MILEA – An Approach for Small Scale Applications

Kathrin Gerhard and Eduard Moser

Robert-Bosch GmbH, Germany
An Approach for Small Scale Applications

Introduction

- Who are we?
  - Team within Bosch, who answers the question:
    “How can we bring a ML algorithm simple and efficient on an embedded device?”
  - Therefore, we developed a library called MILEA:
    - MILEA = Machine Intelligence Library for Embedded Applications

- What is presented?
  1. Environment
  2. Motivation
  3. Implementation
  4. Algorithms and Runtime
  5. Key Facts
MILEA – An Approach for Small Scale Applications

1. Environment

- **Different Controllers:**
  - Example: IFX - 32-bit AURIX™ TriCore™ TC27xx, supporting safety requirements
  - TriCore specification: 300 MHz, FLASH 8MB, RAM 1MB (see: www.infineon.com)
  - also, similar ARM cores or big-endian architectures are supported

- **Real-time** operation to control engine feature

- A **small part** of the processes **uses ML-features:**
  - Neural Net: currently about 5
  - Gaussian Process: about 3
  - SVM: about 2
  - Binary Decision Tree, Random Forest: about 3
  - furthermore, in-house data-based algorithms: about 20

![Engine Control Device]

*300 MHz, FLASH 8MB, RAM 1MB*

*more than 2700 processes*

*each ML process uses a small part of the resources:*

*e.g., < 1ms, FLASH 80 KB, RAM 2KB*
2. Motivation (1)

- **Example**: Virtual Pressure Sensor (within vehicles)
  → **Goal**: Detection of the fast-rising pressure signal

Detecting the right criteria with AI model

Sensor signal of Bosch component

Preventing wrong detection by physical model

Why do we need ML on a microcontroller?
MILEA – An Approach for Small Scale Applications

3. Implementation (1): Two-Phase Deployment Process

1. **Software development:**
coding and updates shall be finished during the first part of the product development

2. **Calibration:**
the second flexible way of deployment is performed via calibration:
→ configuration
→ tuning

MILEA includes the whole description of the ML algorithm into a single “screw”

MILEA especially uses the second phase to allow **flexible** and **easy** changes on the software.
MILEA – An Approach for Small Scale Applications

3. Implementation (2)

1. Single “screw”: The description of a ML algorithm is stored in a configuration file (in FlatBuffers format) and configures the ML model
   - individual FlatBuffers schemes per algorithm

2. Each ML algorithm is an interpreter* (<10KB)

3. The user provides the configuration file and call respective interpreter in a real-time process

4. Two-phase validation
   - Interpreter: validation performed on a wide range of configurations as well as requirement based
   - Model: Use-case validation as part of the product development

*currently, all algorithms are based on floating-point implementation
MILEA – An Approach for Small Scale Applications

4. Algorithms and Runtime (1)

- Supported Algorithms:*

- Example: *Sensor Plausibility Check*
  - detects if data has been manipulated
  - Neural Net:
    - 10 inputs
    - 3 LSTM layers (30, 20, 10 units)
    - 5 dense layers

<table>
<thead>
<tr>
<th>Model</th>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>First LSTM Layer</td>
<td>180 μs</td>
</tr>
<tr>
<td>Total</td>
<td>403 μs</td>
</tr>
</tbody>
</table>

*Further algorithms on request*
4. Algorithms and Runtime (2)

- **Support for Hardware Accelerator: DFA**
  - DFA = DataFlow Architecture
  - MILEA compatible to DFA driver
  - Internally, identical parameters

- **DFA speeds up MILEA performance**
  - FlatBuffers flag: SW execution vs. HW acceleration
  - HW up to ~50x faster, same result

<table>
<thead>
<tr>
<th>Model</th>
<th>SW</th>
<th>HW (DFA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dense Layer:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- 40 inputs, 192 neurons</td>
<td>266 µs</td>
<td>7 µs</td>
</tr>
<tr>
<td>- activation function: ReLU</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
MILEA – An Approach for Small Scale Applications

Key Facts

- Only a **small part** of the processes uses **ML-features**

- **Two-phase** deployment process:
  - initial ML model deployed via FlatBuffers
  - FlatBuffers can be updated in calibration phase and allows **flexible** and **easy** changes of the network topology without new software build

- MILEA SW is **ready for series** and already used in several functions

- MILEA enables easy **access to AI methods** from external machine learning frameworks **for embedded use**

- MILEA has **no HW and SW dependencies**

- Extension with additional AI algorithms possible

MILEA is small, efficient, flexible, and easy to use.
THANK YOU

Contact:

Kathrin Gerhard
E-Mail: kathrin.gerhard@de.bosch.com
MILEA – An Approach for Small Scale Applications

2. Motivation (2)

- **Example: Virtual Pressure Sensor** (within vehicles)
  - Solution: AI is the key!

How can we deploy the neural net on the embedded device?

Neural Network:
- 52 inputs
- 2 hidden dense layers
- 50 outputs
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