Politecnico di Torino
1859

Finanziato dall'Unione europea
NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA

COMPUTER AND CONTROL ENGINEERING

PNRR - Optimizing Compilers for the Deployment of Complex Applications on Heterogeneous Edge Devices

Funded By	MINISTERO DELL'UNIVERSITA' E DELLA RICERCA [P.iva/CF:97429780584] Politecnico di TORINO [P.iva/CF:00518460019]

Supervisor	JAHIER PAGLIARI DANIELE - daniele.jahier@polito.it

Contact	MACII ENRICO - enrico.macii@polito.it

Context of the research activity	How to efficiently execute compute- and data-intensive applications on advanced parallel and heterogeneous hardware is a fundamental problem in today's computing, whose solution is provided by modern compiler infrastructures. This thesis will study and improve such infrastructures, focusing on compilers for deep learning, which represents a key use case for these technologies, due to its potential disruptive impact in many domains (e.g. automotive) and to its peculiar high complexity workload. Progetto finanziato nell'ambito del PNRR M4C2, Investimento 1.4 - Avviso n. 3138 del 16/12/2021 - CN00000023 Sustainable Mobility Center (Centro Nazionale per la Mobilità Sostenibile) - CNMS - CUP E13C22000980001

	The candidate will have the chance to work on two main technologies: Tensor Virtual Machine (TVM) and Multi-Level Intermediate Representation (MLIR). The former is an open-source deep learning compiler and runtime, that aims to accelerate the deployment of machine learning models on a variety of hardware targets, including CPUs, GPUs, and accelerators. MLIR is a novel approach to building reusable compiler infrastructures, whose scope goes beyond machine learning and extends to any kind of domain-specific computing. The candidate will work on applying and extending TVM and MLIR to support the deployment of complex deep learning models on constrained edge devices based on the open-source RISC-V instruction set architecture, ranging from simple microcontroller units (MCUs) to heterogeneous multi-accelerator systems. Detailed objectives:

Objectives	1. To review the existing literature on the execution of deep neural networks (DNNs) on low-power heterogeneous edge devices and to identify potential optimization techniques that can be ported inside TVM or implemented using a dialect optimization step of MLIR. In particular, a set of different optimizations will be developed, which can be either “general-purpose”, i.e., applied to every hardware target, “specific”, i.e., applied to a single target, or “tunable”, i.e., applied to every target but with target-specific parameters. These will include network graphs rewriting optimizations (layer fusion or replacement), individual computational kernel optimizations (loop reordering, tiling, and fusion), and memory management optimizations (DMA, double buffering, etc.). Frameworks for implementing these optimizations, such as polyhedral compilation will be studied and customized for the specific objectives. 2. To study a set of edge-relevant benchmarks and hardware platforms for evaluating the effectiveness of the developed techniques. Examples of benchmarks include the TinyML Perf Suite, which lists four tasks and DNNs and is considered an industrial standard in this field, as well as custom benchmarks related to automotive and smart mobility, energy management, predictive maintenance, etc. The target hardware platforms will include Diana, a System-on-Chip (SoC) developed by KU Leuven that contains a main RISC-V control unit and two DNN hardware accelerators, one 16x16 digital systolic accelerator, and one 1152x512 analog-in-memory computing accelerator; GAP8 and GAP9 by GreenWaves Technologies, characterized by a RISC-V control unit and a small parallel cluster of 8/9 additional RISC-V cores respectively, with a dedicated Level 1 scratchpad memory; GAP9 also includes a programmable DNN-specific digital accelerator. 3. To implement and evaluate the selected optimization techniques on the selected DNNs, benchmarks and low-power hardware platforms. Outline of Work: 1. Familiarization with the target applications and hardware platforms. The three aforementioned target platforms will be configured and tested until the candidate can easily program them. The candidate will also familiarize with the relative Software Development Kits (SDKs) and compilation toolchains. Next, the candidate will study the target applications and DNN models, replicating state-of-the-art results on each of them. 2. Review of the existing literature on the compilation of DNNs for low-power platforms and identification of potential optimization techniques that can be implemented using TVM and MLIR to improve the efficiency of models deployed on the selected hardware targets. This will involve a thorough review of relevant papers, articles, and other sources to identify the most promising optimization techniques. 3. Implementation of the selected optimization techniques in TVM and/or MLIR. 4. Measurement of the performance, energy efficiency or memory occupation improvements obtained through the developed techniques, and analysis of the results to determine the most effective optimization set of steps for each platform. 5. Possible development of new, non-existing, target specific optimizations. The ultimate goal of this research is to improve the efficiency of DNN execution on edge devices, enabling their wider use in a variety of applications. Possible publication venues for this thesis include: - IEEE Transactions on CAD - IEEE Transactions on Computers

	- IEEE Journal on Internet of Things - IEEE Transactions on Emerging Topics in Computing - IEEE Transactions on Parallel and Distributed Systems - IEEE Transactions on Vehicular Technology - ACM Transactions on Embedded Computing Systems - ACM Transactions of Design Automation of Electronic Systems - ACM Transactions on IoT - ACM Transactions on Architecture and Code Optimization

Skills and competencies for the development of the activity	1. Familiarity with programming languages: The project will involve implementing and testing various algorithms and techniques, so experience with programming languages such as Python and C/C++ is needed. 2. Familiarity with embedded systems and computer architectures: Understanding the constraints and limitations of edge devices is important in order to optimize the deep learning models for these platforms. Experience with working on projects involving edge devices would be useful. 3. Familiarity with compilers and compiler optimizations is a nice-to-have, but not a hard requirement, since these concepts will be studied during the first period of the thesis.