Auto-tuning of Hyperparameters by Parallel Search Using Xcrypt

Tatsuro Hanyu† Masatoshi Kawai‡ Takahiro Katagiri‡ Tasuku Hiraishi†3 Tetsuya Hoshino‡ Toru Nagai‡

Graduate School of Informatics, Nagoya University† Information Technology Center, Nagoya University‡
Department of Information and Computer Science, Faculty of Engineering, Kyoto Tachibana University†3

1 Introduction

Optimizing hyperparameters is a crucial task for ensuring both the quality of responses and overall performance. In the context of Convolutional Neural Networks (CNN), fine-tuning hyperparameters such as batch sizes plays a pivotal role in achieving high-quality AI models. However, the associated tuning costs are significant, considering both the time investment and human resources required. On the other hand, parallel processing is indispensable for maximizing the utilization of extensive computing resources. There are two primary approaches to achieve this parallelization: program-level (employing multiple processes or threads) and job-level (multiple executions of the same program with different configurations). Job-level parallelization proves particularly advantageous for thorough parameter exploration and performance evaluation. It enhances overall efficiency by executing numerous identical computational tasks with varying settings. In supercomputer systems, computational resources are managed through a batch scheduler. Users can create job scripts and submit compute tasks to a queue, specifying the necessary resources. These scripts outline task types, required computing resources, execution times, etc. The scheduler then executes the jobs based on resource availability. However, variations in scheduler interfaces and job script syntax across different supercomputer systems can pose challenges, making it challenging to reuse scripts across multiple environments. Addressing this challenge, a Perl-based scripting language named Xcrypt [1][2] has been developed for job-level parallel programming. In our research, we aimed to create an automated system for hyperparameter search using Xcrypt, specifically focusing on auto-tuning (AT) functions [3].

2 Xcrypt

Xcrypt stands as a scripting language designed for orchestrating processes that execute numerous jobs, whether sequentially or in parallel, on a supercomputer. Built upon Perl, it offers a user-friendly approach for simple processes, like parameter sweeps, without necessitating in-depth Perl knowledge. What sets Xcrypt apart is its capability to execute a single script seamlessly across various environments, eliminating concerns about divergent job submission interfaces across supercomputers. The framework supports various extensions, and users have the option to develop custom extensions, such as limiting the number of simultaneous job submissions and specifying job dependencies declaratively.

3 Preliminary Result

Table 1 illustrates the results of auto-tuning (AT) for batch size optimization in ResNet50 using Xcrypt. The exploration spans a batch size range from 1 to 700, resulting in a comprehensive search encompassing a total of 700 configurations.

Table 1: Optimization Result with Xcrypt.

	Batch size	loss	Execution time [s.]
Best	212	8.4289	71.2

4 Conclusion

In this study, Xcrypt has been integrated for hyperparameter optimization specifically applied to ResNet50. In future work, we plan to delve into the exploration of a sophisticated search methodology using Xcrypt.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Number JP19H05662.

REFERENCES

[1] Tasuku Hiraishi, Tatsuya Abe, Takeshi Iwashita, Hiroshi Nakashima. Xcrypt: A Perl Extension for Job Level Parallel Programming. Second International Workshop on High-performance Infrastructure for Scalable Tools WHIST, 2012

[2] Masaru Ueno, Tasuku Hiraishi, Motoharu Hibino, Takeshi Iwashita, Hiroshi Nakashima. Multilingualization Based on RPC for Job-Level Parallel Script Language, Xcrypt. IPSJ Transaction on Programming, Vol. 6, No. 2, pp. 55-68, 2013

[3] Takahiro Katagiri, Daisuke Takahashi, Japanese Autotuning Research: Autotuning Languages and FFT, Proc. of the IEEE, Vol. 106, Issue 11, pp. 2056 – 2067, 2018