C-001

層多重化構造に基づく CNN のハードウェア化に向けた適応的グループ化畳み込み手法
Adaptive Grouped Convolution Method for Hardware CNN Based on Layer Multiplexing Structure

河端 佑一郎† 黒木 修隆† 沼 昌宏†
Yuichiro Kawabata Nobutaka Kuroki Masahiro Numa

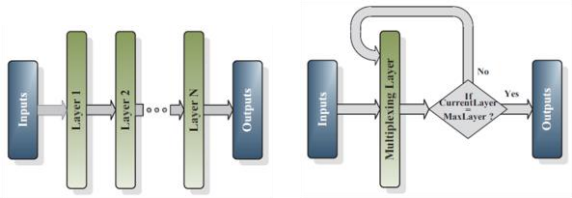
1. はじめに

近年，畳み込みニューラルネットワーク（CNN：Convolutional Neural Network）を用いた物体認識や画像・映像処理に関する研究が，盛んに行われている。CNN の推論および学習には膨大な計算量が必要であり，それらの処理を高速化するアクセラレータとして GPU（Graphics Processing Unit）が広く用いられてきたが，消費電力が大きいという問題がある。そこで，GPU より低消費電力で動作可能な FPGA（Field-Programmable Gate Array）上に CNN 専用の回路を実装することにより，低消費電力化を実現する研究が注目されている。図 1 にニューラルネットワークの回路構成を示す。図 1 (a) の一般的な回路ではパイプライン処理が可能である一方で，図 1 (b) の層多重化回路 [1] では回路規模を削減できる点にメリットがある。そのため，回路規模に制限がある FPGA で CNN を実装するには，層多重化回路が有利であると考えられる。

しかし，表 1 に示す VGG16 [2] における畳み込み層の構造のように，Pooling 層を経ることで出力チャネル数が増加し，特徴マップサイズが小さくなる構造は，特徴マップデータの効率的なメモリ・マッピングを妨げる。例として，図 2 (a) のように，Stage 1 の特徴マップデータを格納し，図 2 (b) のように，Stage 2 の特徴マップデータを格納してメモリ共有を行う場合を考える。ここで，図中における C_k は，ある層における k 番目のチャネルを示している。Stage 2 におけるチャネルが全結合の場合，C_1 から C_n のデータに加えて C_{n+1} から C_{2n} のデータを入力として用いるため，並列演算器数がチャネル数に対して不足する場合，演算結果を書き込むことができない。これは，メモリの共有が不可能であることを示している。また，同一空間座標のデータが不連続に保存されているため，ランダム・アクセスが必要になると同時に，バースト長に収まらないことによるアクセス回数増加を引き起こす可能性がある。

表 1　VGG16 内の畳み込み層 13 層の構造

Stage	Number of Layers	Output Channels	Feature Map Size
1	2	64	224
2	2	128	112
3	3	256	56
4	3	512	28
5	3	512	14



(a) 一般的な回路　　(b) 層多重化回路

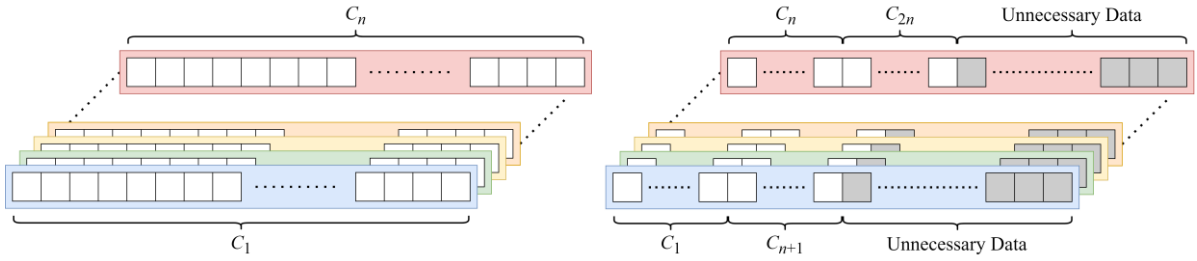
図 1　ニューラルネットワークの回路構成 [1]

そこで本稿では，層多重化回路を用いて効率的に CNN を FPGA 実装することを目的として，適応的にグループ化畳み込みを適用することによって，シーケンシャル・アクセス可能なアドレス・マッピングを行い，演算回路とメモリの両方を共有できるネットワーク設計手法を提案する。

2. 提案手法

2.1　適応的グループ化畳み込み

グループ化畳み込みは，入力された特徴マップを任意のグループ数 G に分割し，それぞれのグループ内に存在する特徴マップのみを利用して畳み込みを行う手法である。この分割によって，1 pixel の特徴マップを出力するために必要な演算量を $1/G$ に削減するとともに，パラメータ数を削



(a) Stage1　　(b) Stage2

図 2　特徴マップデータの格納

† 神戸大学，Kobe University

表 2　各手法による 1 画素当たりの演算量

Stage	DWConv		PWConv	
	従来手法	提案手法	従来手法	提案手法
1	k^2	k^2	n	n
2	k^2	k^2	$2n$	n
3	k^2	k^2	$4n$	n
4	k^2	k^2	$8n$	n
5	k^2	k^2	$8n$	n

表 3　PWConv 層のマッピング結果と必要メモリ容量

実装手法	リソース利用数（リソース利用率）			必要メモリ容量 [Mbit]
	LUT	FF	DSP	
従来手法	144,896 (52.7%)	4,864 (0.9%)	768 (31.5%)	8.84
提案手法				2.10

減する軽量化手法である。グループ数 G は層ごとに任意に設定できるため，各層の畳み込み演算量を均等化するようにグループ数 G を決める。表 2 に，DWConv 層と PWConv 層における，各手法による 1 画素あたりの演算量を示す。ここで，畳み込みのカーネルサイズを $k \times k$，Stage 1 における入力チャネル数を n とする。従来手法の演算量に応じて，Stage 1~5 に対してグループ数をそれぞれ 1，2，4，8，8 に設定する。提案手法によって，演算量が均一化され，離れたアドレスに保存されたデータが不要になると同時に，各データに対するアクセス回数を 1 回に減らせるため，演算に必要なデータアクセス時間を短縮できる。

2.2　レイヤ正規化によるチャネル間結合補正

CNN の学習において，学習の収束を高速かつ安定にする目的で挿入されるバッチ正規化層を，レイヤ正規化層に置き換える。それぞれの正規化層は，バッチ方向，レイヤ方向にサンプリングを行っている。グループ化畳み込みを用いた場合，畳み込みがグループ内でのみ行われるため，チャネル間の結合が弱くなる。そこで，レイヤ正規化を用いることで，疎結合なチャネルを補強することができると考えられる。

3. 実験と評価

3.1　ハードウェア実験と評価

PWConv 層の演算を 64 並列で行う回路を，ZCU102 上にマッピングした結果を表 3 に示す。リソースの最大利用率が LUT における 52.7% であり，余裕をもって収まった。

また，特徴マップデータを保持するメモリを層ごとに共有することで，必要メモリ容量を約 76% 削減した。

3.2　ソフトウェア実験と評価

提案手法の認識精度に対する評価を行う。10 クラス分類のデータセットである CIFAR-10 データセットを用いて，学習と推論を行った。VGG16 に Separable Convolution を適用したネットワークを従来手法とし，従来手法に適応的グループ化畳み込みを適用したネットワークを提案手法として，畳み込み層のパラメータ数を比較した。また，

表 4　畳み込み層のパラメータ数

評価層	従来手法	提案手法
DWConv	33,435	33,435
PWConv	1,634,498	266,432
合計	1,667,933	299,867

表 5　ソフトウェアによる認識精度評価

採用正規化層		認識精度 [%]	
DWConv	PWConv	従来手法	提案手法
Batch	Batch	73.4	70.1
Layer	Batch	**75.1**	70.3
Batch	Layer	72.6	72.1
Layer	Layer	70.3	**72.7**

DWConv 層と PWConv 層の後にある正規化層に関して，バッチ正規化層を適用した場合とレイヤ正規化層を適用した場合の認識精度を比較した。

表 4 に，実装した CNN における畳み込み層のパラメータ数を示す。提案手法では従来手法と比べてパラメータ数を約 82% 削減できることを確認した。また表 5 に，認識精度に関する結果を示す。バッチ正規化層のみを適用した従来手法の認識精度を基準として，適応的グループ化畳み込みを適用することで認識精度が 3.3 pt 低下するが，DWConv 層と PWConv 層の後にレイヤ正規化層を適用することで認識精度の低下を 0.7 pt に抑制する効果を確認した。また，従来手法において DWConv 層の後にレイヤ正規化層，PWConv 層の後にバッチ正規化層を適用した場合が，最も高い認識精度を示した。

4. おわりに

本稿では，CNN の効率的な FPGA 実装のため，層多重化構造に適した適応的グループ化畳み込み手法を提案した。

CNN において，層によってチャネル数が異なり，1 画素あたりの畳み込み演算量が不均一になる問題を，グループ内のチャネル数が等しくなるようにグループ数を設定することで解決した。提案手法によって，1 画素に対するメモリ・アクセス数を均一化し，特徴マップデータを保持するメモリ容量を約 76% 削減する効果を確認した。

また，提案手法によって生じるチャネル間結合が疎になることで，認識精度が低下する問題に関して，畳み込み層に続くバッチ正規化層をレイヤ正規化層に置き換えることでチャネル間の結合を補正し，認識精度を向上させることが可能であることを確認した。

今後の課題として，VGG16 以外の CNN や ViT に対して，提案手法を有効に活用することが挙げられる。

参考文献

[1] F. Ortega-Zamorano, J. M. Jereza, I. Gómeza, and L. Francoa, “Layer multiplexing FPGA implementation for deep back-propagation learning,” IOS Press, Integrated Computer-Aided Engineering, vol. 24, no. 2, pp. 171-185, 2017.

[2] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1556v6, 10 Apr. 2015.