論文 / 著書情報
Article / Book Information

題目(和文)	アーキテクチャ－アルゴリズム協創による小型・高効率ニューラルネットワークアクセラレータの研究
Title(English)	A Study of Highly Compact and Efficient Neural Network Accelerators through Architecture/Algorithm Co-Exploration
著者(和文)	安藤洸太
Author(English)	Kota Ando
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第11941号, 授与年月日:2021年3月26日, 学位の種別:課程博士, 審査員:本村 真人,髙橋 篤司,劉 載勲,中原 啓貴,原 祐子,佐々木 広,高前田 伸也
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第11941号, Conferred date:2021/3/26, Degree Type:Course doctor, Examiner:,,,,,,
学位種別(和文)	博士論文
Category(English)	Doctoral Thesis
種別(和文)	論文要旨
Type(English)	Summary

（博士課程）
Doctoral Program

論 文 要 旨

THESIS SUMMARY

系・コース： Department of, Graduate major in	情報通信 系 情報通信 コース	申請学位（専攻分野）： Academic Degree Requested	博士 Doctor of （ 工学 ）
学生氏名： Student's Name	安藤 洸太	指導教員（主）： Academic Supervisor(main)	本村 真人
		指導教員（副）： Academic Supervisor(sub)	劉 載勲

要旨（英文 800 語程度）
Thesis Summary（approx.800 English Words ）

Deep neural networks have enabled numerous AI applications in various fields, obtaining higher accuracy in many tasks than conventional heuristic approaches. This great success of deep neural networks was attained by advances in hardware technology typified by GPUs and the invention of model structure and training algorithms.

Dedicated hardware accelerators for neural networks on FPGAs and ASICs have been proposed to compute a neural network model in acceptable latency and throughput. On the other hand, the algorithms are continuously evolving, getting more complex and massive; hardware-oriented algorithms to reduce the computational complexity or amount have also been proposed. We expect that hardware-software coevolution drives further development of AI-enabled applications and establishes the next generation of information processing with AI-native computing power.

This thesis introduces programmable neural network processors for edge-side inference and accurate quantized neural network algorithms for embedded systems, with the concepts appropriately verified by FPGA and ASIC prototyping. Through this, we show a practical example and of hardware-software co-optimization, which proves its effectiveness by quantitative evaluation.

First, we prototype a reconfigurable parallel processor for convolutional neural networks. A convolutional neural network consists of two types of computational layers, convolution (Conv) and fully-connected (FC) layers, which have different computational and data structures. We extract a one-to-all parallel multiply-accumulation (MAC) as the common primitive operation in both Conv and FC layers. We construct an ideal parallel computational array with shared and individual buses to conduct this primitive operation. After that, we make this architecture feasible by relaxing internal data rates with an in-array data sharing mechanism and multithreaded accumulators. We discuss the requirements for hardware and model structures for better efficiency through the architecture exploration using this architecture.

We then propose a near-memory processor, named BRein Memory, which improves neural network processing efficiency by eliminating external data movements. It is known that the energy and latency of external memory access set limitations on the performance and efficiency of neural network processing. We attempt to omit the external memory throughout the entire processing of a neural network model. We adopt the binary neural network algorithm, where all the activation and weights are restricted to be −1/+1. It allows the multiplication to be calculated as bitwise XNOR. Thanks to this light-weight arithmetic, we integrate thin processing units between two SRAM macros to close the parallel computation and data movement in it. The computational units can be cascaded to compose a pipeline by matching the symmetric parallelisms in a neural network layer with the SRAM data access pattern, where no scratchpad memory is required. We fabricated a prototype LSI of this architecture with six processing blocks, which can house a 13-layer neural network model at most, achieving 2.3 TOPS/W energy efficiency, the best figure in neural network processor ASICs at the time published. The concept of highly efficient near-memory processing with a quantization algorithm has been proven by prototyping and evaluating this architecture.

Quantized neural networks reduce the computational complexity and memory footprint. However, the accuracy drop is unavoidable, and this could be a fatal problem in some situations. Here, we discuss an accurate yet efficient quantization algorithm that can be used in compact hardware implementation. A neural network processor is a digital signal processor when we see the computation carried in the processing units. The knowledge in signal processing gives us a hint for improving the accuracy of quantized neural networks on compact hardware. Dithering is a technique for low-precision quantization that reduces the quantization errors by representing the source data in a stochastically and spatially distributed manner. Its simplest algorithm is error diffusion, which can be computed by adding each pixel's quantization error to the neighbor. A processing unit in a neural network processor has an adder, which can be appropriated for quantization error accumulation. Based on this idea, we propose a quantization algorithm with error diffusion named Dither NN. Since we can use the adders that the neural network processor has, no additional arithmetic units are required for this extension. We implemented prototype architectures on an FPGA, and we trained convolutional neural network models with and without dithering. We proved the accuracy improvements with a very few additional hardware resource occupation through the evaluation.

This thesis provides a systematic methodology of architecture construction and a practical example of hardware-algorithm co-optimization, focusing on the computation primitives and data delivery patterns through these discussions and prototyping. The key contributions of this thesis include 1) a reconfigurable architecture with coarse-grained data flow switching motivated by the potential common bases among the algorithms, 2) a proof-of-concept near-memory architecture enhanced by a quantization algorithm, and 3) algorithmic optimization of quantization reflecting the hardware structure to achieve higher accuracy without damaging the efficiency. The methodology extracting the essence of computational structure and data topology enables efficient and flexible processors, embodying a form of hardware-software coevolution.