C-020

CNN ハードウェアにおける DRAM アクセス量削減手法

An Approach to Reduce DRAM Access for CNN Hardware

古川 巧†, 望月香那†, 黒田幸作†, 廣瀬哲也†, 黒木修隆†, 沼 昌宏†
Takumi Furukawa†, Kana Mochizuki†, Kousaku Kuroda†,
Tetsuya Hirose†, Nobutaka Kuroki†, and Masahiro Numa†

1. はじめに

近年，画像認識分野で有効な畳み込みニューラルネットワーク（CNN：Convolutional Neural Networks）に注目が集まっている。しかし，CNN は，膨大な処理時間を必要とするため，アクセラレータを用いて処理を高速化する必要がある。このアクセラレータとして，従来は GPGPU（General Purpose Graphic Processing Unit）が利用されることが一般的であったが，大規模なデータセンタ等での利用を前提とした場合，GPGPU による処理では電力効率が低い点に問題がある。そこで，CNN システムをハードウェアで実現するアーキテクチャが提案されている。書き換え可能な FPGA（Field Programmable Gate Array）上のハードウェアで CNN を実装することで，消費電力を約 10 分の 1 に低減できる。一方で，FPGA を用いたアクセラレータにおいても，処理途中で利用するデータは一時的に DRAM に保持される。そのため，扱うデータ量が膨大な CNN では，DRAM アクセスによる消費電力が大きな割合を占める点に問題があった [1]。本稿では，CNN ハードウェアの低消費電力化実現を目的として，畳み込み層の出力を並列化するとともに，次段のプーリング層との間を FPGA 内蔵の BRAM（Block RAM）で接続し，畳み込み層とプーリング層を並列に処理することで DRAM アクセス量を削減する手法を提案する。

2. 提案手法

2.1 従来手法のデータ移動量

図 1 に，通常の処理手順に従った CNN の例を示す。CNN を FPGA で実装する場合，畳み込み層からの出力特徴マップを一度 DRAM に転送し，再び DRAM から FPGA にプーリング層の入力特徴マップとしてデータを転送することが一般的である。N_{ci} 枚の入力特徴マップに対して N_{ci} 枚の重みフィルタを畳み込んだ結果のすべてを加算し，出力特徴マップとする。この処理を N_{co} セットの重みフィルタで行い，N_{co} 枚の特徴マップを生成する。この時，DRAM 入出力データ量定式化のため，次の記号を定義する：

- N_{ci}, S_{ci}：畳み込み層入力特徴マップ数とサイズ
- S_k：重みフィルタのサイズ
- N_{co}, S_{co}：畳み込み層出力特徴マップ数とサイズ
- N_{pi}, S_{pi}：プーリング層入力特徴マップ数とサイズ
- N_{po}, S_{po}：プーリング層出力特徴マップ数とサイズ

ここで，$N_{co} = N_{pi} = N_{po}$，$S_{co} = S_{pi}$ の関係が成り立つ。読込データ量 D_{RDconv} は

$$D_{RDconv} = S_{ci} N_{ci} + S_k N_{ci} N_{co} + S_{pi} N_{pi} \quad (1)$$

で表され，書き出しデータ量 D_{WRconv} は

$$D_{WRconv} = S_{co} N_{co} + S_{po} N_{po} \quad (2)$$

で表される。ここで，式 (2) における S_{co} N_{co} は，式 (1) における S_{pi} N_{pi} と一致し，畳み込み層とプーリング層の間で一時的に DRAM に保存されるデータ量を表す。

2.2 提案手法のデータ移動量

DRAM アクセス量は，畳み込み層の出力特徴マップすべてを FPGA 内蔵の SRAM に読み込むことがもし可能であれば，削減が可能である。しかし，この方法では保持すべきデータ量の増大を招き，その分の SRAM の容量を FPGA 内部に確保することは困難である。そこで図 2 に示すように，中間層のデータの一部を一時的に保持する手法を提案する。M 並列で出力された 1 行分の出力特徴マップのみを FPGA 上の M 並列に接続された BRAM に格納し，1 行分毎のデータをプーリング層に転送して畳み込み層とプーリング層のパイプライン処理を行うことで，SRAM の容量の範囲内で中間の特徴マップを保持することができる。提案手法における読み込みデータ量 D_{RDprop} は

$$D_{RDprop} = S_{ci} N_{ci} + S_k N_{ci} N_{co} \quad (3)$$

で表され，書き出しデータ量 D_{WRprop} は，

$$D_{WRprop} = S_{po} N_{po} \quad (4)$$

で表される。よって，提案手法による DRAM アクセス量削減率 R_{DRAM} は

$$R_{DRAM} = 1 - \frac{D_{RDprop} + D_{WRprop}}{D_{RDconv} + D_{WRconv}} \quad (5)$$

で表される。

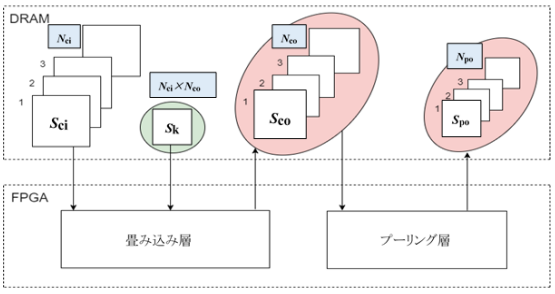

図 1 従来手法のデータ移動

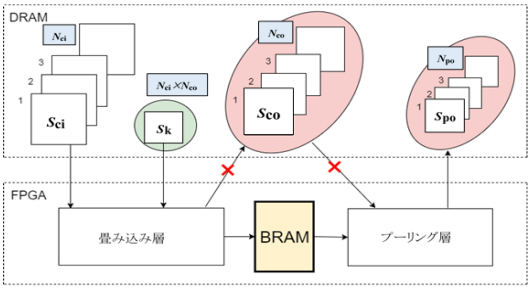

図 2 提案手法のデータ移動

†神戸大学, Kobe University

3. 提案アーキテクチャ

図 3 に，提案する回路の全体構成を示す。回路全体の処理の流れについて説明する。まず，入力特徴マップが DRAM からシフトレジスタを介して M 並列の畳み込み処理部へと転送される。次に，畳み込み処理部から 1 行分のデータが M 並列で BRAM へ出力される。 BRAM に格納されたデータは 1 行分ずつプーリング処理部に流れるように出力され， DRAM へ転送される。提案回路では畳み込み処理部を M 並列で動作させることにより，各 BRAM に保持するデータの容量を特徴マップ 1 行分のデータに抑えている。これにより，容量の小さい SRAM に，畳み込み層の出力特徴マップを保持することを可能にしている。図 4 に，AlexNet [1] に対応したプーリング処理部のアーキテクチャを示す。プーリング処理部では，BRAM から入力特徴マップの隣接するデータが出力され，シフトレジスタを介して前段の比較器で比較される。その後 1 行分のデータが個々の分散 RAM に格納され，後段の比較器で行ごとのデータの比較を行う。プーリング処理部のアーキテクチャは，データを並列に処理するパイプライン形式で構成することで，処理時間を短縮する。

4. 実験と評価

2 章で，提案した処理手順を用いた際の DRAM アクセス量の削減率に関する定式化を行った。畳み込みニューラルネットワークモデルの AlexNet において，提案手法を適応した場合の評価結果を表 1 に示す。AlexNet では，従来手法に比べて畳み込み層とプーリング層の中間特徴マップの DRAM アクセスを抑えられ，AlexNet 全体の DRAM アクセス量を約 35% 削減できることが，計算上確認できた。

次に，Verilog HDL によりハードウェア設計を行い，FPGA にマッピングすることでリソース数の評価を行うとともに，Xilinx 社の Vivado を用いた動作シミュレーションにより， DRAM アクセス量と処理時間の比較を行った。マッピングの結果を表 2 に示す。利用した FPGA のリソース内で提案手法の回路を実装可能であることが確認できた。

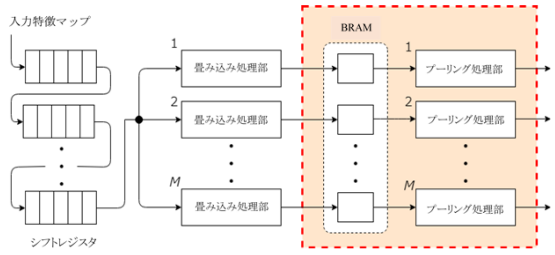

図 3 提案回路の全体構成

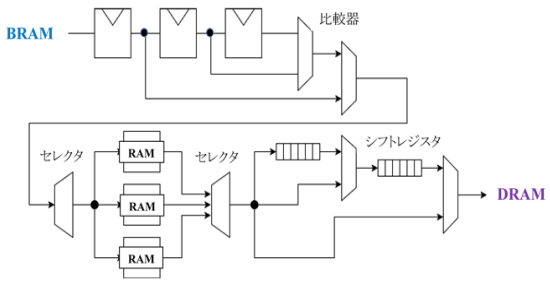

図 4 プーリング処理部のアーキテクチャ

表 1 AlexNet における DRAM アクセス量の見積もり

層	従来	提案	削減率 [%]
conv1+pool1	840,219	259,419	69.13
conv2+pool2	1100,896	727,648	33.91
conv5+pool5	1045,376	958,848	9.28

表 2 マッピング結果

層	LUT [$\times 10^3$ 個]	RAM [$\times 10^3$ 個]	FF [$\times 10^3$ 個]	BRAM [個]
conv1	13	2.0	10	0
conv2	12	1.5	9.7	0
conv3	13	1.4	11	0
conv4	73	1.4	1.3×10^2	0
conv5	73	1.4	1.3×10^2	0
pool1	17	12	37	48
pool2	37	20	94	128
pool5	37	20	92	128

表 3 動作シミュレーションによる処理時間評価

層	従来 [s]	提案 [s]	提案/従来 [%]
pool1	1,580	30.90	1.96
pool2	1,110	7.62	0.69
pool5	3,080	1.88	0.61

また，従来手法と提案手法に対する，動作シミュレーションによる処理時間結果を表 3 に示す。畳み込み層の出力特徴マップ数を M とすると，提案手法では，畳み込み層の出力特徴マップを M 並列かつパイプライン形式で処理することで，すべてのプーリング層で，処理時間が従来手法の約 M 分の 1 になることを確認し，90% 以上の削減可能となることも確認できた。

5. まとめ

本稿では，CNN をハードウェアとして実現する際の低消費電力化を目的として， DRAM アクセス量を削減するアーキテクチャを提案した。従来手法と提案手法を比較して，CNN モデルの AlexNet における DRAM アクセス量の理論値を定式化し，提案アルゴリズムの妥当性を評価した。見積もりの結果，対象のモデルで DRAM アクセス量を約 35% 削減可能であることが確認できた。

Verilog HDL を用いて実際に回路を設計し，Xilinx 社の Vivado を利用してシミュレーションによる評価を行った。その結果，利用した FPGA のリソース内で回路が設計可能であることが確認できた。同時に，畳み込み層での出力の並列処理によって，提案手法は従来手法と比べて，処理時間を 90% 以上削減可能であることを確認した。

参考文献

[1] Y. H. Chen, et al., “Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks,” IEEE Journal of Solid-State Circuits, vol. 52, no. 2, pp. 127-138, Jan. 2017.