International Workshop on Embedded Artificial Intelligence Devices, Systems, and Industrial Applications (EAI)

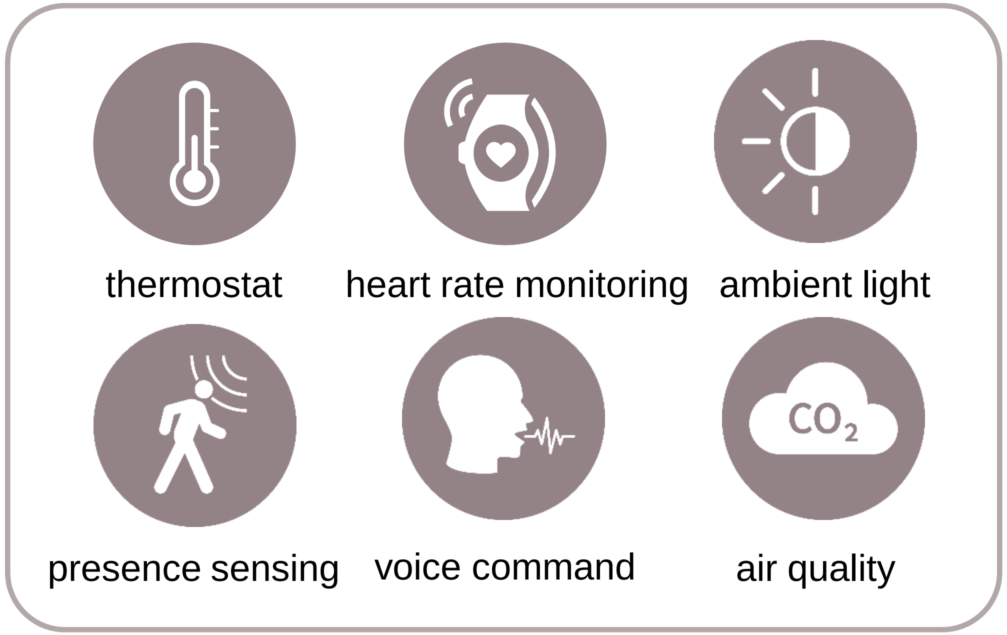
Milan, Italy 19 September 2022

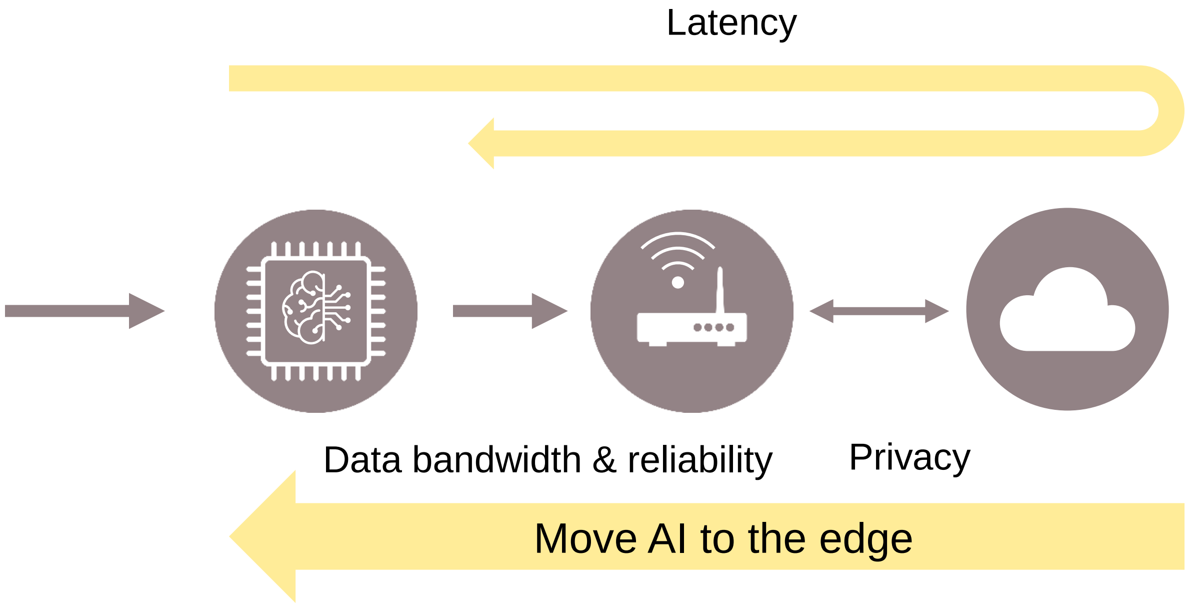
An Embedding Workflow for Tiny Neural Networks on Arm Cortex-M0(+) Cores

Jianyu Zhao, Cecilia Carbonelli, Wolfgang Furtner
Infineon Technologies AG

MILAN 2022
ESSDERC
ESSCIRC

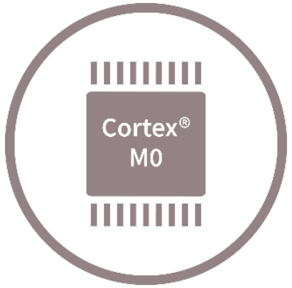
Motivation: AI-enabled IoT Edge Devices



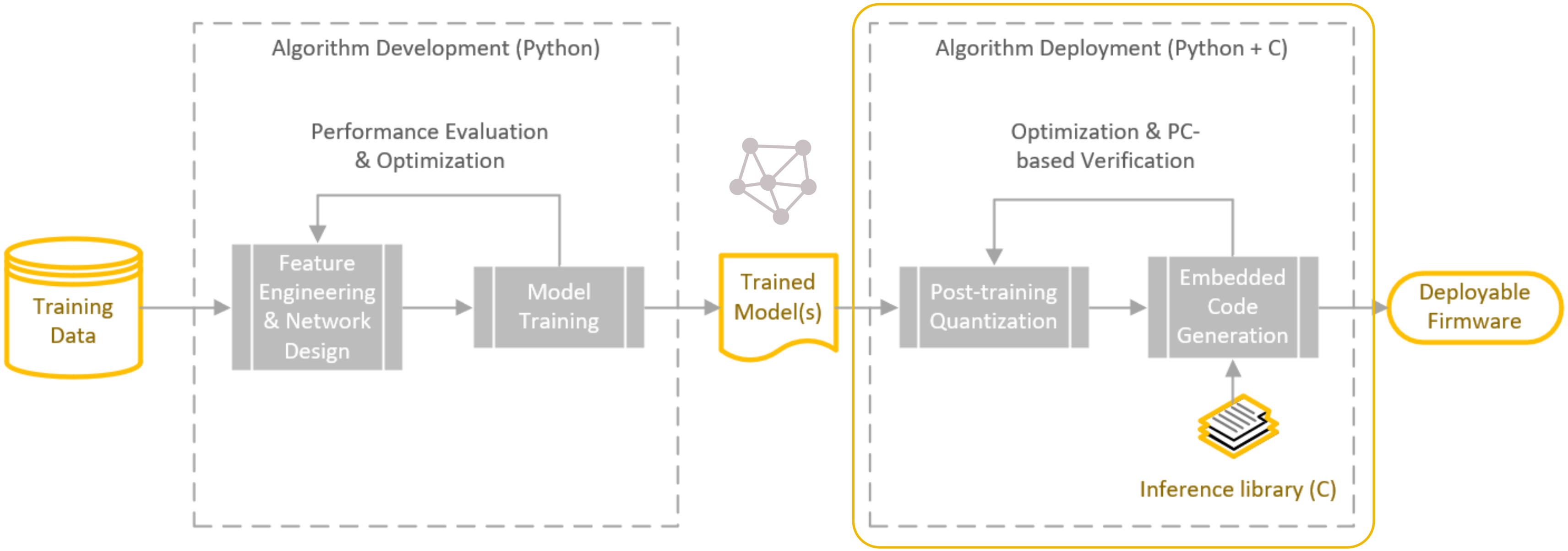


- Always-on and very likely battery-powered: strict power and cost limit, *e.g.*, 1 mW and 1 Euro
- Time series analysis: performance could be improved with fully-connected and/or recurrent networks

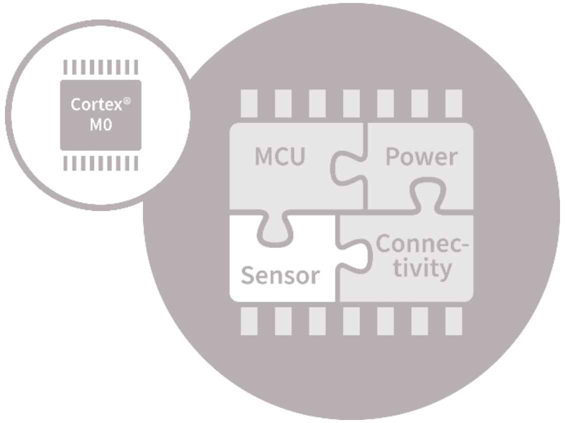
- Arm Cortex-M0(+)
  - widely used for sensor control
  - 32-bit performance at an 8-bit price point
- Challenges
  - very limited computational resource and memory footprint, typically 16 or 32 kB Flash and 4 or 8 kB SRAM
  - no operation system, so the C code needs to run on bare metal



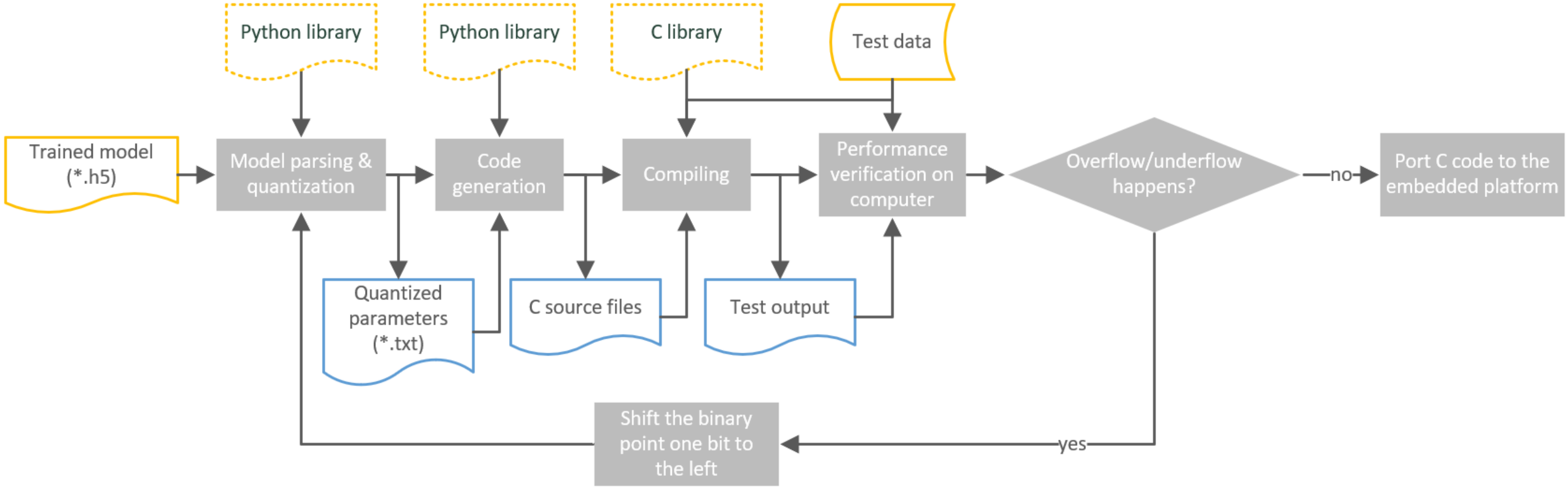
Development and Deployment of Sensor Algorithms



Hardware platforms:



Embedding workflow

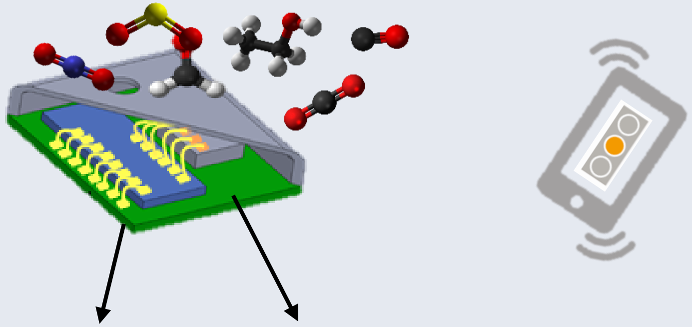


- Reusable Python and C library for network quantization and code generation
- Flexible layer combination, typically `dense` and `GRU` (gated recurrent unit) for time series analysis
- Customizable *bit shift* for each layer

Example: Low-cost Environmental Sensing

Sensor System

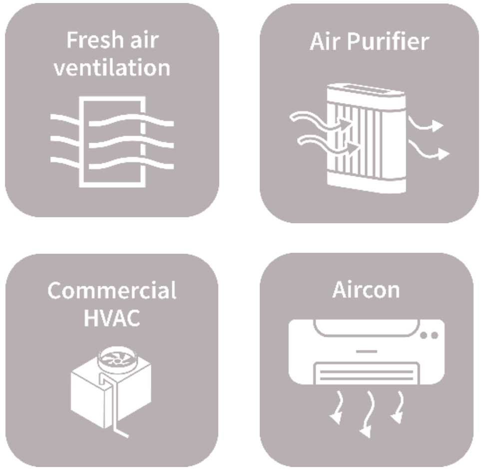
On-sensor signal processing and data analysis on battery-powered devices

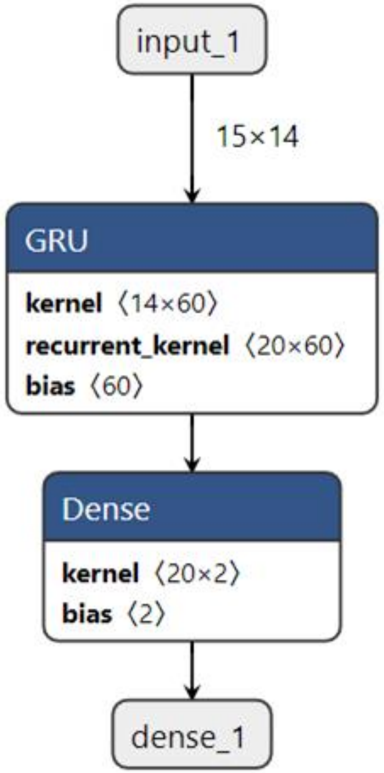


PSoC® Analog Coprocessor (Arm® Cortex®-M0+)

Sensor MEMS

Air quality monitoring Gas Analysis

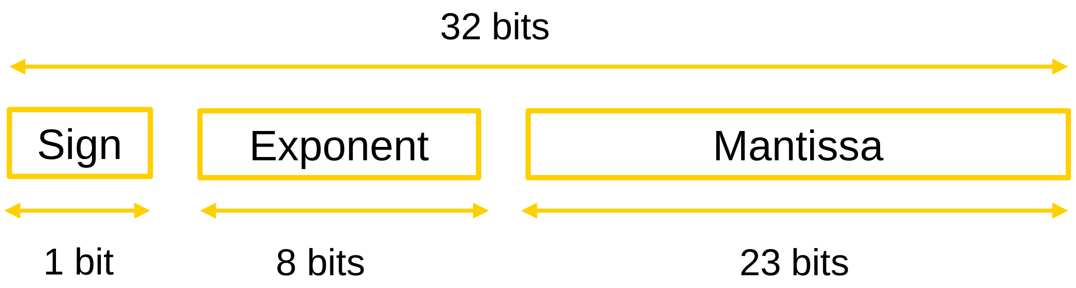




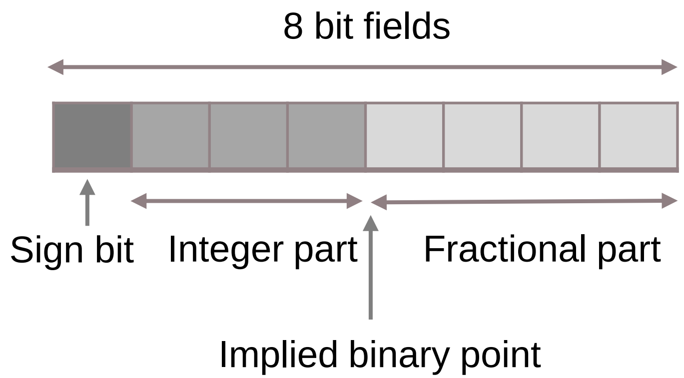
- 14 extracted features
- 15 time steps (5-min time series)
- 20 hidden units
- 2 output concentration values

Example: Model Parsing & Quantization

- Single-precision floating-point representation



- 8-bit fixed-point representation



configuration.txt

```
3
GRU 15 14 20 tanh sigmoid
Dense 20 2 relu
14 2
```

quantized_parameters.txt

```
5
6
--- GRU layer ---
<quantized weights>
<quantized biases>
<quantized weights>
<quantized biases>
<quantized weights>
<quantized biases>
--- Dense layer ---
<quantized weights>
<quantized biases>
```

Example: C library and Code Generation

Files in the C project

```
activation_functions.c
activation_functions.h
fixed_point_operations.c
fixed_point_operations.h
nn_helpers.c
nn_helpers.h
nn_layers.c
nn_layers.h
softmax_functions.c
softmax_functions.h
nn_inference_env.c
nn_inference_env.h
main.c
```

nn_inference_env.c

```
int16_t z1[model_params->layer1->dim_out];

dense_8bit(features, model_params->layer1,
    model_params->n_features, input_shift, z1);
dense_8bit(z1, model_params->layer2,
    model_params->layer1->dim_out,
    model_params->layer1->shift, pred);
```

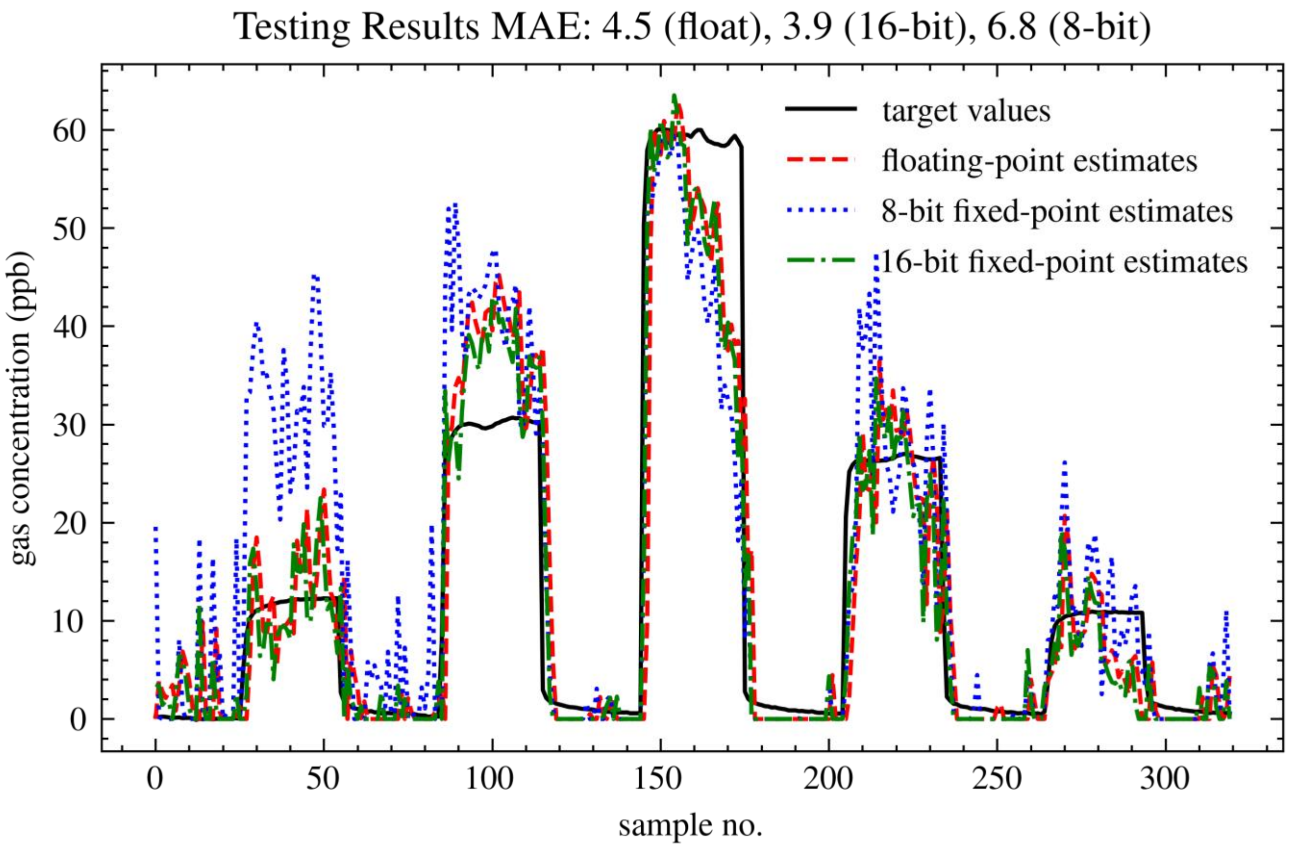
nn_inference_env.h

```
typedef struct {
    const gru_param_8bit_t *layer1;
    const dense_param_8bit_t *layer2;
    const uint8_t n_features;
    const uint8_t n_out;
} model_param_t;

static const model_param_t model_params = {
    &(const gru_param_8bit_t)
    {…}
    &(const dense_param_8bit_t)
    {
        (const int8_t *[])
        {
            (const int8_t []) {16, 6},
            (const int8_t []) {-46, -49},
            (const int8_t []) {68, 79},
             …
            (const int8_t []) {74, 95}
        },
        (const int8_t []) {8, -26},
        ACTIVATION_SIGMOID,
        // dim_out, shift
        2, 6
      },
    14, 2
}
```

nn_layers.h

```
typedef struct {
    const int8_t **weights;
    const int8_t *biases;
    const uint8_t activation;
    const uint8_t dim_out;
    const uint8_t shift;
} dense_param_8bit_t;

typedef struct {
    const int8_t **W_z;
    const int8_t **W_r;
    const int8_t **W_h;
    const int8_t **U_z;
    const int8_t **U_r;
    const int8_t **U_h;
    const int8_t *bias_z;
    const int8_t *bias_r;
    const int8_t *bias_h;
    const uint8_t activation;
    const uint8_t recurrent_activation;
    const uint8_t n_timesteps;
    const uint8_t n_units;
    const uint8_t shift;
}gru_param_8bit_t;
```

Example: Performance Verification



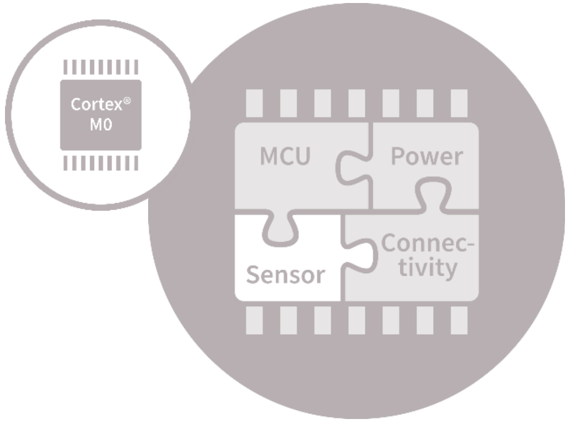
Network implementation	MAE (ppb)	Flash (kB)	SRAM (kB)
Floating-point	4.5	33.5	1.8
16-bit fixed-point	3.9	16.7	1.2
8-bit fixed-point	6.8	8.4	0.8

- 16-bit implementation: 49.3% memory reduction with little change in performance
- 8-bit implementation: 73.9% memory saved with more noisy and less accurate concentration estimates

Summary

- Arm Cortex-M0(+) for AI-enabled IoT edge devices
  - Power-efficient and cost-effective
  - For many applications, more value could be added without additional material cost

- End-to-end workflow for network quantization and code generation
  - Reusable Python and C library
  - Software architecture supporting flexible layer combinations
  - Bit shifts customizable for each layer
- Example project: low-cost environmental sensing
  - Up to 73.9% of the memory footprint was reduced with only a small sacrifice in performance



Part of your life. Part of tomorrow.

Event Organisers

The Key Digital Technologies Joint Undertaking - the Public-Private Partnership for research, development and innovation – funds projects for assuring world-class expertise in these key enabling technologies, essential for Europe's competitive leadership in the era of the digital economy. KDT JU is the successor to the ECSEL JU programme. www.kdt-ju.europa.eu

The AI4DI project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826060. The JU receives support from the European Union’s Horizon 2020 research and innovation programme and the national authorities. www.ai4di.eu

The TEMPO project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826655. The JU receives support from the European Union’s Horizon 2020 research and innovation programme and Belgium, France, Germany, The Netherlands, Switzerland. www.tempo-ecsel.eu

The ANDANTE project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 876925. The JU receives support from the European Union’s Horizon 2020 research and innovation programme and Belgium, France, Germany, The Netherlands, Portugal, Spain, Switzerland. www.andante-ai.eu. ANDANTE has also received funding from the German Federal Ministry of Education and Research(BMBF) under Grant No. 16MEE0116 and 16MEE0117.

Thank You

For your attention

@ Jianyu.Zhao@infineon.com