

# A letter of motivation to the Robot Learning research group

Hiroshi Doyu

December 9, 2022

## Contents

My background	1
My research interests	2
Benefit for the Robot Learning research group	3
Question	3
Reference	3
Appendix	3
Courses . . . . .	3

---

## My background

I have been a Linux Kernel programmer for 20 years, porting Linux Kernel on several new CPUs and implementing low level/embedded drivers (e.g. IOMMU) in the upstream kernel. Those work is all about implementing System Software architecture in traditional Computer Science (CS) (i.e. Operating System) scope, using tools like Assembler, C language and JTAG. Recently I've got interested in a new emerging concept of **Machine Learning compiler** (ML compiler) when I worked at Ericsson Research. Here's our published paper, [A TinyMLaaS Ecosystem for Machine Learning in IoT](#).

Historically ML development was about building up ML models of new algorithms in Python. These pre-trained ML models are running on powerful Cloud GPU in Python execution environment (e.g. Pytorch). The size of ML models has been getting bigger and bigger while ML use cases are expanding from central Cloud computing to Edge computing where computing resources are smaller, diversified

and energy-efficient. There are apparently needs to run ML inferences efficiently on various hardware environments, including the very end of Edge computing (e.g. IoT sensors). ML hardware accelerator at Edge computing needs its tailor made ML compiler to generate optimized executable binary for them, at least, for now. This trend is changing the traditional legacy CS concept (e.g. Compiler) into a new era. For example, there are more needs for domain specific compilers for accelerators used in various specific (tiny) ML use cases. There are 2 aspects of “ML concept” at compilation although they are supplemental.

- Compiling ML models to convert a computational graph at training into the optimized efficient format for inferencing [MLIR](#) and [Apache TVM](#)
- Applying ML methods to optimize compilation process itself (e.g. Reinforcement Learning to find the best outcome). [Apache TVM](#) is doing [this](#).

Right now I’m not employed after Ericsson Research to pursue the possibility to run own business based on our [TinyML](#) solution, under Helsinki university incubation program, [NEXUS](#).

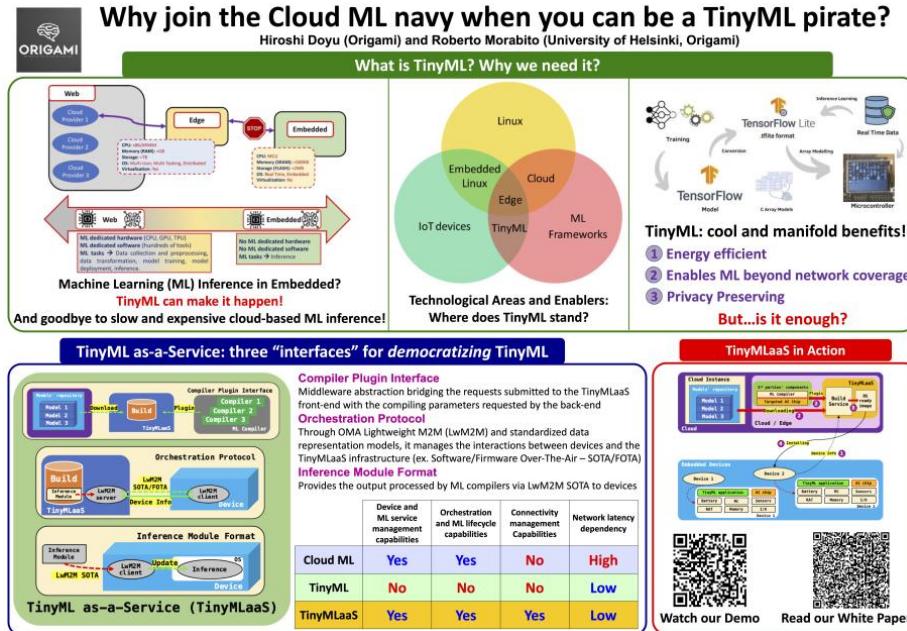


Figure 1: TinyML as-a-Service

## My research interests

I’d like to research further ML compiler area, applying ML algorithms (e.g. RL) to emerging ML compilers, esp. ML HW accelerators. This activity could be

naturally extend to including generation of HW design eventually. I was really entertained by taking both courses, “CS-E4890 - Deep Learning D, Lecture” and “ELEC-E8125 - Reinforcement learning D, Lecture”, at Aalto university.

## Benefit for the Robot Learning research group

Right now I have no idea about the Robot Learning research group. I’d like to learn what kind of synergy we could have.

## Question

I’m not sure that this posted PhD position, [PhD Student Position in Reinforcement Learning and Planning for Hardware Accelerator Design](#), is suitable for my current purpose or not. It would be really nice if we talk about any possible options to research ML compiler area.

## Reference

- [Alexander Ilin](#)

## Appendix

### Courses

I have recently taken the following courses at Aalto university.

- CS-C3240 - Machine Learning D, Lecture
- CS-E407513 - Special Course in Machine Learning, Data Science and Artificial Intelligence D
- CS-E4890 - Deep Learning D, Lecture
- CS-EJ3211 - Machine Learning with Python D
- CS-EJ3311 - Deep Learning with Python D, Lectures
- ELEC-E8125 - Reinforcement learning D, Lecture