Binary-decomposed DCNN におけるハイパーパラメータの最適化に関する研究

TP18004 近藤良太　　指導教授：山下隆義

1. はじめに

Deep Convolutional Neural Network (DCNN) の処理速度の高速化とモデル圧縮を再学習なしに実現する手法として，Binary-decomposed DCNN (B-dCNN)[1] が提案されている．B-DCNN は，Quantization sub-layer による特徴マップの量子化とベクトル分解法による重みの分解により，近似計算に落とし込むことで高速化を実現した．一方で，Quantization sub-layer の量子化ビット数と，ベクトル分解の基底数の最適値はハイパーパラメータとして予め人が設定する必要がある．この組み合わせは，8 層の AlexNet のモデルでは 7^{16} 通りあるため，予め設定することは困難である．そこで本研究では，ハイパーパラメータ探索を用いて量子化ビット数と基底数を同時に自動最適化する手法を提案する．

2. Binary-decomposed DCNN

B-DCNN は処理時間の削減とモデル圧縮を同時に実現するために，ベクトル分解を用いた重みの分解と Quantization sub-layer による特徴マップの量子化を行う．

ベクトル分解は，重みベクトルを二値基底行列 $\mathbf{M}$ とスケール係数ベクトル $\mathbf{c}$ に分解する．ハイパーパラメータである基底数 B が大きい場合，重みの近似性能が高く，識別精度が向上する．しかしながら，パラメータ数が増加する．一方で，基底数 B が小さい場合，パラメータ数が削減されるが重みの近似性能が低下するため，識別精度が低下する．

Quantization sub-layer は，特徴マップ $\mathbf{h}$ の最小値が 0 になるように $\mathbf{h}$ をシフトすることで負値を取り除き，量子化して量子化ビット数 Q の二値にする．ハイパーパラメータである量子化ビット数 Q が大きい場合，特徴マップの表現力が高く，識別精度が向上する．しかし，処理速度は低下する．一方，量子化ビット数 Q が小さい場合，処理速度が高速になるが特徴マップの表現力が乏しいため，識別精度が低下する．

3. 提案手法

B-DCNN は，各層の量子化ビット数，基底数を人が予め設定することで識別精度，モデルサイズ，処理速度を調整する．一般に量子化ビット数と基底数は，実用的な識別精度，処理速度の観点から 2 から 8 の範囲で値を決定している．これらのハイパーパラメータの組み合わせが膨大なため，従来は全層で同じ値としている．本研究では，識別精度，モデルサイズ，処理速度の 3 つを考慮した目的関数を設計し，各層の量子化ビット数と基底数の最適値を導出する．

3.1 目的関数の設計

本研究では，識別精度，モデルサイズ，処理速度の 3 つを考慮して目的関数 C を設計する．量子化ビット数と基底数を最適化する際の目的関数 C を式 (1) のように定義する．

$$C = \alpha|o_p - r_p| + \beta\frac{W_b}{W_f} + \gamma\frac{q_b}{q_f} \tag{1}$$

ここで，o_p は目標精度，r_p は実精度，W_f は重み分解前のモデルサイズ，W_b は重み分解後のモデルサイズ，q_f は通常のネットワークの計算量，q_b は B-DCNN 適用後のネットワークの計算量，α , β , γ は係数である．第 1 項は，目標精度に対しての実精度のズレを表している．第 2 項は，分解後のモデルサイズの圧縮割合である．重み分解前のモデルサイズ W_f と重み分解後のモデルサイズ W_b は，式 (2) により求める．

$$\begin{aligned} W_b &= \frac{\sum_{l=1}^{N}\left(1w_{\mathbf{M}}^{l} + 64w_{\mathbf{c}}^{l}\right)}{8\cdot 1024^3} \\ W_f &= \frac{\sum_{l=1}^{N}\left(32w^{l}\right)}{8\cdot 1024^3} \end{aligned} \tag{2}$$

ここで，N は層の数，$w_{\mathbf{M}}^{l}$ は二値基底行列 $\mathbf{M}$ のパラメータ数，$w_{\mathbf{c}}^{l}$ は，スケール係数ベクトル $\mathbf{c}$ のパラメータ数，w^{l} は重み $\mathbf{w}$ のパラメータ数を示す．モデルサイズを計算する際に二値基底行列 $\mathbf{M}$ は 1 ビット，スケール係数ベクトル $\mathbf{c}$ は 64 ビット，重み $\mathbf{w}$ は 32 ビットであるため，それぞれデータ型を考慮して算出する．このとき，データサイズを $8\cdot 1024^3$ の定数を用いて MB 単位に変換している．第 3 項は，計算量の削減度合いである．通常のモデルの計算量 q_f と B-DCNN 適用後のモデルの計算量 q_b は，式 (3) により求める．

$$\begin{aligned} q_b &= \sum_{l=1}^{N} q^l z^l \\ q_f &= 10\sum_{l=1}^{N} z^l \end{aligned} \tag{3}$$

ここで，q^l は l 層目における量子化ビット数，z^l は l 層目における特徴マップ数，特徴ベクトル数の総乗である．量子化ビット数と特徴マップ数又は特徴ベクトル数の積により，計算量の指標を算出する．このとき，通常のモデルの計算量は量子化ビット数を 10 と仮定して計算している．

各項に掛かる係数は，識別精度，モデルサイズ，処理速度の優先度を表すハイパーパラメータである．式 (1) を最小化する量子化ビット数と基底数の組み合わせが対象のネットワークにおける最適値となる．

3.2 パラメータの最適化

全層の基底数，量子化ビット数を最適化するため，ハイパーパラメータ探索を行う．ハイパーパラメータ探索による最適化は，1 つの目的関数を設定し，それを最小化する組み合わせを探索する．ハイパーパラメータ探索を行うまでの流れを以下に示す．

Step1 ハイパーパラメータ探索に用いる最適化用データセットを作成する．クラスの偏りを防ぐために，全体で 1000 枚程度を確保する．

Step2 Step1 で作成したデータセットを用いて通常のネットワークで推論を行い，精度を算出する．得られた精度を最適化時の目標精度 o_p とする．

Step3 目的関数を最小化するハイパーパラメータ探索を行い，最適値の組み合わせを算出する．

ハイパーパラメータ探索の最適化手法として Successive Halving Algorithm (SHA)[2] を用いる．SHA は，複数回のランダムサーチを試行して目的関数の収束具合の良いハイパーパラメータ以外を枝刈りする．これにより，効率的な探索を可能にする．ハイパーパラメータ探索の手順を図 1 に示す．まず，SHA によって選択された量子化ビット数 Q と基底数 B を B-DCNN に設定する．次に，Step1 で作成した最適化用データセットを用いて推論を行い，識別精度 r_p を算出する．推論時にモデルサイズ W_b と計算量 q_b を取得する．算出した識別精度 r_p，モデルサイズ W_b，計算量 q_b を目的関数 C に代入して評価する．評価結果が以前の評価値より小さい場合に，これらのハイパーパラメータを保存する．なお，選択可能なハイパーパラメータの範囲は，量子化ビット数が [2, 3, 4, 5, 6, 7, 8]，基底数が [2, 3, 4, 5, 6, 7, 8] である．本研究では，探索の試行上限を 1,000 回として最適化処理を行う．

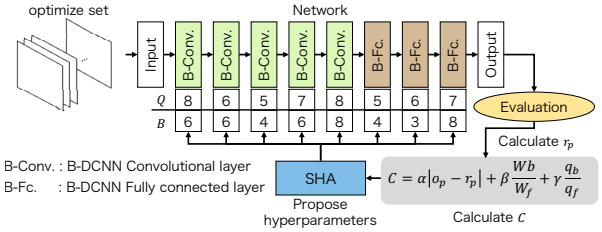

図 1：ハイパーパラメータ探索の手順

表 1：ImageNet データセットによる各モデルとの比較結果

モデル	ハイパーパラメータ	c1	c2	c3	c4	c5	f1	f2	f3	識別精度 [%]	モデル圧縮率 [%]	処理速度 [sec]
1	量子化ビット数	6	6	6	6	6	6	6	6	54.00	81.05	0.700
	基底数	6	6	6	6	6	6	6	6			
2	量子化ビット数	8	8	8	8	8	8	8	8	**55.00**	74.43	0.765
	基底数	8	8	8	8	8	8	8	8			
3	量子化ビット数	6	5	6	6	5	5	7	8	54.10	**87.20**	**0.680**
	基底数	6	7	8	8	4	3	5	8			

表 2：COCO データセットによる各モデルとの比較結果

モデル	ハイパーパラメータ	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	mAP	モデル圧縮率 [%]	処理速度 [sec]
1	量子化ビット数	6	6	6	6	6	6	6	6	6	6	6	6	6	20.54	80.75	**0.471**
	基底数	6	6	6	6	6	6	6	6	6	6	6	6	6			
2	量子化ビット数	8	8	8	8	8	8	8	8	8	8	8	8	8	**27.30**	74.33	0.516
	基底数	8	8	8	8	8	8	8	8	8	8	8	8	8			
3	量子化ビット数	7	6	6	8	6	8	7	7	8	6	4	7	8	23.02	**83.25**	0.485
	基底数	7	6	8	5	7	4	5	6	5	8	5	7	5			

4. 評価実験

提案手法の有効性を確認するために画像分類と物体検出の 2 つのタスクによる評価実験を行う．

4.1 実験概要

画像分類タスクには，ImageNet データセットを用いて Top-1 accuracy により評価する．また，物体検出タスクには COCO データセットを用いて mean Average Precision (mAP) により評価を行う．

タスク毎にモデルを複数用意して識別精度，モデル圧縮率，処理速度を比較する．モデル 1 は，従来手法として量子化ビット数と基底数を 6 で固定した B-DCNN である．モデル 2 は，モデル 1 と同様に量子化ビット数と基底数を 8 で固定した B-DCNN である．モデル 3 は提案手法により最適化した量子化ビット数と基底数を設定した B-DCNN である．目的関数 C の係数は，識別精度の最適化に重点を置き，パラメータ圧縮率の最適化，計算コストの最適化の順で重み付けを行う．そのため，本実験では $\alpha = 10, \beta = 2, \gamma = 1$ とする．

ImageNet データセットを用いた実験では，48,000 枚を検証データ，2,000 枚を最適化処理用データとして用いる．ネットワークモデルは，8 層の学習済みの AlexNet を用いる．このモデルサイズは約 233MB である．COCO データセットを用いた実験では，4,000 枚を検証データ，1,000 枚を最適化処理用データとして用いる．ネットワークモデルは，13 層の学習済みの YOLOv3-tiny を用いる．YOLOv3-tiny は YOLOv3 の畳み込み層を削減した軽量なモデルである．このモデルサイズは 34MB である．

4.2 画像分類タスク：ImageNet

ImageNet データセットを用いた実験結果を表 1 に示す．また，AlexNet における各層のパラメータサイズを図 2 に示す．提案手法はモデル 1 と比較して識別精度が 0.1 ポイント，モデル圧縮率が 6.15 ポイント向上し，処理速度は 0.02[sec] 高速化できている．モデル 2 との比較では，識別精度は 0.9 ポイント低下しているが，モデル圧縮率が 12.47 ポイント向上して処理速度が 0.085[sec] 高速化している．ImageNet は検証データが非常に多く，最適化処理用データの割合が少ない．この条件においても提案手法では，識別精度が向上しており，少ないデータ数でも最適化が可能であることがわかる．

選択された量子化ビット数と基底数に着目すると，畳み込み層の 2,3,4 層目 (c2,c3,c4) と最終層 (f3) が大きな基底数になっている．これは畳み込み層の情報量が多く，小さな基底数を選択した場合，精度に影響するためと考えられる．最終層は大きい値が選択されているため，クラス分類を行う層では高精度な近似が必要であることがわかる．AlexNet は，図 2 のように全結合層 1 層目のパラメータサイズが全体のモデルサイズの 60%を占めている．全結合層 1 層目 (f1) の基底数は 3 になっており，パラメータサイズ

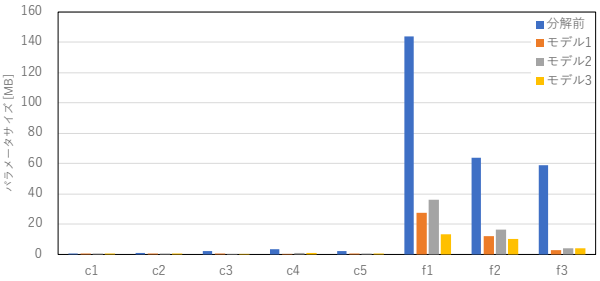

図 2：AlexNet における各層のパラメータサイズ

の大きい層の基底数を積極的に小さくしていることがわかる．このことから，各層のパラメータサイズが全体に占める割合を考慮しつつ最適化できていると言える．

4.3 物体検出タスク：COCO

COCO データセットを用いた実験結果を表 2 に示す．提案手法はモデル 1 と比較して処理速度は 0.014[sec] 低下したが，mAP が 2.48 ポイント，モデル圧縮率が 2.4 ポイント向上している．モデル 2 との比較では，mAP は 4.28 ポイント低下しているが，モデル圧縮率が 8.92 ポイント向上して処理速度が 0.031[sec] 高速化している．

選択された量子化ビット数と基底数に着目すると，全体で高い量子化ビット数が選択されている．物体検出タスクでは複数の物体を 1 枚の画像から検出する．そのため，画像分類タスクと比較して量子化ビット数が精度へ与える影響が大きく，特徴マップの量子化誤差を抑えるために高い値が選択されたと考えられる．

5. おわりに

本研究では，ハイパーパラメータ探索を用いて量子化ビット数と基底数を同時に自動最適化する手法を提案した．識別精度，モデルサイズ，処理速度の 3 つを考慮した目的関数を設計することで精度低下を抑制しつつ，モデル圧縮率と処理速度の向上を実現した．今後は最適化アルゴリズムの改良により，探索時の試行回数の削減や高精度化を図る．また，重み分解の基底数を事前決定する方法を検討する．

参考文献

[1] R. Kamiya, *et al.*, “Binary-decomposed DCNN for accelerating computation and compressing model without retraining”, ICCV Workshop, 2017.

[2] K. Jamieson, *et al.*, “Non-stochastic Best Arm Identification and Hyperparameter Optimization”, AISTATS, 2015.

研究業績

[1] 近藤良太 等, “Binary-decomposed DCNN におけるハイパーパラメータの自動最適化”, ビジョン技術の実利用ワークショップ, 2019.

(他 学会発表 2 件)