Work-in-Progress: Early Power Estimation of CUDA-based CNNs on GPGPUs

Christopher A. Metz
cmetz@uni-bremen.de
University of Bremen
Bremen, Germany

Mehran Goli*
mehran@uni-bremen.de
University of Bremen
Bremen, Germany

Rolf Drechsler*
drechsler@uni-bremen.de
University of Bremen
Bremen, Germany

ABSTRACT

The increasing application of *Machine Learning* (ML) techniques in the *Internet of Things* (IoT) devices has led designers to leverage ML accelerators like GPGPUs in such devices. However, choosing the most appropriate accelerator for such IoT devices is very challenging as they commonly should adhere to tight constraints e.g., low power consumption, long battery lifetime, and low cost of the final products. As a consequence, designing such application-specific IoT devices becomes a non-trivial and difficult task. In this paper, we present a novel approach to estimate power consumption of CUDA-based *Convolutional Neural Networks* (CNNs) on GPGPUs in the design phase. Our approach is able to provide designers with an early prediction of CNNs power consumption up to an absolute error of less than 2% in comparison to the real hardware execution.

ACM Reference Format:
Christopher A. Metz, Mehran Goli, and Rolf Drechsler. 2021. Work-in-Progress: Early Power Estimation of CUDA-based CNNs on GPGPUs. In *2021 International Conference on Hardware/Software Codesign and System Synthesis (CODES/ISSS '21), October 10–13, 2021, Virtual Event, USA*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3478684.3479255

1 INTRODUCTION

Machine Learning (ML) algorithms have been increasingly used in different application-specific *Internet of Things* (IoT) devices ranging from manufacturing to scientific-, health- and security-related applications [2–4, 6].

One of the major challenges that designers are commonly faced during the design phase of such IoT devices is to choose the right ML accelerator e.g., GPGPUs that adhere to the design constraints such as low power consumption, long battery lifetime, and low cost of the final products [10]. As an example, assume that designers need to design an IoT device where its ML application is performed on a GPGPU (as hardware accelerator). If considering the power consumption and battery lifetime of the IoT device as the design constraints, choosing the most appropriate GPGPU that meets the constraints early in the design phase can significantly avoid costly design loops occur. Moreover, in the case of Cloud-based IoT devices where ML-based data processing performs remotely on Cloud-based ML accelerators (i.e., GPGPUs), choosing an appropriate GPGPU can significantly decrease the renting cost and have a direct impact on the cost of the final product.

One possible solution to approach this issue is using power prediction techniques. Existing methods use so-called performance counters [1, 5, 12] to perform power consumption estimation. Therefore, they rely on the run-time data, meaning the ML model must be run once on the target GPGPU that the performance counter results can be measured. However, this can limit the usage of such methods in the early design phase as the GPGPU must already be selected. Moreover, this can increase the required analysis time.

In this paper, we focus on power consumption estimation of CUDA-based CNNs on GPGPU that is one of the most popular ML algorithms in automated manufacturing [9]. We present a novel approach, enabling designers to predict the power consumption of a given CNN without needing to execute it on any GPGPUs. Unlike the existing methods that use performance counters for their prediction, the proposed approach takes advantage of *Parallel Thread Execution* (PTX) code [11] (which is generated at compile time) and GPGPU architectural information. This empowers designers to choose for a given CNN, the most efficient GPGPU in terms of power consumption among the existing models at compile time. The initial experimental results illustrate the effectiveness of our approach in estimating the power consumption of CNNs on GPGPUs where up to an absolute error of less than 2% in comparison to the real hardware execution is achieved.

2 POWER PREDICTION METHODOLOGY

The proposed methodology is illustrated in Fig. 1 which consists of two main stages: 1) Information extraction, and 2) Power predictive model generation.

In the first stage, we compare different Nvidia GPGPUs (different series and types) in terms of architectural information (e.g., CUDA Cores, Memory or L2 Cache) and how a given CNN can load their different components. By this, those GPGPUs' attributes and components which have an impact on performing CNN models are extracted. Next, we analyze the PTX representation of each CNN and extract the

*Also with Cyber-Physical Systems DFKI.

CODES/ISSS '21, October 10–13, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-9076-7/21/10…$15.00
https://doi.org/10.1145/3478684.3479255

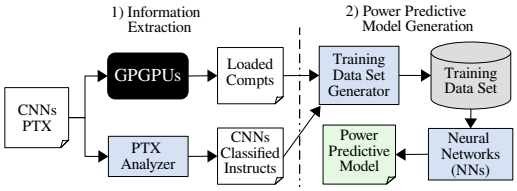

Figure 1: Methodology overview.

instructions that must be loaded into GPGPU's components to run the CNN. These instructions are classified based on their types into different classes.

In the second stage, we build a training data set where the classified extracted CNN instructions and the GPGPU components (that have an impact on performing CNN models) are considered as inputs and the amount of power consumption for each CNN running on the GPGPU as output. The amount of power consumption for each CNN is measured on real GPGPUs. For the training phase, we use four different GPGPUs and 18 different CNNs. In the next step, a neural network is trained with the collected data to create the power consumption predictive model. It takes as inputs the GPGPU architecture and the *PTX Instruction Classes*. The output is the power consumption of the input CNNs executed on GPGPUs. Once the predictive model is generated, the trained neural network can be used to estimate the power consumption of a given CNN on different GPGPU architectures.

3 EXPERIMENTAL RESULTS

Our experimental results demonstrate that the power estimation based on the PTX and GPGPU architecture is promising. Fig. 2 shows the results of our prediction model on estimating the power consumption for ResNet [7] and Densenet [8] variations. On average, our prediction model achieves an *Absolute Error* (AE) of 8.3%. The best-case prediction result belongs to ResNet152 with an AE 0.73%. The worst-case prediction is related to DenseNet201 with an AE of 15.75%. This is due to the fact, that more ResNet variations are included in the training data than e.g., Densnet. By increasing the variation of CNNs in the training data, the prediction can be further improved which is a part of our future works.

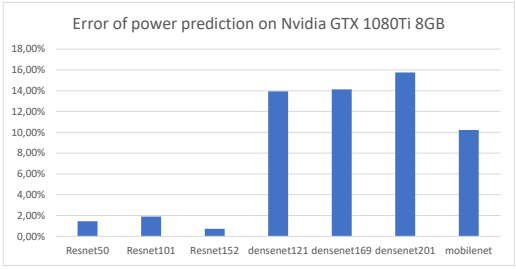

Figure 2: Absolute error of power estimation of different CNNs for Nvidia RTX 1080Ti.

4 CONCLUSION AND FUTURE WORK

In this paper, we presented an early power estimation for CUDA-based CNNs on GPGPUs. We showed how the power consumption of a given CNN model on GPGPUs can be estimated by analyzing its PTX code and the GPGPUs' architectural information. Initial experimental results sound promising.

For future work, we plan to provide a compiler plugin, enabling designers to obtain a power profile based on our prediction model during compilation time. This can significantly help designers to build better ML systems early in the design process. Moreover, we work on preparing more standard CNNs and variations of well-known CNNs to expand our training data set. A more heterogeneous training data will improve our model. Finally, we plan to extend our prediction model to support any kind of CUDA-based applications as well as other non-functional design aspects such as performance estimation.

ACKNOWLEDGMENTS

This work was supported in part by the German Federal Ministry of Education and Research (BMBF) within the project VerSys under contract no. 01IW19001, by the Data Science Center of the University of Bremen (DSC@UB), and by the University of Bremen's graduate school SyDe, funded by the German Excellence Initiative.

REFERENCES

[1] Jianmin Chen, Bin Li, Ying Zhang, Lu Peng, and Jih-kwon Peir. 2011. Statistical GPU power analysis using tree-based methods. In *IGCC*. 1–6.
[2] Zhongke Gao, Yanli Li, Yuxuan Yang, Na Dong, Xiong Yang, and Celso Grebogi. 2020. A Coincidence-Filtering-Based Approach for CNNs in EEG-Based Recognition. *IEEE Transactions on Industrial Informatics* 16, 11 (2020), 7159–7167.
[3] Mehran Goli and Rolf Drechsler. 2020. PREASC: Automatic Portion Resilience Evaluation for Approximating SystemC-Based Designs Using Regression Analysis Techniques. *ACM Trans. Des. Autom. Electron. Syst.* 25, 5, Article 40 (2020), 28 pages.
[4] Mehran Goli, Jannis Stoppe, and Rolf Drechsler. 2018. Resilience Evaluation for Approximating SystemC Designs Using Machine Learning Techniques. In *RSP*. 97–103.
[5] João Guerreiro, Aleksandar Ilic, Nuno Roma, and Pedro Tomás. 2019. Modeling and Decoupling the GPU Power Consumption for Cross-Domain DVFS. *TPDS* 30, 11 (2019), 2494–2506.
[6] Kaiyuan Guo, Lingzhi Sui, Jiantao Qiu, Jincheng Yu, Junbin Wang, Song Yao, Song Han, Yu Wang, and Huazhong Yang. 2018. Angel-Eye: A Complete Design Flow for Mapping CNN Onto Embedded FPGA. *TCAD* 37, 1 (2018), 35–47.
[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition.
[8] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. 2017. Densely Connected Convolutional Networks. In *CVPR*. 2261–2269.
[9] Manuella Kadar and Daniela Onita. 2019. A deep CNN for Image Analytics in Automated Manufacturing Process Control. In *ECAI*. 1–5.
[10] Christopher Metz, Mehran Goli, and R. Drechsler. 2021. Pick the Right Edge Device: Towards Power and Performance Estimation of CUDA-based CNNs on GPGPUs. *ArXiv* abs/2102.02645 (2021).
[11] Nvidia. 2021. Parallel Thread Execution ISA application guide. https://docs.nvidia.com/cuda/parallel-thread-execution/index.html. Accessed: 2021-05-26.
[12] Shuaiwen Song, Chunyi Su, Barry Rountree, and Kirk W. Cameron. 2013. A Simplified and Accurate Model of Power-Performance Efficiency on Emergent GPU Architectures. In *SPDP*. 673–686.