4U-04

カーネル単位でのプルーニングによるディープラーニング専用ハードウェアの動作速度向上

宇津野　祐輔†　　瀬戸　謙修†
東京都市大学　工学研究科†

1　はじめに

近年、CNN(Convolutional Neural Network)を用いたディープラーニング技術は徐々にその応用範囲を拡大しており、組み込みシステム分野においては、例えば車載での応用が期待されている。しかし、CNN は積層化することによって高い認識精度を発揮するが、同時に内部パラメータ（重み）数が増大することにより、演算量が大幅に増加してしまう。そのため電力や搭載可能な面積に制約がある組み込みシステムにおいて、CNN の演算量は実用化への大きな課題となっている。また、CNN の実装手段として FPGA (Field Programmable Gate Array)に注目が寄せられている[1]。FPGA は柔軟に回路構成を変えることができるため、CPU や GPU に比べ電力効率の良い処理が行える。

本研究では、プルーニング（Pruning；枝切り）手法を用いた CNN のパラメータ削減手法を提案する。これにより FPGA 実装された CNN における演算量削減および動作速度向上を目指す。

2　関連研究

CNN の計算量は畳み込み層と呼ばれるフィルタ処理部がそのほとんどを占めており、従来の研究でも様々な手法が提案されてきた。

計算自体の高速化手法としては、固定小数点演算への置換（Darryl ら[2]）、重みおよび特徴マップの演算に用いるバス幅の精度削減（米川・中原[3]）などが存在する。Han らはしきい値未満のパラメータを削除するプルーニング手法を全結合層の重みの各成分に対して用いることで、CNN の認識精度を維持したまま重み配列要素数を約90％削減可能であることを示した[4]。本稿では重みのカーネル単位でのプルーニングを行うことによる畳み込み層における演算量削減手法を提案する。

Improved operating speed of hardware for deep learning by pruning on a per-kernel basis
Yusuke UTSUNO†, Kensyu SETO†
†Tokyo City University. Graduate School of Engineering

3　提案手法

3.1 カーネル単位のプルーニング

畳み込み層における重みとはサイズ k×k の畳み込みフィルタ（カーネル）が M×N 枚格納されている 4 次元配列である（M：出力特徴マップのチャネル数，N：入力特徴マップのチャネル数）。カーネルの枚数(M×N)分だけ畳み込み演算を行うため、特徴マップのサイズを H×W とすると一つの畳込み層の演算量は式(1)の通りとなる[1]。

$$AO_{conv} = M \times N \times W \times H \times k \times k \times 2 + M \times W \times H \quad \cdots (1)$$

本研究では畳み込み演算の回数自体を削減する事により演算量を削減する。しかし、単純に M,N を減らし特徴マップチャネル数を削減してしまうと、CNN の表現力を削いでしまい認識精度に悪影響を及ぼす。そこで、出力への影響度が小さいカーネルを抽出し削除することにより、認識精度と畳み込み演算回数削減の両立を達成する。この手法をカーネル単位のプルーニングといい、畳み込み層ごとに以下の処理を行う。

1. sparsity(カーネルを削除する割合)を設定(ただし、0<sparsity<1)
2. 各カーネルに対してカーネル内重みの絶対値の総和を算出(この総和を cost とする)
3. cost が低いカーネルを順に削除対象に設定、sparsity の割合だけカーネルを削除対象に指定
4. 削除対象になったカーネルのフィルタ内の係数をすべてゼロにする(プルーニング)

プルーニング後はパラメータの改変により CNN の認識精度が低下するため、学習とプルーニングを交互に繰り返す事によって認識精度低下を最小限に抑える。

3.2 重み配列の圧縮

プルーニング後の重み配列は sparsity の割合だけゼロ要素を持つ疎行列となる。FPGA に実装する際に図 1 に示す Coordinate 形式の圧縮処理を施すことによりメモリ量と演算の効率化を図る。FPGA 上で CNN の処理を行う際はプルーニングされた部分の計算をスキップし、残った重みカ

Copyright ©2019 Information Processing Society of Japan.
All Rights Reserved.

ーネルのみ演算処理を行うことにより計算量削減を達成する。

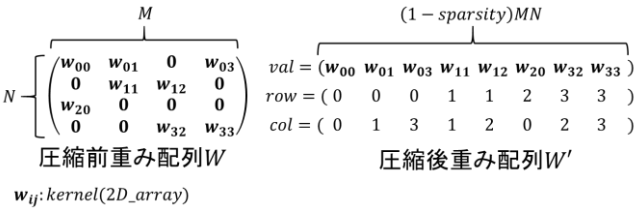

図 1:重み配列の圧縮

また、CNN の学習には Batch Normalization(BN)を用いている。米川・中原は BN の演算をバイアスの演算とマージし、回路リソースを節約する手法を提案した[3]。本稿ではプルーニングした CNN を FPGA に実装する際にこの手法を用いた。

4 実験

本研究では CNN の学習およびプルーニングを Chainer 5.0、CNN の FPGA 実装評価を Xilinx 社の高位合成ツール Vivado HLS 2018.2 を用いて行った。本節では提案手法を適用した場合の CNN の認識精度の評価と高位合成によるハードウェア性能の評価を行う。

4.1 プルーニング手法の認識精度評価

sparsity を様々な値に変え、提案手法によるプルーニングおよび学習を行った時の認識精度を比較した結果を図 2 に示す。プルーニング前に 20epoch 学習させた後、プルーニング→再学習→(繰り返し)のサイクルを 20 回繰り返した。使用したデータセットは Cifar10 である。なお、図中の点線はプルーニングせずに 100epoch 学習させた場合の各ネットワーク構造の認識精度である。

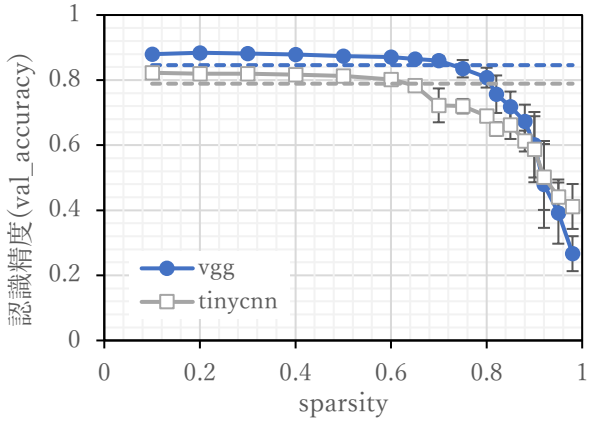

図 2:sparsity による認識精度の変化

今回の実験条件では sparsity>0.7 であれば、最終的な学習結果は認識精度低下が 1.0%未満に収まることが確認できた。これにより提案手法の認識精度維持が可能であることが示された。

4.2 高位合成結果

プルーニングし、重みを圧縮する前後での高位合成後の総動作サイクル数とリソース使用量を比較した。シミュレーションに用いたボードは ZYNQ7 zc702 評価ボード、クロック周期は 10ns、sparsity は 0.4 である。表 1 に高位合成を行った CNN のネットワーク構造を、表 2 に高位合成結果を示す。

表 1:使用したネットワーク構造

layer	layertype	parameter
1	Conv	M=32,N=3,k=3
2	MaxPool	kernel_size=2
3	Conv	M=64,N=32,k=3
4	MaxPool	kernel_size=2
5	FullConnect	

表 2:重み配列圧縮前後での高位合成結果

	圧縮前	圧縮後
サイクル数	13,844,090	8,457,782(61.1%)
BRAM18K	52	52(100%)
DSP48E	3	3(100%)
FF	1703	1590(93.4%)
LUT	4959	4630(93.4%)

重み配列圧縮前に比べて、総動作サイクル数が 38.9%削減できた。sparsity=0.4 なので重み配列サイズは約 60%に削減されたことになるため、この総動作サイクル数の削減は提案手法による効果であると言える。FF(Flip Flop)、LUT(Look Up Table)の使用量が僅かに減少したのは、重み配列圧縮により配列サイズが縮小した事による影響だと考えられる。

5 おわりに

CNN の重み数の削減手法を提案し、高位合成時の総動作サイクル数 38.9%削減を通して CNN 専用回路の動作速度向上を達成した。今後は提案手法の対応ネットワーク構造の拡張や専用回路にループ並列化・パイプライン化等の最適化を施し、さらなる動作速度向上を目指す。

参考文献

[1] Z. Liu, Y. Dou, J. Jiang, J. Xu, S. Li, Y. Zhou, Y. Xu, : Throughput-Optimized FPGA Accelerator for Deep Convolutional Neural Networks : ACM Trans. Reconfigurable Technol. Syst., vol. 10, pp. 17:1-17:23, July 2017

[2] Darryl D. Lin, Sachin S. Talathi,and V. Sreekanth Annapureddy : Fixed Point Quantization of Deep Convolutional Networks : ICLR 2016

[3] H.Yonekawa and H.Nakahara : An On-chip Memory Batch Normalization Free Binarized Convolutional Deep Neural Network on an FPGA : IPDPS 2017 pp.98-105

[4] Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A Horowitz, and William J Dally : EIE: Efficient Inference Engine on Compressed Deep Neural Network. : In ISCA, 2016

Copyright ©2019 Information Processing Society of Japan.
All Rights Reserved.