Resource Scheduling for Real-Time Machine Learning

Suyash Vardhan Singh
University of South Carolina
Columbia, South Carolina, USA
ss121@email.sc.edu

Iftakhar Ahmad
University of South Carolina
Columbia, South Carolina, USA

David Andrews
University of Arkansas
Fayetteville, Arkansas, USA
dandrews@uark.edu

Miaoqing Huang
University of Arkansas
Fayetteville, Arkansas, USA
mqhuang@uark.edu
iahmad@email.sc.edu

Austin R. J. Downey
University of South Carolina
Columbia, South Carolina, USA
austindowney@sc.edu

Jason D. Bakos
University of South Carolina
Columbia, South Carolina, USA
jbakos@cse.sc.edu

Abstract

Data-driven physics models offer the potential for substantially increasing the sample rate for applications in high-rate cyberphysical systems, such as model predictive control, structural health monitoring, and online smart sensing. Making this practical requires new model deployment tools that search for networks with maximum accuracy while meeting both real-time performance and resource constraints. Tools that generate customized architectures for machine learning models, such as HLS4ML and FINN, require manual control over latency and cost trade-offs for each layer. This poster describes a proposed end-to-end framework that combines Bayesian optimization for neural architecture search with Integer Linear Optimization of layer cost-latency trade-off using HLS4ML "reuse factors".

The proposed framework is shown in Fig. 1 and consists of a performance model training phase and two model deployment stages. The performance model training phase generates training data and trains a model to predict the resource cost and latency of an HLS4ML deployment of a given layer and associated reuse factor on a given FPGA. The first model deployment stage takes training, test, and validation data for a physical system–in this case, the Dynamic Reproduction of Projectiles in Ballistic Environments for Advanced Research (DROPBEAR) dataset–and searches the hyperparameter space for Pareto optimal models with respect to latency and workload, as measured by the number of multiplies required for one forward pass. For each of the models generated, a second stage uses the performance model to optimize the reuse factor of each layer to guarantee that the whole model meets the resource constraint while minimizing end-to-end latency.

Table 1 shows the benefit of the reuse factor optimizer that comprises the second stage of the model deployment phase, The results compare the performance of a baseline stochastic search to that of our proposed optimizer for an example model consisting of four convolutional layers, three LSTM layers, and one dense layer. The results show sample stochastic search runs having 1K, 10K, and 1M trials over a total search space of 209 million reuse factor permutations. The stochastic search reaches a point of diminishing returns with latency 205 μs while the optimizer achieves a latency of 190 μs and requires roughly 1000X less search time.

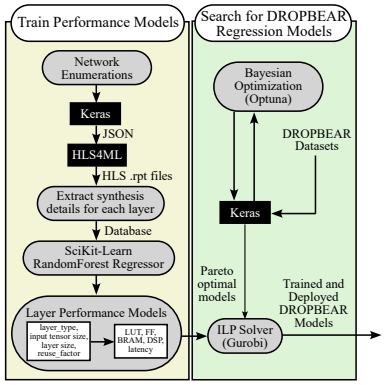

Figure 1: Overview of the tool flow used

Stochastic Search			Proposed ILP Search		ILP vs Stochastic	
Trials	**Search Time (s)**	**Design Latency (μs)**	**Search Time (s)**	**Design Latency (μs)**	**Search Speedup**	**Latency Speedup**
1K	5.03	343.06	4.8	189.84	1.05	1.81
10K	47.67	233.82			9.93	1.23
100K	490.68	227.95			102.23	1.20
1M	4965.65	204.768			1034.51	1.08

Table 1: HLS4ML Deployment Optimizer Versus Stochastic Search

ACM Reference Format:
Suyash Vardhan Singh, Iftakhar Ahmad, David Andrews, Miaoqing Huang, Austin R. J. Downey, and Jason D. Bakos. 2025. Resource Scheduling for Real-Time Machine Learning. In *Proceedings of the 2025 ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA '25), February 27-March 1, 2025, Monterey, CA, USA.* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/3706628.3708848

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1956071.

FPGA '25, February 27-March 1, 2025, Monterey, CA, USA

ACM ISBN 979-8-4007-1396-5/25/02.
https://doi.org/10.1145/3706628.3708848