TSUBAME 共同利用　平成 30 年度　学術利用　成果報告書

利用課題名　HPC を利用した自然言語処理技術の研究
英文：High Performance Computing for Natural Language Processing Technology Research

利用課題責任者
鳥澤 健太郎

所属
国立研究開発法人情報通信研究機構 ユニバーサルコミュニケーション研究所
データ駆動知能システム研究センター
http://www2.nict.go.jp/direct/

邦文抄録（300 字程度）
情報通信研究機構では、対話システム WEKDA を初めとする複数の自然言語処理アプリケーションで、深層学習による様々な分析を行っている。深層学習の高速化には GPGPU を用いることが一般的であるが、近年発表されている大規模なニューラルネットワークは、一枚の GPGPU に学習パラメータを収容することが困難である。そのため本研究では、ニューラルネットワークを分割し複数の GPGPU に格納して分散計算を行う、モデルパラレルと呼ばれる手法によって大規模ニューラルネットワークの学習を試みる。PyTorch など既存の深層学習フレームワークが出力する、ニューラルネットワークの中間表現に基づいてグラフ分割を行い、サブグラフを複数の計算機に転送して学習を行うことに成功した。

英文抄録（100 words 程度）
NICT has been developing language processing applications including WEKDA, where deep learning is intensively used to analyze a large-scale Web archive. Although it is common to use GPGPUs to speed up deep learning, extremely huge modern neural networks contain billions of parameters to learn and single GPGPU cannot store them in its memory. Our approach to learn such huge neural networks is model parallelism, which divides a neural network into smaller parts and distributes them onto multiple compute nodes. In this work, we analyzed an intermediate representation of a neural network that an existing framework outputs and successfully learned the network in a distributed manner.

Keywords: 自然言語処理, 大規模情報分析, テキスト分析, モデルパラレル

背景と目的

深層学習において扱われるニューラルネットワークは、年々大規模化が進んでいる。例えば、昨年 Google から発表され、言語処理分野において注目を集めるネットワーク BERT（参考文献[1]）では、学習におけるミニバッチのサイズを可能な限り小さくしても、一般に入手しやすい GPGPU の 16GB 程度のメモリには学習パラメータを格納できない。巨大なニューラルネットワークを高速に学習するため、これまでに、既存の多くの深層学習フレームワークがデータパラレルと呼ばれる分散処理方式をサポートしてきた。データパラレルでは、入力のミニバッチを分割し、複数の GPGPU で分散して学習を行うことで、高速化と共に、各 GPGPU で計算結果の保持に必要なメモリ量を減少させることができる。しかしこの方式は、巨大な学習データを並列に処理するのに適する一方で、ニューラルネットワークのパラメータが各 GPGPU に複製されるため、超巨大なニューラルネットワークでは GPGPU メモリが不足したり、複数 GPGPU 間でのパラメータの同期のコストが大きくなるという問題がある。

そこで利用者らは、モデルパラレルと呼ばれる並列処理方式によって、超巨大ネットワークを用いた深層学習を行うためのフレームワークを開発してきた。モデルパラレルは、ニューラルネットワークを分割し、複数 GPGPU に配置するため、巨大ニューラルネットワークを処理するのに適している。本研究は、開発したモデルパラレル深層学習フレームワークを、実際に GPGPU を多数搭載した計算機環境で使用し、その機能や速度を改善することを目的とするものである。

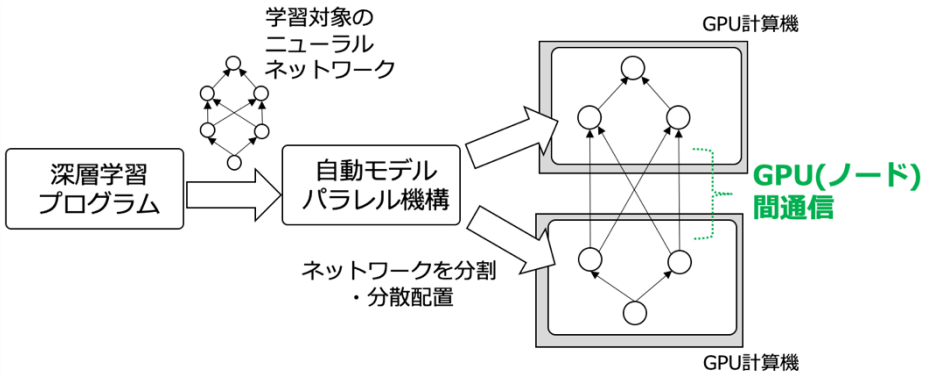

図 1 モデルパラレル用深層学習フレームワークの構成

概要

利用者らは従来からモデルパラレルのための深層学習フレームワークの研究を進めてきたが、利用できる GPU が同一サーバ上のものに限られるなどの制限があった。本研究では、スケーラビリティを向上させるため、複数の GPGPU サーバを利用可能とするよう拡張を行った。

利用者らが開発した、モデルパラレルのための深層学習フレームワークの概要を図 1 に示す。始めに、既存の深層学習フレームワークを用いて実装されたプログラムから、学習対象となるニューラルネットワークの中間表現（IR）を出力させる。ニューラルネットワークの IR として、ONNX[1]がよく知られており、多くのフレームワークから出力可能である。例として、2 つのテンソルを引数に取り、それらを加算した結果を返す単純な関数について、PyTorch[2]が生成する IR を図 2 に示す。こうした IR を出力する機能を備えたフレームワークでは、深層学習プログラムのソースコードの解析や、実行のトレースによって、こうした IR を自動的に出力できる。

```
graph(%a : Tensor
      %b : Tensor) {
  %2 : int = prim::Constant[value=1]()
  %3 : Tensor = aten::add(%a, %b, %2)
  return (%3);
}
```

図2 中間表現（IR）の例

利用者らのフレームワークでは、この IR で記述されたニューラルネットワークのグラフ構造を分割し、複数の GPGPU 計算機に転送する。各 GPGPU 計算機では、使用した IR を受理可能な深層学習フレームワークを用いて学習を行う。順伝播・逆伝播の計算時には、グラフの cut となった部分について MPI でのデータ転送を行う。

結果および考察

PyTorch を用いた深層学習プログラムを対象に、開発したフレームワークを用いてモデルパラレルでの学習を行った。PyTorch が出力した IR を解析し、およそ GPGPU の消費メモリが均等になるようにニューラルネットワークの分割を行い、複数の GPGPU サーバで分散して学習することに成功した。

まとめ、今後の課題

本研究で試作したモデルパラレルのためのニューラルネットワーク分割は、ごく試験的なものに止まる。今後、BERT 等の巨大ニューラルネットワークに対して適用すると共に、ニューラルネットワークの分割アルゴリズムを改善し、大規模化と共に速度向上を図る。

参考文献

[1] Devlin, Jacob and Chang, Ming-Wei and Lee, Kenton and Toutanova, Kristina. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805, 2018.

[1] https://onnx.ai/

[2] https://pytorch.org/