From the cloud to the edge

Jan Jongboom
Edge Impulse
Edge Impulse

Leading development platform for machine learning on edge devices.

>100,000 new projects created in the past year on our community tier.

But... more importantly:

TRUSTED BY LEADING ENTERPRISES

- Oura
- poly
- Sony
- SAIC
- Lexmark
- NASA
- Ecolab
- Brambles
- Bosch
- Know Labs
Edge Impulse

Leading development platform for machine learning on edge devices.

>100,000 new projects created in the past year on our community tier.

But... more importantly:

TRUSTED BY LEADING ENTERPRISES

- Oura
- Poly
- Sony
- SAIC
- Lexmark
- NASA
- Ecolab
- Brambles
- Bosch
- Know Labs
Back to 2019...
Our very first customer

Qura creates a health wearable, originally focused on measuring sleep quality.

Wanted to take a data-driven approach to:
1) understand current accuracy
2) use data to build next generation algorithm.

And... eventually do more than just sleep stages.
So... sleep stage prediction?

Gold-standard labels
A really nice fit for TinyML

A good dataset is data + labels.

Reference device that yields high-quality labels? That's 50% of the work done.
Very common pattern in health

Reference device that can already get the right readings, e.g.:
  * polysomnography for sleep
  * chest-strap for heart rate
  * finger stick for glucose

Want to use ML to make it smaller, cheaper, more power efficient, etc.

Reference device used to derive gold-standard labels.
Using big models to help labeling: YOLOv5
Using big models to help labeling: Whisper

"start": 0.5,
"end": 7.74,
"text": " Hi and welcome to our session here at Spectra 2023 on how to ...",
"confidence": 0.84,
"words": [
  {
    "text": "Hi",
    "start": 0.5,
    "end": 0.66,
    "confidence": 0.87
  },
  {
    "text": "and",
    "start": 0.66,
    "end": 0.8,
    "confidence": 0.65
  },
  {
    "text": "welcome",
    "start": 0.8,
    "end": 0.96,
    "confidence": 0.99
  }
]
Using big models to help labeling: Segment Anything
Putting it into practice
Using cloud-based algorithms to label

Started working with wearable maker in 2022.

Their ML model for sleep stage prediction is licensed and cloud based; with no access to the models.

Customer initial ask:
How could we add features that require processing on-device and in real-time, like 'wake on light sleep'?
Wake on light sleep?

Start wakeup window

Alarm

Wake up here

Awake

REM

Light

Deep

Wake on light sleep?
Step 1: Fetching data from device
Step 2: Setting up pipelines

Fetch data from multiple sources (internal datalake, cloud-labeled data, external reference devices).

Verify that data is correct and complete (all channels available, no truncated data).

Find sleep sessions automatically, categorize and index them (e.g. full nights of sleep + metadata).

Run daily (or on demand).
Catch errors early

Checklists are a very powerful tool.

Multiple devices involved? Find correlation between similar data channels.

Correlation can also be used to time-sync files.
Step 3: Doing some ML on it

Either in Edge Impulse, on SageMaker, in a notebook.

Proof out the use-case, work on features, etc.

Have something that works? Now put it in a pipeline, and the model gets better over time!
Step 4: Getting all this to run on device

PPG

Feature extraction
HR/HRV features, temperature derivative, movement

Machine learning classifier

Temperature

Accelerometer

90% of the work
90% of compute

Sleep analysis
7 h 32 min
Feature extraction

Done by domain experts

Implemented in Matlab or Python

No regard for computational expensiveness in development phase

After completion: hand-over to embedded team (back-and-forth to create features that fit).
EON Compiler + JAX

EON Compiler: Neural network graph to efficient C++ compiler. Hardware-optimizations on Cortex-M, ARC DSPs, ESP32, and a wide variety of NPUs.

JAX: Google project that can compile (a.o.) Python code to an efficient compute graph (with JIT, auto-vectorization).

Together: a way to write code in Python, and compile it down to high-performance, efficient embedded source code.
Python to JAX to EON to C++

Engineer writes code in Python

JAX to compile a compute graph

EON compiles down the graph to C++

Runs efficient on device w/o fw team
Shortening feedback loop

```python
import jax.numpy as jnp

def fn(raw_data, axes_length):
    # data is interleaved so a,b,c,a,b,c so reshape and transpose - this yields one
    # row per axis (e.g. [[a, a], [b, b], [c, c]])
    raw_data = raw_data.reshape((-1, axes_length)).transpose()

    # lowpass filter over the data
    for i in range(0, axes_length):
        raw_data = raw_data.at[i].set(lowpass_filter(raw_data[i], sampling_freq, 3.0, 6))

    # calculate features
    features = []
    features.append(jnp.average(raw_data, axis=-1))
    features.append(jnp.min(raw_data, axis=-1))
    features.append(jnp.max(raw_data, axis=-1))
    features.append(jnp.sqrt(jnp.mean(jnp.square(raw_data), axis=-1)))
    features.append(jnp.std(raw_data, axis=-1))

    # and transpose back and flatten back
    return jnp.array(features).transpose().flatten()
```

Immediately updates, accurate latency/RAM/ROM
(API can also be called directly from Python)
Step 5: Deployment

Planned release in Q3 '23.

Target of 80%+ accuracy.

Deployed as firmware update for existing devices.

All the data and ML infrastructure ready for next steps, model will continue to get better over time.

All infrastructure ready to answer the next questions (e.g. clinical validation of existing algorithms, how good is the PPG=>HR code, what new info can we derive).
Putting it all together

**Dataset creation**
- Aggregate data
- Validate Data
- Prepare Data
- Research data store

**Model development**
- Feature extraction
- Develop and train
- Test and tune

**Automated pipeline**
- Deploy

**Researcher**
- Wearable device
- Reference clinical data

**Edge Impulse**
- New Data
- Active Learning
- Performance monitoring

**Large scale clinical study**
(1000s of subjects)
Recap

It all starts with labels.

Leveraging big models can dramatically reduce time and cost of building datasets.

Have a big model already? Perfect fit for TinyML.
Thank you!

jan@edgeimpulse.com
dgeimpulse.com
This presentation in this publication was presented as a tinyML® EMEA Innovation Forum. The content reflects the opinion of the author(s) and their respective companies. The inclusion of presentations in this publication does not constitute an endorsement by tinyML Foundation or the sponsors.

There is no copyright protection claimed by this publication. However, each presentation is the work of the authors and their respective companies and may contain copyrighted material. As such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

tinyML is a registered trademark of the tinyML Foundation.

www.tinyml.org