平成 27 年度 TSUBAME 産業利用トライアルユース 成果報告書

利用課題名 大規模画像データセットの機械学習のための分散コンピューティング
英文：Distributed Computing for Machine Learning on Large-Scale Image Dataset

利用課題責任者 佐藤 育郎
Ikuro Sato

所属 株式会社デンソーアイティーラボラトリ
Denso IT Laboratory, Inc.
https://www.d-itlab.co.jp/

大規模な画像データセットの深層学習の処理速度の大幅な向上を目的に、ノード分散型の学習方法を開発した。データを多数の GPU で並列的に処理し、インタコネクト通信によりパラメタの更新量を足し合わせる方式を実装したところ、実質的な条件下での学習時のデータの処理速度を、単体の GPU との比較において、30 倍以上高速化することができた。一般物体識別のベンチマーク ILSVRC2012 では 13.89%のトップ 5 エラー率を得た。

We developed node-distributed machine learning method, aiming to accelerate training of massive image dataset for classification. By processing data in parallel on many GPUs and summing over updating quantities of the parameters through interconnect, we have gained more than 30x speedup with respect to single GPU computation. Our deep neural network models scored 13.89% top-5 error rate on ILSVRC2012 benchmark.

Keywords: Deep Learning, Convolutional Neural Network, Image Classification, Asynchronous Distributed Computing

背景と目的

画像認識は、自動車、警備、ロボット、携帯端末といった、多種多様な産業上の応用が期待されており、特に自動運転の実現には不可欠と見なされている技術である。画像認識アルゴリズムの内部で使用されるパラメタは、多くの場合、機械学習の枠組みで最適化される。好ましい認識率の獲得には、大量の実画像データを学習する必要があるが、データ量の増加に伴い、学習アルゴリズムの収束に要する計算時間もまた増加するのが一般的である。データセットの規模によっては、単体の計算機での学習時間が1か月以上にも及ぶことも稀ではなく、この長大な学習時間が産業応用上の課題となっている。

本利用は、大規模画像データセットの深層学習の処理速度の大幅な向上を目的に、処理の分散化を検討したものである。データを多数の GPU で並列的に処理し、インタコネクト通信によりパラメタの更新量を足し合わせる方式を実装したところ、実質的な条件化での学習時のデータの処理速度を、単体 GPU との比較において 30 倍以上高速化することが出来た。

概要

我々は、本トライアルユースを通して、Convolutional Neural Network (CNN)の分散計算アルゴリズムを構築し、これを実現するプログラムを独自開発した。CNN は、2 次元の畳み込み演算を複数層に渡って行うことがその特色であり、その過程において認識に有効な特徴が抽出されていく。特に畳み込み層を多層に渡って反復する Deep CNN がきわめて高い認識能力を持つことが近年の研究によって明らかとなっている。しかしながら多層に渡る畳み込み演算や、膨大な反復計算を必要とする勾配法の利用が、計算負荷を高くしており、さらにデータセットが大規模化することで、学習時間が長大になりすぎる深刻な問題をはらんでいる。

我々はこの問題を解決するために、高速に収束可能な CNN のノード分散型のアルゴリズムを開発した。開発の概要について以下に述べる。

学習アルゴリズムの分散には、一般にデータ並列とモデル並列の二種類の方式があり、我々はデータ並列のみを採用した。データ並列とは、一度の損失関数の

微分計算に使用する複数のデータを、複数の GPU に分散させ処理するものである（ここでは GPU の使用を前提とする）[2,3]。なお、モデル並列は1つのモデルを複数の GPU に分散させ必要に応じて変数を相互に通信し同期をとって処理することで、微分計算を行うものであり、微分計算に必要とされる全変数の容量が GPU メモリを超える場合に必要となる方式である[1,2,3]。

データ並列には、さらに同期型のもの[4]と非同期型のもの[2]があり得る。前者の場合、全スレッドが，それぞれ画像を読み込み，その画像に基づいた微分計算を終えたのち，通信によって微分を加算し，重みパラメタを更新し，最新の重みを全スレッドに配布するサイクルを反復することとなる．利点は，単体の GPU で行う学習に比べて単位時間あたりに処理できる画像枚数が大きく出来る点と，収束の保証がある点にある．一方，非同期型の場合，計算機のリソースを間断なく利用するため，同期型に比べてさらに反復計算の速度を向上させることが可能である利点がある一方，「古い」重みパラメタを使った微分を使うことから理論的に収束の保証がないといった欠点が挙げられる。我々は、重み更新の周期の短縮が高速化にとって最も重要であるという仮説を置き、非同期型の分散方法を採用した。

結果および考察

学習の高速化の度合いは学習器の設計に依存するため、高速化度合いの一般的定量化は難しい。そこで、当社で検討した中で認識性能が良好となったモデルをひとつ選び、その高速化度合いを成果として報告する。比較対象は、提案するマルチノード型の分散学習法と、単一の GPU＆マルチコア CPU を使用した通常の学習法である。前提として、両者とも同じ型の CPU と同じ型の GPU を有するものとする。学習器は、畳み込み層を 11 層持つ CNN である。提案法を使った実験では、48GPU を使用して、この学習器を最適化した。データの処理速度（単位時間内に何画像処理したか）を評価した結果、提案法の方が 30 倍以上高速であった。

提案法の認識性能を、一般物体識別のベンチマーク ILSVRC2012 を使って評価した。これは 1000 クラスの物体を画像から識別する問題であり、データセットには約 128 万枚の訓練データサンプルが含まれる。提案法を使って学習したモデルのアンサンブル識別器のテストデータセットにおけるトップ 5 エラー率（上位 5 件の識別クラスに正解クラスが含まれない率）は 13.89%であった
（ http://image-net.org/challenges/LSVRC/2015/results ）。

まとめ、今後の課題

本トライアルユースでは、大規模データセットの機械学習の、分散計算を利用した高速化に取り組んだ。データを GPU で並列的に処理し、インタコネクト通信によりパラメタの更新量を足し合わせる方式を実装したところ、実質的な条件下でのデータの処理速度を、単体 GPU との比較において 30 倍以上高速化することが出来た。一般物体識別のベンチマーク ILSVRC2012 では 13.89%のトップ 5 エラー率を得た。

認識性能を損なわずに、ボトルネックとなっているインタコネクト通信の負荷を削減することを今後の検討課題としたい。本検討では分散処理アルゴリズムの構築に専念したが、実装面での最適化は充分に検討できていない。実装の工夫による高速化を今後の開発課題としたい。

[1] A. Krizhevsky, et al., “ImageNet Classification with Deep Convolutional Neural Networks”, NIPS 2012.
[2] J. Dean, et al., “Large Scale Distributed Deep Networks”, NIPS 2012.
[3] R. Wu, et al., “Deep Image: Scaling up Image Recognition”, arxiv:1501.02876, 2015.
[4] F. N. Iandola, et al., “FireCaffe: Near-Linear Acceleration of Deep Neural Network Training on Compute Clusters”, arxiv:1511.00175, 2015.