Binary-decomposed DCNN におけるハイパーパラメータの自動最適化

○近藤 良太†，平川翼†，山下隆義†，藤吉弘亘†
†: 中部大学
{kndroa1996@mprg.cs, hirakawa@mprg.cs, takayoshi@isc, fujiyoshi@isc}.chubu.ac.jp

概要： Binary-decomposed DCNN は，特徴マップの量子化と重みの分解により，論理演算を用いた近似計算にすることで，推論の高速化とモデル圧縮を再学習なしに行う手法である．特徴マップは Quantization sub-layer を用いて量子化し，重みはベクトル分解を適用して少量の実数と多数の二値ベクトルに変換する．Quantization sub-layer は量子化ビット数を設定して量子化精度を調整し，ベクトル分解は基底数を設定して近似精度を調整する．これらのハイパーパラメータは各層において任意に設定する必要がある．ネットワークの各層において最適値を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．提案手法は識別精度，モデルサイズ，処理速度を考慮した目的関数を定義し，ハイパーパラメータ探索により量子化ビット数と基底数を同時に最適化する．ImageNet データセットを用いた評価実験では，提案手法が従来手法よりモデル圧縮率が 6.15%，処理速度が 0.02[sec]優れた結果を示した．また，COCO データセットを用いた評価実験では，物体検出タスクにおいても提案手法が有効であることを確認した．

Binary-decomposed DCNN，量子化，重み分解

1. はじめに

Convolutional Neural Network (CNN)は，画像認識[12][15][16]やセマンティックセグメンテーション[17][18]，物体検出[19][20]のタスクにおいて高い性能を達成している．CNN は VGGNet[13]や Deep Residual Net (ResNet)[14]等の深い層から構成されるネットワークが高性能であり，派生モデルが多数提案されている[28][29]．しかしながら，CNN には畳み込み層や全結合層に内積計算が多用されるため，ネットワークが多層化されるにつれてパラメータ数が増加し，多くの計算リソースが必要となる．そのため，組み込み機器やモバイル端末等の限られた計算リソースにおいて CNN の推論を行うには，CNN の識別計算の高速化とモデルサイズの圧縮が必要である．

これらの問題を解決するために，識別計算の高速化とモデルサイズの圧縮を同時に実現する方法として，重みと特徴マップを量子化または二値化，三値化して論理演算に置き換える手法が提案されている[3][10][11]．また，これらの処理を再学習せずに実現する Binary-decomposed DCNN (B-DCNN)[1] や Composite Binary Decomposition Networks (CBDNet)[9]が提案されている．B-DCNN は，精度の低下を最小限に抑えながら，CNN の識別計算の高速化とモデルの圧縮を同時に行う．B-DCNN は，Quantization sub-layer を用いて特徴マップを量子化し，重みにベクトル分解を適用して少量の実数と多数の二値ベクトルに変換する．Quantization sub-layer は量子化ビット数を設定して量子化精度を調整し，ベクトル分解は基底数を設定して近似精度を調整する．これらのハイパーパラメータは対象の層毎に任意で決定する．しかし，ネットワークの各層において最適値を任意で選択することは，組み合わせが無数に存在するため困難である．

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最適化する手法を提案する．目的関数を設計し，最小化問題として定義することで，ハイパーパラメータの最適化手法を適用可能にする．最適化手法により量子化ビット数と基底数の組み合わせを枝刈りしつつ効率的に選出する．

2. 関連研究

DCNN の識別計算の高速化，パラメータの削減方法としては，畳み込み層のカーネルを個別に分解する方法と，カーネル全てを同時に分解する方法がある．前者は MobileNet[26]や ShuffleNet[27]などであり，本研究では後者を対象とする．全てのカーネルを同時に分解する方法は，再学習ありと再学習なしの手法に分類できる．これらの関連研究について，以下に述べる．

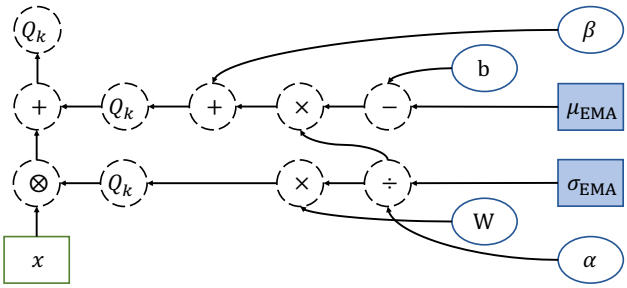

図 1:バッチ正規化層の畳み込み併合の流れ

2.1. 再学習ありの手法

CNN に量子化を適用する方法には，通常のネットワークを量子化ネットワークに変換する Fully Quantized Network (FQN)[2]がある．FQN は，物体検出などの複雑なタスクに低ビット幅の量子化ネットワークを適用する手法である．一般のネットワーク量子化方法は量子化アルゴリズムの適用，Fine-tuning, 量子化モデルの展開の 3 つの流れで行う．FQN では，量子化時の精度低下が発生する段階を調査し，Fine-tuning に重点を置いている．

量子化方法には，線形量子化を用いて重みと特徴マップを量子化する．重みの量子化は出力チャンネルに毎にすることで，精度低下を抑制している．また，特徴マップ量子化後のファインチューニング時に，バッチ正規化層の BN 統計量が精度低下を招くとして値を置き換える処理を行う．BN 統計量μ, σを運動量Mの指数移動平均(EMA)μ_{EMA}, σ_{EMA}に置き換える．置き換え後のバッチ正規化層は次層の畳み込みに併合され，計算量を削減する．バッチ正規化層の畳み込み併合の流れを図 1 に示す．図 1 のように併合処理を行うことで，外れ値による量子化範囲の拡大を防ぎ，分解能の低下を抑える効果がある．これらの処理を加えることで，FQN は 4 ビット幅を適用した量子化ネットワークを実現している．

2.2. 再学習なしの手法

B-DCNN は，再学習なしで重みと特徴マップを二値と少量の実数により近似する．識別計算の高速化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．

まず，あらかじめオフラインで各層の重みをベクトル分解法により，二値と少量の実数に変換する．ベクトル分解は，重みベクトルを基底数Bに基づく二値基底行列$\mathbf{M}$とスケール係数ベクトル$\mathbf{c}$に分解する．このとき，ベクトル分解法として Exhaustive アルゴリズム[5]を使用する．Exhaustive アルゴリズムは，同様のベクトル分解手法である Greedy アルゴリズム[4]より高精度な近似が可能である．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による特徴マップの量子化では，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．Quantization sub-layer の処理の流れを図 2 に示す．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．しかしながら，処理速度は低下

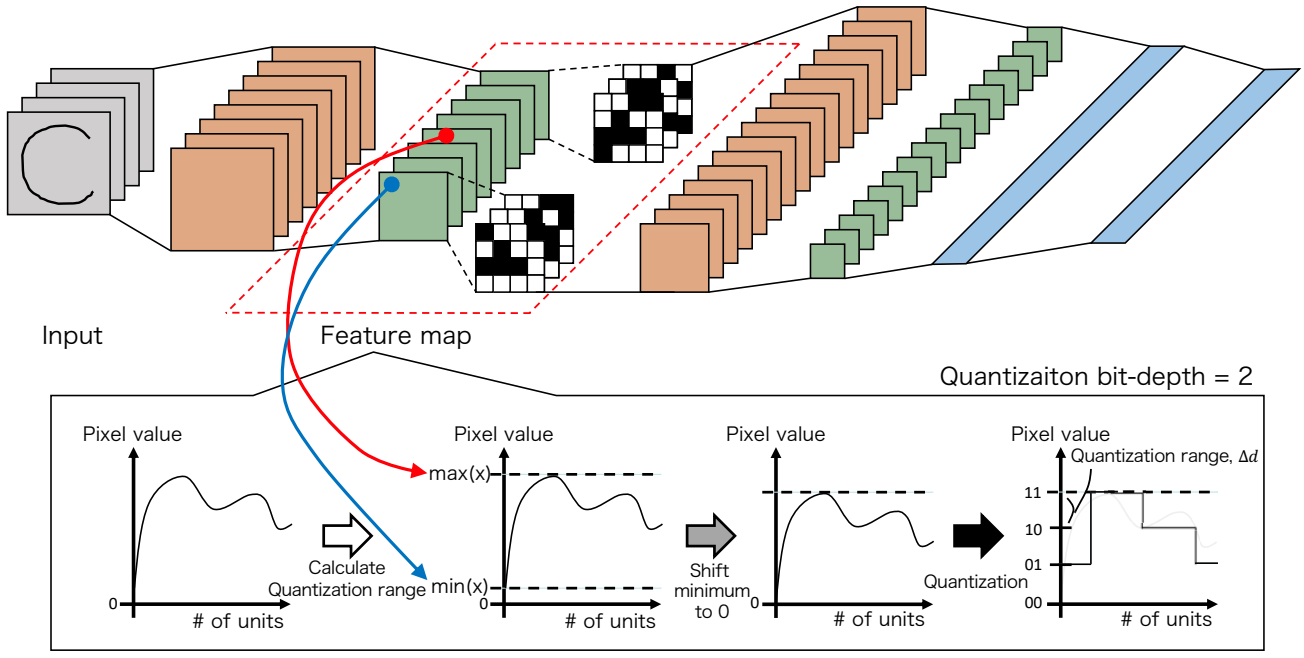

図 2：Quantization sub-layer による特徴マップの量子化

する．一方で，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，任意で各層の量子化ビット数，基底数を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．従来は，これらのハイパーパラメータを全層固定としている．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することで，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

目的関数Cは識別精度，モデルサイズ，処理速度の 3 つを考慮して設計する．量子化ビット数と基底数を最適化する際に使用する目的関数Cを式(1)に示す．

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

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を算出する．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)により計算できる．

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

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．第 3 項は，計算量の削減度合いを算出する．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)により計算できる．

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

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または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，

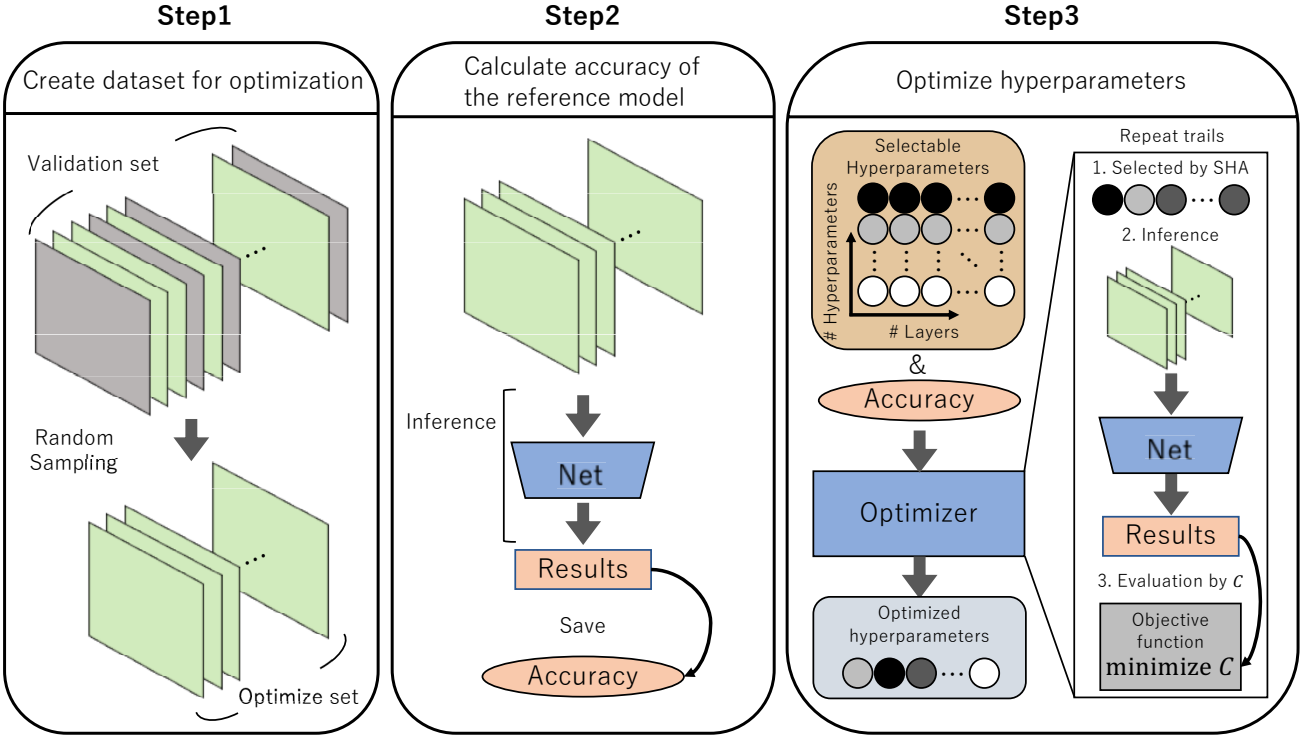

図 3：ハイパーパラメータ探索の手順

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設定する．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. 最適化方法

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最適化の手順を図 3 に示す．また，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　ハイパーパタメータ探索に用いる最適化用データセットを作成

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，目標精度を算出

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

3.2.1. データセットの作成

ハイパーパラメータ探索を実行するために，最適化時に使用するデータセットを作成する必要がある．探索時に対象のデータセット全てを使用すると，推論に膨大な時間を要する．そのため，対象のデータセットからランダムサンプリングを行い，少量のデータセットを作成する．このとき，各クラスの偏りを防ぐために最低でも 1,000 枚程度を確保する．

3.2.2. 対象ネットワークの精度算出

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を使用した際の認識精度を算出する．通常のネットワークとは，B-DCNN を適用する前のネットワークモデルを示す．通常のネットワークから得られた認識精度を最適化時の目標とする．

3.2.3. ハイパーパラメータ探索の実行

ハイパーパラメータ探索に利用する最適化手法として Successive Halving Algorithm (SHA)[21]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．各層における量子化ビット数と基底数の決定までの流れを図 4 に示す．

まず，SHA によって選択された量子化ビット数と基底数を 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，識別精度を算出する．推論時にモデルサイズと計算量を取得する．算出した識別精度，モデルサイズ，計算量を目的関数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，量子化ビット数と基底数を保存する．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]とする．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．

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，CIFAR100[24] と ImageNet[22]データセットを用いる．また，物体検出タスクには COCO[25]データセットを用いる．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．画像分類タスクに

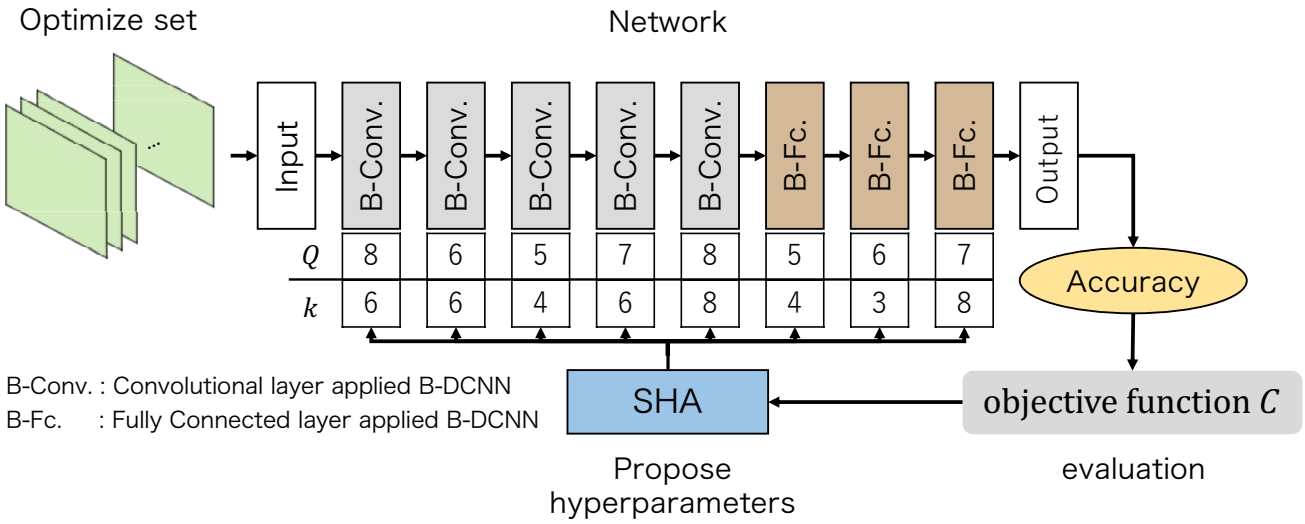

図 4：各層における量子化ビット数と基底数の決定

表 1:CIFAR100 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	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	6	60.00	**79.90**	0.319
	基底数	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	**65.50**	73.20	0.390
	基底数	8	8	8	8	8	8	8	8	8			
3	量子化ビット数	8	7	6	4	7	4	4	4	7	64.25	79.88	**0.295**
	基底数	8	5	4	6	8	6	5	3	8			

表 2:ImageNet データセットにおけ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**	74.73	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 3:COCO データセットにおけ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			

は Top-1 accuracy，物体検出タスクには，mean Average Precision (mAP)を精度の評価方法とする．

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 を使用する．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 を使用する．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．本実験では，$\alpha = 10, \beta = 2, \gamma = 1$として設定した．

CIFAR100 データセットを用いた実験は，50,000 枚を訓練データ，9,000 枚を検証データ，1,000 枚を最適化処理用データとする．ネットワークモデルは，バッチサイズ 64，エポック数 400 で学習した軽量な AlexNet-like を用いる．AlexNet-like は AlexNet をベースに畳み込み層のフィルタサイズやチャンネル数を削減し，バッチ正規化層を加えたモデルである．通常のモデルサイズは約 4MB である．ImageNet データセットを用いた実験は，48,000 枚を検証データ，2,000 枚を最適化処理用データとする．ネットワークモデルは，学習済みのリファレンスモデルとして提供されている AlexNet を用いる．通常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は，4,000 枚を検証データ，1,000 枚を最適化処理用データとする．ネットワークモデルは，学習済みのリファレンスモデルとして提供されている PyTorch 実装の 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．本モデルサイズは 34MB である．

4.2. CIFAR100 による評価結果

CIFAR100 データセットを用いた実験結果を表 1 に示す．ここで，c は畳み込み層，f は全結合層を表している．提案手法はモデル 1 と比較してモデル圧縮率が 0.02%低下したが，識別精度は 4.25%向上し，処理速度は 0.024[sec]高速化している．モデル 2 との比較では，識別精度は 1.25%低下してい

るが，モデル圧縮率が 6.68%向上して処理速度が 0.095[sec]高速化している．このことから，精度低下を抑制しつつモデルサイズと処理速度を考慮した量子化ビット数，基底数が選択されていることがわかる．

選択された量子化ビット数と基底数を見ると，畳み込み層の 1,5 層目と全結合層の最終層で 8,7 と大きい値が割り当てられている．この傾向から，畳み込み層の 1 層目や全結合層へ切り替わる前の層，最終層では高精度な近似が必要とされることがわかる．また，それら以外の層では量子化ビット数と基底数の値が比較的小さいことから，モデルサイズの削減と処理速度の向上に貢献したと考えられる．

4.3. ImageNet による評価結果

ImageNet 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 と比較して識別精度が 0.1%，モデル圧縮率が 6.15%向上し，処理速度は 0.02[sec]高速化している．モデル 2 との比較では，識別精度は 0.9%低下しているが，モデル圧縮率が 12.47%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いが，識別精度が向上しており，提案手法の汎用性が高いことがわかる．

選択された量子化ビット数と基底数を見ると，CIFAR100 を学習した AlexNet-like とは傾向が異なる．畳み込み層の 2,3,4 層目と最終層が 8,7,6 と大きい値を示している. 畳み込み層の情報量が多く，5,4 の小さい値を選択した場合，精度に影響すると考えられる．最終層は CIFAR100 のモデルと同様に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，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．ここで，基底数 3 が選択されており，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て最適化できていると言える．

4.4. COCO による評価結果

COCO データセットを用いた実験結果を表 3 に示す. また，検出の可視化例を図 5 に示す. 図 5 より提案手法は検出されない物体とされる物体が変化していることがわかる．

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%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%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を見ると，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精度への影響が大きく，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ると考えられる．

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．選択された量子化ビット数と基底数から，クラス分類を行う最終層は画像分類に高い近似精度を必要とすることを示した．また、評価指標の違いにより，物体検出には全層で高い量子化精度が必要であることを示した．

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，再学習によるモデルの更なる高精度化を検討する．

参考文献

[1] R. Kamiya, T. Yamashita, M. Ambai, I. Sato, Y. Yamauchi and H. Fujiyoshi, “Binary-decomposed DCNN for accelerating computation and compressing model without retraining”, Workshop on International Conference on Computer Vision, 2017.

[2] R. Li, F. Liang, H. Qin, Y. Wang, R. Fan and J. Ya, “Fully Quantized Network for Object Detection”, The IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[3] M. Ambai and I. Sato, “SPADE: Scalar Product Accelerator by Integer Decomposition for Object Detection”, European Conference on Computer Vision, 2014.

[4] S. Hare, A. Saffari and Philip H S. Torr, “Efficient Online Structured Output Learning for Keypoint-Based Object Tracking”, The Proceedings IEEE Conference of Computer Vision and Pattern Recognition, 2012.

[5] Y. Yamauchi, M. Ambai, I. Sato, Y. Yoshida and H. Fujiyoshi, “Distance Computation Between Binary Code and Real Vector for Efficient Keypoint Matching”, Information Processing Society of Japan Transactions on Computer Vision and Applications, 2013.

[6] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, Andrew G. Howard, H. Adam and D. Kalenichenko, “Quantization and Training of Neural Networks for Efficient Integer-

図 5：COCO データセットにおける検出結果の可視化例

Arithmetic-Only Inference", arXiv preprint arXiv:1712.05877, 2017.

[7] S. Khoram and J. Li, "Adaptive Quantization of Neural Networks", International Conference on Learning Representations, 2018.

[8] Y. Zhou, Seyed-Mohsen. Moosavi-Dezfooli, Ngai-Man. Cheung and P. Frossard, "Adaptive Quantization for Deep Neural Networks", Association for the Advancement of Artificial Intelligence, 2018.

[9] Y. Qiaoben, Z. Wang, J. Li, Y. Dong, Yu-Gang. Jiang and J. Zhu, "Composite Binary Decomposition Networks", Association for the Advancement of Artificial Intelligence, 2019.

[10] R. Mohammad, O. Vicente, R. Joseph and F. Ali, "XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks", European Conference on Computer Vision, 2016.

[11] Forrest N. Candela, Matthew W. Moskewicz, K. Ashram, S. Han, William J. Dally and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size", arXiv preprint arXiv:1602.02830, 2016.

[12] A. Krizhevsky, S. Ilya and Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", Advances in Neural Information Processing Systems, Curran Associates, Inc., 2012.

[13] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", Proceedings of the International Conference on Learning Representations, 2015.

[14] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", The IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[15] D. Han, J. Kim and J. Kim, "Deep Pyramidal Residual Networks", The IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[16] H. Jie, S. Li and S. Gang, "Squeeze-and-Excitation Networks", The IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[17] B. Vijay, K. Alex and C. Roberto, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation", arXiv preprint arXiv:1511.00561, 2015.

[18] Z. Hengshuang, S. Jianping, Q. Xiaojuan, W. Xiaogang and J. Jiaya, "Pyramid Scene Parsing Network", The IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[19] S. Ren, K. He and R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", Neural Information Processing Systems, 2015.

[20] Z. Hengshuang, S. Jianping, Q. Xiaojuan, W. Xiaogang and J. Jiaya, "M2Det: A Single-Shot Object Detector Based on Multi-Level Feature Pyramid Network", Association for the Advancement of Artificial Intelligence, 2019.

[21] K. Jamieson and A. Talwalker, "Non-stochastic Best Arm Identification and Hyperparameter Optimization", The

International Conference on Artificial Intelligence and Statistics, 2015.
[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, Alexander C. Berg and Li. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge”, International Journal of Computer Vision, 2015.
[23] C. Marius, O. Mohamed, R. Sebastian, R. Timo, E. Markus, B. Rodrigo, F. Uwe, R. Stefan and S. Bernt, “The Cityscapes Dataset for Semantic Urban Scene Understanding”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.
[24] A. Krizhevsky, V. Nair and Geoffrey E. Hinton, “Learning Multiple Layers of Features from Tiny Images”, Citeseer, 2009.
[25] Tsung-Yi. Lin and M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. Lawrence Zitnick and P. Dollár, “Microsoft COCO: Common Objects in Context”, European Conference on Computer Vision, 2014.
[26] Andrew G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”, arXiv preprint arXiv: 1704.04861., 2017.
[27] X. Zhang, X. Zhou, M. Lin and J. Sun, “ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices”, The Proceedings IEEE Conference of Computer Vision and Pattern Recognition, 2018.
[28] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, Cheng-Yang. Fu and Alexander C. Berg, “SSD: Single Shot MultiBox Detector”, European Conference on Computer Vision, 2017.
[29] S. Xie, R. Girshick, P. Dollár, Z. Tu and K. He, “Aggregated Residual Transformations for Deep Neural Networks”, The Proceedings IEEE Conference of Computer Vision and Pattern Recognition, 2017.

近藤良太：現在中部大学大学院修士課程在学中，DCNN の高速化とモデル圧縮に関する研究に従事.

平川翼: 2013 年広島大学大学院博士課程前期終了，2014 年広島大学大学院博士課程後期入学，2017 年中部大学研究員(～2019 年)，2017 年広島大学大学院博士後期課程修了．2019 年中部大学特任助教．2014 年独立行政法人日本学術振興会特別研究員 DC1．2014 年 ESIEE Paris 客員研究員(～2015 年)．コンピュータビジョン，パターン認識，医用画像処理の研究に従事.

山下隆義：:2002 年奈良先端科学技術大学大学院大学博士前期課程修了．2002 年オムロン株式会社入社，2009 年中部大学大学院博士後期課程修了(社会人ドクター)，2014 年中部大学講師，2017 年より同大学准教授．人の理解に向けた動画像処理，パターン認識・機械学習の研究に従事.

藤吉弘亘: 1997 年中部大学大学院博士後期課程修了．1997～2000 年米カーネギーメロン大学ロボット工学研究所 Postdoctoral Fellow．2000 年中部大学講師.2004 年より同大学教授．2005～2006 年米カーネギーメロン大学ロボット工学研究所客員研究員，計算機視覚，動画像処理，パターン認識・理解の研究に従事.