# Concurrent Deep Learning Inference with Multiple Models on Single Neural Processing Units

Student

**Michael Schmid**

Introduction: Deep learning applications are slowly but surely making the transition from the stage of fundamental research to its practical applications. Everyday implementations cannot rely on large computer systems with Graphics Processing Units (GPUs), thus Deep Neural Networks (DNNs) are increasingly computed on space-saving and low-cost embedded devices. Embedded devices are subject to various limitations in terms of performance, power consumption and memory requirements.

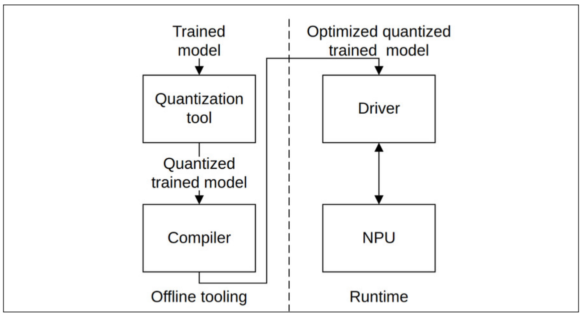
Today, complex systems with advanced sensor fusion are expected to do more than just compute inferences from a single DNN. A system with an embedded device that computes multiple DNNs is demanded. The goal of this project is to investigate the issue of parallel execution of multiple DNNs, using the Arm Ethos-U Neural Processing Unit (NPU).

The Ethos-U NPU is a special class of deep learning processors for which supported DNNs are fused into an optimized operator which is then directly computed on the deep learning Application-Specific Integrated Circuit (ASIC). This process is not preemptive, so it cannot be interrupted. In order not to simply run several DNNs sequentially, a way must be found to split the calculation of one DNN into several parts.

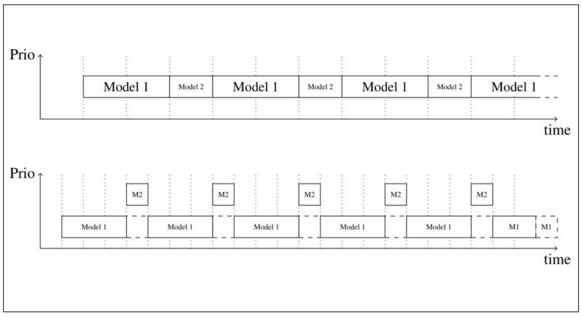
Approach: First, the development environment around the Ethos-U was set up to compute self-trained DNNs on the Corestone-300 simulation platform. Then, different ways to compute a DNN in several parts were shown. The networks were modified in different hierarchy levels of the typical deep learning workflow. After all the functionalities of the modified networks have been tested, a system was built to execute them in parallel with a Real Time Operating System (RTOS).
An appropriate task scheduling is carried out for the execution of DNNs with different inference rates as well as different priorities.

Conclusion: The basic components of such a system are presented and many important tasks could be pointed out. The modification of the DNNs for parallel execution could all be demonstrated successfully and the functionality of the DNNs is still given after the modification.
However, the implementation with the RTOS in combination with the NPU proved to be nontrivial. The correct interaction of the task scheduler and the deep learning implementation of TensorFlow Lite for Microcontrollers could not be applied and guaranteed correctly in all cases. The approach seems to be the right one, but more time needs to be invested to rewrite the driver for the Ethos-U NPU.

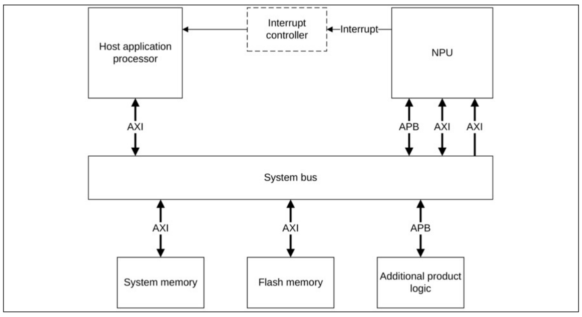
**Tool flow of an embedded Deep Learning system.**
Arm ® Ethos™-U55 NPU Technical reference manual



**Top: Sequential execution of two models. Bottom: Scheduled execution with different priority; Model 2 interrupts Model 1**
Own presentment



**The block diagram of the NPU with the CPU; all function blocks are connected to the same system bus.**
Arm ® Ethos™-U55 NPU Technical reference manual



Advisor
**Prof. Dr. Andreas Breitenmoser**

Subject Area
**Electrical Engineering**

Project Partner
**Sonova AG, Stäfa, Zürich**