Embedded AI Optimization Approaches and Best Practices

Al-Harith Farhad, Fatos Gashi

Motivation

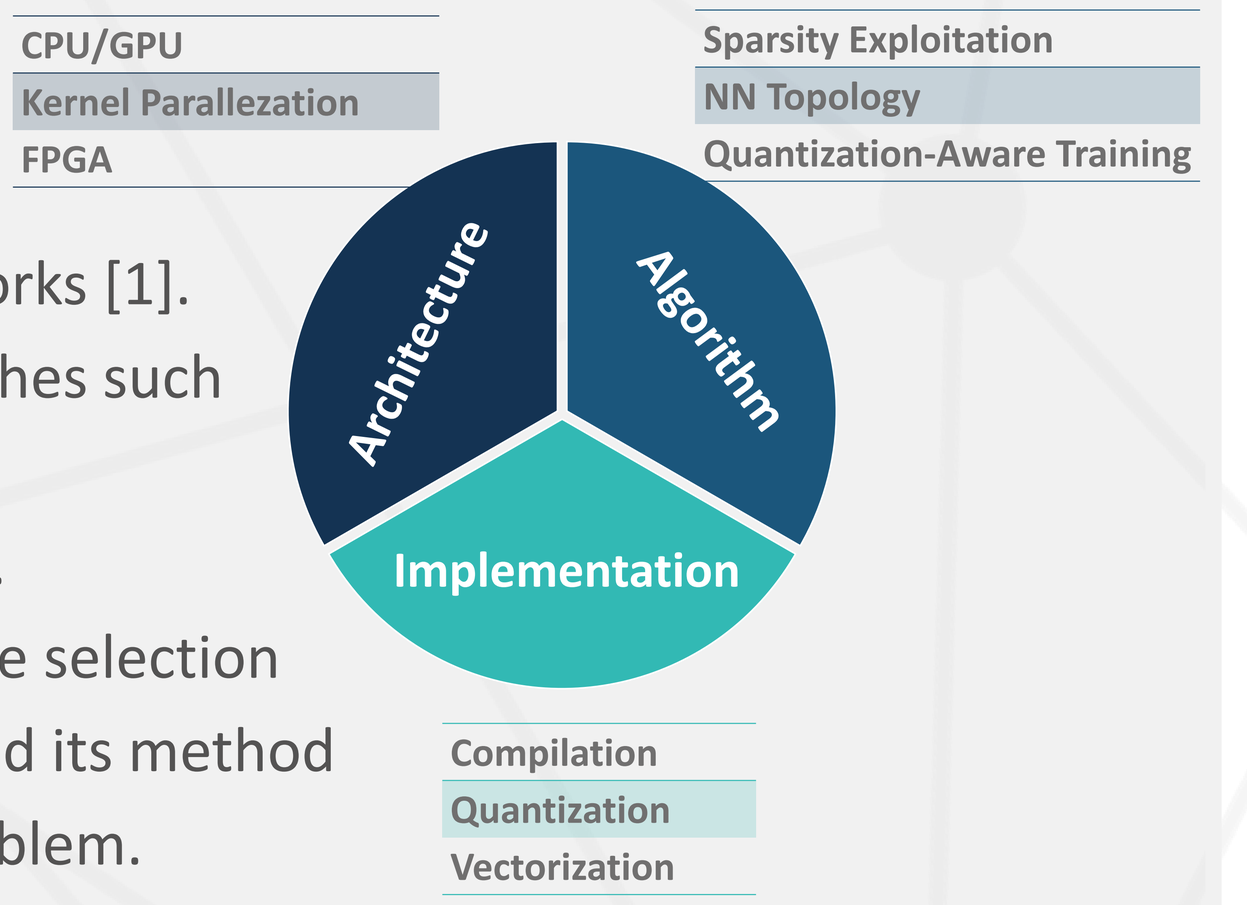
The recent increase in size and complexity of AI model leaves many fields and applications, such as embedded AI, crippled to effectively utilize these models. However, the optimization of these models for edge and small devices can prove essential in using them on-device. This may be achieved with a variety of approaches from choosing the right algorithm and implementation to the hardware architecture used.

Objectives

- Benchmark performance impact of the implementation of various techniques
- Bottleneck analysis of systems
- Identify taxonomy of problems that can be optimized with these techniques
- Map suitable approaches per given the constrains

Application

- Compilation of AI models was shown to achieve better performance than Native implementations in small networks [1].
- Exploration of other hardware approaches such as FPGAs, can also lead to significant Improvement over GPU parallelism [2].
- The trifecta of optimization refers to the selection the optimal architecture, algorithm, and its method of implementation, given a type of problem.



Outlook

- Investigation of optimization methods in comparison to each other
- Resolution of bottlenecks through hardware and software design

References

[1] Thielke, F., & Hasselbring, A. (2019). A JIT compiler for neural network inference. In RoboCup 2019: Robot World Cup XXIII 23 (pp. 448-456). Springer International Publishing.

[2] Rybalkin, V., Ney, J., Tekleyohannes, M. K., & Wehn, N. (2021). When massive GPU parallelism ain't enough: A novel hardware architecture of 2D-LSTM neural network. ACM Transactions on Reconfigurable Technology and Systems (TRETS)

[3] Balasubramanian, K. K., Di Salvo, M., Rocchia, W., Decherchi, S., & Crepaldi, M. (2024). Designing RISC-V Instruction Set Extensions for Artificial Neural Networks: An LLVM Compiler-Driven Perspective.