

DeepSpeed:

深層学習の訓練と推論を劇的に 高速化するフレームワーク

Microsoft DeepSpeed Team
2023 年 6 月 7 日



このスライドでは、我々が研究開発しているDeepSpeedというフレームワークについて、概要をご紹介します。

概要

What is DeepSpeed?

- 数年前から爆発的に広がり
- 話題のChatGPTなども深層学習の成果

- 大規模かつ高速な**深層学習**を容易に実現する様々な機能を持ったソフトウェア
- オープンソースソフトウェアとしてGitHubで公開中
 - [DeepSpeed](#) (メインのレポジトリ)
 - [DeepSpeedExamples](#) (使用例).
 - [Megatron-DeepSpeed](#) (NVIDIAのMegatron-LMと結合したもの).
 - [DeepSpeed-MII](#) (DeepSpeedの高速な推論を容易に利用するためのツール)



メインレポジトリのURL

DeepSpeedのプロジェクトは、Microsoftの[AI at Scale initiative](#)の一部で、次世代AIの機能の大規模な実現を進めています。詳細は[こちら](#)をご覧ください。

ご存じの通り、深層学習は数年前から爆発的に広く使われるようになり、最近話題のChatGPTなどは、その成果の一つです。

DeepSpeedは、その深層学習を、大規模かつ高速に、しかも容易に実行するための様々な機能を持ったソフトウェアです。

DeepSpeedはオープンソースソフトウェアとして、GitHubで公開されており、誰でもご利用いただけます。DeepSpeedを開発している我々のプロジェクトは、MicrosoftのAI at Scale initiativeの一部で、次世代AIの機能の大規模な実現を進めています。

概要

Who uses DeepSpeed? (利用例)

- Microsoft社内では、深層学習の効率化のため広く使用
 - [Turing-NLG: A 17-billion-parameter language model by Microsoft](#)
 - [New Z-code Mixture of Experts models improve quality, efficiency in Translator and Azure AI](#)
 - [Azure empowers easy-to-use, high-performance, and hyperscale model training using DeepSpeed](#)
 - [Video super resolution in Microsoft Edge](#)

In fact, **DeepSpeed** has become the defacto framework for distributed machine learning training
--- **Mark Eugene Russinovich (Azure CTO)** が何度も言及



[What runs ChatGPT? Inside Microsoft's AI supercomputer](#)

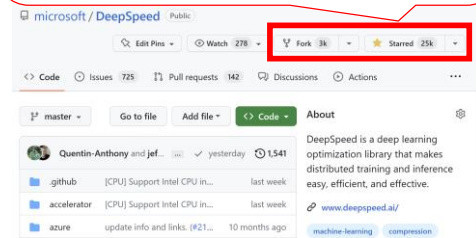
技術の詳細に入る前に、DeepSpeedの利用事例をご紹介します。
Microsoft社内では、深層学習の効率化のため広く使用されており、左にはプロジェクトの例をいくつか示しています。
右側の写真は、AzureのCTOがMicrosoftがどのようにAIのためのスーパーコンピュータを構築しているかについて話しているビデオです。
このビデオの中で、DeepSpeedは、深層学習の並列分散処理における事実上の標準のフレームワークとして、何度も言及されています。

概要

Who uses DeepSpeed? (利用例)

- **もちろん社外でも広く使われています！**
- AI分野のオープンソースソフトウェアとして、もっとも広く使用されているものの一つ
 - 多くの大規模モデルを訓練するために使用 [Megatron-Turing NLG \(530B\)](#), [Jurassic-1 \(178B\)](#), [BLOOM \(176B\)](#), [GLM \(130B\)](#), [YaLM \(100B\)](#).
 - [Hugging Face Transformers](#), [Hugging Face Accelerate](#), [PyTorch Lightning](#), [MosaicML Composer](#), [Determined AI](#) など、多くの著名なオープンソースの深層学習フレームワークのバックエンドとして利用

**25,000スター (GitHubの
Microsoftのオープンソースソフト
ウェアとして11位)
& 3,000フォーク**



通算530万超ダウンロード

How to use DeepSpeed? (使い方)

Webサイト (deepspeed.ai) でドキュメント・チュートリアルを提供

DeepSpeedはオープンソースのソフトウェアなので、もちろん社外でも非常に広く使われており、AI分野のオープンソースソフトウェアとして、もっとも広く使用されているものの一つと言ってもよいでしょう。

BLOOMなどよく知られたものを含む多くの代表的な大規模モデルは、DeepSpeedを使用して訓練されています。また、HuggingFaceをはじめとする多くの著名なオープンソースの深層学習フレームワークのバックエンドとして利用されています。

GitHubのスターは25,000を超えており、Microsoftのオープンソースソフトウェアの中では、11位です。これは深層学習に限らず、すべてのレポジトリを含めた順位であり、一位はVSCode、2位はPowerToys、3位はTypeScript、となっています。DeepSpeedのターゲットとなるAIや深層学習、しかもその高速化や大規模化を必要とする層を考えると、この分野においては非常に高い人気があると言えるでしょう。また、ダウンロード数は530万を超えています。

ドキュメントやチュートリアルは、Webサイトで公開されています。

概要

Who uses DeepSpeed? (利用例)

- 2023年4月にリリースしたDeepSpeed-Chatには極めて大きな反響
 - ChatGPTライクなモデルを学習するためのフレームワーク
- GIGAZINE, ITmediaなど著名なオンラインメディアに掲載

大規模言語モデル（ChatGPTなどに使用されるタイプの深層学習モデル; LLM）のリーダーボードにおいて
**オープンソース勢として
ダントツの存在感**

Leaderboard			
Rank	Model	Elo Rating	Description
1	gpt-4	1225	ChatGPT-4 by OpenAI
2	claude-v1	1205	Claude by Anthropic
3	claude-instant-v1	1153	Claude Instant by Anthropic
4	gpt-3.5-turbo	1143	ChatGPT-3.5 by OpenAI
5	vicuna-13b	1054	a chat assistant fine-tuned
6	vllm-1	1042	PaLM 2 for Chat (chat-bison@001) by Google
7	vicuna-7b	1007	a chat assistant fine-tuned
8	llm-1.7b	980	a dialogue model for academic research by BARR
9	mpt-7b-chat	952	a chatbot fine-tuned from MPT-7B by MosaicML
10	falcon-chat-13b	941	a chat assistant fine-tuned
11	llm-1.7b	937	a model fine-tuned for
12	llm-1.7b	929	an RNN with transform
13	mistral-7b-instruct-1.2b	921	an Open Assistant for
14	chatglm-6b	921	an open bilingual dialo
15	stablelm-tuned-alpha-7b	882	Stability AI language m
16	dolly-v2-13b	866	an instruction-tuned o
17	llm-1.7b	854	open and efficient foundation language models by Meta

[Chat with Open Large Language Models \(lmsys.org\)](#)

2023年4月に公開したDeepSpeed-Chatは、DeepSpeedのこれまでの成果の中でも、特に大きな反響があったものです。これはChatGPTライクなモデルを学習するためのフレームワークで、GIGAZINE, ITmediaなど著名なオンラインメディアに掲載されました。ChatGPT等の、対話を行うタイプのモデルを評価するリーダーボードをみると、トップ数件はGPT-4を含むクローズドなモデルが占めていますが、オープン化されたモデルについては、DeepSpeedの技術を使用して学習したものが大半を占めており、当該分野では圧倒的な存在感を示しています。

科学研究への応用例

- [OpenFold](#)

- [OpenFold library](#) は、DeepMind の AlphaFold 2を、PyTorchを用いて訓練可能のオープンソースフレームワーク
- [GPUメモリ使用量の削減](#)にDeepSpeedを使用

- [GenSLMs](#)

- SARS-CoV-2 (COVID-19)のゲノムの分析用の大規模言語モデル
- 受賞: [2022 ACM Gordon Bell Special Prize for High Performance Computing-Based COVID-19 Research](#) (大規模計算における権威ある賞)
- [GPUメモリ使用量の削減](#)にDeepSpeedを使用

深層学習の課題 – 訓練 (Training) –

深層学習では、大量の訓練データを与えて、モデルを訓練するが、計算量・メモリ量・処理データ量がとにかく多い

- 現実的な時間で訓練を終えるには、GPU等のアクセラレータ（高価）が必要
- モデル規模が大きくなる（=学習パラメータが多くなる）、または訓練データの量が増えると、必要な計算量・メモリ量も増え、多数のGPUを用いた並列処理が必要
→ 数千億パラメータ規模のモデルの訓練には、数千GPUを用いても2ヶ月 (e.g., [MT-NLG](#))



DeepSpeedは深層学習の訓練における様々な技術的課題を解決

課題	DeepSpeed の機能
モデルが大きすぎてGPUメモリに収まらない	ZeRO (Zero Redundancy Optimizer)
数千GPU規模までスケールさせたい	3D parallelism
高い計算効率のモデルを使用したい	DeepSpeed-MoE
GPU間の通信が遅い	Communication Compression
大規模なデータが必要	DeepSpeed Data Efficiency
ChatGPTにも使用されるRLHF 訓練を効率的に実行	DeepSpeed-Chat

DeepSpeedの技術について述べる前に、深層学習の一般的な課題について、簡単にご説明しておきます。

深層学習では、大量の訓練データを与えて、モデルを訓練します。この訓練において、計算量・メモリ量・処理データ量がとにかく多い、という点が、主要な難しさになります。一般には、現実的な時間で訓練を終えるには、GPUなどの高価なアクセラレータが必要になります。また、深層学習のモデルは、その内部にある学習パラメータとよぶ数値データの量でその規模を図るのですが、モデルの規模が大きくなったり、訓練のデータが増えると、多数のGPUを用いた並列処理が必要になります。極端に大きいモデル、例えば数千億パラメータ規模のモデルの訓練には、数千GPUを用いても2ヶ月といった計算時間がかかることもあります。DeepSpeedは、こうした深層学習の計算効率等に関する様々な課題を解決するものです。

ここでは、代表的な課題と、対応するDeepSpeedの技術についてまとめています。それぞれの技術については、後で取り上げます。

深層学習の課題 - 推論 (Inference) -

- 訓練されたモデルを使用するフェーズを推論という
- 実サービスで、多数のユーザによって実行されるのは推論 (e.g., new Bing)
- 訓練より計算量は少ないものの、低レイテンシ（高速な応答）、低コスト化などの要件が重要



DeepSpeedは推論のための様々な機能も提供

課題	DeepSpeedの機能
高速・スケーラブルな推論	DeepSpeed Inference
簡単にデプロイしたい	DeepSpeed-MII
MoEモデルの推論	DeepSpeed-MoE
巨大モデルを高速に推論	DeepSpeed Compression

また、訓練された深層学習モデルを使用するフェーズを、推論といいます。実サービスで、多数のユーザによって実行されるのは推論です。Bing Chatに話しかけると、何かの文章で応答してくれますが、そこで動いているのは推論です。推論は、一般には訓練より計算量は少ないのですが、低レイテンシ、つまり高速な応答が求められますし、多くのユーザに対して提供することから、低コスト化も重要で、訓練とはまた違う技術的課題があります。DeepSpeedでは、この推論についても、様々な機能を提供しています。

概要

Why use DeepSpeed? (何に使える?)

- 数十億~1兆規模のパラメータを持つ超巨大な深層学習モデルの訓練と推論
- 高いスループットと数千GPU規模のスケーラビリティの訓練
- 限られたGPUリソース環境における訓練と推論
- 極めて低レイテンシかつ高スループットな推論
- 高度なモデル圧縮技術による低遅延な推論とモデルサイズ削減

When/Where to use DeepSpeed?

(どういうときに特に有効／よく使われてる?)

- 多様な深層学習モデルの訓練や推論を加速
- 近年もっとも広く使われているTransformerベースのモデルを高速化 (ChatGPTなどもTransformerベース)

DeepSpeedがどういうことに使えるか、大まかにまとめたのがこのリストです。非常に大きいモデルを訓練する、そのためにたくさんのGPUを使う、といった目的から、GPUリソースは限られている中で、できるだけ大きいモデルを計算したい、というもの、レイテンシを下げスループットを上げる、といった目的などに使えます。実際のよくある使われ方としては、すでに深層学習モデルを持っているユーザが、その訓練や推論を高速化したり、モデルを大きくしたり、というものかと思われます。特に、近年広く使われているTransformerベースのモデル、ChatGPTなどもその一種ですが、その高速化にも非常に効果を発揮します。

アウトライン

DeepSpeed の様々な機能を広く浅く紹介

- **システム関連**

- ZeRO series (ZeRO, ZeRO Offload, ZeRO Infinity)
- 3D parallelism
- DeepSpeed Inference
- DeepSpeed-MII

- **モデリング関連 / 横断的トピック**

- DeepSpeed-MoE
- Communication Compression
- DeepSpeed Data Efficiency
- DeepSpeed Compression

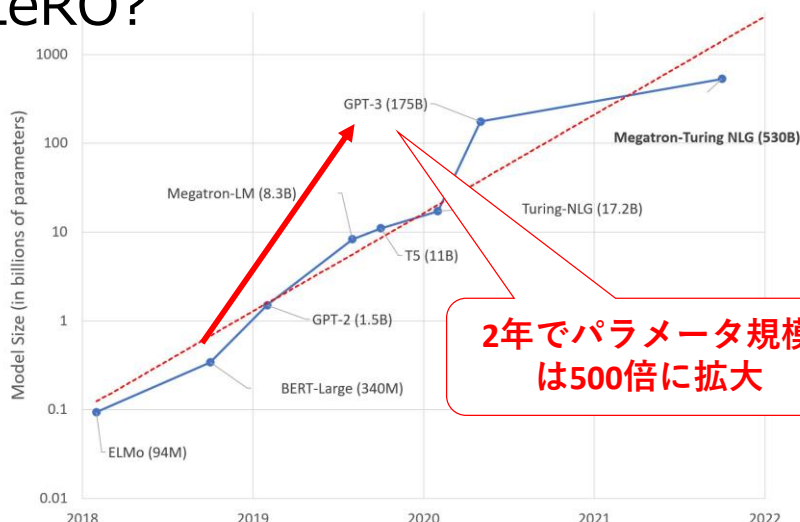
- **DeepSpeed-Chat: ChatGPTライクな大規模モデルを作るためのフレームワーク**

以降では、DeepSpeedの様々な機能について、広く浅く紹介します。

DeepSpeed ZeRO (Zero Redundancy Optimizer)

最初に、DeepSpeedの代表的な機能である、DeepSpeed ZeROについてご説明します。

Why ZeRO?

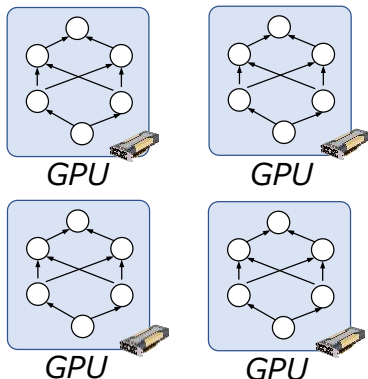


- モデルの規模は劇的に増大
- 高スループット・高スケーラビリティが重要

初めに、ZeROが必要となった背景についてご説明します。近年、深層学習のモデルの規模は、劇的に増大しています。2018年に、自然言語処理分野のブレイクスルーとなったBERTというモデルが発表されて、これは3.4億個の学習パラメータを持ち、当時は非常に大きいモデルとされていました。ところが2年後の2020年には、1750億ものパラメータを持つGPT-3が発表され、たった2年のうちに、モデルの規模は500倍に拡大したことになります。こうした巨大なモデルを訓練あるいは推論するために、効率の良い計算方法や、並列処理の際のスケーラビリティなどが重要な課題となります。

深層学習の並列化

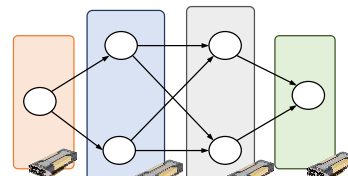
データ並列



- 高い効率を容易に実現
- モデル全体をGPUに複製 → **極めて大きなモデルは学習不可能**

モデル並列

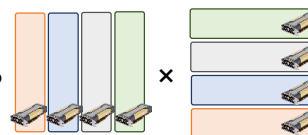
パイプライン並列



深層学習モデルをレイヤごとに分割し
パイプラインで計算

テンソル並列

内部の計算に使われる
巨大な行列を分割

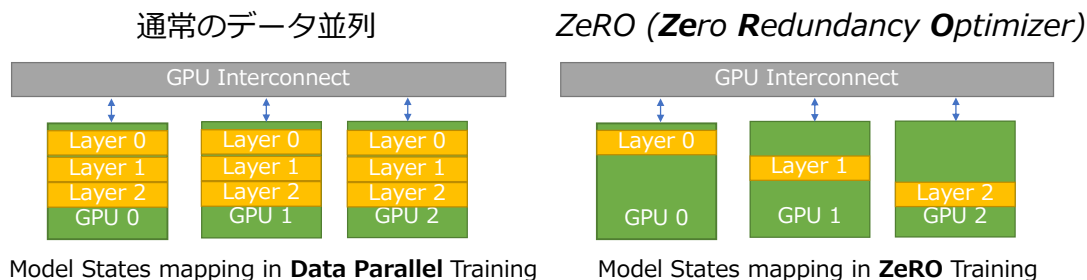


- 大規模なモデルもGPUメモリに格納可能
- モデルの書き換えが必要・高い効率の実現できるモデルに限られる

訓練のための計算を現実的に終えるためには、GPUを複数使った並列計算が重要になります。深層学習の並列化方式は主に二つあり、一つはデータ並列、もう一つはモデル並列です。データ並列は、高い効率を容易に実現できるメリットがありますが、モデル全体をそれぞれのGPUに複製して計算するので、GPT-3のように、非常に大きいモデルの訓練はできません。一方のモデル並列は、モデルを分割して複数のGPUに分けて配置して計算するもので、さらにパイプライン並列・テンソル並列といった方式に分けられます。大規模なモデルもGPUメモリに格納できるのがメリットですが、モデルを定義するプログラムの書き換えが必要だったり、基本的には通信オーバーヘッドの影響で計算の実行効率が下がってしまい、特定のモデルでしか高い実行効率が得られないという難しさがあります。

What is ZeRO?

- データ並列のメモリ利用を効率化
→ 巨大モデルを高い効率で学習、かつどのようなモデルアーキテクチャでも適用可能



- 通常データ並列では重複して持つ学習パラメータを、重複のないように各GPUに格納
- データが必要になった際に、GPU間の通信で必要な部分だけデータを収集し、使い終わったら捨てる

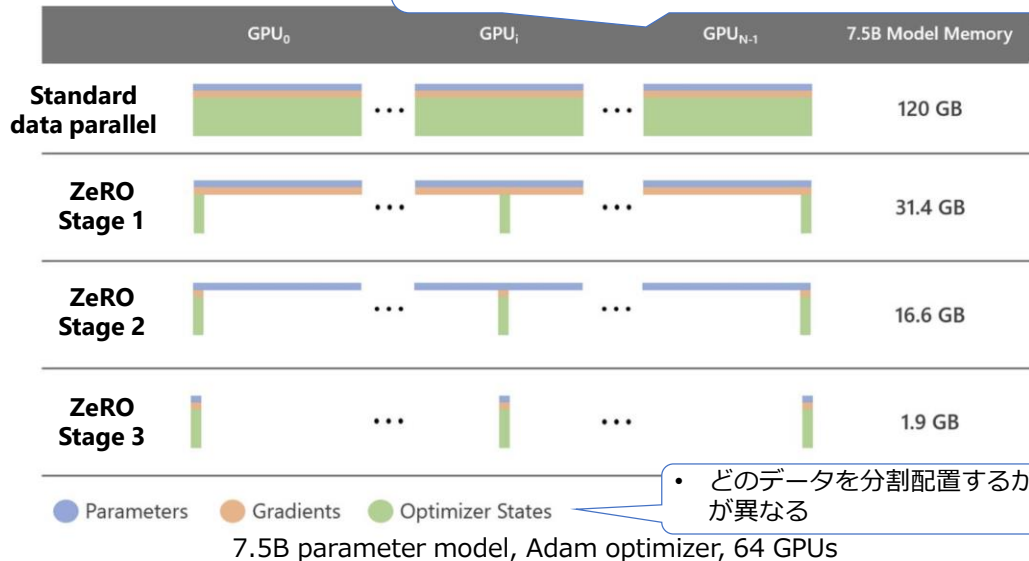
ZeROは、このうちデータ並列のメモリ利用を効率化するものです。高い効率で学習できるうえに、基本的にどのようなモデルにでも適用できる点がメリットです。左の図は通常データ並列で、すべてのGPUに、このレイヤのパラメータを0,1,2を複製して持っています。

右がZeROで、それぞれのGPUで、重複がないようにデータを格納しています。その動作を非常に単純化して説明すると、次のようになります。

Layer0を計算する際には、GPU0がそのデータを他の2つのデータに送信し、共有します。Layer0の計算が終わると、そのデータはGPUメモリ上から削除されます。そして次にLayer1の計算に進み、同様な処理が繰り返されます。これにより、ピークメモリを大幅に小さくできます。

What is ZeRO?

- パラメータを集めるのに通信オーバーヘッドが生じる
- 複数の動作モード (Stage) : 省メモリ効果と通信オーバーヘッドのトレードオフが異なる



多数のGPUに分割して配置されたモデルパラメータを、必要になるたびに通信で集めていると、基本的に通信オーバーヘッドが大きくなってしまいます。そこで、ZeROには、メモリ削減の効果と通信オーバーヘッドのトレードオフが異なる、3つの動作モード (Stage) があります。

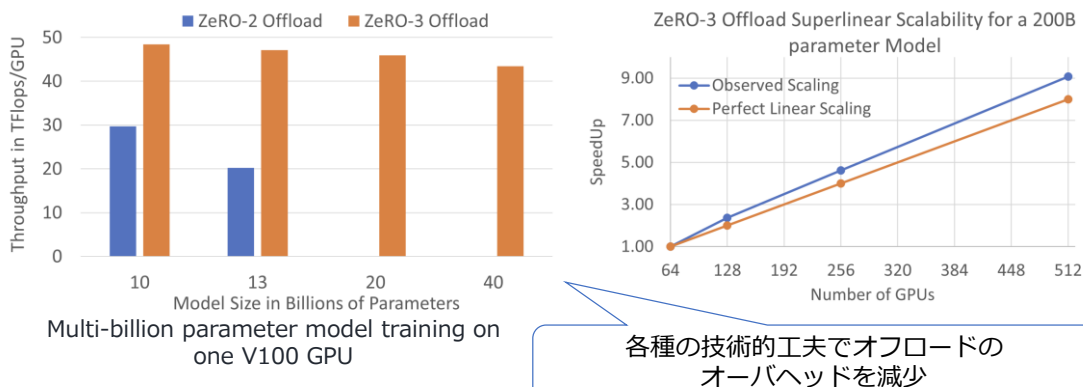
ここでは、75億パラメータを持つモデルを、64基のGPUで分散して学習する際の必要メモリを示しています。深層学習では、メモリ利用の支配的な要素として、パラメータ, Gradients, Optimizer stagesの3つがあります。標準的なデータ並列では、120GBのメモリが必要となります。Stage 1では、このうちOptimizer stagesにだけ、ZeROによるデータの分散配置を適用します。これにより、必要メモリは31.4GBになります。Stage 2では、さらにGradientsにも同じ技術を適用すると、必要メモリは16.6GBになります。Stage 3では、学習パラメータについても同じ技術を適用すると、必要メモリは劇的に減少し、1.9GBになります。Stage1, 2では、従来手法から通信オーバーヘッドをほぼ増やすことなく、必要メモリを大幅に削減できます。Stage3は、さらに多くのメモリを削減できますが、通信オーバーヘッドが大きくなります。使用するGPUのメモリや、訓練するモデルの規模に応じて、適切なStageを選ぶことになります。

	Max Parameter (in billions)	Max Parallelism	Compute Efficiency	Usability (Model Rewrite)
データ並列(DP)	Approx. 1.2	>1000	Very Good	Great
テンソル並列(TP)	Approx. 20	Approx. 16	Good	Needs Model Rewrite
TP + DP	Approx. 20	> 1000	Good	Needs Model Rewrite
パイプライン並列(PP)	Approx. 100	Approx. 128	Very Good	Needs Model Rewrite
PP + DP	Approx. 100	> 1000	Very Good	Needs Model Rewrite
TP + PP + DP	> 1000	> 1000	Very Good	Needs Significant Model Rewrite
ZeRO	> 1000	> 1000	Very Good	Great

これは、一般的な並列化手法とZeROとの比較です。一番上が、基本的なデータ並列ですが、実行効率やスケーラビリティを保ったまま、はるかに大きなパラメータを持つモデルを訓練できるようになります。

ZeRO-Offload - 誰もが巨大なモデルを学習できるように -

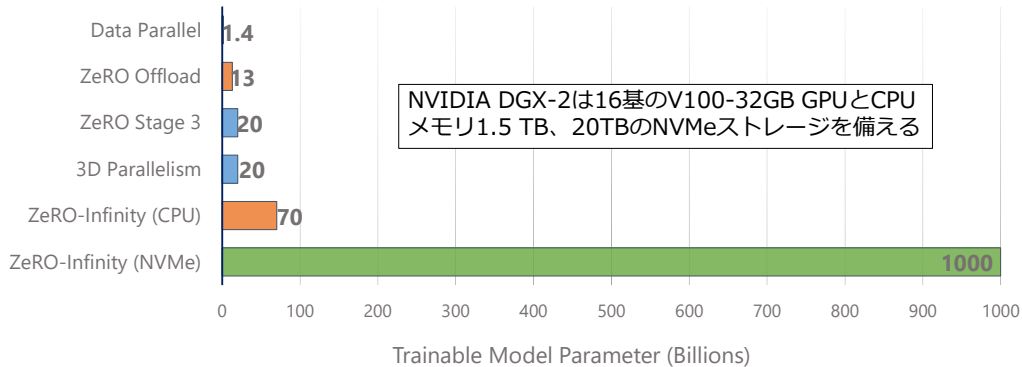
- 学習パラメータ等のデータをGPUメモリからCPUメモリにオフロード
- ZeRO-3との組合せにより、**400億パラメータのモデルを一基のGPUだけで学習**



これまで説明したZeROに加えて、さらに大きなモデルを訓練するための技術がZeRO-Offloadです。学習パラメータ等のデータを、CPUメモリにオフロードすることで、400億パラメータという極めて大規模なモデルでも、1基だけのGPUで学習が可能となっています。こうした技術によって、高価なGPUを多数持つ大企業でなくても、巨大なモデルを学習できます。オフロードには一般的に大きなオーバーヘッドが伴いますが、各種の技術的な工夫により、良好なスケーラビリティを示しています。

ZeRO-Infinity - 極限のモデルサイズへの挑戦 -

- CPUメモリとNVMeストレージの両方にオフロード
- 多くの省メモリ技術を組合せ: ZeRO Stage 3, チェックポインティングなど



1基のNVIDIA DGX-2で学習できる最大モデルサイズ

さらに極限のモデルサイズでの訓練のため、NVMeストレージへのオフロードを行うのが、ZeRO-Infinityです。NVIDIA DGX-2という、16基のGPUを備えたサーバを1台で、1兆パラメータという極端に大きなモデルを学習できることを確認しています。

When/Where to use ZeRO

- 誰でもが巨大モデルを訓練できるように、限られた資源で実行
- バッチサイズ（一度に処理するデータ件数）が大きくなり、効率が改善

Who uses ZeRO

- スタートアップ含む多くの企業が取り組みを紹介
- 主に自然言語処理、StableDiffusionのファインチューニングなど



ZeROは極めて汎用性の高い技術で、これを用いると、従来手法と比べてはるかに巨大なモデルを訓練できます。また、極端に大きなモデルでなくても、学習パラメータのメモリ使用量が減少すると、学習の1サイクルで一度に処理するデータ件数、これをバッチサイズといいます、それを大きくできます。GPUはその仕組み上、多くの件数のデータを処理すると効率が改善するため、実際には省メモリだけでなく、スループットを改善する効果が得られることがあります。ZeROは実際に広く使われており、Webで探しても多くの利用例がみつかります。

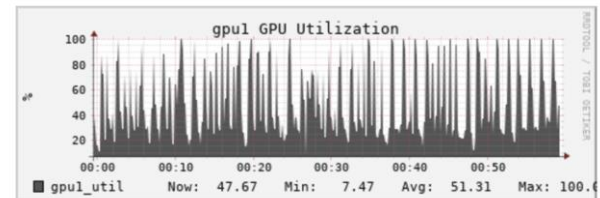
Usecase from Databricks Blog

- [Hugging FaceとDeepSpeedによる大規模言語モデルのファインチューニング](#)

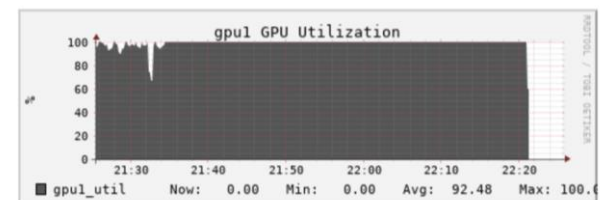
- DeepSpeedによる省メモリ化で大きなバッチサイズで学習が可能に
 - GPUの実行効率が改善
 - 実行時間を1/3に短縮
 - = **コストも1/3!**

深層学習の訓練は
Azureのようなクラウドで
実行されることも多い

DeepSpeedを使用しない場合：



With DeepSpeed:



企業の使用例として、Databricks社のブログに乗っていた例をご紹介します。ここでは、ChatGPTのような大規模モデルの訓練に使われており、DeepSpeedで省メモリ化した結果、バッチサイズが大きくなり、実行時間が1/3になったと記載されています。一定規模のGPUサーバやクラスターは、Azureのようなクラウドで実行されることも多いと思われますが、そこでは実行時間と金銭的なコストは直結していますので、つまりコストを1/3にできたということになります。

How to use ZeRO

- チュートリアル:
 - [Zero Redundancy Optimizer - DeepSpeed](#)
 - [ZeRO-Offload - DeepSpeed](#)
- 多くのフレームワークと結合: [Hugging Face Transformers](#), [Hugging Face Accelerate](#), [PyTorch Lightning](#), [MosaicML Composer](#), [Determined AI](#).



```
# construct torch.nn.Module
model = MyModel()

# wrap w. DeepSpeed engine
engine, *_ = deepspeed.initialize(
    model=model,
    config=ds_config)

# training-loop w.r.t. engine
for batch in data_loader:
    loss = engine(batch)
    engine.backward(loss)
    engine.step()
```

```
ds_config = {
  "optimizer": {
    "type": "Adam",
    "params": {"lr": 0.001}
  },
  "zero": {
    "stage": 3,
    "offload_optimizer": {
      "device": "cpu|nvme"
    },
    "offload_param": {
      "device": "cpu|nvme"
    }
  }
}
```

PyTorchを用いたプログラムに対し、数行のコードの変更でDeepSpeedを適用

ZeROの使用方法については、Webにチュートリアルもありますし、HuggingFaceのTransformersといった代表的なフレームワークなどと結合されており、容易に使用できます。もっとも基本的な使い方では、ここに示したように、PyTorchで記載されたプログラムを数行変更するだけで、ZeROを適用できます。

How to use ZeRO

- ブログ:

- [ZeRO & DeepSpeed: New system optimizations enable training models with over 100 billion parameters](#)
- [Turing-NLG: A 17-billion-parameter language model by Microsoft](#)
- [ZeRO-2 & DeepSpeed: Shattering barriers of deep learning speed & scale](#)
- [DeepSpeed: Extreme-scale model training for everyone - Microsoft Research](#)
- [DeepSpeed ZeRO-3 Offload - DeepSpeed](#)
- [ZeRO-Infinity and DeepSpeed: Unlocking unprecedented model scale for deep learning training - Microsoft Research](#)

- 論文:

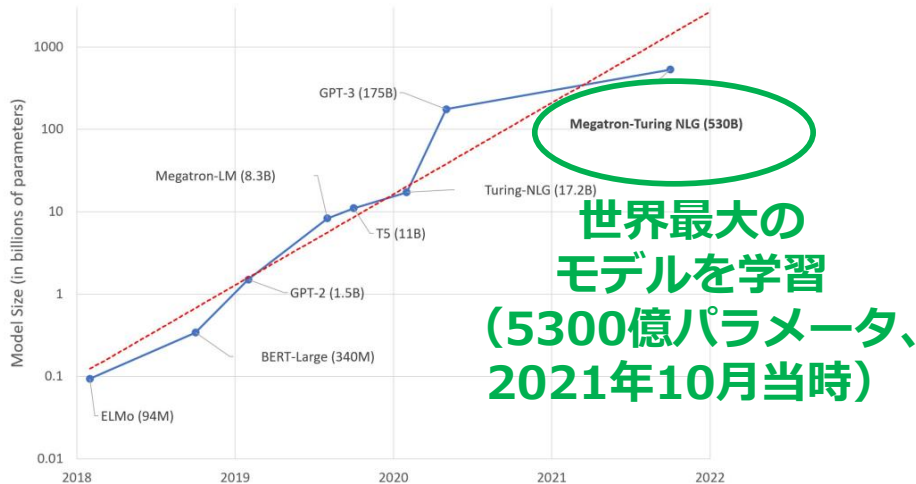
- [ZeRO: memory optimizations toward training trillion parameter models](#)
- [ZeRO-Offload: Democratizing Billion-Scale Model Training](#)
- [ZeRO-infinity: breaking the GPU memory wall for extreme scale deep learning](#)

ZeROについては、ほかにも多くのブログ記事や論文があります。ご興味のあるかたは、こちらもお覧になってください。

3D Parallelism

次に3D Parallelismという技術について説明します

Why 3D Parallelism - 超大規模モデルへのスケールアップ -

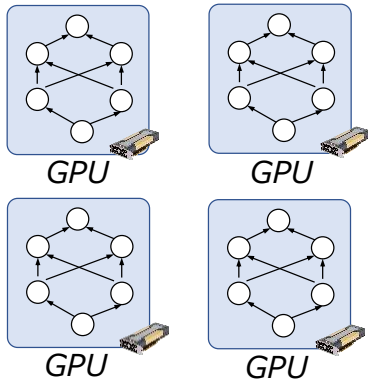


数千億パラメータ規模のモデル・数千GPU規模の学習には、ZeROに加えて、複数種類の並列化の組合せが必要

ZeROは、データ並列の省メモリ化で大規模なモデルを訓練可能にするもので、特に、ZeRO-Infityでは、1兆規模のパラメータのモデルでの訓練を可能にしています。しかし、モデルサイズに加え、訓練データサイズも大きく、数千台という規模のGPUで学習を行うには、ZeROだけでは不十分で、複数種類の並列化の組み合わせが必要です。DeepSpeedのチームでは、これから紹介する3D parallelismを用いて、2021年に、5300億パラメータという当時世界最大サイズのモデルを学習しました。

深層学習の並列化

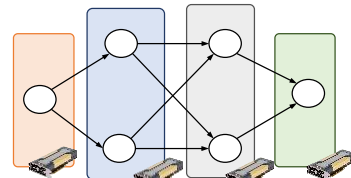
データ並列



- 高い効率を容易に実現
- モデル全体をGPUに複製 → 極めて大きなモデルは学習不可能

モデル並列

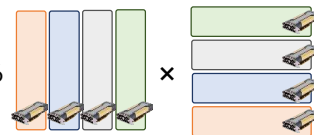
パイプライン並列



深層学習モデルをレイヤごとに分割し
パイプラインで計算

テンソル並列

内部の計算に使われる
巨大な行列を分割



- 大規模なモデルもGPUメモリに格納可能
- モデルの書き換えが必要・高い効率の実現できるモデルに限られる

深層学習の並列化方式は、さきほど紹介したように、データ並列とモデル並列があり、モデル並列はさらに、主にパイプライン並列、テンソル並列と呼ばれる方式が使われています。

What is 3D Parallelism

- ZeROを用いたデータ並列 + パイプライン並列 + テンソル並列の組合せにより、数千GPUを用いた1兆以上のパラメータのモデルが訓練可能に
- ほぼ理想的のスケラビリティを達成

	Max Parameter (in billions)	Max Parallelism	Compute Efficiency	Usability (Model Rewrite)
Data Parallel (DP)	Approx. 1.2	>1000	Very Good	Great
Tensor Parallel (TP)	Approx. 20	Approx. 16	Good	Needs Model Rewrite
TP + DP	Approx. 20	> 1000	Good	Needs Model Rewrite
Pipeline Parallel (PP)	Approx. 100	Approx. 128	Very Good	Needs Model Rewrite
PP + DP	Approx. 100	> 1000	Very Good	Needs Model Rewrite
TP + PP + DP	> 1000	> 1000	Very Good	Needs Significant Model Rewrite
ZeRO	> 1000	> 1000	Very Good	Great

組合せ
+

3D parallelismは、名前のとおり、その3つの並列化方式をすべて組み合わせ、さらにはZeROを併用した方式になります。通常、多くのGPUを用いると、オーバーヘッドも大きくなるのですが、こうした並列化の組み合わせと、各種の実行効率を向上させる工夫により、数百・数千のGPUを使用しながら、理想的なスケラビリティを達成しています。

When/Where to use 3D Parallelism

5300億パラメータモデルの訓練

[Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, the World's Largest and Most Powerful Generative Language Model - Microsoft Research](#)

日本経済新聞

朝刊・夕刊 LIVE Myニュース 日経会社情報

トップ 速報 オピニオン 経済 政治 ビジネス 金融 マーケット マネーのまなび テック 国際 スポーツ 社会・調

大規模言語AIにGoogleやMicrosoft覇権争い 日中に波及

日経産業新聞 + フォローする
2022年2月6日 2:00 [有料会員限定]

保存

NIKKEI
BUSINESS DAILY
日経産業新聞

人間のように巧みに言葉を操る人工知能（AI）は作れるのか。AI開発は米アルファベット傘下のグーグルや、米マイクロソフトなど世界のIT（情報技術）のフロントランナーがこぞって力を入れる「大規模言語モデル」が競争の主戦場となっている。実現

<https://www.nikkei.com/article/DGXZQOUC2175R0R20C22A1000000/>

Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, the World's Largest and Most Powerful Generative Language Model

Published October 11, 2021

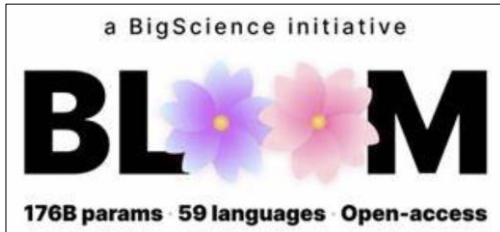
By [Ali Alvi](#), Group Program Manager (Microsoft Turing); [Paresh Kharya](#), Senior Director of Product Management, Accelerated Computing, NVIDIA

マイクロソフトは米エヌビディアと21年10月に5300億のパラメーターを持つAIの開発を明らかにした。曖昧な表現の理解などに優れるという。

DeepSpeedのチームとNVIDIAとの共同で実施した、5300億パラメータモデルの学習は、当時の世界最大サイズだったこともあり、日本でも日経新聞のような一般紙で報道されるなど、話題になりました。

Who uses 3D Parallelism

多くの大規模言語モデル（LLM）は3D-Parallelismを使用: [Jurassic-1 \(178B\)](#), [BLOOM \(176B\)](#), [GLM \(130B\)](#), [YaLM \(100B\)](#).



AI21labs
**Announcing AI21 Studio and Jurassic-1
language models**

AI21 Labs' new developer platform offers instant access to our 178B-parameter language model, to help you build sophisticated text-based AI applications at scale

3D parallelismは、その5300億パラメータのモデル以外にも、BLOOMなど、多くの大規模言語モデルの学習で使われています。

How to use 3D Parallelism

- ブログ:
 - [DeepSpeed: Extreme-scale model training for everyone - Microsoft Research](#)
 - [Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, the World's Largest and Most Powerful Generative Language Model - Microsoft Research](#)
- チュートリアル: [Pipeline Parallelism - DeepSpeed](#)
- 論文: [\[2201.11990\] Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model \(arxiv.org\)](#)

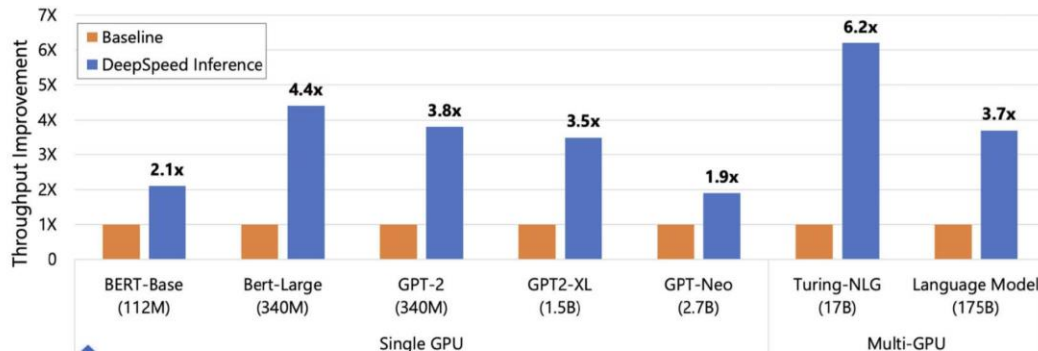
3D Parallelismについても、ブログやチュートリアル、論文がありますので、詳細についてはそちらを確認ください。

DeepSpeed Inference & DeepSpeed-MII

次に、DeepSpeed Inferenceと、DeepSpeed-MIIについて紹介します。

Why DeepSpeed Inference?

- 推論における課題を解決: 1) 大規模モデルの推論のための複数GPU利用, 2) 小さいバッチサイズでの非効率さ, 3) 量子化の適用
- 大規模なTransformerモデルに適用し、最大6倍の高速化（低コスト化）

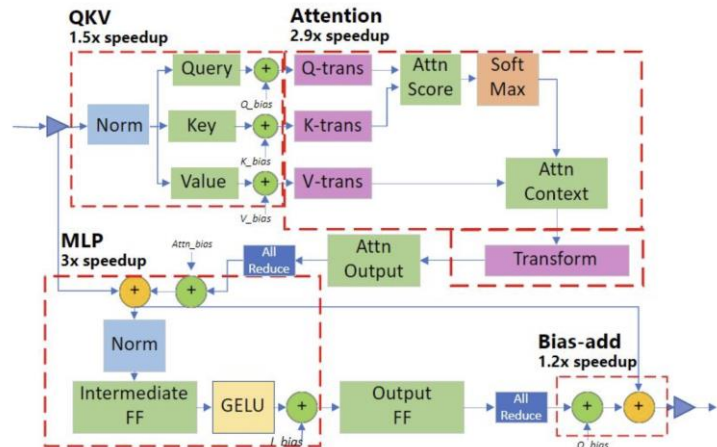


Inferenceとは、このプレゼンテーションでいうところの推論という用語の英語です。

初めに少し触れたように、推論は、訓練より必要な計算やメモリ量は小さいものの、訓練とは違う課題があります。具体的には、大規模モデルのための複数GPU利用、小さいバッチサイズでの非効率、量子化の適用などがあげられます。DeepSpeedは、DeepSpeed Inferenceと呼ぶ推論のための機能を提供しており、この図では、大規模なTransformerに適用し、最大6倍の高速化を確認したという結果を示しています。実際にユーザが使用するサービスで使われるのは基本的に推論であり、推論の高速化は、直接的にサービスの提供コストの削減に貢献します。

What is DeepSpeed Inference?

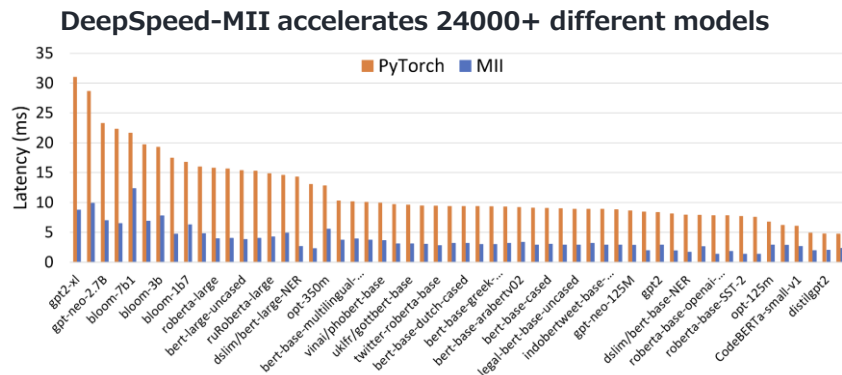
- 推論のためのテンソル並列
- 計算カーネル（GPU実行プログラム）を推論のために最適化し、メモリロード時間を短縮
- 柔軟な量子化のサポート（省ビットでの計算によるデータサイズ削減&計算高速化）



具体的には、DeepSpeed Inferenceでは、推論のためのテンソル並列、計算カーネルと呼ばれる、GPU実行のためのプログラムの最適化、並びに量子化と呼ばれる、少ないビット数での計算のサポートなどの特徴を備えています。

What is DeepSpeed-MII?

- DeepSpeed Inferenceの最適化された推論機能を、数万の深層学習モデルで利用可能に
- MIIで利用することで、元の実装と比べて大幅に低レイテンシ・低コスト化



また、DeepSpeed-MIIは、DeepSpeed Inferenceの推論機能を、数万の深層学習モデルで、容易に利用するためのツールです。MIIからモデルを利用することで、元のモデルの実装を変更することなく、大幅に低レイテンシ・低コスト化できます。この図は、様々なモデルで、もともとのモデルの実装に比べて、MIIを使用すると、大幅にレイテンシが減少したことを示しています。


When/Where to use DeepSpeed Inference and MII?

- HuggingFace (or widely-used model architectures)のモデルを高速化
- 推論サーバ (gRPC, RESTサービス) を簡単に立ち上げ

Who uses DeepSpeed Inference and MII?

- 多くの利用例が報告
- 言語モデル、StableDiffusion, ...

Advancing Machine Learning with DeepSpeed MII and Stable Diffusion

 Jeannine Proctor · Follow
Published in [Bootcamp](#) · 5 min read · Apr 10

Accelerate Stable Diffusion inference with DeepSpeed-Inference on GPUs

[#DIFFUSION](#) [#DEEPSPEED](#) [#HUGGINGFACE](#) [#OPTIMIZATION](#)

November 7, 2022
11 min read
[View Code](#)

In this session, you will learn how to optimize Stable Diffusion for Inference using Hugging Face [Diffusers library](#), and [DeepSpeed-Inference](#). The session will show you how to apply state-of-the-art optimization techniques using [DeepSpeed-Inference](#). This session will

Incredibly Fast BLOOM Inference with DeepSpeed and Accelerate

Published September 16, 2022

[Update on GitHub](#)

 stas · [Follow](#)
 seuss · [Follow](#)
Stas Bekman · Sylvain Gugger

This article shows how to get an incredibly fast per token throughput when generating with the 176B parameter [BLOOM model](#).

Deploying LLMs with Vertex AI

Welcome to an exciting journey of deploying Large Language Models with DeepSpeed on Vertex AI!

 Taha Binhuraib · Follow
6 min read · Apr 23

DeepSpeed InferenceとDeepSpeed-MIIがよく活用されるシーンとしては、HuggingFaceのようなレポジトリで提供されているモデルの高速化があります。また、推論機能をgRPCなどのサーバとして簡単に提供する機能も、サービス提供のために使用されます。実際に利用例は多く報告されています。言語モデルのほか、テキストから画像を生成するStableDiffusionといったモデルも、その高速性から、広く取り上げられています。

How to use DeepSpeed Inference/DeepSpeed-MII?

推論サーバを起動

```
import mii
mii_configs = {"tensor_parallel": 1, "dtype": "fp16"}
mii.deploy(task="text-generation",
           model="bigscience/bloom-560m",
           deployment_name="bloom560m_deployment",
           mii_config=mii_configs)
```

推論サーバの呼び出し

```
import mii
generator = mii.mii_query_handle("bloom560m_deployment")
result = generator.query({"query": ["DeepSpeed is", "Seattle is"]}, do_sample=True, max_new_tokens=30)
print(result)
```

- [DeepSpeed-MII Repository](#) (使い方)
- [DeepSpeed Deep Dive — Model Implementations for Inference \(MII\)](#) (外部ユーザによるチュートリアル)

これは推論のサーバを起動したり、使用するコード例です。本当にこれだけのコードで、文章生成をはじめとする推論を実行するサーバを起動できます。より詳しい使い方については、GitHubレポジトリにある使い方を参照ください。また、外部ユーザが書いてくれたチュートリアルもあります。

How to use DeepSpeed Inference/DeepSpeed-MII?

- ブログ・チュートリアル:
 - [DeepSpeed: Accelerating large-scale model inference and training via system optimizations and compression - Microsoft Research](#)
 - [ZeRO-Inference: Democratizing massive model inference – DeepSpeed](#)
 - [DeepSpeed-MII: instant speedup on 24,000+ open-source DL models with up to 40x cheaper inference – DeepSpeed](#)
 - [Stable Diffusion Image Generation under 1 second w. DeepSpeed MII](#)
 - [Getting Started with DeepSpeed for Inferencing Transformer based Models - DeepSpeed](#)
 - [Automatic Tensor Parallelism for HuggingFace Models - DeepSpeed](#)
- 論文: [DeepSpeed-inference: enabling efficient inference of transformer models at unprecedented scale](#)

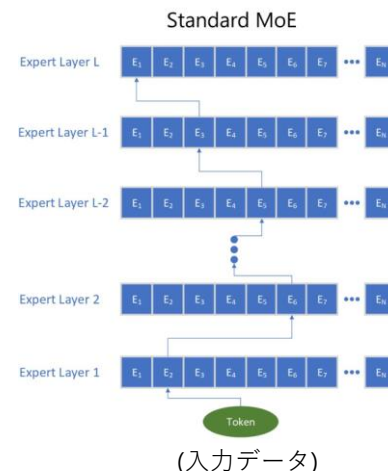
ブログやチュートリアル、論文もあります。技術的な課題やアプローチの詳細については、これらの文献に詳しく記載されています。

DeepSpeed-MoE

これまで主にシステム面にフォーカスした機能を紹介してきました。深層学習の訓練や推論の効率化では、システムだけでなく、使用するモデルのアーキテクチャを工夫したり、あるいはシステムとモデルの両方にまたがるより横断的なアプローチをとる、という方法もあります。ここからは、モデリングのほうにより関係の深い機能について紹介してもらいます。では次の技術として、DeepSpeed-MoEについて紹介します。

Why DeepSpeed-MoE?

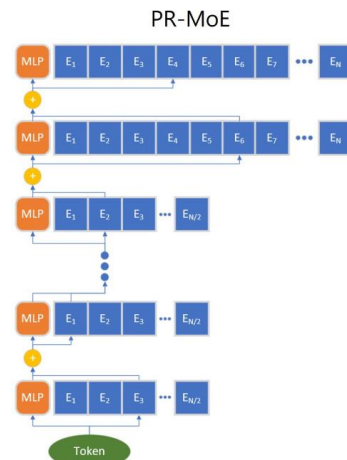
- ZeROや3D parallelismを用いても、大規模モデルの訓練は膨大な計算量、時間、コストが必要
- Mixture of Experts (MoE) と呼ばれる新しいタイプのモデルアーキテクチャでは、入力データに応じてモデルの一部のみを計算
- 一方で課題も：
 - 生成モデル (GPT-3等) での効果が実証されていない
 - メモリは大量に必要
 - 推論の性能が低い



まず背景として、ZeROや3D Parallelismを使用しても、膨大なパラメータを持つ超巨大なモデルの訓練には、同じく膨大な計算量、時間、それに伴うコストが必要です。一方で、訓練を効率化する手法として、近年、Mixture of Experts、略してMoEという新しいモデルアーキテクチャが提案されています。MoEは、必要に応じて、モデルの一部だけを計算することで、計算効率を向上させる手法です。右側の図に示すように、MoEでは、モデルのそれぞれのレイヤで、複数のexpertと呼ばれる単位を持っています。MoEモデルに入力データが渡されると、それぞれのデータごとに特定のexpertだけに渡され、計算はそのexpertだけで行われます。MoEは非常に多数のパラメータを持つモデルを、少ない計算量で処理できることから、近年注目されていますが、課題もいくつかあります。一つは、GPT-3のような生成モデルでの効果が実証されていないこと、次に、多数のExpertを持つために、モデルが大きくなり、メモリが大量に必要なこと、最後に、その独特な仕組みのため、推論の性能が低いことが挙げられます。

What is DeepSpeed-MoE?

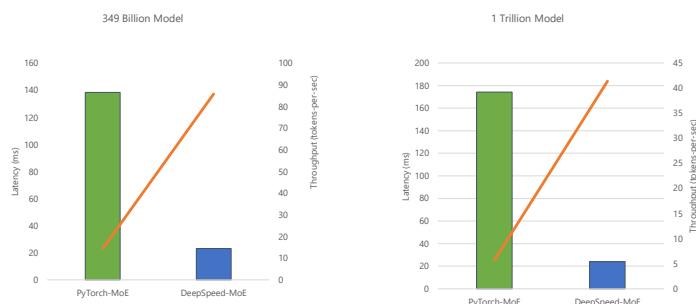
- MoEを生成モデル（GPT-3等）に適用し、**1/5の訓練コスト**で同様のモデル品質を達成
- 新しいアーキテクチャ Pyramid-Residual MoE (PR-MoE): モデルの品質を落とすことなく、**モデルサイズを1.7ー3.2倍縮小**
- Mixture-of-Student: 蒸留(distillation)と呼ばれる技術で、レイヤ数を削減、99%の性能を維持したまま**モデルサイズを1.9倍から3.7倍縮小**



DeepSpeed-MoEは、それらの3つの課題を解決します。まずDeepSpeed-MoEは、GPT-3のような生成モデルに適用して、訓練コストを1/5に削減する一方で、従来手法と同様のモデル品質を達成しました。また、Pyramid-Residual MoEと呼ぶ、右に示すMoEのための新しいアーキテクチャを導入しました。このPyramid-Residual MoEは、従来のMoEよりも効率が高く、モデルの品質を落とすことなく、MoEモデルのサイズを1.7倍から3.2倍に縮小することができました。最後に、蒸留と呼ばれる、モデルの規模を更に縮小するための技術を適用したMixture-of-Studentという仕組みによって、99%の性能を維持したまま、モデルサイズを1.9倍から3.7倍に縮小しました。

What is DeepSpeed-MoE?

- MoEの推論を高速化
 - 異なる並列化方式を効率的に組合せ
 - 通信オーバーヘッドを最小化
 - カーネルの最適化により、GPUメモリバンド幅利用を最大化
- 推論を**7.2倍高速化**、**1兆パラメータのモデルを25ms**で計算



また、DeepSpeed-MoEでは、従来性能が低かったMoEでの推論を最適化しています。具体的には、複数の並列化方式を組み合わせたほか、通信オーバーヘッドの最小化、GPUのメモリバンド幅を向上させるためのカーネルの最適化などを行いました。こうした最適化の結果、推論を7.2倍高速化し、1兆パラメータの超巨大なモデルでも、25ミリ秒での推論を可能にしました。

DeepSpeed-MoE

- When/Where to use DeepSpeed-MoE?
 - MoEモデルを効率的に訓練・推論
- Who uses DeepSpeed-MoE?
 - 活用例: [New Z-code Mixture of Experts models improve quality, efficiency in Translator and Azure AI](#)
- How to use DeepSpeed-MoE?
 - ブログ:
 - [DeepSpeed powers 8x larger MoE model training with high performance](#)
 - [DeepSpeed: Advancing MoE inference and training to power next-generation AI scale](#)
 - チュートリアル:
 - [Mixture of Experts – DeepSpeed](#)
 - [Mixture of Experts for NLG models – DeepSpeed](#)
 - [Getting Started with DeepSpeed-MoE for Inferencing Large-Scale MoE Models](#)
 - 論文: [DeepSpeed-MoE: Advancing Mixture-of-Experts Inference and Training to Power Next-Generation AI Scale](#)

このように、DeepSpeed-MoEは、MoEモデルの訓練と推論の効率を向上させるものです。DeepSpeed-MoEは、すでにMicrosoft社内外で使用されています。ここでは社内の利用例として、Azure AIの翻訳サービスで使われている、Z-code MoEモデルを挙げています。使い方や技術的な詳細については、こちらのブログ、チュートリアル、論文をご覧ください。

Communication Compression

次に、Communication Compressionという技術について説明します

Why Communication Compression?

- 複数のGPUを用いる分散深層学習の訓練では、通信が大きなボトルネック（特に遅いネットワークの場合）
- モデル更新に用いるAdamやLAMBなどのアルゴリズムを分散深層学習で用いる際、GPU間の通信が頻繁に発生
- BERT（言語処理分野の代表的モデル）の訓練において、通信の占める時間は、InfiniBandまたはEthernetの場合でそれぞれ最大 **52%** または **91%**

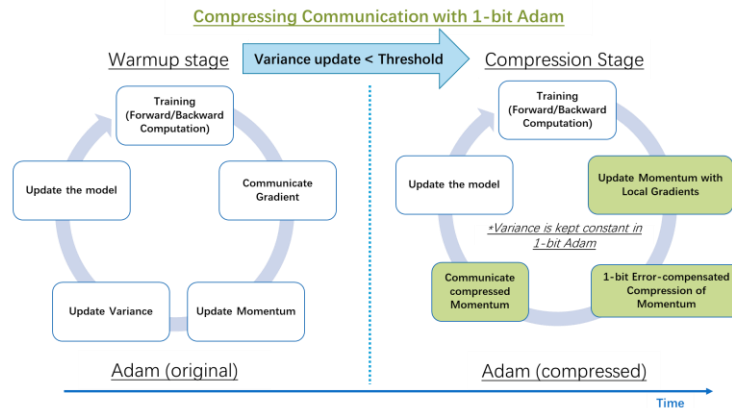
TABLE II
BERT-LARGE PRE-TRAINING SEQLEN 128 PROFILING RESULTS.

Num. node/ GPU	Batch size	Forward (ms)	Backward allreduce (ms)	Backward everything else (ms)	Step (ms)	all-reduce%
Ethernet Cluster:						
64/256	8K	55	3579	117	191	91%
64/256	64K	445	3674	919	215	70%
32/128	8K	112	3759	233	121	89%
16/64	8K	223	3433	464	109	81%
8/32	8K	445	3528	923	38	72%
4/16	8K	881	3436	1827	33	56%
2/8	8K	1773	2087	3696	31	28%
1/4	8K	3532	234	7329	30	2%
InfiniBand Cluster:						
16/128	8K	96	335	179	36	52%
16/128	64K	770	422	1422	32	16%
8/64	8K	192	332	352	34	36%
4/32	8K	384	339	711	31	23%
2/16	8K	768	270	1436	31	11%
1/8	8K	1534	167	2869	31	4%

複数のGPUを用いる分散深層学習の訓練では、特にネットワークが遅い場合、通信が大きなボトルネックになります。具体的には、モデル更新に用いるAdamやLAMBなどのアルゴリズムを分散深層学習で用いる際、GPU間の通信が頻繁に発生します。我々の実験では、言語処理分野の代表的モデルであるBERTの訓練で、通信の占める時間は、InfiniBandまたはEthernetネットワークの場合でそれぞれ最大52% または 91%もの割合を占めていました。

What is Communication Compression?

- 1-bit Adam, 1-bit LAMB, 0/1 Adam
 - 学習パラメータ更新の代表的アルゴリズムであるAdamやLAMBを拡張
 - 圧縮によって通信を大幅に効率化



こうした通信オーバーヘッドを削減するため、我々は1-bit Adam, 1-bit LAMB, 0/1 Adamと呼ぶ三つのアルゴリズムを考案しました。これらは、広く使用されているAdamやLAMBアルゴリズムで、圧縮によって通信を大幅に効率化するものです。例えば1-bit Adamの場合、図に示すように、2段階のアルゴリズムになっています。まずはじめに、warmupの段階では、オリジナルのAdamを使用します。これは、訓練の初期段階では、一般に訓練が不安定なためです。第2段階では、通信の前にAdamアルゴリズムが内部にもつmomentumというデータを圧縮するとともに、モデルの学習速度を保つための複数の技術を組み合わせることで、最終的なモデルの品質を落とさずに、通信オーバーヘッドを減らすことができます。

Communication Compression

- When/Where to use communication compression?
 - Ethernetネットワークのクラスタにおいて、BERTの学習を最大3.3倍高速化
- Who uses communication compression?
 - 比較的遅いネットワーク（Ethernet）でDeepSpeedを運用するユーザによって活用
- How to use communication compression?
 - ブログ: [1-bit Adam](#), [1-bit LAMB](#)
 - チュートリアル: [1-bit Adam](#), [1-bit LAMB](#), [0/1 Adam](#)
 - 論文: [1-bit Adam](#), [1-bit LAMB](#), [0/1 Adam](#)

このCommunication Compressionが通信速度が遅いとき特に有効で、実験ではBERTモデルの学習が最大3.3倍高速化されました。比較的遅いネットワークを用いるクラスタを使用しているユーザによって利用されています。使い方や技術的な詳細については、こちらのブログ、チュートリアル、論文をご覧ください。

DeepSpeed Data Efficiency

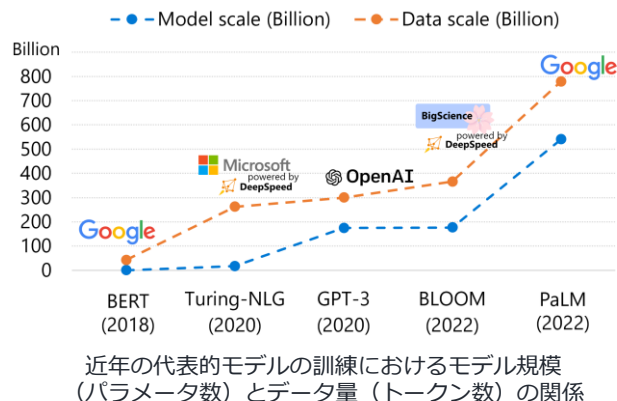
次に紹介するのは、DeepSpeed data efficiencyです。

Why DeepSpeed Data Efficiency?

- モデルサイズに加え、訓練データ量も急速に拡大
- 訓練コストは、モデルサイズ・訓練データ量の両方に比例して増加

- “データ効率”の改善が重要:

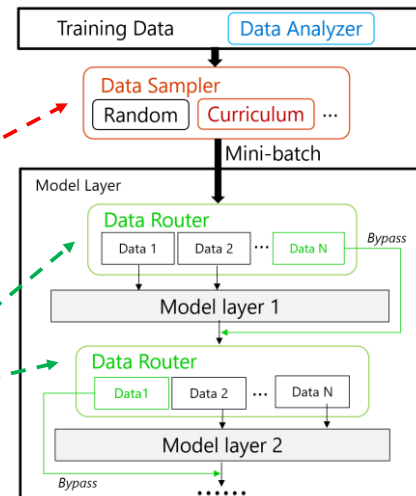
- 少ないデータ量 (= 少ない訓練コスト) で同等の性能を実現
- 同じデータ量でより良い性能を実現



DeepSpeed data efficiencyが必要な理由として、モデルサイズに加え、訓練データ量も急速に拡大している一方で、訓練コストは、モデルサイズ・訓練データ量の両方に比例して増加することが挙げられます。この図では、過去5年の代表的モデルの訓練において、モデル規模と、訓練データ量がいずれも急速に増大していることを示しています。そこで、データ効率を改善する、つまり、少ないデータ量で同等の性能を実現する、または、同じデータ量でより良い性能を実現することが重要になります。

What is DeepSpeed Data Efficiency?

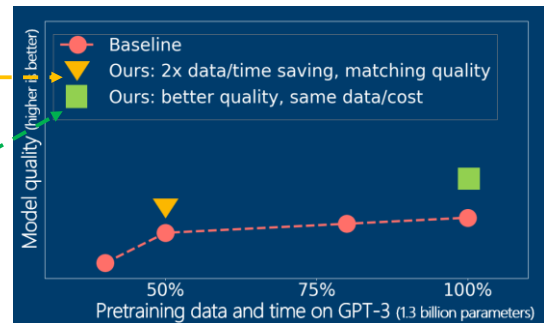
- データをより効率的に活用し、訓練効率とモデル性能を向上させるためのライブラリ
- **curriculum learning**により効率的にデータをサンプリング
 - 簡単なデータから開始し、徐々に難しいデータを学習させる
- モデルレイヤごとに一部の入力データのみで学習させる**random layerwise token dropping (random-LTD)**.
 - 入力データの一部の計算をスキップ



ここで紹介するDeepSpeed data efficiency は、データをより効率的に活用し、訓練効率とモデル性能を向上させるためのライブラリです。右の図に示すように、DeepSpeed data efficiency は、二つの技術から構成されています。一つは curriculum learning という技術で、効率的に訓練データをサンプリングします。具体的には、簡単なデータから訓練を開始し、徐々に難しいデータを学習させます。もう一つは、random layerwise token dropping と呼んでいる技術で、モデルのレイヤごとに入力されるデータを部分的にスキップし、一部のデータのみで学習しデータ効率を改善する。

When/Where to use DeepSpeed Data Efficiency?

- Transformer系のモデルで効果を実証 (GPT-2/3, BERT, ViT).
- GPT-3/BERTでの事前学習(pre-training)において、**半分のデータ量 (= 訓練時間を半減) で同等の性能を達成**
- 同量のデータを用いると、**性能も改善**
- curriculum learningをカスタマイズし、他のタイプのモデルにも適用可能



DeepSpeed data efficiency を使用すると、訓練コストを大幅に節約できます。我々は、GPT-2, GPT-3, BERT, ViTなどのTransformer系のモデルで効果を実証しました。GPT-3とBERTモデルでは、モデルの品質を保ちつつ、半分の訓練データ量を用いて、つまり半分の訓練時間で訓練できることを示しました。さらに、同じ訓練データ量で、より高いモデル品質が得られることも確認しました。curriculum learningに関しては、異なる方法を用いるようにカスタマイズし、他のタイプのモデルにも適用可能です。

DeepSpeed Data Efficiency

Who uses DeepSpeed Data Efficiency?

- Microsoft社内のモデルの訓練に適用
- Sequence Length Warmup(curriculum learning提案手法の一部) は [Amazon](#), [MosaicML](#), [Yandex](#) などを利用

How to use DeepSpeed Data Efficiency?

- ブログ: [DeepSpeed Data Efficiency](#)
- チュートリアル: [DeepSpeed Data Efficiency](#)
- 論文:
 - [The Stability-Efficiency Dilemma: Investigating Sequence Length Warmup for Training GPT Models](#)
 - [Random-LTD: Random and Layerwise Token Dropping Brings Efficient Training for Large-scale Transformers](#)
 - [DeepSpeed Data Efficiency: Improving Deep Learning Model Quality and Training Efficiency via Efficient Data Sampling and Routing](#)

DeepSpeed data efficiencyは、Microsoft社内のモデルの訓練に適用されています。また、curriculum learningの一手法として提案した、Sequence Length Warmupという技術は、Amazon, MosaicML, Yandexなどで利用されています。使い方や技術的な詳細については、こちらのブログ、チュートリアル、論文をご覧ください。

DeepSpeed Compression

次に、DeepSpeed Compressionについて紹介します。

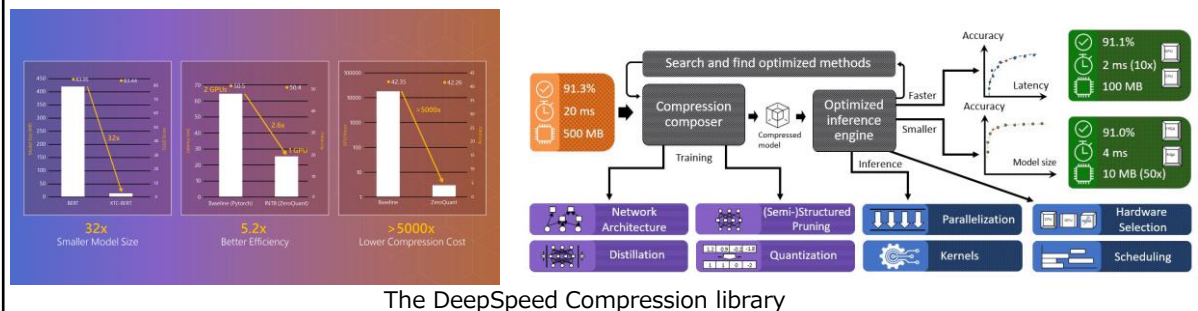
Why DeepSpeed Compression?

- 大規模モデルは推論のコストも大きい
- 前述の通りDeepSpeedは推論も高速化できるが、モデルサイズ（および必要な計算量）は変わらない
→ モデルを圧縮 (compression) できれば、もっと速くできる
- モデル圧縮技術の課題
 - 処理パイプラインが複雑で利用しづらい
 - 圧縮のコストが高い
 - 十分に最適化されたシステムがない
 - 複数の圧縮技術の組合せが難しい

DeepSpeed Compressionが必要となる理由として、大規模モデルでは、推論のコストも大きいことが挙げられます。先に説明した通り、DeepSpeedは推論も効率化しますが、モデルサイズや、モデルサイズによって決まる必要な計算量が変わるわけではありません。そこで、モデルを圧縮することができれば、さらに高速化することを考えます。モデル圧縮技術自体は従来からありますが、従来の技術には複数の問題があります。第一に、既存手法は複雑な処理パイプラインが必要で、利用しづらいことが挙げられます。また、圧縮自体にコストがかかる、実際にレイテンシ短縮の効果を得られるほどにシステム面で十分に最適化されたものがないという問題もあります。さらには、複数の圧縮技術を組み合わせることが困難、という課題もあります。

What is DeepSpeed Compression?

- 新たな圧縮技術
 - XTC: モデルサイズを 1/32 に
 - ZeroQuant: 圧縮のコストを 1/5000 に
- DeepSpeedの最適化された推論システムと統合
- 異なる圧縮技術を組合せ可能



DeepSpeed Compressionは、こうした課題を解決するため、最先端の圧縮技術を提供します。例えば、XTCでは、モデルサイズを 1/32 に縮小します。また、ZeroQuantは、圧縮のためのコストを1/5000にします。また、DeepSpeed Compression はDeepSpeedとシームレスに統合されており、圧縮されたモデルはDeepSpeedの推論システムから効率的に利用できます。最後に、DeepSpeed Compressionでは、複数の圧縮のための手法を組み合わせでき、より大きな効果を得られるようになっています。

DeepSpeed Compression

- When/Where to use DeepSpeed Compression?
 - 大規模モデルを低コストに推論する際に有効
- Who uses DeepSpeed Compression?
 - 静止画像の超解像 (**Bing**) : [Introducing Turing Image Super Resolution](#)
 - 動画の超解像 (**Edge**) : [Video super resolution](#)
- How to use DeepSpeed Compression?
 - ブログ: [DeepSpeed Compression: A composable library for extreme compression and zero-cost quantization - Microsoft Research](#)
 - チュートリアル: [DeepSpeed Model Compression Library - DeepSpeed](#)
 - 論文:
 - [XTC: Extreme Compression for Pre-trained Transformers Made Simple and Efficient](#)
 - [ZeroQuant: Efficient and Affordable Post-Training Quantization for Large-Scale Transformers](#)

DeepSpeed Compression は、巨大モデルの推論を低コストで動かしたいときに特に有効です。これまでに社内外で様々なモデルに利用されてきましたが、利用例としては、BingマップとMicrosoft Edgeのための、画像の解像度を向上させる超解像の技術のモデル推論に貢献しました。使い方や技術的な詳細については、こちらのブログ、チュートリアル、論文をご覧ください。

DeepSpeed-Chat

最後に、DeepSpeed-Chatを紹介します

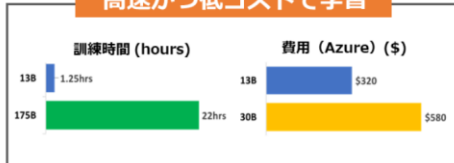
DeepSpeed Chat: ChatGPTライクなモデルを簡単・高速・低コストに、あらゆるスケールで学習



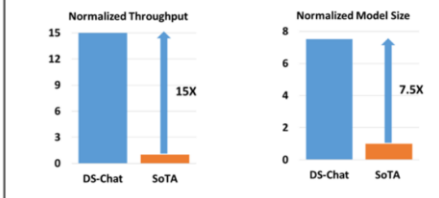
DEEPSPEED CHAT



高速かつ低コストで学習



既存のRLHFシステムと比較して
15倍の高速化、5倍以上大きなモデルを訓練可能



手軽に実行可能

- ChatGPTで用いられるRLHF訓練の完全なend-to-endパイプラインをスクリプト一つで実行

高速・高スケーラビリティ

- ハイブリッドエンジンによって既存のRLHFシステムの15倍の高速化
- あらゆるスケールのモデルで類を見ないコスト削減を達成

大規模モデルのサポート

- GPU1台で100億パラメータ超、複数GPUなら1000億パラメータ超のモデルを学習
- ZeRO, LoRA等の技術を統合

RLHFのための包括的な高速バックエンド

- InstructGPT のパイプラインと様々な大規模モデルのファインチューニングをサポート

DeepSpeed-Chatは、ChatGPTライクなモデルを、簡単・高速・低コスト、かつあらゆるスケールで学習するためのフレームワークです。

Why DeepSpeed-Chat?

- ChatGPTはモデル非公開・OpenAI/Azureで動作
- オープンソースへの強い期待とニーズ
 - それぞれの目的に特化したChatGPTライクなモデルを訓練したい
 - セキュリティ上の理由から、自分の計算環境で動作させたい
- ChatGPTライクなモデルを評価するリーダーボードで、オープンソースモデルの多くが既にDeepSpeedの技術を使用



ChatGPTライクなモデルの訓練を更に効率化するためのオープンソースのフレームワークを提供 : DeepSpeed-Chat

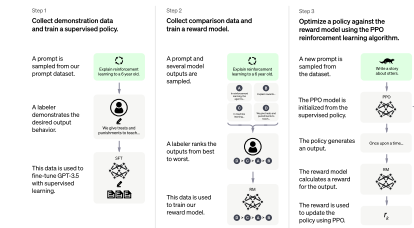
Leaderboard			
Rank	Model	Elo Rating	Description
1	gpt-4	1225	ChatGPT-4 by OpenAI closed source OpenAI/Microsoft
2	claude-v1	1205	Claude by Anthropic closed source Anthropic
3	claude-instant-v1	1153	Claude Instant by Anthropic closed source Anthropic
4	gpt-3.5-turbo	1143	ChatGPT-3.5 by OpenAI closed source OpenAI/Microsoft
5	vicuna-13b	1054	a chat assistant fine-tuned open source, use DeepSpeed technologies
6	vllm-2	1042	PaLM 2 for Chat (chat-bison@001) by Google closed source Google Bard
7	vicuna-7b	1007	a chat assistant fine-tuned open source, use DeepSpeed technologies
8	llama-2-7b	980	a dialogue model for academic research by Meta open source
9	mistral-7b-chat	952	a chatbot fine-tuned from Mistral-7B by Mistral AI open source
10	llm-30b-32k	941	a chat assistant fine-tuned open source, use DeepSpeed technologies
11	alpaca-13b	937	a model fine-tuned from open source, use DeepSpeed technologies
12	RWKV-4-Raven-14B	929	an RWKV with transformer open source, use DeepSpeed technologies
13	mistral-7b-instruct-v0.1	921	an Open Assistant for open source, use DeepSpeed technologies
14	chatgpt-3.5-turbo	921	an open bilingual dialogue open source, use DeepSpeed technologies
15	stability-ai-llm-30b	892	Stability AI language model open source, use DeepSpeed technologies
16	dolly-v2-12b	866	an instruction-tuned open source, use DeepSpeed technologies
17	llama-2-7b	854	open and efficient foundation language models by Meta open source

[Chat with Open Large Language Models \(lmsys.org\)](https://lmsys.org/)

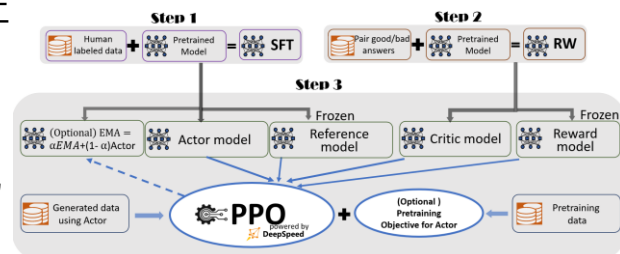
ChatGPTはその公開以来、広い範囲で極めて大きなインパクトを与えていますが、ChatGPTのモデルは非公開であり、OpenAIもしくはMicrosoftのAzureで動作しています。多くの組織、あるいは個人にとって、それぞれの目的に特化したChatGPTライクなモデルを訓練したい、そしてそれをセキュリティ上の理由から、自分の計算環境で動作させたい、というニーズは強くあります。また今日の話の初めに触れましたが、ChatGPTライクなモデルを評価するリーダーボードを見ると、オープンソースモデルの多くが既にDeepSpeedの技術を使用しています。そこで、DeepSpeedチームでは、ChatGPTライクなモデルの訓練を更に効率化するためのオープンソースのフレームワークを提供することになりました。それがDeepSpeed-Chatです。

What is DeepSpeed-Chat

- (OpenAIが公開した情報に基づいて) ChatGPTライクなモデルを3つのステップで訓練するフレームワーク
 - a) 教師付きファインチューニング (Supervised fine-tuning, SFT)
 - b) 報酬モデルのファインチューニング
 - c) RLHF訓練 (Reinforcement Learning with Human Feedback)
- DeepSpeed-Chatで高速かつ低コストでRLHF訓練が可能



ChatGPT

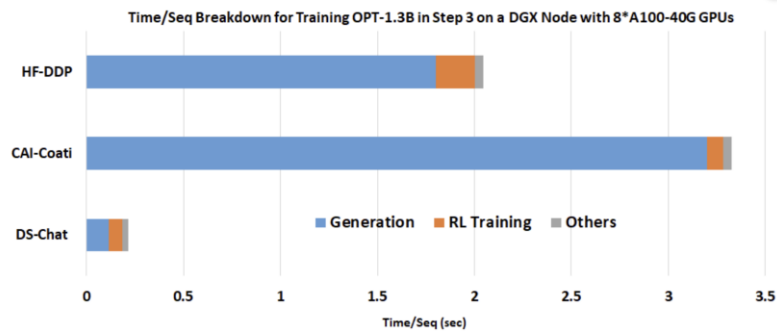


DeepSpeed-Chat

DeepSpeed-Chatは、OpenAIが公開した情報に基づいて、ChatGPTライクなモデルを3つのステップで訓練するフレームワークです。ステップ1とステップ2は、大規模モデルの従来のファインチューニング訓練と似ています。一方、ステップ3のRLHF訓練は、最も複雑な処理を行う部分で、最も訓練コストへ影響する部分です。このステップ3のRLHF訓練を大幅に高速化するのは、DeepSpeed-Chatの最大の貢献です。

What is DeepSpeed-Chat

- RLHF訓練では、複数のモデルを扱いつつ、訓練と推論が両方必要
→ 複数のモデルを同時に訓練・推論できるメモリ管理技術、および訓練と推論の効率的な切り替えを実現し、大幅に高速化
- **既存システムの15倍高速**



RLHF訓練の課題として、複数のモデルを扱いつつ、訓練と推論が両方必要となる点が挙げられます。そのため、複数のモデルを同時に訓練・推論できるメモリ管理技術、および訓練と推論の効率的な切り替えが必要となります。DeepSpeed-Chatでは、これらを最適化し、既存システムの15倍高速に訓練を実現しました。

When/Where to use DeepSpeed-Chat

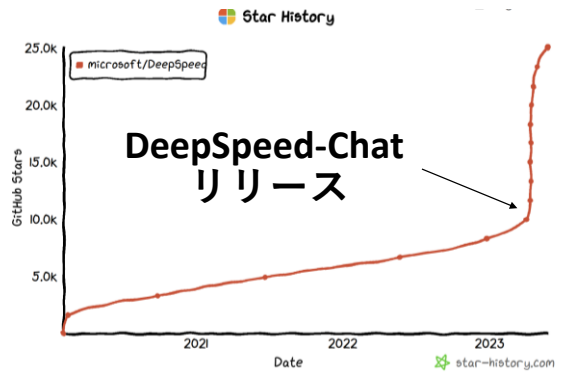
- カスタマイズされたChatGPTライクなモデルを訓練
- 極めて高い実行効率 → 大規模モデルでも低コスト

GPU SKUs	OPT-1.3B	OPT-6.7B	OPT-13.2B	OPT-30B	OPT-66B	OPT-175B
1x V100 32G	1.8 days					
1x A6000 48G	1.1 days	5.6 days				
1x A100 40G	15.4 hrs	3.4 days				
1x A100 80G	11.7 hrs	1.7 days	4.9 days			
8x A100 40G	2 hrs	5.7 hrs	10.8 hrs	1.85 days		
8x A100 80G	1.4 hrs(\$45)	4.1 hrs (\$132)	9 hrs (\$290)	18 hrs (\$580)	2.1 days (\$1620)	
64x A100 80G	31 minutes	51 minutes	1.25 hrs (\$320)	4 hrs (\$1024)	7.5 hrs (\$1920)	20 hrs (\$5120)

Training with 135M tokens on Azure

DeepSpeed-Chatを用いると、ユーザがそれぞれにカスタマイズされたChatGPTライクなモデルを訓練できます。また、これまで何度も触れているように、訓練がAzureのようなクラウドコンピューティングで実行されることを考えると、訓練の実行時間は、コストに直結します。ここで示しているのは、Azureを用いた場合の訓練コストの試算ですが、1750億パラメータという、GPT-3と同じ規模の極端に大きなモデルで、1億以上のトークンデータ量で訓練する場合でも、20時間・5000ドル程度となっています。

Who uses DeepSpeed-Chat?



- 極めて多くのユーザから利用の報告
- ユーザが訓練したモデルがHuggingFace のレポジトリにアップロード



オンラインメディアに多数掲載

DeepSpeed-Chatは、リリース時から非常に多くの注目を集めました。技術的にはかなり専門的なソフトウェアということになると思いますが、先にお伝えしたように、GIGAZINE等のオンラインメディアに多数掲載されました。また冒頭で、現在GitHubのスターが25000以上と記載していましたが、実はDeepSpeed-Chatリリース前は1万前後であったところ、このリリースによって数日で2倍以上という、爆発的な伸びを示しました。多くのユーザが利用について報告しており、またDeepSpeed-Chatを用いて学習したモデルが、HuggingFaceのモデルのレポジトリにアップロードされています。

How to use DeepSpeed-Chat?

コマンド一つで一連の訓練を実行

```
pip install deepspeed>=0.9.0

git clone https://github.com/microsoft/DeepSpeedExamples.git
cd DeepSpeedExamples/applications/DeepSpeed-Chat/
pip install -r requirements.txt

python train.py --actor-model facebook/opt-13b --reward-model facebook/opt-350m --deployment-type single_node
```

訓練したモデルを試すためのサンプルも付属

```
# serve the final model
python chat.py --path ${PATH-to-your-actor-model}
```

[英語ブログ](#), [日本語ブログ](#)
[チュートリアル](#)

Hello, write me a poem about a wise frog.

A wise frog lives in the sky.
He holds court with the other frogs and holds the
meaning of life.
He tells them to avoid traps and look out for each
other.
Frogs are known for their intelligence and se

高度な機能を持つDeepSpeed-Chatですが、使い方は極めて簡単です。環境設定やインストールを済ませれば、Pythonのコマンド一つで3つのステップの訓練が可能なスクリプトが提供されています。また、モデルの訓練後、チャット形式で試すためのサンプルも提供されています。使い方や技術的な詳細については、こちらのブログとチュートリアルをご覧ください。

おわりに

- DeepSpeedの各種技術を簡単にご紹介しました
 - 様々なチャンネルで情報発信や、ユーザ・コラボレータとのコミュニケーションをしています
 - 最新情報を知りたい → Twitterアカウント
[DeepSpeed \(@MSFTDeepSpeed\)](#)
[マイクロソフトDeepSpeed \(@MSFTDeepSpeedJP\)](#) (日本語アカウント)
 - バグレポート等 → [GitHubのIssues](#)
 - 開発に協力する → [GitHubのPull Request \(PR\)](#)
 - 質問・ディスカッション → [GitHubのDiscussion](#)
- GitHubのご連絡・お問い合わせは、英語もしくは英語 + 日本語をお願いします

ここまでDeepSpeedの主要な機能についてご紹介してきました。DeepSpeedチームでは、GitHubやTwitterなど様々なチャンネルで、情報発信や、ユーザ・コラボレータとのコミュニケーションをしています。Twitterでは、日本語のアカウントも運用していますので、ぜひフォローをお願いします。