Deep Learningのための高効率化技術

Technologies for Practical Application of Deep Learning

● 池　敦　● 石原輝雄　● 富田安基　● 田原司睦

あらまし

近年，Deep Learningと呼ばれる機械学習手法が注目されている。Deep Learningは，人間が特徴量をプログラミングする既存手法を圧倒する認識精度を実現できることから，世界的に注目され研究開発が加速している。ニューラルネットワークは，より高精度な認識を実現するため，その層数は年々深くなってきているが，こうした深層化に伴う課題も見えてきている。それは，ニューラルネットワークの学習時間の増大と，メモリサイズ制限によって生じるネットワーク規模の制約である。

本稿では，これらの課題を解決するための技術を紹介する。一つ目は，Deep Learningの高速化を可能にする分散並列技術であり，二つ目は，ニューラルネットワークの大規模化を可能にするメモリ使用の効率化技術である。そして三つ目として，使用するデータサイズそのものを必要最小限自動的にチューニングする技術により，高速化とメモリ効率化の双方を更に加速する専用ハードウェアアーキテクチャーについても解説する。

Abstract

Deep learning, a machine learning method, is attracting more and more attention. Deep learning has gained worldwide attention, and its research and development have accelerated as it achieves recognition accuracy that far surpasses that of conventional methods of extracting features manually. Two issues are affecting its practical application: lengthy training and limited graphical processing unit (GPU) memory. As neural networks are being enlarged to enable higher recognition accuracy, these two issues are becoming more and more serious. In this paper, we introduce three technologies targeting them: distributed parallel processing for faster training, memory optimization for increased GPU memory, and a dedicated hardware engine architecture for data size reduction.

まえがき

今や人工知能の中心的な技術となったDeep Learningは，基本的には3層以上のニューラルネットワーク（Deep Neural Network）を使った機械学習アルゴリズムの総称である。

Deep Learningを実行するためには，三つの要素が必要であると言われている。第一に膨大な学習データ，第二に深いニューラルネットワークを学習できるアルゴリズムの開発，そして第三に，その深いニューラルネットワークの学習を可能にする計算リソースの高性能化である。

Deep Learningの研究・開発には，一般に非常に長い期間が必要となる。これは，ネットワーク構造の決定や学習時の最適化方法に対する厳密な理論がいまだに構築されていないためである。これは，これまで蓄積した経験やノウハウを基に，膨大な解空間の中から試行錯誤を重ねながら探して回る必要があるからとも言えるが，学習処理そのものも膨大な演算を必要とするからである。

毎年，世界的な画像種別コンペティションであるImageNet Large Scale Visual Recognition Challenge（ILSVRC）が行われている。2012年に，カナダ・トロント大学のGeoffrey Hinton教授のチームが，1,200万枚の画像を1,000カテゴリーに分類する画像認識チャレンジに初めてDeep Learningを適用した。[1] その結果，それまでの25.8％だった誤認識率を一気に10％近くも改善させることに成功してトップを飾り，これが今日の第3次Deep Learningブームの火付け役ともなった。それ以降，ILSVRCではトップは全てDeep Learningを使用したものとなった。また毎年改良が続き，2015年にはついに人間の誤認識率である5.1％を下回り，2016年には3％を切るに至った。

Deep Learningの三つの課題

このようなDeep Learningの進歩に伴い，画像認識の誤認識率は著しく低下したが，必要となる学習時間は膨大なものとなっている。2012年のAlexNetでは，1～2週間かけて128万枚の画像を学習させていたが，最近ではネットワーク構成が更に複雑化している。例えば，Microsoft社の「MS ResNet」のような150層という非常に深いニューラルネットワークも提案され，[2] 必然的に学習評価に必要な計算量が引き上げられてしまうため，大きな課題となっている。更に最近では，n個のネットワークを組み合わせて多数決を採るアンサンブルという手法も主流となり，学習そのものがn倍となるなど，計算量の増大に拍車がかかっている［課題1］。

一方，Deep Learningにおいて標準で使われるGPU（Graphics Processing Unit）のメモリサイズは，128 Gバイトを超えることができるCPUに比べるとハイエンドの「Tesla」であっても，8～16 Gバイト程度と限られている。これが巨大なネットワークを使う上での別の大きな課題となってきた［課題2］。

例えば，AlexNetでは0.6 Gバイト(注1) だった必要なメモリ量が，MS ResNetでは8 Gバイト(注1) と，わずか3年で10倍以上に激増している。GPUの搭載メモリ量はそこまで一気には増やせないため，結果として利用可能なメモリ量の範囲内でネットワークを設計するしかなく，Deep Learningの進歩に蓋をする要因にもなりかねない事態となってきている。

さて，Deep Learningでは結果としての認識精度が最も重要であるが，途中の演算に使われるデータの演算精度については，必ずしも最高精度が必要とされない。一般的には32ビットの浮動小数点数が使われるが，最近では16ビットの浮動小数点数や，16ビット/8ビットの固定小数点数を使用しても，それほど認識精度は劣化しないという研究報告が多い。そこで，計算リソース側もこうした16ビット/8ビットの演算機構を追加・強化するものが現れてきている。一方で，単純に演算精度を落とせば認識精度も落ちるため，ネットワークの設計などとのトレードオフで考える必要があり，結果としてここでもより多くの学習が必要となってしまう［課題3］。

このように，年々進歩し，大規模化するDeep Learningであるが，学習時間の長さと扱えるニューラルネットワーク規模への対応がもう一つの大きな課題となっている。富士通では，これまでスーパーコンピュータ「京」(注2) や，その後継プロジェ

（注1） いずれもバッチサイズ＝8の場合。
（注2） 理化学研究所と富士通が共同開発したスーパーコンピュータ。「京」は理化学研究所の登録商標。

クトであるフラグシップ2020などのスーパーコンピュータ開発を手掛けてきている。富士通研究所では，その技術の一部をDeep Learningにも応用することで，これらの課題を大きく改善することに成功した。

以降では，これら課題の解決手法としての，Deep Learningを高効率化する技術について述べる。

Deep Learning高速化のための分散並列技術

Deep Learningでは，ニューラルネットワークの特性を表す近似式を多数組み合わせることで，高度な認識などが実現されている。近似式に出てくる（重みやウェイトと呼ばれる）係数は，学習によって決定される。

一般的な学習は，フォワード処理とバックワード処理の繰り返しで行われる。フォワード処理とは，入力（例えば画像）から出力（画像の認識結果）を生成するまでの一連の流れのことであり，一般にこの処理を推論とも呼ぶ。これに対してバックワード処理とは，この出力を正解と比較し，その差分を用いて各層で使われる近似式の係数を更新していく流れのことである。

Deep Learningで使用される係数の数は膨大であるため，係数を正しく求める学習には数百万個から数千万個に及ぶ大量のデータを使用することがある。このため，1回の学習に数日から数週間という長い時間を要することがDeep Learningを利用する上での課題となっていた。

現在主流の学習方法では，このデータを数十個から数百個ずつのデータのかたまり（ミニバッチ）に分け，ミニバッチ単位で学習を進めていく。係数の修正量は個々のデータを処理することで求められるが，ミニバッチ処理では，ミニバッチ内で得られた修正量の平均値を全体の修正量とする。

ミニバッチ中のデータを，複数のコンピュータで分担して処理する方式をデータ並列という。データ並列では，コンピュータ1台あたりの計算時間を減らすことで高速化を実現する。このとき，修正量の平均値を求めるためにコンピュータ間で結果を共有する。共有のためのコンピュータ間の通信は，分割処理のオーバーヘッドである。したがって，分割処理による計算時間の削減分よりも通信時間が長くなると，分割によってかえって処理時間が長くなる。このため，コンピュータ間の通信時間の削減と，通信処理を計算処理と同時に行うことによる通信時間の隠蔽が重要となる。

筆者らは，これらの課題に対し以下のような方式で対応した。[3]

（1）バックワード処理と通信処理の並列化

Deep Learningで用いられるニューラルネットワークは多層であり，係数の修正量は各層のバックワード処理ごとに得られるため，バックワード処理中に通信処理を行うことが可能となる。最近の並列Deep Learningフレームワークには取り入れられている機能である。

（2）フォワード処理と通信処理の並列化

各層のフォワード処理は，その層の係数の修正が終わっていれば実行可能であるため，フォワード処理中に通信処理を行うことが可能となる。

（3）通信処理のパイプライン化

Deep learningでは大量の計算を行うために，GPGPU（General-Purpose computing on Graphics Processing Units）が利用されることが多い。筆者らは，可能な限りGPUの計算処理を止めないよう，通信処理はCPU側に実装した。通信処理の順序としては，

1. GPUからCPUへ修正用データ転送
2. CPUによる通信処理
3. CPUからGPUへの修正用データ転送

となる。これらがパイプライン処理されるように，修正用データを分割して処理を行う。また通信処理には，修正用データの平均値を求める処理が含

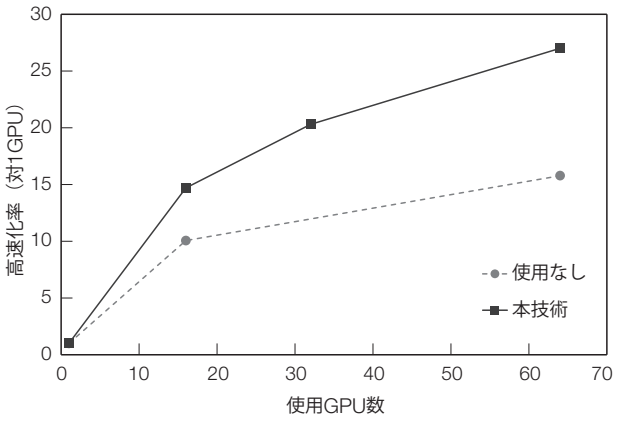

図-1　AlexNetの並列化による高速化効果の比較

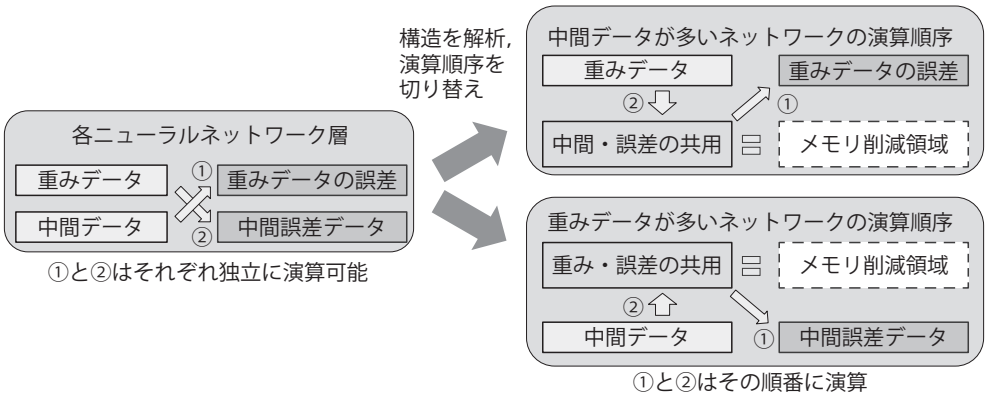

図-2　メモリ効率化技術

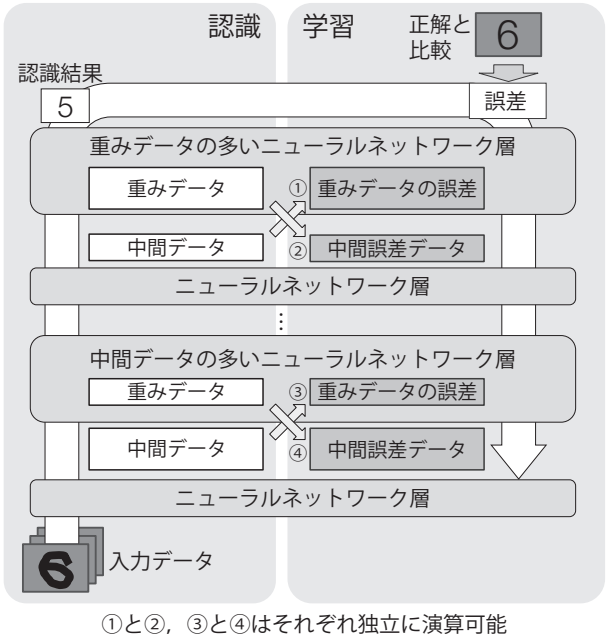

図-3　従来の演算フローとメモリ使用

まれており，通信処理中の計算量も多い。このため，CPU処理もスレッド並列化やSIMD（Single Instruction/Multiple Data）並列化を適用して高速化している。

64個のGPUを使用した場合において，これらの技術により学習速度が1.8倍に向上することを確認した（**図-1**）。

ニューラルネット大規模化のための メモリ効率化技術

Deep Learningでは，膨大な演算処理を行うためにGPUが用いられている。GPUとCPU間の通信バンド幅は少ないため，GPUの高速な演算性能を活用するには，一連の演算に使用するデータを可能な限りGPUの内部メモリに格納しなければならない。しかし，ハイエンドのGPUに搭載できるメモリ量（8 ～ 16 Gバイト）はCPUに搭載できるメモリ量（数百Gバイト）よりも少ないため，高速に学習できるニューラルネットワークの規模は制限されるという課題がある。

この課題を解決するために，1台のGPUで計算できるニューラルネットワークの規模を拡大するメモリ効率化技術を開発した。ニューラルネットワークでは，各層での学習処理において，誤差逆伝搬法のために行う重みデータから中間誤差データを求める演算と，次の重みデータを更新するために行う中間データから重みデータの誤差を求める演算の両方を行う必要がある。

本効率化技術では，これらの演算が独立して実行できることに着目した。すなわち，学習の開始時にニューラルネットワークの各層の構造を解析し，中間誤差データと重みデータの誤差を格納するために確保されるメモリ量を見積もる。次に，より大きなデータを配置するメモリ領域が再利用できるように演算の処理順序を切り替えることにより，メモリ使用量を削減する（**図-2**）。[4]

図-3は，ニューロンデータの多い層（例：畳み込み層）や，重みデータの多い層（例：全結合層）などを含む，ニューラルネットワークにおける従来の演算フローとメモリ使用を示す。手書き入力データ「6」を入力とし，ニューラルネットの認識結果として「5」と誤認識され，それを正解データと比較した誤差から，ニューラルネットの重みを

更新している。**図-4**は，メモリ効率化技術における演算フローとメモリ使用を例として示している。

従来は，認識と学習に必要なメモリ領域を全て確保しているのに対し，本技術では演算順序を数字の順に行うことで，各層でメモリ使用量の大きい方の演算を先に行う。その演算を先に終わらせることで，そのメモリ領域を再利用し，使用メモリ量を削減している。

このメモリ効率化技術を，分散並列化技術と同様にオープンソースソフトウェアのDeep Learningフレームワーク「Caffe」に実装し，GPUの内部メモリ使用量を計測した。研究分野で広く使用

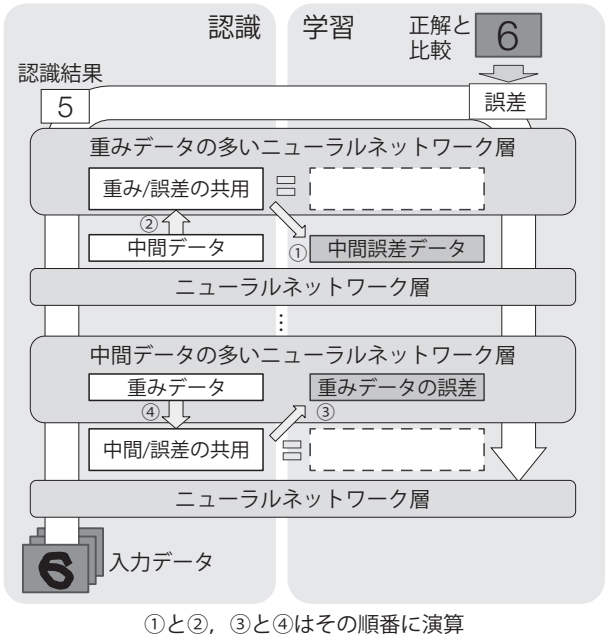

図-4　メモリ効率化技術の演算フローとメモリ使用

されている画像認識用ニューラルネットワークAlexNetやVGGNetを用いた評価では，本技術適用前と比較して40％以上のメモリ使用量が削減できることを確認した。本技術により，削減できたメモリ領域をニューラルネットワークのレイヤー数やニューロン数の増加に使用することで，GPU 1台あたり最大で約2倍の規模のニューラルネットワークを学習することが可能になった。また，1回の学習処理における処理枚数を倍に増やすことで，画像認識の精度が4％向上することも確認した。[5]

Deep Learningの専用ハードウェア技術

推論時の認識率を高くするために，ニューラルネットワークの規模（層数）は大きくなる傾向にあり，学習処理量も増加傾向にある。学習処理はGPGPUで行うのが一般的であるが，1チップあたり200 ～ 300 Wの電力を消費する。このため，これをクラスタ並列化して高速化を図る場合，利用できる電力量から使用可能なGPUの数に制限を受け，電力性能が全体の処理性能を決める要素になっている。例えば，2倍の電力性能を持つチップであれば，利用可能な電力が同じであっても2倍の処理が可能になる。

現在の学習処理では，演算精度の不足などの点を考慮しなくてもよいなどの理由から，32ビット浮動小数点数で演算を行うことが主流になっている。「Deep Learning高速化用分散並列技術」の章で述べたように，マルチチップでの並列化処理ではチップ間の通信が必要になるが，この通信時間を短くすることは並列処理の高速化を可能にする。

このような理由から，今回Deep Learningにおけ

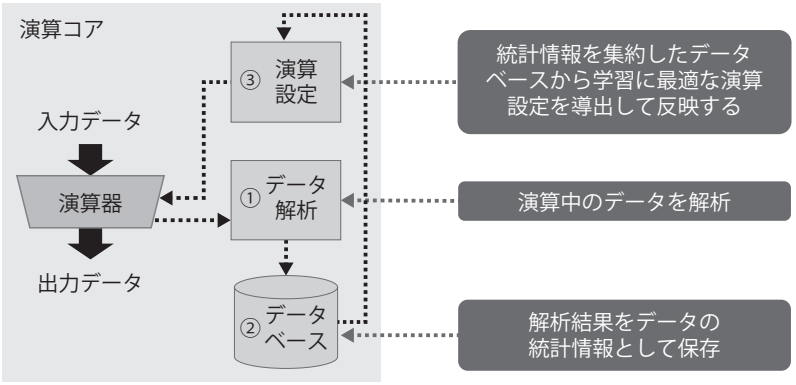

図-5　演算コアによる演算精度の向上

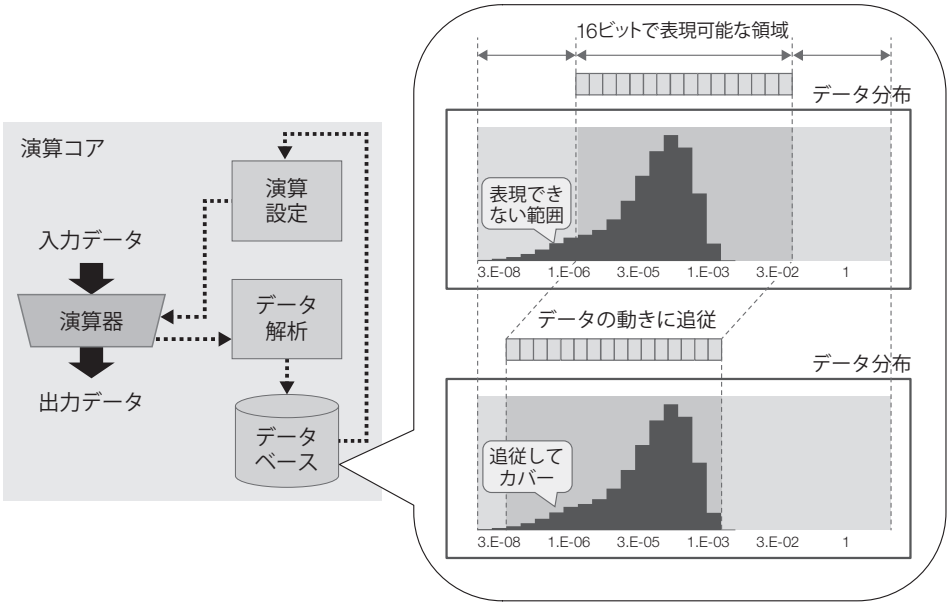

図-6　統計情報を用いた演算設定の最適化

る学習処理の電力性能向上を狙い，より少ないビット幅で演算精度を確保するハードウェアを開発した。[6] この学習用ハードウェアの演算コア（**図-5**）には，①演算中のデータを解析するブロック，②解析したデータの分布を保存するデータベース，③演算の設定を保持するブロックを持つ。データ解析ブロックでは，Deep Learningの学習中に演算器の出力データをリアルタイムに解析して，データ分布を表す統計情報としてデータベースに保存する。そして，その分布からDeep Learningの学習精度を向上させるために，十分な演算精度を保つことができるように学習に最適な設定をして演算を進める（**図-6**）。

これにより，処理データ単位を32ビットから16ビットまたは8ビットに削減することで，メモリ，バス，演算器をそれぞれ2倍，4倍利用することが可能になる。同時に，更に細かい単位での必要精度を把握することにより，チップ間通信での時間圧縮を可能にしている。これらにより，Deep Learningにおいて学習処理の電力性能を2 ～ 4倍に向上できる。また，これらの機能をライブラリ化することにより，各フレームワークからの利用を容易にしている。

む　す　び

本稿では，Deep Learning技術の概要と三つの課題，そしてそれらの解決手法として分散並列化技術とメモリ使用の効率化技術，およびデータサイズ削減のための専用ハードウェアのアーキテクチャーについて述べた。

富士通研究所では，今後発展形として両者の組み合わせや，Caffe以外のフレームワークへの展開，そして富士通が開発中のDLU（Deep Learning Unit）への適用など，Deep Learning分野の技術発展に今後も寄与すべく研究開発を進めていく。

参考文献

(1) G. E. Hinton et al. : Improving neural networks by preventing co-adaptation of feature detectors. Neural and Evolutionary Computing 2012, Vol.1207.0580, p.1-18, 2012. https://arxiv.org/abs/1207.0580

(2) K. He et al. : Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016, p.770-778. https://arxiv.org/abs/1512.03385

(3) 山崎雅文ほか：MPIを用いたDeep Learning処理高速化の提案．IPSJ SIG Technical Report, Vol.2016-HPC-155, No.6 (August 2016).

(4) K. Shirahata et al. : Memory reduction method for deep neural network training. IEEE MLSP 2016.

（5） K. Shirahata et al.：Memory Reduction Method for Training Very Deep Neural Networks on a GPU. GPU Technology Conference（GTC）2017.
（6） 伴野 充ほか：DNN学習向けプロセッサの電力効率を向上する低精度演算技術の提案．xSIG2017.

著者紹介

池　敦（いけ　あつし）
コンピュータシステム研究所
次世代コンピュータシステムプロジェクト
システムアーキテクチャーの研究に従事。

石原輝雄（いしはら　てるお）
コンピュータシステム研究所
次世代コンピュータシステムプロジェクト
知能コンピューティングの研究に従事。

富田安基（とみた　やすもと）
コンピュータシステム研究所
次世代コンピュータシステムプロジェクト
知能コンピューティングアーキテクチャーの研究に従事。

田原司睦（たばる　つぐちか）
コンピュータシステム研究所
次世代コンピュータシステムプロジェクト
高性能システムアーキテクチャーの研究に従事。