経済産業大臣賞

大規模深層学習のための
自動並列処理ソフトウェア RaNNC

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

主任研究員　田仲 正弘

東京大学大学院　情報理工学研究科

教授　田浦 健次朗

東京大学　情報基盤センター

教授　塙 敏博

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

副研究所長　鳥澤 健太郎

1．緒　言

本稿では、応募者らが開発し、昨年よりGitHub（https://github.com/nict-wisdom/rannc）にてフリーソフトとして公開している[1] RaNNC（Rapid Neural Network Connector）について述べる。RaNNCは、大量の計算機と高レベルなスキルが必要とされる大規模な深層学習を大幅に容易にし、より多くの人々に大規模深層学習によるAIを活用してもらうことを目的に開発したものである。

近年、深層学習は、ニューラルネットの大規模化により顕著な発展を遂げ、GPT-3［Brown 2020］等の巨大ニューラルネットは、もはや人間レベルの高品質なテキストを自動で書き下せる他、画像認識、対話システム等、多様な技術で実活用され始めている。一方、我が国の関連技術は、米国等の後塵を拝しており、例えば、日本発の深層学習ソフトウエアとして有名なChainer［Tokui 2019］は、PyTorch［Paszke 2019］等の米国の類似ソフトウェアがデファクトスタンダードとなったため、開発を中止している。深層学習の急速な普及からすれば、こうした状況は将来、日本全体に悪影響を及ぼしかねない。こうした状況の打開を目指した、日本における深層学習研究とビジネスの促進は、RaNNCを開発したもう一つの目的である。

ニューラルネットは、各種の数値計算と大量の学習パラメータ（数値データ）からなるが、その学習には膨大な計算が必要であり、前述のPyTorch等、深層学習専用ソフトウェアによってニューラルネットを定義し、深層学習に特化した計算機であるGPUを用いて学習が行われる。ニューラルネットの性能は、学習パラメータの数を増加させ、巨大化させることにより、劇的に向上するとされるが［Kaplan 2020］、そうした巨大化は必要な計算の量やメモリ量の増加に直結する。このため、巨大ニューラルネットの学習を現実的な時間で完了するには、計算や学習パラメータの記憶を多数のGPUに分担させ、並列に計算を実行して、学習を高速化することが必須である。

応募者らが開発したRaNNCは、上に述べたような、多数のGPUに計算や学習パラメータの記憶を分担させ、並列で学習を行うプロセスをほぼ完全に自動化するものである（図1）。一方で、後述するように、既存のソフトウエアでは、利用者が、細かな計算や各々の学習パラメータの記憶を、複数あるGPUのどれで行うかといった分担を決定し、通常、1台のGPUを想定して記述されるニューラルネット定義のプログラムの中に、そうした分担の指定を書き込む必要がある。加えて、この分担の仕方によって学習時の計算速度は大きく変わるが、そもそも、どのように計算を複数のGPUに分担させると高速化できるかは、ハイレベルな専門家にも自明ではなく、最終的な分担の決定には、長期に渡る試行錯誤が必要となる。例えばGPT-3は、1,750億個もの学習パラメータを持つが、こうした規模のニューラルネットの学習には、おそらく、多数の並列処理の技術者による、数ヶ月単位のチューニングを要したと推測される。

一方RaNNCは、ニューラルネット内の処理やGPUの計算速度、搭載メモリ量などを自動的に分析、勘案し、また、GPUへの分担決定等の各種の試行錯誤を自動化するため、開発者がニューラルネット定義のプログラムを一切書き換えることなく、高速な学習を実行できる。つまり、専門家のチューニング作業が一切不要になり、大規模な深層学習を劇的に容易化、ローコスト化できる。実際に我々は、RaNNCを用いて、一台のGPUを想定して記述されたニューラルネット定義を一切修正せずに、GPT-3を超える2,000億超の学習パラメー

[1] ライセンスはMITライセンスとしており、商用目的を含め、無償で利用できる。

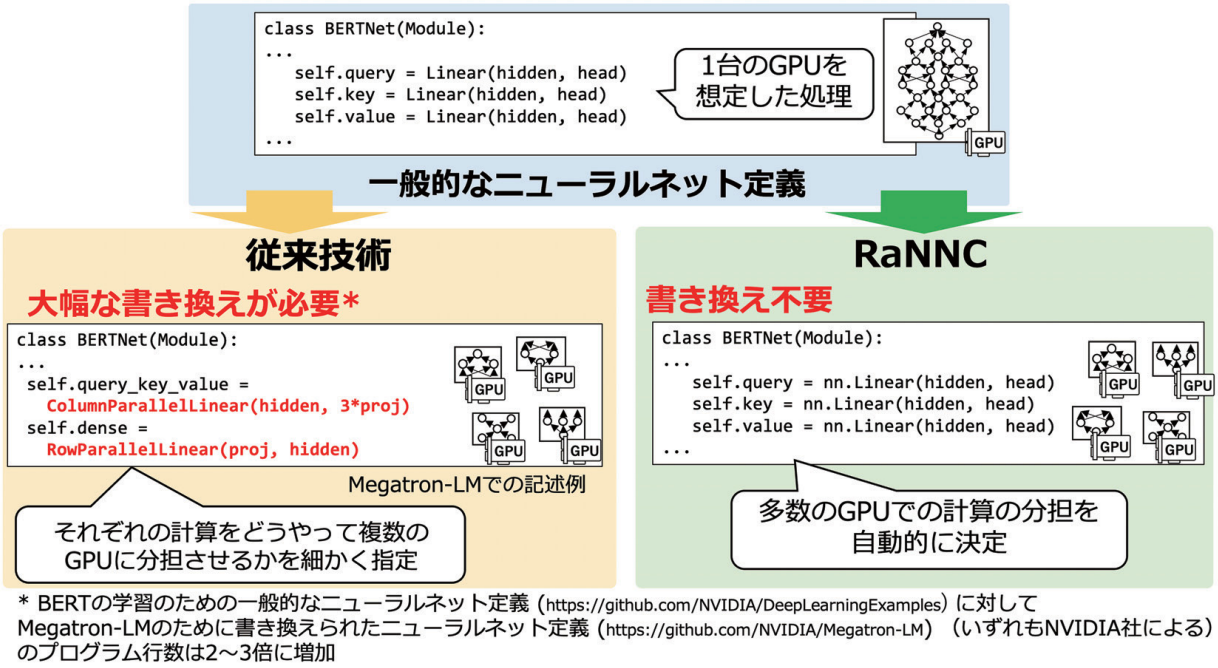


図1：従来技術との比較

タを持つニューラルネットが、192台の GPU で学習可能なことを確認している。

また、広く使用されている既存ソフトウェアである Megatron-LM[Shoeybi 2019]や Mesh-TensorFlow [Shazeer 2018] は、GPT-3 や BERT [Devlin 2019] など、Transformer [Vaswani 2017] と呼ばれる特定のニューラルネットの種類にしか適用できないのに対し、RaNNC は基本的に適用できるニューラルネットの種類に制約がない点でも優れている。例えば、応募者らは畳み込みニューラルネットと呼ばれるニューラルネットと BERT を組み合わせた BERTAC[Oh 2021] を提案しているが、そのようなニューラルネットのパラメータ数を増加させて学習する場合には、Megatron-LM 等は利用できず、RaNNC を用いる必要がある。

RaNNC は、PyTorch の開発を主導する Facebook が主催する PyTorch Annual Hackathon 2021（110ヶ国から1,947人が参加、応募65件）で、First Place（第一位、PyTorch Developer Tools & Libraries 部門）を受賞しており、並列処理分野のトップレベルの国際会議 IPDPS 2021（IEEE International Parallel and Distributed Processing Symposium）でも発表されている [Tanaka 2021] ほか、日本語での講演ながら、NVIDIA が主催する GTC（GPU Technology Conference）で招待講演を行うなど、対外的にも高く評価されている。また、スーパーコンピュータ富岳で RaNNC を稼動させる計画も進んでいる。RaNNC は2021年3月に公開されたが、海外からもフィードバックが寄せられており、大量の GPU を保有する利用者でなければ使用できないにも関わらず、ダウンロード数の測定を始めた2021年10月下旬から本稿執筆時点の2022年3月までで約600件ダウンロードされるなど、今後広く普及が進んでいくと期待される。

また、応募代表者らの所属する NICT では、これまで BERT 等の深層学習技術を用いて、多様な質問に膨大な Web 情報を用いて回答する大規模 Web 情報分析システム WISDOM X（https://www.wisdom-nict.jp/ にて現在一般公開中）や、高齢者の健康状態を、雑談を交えつつ対話を通じてチェックし、高齢者介護を支援するためのマルチモーダル音声対話システ

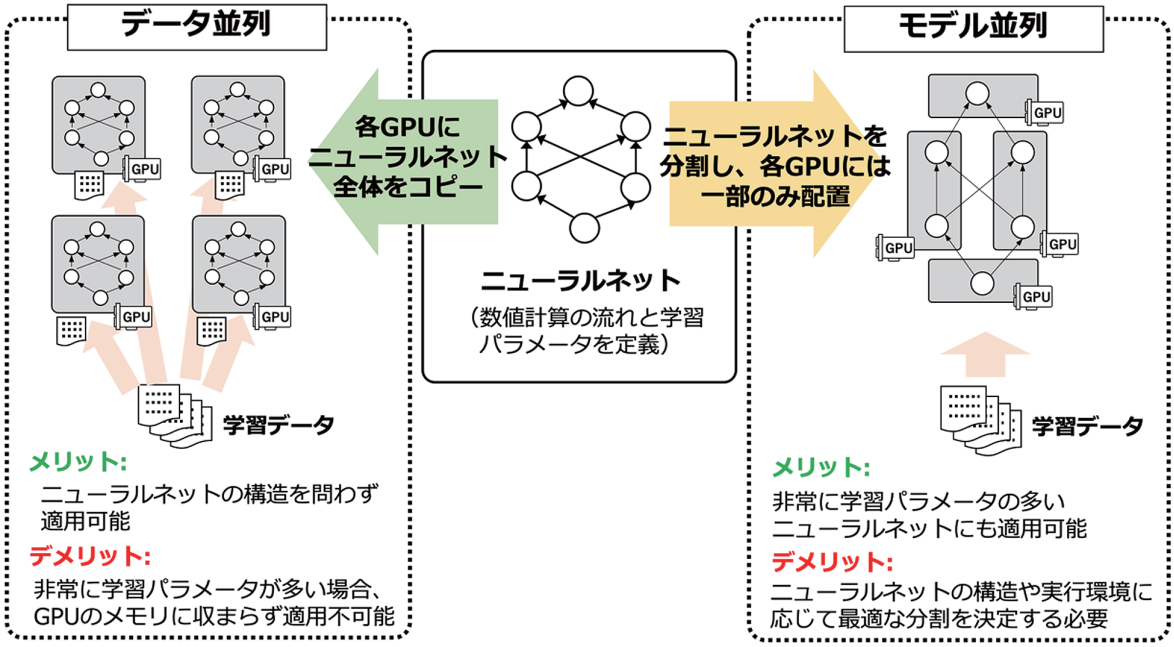


図2：深層学習の並列化方式

ム MICSUS(“MICSUS”と検索すると、YouTube でデモ動画等を閲覧できる)等を開発してきた。現在、これらのシステムの開発経験に基づき、RaNNC を用いて、200億パラメータの日本語対応の巨大ニューラルネットを学習している(商用化時の運用コスト低減に配慮しつつ、最新の研究成果に基づき、単純にパラメータを増大させるよりも、高品質な学習データを用い、より長期にわたる学習を行う方針を採っている)。こうした巨大ニューラルネットは、将来、MICSUS 等と併せ、幅広くライセンス等を行っていく。

2. 背景：巨大ニューラルネットワークの学習

近年の巨大ニューラルネットの隆盛のきっかけは、2018年における、3.4億個の学習パラメータを持つニューラルネット BERT の発表である。BERT は、アプリケーションごとに特化された異なる構造のニューラルネットを用いるという当時の常識を打ち破り、テキストの分類や質問応答等、多くのアプリケーションで世界最高性能を叩き出した[2]。また前述の GPT-3は、1,750億個ものパラメータを持ち、ブログや対話等の様々な分野で高品質なテキストを生成するなどして、世界を驚かせた[3]。

こうした巨大ニューラルネットの学習は、前述したように、多数の GPU を使った並列処理によって高速化する必要があるが、こうした並列化にはデータ並列とモデル並列の2種類がある(図2)。PyTorch を含む多くの深層学習ソフトウェアは、学習データを分割して、複数の GPU に割り当てるデータ並列のみをサポートしているが、各々の GPU のメモリにニューラルネットの学習パラメータ全体をコピーする必要があることから、全学習パラメータが1台の GPU のメモリに格納できない GPT3のような巨大ニューラルネットの学習を、データ並列だけで行うことは不可能である。

---

[2] https://ainow.ai/2019/05/08/166723/ 等、多くの解説記事が公開されている。
[3] https://www.intellilink.co.jp/column/ai/2021/031700.aspx 等で、多くの事例が挙げられている。

一方、モデル並列では、ニューラルネットを分割し、分割で得られたより小さなニューラルネット（部分ニューラルネットと呼ぶ）を各々のGPUに割り当て、それらのGPUが互いに通信し、連携しつつ学習を行う。各GPUは、分割された部分ニューラルネットに関係する学習パラメータのみをメモリに記憶すれば良いため、学習パラメータの多い巨大ニューラルネットの学習も可能になる。一方で、分割の仕方によっては、GPU同士のデータ通信に要する時間ばかりが長くなり、多数のGPUを使っても全体の計算速度が低下する。モデル並列では通常、そうした速度低下を避けるため、通信の頻度や量、各々のGPUで行う計算の量の間でバランスを取る必要があるが、RaNNC以前の既存ソフトウェアでは、そうしたバランスを取る作業は人間の開発者が自ら行い、その設定をPyTorch等のために書かれたニューラルネットの定義に事細かに記載する必要があった。また、適切なバランスを見いだすには、GPUやそれを繋ぐネットワークの性能に加えて、ニューラルネットの特性をまず踏まえる必要があるが、それだけでは処理時間の予測は難しく、多数の実験、試行錯誤が必要になる。要するに、特に日本においては希少な人材であるところの、ニューラルネットとGPU、並列計算のいずれに関しても深い知識を持った多数の技術者が、多くの試行錯誤を通じて分割を決定する必要があり、これが、巨大ニューラルネットが一部の有力海外企業等でしか開発されていない理由の一つである。RaNNCはこうしたニューラルネットの分割等の一連の作業を自動化し、より多数の人、組織が巨大ニューラルネットを開発することを可能とする。

3．RaNNCによる自動並列化

表1に、モデル並列による巨大ニューラルネット学習を行う著名な既存ソフトウェアを挙げる。これらを使った学習を行うには、前述の通り、様々な決定を人手で行う必要があるが、決定すべき要素をより詳細に見ていくと以下のようになる。(1) ニューラルネットをどのよ

表1：既存ソフトウェアとの比較

	自動化			適用できるニューラルネット種類
	分割方法決定	コピー数決定	GPU配置決定	
Megatron-LM（NVIDIA）	×	×	×	Transformerに限定
Mesh-TensorFlow（Google）	×	×	×	Transformerに限定
GPipe（Google）	×	×	×	制限無し
PipeDream-2BW（スタンフォード大、Microsoft）	×	○	×	制限無し
RaNNC（応募者らが開発）	○	○	○	制限無し

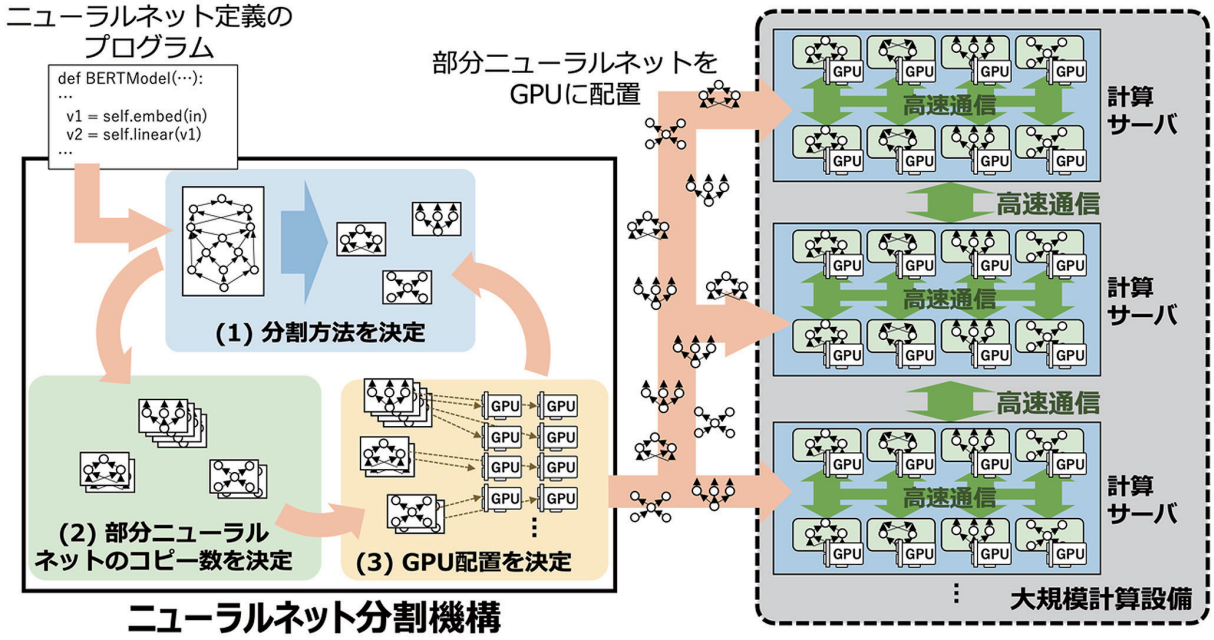


図3：RaNNC による自動並列化

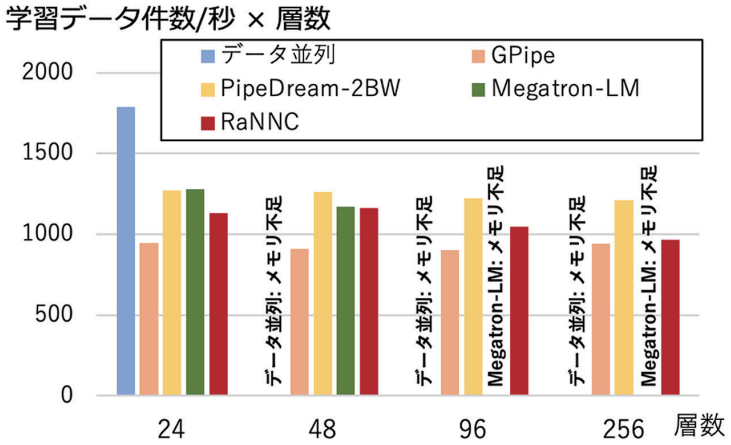
うに複数の部分ニューラルネットに分割するかを決定する。(2)得られた部分ニューラルネットを、何台の GPU にコピーするかを決定する。多くの場合、部分ニューラルネットごとにコピーを複数作成し、データ並列のように、それぞれ異なる学習データに関する計算を分担させることで高速化できる。(3) 各部分ニューラルネット (コピーを含む) をどの GPU に配置するかを決定する。通常、1台の計算サーバは複数の GPU を備えており、頻繁に通信する部分ニューラルネット同士を、高速に相互通信可能な同一の計算サーバ内の GPU に配置することで、全体の処理を高速化できる。

なお、これら3種の決定は相互に依存しており、一見すると各 GPU の分担が不均等で、低い処理速度しか得られなさそうな分割方法が、コピー数や GPU 配置によっては、実際には極めて高速な学習を実現することもある。このため、高い処理速度を得るためには、試行錯誤を通じて良い組み合わせを探しながらチューニングするほかない。なお、PipeDream-2BW は、分割可能な箇所を人間が指定すると、限られた組み合わせ (せいぜい数十パターン程度) から、分割数やコピー数を自動で決定する機能を持つが、可能な分割数等は予め人間が指定しておく必要がある。

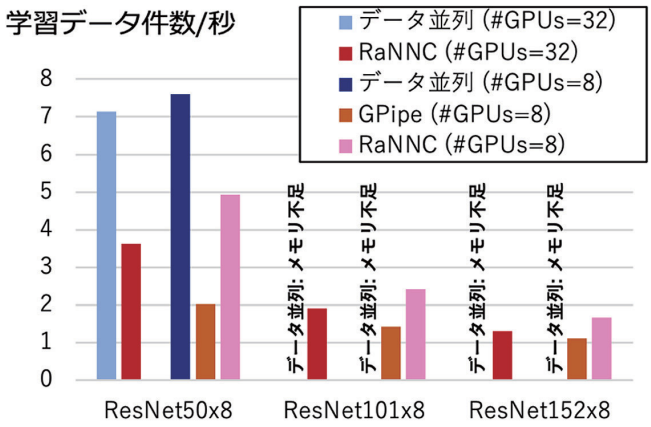
一方 RaNNC は、様々な分割方法・コピー数・GPU 配置の組み合わせについて、可能な限り全体として高速な学習が行えるような決定を行う。これらの組み合わせは膨大な数に上るため、動的計画法[4]を用いたアルゴリズムによりまず有望な組み合わせ (数千～数万パターン程度) に絞り込んだ上で、それぞれの組み合わせについて、学習時の計算を部分的に実行しながら自動的に試行錯誤し、高速な学習速度を達成できるものを選択する (図3)[5]。

[4] 組み合わせによって生まれる選択肢が膨大にあり、単純に一つ一つ調べて最適なものを探すことが不可能な場合にしばしば使用される計算手法の一つ。与えられた問題を、より単純な問題の集まりに分解し、単純な問題を順番に解きながら、その結果を効率よく記録することで、チェックするべき選択肢を大幅に絞ることができる。

[5] RaNNC による分割の決定には、200億パラメータのニューラルネットで4時間程度、2000億パラメータのニューラルネットでおよそ1日を要した。



(a) 大規模BERT



(b) 大規模ResNet

図4：既存ソフトウェアとの速度比較

以下では、RaNNC の有効性を評価するため、RaNNC 及び Megatron-LM、GPipe [Huang 2018]、PipeDream-2BW [Narayanan 2020] を用いて巨大ニューラルネットを学習し、その学習速度を比較した結果を報告する[6]。既存ソフトウェアはいずれも、ニューラルネット定義を書き換える必要があるが、実験では、各ソフトウェアの開発者自身が作成し、かつ高速化のためにチューニングしたニューラルネット定義を使用した。一方、RaNNC を用いる場合、1台の GPU を想定した一般的なニューラルネット定義（応募者以外が開発し、オープンソースで公開したもの）を使用している。

学習パラメータを最大129億まで増加させた BERT[7] を、32台の GPU で学習させた実験では、Megatron-LM と比較して、約5倍の学習パラメータを持つニューラルネットが学習できることが確認でき（図4（a））、また、Megatron-LM で学習可能な規模のニューラルネット

[6] 詳細な実験設定については、応募者らの論文 [Tanaka 2021] に記載している。なお、表1に示した Mesh-TensorFlow は、ベースとなる計算エンジンに Google が開発した TensorFlow [Abadi 2016] を使用しており、PyTorch を用いている RaNNC 及びその他のソフトウェアとは、同一のニューラルネットでも、計算速度やニューラルネット定義が大きく異なり、比較が困難なため、実験には用いていない。

[7] BERT のパラメータ数は、主に隠れ層と呼ばれる部分のサイズと数によって決まる。本実験では、隠れ層サイズを2048（原論文の2倍）とし、層数を変化させながら実行速度を調査した。

では、ほぼ同等の速度となった[8]。GPipe との比較では、いずれの設定でも RaNNC の方が高速であった。PipeDream-2BW は、RaNNC よりも幾分高速であったが、複数の GPU での計算結果の収集を待たずに、次の計算ステップに進むことで高速化しているため、計算の精度が低くなり、学習性能が低下する傾向があることが知られている。

また、画像分類における代表的なニューラルネットである ResNet[He 2016] の学習パラメータを最大37億個まで増加させ[9]、GPipe との比較を行ったところ[10]、RaNNC の方が1.5～2.5倍程度高速であった(図4(b))。

こうした結果は、ほぼ完全自動でニューラルネットの分割、高速な並列計算を行えるという RaNNC の大きなコスト低減効果と合わせて、RaNNC の価値を示すものである。

4．結　言

応募者らが開発した RaNNC は、巨大ニューラルネットの学習を劇的に容易にするものである。今後、NICT の GPU やスーパーコンピュータ富岳を活用し、GPT-3に代表される Few-shot(Zero-shot) 学習を行うものを含め、巨大ニューラルネットの学習なども進めていく。

参考文献

[Brown 2020] Brown, T. B., Mann, B., Ryder, N., Subbiah, et al. (2020).“Language Models are Few-Shot Learners.”arXiv preprint, arXiv:2005.14165.

[Tokui 2019] Tokui, S, et al. (2019).“Chainer: A Deep Learning Framework for Accelerating the Research Cycle.”Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2002−2011.

[Paszke 2019] Paszke, A., Gross, S., Massa, F., et al. (2019).“PyTorch: An Imperative Style, High-Performance Deep Learning Library.”In Advances in Neural Information Processing Systems 32 (NIPS 2019), pp. 8024−8035.

[Kaplan 2020] Kaplan, J., McCandlish, S., et al. (2020).“Scaling Laws for Neural Language Models.”arXiv preprint, arXiv:2001.08361.

[Shoeybi 2019] Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., and Catanzaro, B. (2019).“Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism.”arXiv preprint, arXiv:1909.08053.

[Shazeer 2018] Shazeer, N., Cheng, Y., Parmar, N., et al. (2018).“Mesh-TensorFlow: Deep Learning for Supercomputers.”In Advances in Neural Information Processing Systems 31 (NIPS 2018), pp.

[8] Megatron-LM の最新バージョンには、処理速度が遅くなる代わりにメモリ使用量を削減する機能があり、より大きなネットワークを学習できる可能性があるが、本実験を実施した2020年8月時点ではそうした機能がなく、使用していない。

[9] 畳み込み層と呼ばれる箇所のパラメータを増加させた。

[10] BERT を用いた比較実験に用いた PipeDream-2BW や GPipe は、本来のアイデアは各種のニューラルネットに適用可能であるが、実際のプログラムが BERT に特化されており、ResNet で動作させることができなかった。そのためこの比較では、GPipe のアイデアを汎用的に実現した torchgpipe(https://github.com/kakaobrain/torchgpipe) を用いた。torchgpipe はモデル並列のみに対応し、データ並列と併用できず、また複数サーバの GPU を利用できない。そのため、サーバ1台(GPU8台) を用いたモデル並列で実行した。前述の通り、Megatron-LM は Transformer からなるニューラルネットにしか適用できないため、ResNet を用いた実験には使用していない。

10435−10444.
[Devlin 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” In Proceedings of the 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), pp. 4171−4186.
[Vaswani 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., et al. (2017). “Attention is All You Need.” In Advances in Neural Information Processing Systems 30 (NIPS 2017), pp. 5998−6008.
[Oh 2021] Oh, J.-H., Iida, R., Kloetzer, J., and Torisawa, K. (2021). “BERTAC: Enhancing Transformer- based Language Models with Adversarially Pretrained Convolutional Neural Networks.” In Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021), pp. 2103−2115.
[Tanaka 2021] Tanaka, M., Taura, K., Hanawa, T., and Torisawa, K. (2021). “Automatic Graph Partitioning for Very Large-scale Deep Learning.” In Proceedings of the 35th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2021), pp. 1004−1013.
[Huang 2018] Huang, Y., Cheng, Y., Bapnao, A., et al. (2018). “GPipe: Efficient Training of Giant Neural Networks using Pipeline Parallelism.” arXiv preprint, arXiv:1811.06965.
[Narayanan 2020] Narayanan, D., Phanishayee, A., Shi, K., Chen, X., and Zaharia, M. (2020). “Memory-Efficient Pipeline-Parallel DNN Training.” arXiv preprint, arXiv:2006.09503.
[Abadi 2016] Abadi, M. et al. (2016). “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.” arXiv preprint, arXiv: 1603.04467.
[He 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). “Deep Residual Learning for Image Recognition.” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), pp. 770−778.