A-02

原子スイッチを用いた再構成可能デバイスによる CNN の実装の実現可能性検討

A Feasibility Study on the Implementation of CNN using Atom-switch-based Reconfigurable Device

久富 亘† Wataru Kutomi
越智 裕之†‡ Hiroyuki Ochi

1. はじめに

高度なパターン認識の実現を目指して，ディープラーニングが研究されている．ディープラーニングは，ニューラルネットワークを多層的に構成した機械学習の方法論である。近年音声、画像を対象とした識別問題に対し，ディープラーニングが適用されつつある．なかでもディープラーニングの一つである畳み込みニューラルネットワーク (Convolutional Neural Network, CNN) は，画像認識分野で他の手法を圧倒する識別率を達成し，注目されている．

CNN は膨大な計算量を要するため，汎用 CPU で実装すると多大な演算時間を要する．そのため演算性能を改善することが重要である．文献[1]では，GPU で実装することで演算性能を改善している．しかし，一般に GPU での実装は消費電力が大きい．そこで我々は、演算性能も高く低消費電力な再構成可能デバイスによる CNN の実装について検討する．

再構成可能デバイスの中でも，FPGA は近年最も成長を遂げてきたデバイスの一つである．FPGA は，論理ブロックと入出力用ブロック，それらを接続するための配線から構成されている．そしてこれらの資源がプログラマブルであるため，自由な配線，自由な論理を構成することができる．しかし FPGA に搭載されている乗算器の数が潤沢でないため，処理の並列化には限界がある．

近年，新しいナノデバイスである原子スイッチを使用した再構成可能デバイスが注目されている．従来の FPGA は資源のプログラマビリティをパストランジスタと SRAM を用いて実現しているのに対し，原子スイッチを使用すれば配線層のみで実現できるために総ロジック面積を少なくすることができる．文献 [2] では，SRAM ベースの FPGA にくらべ，約 78%の回路規模の縮小が報告されている．

本稿では，原子スイッチを応用した FPGA の特徴をいかした CNN の実装について検討する．CNN は畳み込みとプーリングが最も支配的な部分であるため，この二つの処理を提案手法で実装し，処理の高速化を行う．そして実装に必要な資源を明らかにしたうえで，実装の実現可能性について検討する．

以下，2 章で CNN と原子スイッチの説明を行い，3 章で提案手法の説明，4 章ではハードウェアの実現可能性を検討し，最後に 5 章でまとめを述べる．

2．背景

本章では，実装対象である CNN と原子スイッチの概要について述べる．

2.1　畳み込みニューラルネットワーク (Convolutional Neural Network; CNN)

CNN は特に画像認識で応用されているディープラーニングである．2012 年，物体の認識率を競うコンテスト(ILSVRC)において Hinton 率いるトロント大学が CNN によってエラー率約 17% で従来手法 (エラー率約 27%) を凌駕する結果を出した [3]．一般的な CNN は，畳み込みとプーリングを複数組み合わせて構成することで高い識別率を実現している．

2.1.1　畳み込み処理

畳み込みとは，入力画像と重みとよばれるフィルタ間の積和演算 (式 2.1)である (図 2.1).

$$u_{ij} = \sum_{p=0}^{H-1}\sum_{q=0}^{H-1} x_{i+p,j+q} w_{pq} \tag{2.1}$$

ここで入力画像サイズを $I \times I$ 画素とし，画素をインデックス(i, j) (i = 0, 1, …, I–1, j = 0, 1, …, I–1)で表し，フィルタの画像サイズを $H \times H$ 画素とし同様に，画素をインデックス (p, q) (p = 0, 1, …,H–1, q = 0, 1, …, H–1) で表す．そして入力画像とフィルタの画素値をそれぞれ x，h とする．畳み込みはフィルタ画像の濃淡パターンと類似の濃淡パターンが入力画像のどこにあるかを検出するはたらきがある．つまり，フィルタが表している特徴量を画像から抽出するのである．

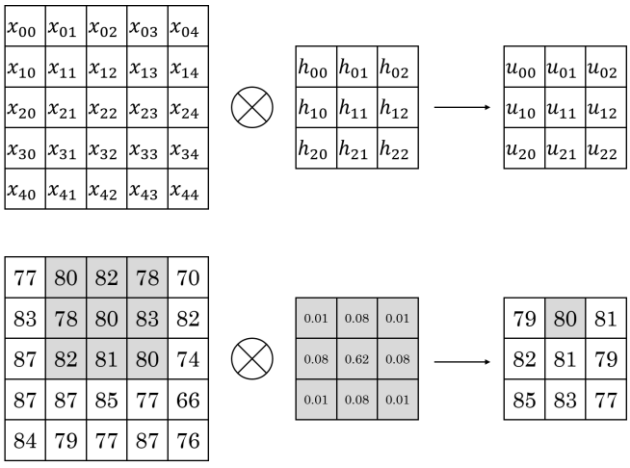

図 2.1: 畳み込み処理の概要

図 2.1 のように畳み込みは入力画像にフィルタを重ねたとき，画像とフィルタの重なり合う画素どうしの積を求め

† 立命館大学大学院情報理工学研究科, Graduate School of Information Science and Engineering, Ritsumeikan University
‡ 立命館大学情報理工学部, College of Information Science and Engineering, Ritsumeikan University

て，フィルタ全体の和を求めている．したがって画像からフィルタがはみ出すような位置に重ねることができず，画像内にフィルタ全体が収まる範囲内でフィルタを動かすと，畳み込み結果である出力画像サイズは，入力画像よりも小さくなる．このとき，出力画像サイズは式 2.7 で表すことができる．

$$(I-K+1)/(stride) \tag{2.2}$$

stride とはフィルタが動く幅を指しており，本稿で表現されている畳み込みはすべて，stride = 1 である．

2.1.2 プーリング処理

プーリングは，通常畳み込みの直後に配置される．プーリング処理には，データサイズを減らし対象領域の些細な特徴量の違いを吸収し，その領域内の代表的な特徴量を抽出するはたらきがある．またその処理にはいくつか方法が存在し，最大プーリングという処理が一般的である．図 2.2 では，4×4 の入力に対して，2×2 の領域ごとに分割する．そしてそれぞれの領域内から最大値を抽出し，2×2 の画像を出力する．この場合，2×2 の領域ごとに処理を行うことによって，データサイズが 16(= 4×4) から 4 (= 2×2) と 1/4 になる．

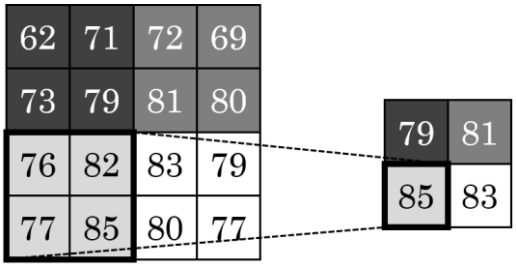

図 2.2: プーリング処理の概要

2.1.3 CNN の支配的な処理

前節でも述べたように，CNN は畳み込みとプーリングが多層にわたって構成されている．本節では， CNN ツールである Caffe [4] を用いて，汎用 CPU 上で CNN の処理時間を測定する．テストモデルとして，MNIST とよばれる手書き数字のデータセットを使用する．MNIST は，28×28 ピクセルのグレースケール画像である．MNIST における CNN の識別ネットワークは図 2.3 のとおりである．表 2.1 は各処理層の入出力パラメータである．ここで conv と pool は畳み込みとプーリングであり，ip と ReLU は全結合と活性化関数である．表 2.1 のパッチサイズについて conv の場合はフィルタサイズ，pool の場合は対象領域サイズを示している．

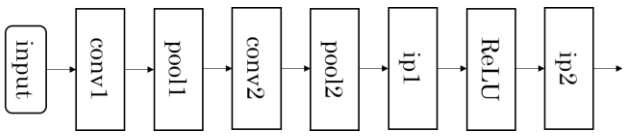

図 2.3: MNIST における CNN ネットワーク

表 2.0.1: 各処理層の入出力パラメータ

	入力サイズ (*I*×*I*×*K*)	出力サイズ (*I*×*I*×*K*)	パッチサイズ (*P*×*P*)
conv1	28 x 28 x 1	24 x 24 x 20	5 x 5
pool1	24 x 24 x 20	12 x 12 x 20	2 x 2
conv2	12 x 12 x 20	8 x 8 x 50	5 x 5
pool2	8 x 8 x 50	4 x 4 x 50	2 x 2
ip1	4 x 4 x 50	1 x 1 x 500	---
ReLU	1 x 1 x 500	1 x 1 x 500	---
ip2	1 x 1 x 500	1 x 1 x 10	---

表 2.2: 各処理層の処理時間と比率

	処理時間 [ms]	比率 [%]
conv1	10.67	23.6
pool1	7.04	15.5
conv2	21.60	47.8
pool2	2.40	5.3
ip1	3.28	7.2
ReLU	0.040	0.1
ip2	0.18	0.5

CNN ネットワークにおいて各畳み込み層のフィルタの値をランダムに初期化し，識別を 100 回行ったときの処理時間の平均値とその比率を表 2.2 に示す．全体の 71.4% を conv が占めており，19.8% を pool が占めている．conv1 に比べ，conv2 の処理時間が多いのはフィルタの数が conv1 より多いことが主な要因である．pool において pool1 が行う処理数は 240(=12×20) に対して，pool2 は 200(=4×50) であるため，pool2 の処理時間が少ない．

2.2 原子スイッチ

原子スイッチとは，金属による架橋とその消滅を利用し

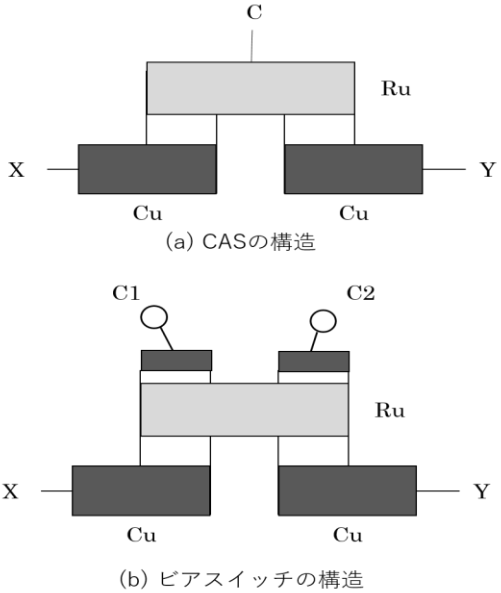

図 2.4: CAS とビアスイッチの構造

た不揮発性のスイッチ素子である[2]．原子スイッチは，銅(Cu) とルテニウム(Ru) と，それらに挟まれた固体電解質からなる．Cu からプラスの電圧が印加されることで，固体電解質の間に Cu から Ru にかけて Cu イオンが析出し，架橋が生成される．また，Cu からマイナスの電圧が印加されることで，固体電解質の間ある架橋が消滅する．

従来の FPGA のプログラマブルスイッチは，ON，OFF の構成情報を SRAM が保持し，その情報を用いパストランジスタによって導通および非導通を実現する．一方原子スイッチは，スイッチングと構成情報保持が一体化しており，原子スイッチ素子のみで回路内のスイッチングを実現可能であるため，回路面積が大幅に削減される．文献 [2] では，SRAM ベースの FPGA に比べ，約 78%の回路面積の削減が報告されている．

また原子スイッチを用いた再構成可能アーキテクチャが提案されている [2]．再構成可能アーキテクチャを構成するために，文献[2]では，原子スイッチを図 2.4(a) のように Complementary Atom Switch (CAS) という形に置いて実現している．これは原子スイッチを二つ逆向きに直列接続させた形になり，原子スイッチ単体よりも信頼性が向上する．

しかし CAS でクロスバスイッチを実現するには，アクセストランジスタを必要とした．スイッチよりもアクセストランジスタの方が大きいため，スイッチ当たりの面積が増大してしまう．文献[7]では，ビアスイッチを用いた再構成可能アーキテクチャを提案している．ビアスイッチは図 2.4(b)のように CAS と 2 つのバリスタで構成されており，アクセストランジスタを用いることなくクロスバスイッチを実現した．

3. 提案手法

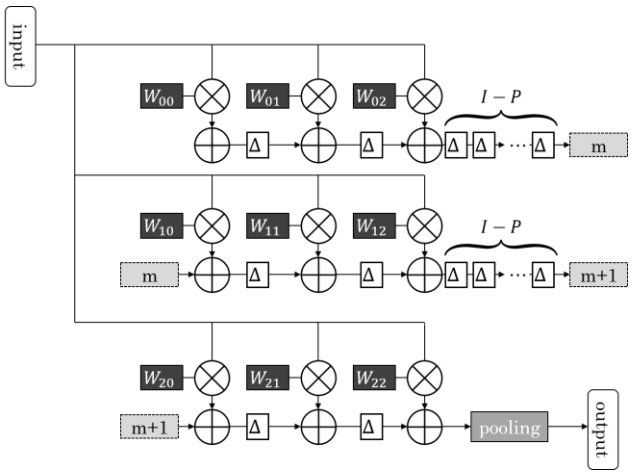

図 3.1: 提案手法の概要

前章でも述べたように，畳み込み処理が CNN において最も支配的な処理である．この畳み込みについてハードウェアで実装することで，処理の高速化を図る．畳み込み処理は積和演算の繰り返しであるため，積和演算を並列処理することが重要である．しかし FPGA に搭載されている乗算器は豊富ではないため，畳み込み処理の並列化には限界がある．文献[5]では原子スイッチを用いることで，乗算器やメモリを潤沢に搭載したアーキテクチャを提案している．本章ではこのアーキテクチャを用いた畳み込みの実装手法について提案する．

我々は文献[6]を参考にして，畳み込みおよびプーリング処理を実行する方式を提案する (図 3.1)．入力画像は外部から受け取り，また 1 クロックごとに 1 ピクセルだけ受け取ることを前提とする．図 3.1 における W はフィルタ係数，I は入力画像の幅，P はフィルタの幅，Δ はレジスタを指している．入力画素を受け取るたびに，乗算を行いその演算結果を次の積和演算器にシフトしていく．P 回だけ積和演算を行ったあと，$I-P$ 回シフトを行い，また積和演算を P 回だけ行う．これを P 回繰り返したあと，その出力結果でプーリング処理を行い，画素値を出力する．図 3.1 ではフィルタの幅が 3 で，フィルタサイズが 9(=3 × 3) である．つまり，3 回積和演算行ったあと，$I-3$ 回シフトを行う流れを 3 回行うと畳み込み結果が出力される．このとき乗算器が 9 個あれば実装することができ，フィルタの枚数に応じて並列化を行うことで，処理の高速化が見込める．

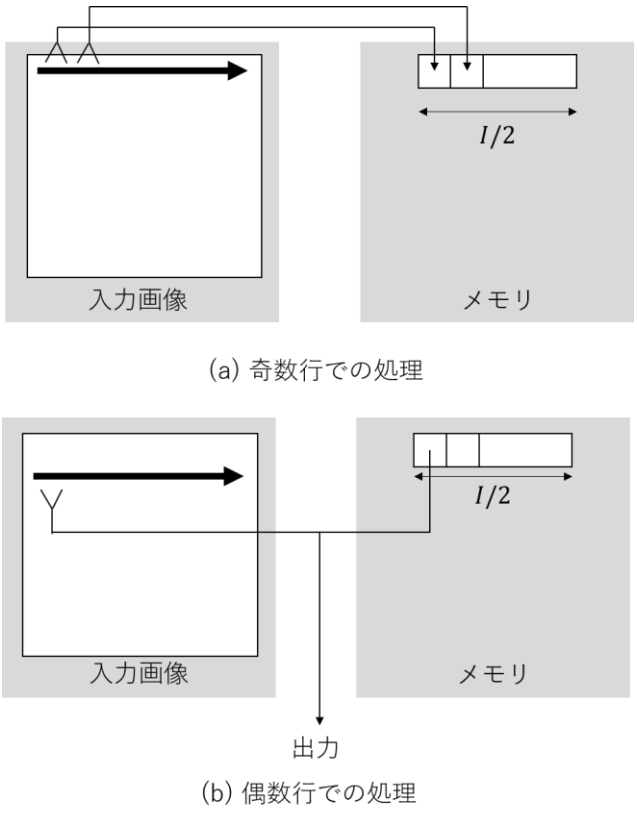

図 3.2: プーリング処理の概要

プーリング処理の概要を図 3.2 に示す．図 3.2(a)は，奇数行の画像を入力として，受け取ったときの処理である．2 画素を比較して，大きい値をメモリに保持する．このメモリ容量は $I/2$ である．図 3.2(b)は偶数行の処理である．2 画素を比較した大きい値とメモリが保持している値を比較して，大きい値を出力する．

4. 性能評価と実現可能性検討

本章では 3 章で提案した実装手法を用いて，性能評価と必要な資源を明らかにしたうえで，原子スイッチを用いた再構成可能アーキテクチャで実現できるかについて検討する．

4.1 性能評価

表 4.1: 提案手法のサイクル数と処理時間

	提案手法		Caffe 処理時間
	サイクル数	**処理時間**	
conv1, pool1	**793**	**7.93 μs**	**12.64 ms**
conv2, pool2	**148**	**1.48 μs**	**25.03 ms**

2.1 節で述べた MNIST の CNN ネットワークを用いて，提案手法の性能評価を行う．conv1 は，表 2.1 に示した通り入力画像サイズが 28×28 で，フィルタサイズが 5×5 である．conv1 と pool1 は 5 画素分だけ積和演算を行ったあと，23(=28-5)回シフトを行い，下段の積和演算器に入力される．この流れを 5 段構成にすることで，conv1 と pool1 を実現している．conv2 もフィルタサイズが 5×5 であるため，同様に実装することができるが，入力画像サイズが 12×12 であるため 5 画素分の積和演算を行ったあとシフトする回数は，7(=12-5) である．

conv1，pool1 と conv2，pool2 の処理に要するサイクル数と処理時間を表 4.1 に表す．ここで，提案手法の動作周波数を 100MHz とし，Caffe 実行に使用した CPU は Intel core i5-6300U @2.40GHz 2.50GHz である．提案手法における畳み込みは最大限並列化を行っていて，conv1 は約 1600 倍，conv2 は約 1700 倍 Caffe よりも高速であることがわかる．

4.2 実現可能性の検討

表 4.2: 提案手法で要する乗算器とメモリの諸元

	conv1, pool1	conv2, pool2
乗算器数	**500**	**25,000**
メモリ語数	**28,800+240 語**	**200 語**

提案手法で要する乗算器数とメモリ語数を表 4.2 に示す．conv1 においてフィルタサイズが 5×5，そしてフィルタ枚数が 20 枚であるため要乗算器数は 5×5×20 = 500 個となる．メモリ語数は畳み込みの出力画像を一時的に保存するときとプーリング処理で一時的に保存するときに必要とする語数である．conv1 の出力画像サイズは 24×24 で出力枚数が 20 枚であるため，必要とするメモリ語数は 24×24×20 = 28,800 語である．pool1 で必要とするメモリ語数は 12×20 = 240 語である．conv2 はフィルタサイズが 5×5，フィルタ枚数が 50 枚であり，入力画像枚数が 20 枚であるため要乗算器数は，5×5×50×20 = 25,000 個である．pool2 で要するメモリ語数は，4×50 = 200 語である．

文献[5] で提案されているアーキテクチャは 1 タイルにつき乗算器が 1 つ搭載されており，1 タイル当たりの面積は 5.319μm^2 である．チップ面積が 25mm^2 ならばタイル数は 4,700 個，100mm^2 ならばタイル数は 18,800 個になる．チップ面積が 133mm^2 以上ならば，実装が可能であるといえる．

5. おわりに

本稿では，乗算器やメモリを潤沢に利用することを前提として，原子スイッチを用いた再構成可能デバイスによる CNN の実装手法について提案した．提案手法は文献[6]を基に，処理の並列化を最大限行い，畳み込み処理とプーリング処理に要する資源を明らかにした．チップ面積が 133mm^2 以上ならばアーキテクチャの実装が可能であることを示した．今後は並列化について考察をしたうえで，チップ面積の削減や配線を含めた実現可能性について評価を行いたい．

参考文献

[1] S. Puri, P. Simard, K. Chellapilla, “High performance convolutional neural networks for document processing,” Tenth International Workshop on Frontiers in Handwriting Recognition, Oct 2006.

[2] M. Miyamura, M. Tada, T. Sakamoto, N. Banno, K. Okamoto, N. Iguchi, H. Tada“First demonstration of logic mapping on nonvolatile programmable cell using complementary atom switch,” IEDM, 2012.

[3] A. Krizhevsky, I. Sutskever, G. E. Hinton, “Imagenet classification with deepconvolutionak neurak networks,” *Advances in neural information processing systems* , 2012, pp. pp1097-1105.

[4] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” Proceedings of the 22nd ACM international conference on Multimedia, 2014.

[5] J. Hotate, T. Kishimoto, T. Higashi, H. Ochi, R. Doi, M. Tada, T. Sugibayashi, K. Wakabayashi, H. Onodera, Y. Mitsuyama, M. Hashimoto, “A highly-dense mixed grained reconfigurable architecture with overlay crossbar interconnect using via-switch” 26th International conference on field programmable logic and applications, 2016.

[6] C. Farabet, C. Poulet, Y. LeCun, “CNP: An FPGA-based processor for convolutional networks,” 2009 International Conference on Field Programmable Logic and Applications, 2009.

[7] M. Tada, T. Sugibayashi, K. Wakabayashi, H. Onodera, Y. Mitsuyama, M. Hashimoto, M. Hashimoto, H. Ochi, N. Banno, K. Okamoto, N. Iguchi, T. Sakamoto, Y. hada, Y. Tsuji, “A novel two-varistors (a-Si/SiN/a-Si) selected complementary atom switch (2V-1CAS) for nonvolatile crossbar switch with multiple fan-outs” Dig. IEDM, pp32-35, 2015