# ハードウェアディープラーニングアクセラレータの研究動向

Research Trends of Hardware Deep Learning Accelerators

植吉 晃大 Kodai Ueyoshi　高前田 伸也 Shinya Takamaeda　池辺 将之 Masayuki Ikebe　浅井 哲也 Tetsuya Asai　本村 真人 Masato Motomura

北海道大学 情報科学研究科
〒 060–0814 北海道札幌市北区北 14 条西 9 丁目
Graduate School of Information Science and Technology, Hokkaido University
Kita 14, Nishi 9, Kita-ku, Sapporo, Hokkaido, 060–0814
E-mail: ueyoshi@lalsie.ist.hokudai.ac.jp

## 1　まえがき

近年、ディープラーニングのハードウェア実装の研究が盛んに行われており、より面積・電力効率の良いアクセラレータが求められている。特に、畳み込みニューラルネットワーク (Convolutional Neural Network: CNN) の認識機能に特化したアクセラレータが多数提案されている。本稿では、それぞれのアーキテクチャの特性を調べ、その動向を解説する。さらに、ハードウェア志向なニューラルネットワークを取り上げ、これらを組み合わせることで、今後どのように効率的なアクセラレータを探索していくべきかを調査し、報告する。

## 2　畳み込みニューラルネットワークのハードウェアアクセラレータ

近年、ディープラーニングのハードウェア組み込みシステムの需要が高まってきている。これは、知能的計算技術の車載応用や IoT デバイスへの普及において、面積・電力効率の良い組込みシステムが求められているからである。特に、学習済みの畳み込みニューラルネットワークを用いた識別高速化のためのアクセラレータが期待されている。

CNN は、畳み込み層を持ったニューラルネットワークで、通常のニューラルネットワークと同様に順伝播と誤差逆伝播法を用いて、クラス分類の結果から、教師有り学習を可能とする。この認識精度が画像認識の権威あるコンテスト ILSVRC2012 で非常に高い成果を打ち出した [1]。それ以降、研究が盛んに進められ、実用化が進められている。しかし、そのネットワーク規模は年々増加傾向にあり、これを現実の制約下で実用するには、面積・電力効率の優れた専用アクセラレータが必要である。

一般的なニューラルネットワークアクセラレータは、積和演算を並列に行なう構造をとり、メモリバンド幅ボトルネックとなっていたが、CNN の演算の 90%を占める畳み込み演算は、多くのパラメータが再利用性を有しているため、演算処理とメモリバンド幅のバランスが重要となる。これまでに、この畳込み演算に着目した種々の専用アクセラレータが研究されている。これらは主に 4 種のパターンに大別できる。

(A)Park ら [2] や Qiu ら [3] は重み (Weight) をローカルレジスタに持ち、画素 (Activation) または途中結果 (Psum) をグローバルに伝播させる手法を取った。これらは、CNN の重みの再利用性を最重要視した構成で、重みの読み出し回数を最小化することができる。

(B)Zidong らの ShiDaDiannao[4] は途中結果をローカルレジスタに持ち、画素または重みをグローバルに伝播させた。これにより、画素と重みの連続性を活かしながら、途中計算をローカルに保持することで計算結果の移動コストを最小化させている。

以上の (A)(B) は要素一つに優先事項を偏らせているため、ネットワーク構造の変化に弱い。そこで、(C)Chen らの Eyeriss[5] は重みと画素を行毎にグループ化して大局的に最適化している。行毎にローカルバスを設け、二次元構造を最大限に利用することで、グローバルバッファからの読み出し回数を最小化させることで省電力化を実現した。さらに、各ニューロンの出力がほとんど 0 という特性を用いた圧縮表現と演算スキップ機能を設けることで、効率化を図っている。

しかし、これら 3 種はネットワーク構造が畳み込みに特化してしまい、汎用性に欠ける。そのため例外処理をしたり、固定のネットワークを用いることでこの問題を回避している。一方、(D)Zhang らは、任意のネットワーク構造から最高性能を見つけ出す探索方法を紹介し、FPGA によって実装している [6]。この探索手法では、ローカルレジスタを持たず、グローバルバッファからの任意に入力を選択する。Zhang らは、各層の演算を可能とする演算器のルーフラインモデルを組み合わせ的に洗い出し、与えられたメモリバンド幅と面積制約の中で最も高い計算性能とその構造を算出している。

このように、ハードウェア化による CNN の高速化手法が提案されている。しかし、急激に進化するニューラルネットワーク構造に対応するには、性能や電力効率だけではなく、高い汎用性を伴った高速化技術が今後の重要となる。

## 3　ディープニューラルネットワークの軽量化

ハードウェア化による高速化の一方で、ニューラルネットワークそのものの軽量化の取り組みが行われている。そこで、実装するネットワーク規模を小さくする、圧縮手法が研究されている。枝刈りやクラスタリングによって、ネットワークの表現を減らす事ができる [7][8]。

また、認識精度を落とさずに実装する計算コストを削減する手法がある。Google や NVIDIA は、従来の float 演算から、bit 精度を落とした計算機を発表した。さら

に、bit 精度削減の量子化手法も多数発表されている。その中でも、究極的に二値にまで削減したものが二値化ニューラルネットワークである [9]。

二値化を行えば、積和演算における乗算を XNOR ゲート一つで代用でき、加算器も小さく、さらにメモリバンド幅も大幅に小さくすることができる。しかし、二値化ニューラルネットワークの学習を収束させるには、バッチノーマライゼーションと呼ばれる、正規化手法が用いられる [10]。これにより、値が二値の閾値領域近傍へ分布するように正規化を行なうことで学習を収束させることができる。これは、認識時にも用いられるため、この演算がボトルネックとなり得る。そこで、認識時に限っては符号部のみが得られれば良いという着想から、この演算を簡易化させることができる手法が報告されている [11]。

我々は、この手法を応用して、Zhang らの最適化探索を二値化に適用させた。二値化によって極端に面積・バンド幅効率の改善が得られることから、CNN の複数個存在する並列性を組み合わせることで、より最適な構成をとる事ができることを示す。また、この並列性を用いると、層毎の構造の変化量が大きくなるに連れ、静的な構成と動的な構成での性能差が大きくなる。そのため、インターコネクト部の汎用化とそのオーバーヘッドのトレードオフを考察した。

## 4 まとめ

本稿では、ディープラーニング専用ハードウェアアクセラレータの動向を CNN の認識機能に着目して報告した。また、ニューラルネットワークのハードウェア志向アルゴリズムを取り上げた。これらを用いたハードウェア探索手法を発表では解説する。

## 参考文献


[1] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems 25, eds. by F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, pp.1097–1105, Curran Associates, Inc., 2012.

[2] S. Park, S. Choi, J. Lee, M. Kim, J. Park, and H.J. Yoo, "14.1 a 126.1mw real-time natural ui/ux processor with embedded deep-learning core for low-power smart glasses," 2016 IEEE International Solid-State Circuits Conference (ISSCC), pp.254–255, Jan. 2016.

[3] J. Qiu, J. Wang, S. Yao, K. Guo, B. Li, E. Zhou, J. Yu, T. Tang, N. Xu, S. Song, Y. Wang, and H. Yang, "Going deeper with embedded fpga platform for convolutional neural network," Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, pp.26–35, FPGA '16, ACM, New York, NY, USA, 2016. http://doi.acm.org/10.1145/2847263.2847265

[4] Z. Du, R. Fasthuber, T. Chen, P. Ienne, L. Li, T. Luo, X. Feng, Y. Chen, and O. Temam, "Shidiannao: Shifting vision processing closer to the sensor," Proceedings of the 42Nd Annual International Symposium on Computer Architecture, pp.92–104, ISCA '15, ACM, New York, NY, USA, 2015.

[5] Chen, Yu-Hsin and Krishna, Tushar and Emer, Joel and Sze, Vivienne, "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," IEEE International Solid-State Circuits Conference, ISSCC 2016, Digest of Technical Papers, pp.262–263, 2016.

[6] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, "Optimizing fpga-based accelerator design for deep convolutional neural networks," Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, pp.161–170, FPGA '15, ACM, New York, NY, USA, 2015.

[7] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M.A. Horowitz, and W.J. Dally, "Eie: Efficient inference engine on compressed deep neural network," Proceedings of the 43rd International Symposium on Computer Architecture, pp.243–254, ISCA '16, IEEE Press, Piscataway, NJ, USA, 2016. https://doi.org/10.1109/ISCA.2016.30

[8] S. Han, H. Mao, and W.J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," International Conference on Learning Representations (ICLR), 2016.

[9] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," CoRR, vol.abs/1609.07061, 2016. http://arxiv.org/abs/1609.07061

[10] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift.," ICML, eds. by F.R. Bach and D.M. Blei, vol.37, pp.448–456, JMLR Workshop and Conference Proceedings, JMLR.org, 2015.

[11] H.I. Hiroki Nakahara, Haruyoshi Yonekawa and M. Motomura, "A batch normalization free binarized convolutional deep neural network on an fpga," 25th ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (ISFPGA), 2017 (to appear).