Application Specific Domain Co-design Hardware Accelerator IP for Deep Learning Enabled Internet-of-Things

Summary

Currently, DeepLearning (DL) is becoming ubiquitous in everyday life. In particular, Convolutional Neural Networks (CNNs) are already present in many applications related to computer vision ranging from medicine to autonomous driving, neural language processing, and finance. However, these algorithms employ very deep networks to achieve impressive performance, requiring a significant computational power, both during the training and inference time. A single inference of a DL model may require billions of multiply-and-accumulated operations, making the DL extremely compute- and energy-hungry.

The growing desire to adopt these algorithms in an increasing number of applications and the evolution of hardware devices is recently shifting the inference process from the cloud to the edge, i.e., on embedded systems with limited resources. In addition, this paradigm shift can offer the user lower latency and greater privacy. In-device CNN processing presents several benefits.

Since the data does not have to be transmitted to a central processing unit, it remains local, ensuring better user privacy and more deterministic latency. In addition, the absence of transmission reduces power consumption and the need for a network infrastructure capable of handling large amounts of data. At the same time, however, edge computing demands addressing the gap between the computational and memory-hungry CNNs requirements and limited hardware and energy-constrained embedded platforms. Consequently, the need for cost-effective hardware platforms and co-design optimization techniques capable of implementing energy-efficient DL execution arises.

Achieving the objective of enabling minimal resource systems for CNNs algorithms relies on the ability to optimize and co-design accelerator architectures. This dissertation proposes an in-depth analysis and design development for the acceleration of CNN algorithms for edge computing. The analysis covers all critical aspects of acceleration, from algorithmic optimization to architectural exploration and hardware architecture development. As widely discussed, IoT, or edge computing, imposes stringent constraints on area, throughput, and energy efficiency.

In order to optimize a hardware architecture on these three key metrics, the focus must be placed on a specific application. Identifying the specific application domain enables software-hardware co-design techniques to maximize performance.

This thesis is mainly divided into two parts, each proposing a domain-specific solution, the first aimed at training and the second at inference. The first part concerns the optimization of the training and generalization phase in the context of continuous learning (CL). CL requires the ability to learn from a continuous stream of data that may also include new categories of objects. Extreme Learning Machine (ELM), an alternative to Backpropagation, is adapted and applied to CL to solve the challenges presented by the latter, such as latency, accuracy, and resources (memory, algorithmic complexity, etc.). This approach shows encouraging results, for instance, in the case of CaffeNet it out-performed the current state-of-the-art methodologies, reaching an accuracy of 47%.

The second example introduces two implementations of the Serial-MAC-engine (SMAC-engine), a fully-digital hardware accelerator for inference of quantized CNNs suitable for integration in a heterogeneous System-

on-Chip (SoC). With scalable performance, the SMAC engine supports configurable precision for weights (8/6/4 bits) and activations (8/4 bits). Results in 65 nm technology demonstrate that the serial-MAC approach enables the accelerator to achieve a maximum throughput of 14.28 GMAC/s, consuming 0.58 pJ/MAC @ 1.0 V when operating at a precision of 4 bits for weights and 8 bits for activations. The architecture is then further developed to take advantage of the sparsity of activations to improve performance, reaching peaks of 56.88 GMAC/s for Pa=8 and Pw=4 and sparsity 80%.

The structure of the thesis is as follows:

- Chapter one introduces the context and the motivation for this work.
- Chapter two introduces a crucial background part related to architectural exploration for developing Deep Learning enabled hardware accelerators.
- Chapter three presents an alternative to backpropagation that can be used in the case of Continual Learning to create a methodology that allows a DL model to learn continuously.
- Chapter four consists of 2 macro sections. The first describes developing and implementing a hardware architecture for CNN acceleration focused on the area, and power optimization called Serial Multiply and Accumulate engine (SMAC engine), while the second further develop this architecture leveraging activation sparsity.
- Chapter five briefly introduces the challenges and new trends in the verification world.
- Chapter six concludes the dissertation, summarizing the main achievements and contribution of our research.