GMap : An Open-source Efficient Compiler for Mapping any Network onto any Neuromophic Chip

Jimmy Weber, Melika Payvand
Institute of Neuroinformatics, University of Zurich and ETH Zurich, Switzerland.

Summary. Here, we address the problem of mapping networks onto neuromorphic systems by proposing a versatile, easy-to-use and open-source compiler that can efficiently map any arbitrary connectivity matrix to various hardware architectures, while respecting their constraints. The compiler enables the research community to evaluate the compatibility of pre-trained networks with existing hardware or assess the feasibility of implementing a given network onto new hardware architectures. By accommodating diverse hardware configurations, our solution enhances flexibility in deploying neural networks to edge computing.

Event-driven neuromorphic devices offer a promising solution to the stringent power and memory requirements of embedded systems at the edge of the sensors. These systems consist of parallel processing neurons that receive and transmit information in the form of all-or-none events or spikes, similar to the biological systems they are inspired by. To scale these systems, multi-core architectures are used where the number of neurons per core, the number of connections, and connectivity schemes between cores serve as typical design parameters.

Often, the neural networks for certain tasks are trained off-chip, and are then transferred onto the chips for inference. However, given the diverse architectures of neuromorphic systems [1], performing specific hardware-aware training for each system can be highly time and effort consuming. An alternative solution is to directly map the pre-trained network onto the chips, but this introduces a mapping problem. Ideally, we need a method to transfer a hardware-agnostic trained network to a chip while respecting its constraints. Figure 1 illustrates an example of this mapping problem.

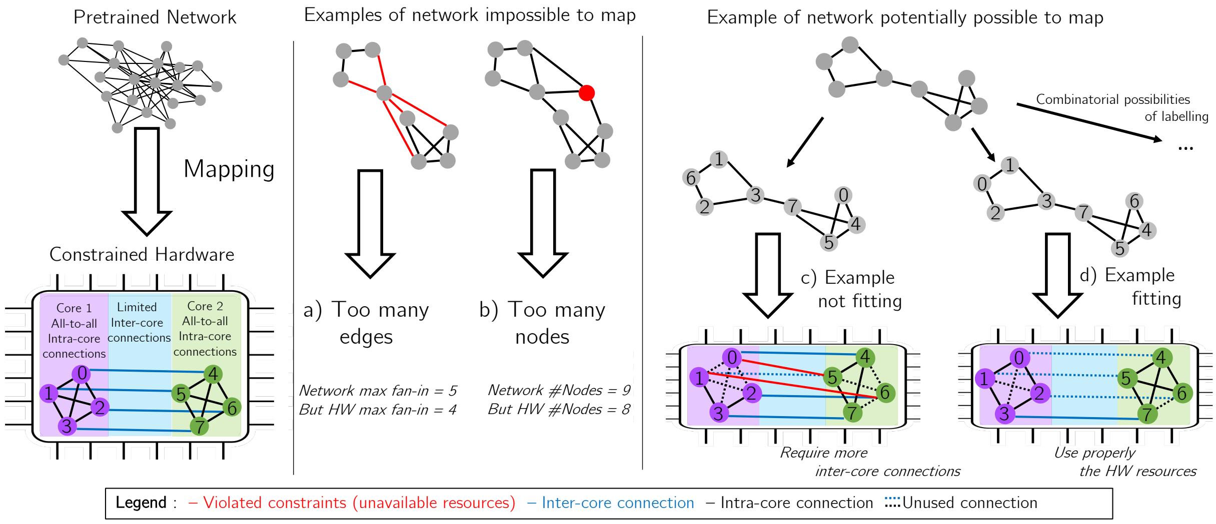

Figure 1: Example of mapping a network onto a constrained hardware. In this example, the hardware has two cores with all-to-all intra-core connections and only sparse inter-core connections. Certain networks are impossible to map onto the hardware (cases a. and b.), while others may be potentially mappable depending on their labeling. Among all the combinatorial possibilities of labeling for the latter, some fit within the hardware constraints (case d.), while others do not (case c.).

Existing solutions for this compilation problem tend to be highly specific to certain hardware or rely on approximations of hardware constraints [4], such as detecting highly-clustered groups of neurons for multi-core chip mapping. These solutions may not be optimal for an arbitrary custom-built hardware configuration. This highlights the need for more generalized mapping techniques that can efficiently map neural networks to a variety of neuromorphic hardware architectures, enabling greater flexibility in deploying neural networks to edge computing devices.

To address the mapping problem in neuromorphic systems, we propose GMap, a versatile, easy-to-use, and open-source compiler capable of mapping any arbitrary connectivity matrix to any arbitrary hardware architecture while respecting its constraints. This enables the research community to easily evaluate the compatibility of pre-trained networks with existing hardware or assess the feasibility of implementing a given network on new hardware architectures. Our algorithm can map networks onto hardware platforms characterized by the number of cores, neurons per core, and maximum fan-in and fan-out. Furthermore, our solution can accommodate more complex routing architectures, as demonstrated by its effectiveness on the mixed-signal DYNAP-SE chip [2].

GMap is an adaptation of the simulated annealing approached presented in [3] to identify the best mapping. The algorithm is a meta-heuristic optimization based on probabilistic approach for approximating the global optimum. It starts with a wide exploration of the search space and gradually transitions to a greedy search as the algorithm progresses towards convergence. It is important to note that it may not always be possible to find a mapping that satisfies all hardware constraints. In such cases, the compiler raises an error, and returns the mapping with the fewest constraint violations.

Given that the algorithm is a heuristic search method, it provides a sub-optimal solution. As shown in Figure 2, to benchmark the accuracy of the solution, we compared it to the actual optimal solution, found using a brute-force technique. However, this is a combinatorial optimization problem, so ground truth comparison was only possible with small networks. In contrast, the proposed algorithm has a complexity of $O(n^2)$, making it efficient for larger networks.

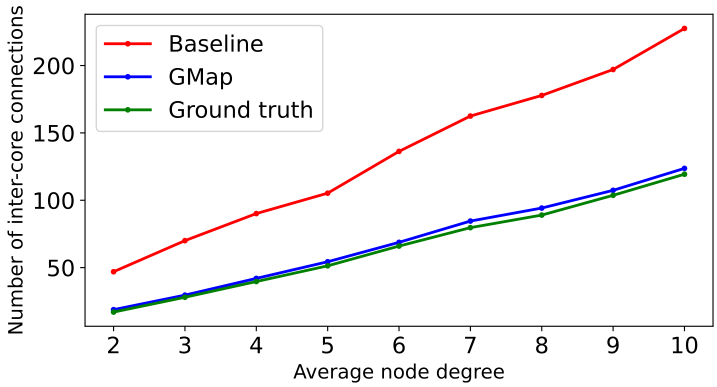

Figure 2: Comparison of the number of constraints violated for the mapping of networks with different average node degree. The show-case is for mapping a 32-node small-world network onto a 4-core neuromorphic hardware. The baseline represent a random mapping and the ground truth is the optimal result obtained with the brute-force approach.

In conclusion, we have developed a general algorithm for mapping any neural networks on any neuromorphic hardware. The input to the algorithm are the architectural parameters of the hardware and the network connectivity, and the output is whether the network maps on the hardware and if yes, a solution for the mapping. This easy-to-use tool for the neuromorphic community is open-source and can be found on GitHub/EIS-Hub/GMap-compiler.

References

[1] Arindam Basu, Lei Deng, Charlotte Frenkel, and Xueyong Zhang. Spiking neural network integrated circuits: A review of trends and future directions. In *2022 IEEE Custom Integrated Circuits Conference (CICC)*, pages 1–8. IEEE, 2022.

[2] Saber Moradi, Ning Qiao, Fabio Stefanini, and Giacomo Indiveri. A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (dynaps). *IEEE transactions on biomedical circuits and systems*, 12(1):106–122, 2017.

[3] Peter JM Van Laarhoven, Emile HL Aarts, Peter JM van Laarhoven, and Emile HL Aarts. *Simulated annealing.* Springer, 1987.

[4] Chao Xiao, Jihua Chen, and Lei Wang. Optimal mapping of spiking neural network to neuromorphic hardware for edge-ai. *Sensors*, 22(19):7248, 2022.