tinyML® EMEA
Enabling Ultra-low Power Machine Learning at the Edge

June 26 - 28, 2023

www.tinyML.org
pip install edgeimpulse

A programmatic approach to automate your MLOps Pipelines

Louis Moreau
Developer Relations Lead Engineer, Edge Impulse

June 27, 2023
Introduction

We started our journey by building tools to ease AI for embedded engineers.
Now we also build tools for domain experts to deploy models on edge devices...
profile()
target: cortex-m4f-80mhz
RAM: 39.1 kB
flash: 37.6 kB
latency: 145 ms

Python SDK
deploy()
profile()

target: cortex-m4f-80mhz
RAM: 39.1 kB
flash: 37.6 kB
latency: 145 ms

Python SDK
deploy()
profile()
target: cortex-m4f-80mhz
RAM: 39.1 kB
flash: 37.6 kB
latency: 145 ms
TensorFlow

Keras

Python SDK

EDGE IMPULSE

profile()

target: cortex-m4f-80mhz
RAM: 39.1 kB
flash: 37.6 kB
latency: 145 ms

deploy()
deploy()
profile()
target: cortex-m4f-80mhz
RAM: 39.1 kB
flash: 37.6 kB
latency: 145 ms
Python SDK
Target:

cortex-m4f-80mhz

RAM: 39.1 kB
flash: 37.6 kB
latency: 145 ms

Python SDK

deploy()

profile()
Edge Impulse
Python SDK

How it works?
### Configure the Edge Impulse Python SDK

```python
!pip install edgeimpulse

```{py}
import edgeimpulse as ei

ei.API_KEY = api_key
```

### Profile

```python
profile = ei.model.profile(model="./my_model/",
                          device='cortex-m4f-80mhz')
print(profile.summary())
```

### Deploy

```python
deploy = ei.model.deploy(model="./my_model/",
                         model_output_type=ei.model.output_type.Classification(),
                         deploy_target="zip",
                         output_directory="./")
```
**Configure the Edge Impulse Python SDK**

```python
!pip install edgeimpulse

import edgeimpulse as ei

ei.API_KEY = api_key
```

**Profile**

```python
profile = ei.model.profile(model="./my_model/",
                           device='cortex-m4f-80mhz')

print(profile.summary())
```

**Deploy**

```python
deploy = ei.model.deploy(model="./my_model/",
                         model_output_type=ei.model.output_type.Classification(),
                         deploy_target="zip",
                         output_directory="./")
```
Configure the Edge Impulse Python SDK

```python
!pip install edgeimpulse

import edgeimpulse as ei

ei.API_KEY = api_key
```

Profile

```python
profile = ei.model.profile(model="./my_model/",
device='cortex-m4f-80mhz')
print(profile.summary())
```

Deploy

```python
deploy = ei.model.deploy(model="./my_model/",
model_output_type=ei.model.output_type.Classification(),
deploy_target="zip",
output_directory="./")
```
Example - Profile

```python
[43] profile = ei.model.profile(model="saved_model_float32.zip",
                               device='cortex-m7-216mhz')

print(profile.summary())

Target results for float32:
-------------------------------
{
    "device": "cortex-m7-216mhz",
    "tfliteFileSizeBytes": 863312,
    "isSupportedOnMcu": true,
    "memory": {
        "tflite": {
            "ram": 399257,
            "rom": 927576,
            "arenaSize": 398905
        },
        "eon": {
            "ram": 328776,
            "rom": 882432
        }
    },
    "timePerInferenceMs": 75
}
```
Configure the Edge Impulse Python SDK

```python
# pip install edgeimpulse

import edgeimpulse as ei

ei.API_KEY = api_key

profile = ei.model.profile(model="./my_model/",
device='cortex-m4f-80mhz')
print(profile.summary())
```

Profile

Deploy

```python
deploy = ei.model.deploy(model="./my_model/",
model_output_type=ei.model.output_type.Classification(),
deploy_target="zip",
output_directory="./")
```
Example - Quantize & deploy

Deploy

```python
# Generate the representative data for the quantized model
import glob, cv2
import numpy as np
X_data = []
files = glob.glob("test-set/*.jpg")
for myFile in files:
    image = cv2.imread(myFile)
    resized = cv2.resize(image, (96,96), interpolation = cv2.INTER_AREA)
    X_data.append(resized)

print('X_data shape:', np.array(X_data).shape)

X_data shape: (36, 96, 96, 3)
```
# Set model information, such as your list of labels
deploy_filename = "generated CPP.zip"
labels = ['cotton stem', 'epidermis onion', 'housefly legs', 'unknown', 'wood stem']
model_output_type = ei.model.output_type.Classification(labels=labels)

# Create C++ library with trained model
deploy_bytes = None
try:
    deploy_bytes = ei.model.deploy(model="saved_model.zip",
                                  model_output_type=model_output_type,
                                  deploy_model_type="int8",
                                  representative_data_for_quantization=np.array(X_data, dtype="float32"),
                                  engine="tflite-eon",
                                  deploy_target="zip",
                                  output_directory="./")

except Exception as e:
    print(f"Could not deploy: {e}")

# Write the downloaded raw bytes to a file
if deploy_bytes:
    with open(deploy_filename, 'wb') as f:
        f.write(deploy_bytes)
The Python SDK is built on top of the Edge Impulse Python API bindings, the edgeimpulse-api package.

These are Python wrappers for all of the web API calls available to interact with Edge Impulse projects programmatically (i.e. without needing to use the Studio graphical interface).
Need more?

```python
python -m pip install edgeimpulse-api

from edgeimpulse_api import Configuration, ApiClient, ProjectsApi

# Settings
host = "https://studio.edgeimpulse.com/v1"
api_key = "ei_dae2...

# Create a client object that can connect to our project
config = Configuration(host=host, api_key={"ApiKeyAuthentication": api_key})
client = ApiClient(config)

# Get info about the project
projects = ProjectsApi(client)
project_list = projects.list_projects()  
print(project_list.projects[0])
```
Edge Impulse
Python SDK

Integrations and automation
Integrations & automation

Designed to help ML practitioners with every stage of their workflow

- Integrates easily in **existing ML workflows**
- Unlocks **feature engineering** and fasten feedback loops
- Empowers **model optimization** and deployment to any device
Integrations & automation

CI/CD pipelines

- CI/CD is one of the critical factors for delivering fully tested and up-to-date software or firmware.
- We developed a GitHub Action to easily profile, build and deploy your Edge Impulse model.
Health ML automation example

Automated pipeline

Dataset creation
- Aggregate data
- Validate data
- Prepare data

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

Deploy

Large scale clinical study (1000s of subjects)

Researcher

Health device

Reference clinical data

Edge Impulse

New data
Active learning
Performance Monitoring

Device
Edge Impulse Python SDK

Recap
Convert Python Models into Optimized C++

- Profile on-device performance of any trained model
- Analyze the impact of architectural decisions
- Generate optimized C++ libraries
- Deploy to edge devices
Copyright Notice

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