パラメータ削減と復元による深層学習モデルの適応的な圧縮

佐々木健太† 佐々木勇和† 鬼塚 真†

† 大阪大学大学院情報科学研究科 〒 565–0871 大阪府吹田市山田丘 1-5
E-mail: †{sasaki.kenta,sasaki,onizuka}@ist.osaka-u.ac.jp

あらまし 深層学習モデルの利用には多くの計算資源やメモリ，電力が必要であり，特に携帯端末などのハードウェア資源の限られた環境で利用する際に問題となる．そのため，モデルの高い性能をできる限り維持しつつパラメータを削減し効率化を図る研究が多く行われている．パラメータの枝刈りと復元を交互に繰り返す既存の手法では，枝刈りを行った後にネットワーク構造を固定して再学習を行う手法が用いられており，パラメータを大きく削減した際にモデルの性能を維持することが難しいという問題点がある．そこで本稿では，ネットワークの性能を維持したままパラメータを大きく削減することのできる動的なパラメータの枝刈り手法を利用してパラメータの削減と復元を段階的に行うことによって，パラメータを大きく削減した際の深層学習モデルの性能向上に取り組む．

キーワード 深層学習，ニューラルネットワーク，圧縮

1. はじめに

近年，深層学習モデルは画像認識等の分野で高い性能を発揮し注目されている [1]．深層学習の応用分野が広がるにつれて，これまで深層学習モデルがあまり利用されてこなかったスマートフォンやドローンなどの多様な環境において深層学習モデルを利用したいという要求が高まりつつある [2]．一方で，多くの深層学習モデルはパラメータの数が非常に膨大であるため，モデルを利用する際の計算コストやメモリ使用量，消費電力が大きく，特に携帯端末などのハードウェア資源の限られた環境で利用する際に問題となる．深層学習モデルのパラメータは冗長性が非常に高いことが Denil らによる研究 [3] で明らかにされており，モデルの性能をできる限り維持しつつパラメータを削減し効率化を図る様々な手法が提案されている [4]．本稿では特に枝刈りと復元を繰り返すことで段階的に深層学習モデルを圧縮する手法に注目する [5]．

Han らによる研究 [6] では，枝刈りによってパラメータを削減した後に削減したパラメータの復元を行うことでモデルの性能を向上させる手法が提案されており，CNN や RNN 等の多様な深層学習モデルにおいてモデルの性能が向上することを確認している．Jin らの研究 [5] では，パラメータの削減と圧縮前のネットワーク構造の復元を交互に繰り返すことによって段階的にモデルを圧縮する手法を提案している．これら 2 つの既存研究においては，パラメータを削減する際に，枝刈りを行いネットワークの構造を確定させた後に再学習を行う手法を用いている．しかしながら，このパラメータ削減手法は，学習中にパラメータの削減と復元を行う動的な枝刈り手法と比べてモデルの性能を維持しつつパラメータを大きく削減することが難しいという問題点がある [7]．また，既存研究ではパラメータの復元を行う際に常に元のパラメータを全て復元しているため，段階的にパラメータを削減した際，特に圧縮の終盤においてネットワーク構造が大きく変動し，モデルの学習に悪影響を与える可能性がある．

そこで本稿では Jin らの手法 [5] を発展させ，動的な枝刈り手法である Dynamic Network Surgery [7] を用いて，復元するパラメータ数を徐々に減らしながらパラメータの削減と復元を交互に繰り返す手法を提案する．MNIST データセットと CIFAR-10 データセットでの学習を行った 2 つのモデルに対して提案手法を適用する実験を行い，提案手法がモデルの性能を維持しつつパラメータを大きく削減することができ，通常の Dynamic Network Surgery と同程度の圧縮性能を持つことを確認した．

以降の本稿の構成は以下の通りである．まず，2. で関連研究について述べる．そして，3. で用いる手法の詳細について説明し，4. において実験の設定と結果を示す．最後に，5. において結論と今後の課題について述べる．

2. 関連研究

深層学習モデルを圧縮する様々な手法が提案されている [4]．本稿では，特に深層学習モデルのパラメータを枝刈りすることによってパラメータ数を削減する手法に注目する．深層学習モデルのパラメータを枝刈りする手法として，モデルの損失関数の二階偏微分を利用して枝刈りを行う Optimal Brain Damage [8] が知られている．しかしながら，二階偏微分の計算は計算コストが大きいため大規模で複雑なモデルへの適用が難しい．近年の大規模なモデルに対して枝刈りによってパラメータ削減を行なった研究として Han らによる研究 [9] が存在する．Han らによる研究では，絶対値の小さなパラメータを枝刈りと再学習を繰り返すことで，AlexNet のパラメータ数を元のモデルの約 11% 程度まで削減することに成功している．また，この枝刈り手法を重みの量子化と共有化，そしてハフマン符号化と組み合わせた Deep Compression [10] が Han らによって提案されており，モデルのさらなる圧縮に成功している．また，Guo ら [7] は，再学習のプロセスに枝刈りの操作を組み込むことで学習中に動的にパラメータの枝刈りと復元を行う Dynamic Network Surgery を提案しており，AlexNet においてパラメー

タ数を元のモデルの 約 5.7% 程度まで削減することに成功している．

複数の研究において，深層学習モデルのパラメータの枝刈りと枝刈りしたパラメータ全ての復元を交互に繰り返すことによって，モデルの性能を向上させることができることが報告されている．Han らによる研究 [6] においては，一度パラメータの絶対値による枝刈りを行なってパラメータを削減し，その後全てのパラメータを復元して学習させ直すことによって元のモデルよりも高い予測性能を持つモデルが得られることが画像分類やキャプション生成，そして音声認識のタスクを行う多様な深層学習モデルでの実験において確認されている．Han らは Sparse 化と Dense 化を行うことでモデルの性能が向上する要因として，パラメータの最適化において鞍点を回避できる点，圧縮によって目的関数の形状が滑らかでノイズに強い低次元での最適化となる点，重みを初期化する複数の機会がある点，そしてパラメータ同士の共適応を防ぐことができる点などが推測できるとしている．しかしながら，Han らの研究はパラメータの削減を目的としたものではなく，最終的に元のモデルのパラメータを全て復元している．Jin らによる研究 [5] では，パラメータの絶対値を用いた枝刈りと元のモデルへの復元を交互に繰り返しながらモデルを段階的に圧縮する手法を提案している．本稿では，この手法を発展させ，動的なパラメータの枝刈りとパラメータの復元を段階的に行う手法を提案する．

また，パラメータの枝刈りと復元を繰り返す手法に関連が深いと考えられる手法として，深層学習モデルの学習の際にランダムに選んだパラメータの一部を一時的に無効化しながら学習を行う事で汎化性能を向上させる Dropout [11] がある．本稿で注目した圧縮手法はモデルの圧縮を目的としているという点で Dropout と異なっている．モデルの圧縮が目的であるため，本稿で注目した圧縮手法においては削減するパラメータが絶対値により決定され，パラメータの削減は一時的なものではなく継続的なものであるという特徴がある．

深層学習モデルを圧縮するその他の手法としては，行列分解を用いて近似を行う手法 [12] やパラメータを低精度化してより少ない bit 数で表現する手法 [13] [14] などが提案されている．また，大規模な深層学習モデルの出力を利用して小規模なモデルの学習を行う蒸留と呼ばれる手法 [15] も提案されている．

3. 手　　法

本稿では Jin らによって提案されているパラメータの枝刈りと復元を交互に繰り返す手法 [5] を発展させた手法を提案する．本稿では，枝刈りの手法の中でも高い圧縮性能を達成している Dynamic Network Surgery [7] を Jin らの手法のパラメータの削減と復元のフェイズに組み込む．本章では，まず Dynamic Network Surgery 及びパラメータの枝刈りと復元を交互に繰り返す既存手法についてそれぞれ説明した後に，提案手法の詳細について述べる．

3.1 Dynamic Network Surgery

Dynamic Network Surgery は，パラメータの枝刈りと復元を学習のプロセスに組み込むことで学習中に枝刈りと復元を行う動的な枝刈り手法である．Dynamic Network Surgery のアルゴリズムを Algorithm1 に示す．Dynamic Network Surgery では学習のイテレーションごとに絶対値が閾値以下となったパラメータの削減と，絶対値が閾値を上回ったパラメータの復元を行う．Dynamic Network Surgery においては，パラメータの枝刈りをマスク行列を用いて表現する．そのため，絶対値が閾値以下となり，枝刈りによって無効化されているパラメータについても，パラメータの値が保持されている．順伝播と逆伝播はマスク行列を用いて枝刈りで保持されているパラメータのみを用いて行い，パラメータの更新の際は枝刈りで削除されたパラメータを含む全てのパラメータを更新する点が特徴である．更新によって絶対値が閾値以下となったパラメータはマスク行列の該当箇所を更新することによってモデルから削除され，新たに閾値を上回ったパラメータについては同様にマスク行列を更新することで復元を行う．Dynamic Network Surgery では，ネットワークの構造が頻繁に変更されるのを防ぎ，手法の安定性を高めるために，枝刈りの閾値と復元の閾値は別の値を設定する．また，枝刈りと復元を確率的に行い，マスク行列の各要素が更新される確率を徐々に低下させることでモデルの収束を助けている．

Algorithm 1 Dynamic Network Surgery [7] をもとに作成

Input: $\boldsymbol{X}$: Training data, $\hat{\boldsymbol{W}}$: Parameters of the Network,
$\boldsymbol{T}$: Thresholds for each layer
α: Base learning rate, f: Learning policy,
σ: Pruning and splicing policy,
m: Maximum number of iterations,
Output: $\boldsymbol{W}, \boldsymbol{M}$: The updated parameters and binary masks
Initialization: $\boldsymbol{W} \leftarrow \hat{\boldsymbol{W}}$, $\boldsymbol{M} \leftarrow \boldsymbol{1}$, $iter \leftarrow 1$, $\beta \leftarrow f(\alpha,\ iter)$
while $iter \leqq m$ **do**
Choose a minibatch of network input from $\boldsymbol{X}$
Forward propagation and backpropagation
with the masked parameters
Update $\boldsymbol{M}$ by $\boldsymbol{T}$ and magnitude of current $\boldsymbol{W}$
with the probability of $\sigma(iter)$
Update $\boldsymbol{W}$ by gradient descent algorithm
with the learning rate β
$iter \leftarrow iter + 1$, $\beta \leftarrow f(\alpha,\ iter)$
end while

3.2 深層学習モデルの段階的な圧縮手法

Jin らによって提案されている手法 [5] は，パラメータを削減する Sparse 化フェイズと削減したパラメータを復元する Dense 化フェイズの 2 つのフェイズを交互に繰り返すことで段階的なモデルの圧縮を行う．まず，Sparse 化フェイズおいては，モデルのパラメータの中で絶対値の小さなパラメータの枝刈りを行い，枝刈り後にモデルの再学習を行う．この時，保持するパラメータを確定しネットワーク構造を更新した後に再学習を行う手法を用いる．続く Dense 化フェイズにおいては，Sparse 化フェイズで枝刈りした全てのパラメータの復元と再学習を行う．Dense 化フェイズにおいて，Sparse 化フェイズで枝刈りされたパラメータ全てを 0 で再初期化して圧縮前のネットワーク構造

を復元する．Sparse 化フェイズにおいて保持するパラメータを段階的に減らしながら 2 つのフェイズを繰り返すことで段階的なモデルの圧縮を行う．Jin らの手法で用いられている枝刈りを行った後にネットワーク構造を固定して再学習を行う手法は，パラメータを大きく削減した際にモデルの性能を維持することが難しいと言う問題点がある．また，Dense 化フェイズにおいて，常に全てのパラメータを復元しており，特に圧縮の終盤においてネットワークの構造が大きく変動してしまうため学習に悪影響を与えている可能性がある．

3.3 提 案 手 法

本稿では，Jin らの手法を発展させることでモデルの圧縮性能を向上させることを目指す．本稿で提案する手法と Jin らの手法との主な違いは以下の 3 点である．(1) 枝刈りを行いネットワークの構造を確定させた後に再学習を行う手法ではなく再学習のプロセスに枝刈りを組み込んだ動的な枝刈り手法である Dynamic Network Surgery を用いる．Dynamic Network Surgery は枝刈りと再学習が独立した枝刈り手法と比べて高い圧縮性能を達成している．(2)Dense 化フェイズにおいて枝刈りしたパラメータを復元する際に，圧縮前のネットワーク構造の完全な復元を行わず，徐々に復元する結合を減らす．これによって，圧縮の終盤においてネットワーク構造が過度に変動することを防ぐ．(3) パラメータを復元する際にゼロで初期化せず，Dynamic Network Surgery のアルゴリズムにおいて保持されている値を復元する．提案手法の概要を図 1 に示し，そのアルゴリズムをアルゴリズム 2 に示す．本稿で提案する手法は Jin らの手法と同様に Sparse 化フェイズと Dense 化フェイズの 2 つのフェイズで構成される．まず，Sparse 化フェイズでは枝刈りの閾値を比較的大きな値に設定して Dynamic Network Surgery を行うことで，パラメータ数を大きく削減する．その際には，その時点のパラメータの行列とマスク行列を用いて，有効なパラメータの一定の割合が Dynamic Network Surgery の最初の閾値処理で削減されるよう閾値を設定する．次の Dense 化フェイズでは，閾値を引き下げて再び Dynamic Network Surgery による学習を行うことで Sparse 化フェイズで削減されたパラメータの一部を復元する．この際も，その時点のパラメータの行列とマスク行列を用いて，その時点で有効なパラメータ数に対して一定の割合が Dynamic Network Surgery の最初の閾値処理で復元されるように閾値を設定する．これにより，有効なパラメータ数が減るにつれて復元されるパラメータ数も減少することとなり段階的に Dense 化フェイズ後のモデルのパラメータ数を減らすことができる．パラメータを復元する際にはパラメータの再初期化は行わず，Dynamic Network Surgery のアルゴリズムにおいて保持されている値を引き継ぐ．この Sparse 化フェイズと Dense 化フェイズを交互に繰り返すことによって段階的にモデルの圧縮を行う．

Algorithm 2 提案手法

Input: $\boldsymbol{X}$: Training data, $\hat{\boldsymbol{W}}$: Pretrained parameters,
n: Maximum number of steps
Output: $\boldsymbol{W}, \boldsymbol{M}$: The updated parameters and binary masks
$\boldsymbol{T}$: Thresholds for each layer
Initialization: $\boldsymbol{W} \leftarrow \hat{\boldsymbol{W}}$, $step \leftarrow 1$, $\boldsymbol{M} \leftarrow \boldsymbol{1}$
while $step \leqq n$ **do**
Sparse Phase:
Update $\boldsymbol{T}$ *with current* $\boldsymbol{W}$ *and* $\boldsymbol{M}$ *to reduce the parameters*
Update $\boldsymbol{W}, \boldsymbol{M}$ *by Dynamic Network Surgery with* $\boldsymbol{T}$
Dense Phase:
Update $\boldsymbol{T}$ *with current* $\boldsymbol{W}$ *and* $\boldsymbol{M}$ *to restore the parameters*
Update $\boldsymbol{W}, \boldsymbol{M}$ *by Dynamic Network Surgery with* $\boldsymbol{T}$
$step \leftarrow step + 1$
end while

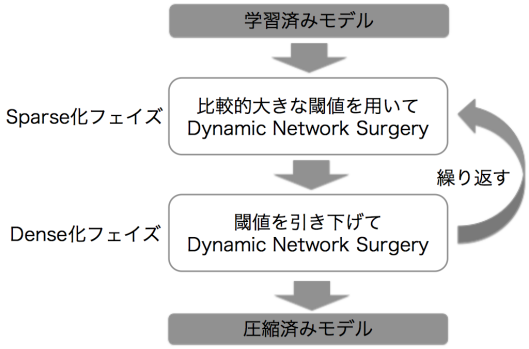

図 1　パラメータの削減と復元を繰り返す手法

4. 実　　験

4.1 実 験 設 定

本稿では，深層学習フレームワークである Caffe [16] とその Python 向けのインターフェースである Pycaffe を用いて実験を行なった．また，Dynamic Network Surgery の実装には Guo らによって公開されているソースコード (注1) を利用した．データセットとして，手書き数字の画像のデータセットである MNIST データセット (注2) と 10 カテゴリの画像データセットである CIFAR-10 データセット [17] を用いた．MNIST データセットには 60,000 枚の訓練用データと 10,000 枚のテスト用データが含まれる．本稿での実験では，MNIST データセットの訓練用データのうち 10%にあたる 6,000 枚の画像を検証用データとし，残る 54,000 枚の画像を訓練用データとした．また，CIFAR-10 データセットには 50,000 枚の訓練用データと 10,000 枚のテスト用データが含まれる．本稿の実験では，CIFAR-10 データセットの訓練用データのうち 10%にあたる 5,000 枚の画像を検証用データとし，残る 45,000 枚の画像を訓練用データとした．

本稿での実験では，いずれのモデルについても学習を行う際はミニバッチによる確率的勾配降下法を用いた学習を行い，その際 L2 ノルムによる正則化とモメンタム法を適用した．また誤差関数としては交差エントロピー誤差関数を用いた．出力層以外の活性化関数としては正規化線形関数を用い，出力層の活性化関数としてはソフトマックス関数を用いた．MNIST デー

(注1)：https://github.com/yiwenguo/Dynamic-Network-Surgery
(注2)：http://yann.lecun.com/exdb/mnist/

表 1　LeNet-5 の各層のパラメータ数

	パラメータ数	モデル全体に占める割合
畳込み層 1	0.5K	0.12%
畳込み層 2	25K	5.81%
全結合層 1	400K	92.92%
全結合層 2	5K	1.16%
モデル全体	430.5K	

表 2　CIFAR-10 の実験で用いたモデルの各層のパラメータ数

	パラメータ数	モデル全体に占める割合
畳込み層 1	2.4K	2.68%
畳込み層 2	25.6K	28.62%
畳込み層 3	51.2K	57.25%
全結合層 1	10.2K	11.45%
モデル全体	89.4K	

タセットでの実験では，2 層の畳込み層と 2 層の全結合層からなるニューラルネットワークである LeNet-5 を圧縮の対象とした．LeNet-5 の各層のパラメータ数とそのモデル全体に占める割合を表 1 に示す．圧縮前のモデルの学習の際には，Caffe の MNIST チュートリアル (注3) の設定を用いて学習を行なった．ただし，学習のイテレーション数のみ 1 万イテレーションから 10 万イテレーションに変更した．異なる初期値で 3 回学習を行った中でテストデータにおける性能が最も高いモデルを圧縮の対象とした．今回圧縮の対象としたモデルのテストデータにおける予測精度は 99.15%であった．CIFAR-10 データセットでの実験では，Caffe によって提供されている 3 層の畳込み層と 1 層の全結合層から構成されるモデル (注4) を圧縮の対象とした．このモデルの各層のパラメータ数とそのモデル全体に占める割合を表 2 に示す．圧縮前の学習の際には，正則化の係数を 0.004，モメンタムの係数を 0.9 に設定し，バッチサイズは 100 として学習率 0.001 での学習を 8 万イテレーション行なった後に学習率 0.0001 での学習を 2 万イテレーション行った．LeNet-5 での実験と同様に異なる初期値で 3 回学習を行った中でテストデータにおける性能が最も高いモデルを圧縮の対象とした．今回圧縮の対象としたモデルのテストデータにおける予測精度は 81.53%であった．

これらのモデルに対して，通常の Dynamic Network Surgery と提案手法をそれぞれ用いてパラメータを削減し，パラメータ削減後のモデルのテストデータでの予測精度について評価を行った．通常の Dynamic Network Surgery については，元のモデルからの圧縮を閾値を変えて複数回パラメータ削減を行い，パラメータ削減後のテストデータにおける予測精度を評価した．提案手法については，Sparse 化と Dense 化を MNIST データセットでの実験ではそれぞれ 7 回ずつ，CIFAR-10 データセットでの実験ではそれぞれ 5 回ずつ繰り返すことで徐々にパラメータ数を削減し，それぞれのステップの Sparse 化フェイズ及び Dense フェイズ終了時のモデルのテストデータにおける予測精度を評価した．いずれのモデルでの実験においても，Dynamic Network Surgery を適用する際には，Guo らの手法に従い全結合層と畳込み層の枝刈り及び復元を別々に行った．具体的には，畳込み層と全結合層の一方を学習する際にもう一方の学習率を 10 分の 1 に設定することでそれぞれの層の枝刈りと復元を行った．層ごとに異なるパラメータの削減割合を適切に設定することで精度を維持したままより多くのパラメータを削減することが可能である．しかし，今回の実験ではいずれの手法についても層ごとにパラメータの削減割合を調整することは行わず，Dynamic Network Surgery を適用する際の最初の閾値処理において，全ての層で同じ割合だけパラメータが削減されるよう閾値の設定を行った．閾値の設定の際には，Dynamic Network Surgery を行う際の最初の閾値処理において，各層のパラメータの数が 1 を下回ることがないように閾値を設定した．

通常の Dynamic Network Surgery を適用する際には，いずれのモデルでの実験でも，圧縮前のモデルの学習時と同じ設定で畳込み層と全結合層のそれぞれについて合計 10 万イテレーションずつの学習を行い合計 20 万イテレーションの学習を行った．提案手法を適用する際にも全結合層と畳込み層のそれぞれについて 10 万イテレーションの学習を行ったが，Sparse 化フェイズと Dense 化フェイズそれぞれで再学習を行う必要があるため，1 回のステップでは Dynamic Network Surgery の倍の 40 万イテレーションの学習を行う．その他の学習率とその減衰方法，正則化やモメンタムの係数等のパラメータについては圧縮前のモデルの学習の際と同じ値に設定し，2 つの手法で同じ値を利用した．また，提案手法では，Sparse 化フェイズにおいては，Dynamic Network Surgery の最初の閾値処理においてパラメータの数がその Sparse 化フェイズの前の 12.5%になるように閾値を設定し，Dense 化フェイズにおいてはパラメータの数がその Dense 化フェイズの前の 2 倍になるように閾値を設定し実験を行なった．

4.2　実 験 結 果

2 つのモデルに対して提案手法を適用した際の各ステップの Sparse 化フェイズと Dense 化フェイズの終了時におけるモデルのパラメータの維持率とテストデータでの予測精度を図 2 及び図 3 に示す．パラメータの維持率とは，手法を適応する前のモデルのパラメータの数に対するその時点で削減されていないパラメータの数の割合である．また，手法を適応する前のモデルのテストデータにおける予測精度も参考のためにプロットしている．いずれのモデルでの実験においても，提案手法を用いて圧縮前のモデルと同程度の予測精度を維持しつつ，パラメータ数を大きく削減することができた．また，削減および復元されるパラメータが徐々に減少し，圧縮の後半において，モデルのパラメータ数が大きく変動することを防ぐことができていることが確認できた．しかし，Dynamic Network Surgery の再学習によるパラメータの復元と削減の効果によって，各フェイズで削減又は復元するパラメータの割合が一定とはならないことが明らかとなった．各フェイズで削減又は復元されるパラメータの数をコントロールするためには，各フェイズでの閾値

(注3)：http://caffe.berkeleyvision.org/gathered/examples/mnist.html
(注4)：https://github.com/BVLC/caffe/blob/master/examples/cifar10/cifar10_full_train_test.prototxt

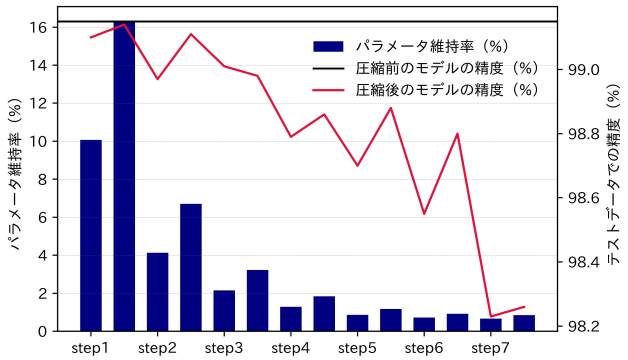

図 2　MNIST データセットを用いた実験における
各ステップでのパラメータ維持率と精度

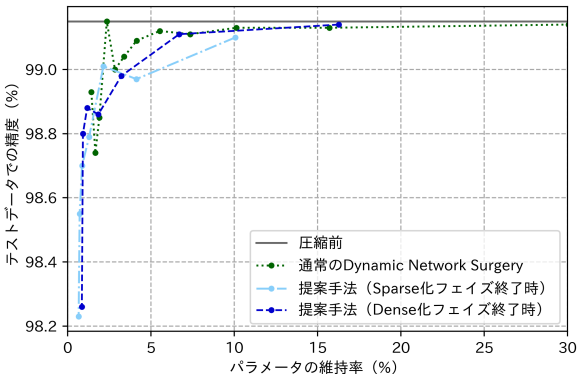

図 4　MNIST データセットを用いた実験における
通常の Dynamic Network Surgery との比較

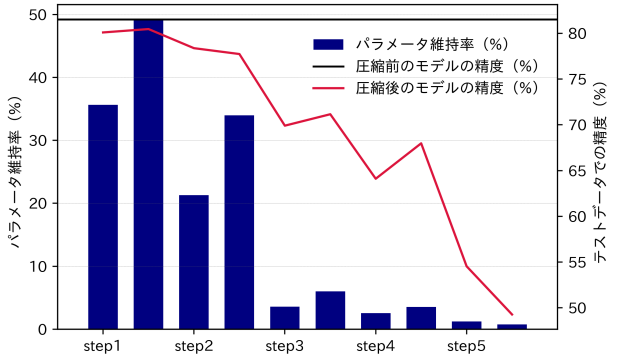

図 3　CIFAR-10 データセットを用いた実験における
各ステップでのパラメータ維持率と精度

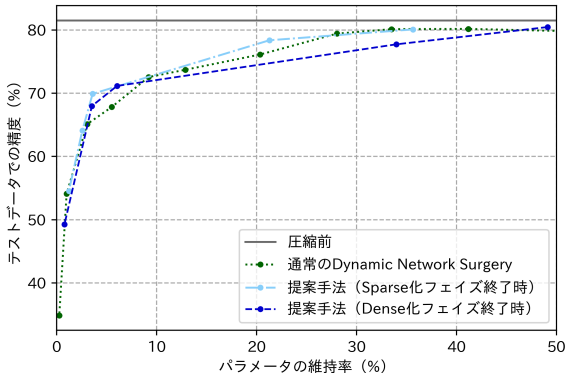

図 5　CIFAR-10 データセットを用いた実験における
通常の Dynamic Network Surgery との比較

の更新を適切に行う必要があると考えられる．

次に，提案手法と通常の Dynamic Network Surgery の性能を比較するために，それぞれの手法におけるパラメータ維持率とモデルのテストデータにおける予測精度をプロットしたグラフを図 4 及び図 5 に示す．圧縮前のモデルの精度を「圧縮前」として，通常の Dynamic Network Surgery による元のモデルの圧縮を閾値を変えて複数回行った結果を「通常の Dynamic Network Surgery」として，提案手法によって段階的な圧縮を行った際の各ステップの Sparse 化フェイズ終了時のパラメータ維持率と精度を「提案手法（Sparse 化フェイズ終了時）」として，Dense 化フェイズ終了時のパラメータ維持率と精度を「提案手法（Dense 化フェイズ終了時）」としてプロットしている．いずれのモデルにおいても提案手法の圧縮性能は通常の Dynamic Network Surgery と大きな差は無く，Dynamic Network Surgery の性能を上回るには手法にさらなる改善が必要となる結果となった．

5. 結　　論

本稿では，深層学習モデルのパラメータの動的な枝刈り手法である Dynamic Network Surgery を用いてモデルのパラメータの枝刈りと復元を繰り返すことで段階的にパラメータを削減する手法を提案した．MNIST データセットと CIFAR-10 データセットでの学習を行なった 2 つのモデルを対象とした実験において，提案手法がモデルの性能を維持しつつパラメータを大きく削減することができ，通常の Dynamic Network Surgery と同程度の圧縮性能を持つことを確認した．今後は，特に Dense 化を行った際に，既存研究 [6] [5] で報告されているような圧縮前のモデルからの精度の向上が確認できなかった点について要因を見極め，手法の改善に取り組みたい．

文　　献

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Computing Research Repository*, Vol. abs/1512.03385, , 2015.

[2] Antonio Loquercio, Ana Isabel Maqueda, Carlos R. Del Blanco, and Davide Scaramuzza. Dronet: Learning to fly by driving. *IEEE Robotics and Automation Letters*, 2018.

[3] Misha Denil, Babak Shakibi, Laurent Dinh, Marc'Aurelio Ranzato, and Nando de Freitas. Predicting parameters in deep learning. In *Advances in Neural Information Processing Systems*, pp. 2148–2156, 2013.

[4] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. A survey of model compression and acceleration for deep neural networks. *arXiv preprint arXiv:1710.09282*, 2017.

[5] Xiaojie Jin, Xiaotong Yuan, Jiashi Feng, and Shuicheng

Yan. Training skinny deep neural networks with iterative hard thresholding methods. *arXiv preprint arXiv:1607.05423*, 2016.

[6] Sharan Narang Huizi Mao Enhao Gong Shijian Tang Erich Elsen Peter Vajda Manohar Paluri John Tran Bryan Catanzaro William J. Dally Song Han, Jeff Pool. Dsd: Dense-sparse-dense training for deep neural networks. *International Conference on Learning Representations (ICLR)*, 2017.

[7] Yiwen Guo, Anbang Yao, and Yurong Chen. Dynamic network surgery for efficient dnns. In *Advances In Neural Information Processing Systems*, pp. 1379–1387, 2016.

[8] Yann LeCun, John S Denker, Sara A Solla, Richard E Howard, and Lawrence D Jackel. Optimal brain damage. In *Advances in Neural Information Processing Systems*, Vol. 2, pp. 598–605, 1989.

[9] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, pp. 1135–1143, 2015.

[10] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. *International Conference on Learning Representations (ICLR)*, 2016.

[11] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, Vol. 15, No. 1, pp. 1929–1958, 2014.

[12] Emily Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. *Computing Research Repository*, Vol. abs/1404.0736, , 2014.

[13] Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. *Computing Research Repository*, Vol. 392, , 2015.

[14] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in Neural Information Processing Systems*, pp. 3123–3131, 2015.

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[16] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of ACM Multimedia*, pp. 675–678, 2014.

[17] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Tech Report*, 2009.