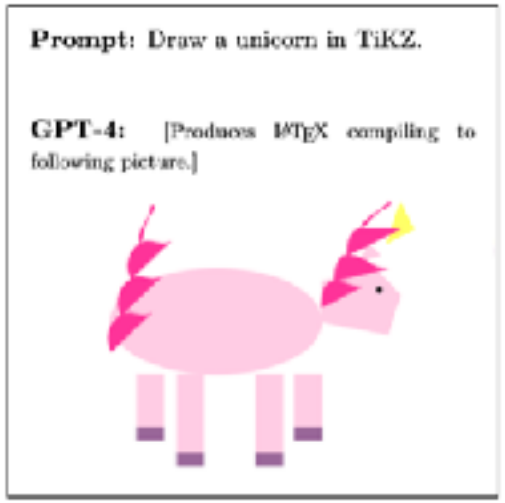
日本語に強い大規模言語モデルSwallow

学術国際情報センター
横田理央
rioyokota@gsic.titech.ac.jp

2024/05/18 SEAGAIA MEETING 2024 IN 宮崎

最新の生成AIの性能

ChatGPT (OpenAI)



出典：Sparks of Artificial General Intelligence: Early experiments with GPT-4

LaTeXでユニコーン(一角獣)が描けるということは？

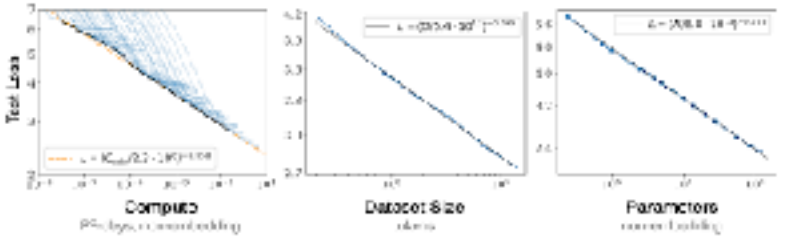
- この時点でGPT-4はマルチモーダルではなかった
- 文字情報からユニコーンが何かを理解
- ユニコーンの図形情報を再構築
- TiKZの文法を正確に習得

Gemini (Google)

出典：Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context

ChatGPT公開以前の流れ

2019/06/22 MicrosoftがOpenAIに10億ドル(当時 約1100億円)を投資
2020/01/23 OpenAIが言語生成モデルに関するScaling Lawの論文を発表
2020/03/28 OpenAIが言語生成モデルGPT3に関する論文を発表
2020/06/11 OpenAIがGPT3のAPIを公開
2020/09/22 OpenAIがMicrosoftに対してGPT3のソースコードを公開
2021/01/05 OpenAIが画像+言語モデルCLIPと画像生成モデルDALL-Eを発表
2021/06/29 GithubがGPT3にコードを追加学習したCodexを利用したCopilotを発表
2021/07/28 Free Software FoundationがGithub Copilotに対するライセンス上の懸念を表明
2021/09/10 Naverが言語生成モデルHyperClovaを発表
2022/02/22 DeepMindがコード生成モデルAlphaCodeを発表
2022/03/29 DeepMindが言語生成モデルChinchillaを発表
2022/05/23 Googleが画像生成モデルImagenを発表
2022/07/20 OpenAIが画像生成モデルDALL-E2を発表
2022/07/22 BigScience (HuggingFace,CNRS,GENCI)が多言語モデルBloomを発表
2022/08/04 精華大学が言語生成モデルGLM-130Bを発表
2022/08/22 Stability AIが画像生成モデルStable Diffusionを発表
2022/11/10 DeepMindが汎用モデルGatoを発表
2022/11/15 DeepMindが画像+言語モデルFlamingoを発表

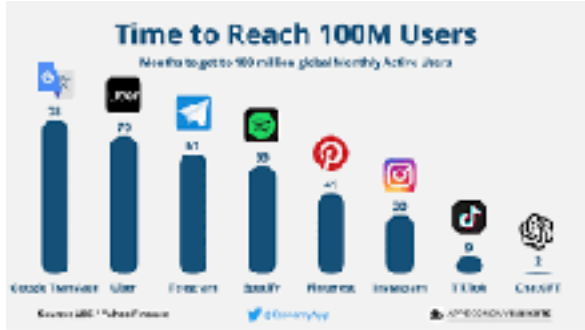


ChatGPT公開以降の時系列

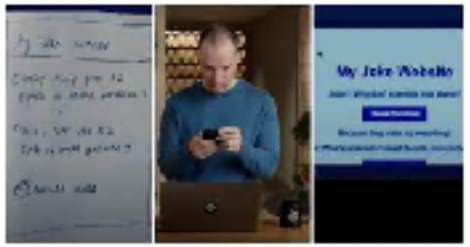
2022/11/30 OpenAIがChatGPTを発表
2022/12/04 公開後5日で100万ユーザを突破
2022/12/05 Stack OverflowがChatGPTで生成された投稿を禁止
2022/12/21 GoogleがChatGPTの自社への脅威に対してCode Red(非常事態)を宣言
2023/01/23 MicrosoftがOpenAIに100億ドル(当時 約1.3兆円)を投資 →


Microsoft to Invest $10 Billion in OpenAI, the Creator of ChatGPT
The tech giant aims to remain at the forefront of generative artificial intelligence with its partnership with OpenAI.

2023/02/01 OpenAIが月額$20の有料サービスChatGPT Plusを開始
2023/02/02 公開後約２ヶ月で1億ユーザを突破 →



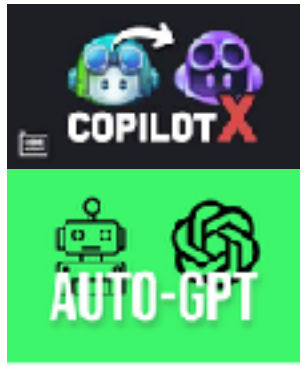
2023/02/03 自民党AIの進化と実装に関するプロジェクトチーム (第１回)
2023/02/06 GoogleがChatGPTに対抗して対話型AIのBardを限定公開
2023/02/07 Microsoftが検索エンジンBingにChatGPTを導入
2023/02/09 ChatGPTが米国の医師国家試験に合格できるレベルとの論文が発表される
2023/02/24 Metaが言語生成モデルLLaMAのソースコードを非商用ライセンスで公開
2023/03/01 OpenAIがChatGPTとWhisper(音声認識)のAPIを公開 (ユーザデータは学習に使わないことを宣言)
2023/03/09 MicrosoftがAzure OpenAI ServiceでChatGPTを提供開始
2023/03/10 Googleが言語生成モデルPaLM2を発表
2023/03/14 ChatGPT PlusでGPT-4が利用可能に →



2023/03/14 GoogleがPaLM2のAPIを公開

ChatGPT公開以降の時系列

2023/03/16 Microsoft 365 CopilotでWord,Outlook,TeamsなどからAIアシスタントが利用可能に
2023/03/20 ChatGPTから氏名,email,住所,カード番号などの個人情報が漏えい
2023/03/20 GitHubがGPT-4を搭載したCopilot Xを公開 →
2023/03/30 ChatGPTのAPIを利用したAI agentであるAuto-GPTが公開 →
2023/03/30 UC Berkeley,CMU,StanfordなどがGPT-4の会話データで微調整したVicunaを発表
2023/03/31 イタリアがChatGPTの利用を禁止 (4/28に撤回)
2023/05/19 G7広島サミットにて「広島AIプロセス」立ち上げ
2023/05/21 LLM勉強会 (第１回)
2023/06/01 「富岳」政策対応枠 利用開始
2023/06/05 TTI (Abu Dhabi)がFalcon-40Bのソースコードを商用ライセンスで公開 (09/06に180Bも)
2023/07/11 Anthropicが言語生成モデルClaude2を発表
2023/07/18 Metaが言語生成モデルLLaMA2のソースコードを限定的な商用ライセンスで公開 →
2023/09/07 Turingが画像+言語モデルHeronを公開
2023/09/25 AlibabaがQwen-14Bのソースコードを限定的な商用ライセンスで公開
2023/10/02 AmazonがAnthropicに12億ドル(約1800億円)を投資
2023/10/03 産総研の生成AI開発支援プログラムに国立情報学研究所(NII)とELYZA社が採択
2023/10/26 Google, Microsoft, Anthropic, Open AI が AI Safety Fundに10億ドル(1500億円)を投資
2023/10/30 米国がAI規制に関する大統領令を発令



ChatGPT公開以降の時系列

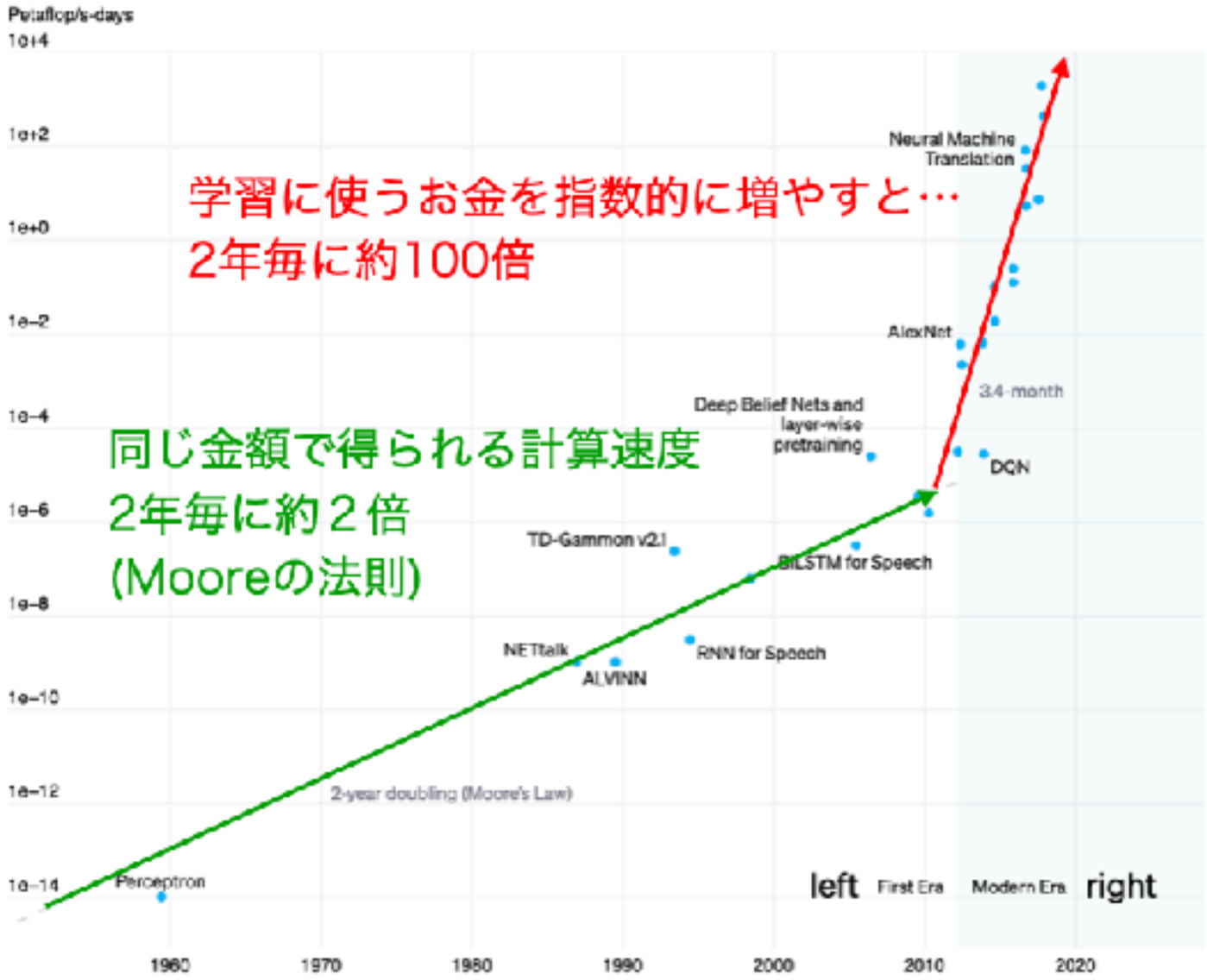
2023/11/17 Sam AltmanがOpenAIから解任→Microsoftに移る→OpenAIに戻る
2023/11/30 AlibabaがQwen-72Bのソースコードを限定的な商用ライセンスで公開
2023/12/06 AI Alliance立ち上げ：日本からは東大,慶應,SB Institute, sakana.ai, Sonyなどが参加
2023/12/07 GoogleがGemini, Alphacode2を発表
2023/12/08 Elon Muskが立ち上げたAIベンチャーXがGrokを発表
2024/12/19 東工大・産総研が Swallow-7B,13B,70Bを限定的な商用ライセンスで公開
2023/12/21 RinnaがNekomataのソースコードを限定的な商用ライセンスで公開
2024/01/02 Mistral AIがMixtral 8x7Bのソースコードを商用ライセンスで公開
2024/02/15 GoogleがGemini Pro 1.5を発表
2024/02/16 OpenAIが動画生成モデルSoraを発表
2024/02/21 GoogleがGemmaのソースコードを商用ライセンスで公開
2024/02/26 Mistral AIがMistral Largeを発表
2024/03/04 AnthropicがClaude3を発表
2024/03/10 Sam AltmanがOpenAI CEO復帰
2024/03/11 東工大・産総研が Swallow-MS, Swallow-MXを商用ライセンスで公開
2024/03/12 Elon MuskがGrokをオープンソースにするとX上で宣言
2024/03/12 ELYZAがELYZA-70Bのソースコードを商用ライセンスで公開
2024/04/18 Metaが言語生成モデルLLaMA3のソースコードを限定的な商用ライセンスで公開
2024/04/23 Microsoftが小型のPhi-3を発表

深層学習の大規模化

2012: **AlexNet**
$500のGPU x 2
= **$1,000**
を使って**5日**間学習

2022: **GPT-4** (予測値)
$8,000のGPU x 25,000
= **$200,000,000**
を使って**90日**間学習

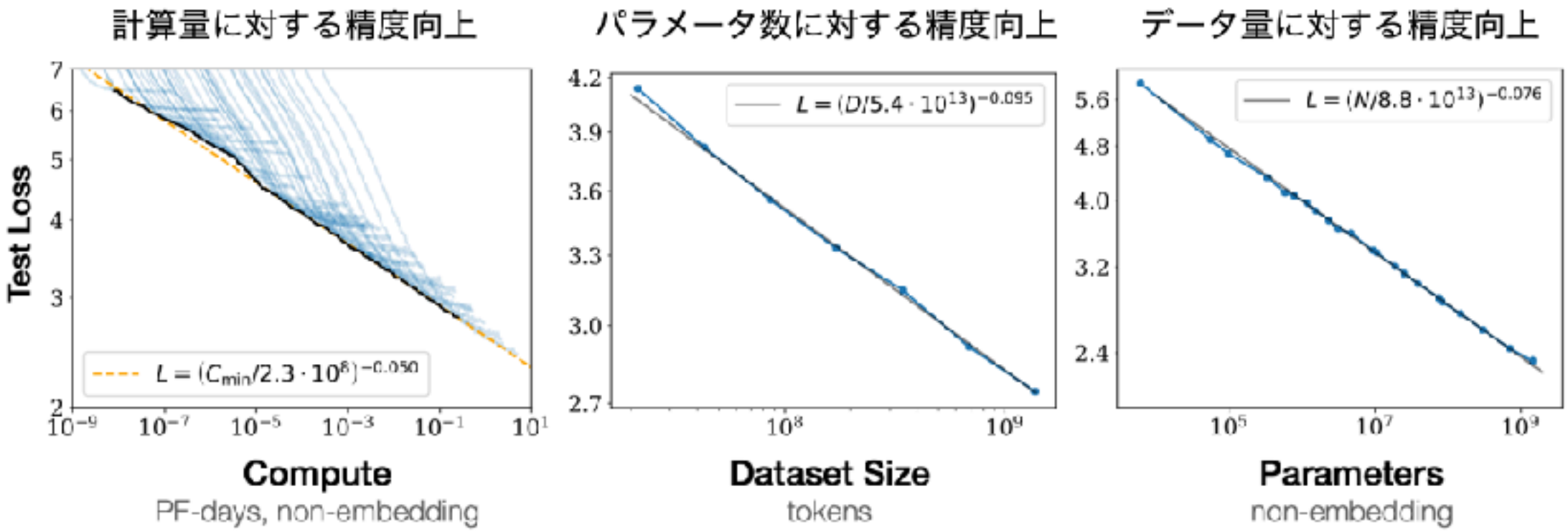
なぜこんなにお金をかける
ようになったのか？



出典：OpenAI: AI and compute

予測可能な費用対効果(スケール則)

・計算量 ∝ パラメータ数 x データ量
・計算量 ∝ GPU数 x 計算時間
・例：TSUBAME4.0だと 1 GPU x 1 時間 = 15.625円 (学内限定)
・TSUBAME4.0でGPT-4を学習するには2.8億円
・データやモデルの質を改善をすればもっと安くなる
・工夫をしなくても最低でもこの費用対効果の下限は保証される



出典：Scaling Laws for Neural Language Models

大規模言語モデルSwallow

Metaが開発した英語のLLMであるLlama2
に日本語を継続学習させた

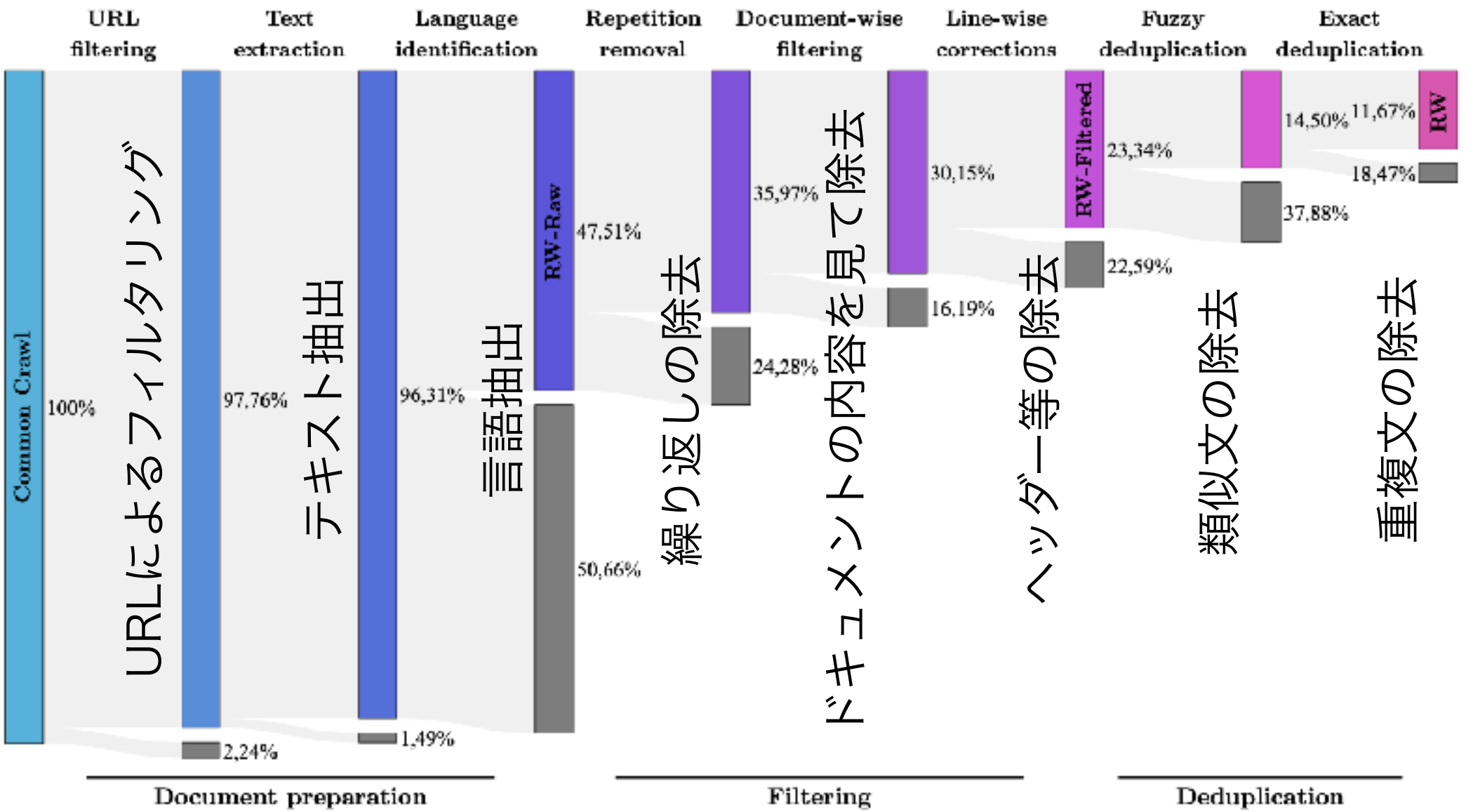
Metaが英語で学習した知識を使って
日本語で会話できるようになった

岡崎研：日本語データの収集と精製
・ネット上のあらゆるデータをクロールして
集積したCommonCrawlの2020年から
2023年にかけて収集された約634億ページ
のデータを色々な手法でフィルタリング

横田研：大規模分散学習
・何百GPUを同時に使って1ヶ月も学習を
する際の高速化・並列化・頑健化

データの前処理(岡崎研)

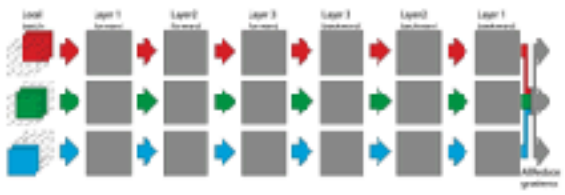
大きい単位から始めて、徐々に小さい単位へ



出典：The RefinedWeb Dataset for Falcon LLM

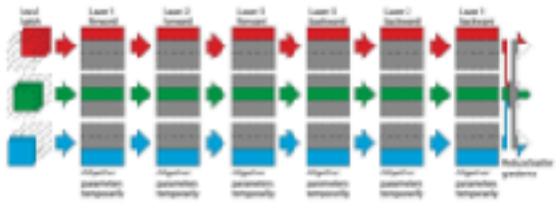
分散並列学習（横田研）

データ並列 (DP)



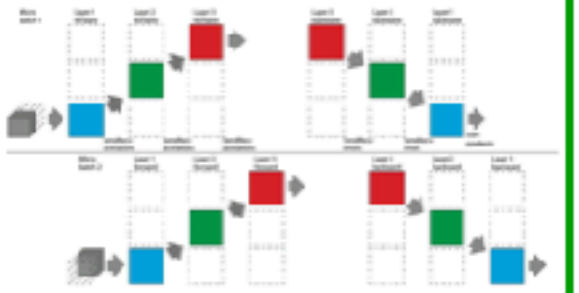
データ：分散
モデル：冗長
通信内容：勾配
通信形式：AllReduce
通信頻度：ステップ毎
長所：実装が簡単
短所：ラージバッチ問題
　　メモリ消費量

ZeRO (FSDP)



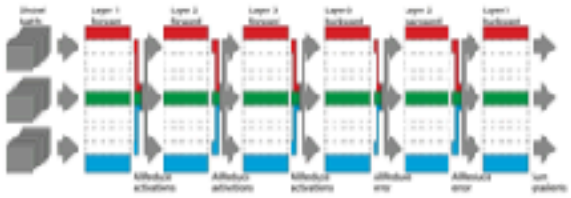
データ：分散
モデル：一時的に分散
通信内容：勾配＋重み
通信形式：ReduceScatter
　　＋AllGather
通信頻度：層毎
長所：実装が簡単
　　省メモリ
短所：ラージバッチ問題

パイプライン並列 (PP)



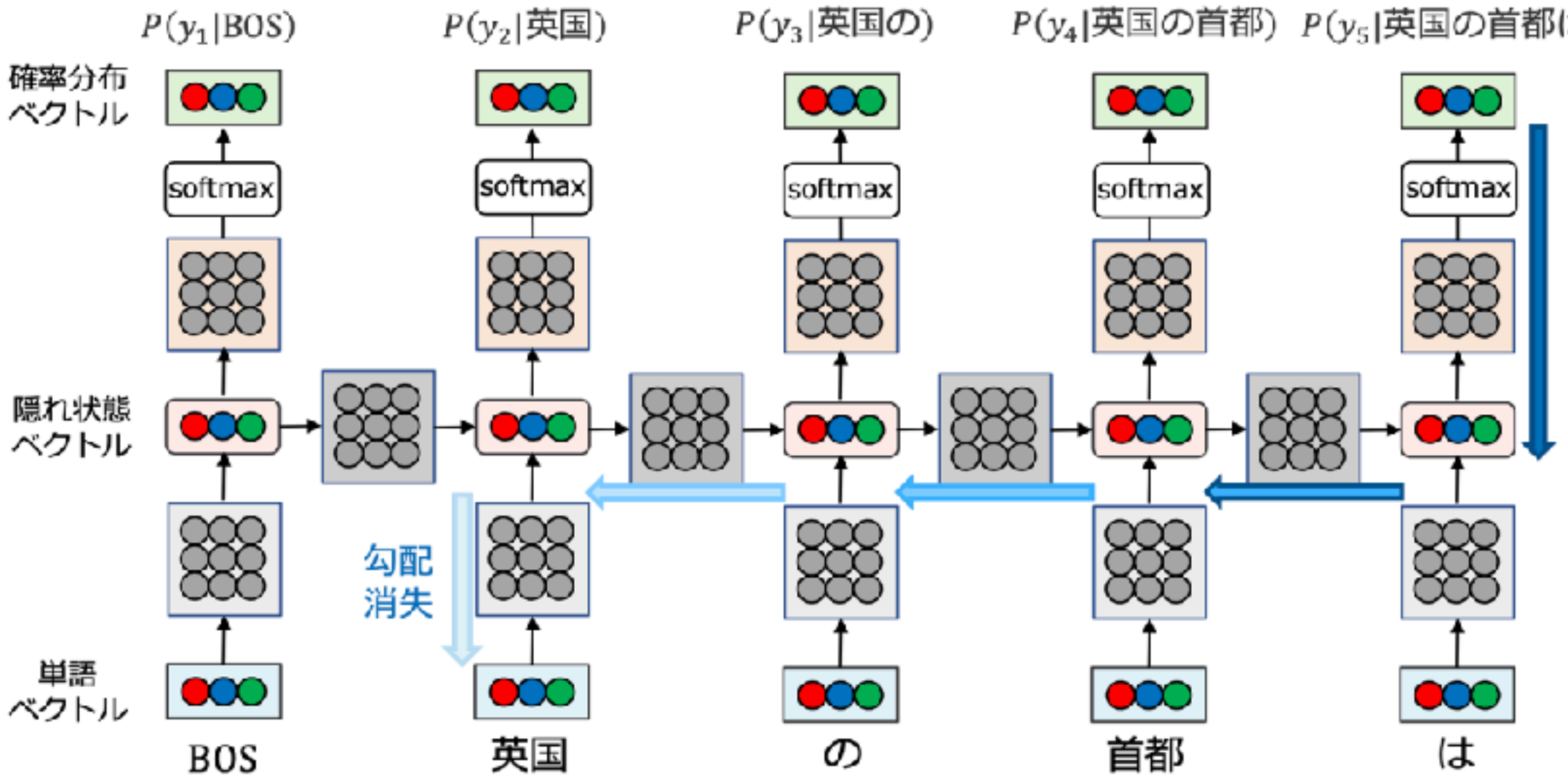
データ：冗長
モデル：分散
通信内容：活性
通信形式：SendRecv
通信頻度：層毎
長所：省メモリ
　　演算量低減
短所：パイプラインバブル

テンソル並列 (TP)



データ：冗長
モデル：分散
通信内容：活性
通信形式：AllReduce
通信頻度：層毎
長所：省メモリ
　　演算量低減
短所：通信オーバーヘッド
　　オーバーラップ不可
　　実装が複雑

言語モデル：Recurrent Neural Network (RNN)



T Mikolov, M Karafiát, L Burget, J Černocký, S Khudanpur. 2010. Recurrent Neural Network Based Language Model. In *INTERSPEECH*, pp. 1045-1048.

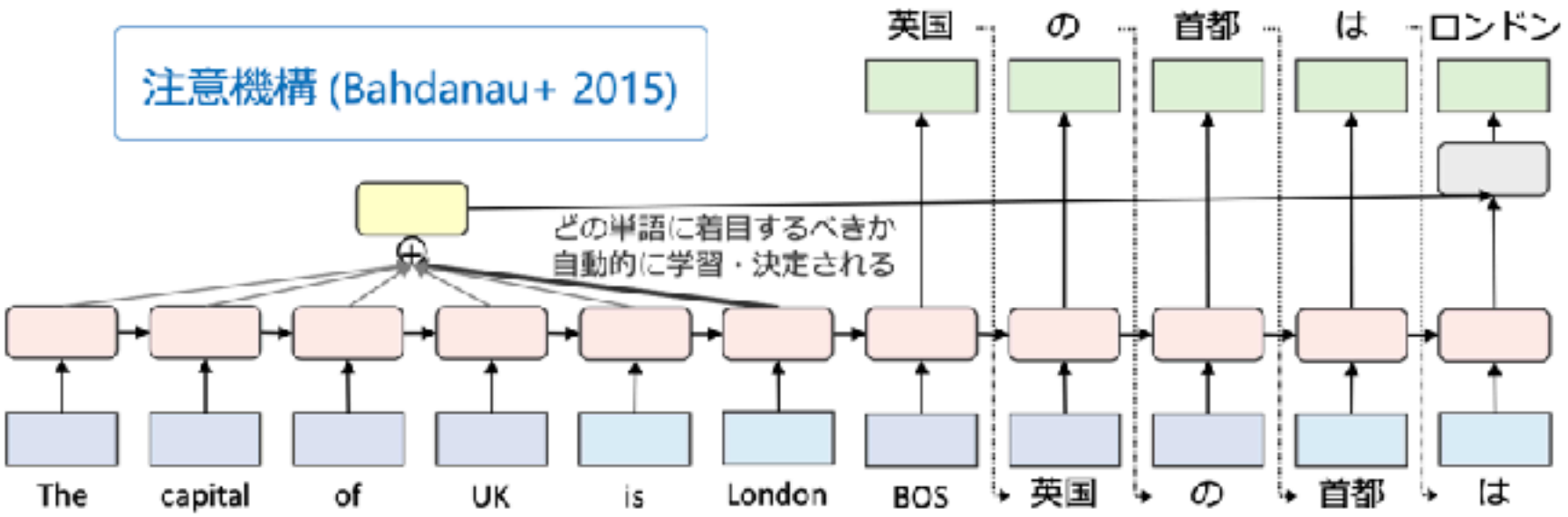
出典：岡崎先生：大規模言語モデルの驚異と脅威

機械翻訳モデル：注意機構

- 2014年頃から深層学習に基づく機械翻訳の研究が盛んに (Sutskever+ 2014)
- 機械翻訳モデルと言語モデルのアーキテクチャは似ている
- 大規模言語モデルの基盤となるアイディア（例：注意機構）が次々と生み出される

☑ 注意機構により、固定長のベクトルだけを用いるのではなく、入力単語の情報を柔軟に参照しながら翻訳単語の予測を行えるようになり、長い入力文の翻訳精度が向上した

☹ 入力文中の単語間、出力文中の単語間の長距離依存を考慮しにくい



I Sutskever, O Vinyals, Q V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *NIPS*, pp. 3104–3112.
D Bahdanau, K Cho, Y Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR*.

出典：岡崎先生：大規模言語モデルの驚異と脅威

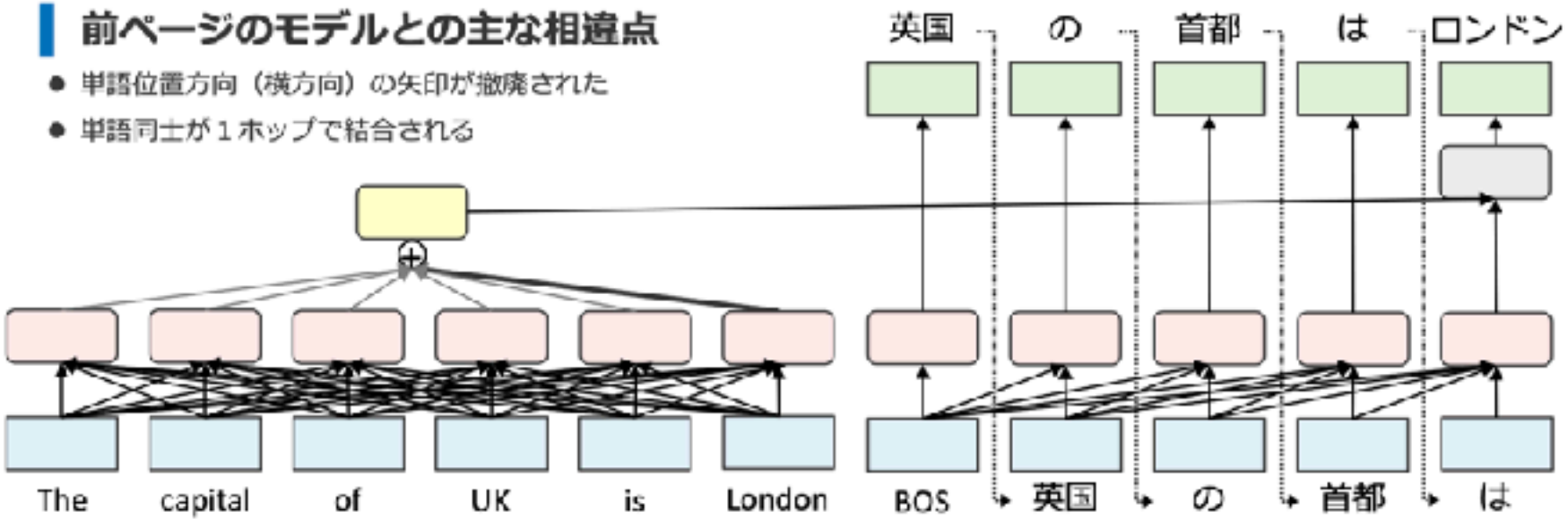
言語モデル：Transformer

- 自己注意だけで単語間の情報を統合するモデル
- 位置エンコーディング、マルチヘッド注意、残差結合、層正規化などの工夫を盛り込む

- ☑ 単語間の情報の統合に要するコストが距離に依らない（長距離依存を扱いやすい）
- ☑ 並列計算で実装しやすい（GPUやTPUなどのハードウェアを活用しやすい）
- ☑ 大規模言語モデルに限らず、自然言語処理以外の分野も含めて、汎用的に用いられる基盤アーキテクチャとなった

前ページのモデルとの主な相違点

- 単語位置方向（横方向）の矢印が撤廃された
- 単語同士が１ホップで結合される



A Vaswani, N Shazeer, N Parmar, J Uszkoreit, L Jones, A N. Gomez, L Kaiser, I Polosukhin. 2017. Attention is All You Need. In *NIPS*, pp. 5998–6008.

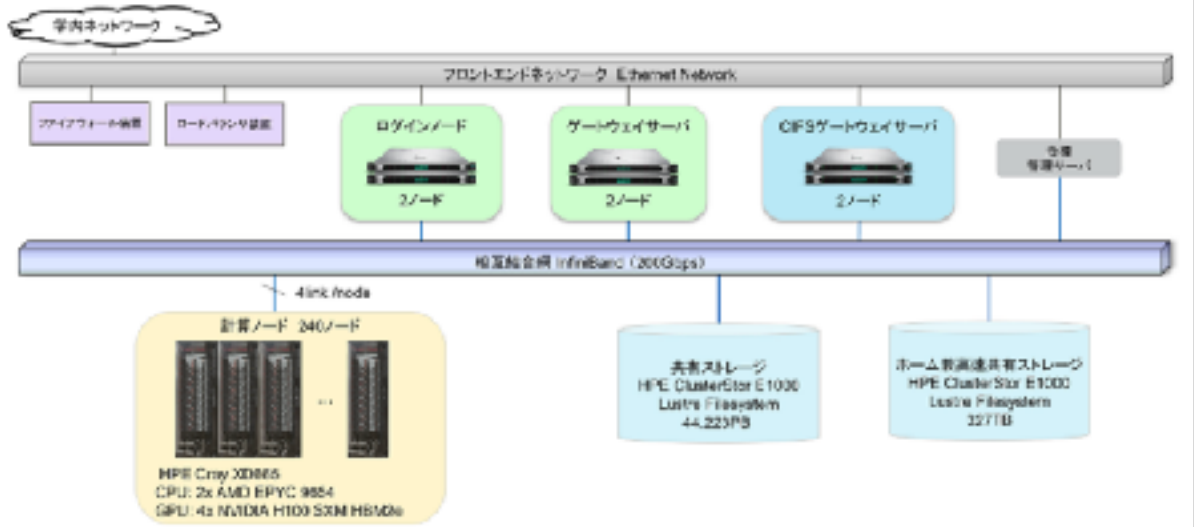
出典：岡崎先生：大規模言語モデルの驚異と脅威

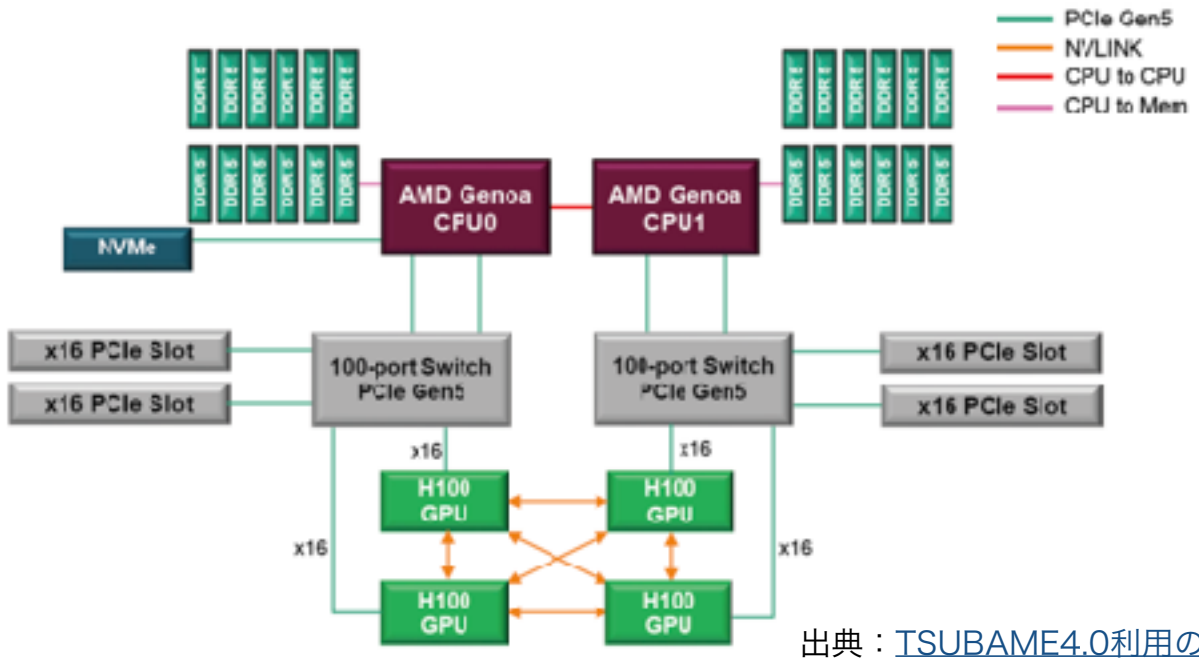
スーパーコンピュータの構造

構成要素	製品・性能
CPU	AMD EPYC 9654 2.4GHz × 2 Socket
コア数/スレッド数	96コア / 192スレッド × 2 Socket
メモリ	768GiB (DDR5-4800)
GPU	NVIDIA H100 SXM5 94GB HBM2e × 4
SSD	1.92TB NVMe U.2 SSD
インターコネクト	InfiniBand NDR200 200Gbps × 4

普通にPythonを使っているとGPUを
1個しか同時に使えない

分散並列処理をする場合はGPU間の
通信を気にする必要がある



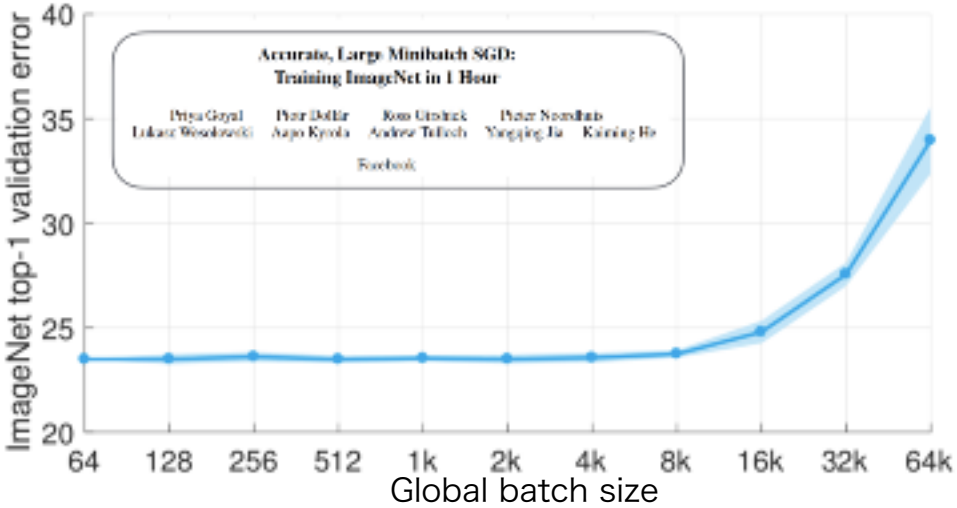


出典：TSUBAME4.0利用の手引き

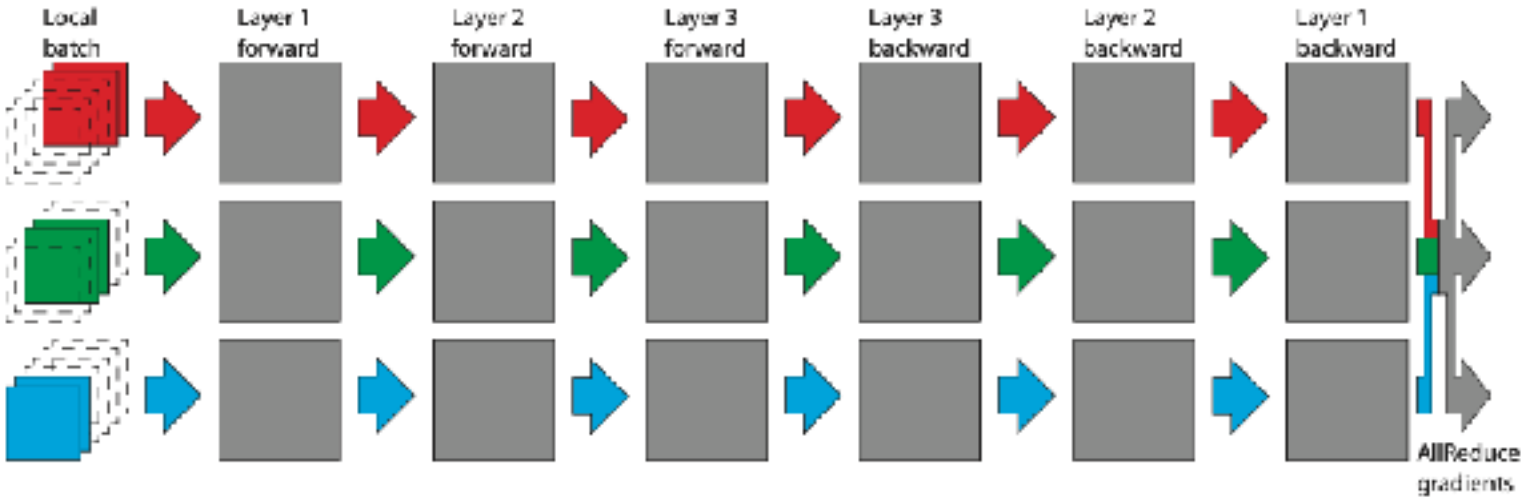
データ並列

Local batch：各プロセスのバッチ → 小さいとFLOP/sが出ない → メモリ上限まで増大
Global batch：全プロセスのバッチ → プロセス数に比例

Global batch sizeが増大すると汎化性能が低下
→ 解決策
・途中からbatch sizeを上げる
・特殊な最適化手法を使う (LARS,LAMB)
・特殊な正則化項を加える (勾配分散)



Accurate, Large Minibatch SGD:Training ImageNet in 1 Hour
https://arxiv.org/abs/1706.02677



ZeRO (Zero Redundancy Optimizer)

データ並列におけるモデルの冗長性を緩和 → 消費メモリはプロセス数に反比例

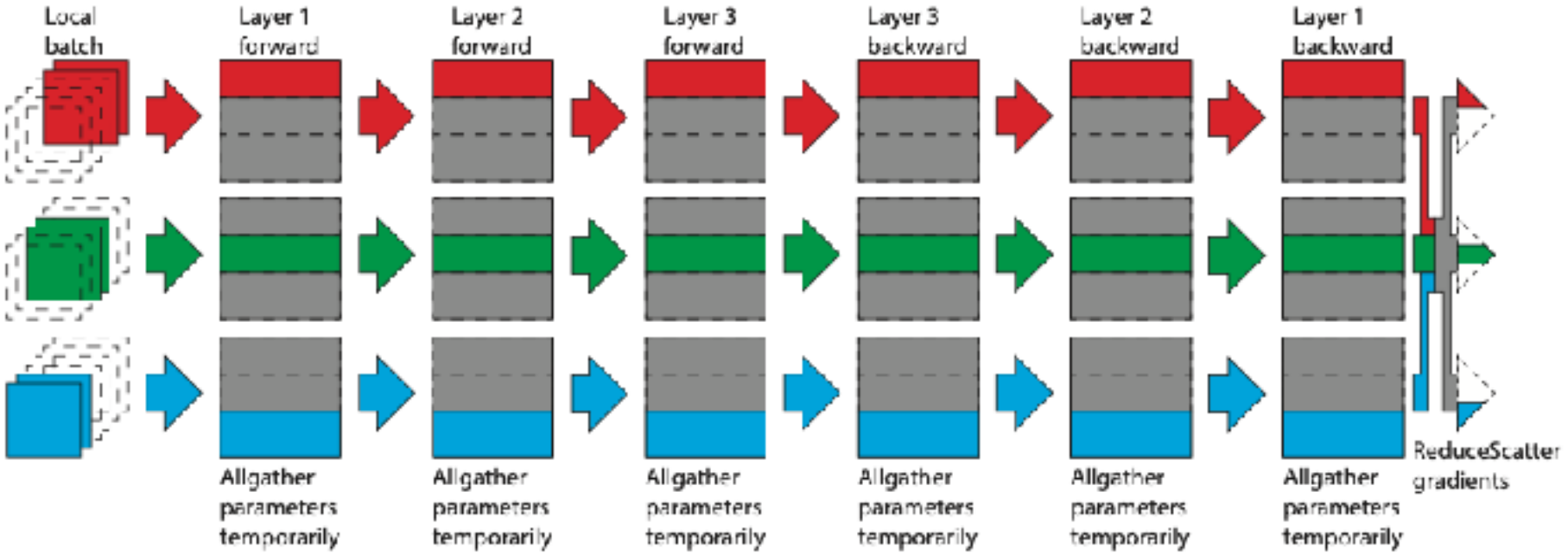
ZeRO-1：Optimizer stateを分散

ZeRO-2：Optimizer state＋勾配を分散

ZeRO-3：Optimizer state＋勾配＋重みを分散（重みをAllGather）

演算量はデータ並列と同じ

バッチサイズが増大する問題もデータ並列と同じ



パイプライン並列（層間並列）

Local batchをさらに細かくmicro batchに分けてパイプライン処理する
隣の層を担当するプロセスとの近接通信
自分が担当する層のみを計算する → メモリも演算もプロセス数に反比例
パイプラインバブルを低減するために様々な手法が提案されている

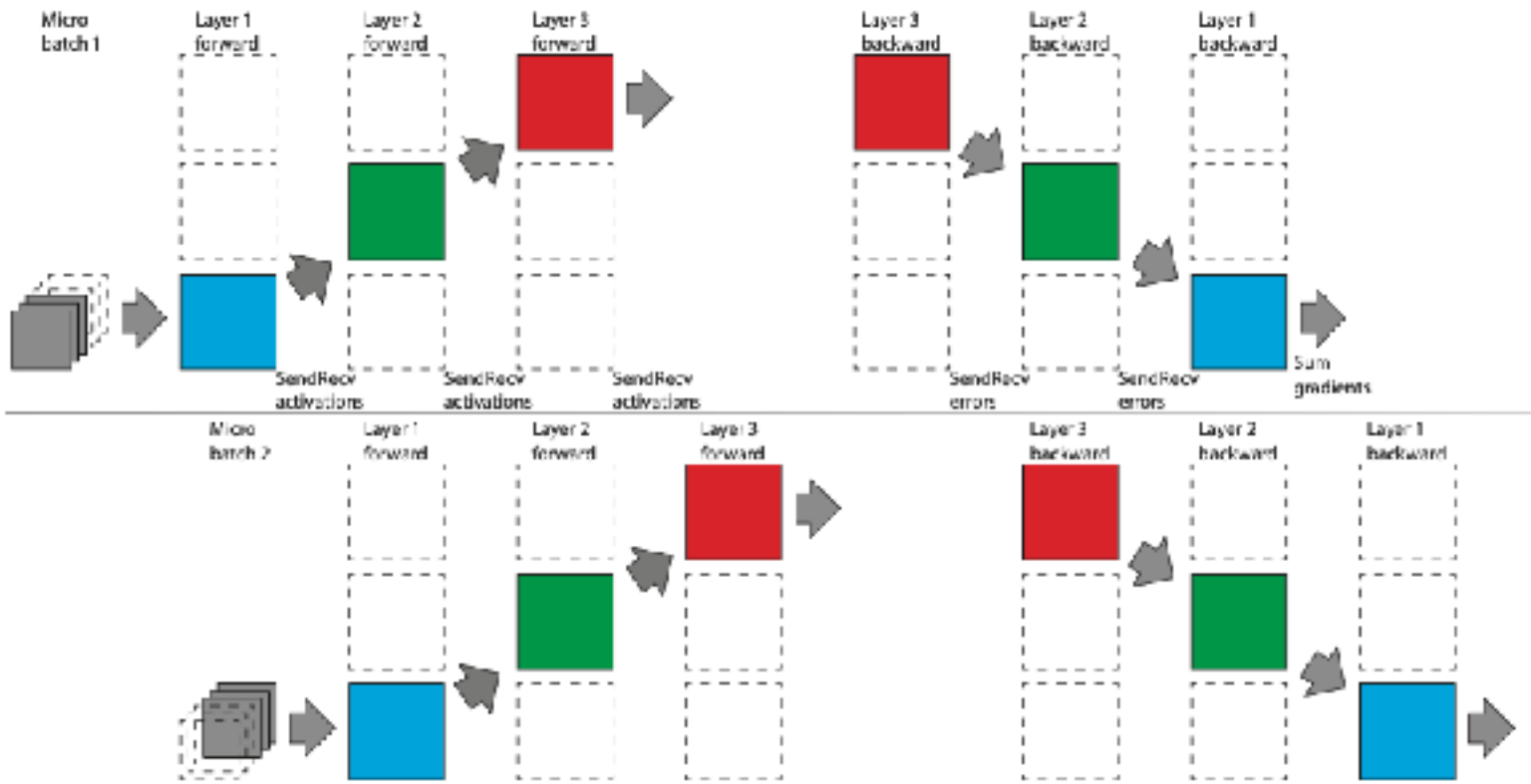
・GPipe：micro batch
[https://arxiv.org/abs/1811.06965]

・PipeDream：非同期
[https://arxiv.org/abs/1806.03377]

・Interleaved 1F1B：層数を
プロセス数x2に分割
[https://arxiv.org/abs/2104.04473]

・Chimera：双方向パイプライン
[https://arxiv.org/abs/2107.06925]

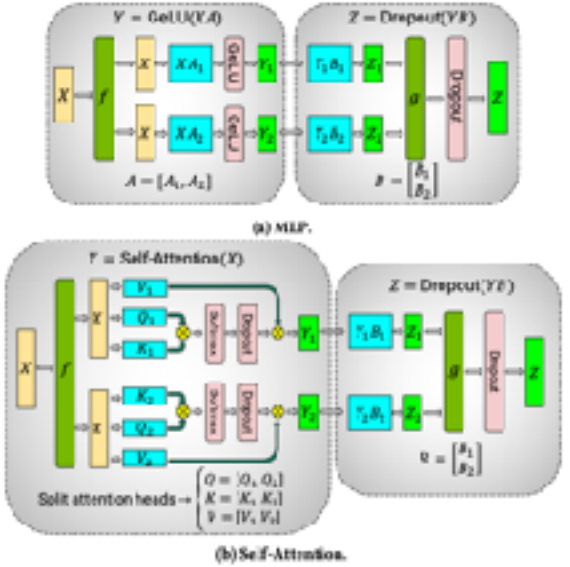
・ZeroBubble：Backward
のW微分とx微分を分ける
[https://arxiv.org/abs/2401.10241]



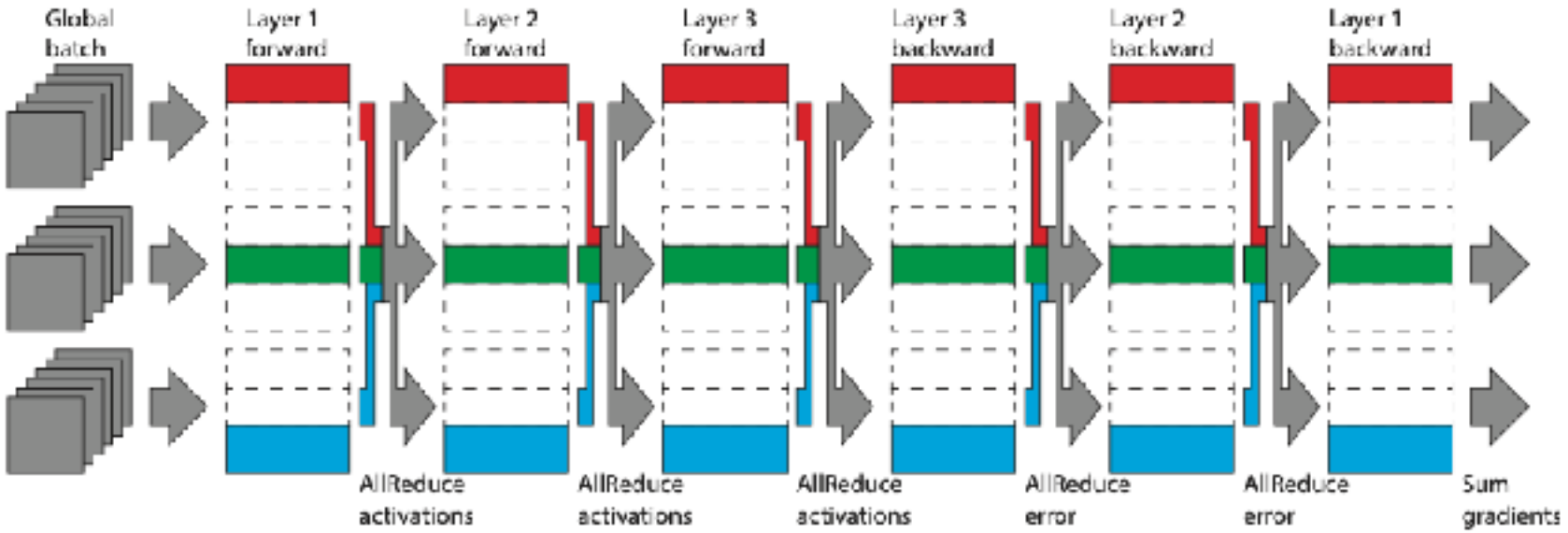
テンソル並列（層内並列）

層内の行列積の分散並列処理 → メモリも演算もプロセス数に反比例
→ 実装がモデルアーキテクチャに依存するため自動並列化が困難 →

次の層の計算をするには必ず活性の通信が必要
通信をするには前の層の計算が完了していることが必要
→つまり、計算と通信をオーバーラップする余地が全くない

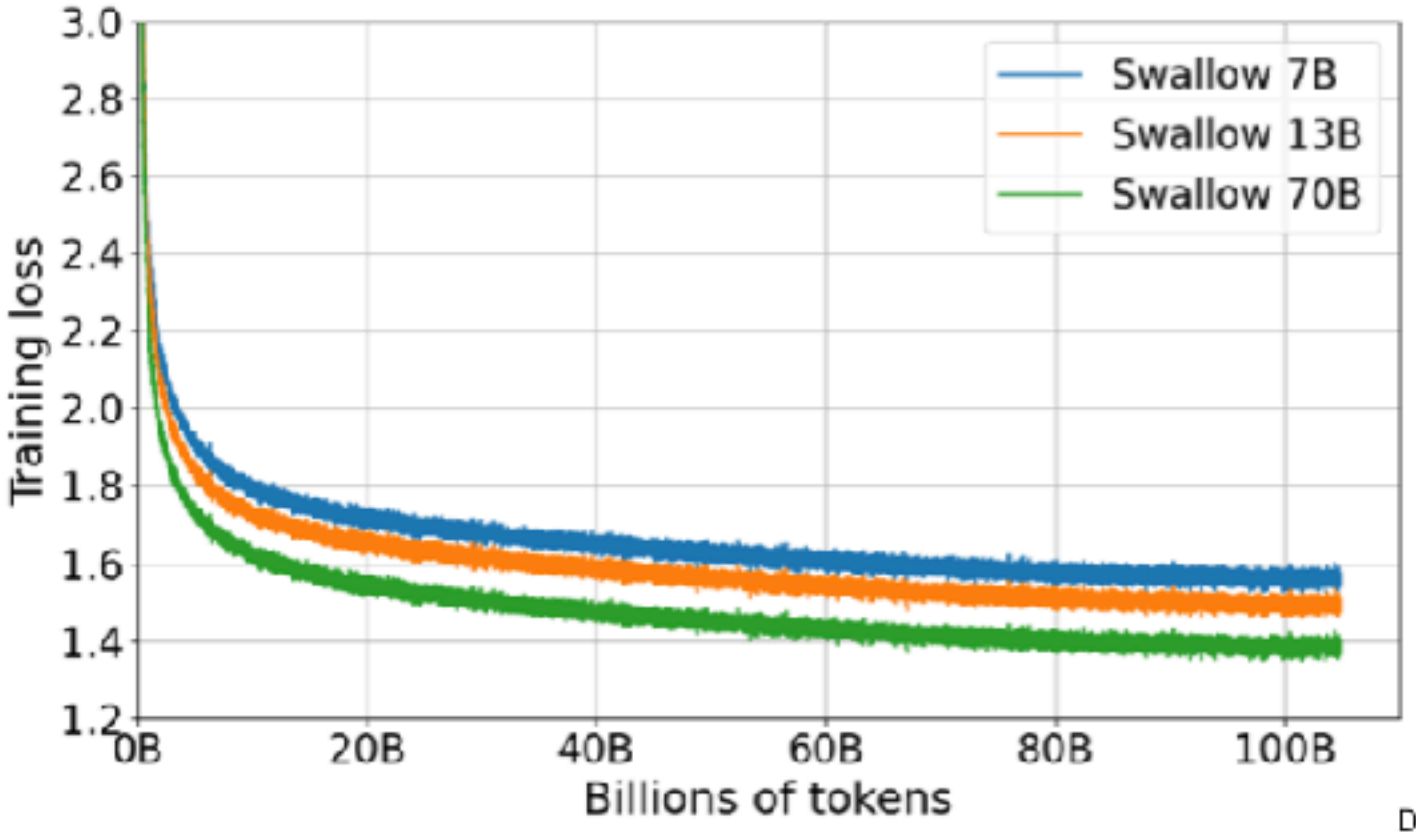


Efficient Large-Scale Language Model Training on GPU Clusters Using Megatron-LM
https://arxiv.org/abs/2104.04473



Swallowの学習

Parameters	Embedding size	Att. heads	Layers	Context	GQA	Tokens	Batch size	LR
7B	4096	32	32	4096	No	100B	1024	1.0×10^{-4}
13B	5120	40	40	4096	No	100B	1024	1.0×10^{-4}
70B	8192	64	80	4096	Yes	100B	1024	5.0×10^{-5}



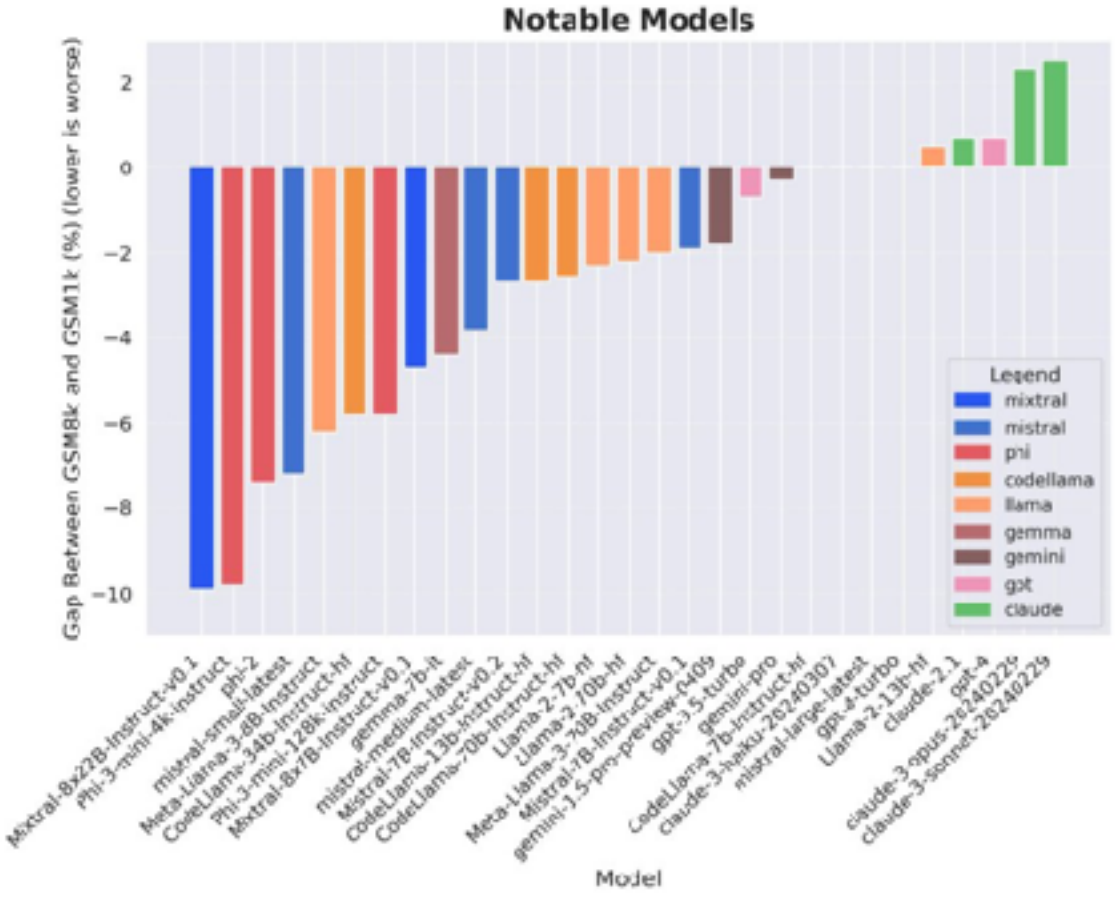
DALL-E generated image of swallows playing with a llama in Tokyo

日本語 LLM ベンチマークにおける性能

出典：Nejumi Neo Leaderboard

Swallowは現在国産LLMの中で最高性能

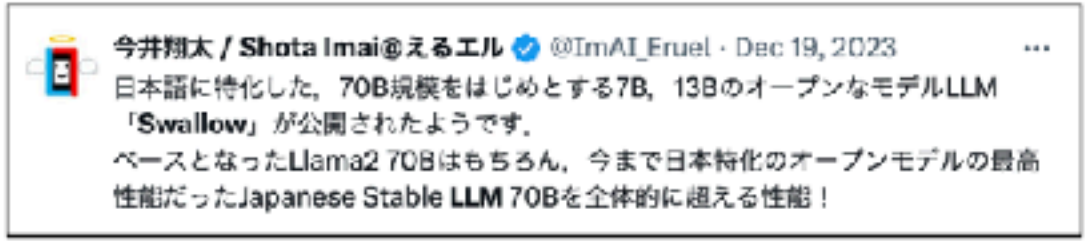
ただし、ベンチマークが全てではない



出典：A Careful Examination of Large Language Model Performance on Grade School Arithmetic

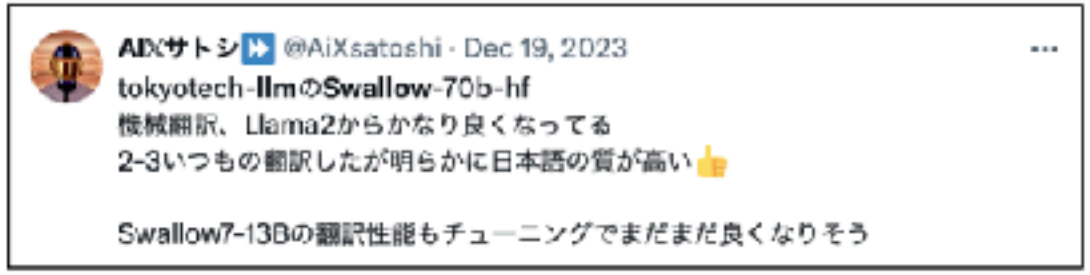
	run_name	AVG	AVG_jaster	AVG_MT_bench
20	gpt-4-0125-preview	0.7722	0.6463	8.981
9	gpt-4-turbo-2024-04-09	0.769	0.6343	9.038
2	anthropic.claude-3-opus-20240229-v1:0	0.7508	0.6178	8.837
48	gpt-3.5-turbo	0.6701	0.5161	8.241
6	Qwen/Qwen1.5-72B-Chat	0.6605	0.5016	8.194
16	mistral-large-2402	0.6549	0.5485	7.612
28	gemini-pro	0.6402	0.5636	7.169
4	CohereForAI/c4ai-command-r-plus	0.6338	0.5369	7.306
10	meta-llama/Meta-Llama-3-70B-Instruct	0.5293	0.3462	7.125
39	stabilityai/StableBeluga2	0.5283	0.4111	6.456
43	mistralai/Mixtral-8x7B-Instruct-v0.1	0.5006	0.2774	7.238
8	augmxnt/shisa-gamma-7b-v1	0.4911	0.5203	4.619
5	01-ai/Yi-34B-Chat	0.4839	0.3022	6.656
42	tokyotech-llm/Swallow-70b-instruct-hf	0.4712	0.5036	4.387
13	Rakuten/RakutenAI-7B-chat	0.393	0.2548	5.312
23	lightblue/qarasu-14B-chat-plus-	0.3846	0.1437	6.256
46	stabilityai/japanese-stablelm-instruct-	0.3732	0.2432	5.031
37	rinna/nekomata-14b-instruction	0.3644	0.4375	2.912
33	elyza/ELYZA-japanese-Llama-2-13b-	0.3278	0.1506	5.05
41	cyberagent/calm2-7b-chat	0.2716	0.1057	4.375
44	matsuo-lab/weblab-10b-instruction-sft	0.234	0.2718	1.963

Swallowユーザの声



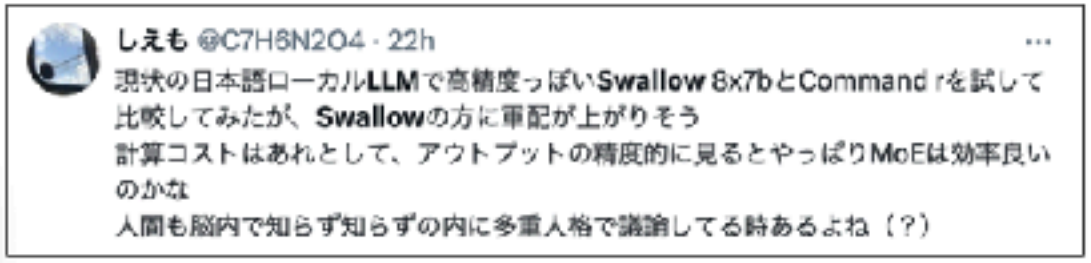
今井翔太 / Shota Imai@えるエル @ImAI_Eruel · Dec 19, 2023
日本語に特化した，70B規模をはじめとする7B，13Bのオープンなモデル**LLM**「**Swallow**」が公開されたようです．
ベースとなったLlama2 70Bはもちろん，今まで日本特化のオープンモデルの最高性能だったJapanese Stable **LLM** 70Bを全体的に超える性能！

しゅんけー @shunk031 · Dec 19, 2023
もはや @okoge_kaz さんの独壇場のように感じる…（最近見るLLM関連記事すべて彼が噛んでるように見える、凄すぎる）> **Swallow**: LLaMA-2 日本語継続事前学習モデル | Kazuki Fujii zenn.dev/tokyotech_lm/a... #zenn

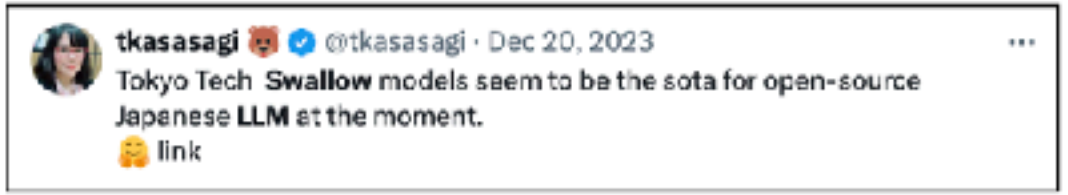


AIXサトシ @AiXsatoshi · Dec 19, 2023
tokyotech-llmの**Swallow**-70b-hf
機械翻訳、Llama2からかなり良くなってる
2-3いつもの翻訳したが明らかに日本語の質が高い👍

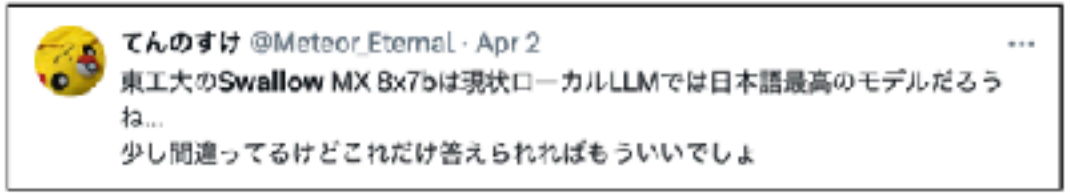
Swallow7-13Bの翻訳性能もチューニングでまだまだ良くなりそう



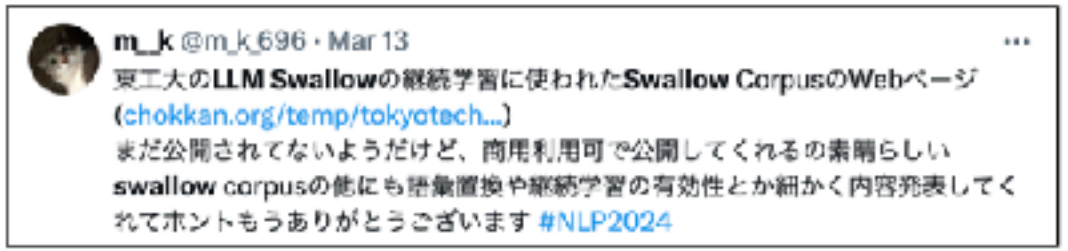
しえも @C7H8N2O4 · 22h
現状の日本語ローカル**LLM**で高精度っぽい**Swallow** 8x7bとCommand rを試して比較してみたが、**Swallow**の方に軍配が上がりそう
計算コストはあれとして、アウトプットの精度的に見るとやっぱりMoEは効率良いのかな
人間も脳内で知らず知らずの内に多重人格で議論してる時あるよね（？）



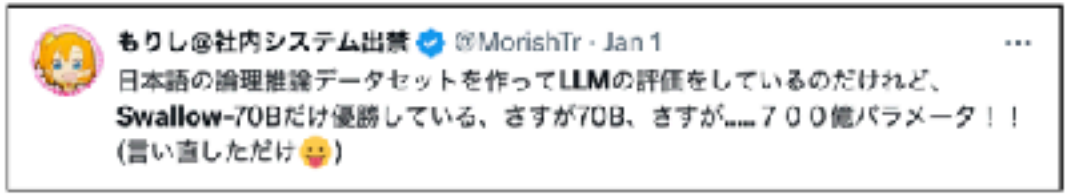
tkasasagi @tkasasagi · Dec 20, 2023
Tokyo Tech **Swallow** models seem to be the sota for open-source Japanese **LLM** at the moment.
🤗 link



てんのすけ @Meteor_Eternal · Apr 2
東工大の**Swallow** MX 8x7bは現状ローカルLLMでは日本語最高のモデルだろうね…
少し間違ってるけどこれだけ答えられればもういいでしょ



m_k @m_k_696 · Mar 13
東工大の**LLM Swallow**の継続学習に使われた**Swallow** CorpusのWebページ (chokkan.org/temp/tokyotech...)
まだ公開されてないようだけど、商用利用可で公開してくれるの素晴らしい
swallow corpusの他にも単語置換や継続学習の有効性とか細かく内容発表してくれてホントもうありがとうございます #NLP2024



もりし@社内システム出禁 @MorishTr · Jan 1
日本語の論理推論データセットを作って**LLM**の評価をしているのだけれど、**Swallow**-70Bだけ優勝している、さすが70B、さすが……７００億パラメータ！！
(言い直しただけ😛)

まとめ

大規模言語・画像モデルなどの生成AI技術は、2年に100倍の勢いで成長している

GPT-4は既に多くのタスクで平均的な人間を上回る（弁護士試験や医師国家試験にも合格できる）

最近の大規模言語モデルの事前学習は、世界最大級のスパコンを使っても何ヶ月もかかる

インターネットのデータは学習に役に立たないものがほとんど

いかに綺麗なデータを精製するかが重要（データが21世紀の原油だとするとその精製工場が重要）

何百GPUも同時に使いこなすには様々な並列化手法を組み合わせる必要がある

ベンチマークのスコアよりも使っているユーザの声が大事

分野の違う研究室がチームを組んで大きなプロジェクトに取り組むことの重要性

Tokyo Tech

Thank You