5J-02

大規模言語モデルの分散並列学習

藤井 一喜 [1] 横田 理央 [2]
[1] 東京工業大学 情報理工学院　[2] 東京工業大学　学術情報国際センター

1 はじめに

近年、様々な研究機関や企業で大規模言語モデル (LLM) の開発が行われている。これらのモデルは、人間に近い言語理解能力と生成能力、さまざまな分野への適用可能性を示し、大きな注目を集めている。大規模言語モデルの効率的な学習には、分散並列学習が不可欠であり、学習効率に大きく影響する重要な要素である。本論文では、Llama 2 から継続事前学習を行った日本語モデル「Swallow [1)]」の開発に使用した分散並列学習手法について報告する。また学習に使用した実装を公開する。 [2)] 継続事前学習の有効性に関する議論は、別の論文を参照されたい [1]。

2 分散学習

大規模言語モデルを 1 つの GPU で学習することは、GPU メモリの制約と学習に要する時間の両方から困難である。GPU メモリの点では、最新の H100 80GB を利用しても、今回学習した 7B のモデルを学習することは困難である。また、仮にモデルパラメータと勾配、オプティマイザの状態 (Optimizer State) が 1 枚の GPU に収まったとしても、1 枚の GPU では学習を完了するために非現実的な時間を要することになる。そのため、今回の学習では、データ並列とモデル並列を併用した分散並列学習を採用した。

2.1 学習環境

学習には、産総研の AI 橋渡しクラウド (ABCI) を利用した。混合精度（bfloat16）を採用し NVIDIA A100 ノードを複数台使用し分散並列学習を行った。各ノードは NVIDIA A100 40GB GPU を 8 基を搭載し、ノード間は InfiniBand HDR にて接続されている。

2.2 学習ライブラリ

学習には Megatron-LM [3)] を利用した。3D 並列化 (3D Parallelism) が利用できる点、分散学習設定のセクションで触れる工夫が実装されている点、ライブラリとしての成熟度を加味し選定を行った。

2.3 分散学習設定

本研究では Llama 2 7B、13B、70B から継続事前学習を行った。効率的な学習を行うために、データ並列（Data Parallelism）、テンソル並列化（Tensor Parallelism）、パイプライン並列化（Pipeline Parallelism）を統合した 3D 並列化（3D Parallelism）を採用し、高い計算効率と効率的なメモリ利用を目指した。表 1 に各モデルサイズにおける分散学習設定を示す。他にも、以下に挙げる工夫を取り入れた。 [4)]。

表 1 モデルパラメータごとの分散学習設定: DP、TP、PP、SP はそれぞれデータ並列 (Data Parallelism)、テンソル並列 (Tensor Parallelism)、パイプライン並列 (Pipeline Parallelism)、シークエンス並列 (Sequence Parallelism) を表す。

Params	DP	TP	PP	SP	Distributed Optimizer
7B	16	2	2	✓	✓
13B	8	2	4	✓	✓
70B	4	8	8	✓	✓

効率的なメモリ消費　Megatron-LM の Distributed Optimizer を用いて、オプティマイザの状態（optimizer state）をデータ並列プロセス間に分散配置し、冗長性を排除することで、必要なメモリ使用量を削減した。Distributed Optimizer は Reduce Scatter と All Gather を利用し、効率的に通信を

1) https://tokyotech-llm.github.io/swallow-llama
2) https://github.com/rioyokotalab/Megatron-Llama2
3) https://github.com/NVIDIA/Megatron-LM
4) いくつかの実験では、実験途中で使用ノード数の変更が行われたため、データ並列数が 2 倍もしくは 1/2 倍されている期間が存在する。これは本論文で説明する実験以外にも同時に進めた実験があるためで、限られた計算資源・期間で有望そうな設定を優先してモデル構築を進めたからである。

行うため、通常のデータ並列と同じ通信コストにも関わらずメモリ削減が行える。

トポロジーを考慮した 3D マッピング 3D 並列化において、Transformer ブロックはパイプライン並列により複数の GPU に分散配置され、さらにテンソル並列により層内のパラメーターは分散配置される。この際、Megatron-LM[2] で提案されているように、通信を多く必要とする分散手法のワーカー（テンソル並列ワーカー）はノード内に配置した。これは、ノード内の通信は NVLink によりノード間通信よりも高速であるためである。また、データ並列の勾配平均化のための通信を考慮して、データ並列ワーカーも可能な限りノード内に配置した。パイプライン並列は他の並列化手法と比較して通信量が少ない P2P(Point-to-Point) 通信であるため、パイプラインステージはノード間で配置した。

メモリ効率化のための 1F1B の採用 1F1B(one forward pass followed by one backward pass) のパイプライン並列である PipeDream-Flush[3] を利用することで、パイプラインステージ数以下のマイクロバッチ数しか活性 (activation) を必要としないようにし、GPipe[4] よりもメモリ効率を上昇させた。

Sequence Parallelism による並列化 テンソル並列は Self-Attention や MLP ブロックを並列化するが、レイヤーノーム (Layer-Norms) とドロップアウト (Dropouts) は並列化しないため、テンソル並列プロセス間でこれらが冗長にメモリ上に存在する。これを効率化するために、シークエンス並列 (Sequence Parallelism) [5] を利用した。シークエンス並列は、テンソル並列と同時に用いることで、通信コストのオーバーヘッドなしにメモリの効率化を行える。

2.4 計算効率

実際に学習を行った際のモデルパラメータごとの TFLOPS を表 2 に示す。実行効率として、70B のモデルにおいて 50%以上を記録しており、効率的に学習を行えたと考える。

表 2 モデルパラメータごとの TFLOPS/GPU

Params	ノード数	TFLOPS/GPU	実行効率
7B	4	134	43.0 %
13B	8	143	45.8 %
70B	32	158	50.6%

3 おわりに

本研究では、最大で 70B のモデルの効率的な学習を行った。3D 並列化は大規模言語モデルの事前学習、継続事前学習を効率的に行うために重要な技術であり、本研究においてもその有効性が示された。しかしながら、3D 並列化は実装が難しく、あらゆる言語モデルの学習に向いているとは言えない。今後は、3D 並列化に加え FSDP(Fully Sharded Data Parallel) のようにデータ並列だけで 10B 前後のモデルを負担が少なく学習できるように、ライブラリ開発を進め、LLM の開発に貢献したいと考えている。

謝辞

LLM の継続事前学習の実験では、国立研究開発法人産業技術総合研究所が構築・運用する AI 橋渡しクラウド（ABCI: AI Bridging Cloud Infrastructure）の「大規模言語モデル構築支援プログラム」の支援を受けました。本研究は、JST、CREST、JPMJCR2112 の支援を受けたものである。

参考文献

[1] 藤井一喜, 中村泰士, Mengsay Loem, 服部翔, 飯田大貴, 水木栄, 平井翔太, 大井聖也, 横田理央, 岡崎直観. 継続事前学習による日本語に強い大規模言語モデルの構築. 言語処理学会第 30 回年次大会 (NLP2024), 2024.

[2] Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Anand Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, Amar Phanishayee, and Matei Zaharia. Efficient Large-Scale Language Model Training on GPU Clusters Using Megatron-LM. In **Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis**, pp. 1–15, 2021.

[3] Deepak Narayanan, Amar Phanishayee, Kaiyu Shi, Xie Chen, and Matei Zaharia. Memory-Efficient Pipeline-Parallel DNN Training. In **International Conference on Machine Learning**, pp. 7937–7947. PMLR, 2021.

[4] Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, et al. Gpipe: Efficient Training of Giant Neural Networks using Pipeline Parallelism. **Advances in neural information processing systems**, Vol. 32, , 2019.

[5] Vijay Anand Korthikanti, Jared Casper, Sangkug Lym, Lawrence McAfee, Michael Andersch, Mohammad Shoeybi, and Bryan Catanzaro. Reducing Activation Recomputation in Large Transformer Models. **Proceedings of Machine Learning and Systems**, Vol. 5, , 2023.