1A-02

# 手書き数字認識ニューラルネットワークにおける近似乗算器の評価

佐藤　寿倫†
福岡大学†

請園　智玲‡
福岡大学‡

## 1. はじめに

デナードスケーリングが有効だった時代は去り，電源電圧を下げることでの消費電力削減は不可能になった．代替となる選択肢として，近似計算は有望な方法である．意図的に演算精度を犠牲にすることで，消費電力を削減する．我々は以前に近似乗算器を提案したが，演算器単体についての評価に留まっていた[1]．本稿ではその近似乗算器の実用性を確認するために，手書き数字認識アプリケーションを用いて，その評価を行う．画像鮮鋭化処理を用いての評価は，すでに[2]で報告済みである．

## 2. 近似乗算器

乗算操作は，(1)部分積の生成，(2)部分積の加算圧縮，そして(3)最終加算の3ステージから成る．我々が提案している近似乗算器[1]の第2と第3ステージを図1に示す．●が部分積である．第2ステージでは圧縮にORゲートを用いて近似している．二つの部分積 $PP_i$ と $PP_j$ の加算は次式で表現できる．

$$PP_i + PP_j = (PP_i \vee PP_j) + (PP_i \wedge PP_j)$$
$$= p_{i,j} + q_{i,j}$$
$$p_{i,j} = PP_i \vee PP_j$$
$$q_{i,j} = PP_i \wedge PP_j$$

$p_{i,j}$（■）を近似和とみなす．一方，$q_{i,j}$ は近似和の誤差を補正する目的で利用する．上記の演算後の1行分をPおよびQで表す．8行の部分積は4行の P1～P4 に圧縮され，それぞれに対応するQ1～Q4 には OR 操作を施す（◆）．この操作を残り2回繰り返すと，1つの近似積P7と3つの誤差補正ベクトルが得られる．後者について図の操作を行うと一つの誤差補正ベクトル（▲）が得られる．これらを第3ステージで加算すれば最終積（○）が得られる．ここで上位4ビットには正確な加算を施すが，続く7ビットには桁上げをマスク可能な加算器 CMA[1]を利用する．マスクの設定を変えることで，精度可変な近似乗算器を実現している．以上の操作の詳細は[1]を参照されたい．

ほとんどの近似演算器は符号無し数を扱っている．アプリケーションによっては符号付き数を扱う必要がある．配列型乗算器で符号付き数を扱うために，Baugh-Wooley アルゴリズム[3]が利用できる．残念ながら後に述べるように，このアルゴリズムは近似乗算器では効果が無く，頻繁に演算後の符号に誤りを生じる．符号付き整数では二の補数表現を採用することが一般的であるが，OR 演算で加算圧縮を行っていることから，桁上げが失わるためである．このため naïve な方法[4]で符号付き数を扱うことにする．被乗数あるいは乗数が負数の場合には正数に変換し，符号無し数で近似乗算を実施し，必要に応じて結果の符号を操作する．この方法では回路の追加が必要になるが，それによる回路面積と消費電力の増加は小さいと報告されている[4].

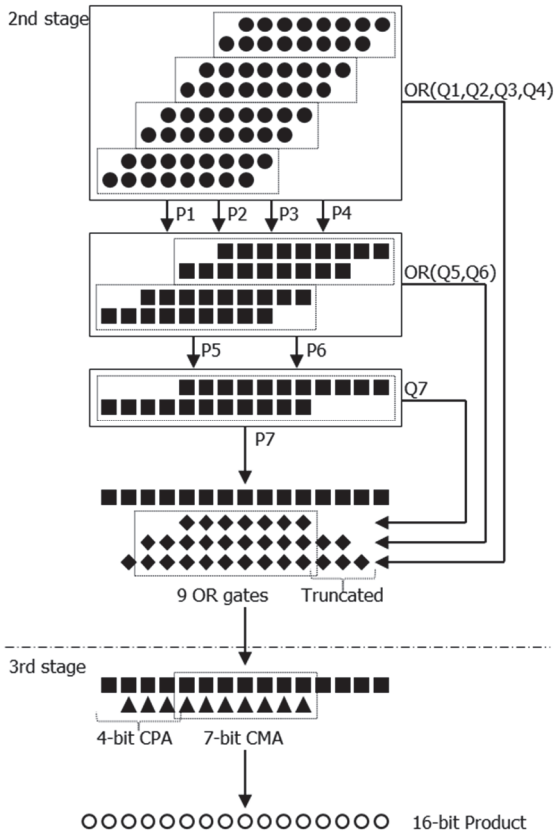


図 1. (8b x 8b)-近似乗算器

Evaluation of an Approximate Multiplier Using CNN-based Handwritten Digit Recognition
†Toshinori Sato, Fukuoka University
‡Tomoaki Ukezono, Fukuoka University

## 3. 評価

まず，MRED（Mean Relative Error Distance）[5]で精度を評価する．ED（Error Distance）は正確値（M）と近似値（M’）との差で，$ED = |M' - M|$である．RED（Relative ED）はEDをMで割った値で，$RED = ED/M = |M' - M|/M$である．MRED（Mean RED）はREDの平均である．

次に，畳み込みニューラルネットであるLeNet-5[6]上の手書き数字認識アプリケーションで評価する．このアプリケーションでは符号付き数を扱う必要がある．図2に示すLeNet-5をdarknet[7]で構築する．活性化関数にはtanhを用いる．入力は$32 \times 32$ピクセルの画像である．単精度の浮動小数点演算で学習する．学習用データはMNIST[8]として提供される 60,000 枚の画像である．推論時でのみ近似乗算を利用する．畳み込み層と全結合層での乗算でのみ，8 ビット固定小数点演算を採用し，加えて近似計算を利用する．LeNet-5 では 341K 回の積和演算が実施され[9]，畳み込み演算が全処理の 90%を占めている[10]．推論時のテストデータはやはり MNIST として提供されている10,000枚の画像である．認識正解率を演算精度の評価尺度として採用する．

表1にMREDと認識精度をまとめる．2行目はBaugh-Wooley アルゴリズム[3]を採用し桁上げを全くマスクしない構成時の結果であり，3〜10 行目は naïve な方法[4]を採用し CMA でのマスク設定を変えた時の結果である．

2〜3 列目が，$2^8 \times 2^8 = 65{,}536$通りに対して，それぞれ符号無し数（0〜255）[1]と符号付き数（-128〜127）を扱う場合での MRED である．後者は前者と比較して最大で1.51倍となっている．後者は実質的には符号無し数を扱っており，且つ被乗数と乗数は7ビット相当であるにも関わらず，MRED が悪化している点は興味深く，近似乗算器にとって符号付き数の扱いが弱点であることが確認できる．

表1の4列目は手書き数字認識精度である．認識精度は，単精度浮動小数点演算時に 97.37%で，固定小数点での正確な演算時に 96.64%であった．これらと比較すると，Baugh-Wooley アルゴリズム採用時の精度は全く実用的では無いが，naïve な方式採用時のそれらは十分実用的であると思われる．後者では実質的に符号無し数を扱っていることから現状の近似乗算器は符号付き数の扱いが苦手であることが再確認出来，この解決が大きな課題である．近似乗算器を最高精度の構成（$\overline{mask} = 1111111$）から最低精度の構成（$\overline{mask} = 0000000$）に変えることで，消費電力を19.8%削減可能である[1]．例えば，バッテリー

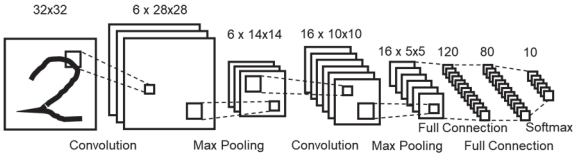


図 2. LeNet-5 ニューラルネットワーク

表 1. 評価結果

| $\overline{mask}$ | MRED (%) | | Recognition Rate (%) |
|---|---|---|---|
| | unsigned[1] | signed | |
| B & W | | 287 | 10.30 |
| 1111111 | 2.51 | 2.32 | 96.37 |
| 1111110 | 2.66 | 2.72 | 95.81 |
| 1111100 | 2.97 | 3.54 | 95.38 |
| 1111000 | 3.56 | 5.03 | 95.34 |
| 1110000 | 4.61 | 6.93 | 95.34 |
| 1100000 | 6.06 | 9.17 | 95.34 |
| 1000000 | 7.92 | 11.5 | 95.34 |
| 0000000 | 10.1 | 13.4 | 95.34 |

残量が少なくなった際に手書き数字認識精度が1%低下することを許容すれば，そのデバイスの利用可能時間を25%延長可能なわけである．

## 4. おわりに

手書き数字認識アプリケーションを用いて，我々が提案している近似乗算器を評価した．当該アプリケーションは符号付き数を扱う必要があるが，Baugh-Wooley アルゴリズムは有効では無かった．Naïve な方式を採用すれば実用的な認識精度が得られることが確認され，限定的ではあるが，提案している近似乗算器の有用性が確認できた．符号付き数を扱うことが可能な近似乗算器を検討することが，喫緊の課題である．

## 謝辞

本研究はJSPS科研費 JP17K00088の助成を受け，その一部は福岡大学研究推進部の研究経費（課題番号 175007，177005）によるものである．ヨウ ドウキンは有意義な議論に参加した．

## 参考文献

[1] 井上，http://id.nii.ac.jp/1001/00183856/
[2] H. Baba, doi:10.1109/ISVLSI.2018.00109
[3] C. Baugh, doi:10.1109/T-C.1973.223648
[4] S. Hashemi, doi:10.1109/ICCAD.2015.7372600
[5] C. Liu, doi:10.7873/DATE.2014.108
[6] Y. LeCun, doi:10.1109/5.726791
[7] J. Redmon, https://pjreddie.com/darknet/
[8] Y. LeCun, http://yann.lecun.com/exdb /mnist/
[9] V. Sze, doi:10.1109/JPROC.2017.2761740
[10] J. Emer, ISCA Tutorial, 2017.