BIP in Embedded Systems

TinyML: porting Machine Learning to MCU's

Thomas Herpoel

February 15, 2024

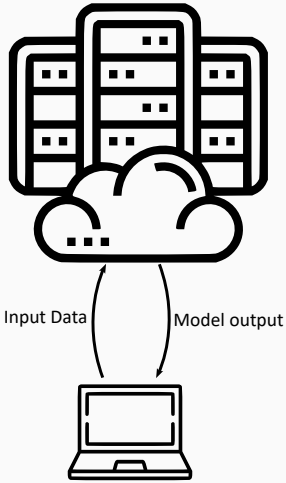
École d'ingénieurs de la HELHa

HELHa
Haute École Louvain en Hainaut

TinyML

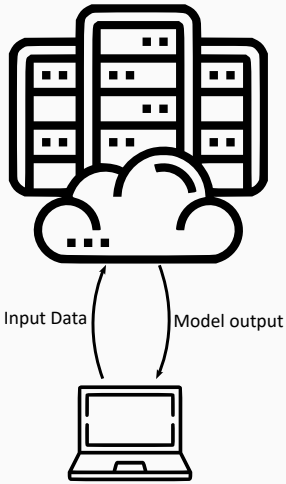
Deployment of machine learning models on resource-constrained devices, such as microcontrollers and embedded systems.

Typical ML approach



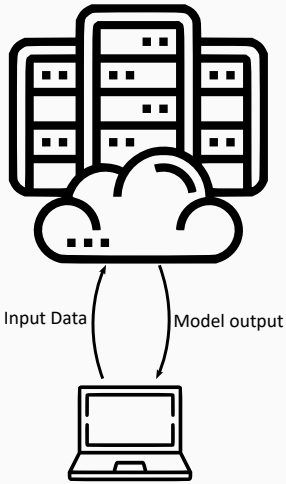
- Model is trained and run in the cloud

Typical ML approach



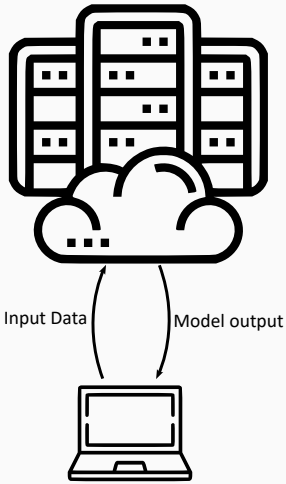
- Model is trained and run in the cloud
- Input data and model outputs are in the wild

Typical ML approach



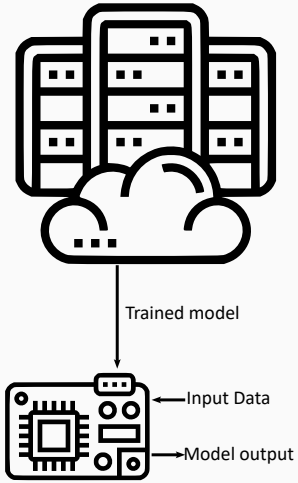
- Model is trained and run in the cloud
- Input data and model outputs are in the wild
- Requires heavy infrastructure

Typical ML approach



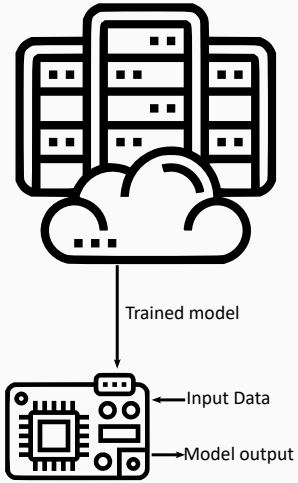
- Model is trained and run in the cloud
- Input data and model outputs are in the wild
- Requires heavy infrastructure
- **Latency!**

ML for embedded systems



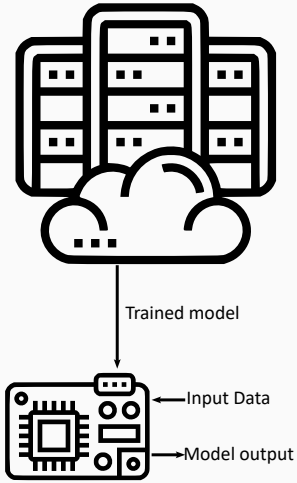
- Model is created and trained on AI specific powerful hardware

ML for embedded systems



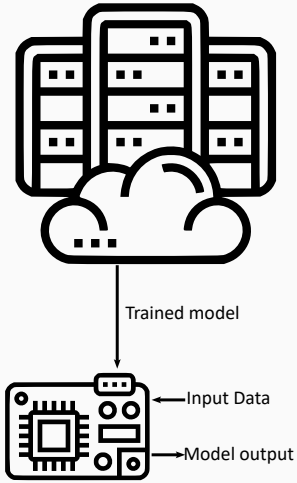
- Model is created and trained on AI specific powerful hardware
- Input data and model outputs stay local

ML for embedded systems



- Model is created and trained on AI specific powerful hardware
- Input data and model outputs stay local
- Low power in use

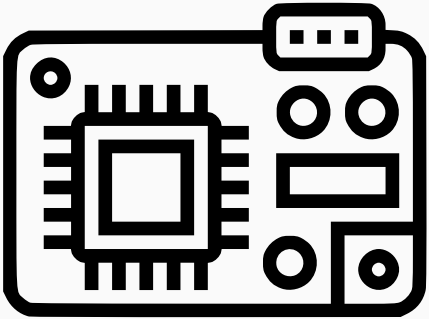
ML for embedded systems



- Model is created and trained on AI specific powerful hardware
- Input data and model outputs stay local
- Low power in use
- **Latency compatible with real-time constraints**

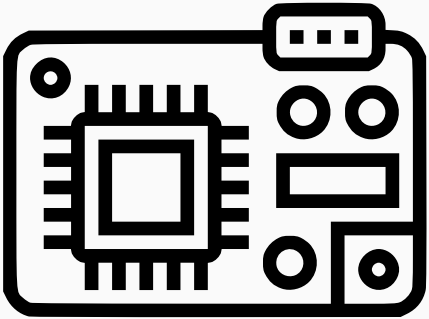
Challenges

- Limited work memory (RAM)
- Limited storage (Flash)



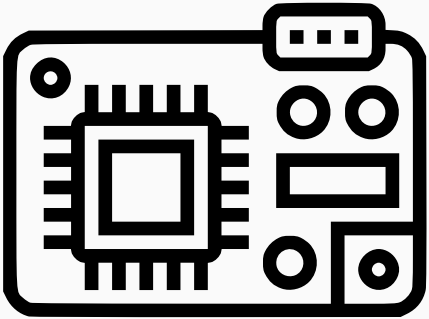
Challenges

- Limited work memory (RAM)
- Limited storage (Flash)
- Limited processing power (CPU)

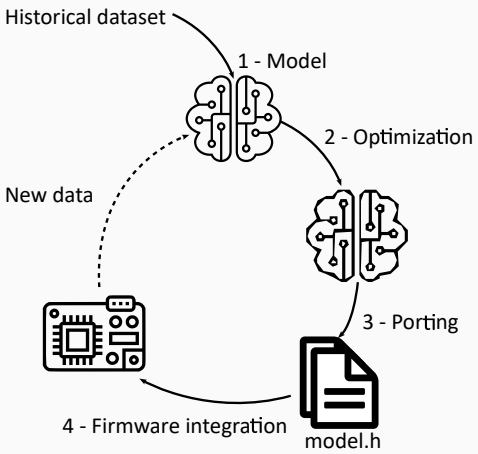


Challenges

- Limited work memory (RAM)
- Limited storage (Flash)
- Limited processing power (CPU)
- Low power availability

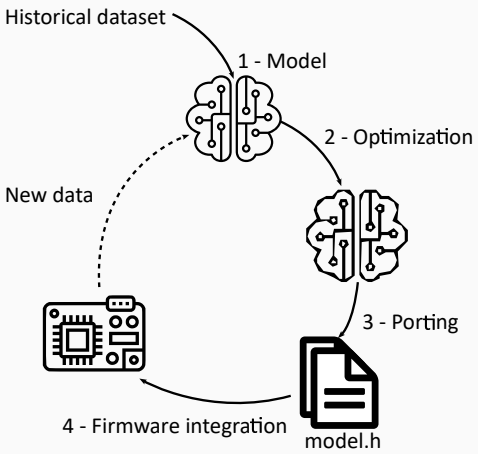


Workflow



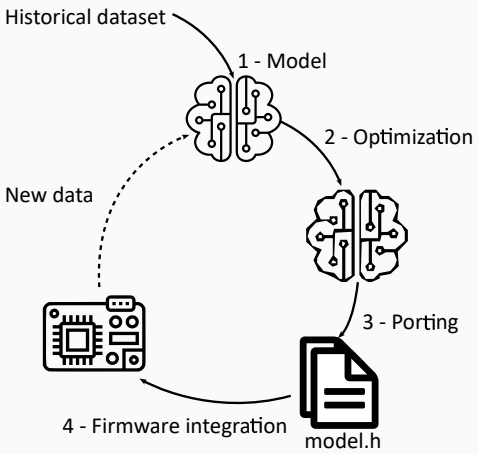
1. **Model definition and training**
   Important to keep the tinyml goal in mind!

Workflow



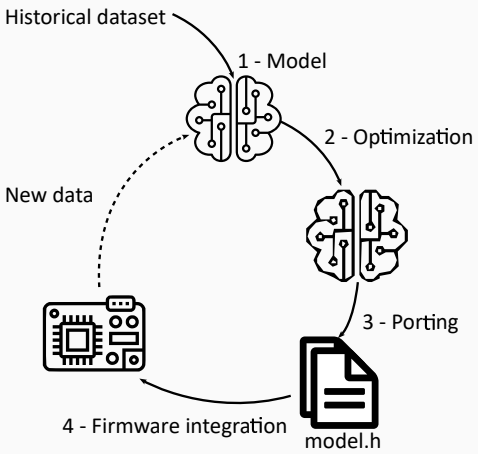
1. **Model definition and training**
   Important to keep the tinyml goal in mind!
2. **Model optimization**
   Quantization, Pruning, Compression

Workflow



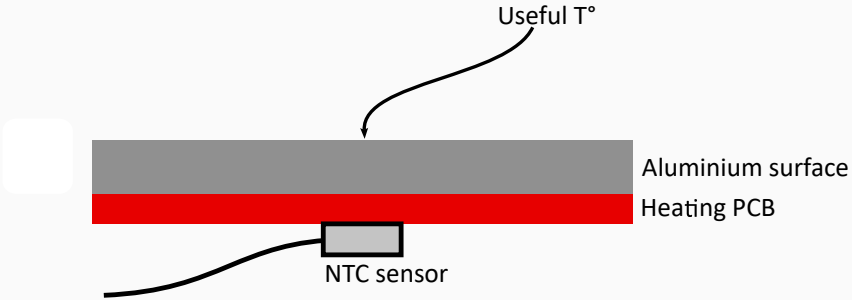
1. **Model definition and training**
   Important to keep the tinyml goal in mind!
2. **Model optimization**
   Quantization, Pruning, Compression
3. **Model conversion for embedded device**
   Output is C code + framework library

Workflow

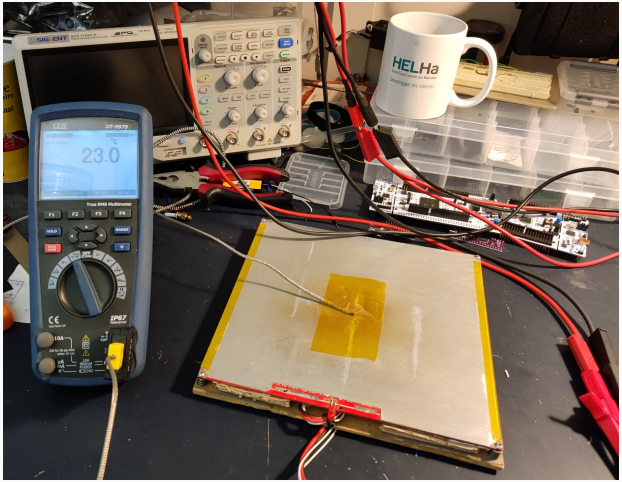


1. **Model definition and training**
   Important to keep the tinyml goal in mind!
2. **Model optimization**
   Quantization, Pruning, Compression
3. **Model conversion for embedded device**
   Output is C code + framework library
4. **Integration into Firmware**

Demo - the problem



Demo - creating dataset



Demo - creating dataset

Temp [C]	NTC [V]
22.6	1.881
23.1	1.87
24.1	1.836
26.4	1.755
28.5	1.671
31	1.573
36.8	1.32
37.2	1.229
43.8	0.986
44.5	0.97
45.2	0.93
52.3	0.747
53.3	0.705
54.4	0.656
62.4	0.492

Demo - test circuit