edge Impulse project!

Great! Here’s how you get started with accelerometer data:

Connect a development board
Get started with real hardware from Nordic, Arduino, OpenMV, ST, Eta Compute, Himax and Silabs, or connect any development board with the Data Forwarder.

**Connect your development board**

Tutorial: continuous motion recognition
Follow our end-to-end tutorial to collect accelerometer data, train a model, and deploy it back to your device to analyze movement in real-time.

**Read the tutorial**

I know what I’m doing, hide this wizard!

Let’s get started!
Marcelo Rovai / TinyML4D - Project Setup

This is your Edge impulse project. From here you acquire new training data, design impulses and train models.

Creating your first impulse (0% complete)

Acquire data
Every Machine Learning project starts with data. You can capture data from a development board or your phone, or import data you already collected.

Design an impulse
Teach the model to interpret previously unseen data, based on historical data. Use this to categorize new data, or to find anomalies in sensor readings.

Deploy
Package the complete impulse up, from signal processing code to trained model, and deploy it on your device. This ensures that the impulse runs with low latency and without requiring a network connection.

Sharing
Your project is private.

Make this project public

Summary

DEVICES CONNECTED 0

DATA COLLECTED 0

Collaborators

Marcelo Rovai
- Pre-Processing Data
- Design a Model
- Train a Model
These are devices that are connected to the Edge Impulse remote management API, or have posted data to the ingestion SDK.

No devices connected yet.

Learn how to connect a new device
Gesture Classification

- Manual gestures ("labels"):  
  - up-down  
  - left-right  
  - circle  
  - idle
Gesture Classification

• Manual gestures ("labels"):  
  • up-down  
  • left-right  
  • circle  
  • idle

• Data: collect & test using accelerometer as sensor
Collect data

You can collect data from development boards, from your own devices, or by uploading an existing dataset.

- Connect a fully supported development board
  - Get started with real hardware from a wide range of silicon vendors - fully supported by Edge Impulse.
  - Browse dev boards

- Use your mobile phone
  - Use your mobile phone to capture movement, audio or images, and even run your trained model locally. No app required.
  - Show QR code

- Use your computer
  - Capture audio or images from your webcam or microphone, or from an external audio device.
  - Collect data

- Data from any device with the data forwarder
  - Capture data from any device or development board over a serial connection, in 10 lines of code.
  - Show docs

- Upload data
  - Already have data? You can upload your existing datasets directly in WAV, JPG, PNG, CBOR, CSV or JSON format.
  - Go to the uploader

- Integrate with your cloud
  - The enterprise version of Edge Impulse integrates directly with the data stored in your cloud platform.
  - Contact us
Collect data

You can collect data from any smartphone. From your smartphone go to this URL, or scan the QR code below.
Collect data

Device phone_kq6ray4k is now connected

Get started!
These are devices that are connected to the Edge Impulse remote management API, or have posted data to the ingestion SDK.

<table>
<thead>
<tr>
<th>NAME</th>
<th>ID</th>
<th>TYPE</th>
<th>SENSORS</th>
<th>REMOTE</th>
<th>LAST SEEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>phone_kq6ray4k</td>
<td>phone_kq6ray4k</td>
<td>MOBILE_CLIENT</td>
<td>Accelerometer, Microphone</td>
<td>✔</td>
<td>Today, 12:06:04</td>
</tr>
</tbody>
</table>

© 2021 EdgeImpulse Inc. All rights reserved
No devices connected

Device

Collected data

No data collected yet

Let's collect some data
Collect Data

DATA ACQUISITION (TinyML4D - PROJECT SETUP)

Training data  Test data

Did you know? You can capture data from any device or development board, or upload your existing datasets - Show options

DATA COLLECTED

LABELS 0

Record new data

Device  phone_kqGray4k

Label  up_down

Sample length (ms.)  10000

Frequency  62.5Hz

Sensor  Accelerometer, Microphone, Camera

Start sampling

RAW DATA

Click on a sample to load...
Collect Data

EDGE IMPULSE

Training data  Test data

Did you know? You can capture data from any device or development board, or upload your existing datasets - Show options

DATA COLLECTED
10s

LABELS
1

Record new data

Device:

Phone.kg454y4k

Label:

up_down

Sample length (ms): 10000

Sensor:

Accelerometer

Frequency: 62.5Hz

Start sampling

Collected data:

SAMPLE NAME  LABEL  ADDED  LENGTH
up_down.2gbe7ljv  up_down  Today, 12:36:16  10s

RAW DATA:

up_down.2gbe7ljv

Graph showing raw data with accelerometer and gyroscope readings.
Collect Data
Pre-Processing Data
Design a Model
Train a Model
An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.
Add a processing block

**Spectral Analysis**
Great for analyzing repetitive motion, such as data from accelerometers. Extracts the frequency and power characteristics of a signal over time.

**Flatten**
Flatten an axis into a single value, useful for slow-moving averages like temperature data, in combination with other blocks.

**Image**
Preprocess and normalize image data, and optionally reduce the color depth.

**Audio (MFCC)**
Extracts features from audio signals using Mel Frequency Cepstral Coefficients, great for human voice.

**Audio (MPE)**
Extracts a spectrogram from audio signals using Mel-filterbank energy features, great for non-voice audio.

**Spectrogram**
Extracts a spectrogram from audio or sensor data, great for non-voice audio or data with continuous frequencies.

**Audio (Syntiant)**
Syntiant only. Compute log Mel-filterbank energy features from an audio signal.

**Raw Data**
Use data without pre-processing. Useful if you want to use deep learning to learn features.
An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.
An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.

### Add a learning block

Some learning blocks have been hidden based on the data in your project.

<table>
<thead>
<tr>
<th>Description</th>
<th>Author</th>
<th>Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification (Keras)</td>
<td>EdgeImpulse Inc.</td>
<td></td>
</tr>
<tr>
<td>Learn patterns from data, and can apply these to new data. Great for categorizing movement or recognizing audio.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anomaly Detection (K-means)</td>
<td>EdgeImpulse Inc.</td>
<td></td>
</tr>
<tr>
<td>Find outliers in new data. Good for recognizing unknown states, and to complement classifiers.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Regression (Keras)</td>
<td>EdgeImpulse Inc.</td>
<td></td>
</tr>
<tr>
<td>Learn patterns from data, and can apply these to new data. Great for predicting numeric continuous values.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.

**Time series data**
- Axes: accX, accY, accZ
- Window size: 2000 ms
- Window increase: 200 ms
- Frequency (Hz): 62.5
- Zero-pad data: 

**Spectral Analysis**
- Name: Spectral features
- Input axes: accX, accY, accZ

**Neural Network (Keras)**
- Name: NN Classifier
- Input features: Spectral features
- Output features: 4 (circle, idle, left_right, up_down)

**Output features**
- 4 (circle, idle, left_right, up_down)

Add a processing block
Add a learning block

Save Impulse
Preprocess Data
Preprocess Data

Spectral features

Training set
- Data in training set: 6m 23s
- Classes: 4 (circle, idle, left_right, up_down)
- Window length: 2000 ms.
- Window increase: 80 ms.
- Training windows: 3,782

Feature generation output
- Scheduling job in cluster...
- Job started
- Creating windows from 42 files...
- Creating windows from files...
- Creating 3783 windows: circle: 879, idle: 3881, left_right: 969, up_down: 863
- Job completed

Feature explorer (3,783 samples)
- X Axis
- Y Axis
- Z Axis
  - accx RMS
  - accy RMS
  - accz RMS

On-device performance
- Processing time: 7 ms.
- Peak RAM usage: 5 KB
Preprocess Data

Design a Model

NN CLASSIFIER (TINYMLD - PROJECT SETUP - GESTURE CLASSIFICATION)

#1 - Click to set a description for this version

Neural Network settings

Training settings

- Number of training cycles (EPOCHS): 20
- Learning rate (Lr): 0.0005

Neural network architecture

- Input layer (33 features)
- Dense layer (20 neurons)
- Dense layer (10 neurons)
- Add an extra layer
- Output layer (4 features)

Training output

- Input
- Input layer
- Dense
- kernel (33x20)
- bias (20)
- ReLU
- Dense
- kernel (20x10)
- bias (10)
- ReLU
- Dense
- kernel (10x4)
- bias (4)
- Softmax
- y_pred
Pre-Processing Data

Design a Model

Train a Model

Dataset

Acquire Training Data

Train Machine Algorithms

Impulse

Edge

Device

Collect

Train

Deploy

Validate

Tests

Embed On Edge

Test Data Flow

• Pre-Processing Data
• Design a Model
• Train a Model
## Test data

Set the 'expected outcome' for each sample to the desired output. This will automatically score the Impulse.

<table>
<thead>
<tr>
<th>SAMPLE NAME</th>
<th>EXPECTED OUTCOME</th>
<th>LENGTH</th>
<th>ACCURACY</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>testing.2800f...</td>
<td>up_down</td>
<td>9s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>testing.287w...</td>
<td>left_right</td>
<td>10s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>testing.287voc...</td>
<td>left_right</td>
<td>10s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>testing.287q...</td>
<td>up_down</td>
<td>10s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>up_down.287...</td>
<td>up_down</td>
<td>10s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>idle.2872mc6</td>
<td>idle</td>
<td>10s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>testing.285gm...</td>
<td>up_down</td>
<td>10s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>testing.285ga...</td>
<td>idle</td>
<td>4s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>testing.285gl...</td>
<td>circle</td>
<td>10s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>testing.285g3...</td>
<td>left_right</td>
<td>10s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>left-right.285f...</td>
<td>left_right</td>
<td>10s</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Test data

Set the 'expected outcome' for each sample to the desired outcome to automatically score the impulse.

<table>
<thead>
<tr>
<th>SAMPLE NAME</th>
<th>EXPECTED OUTCOME</th>
<th>LENGTH</th>
<th>ACCURACY</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>testing.2880f...</td>
<td>up_down</td>
<td>9s</td>
<td>100%</td>
<td>92 up_down</td>
</tr>
<tr>
<td>testing.287v...</td>
<td>left_right</td>
<td>10s</td>
<td>100%</td>
<td>95 left_right</td>
</tr>
<tr>
<td>testing.287v...</td>
<td>left_right</td>
<td>10s</td>
<td>100%</td>
<td>97 left_right</td>
</tr>
<tr>
<td>testing.287v...</td>
<td>up_down</td>
<td>10s</td>
<td>100%</td>
<td>97 up_down</td>
</tr>
<tr>
<td>up_down.287...</td>
<td>up_down</td>
<td>10s</td>
<td>100%</td>
<td>97 up_down</td>
</tr>
<tr>
<td>idle.287h2mc6</td>
<td>idle</td>
<td>10s</td>
<td>100%</td>
<td>97 idle</td>
</tr>
<tr>
<td>testing.285g...</td>
<td>up_down</td>
<td>10s</td>
<td>100%</td>
<td>97 up_down</td>
</tr>
<tr>
<td>testing.285g...</td>
<td>up_down</td>
<td>10s</td>
<td>100%</td>
<td>97 up_down</td>
</tr>
<tr>
<td>testing.285g...</td>
<td>up_down</td>
<td>10s</td>
<td>100%</td>
<td>97 up_down</td>
</tr>
<tr>
<td>testing.285g...</td>
<td>up_down</td>
<td>10s</td>
<td>100%</td>
<td>97 up_down</td>
</tr>
<tr>
<td>left-right.285f...</td>
<td>left-right</td>
<td>10s</td>
<td>98%</td>
<td>95 left-right, 1 idle</td>
</tr>
</tbody>
</table>

### Model testing output

Classifying data for NN Classifier...
Copying features from processing blocks...
Copying features from DSP block...
Copying features from DSP block OK
Classifying data for float32 model...
Scheduling job on cluster...
Job started
Classifying data for NN Classifier OK
Job completed

### Model testing results

**Accuracy**: 99.90%

- **Confusion Matrix**:

<table>
<thead>
<tr>
<th></th>
<th>CIRCLE</th>
<th>IDLE</th>
<th>LEFT_RIGHT</th>
<th>UP_DOWN</th>
<th>UNCERTAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIRCLE</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>IDLE</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>LEFT_RIGHT</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>UP_DOWN</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>UNCERTAIN</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
</tr>
</tbody>
</table>

- **F1 Score**: 1.90

### Feature explorer

- **Axes**: acce RMS, acce RMS, acce RMS
- Pre-Processing Data
- Design a Model
- Train a Model
You can collect data from this device from the **Data acquisition** page in the Edge Impulse studio.

- Collecting images?
- Collecting audio?
- Collecting motion?

Switch to classification mode

**Building project...**

**Downloading deployment...**
Make Inferences

Starting in 2 seconds...

Sampling...

Sampling...

Sampling...
Summary
Additional Free Resources

Google CoLab

Google Colaboratory or CoLab for short, allows you to write and execute Python in your browser, with zero configuration required, free access to GPUs and easy sharing. Google Colab is also an online integrated developer environment to design, train, and test our machine learning models. Here is an introduction to Google Colab. Watch Jake VanderPlas from Google give a wonderful intro to Colab.

Python for Data Science and ML Review

- A Whirlwind Tour of Python by Jake VanderPlas (e-book content)
- Learn the most important language for data science: Kaggle Python Tutorial
- Use TensorFlow and Keras to build and train neural networks for structured data: Kaggle Intro to Deep Learning

Hackster TinyML Tutorials

- "Listening Temperature" with Arduino Nano (Audio)
- Motion Recognition Using Raspberry Pi Pico (Accelerometers)
- Coffee disease classification with Seeed Maix Bit RISC-V board (Vision)

Imagine 2021 Day 3: Community Showcase
SciTinyML - ICTP workshop
Scientific Use of Machine Learning on Low Power Devices

Setting up the software tools

Prof. Marcelo José Rovai
UNIFEI - Universidade Federal de Itajubá, Brazil
Web: https://github.com/Mjrovai