重み係数のスパース化による深層ニューラルネットワークのコンパクト化技術

DNN Compaction Method Eliminating Zero Weight Coefficients

谷口　敦司 YAGUCHI Atsushi　浅野　渉 ASANO Wataru　谷沢　昭行 TANIZAWA Akiyuki

深層学習で用いられる多層のニューラルネットワーク（DNN：Deep Neural Networks）は，画像認識などの処理で高い性能を実現している。しかし，DNNは大規模・複雑化する傾向があるため，演算能力や搭載メモリー量の制約があるエッジデバイス上で動作させることが困難であった。

東芝グループは，一般的な学習条件下で，DNNにおける多数の重み係数のうち一部が自動的にゼロ近傍に収束する（スパース化）現象を発見した。国立研究開発法人 理化学研究所（以下，理研と略記）と共同で，その発生原理を解明するとともに，学習後にゼロとなった重み係数を削除することでDNNのデータ量を削減するコンパクト化技術を開発した。画像認識の公開データセットを用いた実験の結果，この技術が認識精度の低下を抑えながら，重み係数を約80％削減できることを確認した。

Multilayer neural networks for deep learning, or so-called deep neural networks (DNNs), deliver superior performance in various applications including image recognition. However, as DNNs often tend to increase in scale and complexity, requiring higher processing performance, it is difficult to implement applications using them on edge devices with limited computation power and memory capacity.

While investigating how to make such large-scale DNNs more compact, the Toshiba Group discovered a phenomenon in which a portion of the large number of weight coefficients of a DNN automatically converge to zero when training a DNN under general conditions. In cooperation with the Institute of Physical and Chemical Research (RIKEN), we have elucidated the principle of this phenomenon and developed a DNN compaction method to reduce the volume of data in a DNN by eliminating those zero weight coefficients after training. Experiments on image recognition using open datasets have verified that this method achieves a reduction in the weight coefficients of a DNN of more than 80% compared with conventional methods while maintaining recognition accuracy.

1. まえがき

近年，様々な分野に機械学習が適用されるようになり，画像認識などをより精度良く行うために，DNNを用いた深層学習の導入が進んでいる。DNNは高い性能を実現できるが，大規模・複雑化するため，処理に使用するデバイスに制約があった。そこで東芝グループは，重み係数のスパース化を利用してDNNのデータ量を削減するコンパクト化技術を開発した。ここでは，開発したコンパクト化技術の概要と，評価結果について述べる。

2. DNNの高性能化と実用上の問題点

ニューラルネットワークは，脳内のニューロン同士の接続関係に着想を得たモデルあり，ニューロンを模した複数のノードから成る入力層，中間層，及び出力層から構成される（**図1**）。各ノードは前後の層のノードと連結しており，重み係数と呼ばれるパラメーターが与えられている。ニューラルネットワークは，入力層と出力層に与えるデータセットに合わせて，これらの重み係数を調整することで様々な関数を表現できる。

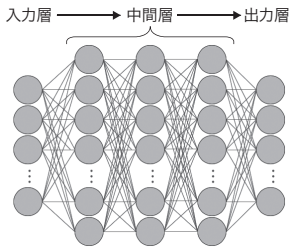

図1．ニューラルネットワークの模式図
中間層を多層化したニューラルネットワークを，一般にDNNと呼ぶ。
Schematic diagram of neural network structure

この重み係数の調整手順は学習と呼ばれる。例えば，写真に写る物体を認識する場合には，画像をベクトル化したデータを入力層に，物体のラベルを0と1の離散値で表したベクトルデータを出力層に与えて学習を行う。もし，学習後のニューラルネットワークが精度良く認識できるとすると，中間層には入力層のデータから抽出された認識に有効な情報（特徴量）が含まれているはずである。このように，目的と

なる関数を表現する上で重要な情報を抽出する手順は特徴抽出と呼ばれる。

一般に，中間層の数を増やすほど複雑な特徴を抽出することが可能になり，認識性能が向上すると考えられる。しかし従来は，多層化することで学習が難しくなる，計算量が増加する，などの問題があり，期待する性能を実現することができなかった。そのため，特徴量を抽出する部分は目的に合わせて人が設計し，最終的な分類などの処理だけを機械学習で決定するのが一般的だった。

これに対して近年，DNNの新たな構造の開発や計算機の処理性能の向上により，特徴抽出を含む入力から出力に至る過程を一気通貫で学習することが可能になってきた。このようなDNNを学習する手法は，深層学習と呼ばれる。DNNの性能が，人手で設計したものを上回る事例が多数報告されており[1]，音声認識や，機械翻訳，画像認識などの様々な用途への適用が検討されている。しかし，高い性能を実現するDNNは，大規模・複雑化する傾向があり，演算能力やメモリー量が限られたデバイス上で動作させることが困難であった。

3. 深層学習とDNNのコンパクト化

深層学習は，重み係数を含む学習可能なパラメーターΘを，式(1)を最適化することによって学習する。

$$\min_{\Theta} \frac{1}{N}\sum_{n=1}^{N} L(x_n, y_n, \Theta)+\lambda R(\Theta) \qquad (1)$$

ここで，x_n，y_n（n=1，2，…，N）はDNNの入力層と出力層に与えるベクトルデータ，Nは学習に用いるサンプル数を表す。L（x_n，y_n，Θ）は，DNNの出力と学習時に出力層に与えるデータ（教師データ）との乖離（かいり）を表す損失関数，R（Θ）は学習データへの過適合（注1）を抑制する正則化と呼ばれる作用を施す関数である。λは，あらかじめ固定値を設定するパラメーターであり，大きいほど正則化の効果が高い。

式(1)は，目的関数を微分することで得られる勾配ベクトルの逆方向に，パラメーターを逐次更新することで最適化する。具体的には，学習データからランダムに選択したM（$\ll N$）個のサンプルだけを用いて一度の更新を行う確率的勾配降下法（SGD：Stochastic Gradient Descent）が，広く用いられる。

図1に示す全結合型のDNNの場合に中間層のノード数をkとすると，DNNの重み係数の数は，中間層の数を一つ増やすごとにk^2オーダーで増加する。例えば，k=1,000として，それぞれの重み係数を単精度浮動小数点型（32ビット）で保存する場合，1層ごとにデータ量が約4Mバイト増加することになる。そのため，できるだけ性能を落とさずにDNNのデータ量を削減するコンパクト化技術が，近年盛んに研究されている。

代表的なものの一つ目は，DNNの学習後に重要度の低い重み係数を削除する枝狩り手法[2]である。しかし，この手法は，削除する重み係数の選択基準や閾値（しきいち）の設定が難しいという問題があった。

二つ目は，学習中に重み係数をゼロ近傍に収束させるスパース化手法[3],[4]である。スパース（疎）はまばらという意味であり，機械学習や統計学の分野ではデータの大半がゼロで，意味のあるものは少数に限られることを指す。重み係数のスパース化は，正則化関数としてスパース化を促す関数を用いることで実現するのが一般的である。例えば，重み係数の絶対値和が用いられる。しかし，このようなスパース化を促す関数は，微分不可能な点を含むことから，最適化が難しかった。

4. 重み係数のスパース化を活用したDNNのコンパクト化

東芝グループは，スパース化を促す特別な正則化関数を利用せず，重み係数の二乗和を用いるL_2正則化といった一般的な条件で深層学習を行う際にも，スパース化が発生する現象（**図2**）を発見し，理研と共同でその発生原理を解明した[5]。また，学習後にゼロとなった重み係数は認識結果に影響しないことを利用し，学習後にそれらの重み係数を削除するだけで，DNNを容易にコンパクト化できる技術を開発した。以下では，スパース化の発生原理と，開発したコンパクト化技術の概要について述べる。

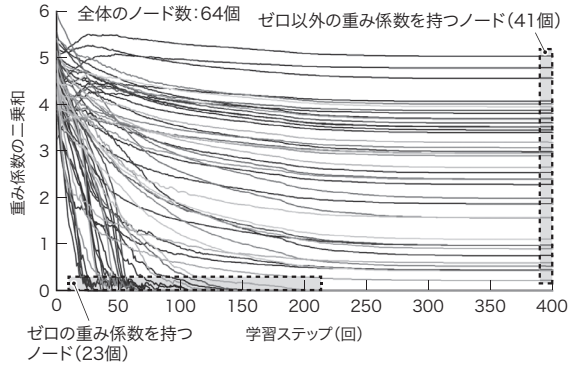

図2. 学習時の重み係数二乗和の推移

L_2正則化により，64個中23個のノードの重み係数がゼロに収束することを確認した。

Example of changes in sum of squares of weight coefficients when training DNN

（注1） 学習データに当てはまり過ぎて，未知のデータに対する精度が低い状態。

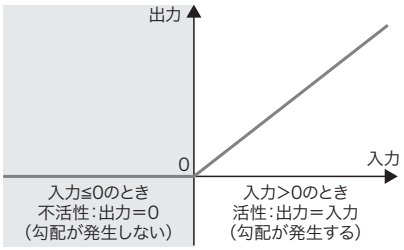

図3. ReLU

ReLUは，入力がゼロ以下の場合は出力がゼロとなり，勾配は発生しない。

Rectified linear unit (ReLU) used for activation function

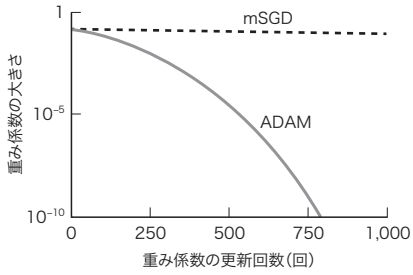

図4. mSGDとADAMで最適化した際の重み係数の減衰速度の比較

ADAMはmSGDよりも速く減衰する。

Comparison of convergence speed of weight coefficients obtained by momentum stochastic gradient descent (mSGD) and adaptive moment estimation (ADAM) methods

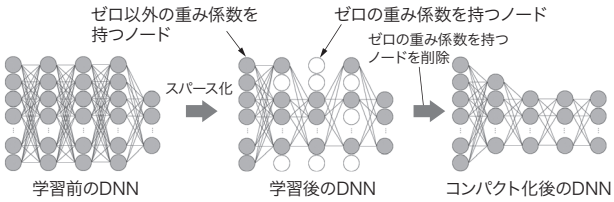

図5. DNNコンパクト化の手順

学習後にスパース化によりゼロになった重み係数を持つノードを削除するだけで，容易にコンパクト化できる。

Flow of DNN compaction processes

4.1 スパース化の発生条件

以下の三つを同時に満足する条件でDNNを学習する場合に，重み係数のスパース化が発生する。

(1) 中間層のノードごとの活性化関数として，ReLU（Rectified Linear Unit）[6]を使用

(2) 正則化関数として，L_2正則化を使用

(3) 最適化方法として，ADAM（Adaptive Moment Estimation）[7]を使用

ここで活性化関数とは，DNNの表現能力を高めるために各ノードに適用される非線形関数である。ReLUは入力が正の場合は恒等写像を，ゼロ以下の場合はゼロを出力する関数である（**図3**）。L_2正則化は，重み係数の値を小さく抑えることで学習データへの過適合を抑制する効果がある。ADAMは，確率的な最適化手法の一つであり，重み係数の更新ステップ幅を自動的に調整する働きを持つ。これらは深層学習において一般的に用いられる条件であるが，スパース化を発生させることは今まで報告されていなかった。

4.2 スパース化の発生原理

学習における重み係数の更新は，式(1)の損失関数の勾配と正則化関数の勾配の和によって決定される。損失関数の勾配は重み係数を学習データへ適合させ，L_2正則化の勾配は重み係数を小さくする（ゼロに減衰させる）働きがある。図3に示すように，ReLUは入力がゼロ以下の場合は不活性で損失関数の勾配が発生しないという特徴があるが，ReLUは多くの場合不活性であるとの報告[6]もある。そのような場合は損失関数の勾配は小さくなり，正則化関数の勾配の影響が相対的に強くなる。したがって，活性化しにくいReLUの入力につながる重み係数は，ゼロに減衰しやすくなると考えられる。

更に，ゼロに減衰する速度が重要である。通常のSGDに慣性項を導入したmomentum-SGD（mSGD）[8]を用いて最適化を行った場合は，スパース化が発生しない。これは，理論的なゼロへの減衰速度が，ADAMは二重指数オーダーであるのに対し，mSGDは指数オーダーであるためと考えられる。**図4**に示すように，ADAMとmSGDでは減衰速度に大きな差があることが確認できる。これらをまとめると，活性化しにくいReLUの入力につながる重み係数について，L_2正則化によるゼロへの減衰がADAMによって高速化されることが，スパース化の発生原理である。

4.3 DNNのコンパクト化

活性化しにくいReLUの入力につながる重み係数はDNNの出力に与える影響が小さい。前述のように，そのような重み係数はスパース化によってゼロとなることから，重要度の低い重み係数が学習中に自動的に削除されると考えられる。したがって，枝狩り手法と異なり，学習後に削除する重み係数を選択する必要はなく，ゼロになった重み係数を削除するだけである（**図5**）。また，重み係数は不規則にスパース化するわけではなく，活性化しにくいReLUの入力につながる全ての重み係数が同時にスパース化する。したがって，図5のようにノード単位での削除が可能であり，コンパクト化の効果が高い。ここで，スパース化の強度は式(1)のパラメーターλを調整することで制御できる。

5. コンパクト化の評価実験

画像認識の公開データセットを用いて，開発したコンパク

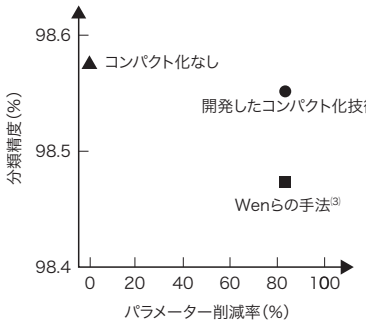

図6. MNISTの手書き数字のデータセットを用いた認識実験の結果
開発したコンパクト化技術は，分類精度の低下を抑えながら，パラメーター数を83.7％削減できた。
Results of experiments on handwritten digit recognition using Mixed National Institute of Standards and Technology (MNIST) dataset

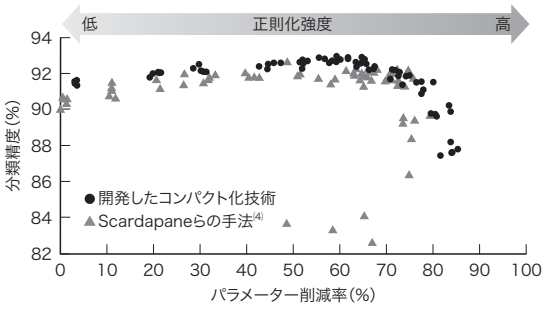

図7. CIFAR-10の一般物体のデータセットを用いた認識実験の結果
開発したコンパクト化技術は，分類精度を維持しながら，正則化強度によってパラメーター削減率を調整できる。
Results of experiments on object recognition using Canadian Institute for Advanced Research (CIFAR)-10 dataset

ト化技術と，スパース化を促す正則化関数を用いたコンパクト化手法[3], [4]との比較実験を行った[5]。手書き数字（0～9）のデータセット（MNIST）[9]を用いた，4層の全結合型のDNNに対する実験の結果を，**図6**に示す。開発した技術は，コンパクト化なしのDNNに比べて分類精度0.02％の低下で，パラメーター数を83.7％削減できた。これは，Wenらの手法[3]と同等のパラメーター削減率であり，分類精度の低下を小さく抑えられていることを確認した。

一般物体認識のデータセット（CIFAR-10）[10]を用いた，16層の畳み込み型のDNNに対する実験の結果を**図7**に示す。開発した技術は，認識精度の低下が約1％以内の範囲では，正則化強度を変更することでDNNのパラメーター数を30～70％程度削減できた。また，Scardapaneらの手法[4]よりも分類精度が高いことを確認した。

6. あとがき

深層学習において，ある条件で一部の重み係数がスパース化する現象を発見し，それを活用したDNNのコンパクト化技術を開発した。この技術を用いると，一般的な学習条件の設定と学習後の不要な重み係数の削除によりコンパクト化できるので，従来に比べて容易にDNNのコンパクト化が可能である。画像認識の公開データセットを用いた実験で，この技術が，従来技術よりも認識精度の低下を抑えながら，DNNをコンパクト化できることを確認した。

今後，自動運転向け画像認識システムなど様々な組み込み機器やエッジデバイスにおける高度なDNNの活用に向けて，更なる研究開発を進めていく。

文　献

(1) LeCun, Y. et al. Deep learning. Nature. 2015, **521**, p.436–444.
(2) Li, H. et al. "Pruning Filters for Efficient ConvNets". Proc. International Conference on Learning Representations (ICLR). Toulon, France, 2017-04, ICLR. 2017, p.1–13.
(3) Wen, W. et al. "Learning Structured Sparsity in Deep Neural Networks". Proc. Conference on Neural Information Processing Systems (NeurIPS). Barcelona, Spain, 2016-12, NeurIPS. 2016, p.2082–2090.
(4) Scardapane, S. et al. Group Sparse Regularization for Deep Neural Networks. Neurocomputing. 2017, **241**, p.81–89.
(5) Yaguchi, A. et al. "Adam Induces Implicit Weight Sparsity in Rectifier Neural Networks". Proc. IEEE International Conference on Machine Learning and Applications (ICMLA). Orland, FL, 2018-12. 2018, p.318-325.
(6) Glorot, X. et al. "Deep Sparse Rectifier Neural Networks". Proc. International Conference on Artificial Intelligence and Statistics (AISTATS). Ft. Lauderdale, FL, 2011-04. 2011, p.315–323.
(7) Kingma, D. P. ; Ba, J. "Adam: A Method for Stochastic Optimization". Proc. International Conference on Learning Representations (ICLR). San Diego, CA, 2015-05. 2015, p.1–15.
(8) Qian, N. On the momentum term in gradient descent learning algorithms. Neural Networks. 1999, **12**, 1, p.145–151.
(9) LeCun, Y. et al. Gradient-based learning applied to document recognition. Proc. IEEE. 1998, **86**, 11, p.2278–2323.
(10) Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto. 2009, 58p.

谷口　敦司　YAGUCHI Atsushi
研究開発本部 研究開発センター
アナリティクス AI ラボラトリー
Analytics AI Lab.

浅野　　渉　ASANO Wataru
東芝インフラシステムズ（株）
インフラシステム技術開発センター
自動化・画像応用システム開発部
Toshiba Infrastructure Systems & Solutions Corp.

谷沢　昭行　TANIZAWA Akiyuki
研究開発本部 研究開発センター
アナリティクス AI ラボラトリー
映像メディア学会会員
Analytics AI Lab.