ニューラルネットワークの全結合層における パラメータ削減手法の比較

佐々木健太† 佐々木勇和†† 鬼塚 真††

† 大阪大学工学部電子情報工学科 〒 565–0871 大阪府吹田市山田丘 2-1
†† 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 ディープニューラルネットワークは画像認識や音声認識などの分野で高い精度を達成している．一方で，大規模な環境で学習したディープニューラルネットワークを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は，その計算コスト，消費電力及びメモリ使用量などが技術的な課題となる．これらの課題の一因となり得るのがディープニューラルネットワークの膨大な数のパラメータである．ディープニューラルネットワークのパラメータは多くの場合で過剰であることが知られており，パラメータを削減する様々な手法が提案されている．そこで本稿では，結合単位でパラメータを削減する手法とユニット単位でパラメータを削減する手法の 2 つの既存手法に注目する．これら 2 つの既存手法を 3 種類のネットワークの全結合層に対して適用し，パラメータ削減後の認識精度の観点から比較を行う．

キーワード ニューラルネットワーク，圧縮，深層学習

1. はじめに

近年，画像認識や音声認識の分野においては多層のニューラルネットワークであるディープニューラルネットワーク（以下 DNN）が高い精度を達成しており，自動運転や機械翻訳など様々な分野への応用が期待されている．DNN の応用分野が広がるにつれて，スマートフォンやウェアラブル端末，ドローンなどのこれまで DNN が利用されてこなかった幅広い環境においても，DNN を利用したいという要求が高まりつつある．しかしながら，多くの DNN は豊富なハードウェア資源を持つ計算機上での動作を前提としており，スマートフォンなどのハードウェア資源の限られた環境上で実行する際には，計算コスト，消費電力およびメモリ使用量の大きさ等が技術的な課題となる．十分なハードウェア資源を持つサーバーにデータを送信してサーバー側で処理を行う手法も考えられるが，そのためには当然ながら通信環境が必須となり，通信帯域や遅延，プライバシーやセキュリティーなどの問題が生じる [1]．そのため DNN の用途によっては，外部にデータを送信することなくクライアント側で処理を行うことが望まれる場合があると考えられ，その場合には DNN を利用することのハードウェア要求の高さが問題となる．

DNN のハードウェア要求の高さの一因として，DNN のパラメータ数の多さが挙げられる．DNN のパラメータとはネットワークの結合の重みとバイアスの値である．表 1. に代表的なニューラルネットワークのパラメータ数を示す．表 1. に示すように，DNN は非常に多くのパラメータを持っており，DNN のモデルサイズが増大する要因となっている．

表 1 ニューラルネットワークのパラメータ数（ [2] をもとに作成）

ネットワーク名	データセット	パラメータ数	タイプ
LeNet-300-100 [3]	MNIST	266K	FFNN(注1)
LeNet-5 [3]	MNIST	431K	CNN(注2)
AlexNet [4]	ImageNet	61M	CNN
VGG-16 [5]	ImageNet	138M	CNN
GoogLeNet [6]	ImageNet	13M	CNN
ResNet-50 [7]	ImageNet	25M	CNN
ResNet-152 [7]	ImageNet	60M	CNN
SqueezeNet [8]	ImageNet	1.2M	CNN
DeepSpeech [9]	WSJ	8M	RNN(注3)
DeepSpeech-2 [10]	Baidu internal	67M	RNN
NeuralTalk [11]	Flickr-8K	6.8M	RNN+LSTM(注4)

ここで，LeNet-300-100 [3]，LeNet-5 [3] は手書き数字の認識を行うネットワークであり，AlexNet [4]，VGG-16 [5]，GoogLeNet [6]，ResNet-50 [7]，ResNet-152 [7]，SqueezeNet [8] は一般画像の認識を行うネットワークである．そして，DeepSpeech [9]，Deepseech2 [10] は音声認識を行うネットワークであり，NeuralTalk [11] は画像から説明文の生成を行うネットワークである．最近では，主に画像認識分野において GoogLeNet や SqueezeNet などのように，パラメータ数の非常に多い全結合層を極力用いず畳込み層を主体としたネットワークが多く提案されており，これらのネットワークは AlexNet や VGGNet などの畳込み層と複数の全結合層を併用するネットワークよりもパラメータ数を抑えることができる傾向にある．特に SqueezeNet はネットワークの設計段階からパラメータ数を減らすことを念

（注1）: Feedforward neural network
（注2）: Convolutional neural network
（注3）: Recurrent neural network
（注4）: Long short-term memory

頭に置いて作られたネットワークであり，AlexNet の 50 分の 1 という非常に少ないパラメータ数で AlexNet と同程度の精度を達成している．パラメータ数が多いことは，モデルを保存するために多くのストレージを必要とするだけでなく，実行時にはメモリ使用量が増加し，DRAM へのアクセスが増えることによる消費電力の増加にも繋がることが指摘されている [12] [13]．一方で，DNN のパラメータの多くが冗長であることが知られている [14]．そのため，予測の精度を維持したまま DNN のパラメータ数を削減する様々な手法が提案されている [1,12,15–20]．

本稿では，DNN のパラメータ削減手法のうち，先行研究において十分な比較が行われていない DNN の不要な結合あるいはユニットを削減（Pruning）する手法に注目する．結合単位でパラメータを削減する手法とユニット単位でパラメータを削減する手法の 2 つの既存手法 [12] [16] について，LeNet-300-100，LeNet-5，そして ImageNet のデータセットにおいて学習を行った後に，Oxford 102 Category Flower Dataset 向けに転移学習を行った CaffeNet の 3 種類のネットワークの全結合層に対して適用し，パラメータ削減後の認識精度の観点から評価を行う．

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. で本稿で用いた手法について説明し，4. において実験の設定と結果について述べる．最後に，5. において結論を述べたうえで今後の課題についても述べる．

2. 関 連 研 究

本稿が注目するニューラルネットワークの不要な結合やユニットの削減を行う手法については，主にニューラルネットワークの汎化性能を向上させる手段として古くから様々な手法が提案されており，それらは Reed らによる研究 [21] にまとめられている．Reed らの研究 [21] では，結合の削減を行う手法の多くは大きく 2 つのグループに分けられるとしている．

1 つ目のグループは学習済みのネットワークを用いて，ある結合（あるいはユニット）の削減がネットワークの損失関数にどの程度影響を与えるのかを見積もることによって削減する結合を決める手法のグループである．すべての結合についてその削減がネットワークの損失関数に与える影響を計算することは大規模なネットワークでは難しいため，様々な近似的な手法が提案されている．代表的なものとしては，ネットワークの損失関数の各重みについての二階偏微分を利用して結合の重要度を測る Optimal Brain Damage と呼ばれる手法 [22] が挙げられる．しかしながら，二階偏微分の計算は計算コストが大きく，大規模なネットワークでの利用には不向きである．もう一方のグループは，ネットワークの学習の際に損失関数に何らかのペナルティ項を追加することで学習の途中でより小規模なモデルを学習させる手法のグループである．主にネットワークの過学習を防ぐ正則化の手法として用いられる重み減衰（Weight Decay）[23] などがこちらのグループにあたる．また，遺伝的アルゴリズムを用いた手法などどちらのグループにも属さない手法についてもまとめられている．しかしながら，これらの研究はパラメータ数の削減を目的としたものではなく，ネットワークの表現能力を制限することによって過学習を防ぐことでネットワーク汎化性能を向上させること，あるいはより少ない学習データで効率的に学習を行うことなどを主要な目的としている点で，本稿の方向性と異なっている．

ニューラルネットワークの大規模化・複雑化が進んでいる状況下で，いくつかの DNN のパラメータについて非常に冗長性が高いことが Denil らによる研究 [14] によって明らかにされている．このような事実から，学習済みの DNN において不要なパラメータの削減を行うことによって DNN のモデルサイズを圧縮する手法の研究が近年盛んにおこなわれており，以下でそれらの研究の一部について述べる．

近年の大規模なネットワークに対してパラメータ数の削減を目的として結合の削減を行った研究として Han らによる研究 [12] がある．Han らによる研究では，重みの絶対値を評価指標とした結合の削減と再学習を繰り返すことで，AlexNet や VGGNet などの大規模なネットワークにおいても精度を維持したままパラメータ数を 9 倍から 13 倍圧縮することに成功している．そして，この手法を他の手法と組み合わせた Deep Compression と呼ばれる手法 [1] も Han らによって提案されており，モデルサイズをさらに圧縮することに成功している．また，圧縮したモデルを効率的に処理するためのハードウェアも Han らによって提案されている [24]．また，Han らによる結合の削減の手法では，一度削減された結合は再学習を行っても復元されることはなかったが，Guo らによる研究 [15] では，Han らの手法を発展させ再学習中に動的に結合の削減と復元を行うことによって AlexNet において精度を維持したままパラメータを 17.7 倍圧縮することに成功しており，パラメータ削減時の再学習に必要なエポック数も Han らによる手法と比べて少なくて済むことが報告されている．

また，結合単位ではなくユニット単位でパラメータの削減を行う手法も提案されている．He らによる研究 [16] では，ユニットの削減を行う際の評価指標として 3 種類の評価指標を提案しそれらの比較を行っており，He らの研究で対象としたネットワークにおいては，ユニットの出力側の結合の重みの値の L1 ノルムを用いた評価指標が最も良い性能を示したことが報告されている．また，ユニット単位でパラメータを削減するその他の手法としては，重みの値が類似しているユニット同士を統合する手法 [17] も提案されている．

ここまで述べた手法は，本稿で注目する何らかの手段でネットワークの結合，あるいはユニットの削減を行う手法のグループであったが，結合，あるいはユニットの削減を行う手法以外にもニューラルネットワークのパラメータ数を削減するための様々な手法が提案されている．例としては，Denton らによる研究 [18] が挙げられる．Denton らの研究では，ニューラルネットワークのパラメータ行列を行列分解の手法のひとつである SVD（Singular Value Decomposition）を用いて近似を行うことによってパラメータ数を削減する手法が提案されている．しかしながら，Denton らの手法は，パラメータの削減よりもニューラルネットワークによる推論にかかる時間の短縮に重点を置いており，Pruning を行う手法と比べるとパラメータ数の削減効果は大きくないことが報告されている [1]．

また，異なる結合同士で重みの値を共有する手法 [19] も Chen らによって提案されている．Chen らによる手法はネットワークの結合をハッシュ関数を用いてランダムにグループ化し，同じグループの結合で一つのパラメータを共有することによって DNN のモデルサイズを削減することに成功している．

パラメータを表現するための bit 数を削減する手法 [25] [26] についても数多くの研究がなされており，重みの値を二値化する手法 [27] なども提案されている．これらの研究の多くはネットワークの推論時だけでなく学習時の効率化も主な目的としている点で本稿で注目した学習済みのモデルを圧縮する手法と大きく異なっている．また，パラメータの数を減らすのではなくパラメータ当たりのデータサイズを削減しているという点でも本稿が注目した手法とアプローチが異なっている．

DNN のモデルを圧縮するためのその他の手法としては，訓練済みの大規模なネットワークの出力を利用して，より小規模なネットワークを学習させる蒸留と呼ばれる手法 [20] なども提案されている．

本章では DNN のモデルサイズの圧縮に関連する研究について述べたが，結合単位でパラメータを削減する手法とユニット単位でパラメータを削減する手法の比較は多くは行われていない．また，難しいタスクを学習したネットワークモデルを転移学習によってより易しいタスク向けに学習させ直した際には，ネットワークのパラメータに冗長性が生じると考えられるが，転移学習を行ったネットワークモデルに対してパラメータの削減を行った研究もほとんど行われていない．そこで，本稿では結合単位でパラメータを削減する手法とユニット単位でパラメータを削減する手法の 2 つの既存手法 [12] [16] を難しいタスクから比較的易しいタスクへの転移学習を行ったネットワークを含む 3 つのネットワークに対して適用して評価を行った．

3. 手法の説明

本稿では，ニューラルネットワークのパラメータを削減する手法の中でも，既存研究において十分な比較が行われていない結合単位でパラメータの削減を行う手法とユニット単位でパラメータの削減を行う手法に注目する．本稿では，これら 2 種類の既存手法を全結合層のパラメータがネットワーク全体のパラメータ数の大部分を占める LeNet-300-100 と LeNet-5，Oxford 102 category flower dataset 向けに学習を行った CaffeNet の 3 種類のネットワークの全結合層に対して適用し，パラメータ削減後の認識精度の観点から比較を行う．

3.1 結合単位でのパラメータの削減

本稿では結合単位でのパラメータの削減については Han らによる研究 [12] で提案されている手法を用いた．

結合単位でのパラメータの削減は図 1 に示されるように，ネットワークの不要な結合を削減することによってパラメータの削減を行う手法である．

結合単位でのパラメータの削減については Han らによる研究 [12] で提案されている手法を用いる．Han らによる研究では全結合層だけでなく畳込み層に対しても結合の削減を適用しているが，本稿では全結合層に対してのみ結合の削減を行い，

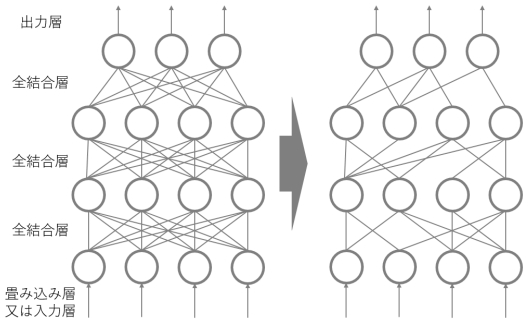

図 1　結合単位でのパラメータの削減

畳込み層の圧縮については今後の課題とする．Han らによって提案された手法の概要を図 2 に示す．Han らによって提案された手法は図 2 に示すような 3 つのステップから構成される．

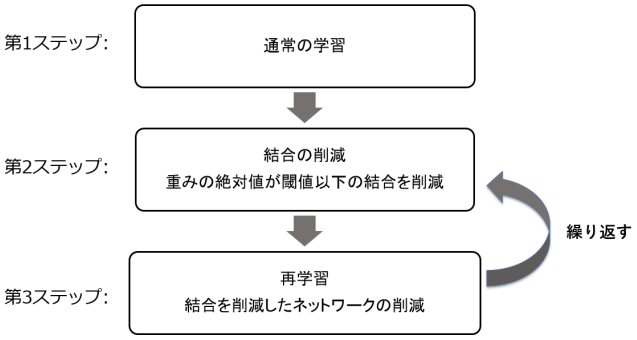

図 2　結合の削減の手順（ [12] をもとに作成）

第 1 ステップでは，通常のネットワークの学習と同様に学習を行う．第 2 ステップでは，第 1 ステップで学習した重みの絶対値が閾値よりも小さな結合を削減する．結合の削減に用いる閾値については [12] で提案されている手法に従い，$\alpha\sigma_l$ とした．但し，α はネットワーク全体のパラメータ削減率を調整するために選ぶパラメータであり，σ_l は各層の結合の重みの標準偏差である．本稿では，このパラメータの α を変えて実験を行うことで最終的なパラメータ削減率を調整して評価を行った．また，Han らによる研究においては，層によってパラメータの削減によるネットワークの認識精度への影響が異なっていることが示されており，レイヤー単位で結合の削減を行った場合の精度の低下を確認する予備的な解析を行うことによって層ごとに閾値を調整を行っているが，各層の閾値について最適な値を決定するのは非常に難しい問題であることから，本稿ではネットワークのすべての層において同じ値のパラメータ α を用いる．第 2 ステップにおいて結合の削減を行ったネットワークは予測の精度が大きく低下してしまうため，第 3 ステップにおいて結合の削減されたネットワークの再学習を行いパラメータの値を調整する．第 3 ステップの再学習においては，重みの値の初期化は行わず，前のステップで学習済みの重みの値を初期値として学習を行う．また，第 3 ステップでの学習時のデータセットとしては第 1 ステップで用いたデータセットと同じものを用いた．第 2 ステップの結合の削減と第 3 ステップの再学習を繰り返すことによって段階的に結合の数を減らすことができる．

3.2 ユニット単位でのパラメータの削減

次に，ユニット単位でのパラメータの削減手法について説明する．ユニット単位でのパラメータの削減は図 3 に示されるように，ネットワークの不要なユニットを削減することによってパラメータの削減を行う手法である．

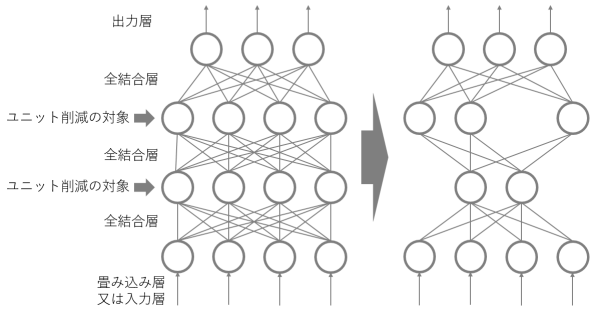

図 3　ユニット単位でのパラメータの削減

ユニット単位での結合のパラメータの削減については He らによる研究 [16] で提案されている手法を参考にする．図 4 にユニット単位でのパラメータ削減手法の概要を示す．ユニット単位でのパラメータ削減手法は図 4 に示す 3 つのステップから構成される．

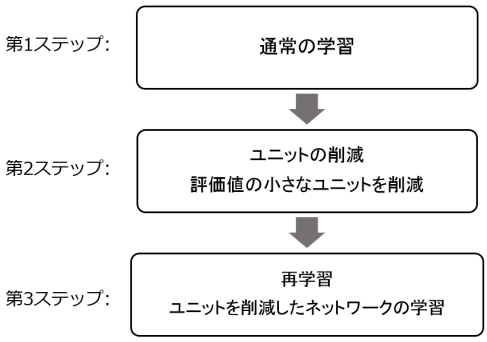

図 4　ユニットの削減の手順

ユニット単位でパラメータを削減する手法においても，結合単位でパラメータを削減する手法と同様に，まず通常のネットワークの学習と同様に学習を行う．そして，学習したネットワークにおいて，各層のユニットについてその重要度を測るための評価値を求め，評価値についてユニットのソートを行う．そして，設定したユニットの削減率に従って評価値の低いユニットからユニットの削減を行う．本稿ではユニットの削減率を変化させることで，パラメータの削減率を調整して評価を行った．ユニットの削減を行ったネットワークは精度が大きく低下するため，ユニットの削減を行った後に再学習を行う．その際には，結合単位でパラメータを削減する手法と同様に，学習済みの重みを初期値として，最初の学習で用いたデータセットと同じデータセットで学習を行う．He らによる研究ではニューラルネットワークのユニットを削減する際の評価指標として 3 種類の評価基準を比較し，He らが対象としたネットワークにおいては注目するユニットの出力側の重みの L1 ノルムを用いた *onorm* という指標が最も効果的であったとしている．l 番目の層の i 番目のユニットの *onorm* は以下のように定式化されている．

$$onorm(l,i) \;=\; \frac{1}{N^{l+1}} \sum_{j=1}^{N^{l+1}} |w_{ij}^{l+1}| \qquad (1)$$

ここで N^l は l 番目の層の出力ユニットの数を表し，w_{ij}^l は l 番目の層の結合の重みを表す．本稿ではユニットを削減する際の評価基準としてこの *onorm* を採用した．He らはユニットの削除は他のユニットの *onorm* に影響を与えることを指摘しつつも，*onorm* による評価を一度だけ行ってユニットを *onorm* 順にソートしてより *onorm* の小さいユニットを一度に削減した後に再学習を行っている．本稿でも同様に *onorm* による評価と削減を一度のみ行う．本稿においては，ユニットの削減は図 3 に示すようにネットワークの全結合部分の両端である入力と出力に相当するユニットに対しては適用しない．

4. 実　　験

4.1 データセット

まず，LeNet-300-100 及び LeNet-5 での実験においては手書き数字の画像データセットである MNIST データセット(注5)を用いた．MNIST データセットは 10 カテゴリのラベル付けがなされた 28×28 ピクセルのグレースケール画像のデータセットであり，6 万枚の訓練用データセットと 1 万枚のテスト用データセットで構成される．

CaffeNet においては，Nilsback ら [28] によって作成された Oxford 102 Category Flower Dataset を用いた．このデータセットは 102 カテゴリのラベル付けがなされた花のカラー画像のデータセットであり，1,020 枚の訓練用データセットと 6,149 枚のテスト用データセット，そして 1,020 枚の検証用データセットで構成されている．本稿では，後述の Caffe Model Zoo に記載されている学習方法に従い，訓練用データセットとテスト用データセットを入れ替えて実験を行い（すなわち，6,149 枚のデータを訓練用に，1,020 枚のデータをテスト用に用いた），検証用データセットについては利用しなかった．

4.2 パラメータ削減の対象としたネットワーク

4.2.1 LeNet-300-100

LeNet-300-100 は，3 つの全結合層のみで構成される比較的構造の単純なネットワークである．学習率や正則化項の係数などの学習時のパラメータについては Caffe の MNIST チュートリアルを参考に設定した．異なる初期値で 5 回学習を行い，テスト用データ 1 万件において最も高い精度を達成したモデルをパラメータ削減の対象とした．LeNet-300-100 の各層のバイアスを除くパラメータ数を表 2 に示す．

4.2.2 LeNet-5

LeNet-5 は全結合層のみで構成されている LeNet300-100 と異なり，畳込み層と全結合層を併用したネットワークである．各層のバイアスを除くパラメータ数を表 3 に示す．学習率など

(注5)：http://yann.lecun.com/exdb/mnist/

表 2　LeNet-300-100 の各層のバイアスを除くパラメータ数

	パラメータ数	ネットワーク全体に占める割合
全結合層 1	235.2K	88.35%
全結合層 2	30K	11.27%
全結合層 3	1K	0.38%
合計	266.2K	

の学習時のパラメータについては Caffe の MNIST チュートリアルを参考に設定した．LeNet-5 においても LeNet-300-100 と同様に異なる初期値で 5 回学習を行い，テスト用データにおいて最も高い精度を達成したモデルをパラメータ削減の対象とした．

表 3　LeNet-5 の各層のバイアスを除くパラメータ数

	パラメータ数	ネットワーク全体に占める割合
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%
合計	430.5K	

4.2.3 Oxford 102 Category Flower Dataset を学習させた CaffeNet

CaffeNet とは，Krizhevsky らによって提案され，2012 年に画像認識のコンペティションである ImageNet Large Scale Visual Recognition Challenge(ILSVRC) [29] において優勝した AlexNet に対して Pooling 層と Normalization 層を入れ替えるなどの若干の変更を行ったネットワークであり，Caffe によって提供されている．本稿では，ImageNet のデータセットではなく，Oxford 102 Category Flower Dataset を学習させた CaffeNet に対してパラメータの削減を行った．Oxford 102 Category Flower Dataset を学習する CaffeNet は Caffe Model Zoo(注6)で提供されている．CaffeNet の出力層のユニットの数が ImageNet のタスクに合わせて 1000 ユニットなのに対して，本稿で用いたネットワークは Oxford 102 Category Flower Dataset に合わせて出力ユニットの数が 102 ユニットに変更されている．本稿では，Caffe Model Zoo で提供されているモデルの学習方法 (注7)に従って ImageNet データセットで学習を行った CaffeNet を Oxford 102 Category Flower Dataset 向けに fine-tuning による転移学習を行った．手順としては，まず 1,000 カテゴリのラベル付けがなされた ImageNet の ILSVRC2012 データセットを用いて CaffeNet の学習を行う．本稿では，ILSVRC2012 データセットでの学習は実際には行わず，学習済みのモデルをダウンロードして利用した (注8)．この学習済みモデルは ILSVRC2012 の検証用データセットにおいて 57.4% の認識精度（top-1）を達成している．次に，ネットワークの最後の全結合層の出力ユニットの数を Oxford 102 Category Flower Dataset に合わせて 1,000 から 102 に変更し，この層のパラメータのみを初期化する．最後に，Oxford 102 Category Flower Dataset の訓練用データ 6,149 枚を用いてバッチサイズ 50 で 5 万イテレーションの学習を行った．この時，ネットワークの最後の全結合層は他のレイヤーの 10 倍の学習率で学習を行った．学習率などのパラメータの設定についても Caffe Model Zoo の記述に従った．本稿では Caffe Model Zoo の手法と異なり，ネットワークの過学習を防ぐために学習時に全結合層のユニットを確率的に無視して学習を行う手法である DropOut [30] は適用しなかった．初期値を変えて 3 回学習を行い，テスト用データにおいて最も高い精度を達成したモデルをパラメータ削減の対象とした．このモデルのテストデータにおける認識精度 (top-1) は 91.82% であった．本稿で用いたネットワークの各層のバイアスを除くパラメータ数を表 4 に示す．

表 4　CaffeNet(Modified) の各層のバイアスを除くパラメータ数

	パラメータ数	ネットワーク全体に占める割合
畳込み層 1	35K	0.06%
畳込み層 2	307K	0.54%
畳込み層 3	885K	1.54%
畳込み層 4	664K	1.16%
畳込み層 5	442K	0.77%
全結合層 1	38M	65.91%
全結合層 2	17M	29.29%
全結合層 3	418K	0.73%
合計	57M	

4.3 実験設定

本稿では，ディープラーニングフレームワークである Caffe [31] とその Python 向けインターフェースである PyCaffe を用いて実験を行った．

本稿では，すべてのネットワークの学習・再学習において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．

まず，LeNet-300-100 において結合の削減を行う際には，削減の閾値を決めるパラメータ α を 0.6 から 2.8 の範囲で変えて実験を行い，結合の削減と再学習 10 万イテレーションのセットをそれぞれ 5 回ずつ繰り返した後の最終的なパラメータ削減率と精度を評価した．ユニットの削減においては，各層について全てのユニットを He らによる研究で用いられた評価指標である *onorm* の低いものから削減し，その後 10 万イテレーションの再学習を行うことによってユニットの削減を行った．LeNet-5 においても同様に，結合の削減を行う際には，削減の閾値を決めるパラメータ α を 0.7 から 2.8 の範囲で変えて実験を行い，結合の削減と再学習 10 万イテレーションのセットをそれぞれ 5 回ずつ繰り返した後の最終的なパラメータ削減率と精度を評価した．LeNet-5 においても LeNet-300-100 と同様にバッチサイ

(注6)：https://github.com/BVLC/caffe/wiki/Model-Zoo
(注7)：https://gist.github.com/jimgoo/0179e52305ca768a601f
(注8)：https://github.com/BVLC/caffe/tree/master/models/bvlc_reference_caffenet

ズは 64 を用いた．ユニットの削減においては，削減の対象となる層のユニットを評価指標である *onorm* が低いユニットから削減し，その後 10 万イテレーションの再学習を行うことによってユニットの削減を行った．LeNet-300-100 および LeNet-5 の再学習を行う際の学習率や正則化項の係数などのパラメータについては，初期の学習を行った際に用いた Caffe の MNIST チュートリアルのパラメータを参考に設定した．Oxford 102 Category Flower Dataset を学習させた CaffeNet において結合の削減を行う際には，削減の閾値を決定する際のパラメータの α を 0.5 から 2.6 の範囲で変えて実験を行い，結合の削減と再学習 5 万イテレーションのセットをそれぞれ 3 回ずつ繰り返した後の最終的なパラメータ削減率と認識精度を評価した．ユニットの削減においては，各層について全てのユニットを評価指標である *onorm* が低いユニットから削減し，その後 5 万イテレーションの再学習を行うことによってユニットの削減を行った．その際，削減の対象となる全てのユニット層においてユニットの削減率が等しくなるようにユニットの削減を行った．Oxford 102 Category Flower Dataset を学習させた CaffeNet の再学習を行う際の学習率や正則化項の係数などのパラメータについては，初期の学習を行った際に利用した Caffe Model Zoo のパラメータを参考に設定した．

4.4 評価結果

3 つの全結合層から構成される LeNet-300-100 に対して結合の削減とユニットの削減それぞれの手法でパラメータの削減を行なった場合の全結合層のパラメータの削減率とテストデータにおける認識精度の関係を図 5 に示す，LeNet-300-100 のパラメータ削減前のテストデータにおける認識精度は 98.34% であり，この値を「baseline」としてプロットしている．「baseline」については，パラメータの削減を行っていないが比較のため図中に直線でプロットしている．また，学習済みのネットワークのパラメータを削減するのではなく，最初からパラメータ数の少ないネットワークを学習させた際のパラメータ削減率と精度の関係も「小規模モデル」としてプロットしている．その際には，ユニット単位でのパラメータの削減を行ったネットワークとネットワークの構造が同じになるようにネットワークの定義し，異なる初期値で 5 回学習を行い最も精度の高かったものをプロットした．

LeNet-300-100 においては入力側から見て 1 つ目の全結合層がネットワーク全体のパラメータの 90% 近くを占めており，2 つめの全結合層のユニットの削減はパラメータ削減率及び削減後の精度に与える影響が小さいことから，LeNet-300-100 におけるユニット単位でのパラメータの削減の手法においては，2 つめの全結合層のユニットの削減率を 70% に固定し，1 つ目の全結合層のユニットの削減率を変化させることによってネットワーク全体での削減率を調整した．

次に，LeNet-5 に対して結合の削減とユニットの削減それぞれの手法でパラメータの削減を行なった場合の全結合層のパラメータの削減率と認識精度の関係を図 6 に示す，パラメータ削減前のネットワークの認識精度を「baseline」としてプロットし，LeNet-300-100 と同様の手順で最初からパラメータ数の少ないネットワークを学習させた際のパラメータ削減率と精度の関係も「小規模モデル」としてプロットしている．LeNet-5 のパラメータ削減前の認識精度は 99.10% であった．

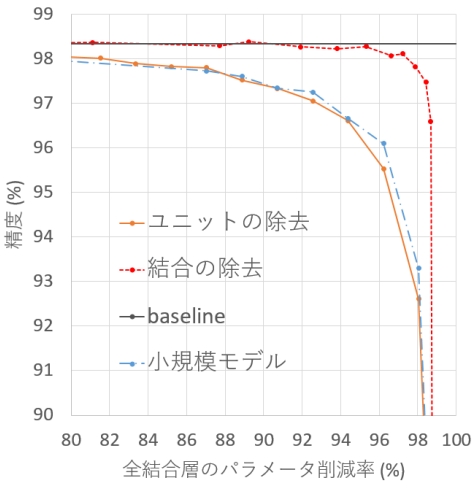

図 5　LeNet-300-100 におけるパラメータ削減の結果 (注9)

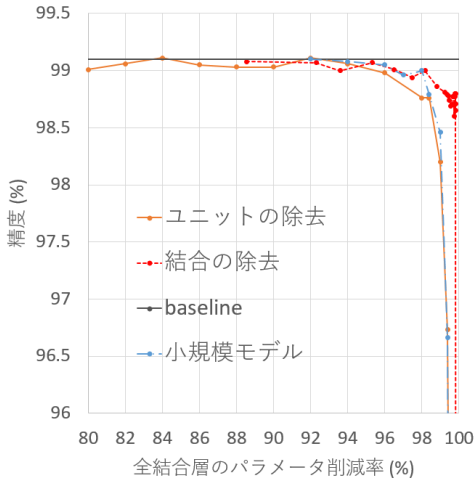

図 6　LeNet-5 におけるパラメータ削減の結果 (注9)

LeNet-300-100 と LeNet-5 においては，いずれのネットワークにおいても結合単位でのパラメータの削減はユニット単位でのパラメータの削減に比べて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に成功していることが分かった．削減後のネットワークにおいて結合単位での削減はユニット単位でのパラメータの削減と比べてより多くのユニットを保持できるために，より高い表現力を維持できる点が一因として考えられる．特に，ユニット単位でのパラメータの削減に関しては，大規模なモデルのパラメータを削減するのではなく，最初からパラメータ数の少ないモデルを学習させた場合と認識精度にほとんど差が無い結果となった．MNIST のデータセットにおいてはネットワークの初期値によらず，小規模なネットワークでも十分学習を行うことが可能であり，大規模なネットワー

(注9)：ただし「baseline」はパラメータ削減を行っていないが比較のため直線でプロットしている

クで学習した重みを利用することによる恩恵が得られなかったためであると考えられる．LeNet-5 での実験結果においてLeNet-300-100 での実験結果よりも各手法間で性能の差が小さいのは，LeNet-5 においては畳込み層のパラメータが削減されずに残っているため全結合層のパラメータの削減がネットワークの精度に与える影響が比較的小さいためであると考えられる．

最後に，Oxford 102 category flower dataset 向けに学習を行った CaffeNet に対して結合の削減の手法とユニットの削減の手法をれぞれを用いてパラメータの削減を行った場合の全結合層のパラメータ削減率と認識精度の関係を図 7 に示す．パラメータ削減前のネットワークの認識精度を baseline としてプロットしている．CaffeNet のパラメータ削減前の Oxford 102 category flower dataset のテストデータにおける認識精度（top-1）は 91.82% であった．CaffeNet においては ImageNet で学習したモデルからの転移学習を行っているため，小規模なモデルを最初から学習した場合についての実験は行っていない．また，結合単位でのパラメータの削減を行った際のパラメータ α と各層のパラメータ削減率とネットワークの認識精度は表 5 のようになった．

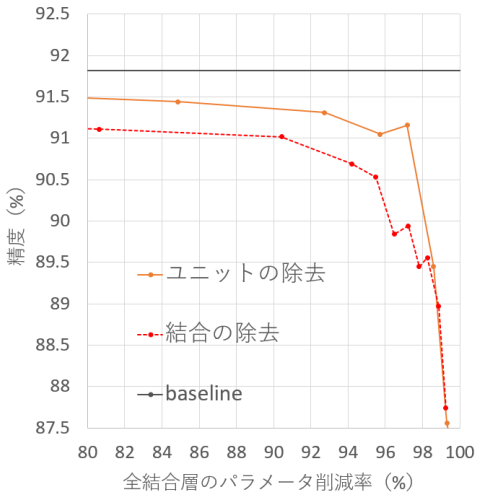

図 7　CaffeNet におけるパラメータ削減の結果 (注9)

Oxford 102 category flower dataset 向けに学習を行った CaffeNet においては，結合単位でのパラメータ削減がユニット単位でのパラメータ削減にやや劣る結果となった．LeNet-300-100 及び LeNet-5 での実験と異なる結果が出た要因としては Oxford 102 category flower dataset 向けに学習を行った CaffeNet においてはパラメータ削減前のネットワークにおけるパラメータの冗長性が非常に大きいという点や，結合の削減と再学習のセットの繰り返しの回数，各層のパラメータ削減率などいくつかの要因が考えられるため，この点についてはさらなる検証を必要とする．

また，CaffeNet における実験においてはパラメータ削減後の予測精度がパラメータ削減前のモデルをやや下回る結果となった．再学習時のハイパーパラメータの設定が原因である可能性も考えられるが，再学習に用いたデータセットのサイズがパラメータを削減したモデルの学習を行うのに十分でなかった可能性も考えられる．データセットのサイズが精度の低下の原因であると仮定すると，大規模な学習データが存在する転移学習前のデータセットにおいてパラメータの削減と再学習を行った後で転移学習を行うことによってより高い予測精度を維持できると期待できる．

表 5　CaffeNet における結合単位でのパラメータ削減時の各層のパラメータ削減率

	パラメータ削減率（%）				
α	全結合層 1	全結合層 2	全結合層 3	全結合層全体	精度 (%)
0.5	49.54	47.13	57.21	48.86	91.66
0.6	57.54	55.14	64.99	56.86	91.39
0.8	70.96	69.04	76.44	70.42	91.16
1.0	80.97	79.77	83.74	80.62	91.11
1.3	90.51	90.27	90.01	90.43	91.02
1.5	94.16	94.28	92.45	94.18	90.69
1.6	95.44	95.64	93.38	95.49	90.53
1.7	96.41	96.66	94.20	96.47	89.84
1.8	97.17	97.43	94.88	97.23	89.94
1.9	97.75	98.02	95.45	97.81	89.45
2.0	98.20	98.45	95.94	98.26	89.56
2.2	98.80	99.03	96.74	98.85	88.97
2.4	99.19	99.40	97.41	99.24	87.74
2.6	99.46	99.64	97.95	99.50	87.08

5. 結　　論

本稿では，ニューラルネットワークの結合単位でパラメータを削減する手法とユニット単位でパラメータを削減する手法の 2 つの既存手法を LeNet-300-100，LeNet-5，そして Oxford 102 Category Flower Dataset 向けに学習を行った CaffeNet の 3 種類のネットワークの全結合層に対して適用し評価を行った．その結果，LeNet-300-100 及び LeNet-5 においては結合単位でのパラメータを削減する手法がより優れた性能を示したのに対して，Oxford 102 Category Flower Dataset 向けに学習を行った CaffeNet においてはユニット単位でのパラメータの削減を行う手法がやや優れた性能を示す結果となり，対象とするネットワーク等の条件によって良い性能を示すパラメータ削減手法が異なることが明らかとなった．また，本稿においては各層ごとのパラメータの削減率の調整は基本的に行っていないが，今後の課題として各層のパラメータの冗長性に応じてパラメータの削減率を調整することも検討したい．その際には，各層のパラメータの削減率をどのように決定するのかが課題となる．

文　　献

[1] S. Han, H. Mao, and W.J. Dally, “Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding,” International Conference on Learning Representations (ICLR), 2016.

[2] S. Han, J. Pool, S. Narang, H. Mao, S. Tang, E. Elsen, B. Catanzaro, J. Tran, and W.J. Dally, “DSD: regularizing deep neural networks with dense-sparse-dense training flow,” Computing Research Repository, vol.abs/1607.04381, 2016.

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol.86, no.11, pp.2278–2324, 1998.

[4] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems, pp.1097–1105, 2012.

[5] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Computing Research Repository, vol.abs/1409.1556, 2014.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," Proceedings of CVPR, pp.1–9, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Computing Research Repository, vol.abs/1512.03385, 2015.

[8] F.N. Iandola, M.W. Moskewicz, K. Ashraf, S. Han, W.J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size," Computing Research Repository, vol.abs/1602.07360, 2016.

[9] A.Y. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A.Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," Computing Research Repository, vol.abs/1412.5567, 2014.

[10] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A.Y. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A.Y. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, and Z. Zhu, "Deep speech 2: End-to-end speech recognition in english and mandarin," Computing Research Repository, vol.abs/1512.02595, 2015.

[11] A. Karpathy, and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.3128–3137, 2015.

[12] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," Advances in Neural Information Processing Systems, pp.1135–1143, 2015.

[13] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International, pp.10–14, 2014.

[14] M. Denil, B. Shakibi, L. Dinh, M. Ranzato, and N. de Freitas, "Predicting parameters in deep learning," Advances in Neural Information Processing Systems, pp.2148–2156, 2013.

[15] Y. Guo, A. Yao, and Y. Chen, "Dynamic network surgery for efficient dnns," Advances In Neural Information Processing Systems, pp.1379–1387, 2016.

[16] T. He, Y. Fan, Y. Qian, T. Tan, and K. Yu, "Reshaping deep neural network for fast decoding by node-pruning," 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.245–249, 2014.

[17] S. Srinivas, and R.V. Babu, "Data-free parameter pruning for deep neural networks," Computing Research Repository, vol.abs/1507.06149, 2015.

[18] E. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus, "Exploiting linear structure within convolutional networks for efficient evaluation," Computing Research Repository, vol.abs/1404.0736, 2014.

[19] W. Chen, J.T. Wilson, S. Tyree, K.Q. Weinberger, and Y. Chen, "Compressing neural networks with the hashing trick," Computing Research Repository, vol.abs/1504.04788, 2015.

[20] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," arXiv preprint arXiv:1503.02531, 2015.

[21] R. Reed, "Pruning algorithms-a survey," IEEE transactions on Neural Networks, vol.4, no.5, pp.740–747, 1993.

[22] Y. LeCun, J.S. Denker, S.A. Solla, R.E. Howard, and L.D. Jackel, "Optimal brain damage," Advances in Neural Information Processing Systems, vol.2, pp.598–605, 1989.

[23] D.C. Plaut, S.J. Nowlan, and G. Hinton, "Experiments on learning by back propagation," , 1986.

[24] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M.A. Horowitz, and W.J. Dally, "EIE: efficient inference engine on compressed deep neural network," Computing Research Repository, vol.abs/1602.01528, 2016.

[25] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," Computing Research Repository, vol.392, 2015.

[26] V. Vanhoucke, A. Senior, and M.Z. Mao, "Improving the speed of neural networks on cpus," Proceedings of Deep Learning and Unsupervised Feature Learning NIPS Workshop, vol.1, p.4, 2011.

[27] M. Courbariaux, Y. Bengio, and J.P. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," Advances in Neural Information Processing Systems, pp.3123–3131, 2015.

[28] M.E. Nilsback, and A. Zisserman, "Automated flower classification over a large number of classes," Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing, 2008.

[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," International Journal of Computer Vision, vol.115, no.3, pp.211–252, 2015.

[30] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," Journal of Machine Learning Research, vol.15, no.1, pp.1929–1958, 2014.

[31] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," Proceedings of ACM Multimedia, pp.675–678, 2014.