Predicting Performance of Deep Neural Network Schedules Across Accelerator Designs

Charles Hong (Advisor: Yakun Sophia Shao)
charleshong@berkeley.edu
University of California, Berkeley

1 INTRODUCTION

Specialized deep neural network (DNN) accelerators, such as Google's Tensor Processing Unit (TPU), are becoming more and more common. In order to use these accelerators, compilers must schedule nested loop workloads, such as convolutional neural network layers, by defining their ordering, tiling, and spatial mapping. These schedules are often not optimal [2].

By accurately modeling performance and energy usage across different DNN layers and accelerator architectures, we can more precisely predict DNN performance on hardware configurations that have not yet been deployed. This would allow for rapid software-hardware co-design of machine learning compilers and DNN accelerators, bypassing long simulation times that are often a barrier to optimal hardware design [6]. In addition, such a model could allow for co-scheduling of DNN layers on a partitioned accelerator, providing an additional level of parallelism to spatial accelerators like the TPU. The goal of this work is to understand whether it is possible to build such a generalizable model while maintaining interpretability, as this can help inform future design choices.

2 BACKGROUND AND RELATED WORK

This project builds on CoSA (Constrained Optimization for Spatial Accelerators) [2], which focuses on identifying an optimization strategy for DNN layer scheduling on the Simba accelerator [5]. CoSA uses mixed integer programming techniques to solve for the optimal performance and energy given resource constraints. To evaluate a schedule, CoSA integrates Timeloop [4], which provides microarchitecture- and technology-specific energy models that help estimate performance and energy of the accelerator.

We also address limitations of other prior work. One such work is A Learned Performance Model for Tensor Processing Units [3], which focuses on building an end-to-end performance model for a neural network, targeted specifically to the TPU. A similar work is Mind Mappings [1], which couples neural network-based performance models with gradient-based search to perform DNN layer schedule search. These works motivate our work by indicating that a surrogate machine-learning based model for neural network performance prediction can provide enough accuracy to inform neural network scheduling on accelerators. In contrast to these two works, our performance model seeks to predict parallel performance on arbitrary accelerator configurations, with the goal of enabling faster hardware-software codesign. We also present new insights on what types of models can be used.

A third state-of-the-art work is Apollo [6], which explores the use of various iterative optimization techniques to optimize accelerator architecture. We build upon works like Apollo by taking into account DNN layer scheduling in addition to hardware parameters.

3 OUR APPROACH

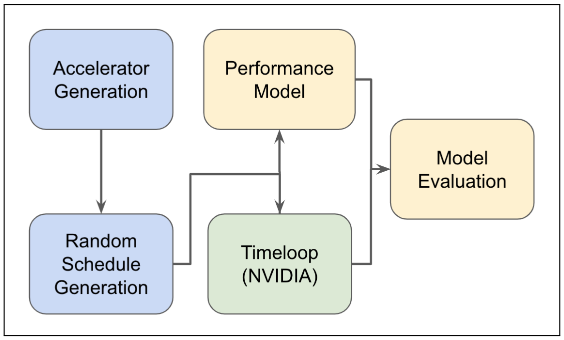

Figure 1: Summary of the approach taken.

A significant portion of this project's contribution is in building infrastructure that allows the fast evaluation of different machine learning models on different accelerator designs and neural network layer schedules. In order to model DNN layer performance on varying hardware, we generate a wide range of accelerator descriptions that can be passed to Timeloop. Parameters include number of parallel processing elements (PEs), as well as the size of on-chip memory, divided amongst the accumulation weight, input, and global buffers. We also use the nested loop representation of a convolutional neural network layer to randomize the following schedule attributes:

- **Loop permutation**, the order of the dimensions in the computational loop nest can be permuted depending on the specific problem dimensions.
- **Memory mapping**, how much of each problem dimension should be mapped to each memory level. This determines buffer usage.
- **Spatial mapping**, how much parallel computation is allotted to a dimension. This determines parallel spatial resource utilization.

We sample schedules across different accelerator configurations to build a machine learning model that will take into account both hardware and software features. So far, around 6 million schedules have been generated.

4 KEY RESULTS AND CONTRIBUTIONS

In this project, we evaluate three main machine learning models on the task of predicting deep neural network layer runtime on new, unseen schedules across different accelerator configurations. After a thorough data analysis, we find that surprisingly, when combined with a compressed, analytically preprocessed feature set, the most

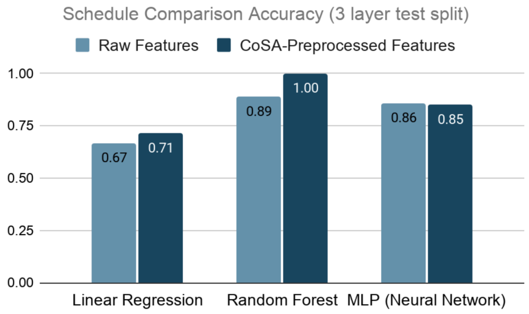

Figure 2: Schedule comparison accuracy for the three model types, trained on 3000 schedules/layer on 5 different accelerators.

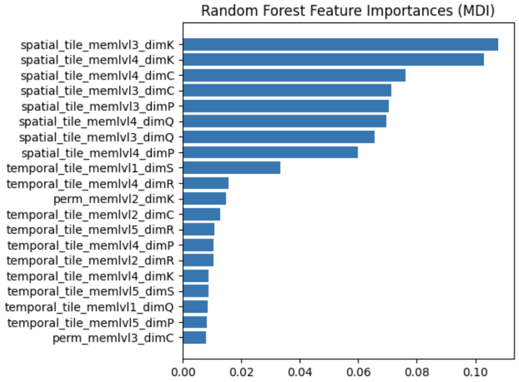

Figure 3: Random forest feature importance of the original 132 input features, using the MDI method.

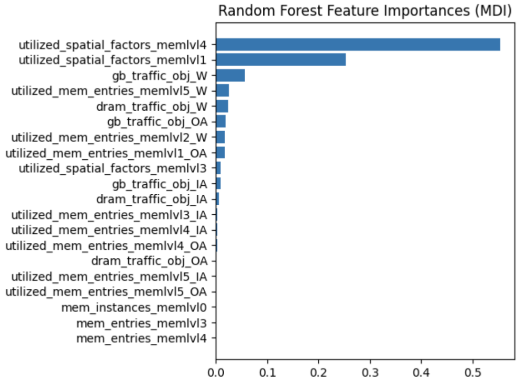

Figure 4: Random forest feature importance of the compressed set of 36 input features, using the MDI method.

Table 1: Models used

Model type	Interpretability	Training time
Linear regression	High	Low
Random forest	Medium	High
Multi-layer perceptron	Low	High

accurate of the three models is the random forest. This compressed feature set both increases accuracy and decreases training time of the random forest model. We also show the usefulness of Mean Decrease in Impurity (MDI) feature importance analysis with random forest models, utilizing the method to identify the most predictive input features. Finally, we show that a random forest model has high potential to predict layer runtime on unseen accelerator designs, and that statistical methods can be used to avoid schedules that break hardware constraints.

5 FUTURE WORK

There are many potential avenues for extending this project. The analysis presented is meant to serve as a building block towards future exploration of hardware-software co-optimization for deep neural network accelerators. Further analysis of the models themselves could provide new insights into the importance of specific accelerator or schedule parameters. With further tuning, the models could be part of generative or reinforcement learning-based methods for generating new accelerator designs or more optimally parallel schedules.

REFERENCES

[1] Kartik Hegde, Po-An Tsai, Sitao Huang, Vikas Chandra, Angshuman Parashar, and Christopher W. Fletcher. 2021. Mind Mappings: Enabling Efficient Algorithm-Accelerator Mapping Space Search. In *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS).*

[2] Qijing Huang, Minwoo Kang, Grace Dinh, Thomas Norell, Aravind Kalaiah, James Demmel, John Wawrzynek, and Yakun Sophia Shao. 2021. CoSA: Scheduling by Constrained Optimization for Spatial Accelerators. In *International Symposium on Computer Architecture (ISCA '21).*

[3] Samuel J. Kaufman, Phitchaya Mangpo Phothilimthana, Yanqi Zhou, Charith Mendis, et al. 2021. A Learned Performance Model for Tensor Processing Units. In *Proceedings of the 4th MLSys Conference.*

[4] Angshuman Parashar, Priyanka Raina, Yakun Sophia Shao, Yu-Hsin Chen, et al. 2019. Timeloop: A Systematic Approach to DNN Accelerator Evaluation. In *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS).* 304–315.

[5] Yakun Sophia Shao, Jason Cemons, Rangharajan Venkatesan, Brian Zimmer, et al. 2019. Simba: Scaling Deep-Learning Inference with Chiplet-Based Architecture. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture.*

[6] Amir Yazdanbakhsh, Christof Angermueller, Berkin Akin, Yanqi Zhou, et al. 2020. Apollo: Transferable Architecture Exploration. In *ML for Systems Workshop at NeurIPS 2020.*