# Optimization Tools for ConvNets on the Edge

**Valentino Peluso**

**Abstract—**The advancement of low power technologies and design strategies for integrated circuits, together with the improvement of wireless communication systems and infrastructures, enabled a massive deployment of smart IoT sensors able to sense the physical world. Meanwhile, thanks to the recent breakthroughs in Artificial Intelligence (AI), Convolutional Neural Networks (ConvNets) in particular, computers took a further step towards the human intelligence, acquiring skills like autonomous learning and decision making.

The integration of such AI technologies into the end-nodes of the IoT is the premise for a new paradigm—Artificial Intelligence of Things (AIoT)—where sensors will evolve from passive data collectors to active intelligent devices able to infer the meaning of data locally. This shift will thus enable the design of more efficient, scalable, and secure digital ecosystems.

The migration from the cloud to the edge devices poses several issues due to the complexity of modern ConvNet models. The quest for high accuracy has brought the design of ConvNets with billions of hidden parameters and millions of arithmetic operations, thus preventing their deployment on resource-constrained, low-power devices. To tackle this problem, there is a wide consensus that the design of portable ConvNets needs a proper understating of the hardware resources available from the early stage of the training process. Specifically, the optimization of ConvNets should encompass a multi-objective problem formulation, involving extra-functional metrics like memory, energy, and power, besides accuracy. Rather than improving accuracy, the goal is to identify the trade-off frontiers in the design space to pick the best solution meeting the resource constraints. In practice, this can be achieved with algorithmic transformation built upon compression methods that exploit the intrinsic resiliency of neural networks to identify and remove those parts of the model with less contribution to accuracy.

The search for optimality, however, gets challenging due to several reasons. Just like for training, the lack of a closed-form solution to describe the dynamics of the learning flow makes the optimization loop slow and uncertain. Moreover, the high number of dimensions to explore introduced an additional level of complexity overlooked by most of the existing works. A problem formulation neglecting these aspects might result too weak, or unsuited, for real-life applications.

This optimization problem recalls the design of digital integrated circuits (ICs), where multiple conflicting constraints should be addressed, like area, power, and performance. Which of these dimensions gets the highest priority depends on the use-cases, the cost requirements, and in general on the design specifications. For instance, some use-cases require fast processing, whereas those applications with relaxed timing constraints are often limited in area and power consumption. To serve this purpose, the EDA tools for the IC segment have been engineered in a modular way, providing a collection of computer-aided methods with specific goals and formulations. Designers are free to build their own pipeline, integrating the most suited tools depending on their needs.

Applying the same approach to the optimization of ConvNets seems a natural choice and it is exactly the main topic of this dissertation. Indeed, a *one-size-fits-all* solution does not exist due to the diversity of applications and the hardware back-end. Rather, dedicated solutions are needed for the analysis and optimization of memory, energy, and power, and their integration undergoes a vertical implementation, from software to hardware.

Moreover, compression methods originally applied at design-time can be operated at run-time, leading ConvNets to become dynamic algorithms that modulate the resource usage depending on external triggers raised at the application level (e.g. the battery level or the severity of the task). Once again, whereas the adaptive/dynamic control of resources is a well-known standard in hardware design (e.g. dynamic power management), it is a less explored field in the optimization of ConvNets. For this reason, special attention is devoted to this topic, demonstrating that algorithmic knobs for run-time reconfiguration introduce additional degrees of freedom in the optimization space.

This dissertation is organized into three main parts, each of them focusing on a specific design goal. In the first part, it focuses on aggressive memory optimization which is particularly suited for devices with extreme memory constraints (<1MB). It first presents *Prune and Quantize*, a smart heuristic to explore the memory-accuracy space when neural network compression is pushed towards the deep memory region. Then, it introduces *Encoding-Aware Sparse Training*, a novel training technique for sparse ConvNets designed to maximize the compression rate of standard encoding algorithms.

In the second part, the energy optimization problem is addressed elaborating the idea of *Adaptive ConvNets*, a solution that allows ConvNets to trade accuracy for energy at run-time. Two different implementations conceived for software-programmable neural accelerators with mixed-precision arithmetic are discussed and validated.

In the third part, the focus shifts on power optimization, with emphasis on dynamic power and thermal management. A novel power distribution scheme named *FINE-VH* is presented, together with an automatic design methodology and the integration in a standard EDA flow, which enables a more efficient Dynamic-Voltage-Frequency Scaling (DVFS) policy. At last, the efficacy of ConvNets under thermal and accuracy constraints is assessed using DVFS as the main control knob.

Overall, the technical contributions described in this dissertation offer a collection of design&optimization tools for ConvNets to (i) assess at the design-time both functional and extra-functional metrics, (ii) explore different dimensions of the design-space (iii) identify the optimal solution that can meet the hardware requirements.