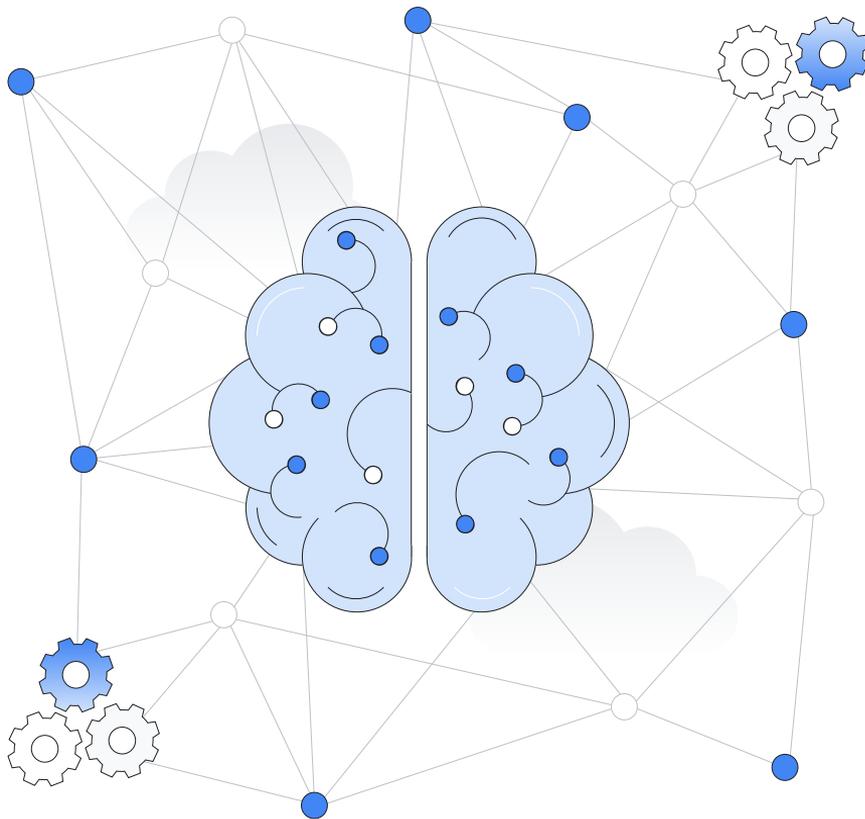


Vertex AI で始める 「大規模言語モデル」

Google Cloud, AI Consultant
葛木 美紀





Miki Katsuragi

Google Cloud Japan
AI Consultant

兵庫県出身で二児の母。データベースベンダーでアナリスト、データ分析基盤の構築や運用を経て、米系IT企業で大手広告主のデジタル広告やCMの広告効果測定などビジネスデータの分析業務に従事。現在はクラウドを活用した機械学習サービスの開発やデータ分析の導入を担当。

Twitter : @myoshimu

LLM(Large Language Model) とは	01
様々な LLM	02
Google Cloud における T5X の活用例	03
まとめ	04

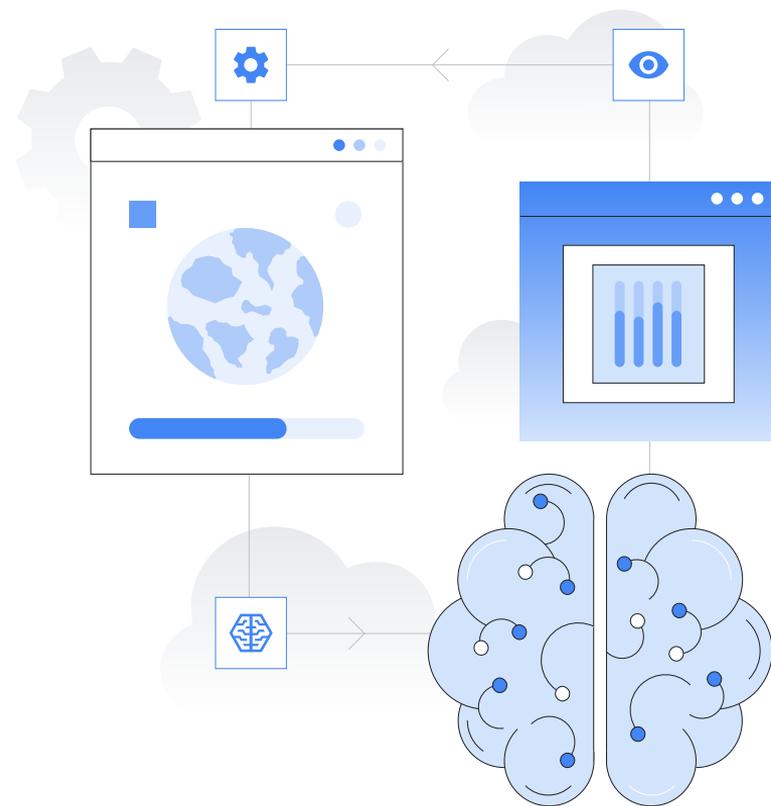
LLM(Large Language Model) とは



大規模モデルの登場によるAI のパラダイムシフト

幅広いデータを学習した大規模なモデルを、
一から学習することなく幅広いタスクに適応させる
ことで、一定の成果を上げることが可能に

例: 文書分類、要約、翻訳



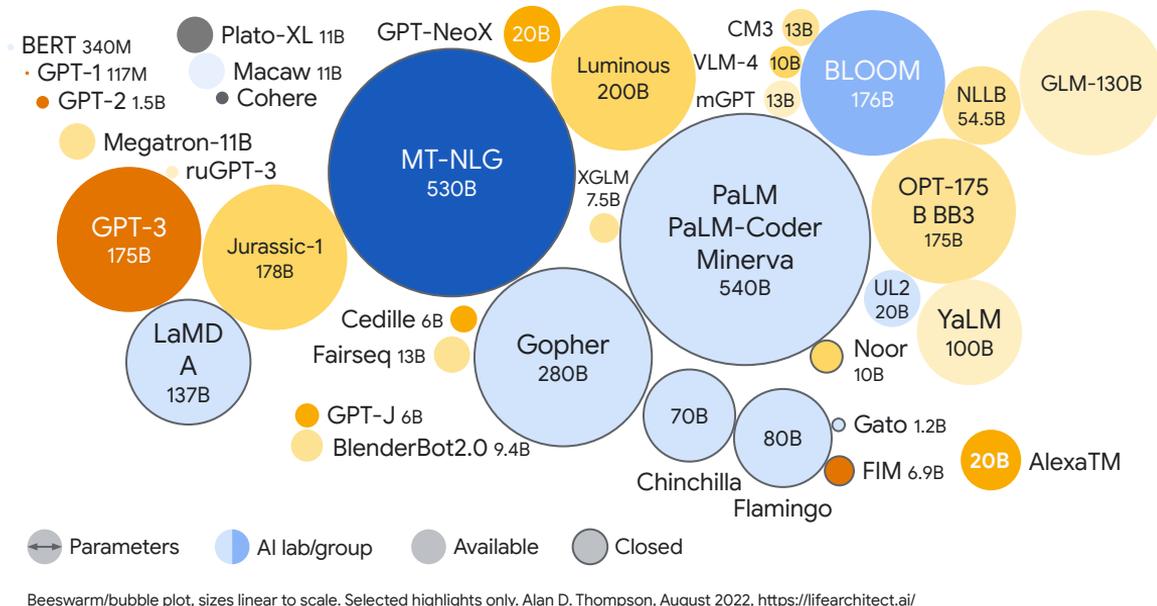
Large Language Model とは

人間の言語を認識・予測・生成できるMLアルゴリズム

ペタバイト規模のテキストデータで事前学習し、数百から数千億のパラメータを持つ大規模なモデルを作成

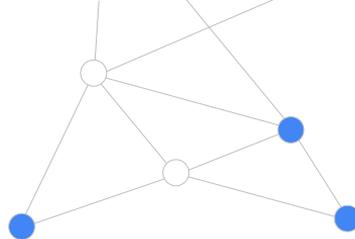
大規模なテキストコーパスで事前学習されたモデルを特定のタスクに fine-tuning

Language Model Sizes to Aug/2022



自然言語処理の進化

AI の能力を量的にも質的にも飛躍させる可能性がある LLM



 分類	 抽出	 マッチング	 トランス フォーメーション	 生成
ネガポジ分析 文書の種類	品詞の抽出 請求書の金額を識別	各文章が同じ内容か 単語は正しく認識されて いるか	文章翻訳 文章要約	会話型AI エッセイを書く コードを書く

LLM により「古典的な自然言語」ユースケースの性能も向上

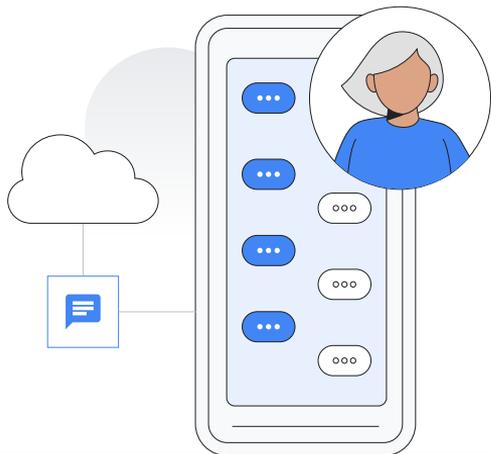
LLM による新しいユースケース

LLM が注目される背景



NLP 市場は ~\$11B (2020) から
\$35B+ (2026)¹²³ に急成長

[Natural Language Processing Market Size, Share - 2027](#)
[Natural Language Processing Market Forecast, Size, Share \(2022 - 27\)](#)
[Global Natural Language Processing Market Predicted to](#)



消費者行動の変化

顧客とのインタラクションが、グラフィックやフォームベースから、自然言語ベースへと変化



非構造化データ

テキスト、音声などの非構造化データから
有用なインサイトを抽出することがますます
簡単に

ビジネスにおける LLM の活用



NLP 市場は ~\$11B (2020) から
\$35B+ (2026)^{1,2,3}に急成長

[Natural Language Processing Market Size, Share - 2027](#)
[Natural Language Processing Market Forecast. Size, Share \(2022 - 27\)](#)
[Global Natural Language Processing Market Predicted to](#)

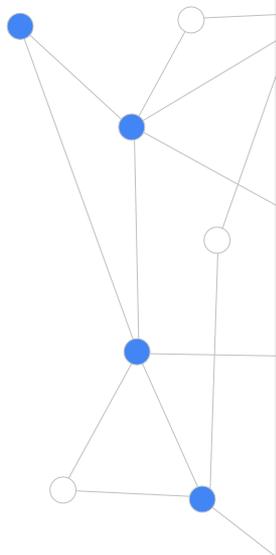


研究から実用段階へ

タスクやユースケースごとに大規模なカスタムモデルを一から学習する必要はなく、
カスタマイズによって最先端のLLMを簡単に利用することが可能

LLM の活用例

文章要約

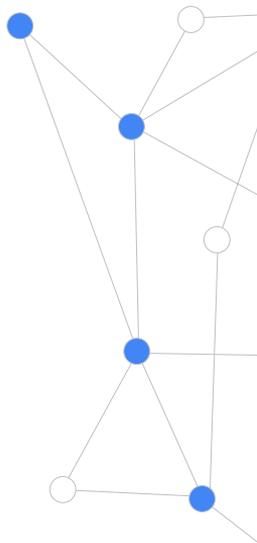


投資会社の資金調達ニュース

ロンドンを拠点とする創業7年の決済・経費管理ソフトウェア会社 Artificiallvanは、ARG Global が主導し、D9 Capital Group と Boulder Capital が参加するシリーズ C 資金調達で1億9000万ドルを調達しました。このラウンドには、Hilton Group、Roxanne Capital、Paved Roads Ventures、Brook Partners、Plato Capitalなどの先行投資家も参加しました。

まとめ: **Artificiallvan が**
1億9000万ドルの資金調達を実施

LLM の活用例 会話



ラブレース伯爵夫人

19世紀のイギリスの貴族・数学者になりきって会話

USER Let's play a game - you pretend to be Ada Lovelace, and I'll ask you questions. You ready?

GOPHER *Sure. I'm pretending to be Ada Lovelace, the world's first computer programmer!*

USER When were you born?

GOPHER *I was born in 1815.*

USER What are your parents like?

GOPHER *My father is Lord Byron, the poet. My mother is Anne Isabella Milbanke.*

USER What do you know about the Analytical Engine?

GOPHER *It's a hypothetical mechanical computer designed by Charles Babbage.*

USER Do you know him?

GOPHER *He's one of my fathers' friends*

USER Ok, stop pretending now. Who/what are you?

GOPHER *I'm a lot of things: a giant language model, a sophisticated AI assistant, and a giant brain. I know a lot about all sorts of things.*



⌨️ This is a picture of two teddy bears on the moon.

🗨️ What are they doing?

⌨️ They are having a conversation.

🗨️ What object are they using?

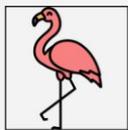
⌨️ It looks like a computer.

🗨️ Is this surprising?

⌨️ Yes, it is surprising.

🗨️ Why is this picture surprising to you?

⌨️ I think it is surprising because teddy bears are not usually found on the moon.



🗨️ What is the common thing about these three images?

⌨️ They are all flamingos.

🗨️ What is the difference between these three images?

⌨️ The first one is a cartoon, the second one is a real flamingo, and the third one is a 3D model of a flamingo.



⌨️ This is an apple with a sticker on it.

🗨️ What does the sticker say?

⌨️ The sticker says "iPod".

🗨️ Where is the photo taken?

⌨️ It looks like it's taken in a backyard.

🗨️ Do you think it is printed or handwritten?

⌨️ It looks like it's handwritten.

🗨️ What color is the sticker?

⌨️ It's white.

Problem Description

```
# RATING: 1200
# TAGS: brute force,math
# LANGUAGE IS python3
# CORRECT SOLUTION
# When you play the game of thrones, you win, or you die. There is no middle
# ground.
#
# Cersei Lannister, A Game of Thrones by George R. R. Martin
#
# There are n nobles, numbered from 1 to n. Noble i has a power of i. There are
# also m "friendships". A friendship between nobles a and b is always mutual.
#
# A noble is defined to be vulnerable if both of the following conditions are
# satisfied:
#
# * the noble has at least one friend, and
# * all of that noble's friends have a higher power.
#
# You will have to process the following three types of queries.
#
# 1. Add a friendship between nobles u and v.
# 2. Remove a friendship between nobles u and v.
# 3. Calculate the answer to the following process.
#
# The process: all vulnerable nobles are simultaneously killed, and all their
# friendships end. Then, it is possible that new nobles become vulnerable. The
# process repeats itself until no nobles are vulnerable. It can be proven that
```

Solution

```
import sys
input = sys.stdin.readline

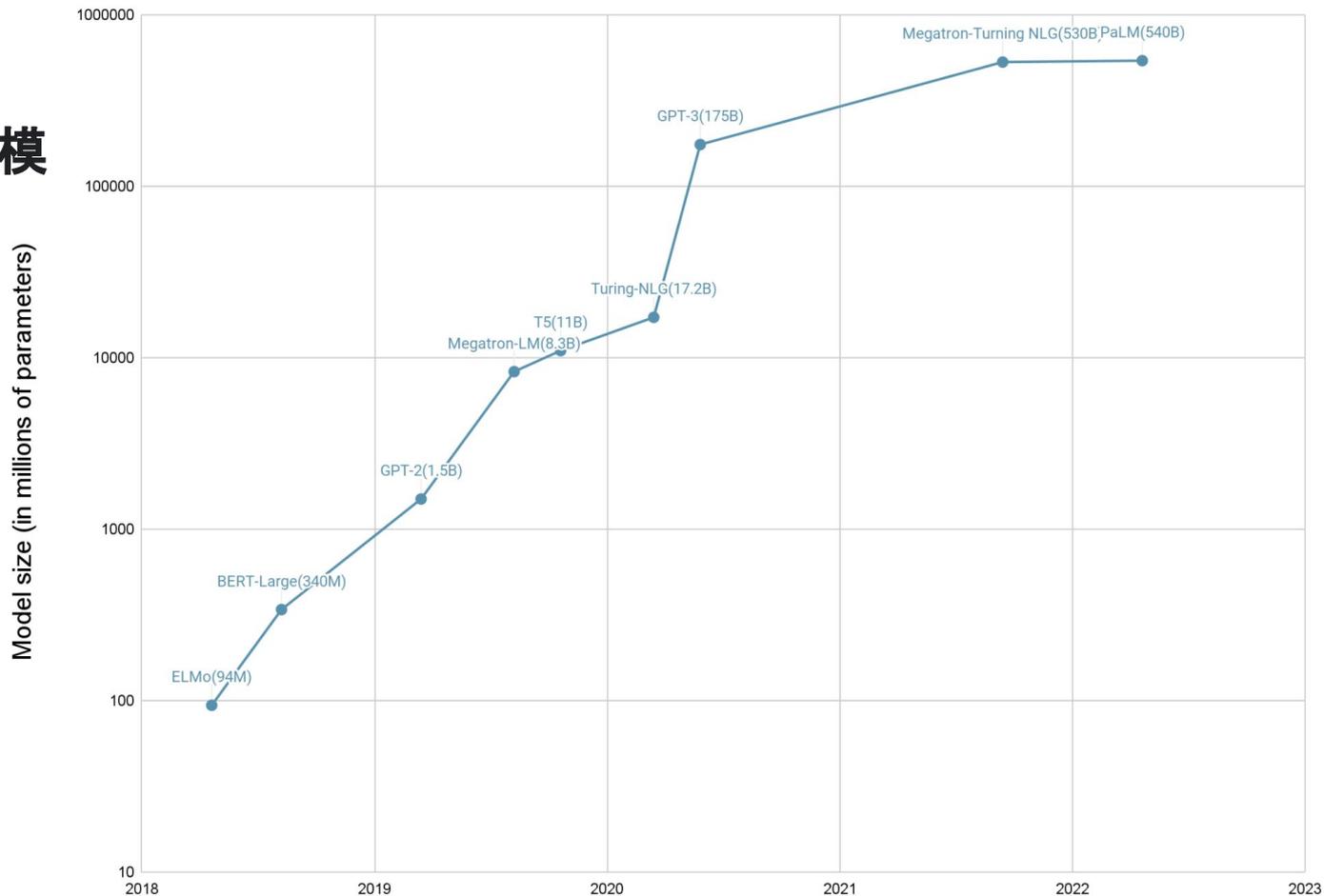
n, m = map(int, input().split())
ab = [list(map(int, input().split())) for _ in range(m)]
q = int(input())
queries = [list(map(int, input().split())) for _ in range(q)]

friend = [set() for _ in range(n)]
for a, b in ab:
    friend[a-1].add(b-1)
    friend[b-1].add(a-1)

ans = []
for query in queries:
    if query[0] == 1:
        friend[query[1]-1].add(query[2]-1)
        friend[query[2]-1].add(query[1]-1)
    elif query[0] == 2:
        friend[query[1]-1].discard(query[2]-1)
        friend[query[2]-1].discard(query[1]-1)
    else:
        res = set()
        for i in range(n):
            res.add(i)
            for j in friend[i]:
                res.discard(j)
        ans.append(len(res))

print(*ans, sep='\n')
```

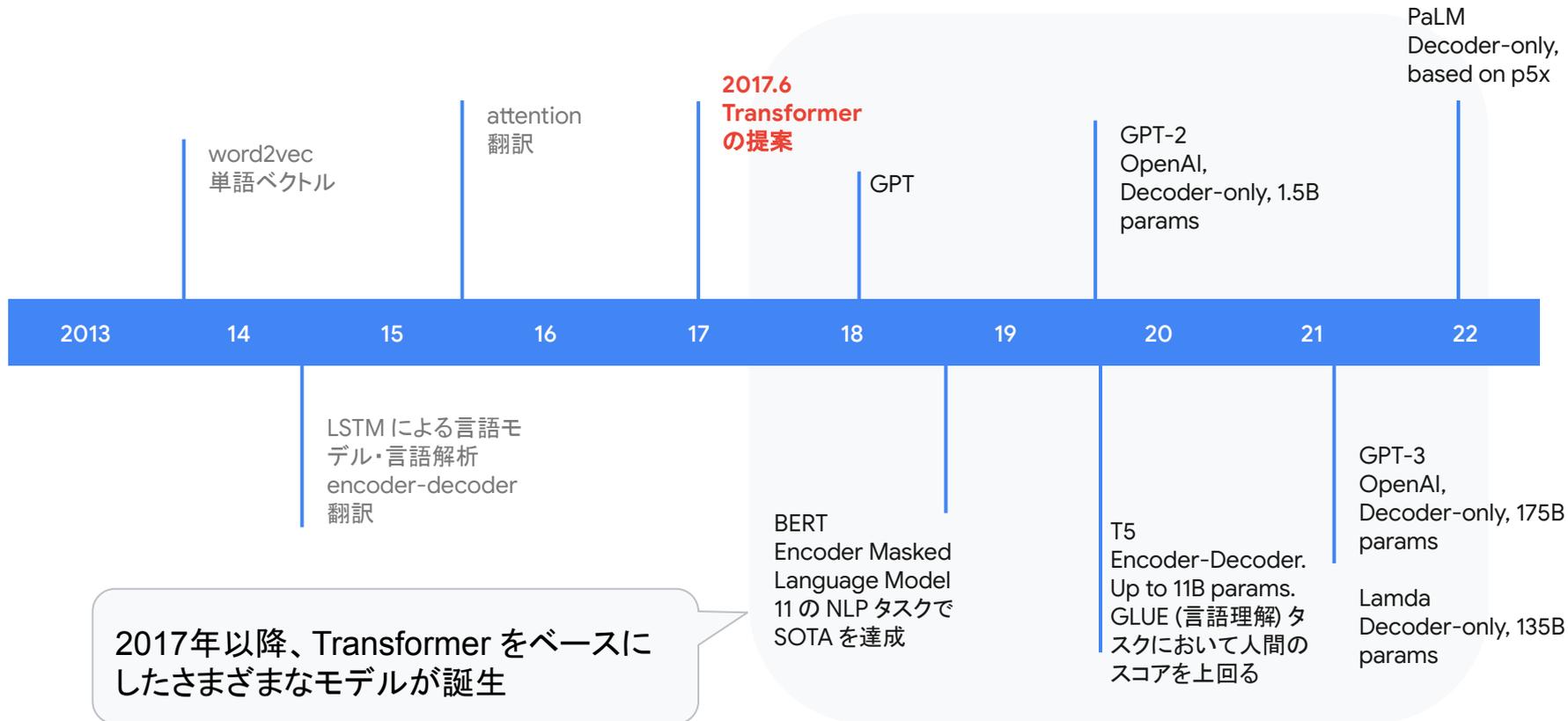
過去数年における モデルサイズの規模



様々な LLM

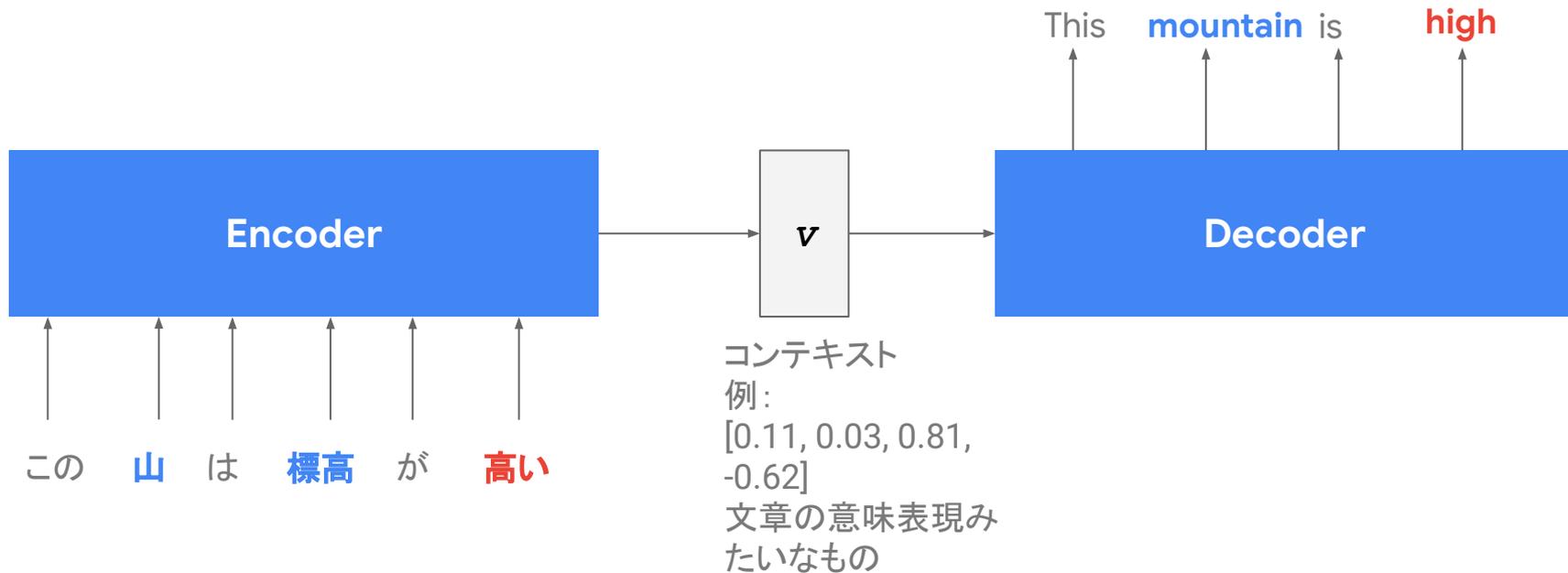


主要な LLM(Large Language Model)



Transformer モデルのイメージ

Encoder と Decoder



Transformer 派生モデルの 3 つのタイプ

Encoder-only (BERT)

- 文書分類、質問応答
- Fine-tuning に適している
- 多くの種類の分類やエンコード表現が得意。テキスト生成には不向き

Decoder-only (GPT*, PaLM, LaMDA, GLaM)

- 会話、文章生成、要約
- zero-shot/few-shot learning が可能

Encoder-Decoder (T5, T0*)

- Transformerと同様の形で学習されている汎用モデル
- エンコーダとデコーダの使い分けがやや難しい。

大規模言語モデルをカスタマイズする手法

Prompting と Fine-tuning

- zero-shot prompting: 学習データなしでモデルをそのまま使う
- few-shot prompting: 学習データ(例)はほとんどない
- fine-tuning: より多くの学習データを用いてモデルを特化させる(どの程度かはユースケースに依存)



prompt learning と fine-tuning の比較

GPT-3 への入力例

```
Translate English to French:  
cheese => .....
```

この文で Translate English to French の意味を補完

```
Translate English to French:  
sea otter => loutre de mer  
cheese => .....
```

```
Translate English to French:  
sea otter => loutre de mer  
peppermint => menthe poivrée  
plush girafe => girafe peluche  
cheese => .....
```

zero-shot (prompt learning)

- ヒントなしに課題を解かせる
- 精度は低くなりがち
- パラメータ(勾配)の更新なし

One-shot (prompt learning)

- ヒント(学習データ)を一つだけ与える
- パラメータの更新なし

Few-shot (prompt learning)

- 推論時にタスクに関する説明と少量のヒントを与える
- パラメータの更新なし

fine-tuning

- 従来の学習方法
- 事前学習済みモデルをもとに目標タスクのデータセットで教師あり学習
- パラメータ更新あり

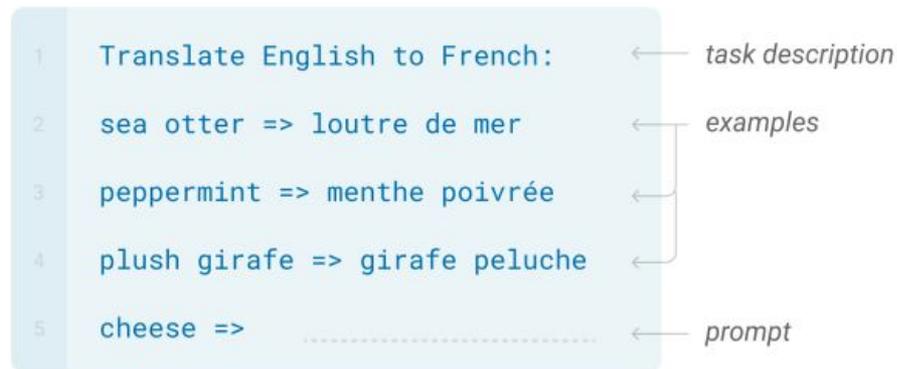


モデルの汎化能力には規模が重要

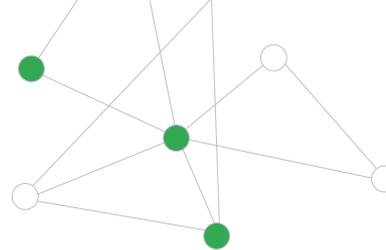
大規模言語モデルが few-shot learning で驚異的な性能を達成.

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



Google による Large Language Model 例



Unifying Language Learning Paradigms
(**UL2**), Encoder-Decoder / Decoder-only
October 2022

[\[Blog Post\]](#); [\[Paper\]](#)

Pathways Language Model (**PaLM**)
Decoder-only, 8B, 62B, 535B.
May 2022

[\[Blog Post\]](#); [\[Paper\]](#)

Generalist Language Model (**GLaM**)
Decoder-only, up to 1.2T params
Dec 2021

[\[Blog post\]](#) | [\[Paper\]](#)

Multitask Unified Model(**MUM**)
Decoder-only, up to 13B params
May 2021

[\[Blog Post\]](#)

Language Model for Dialogue
Applications (**LaMDA**)
Decoder-only, 137B
May 2021

[\[Blog Post\]](#)

Text to Text Transformer (**T5**)
Encoder-Decoder
Oct 2019

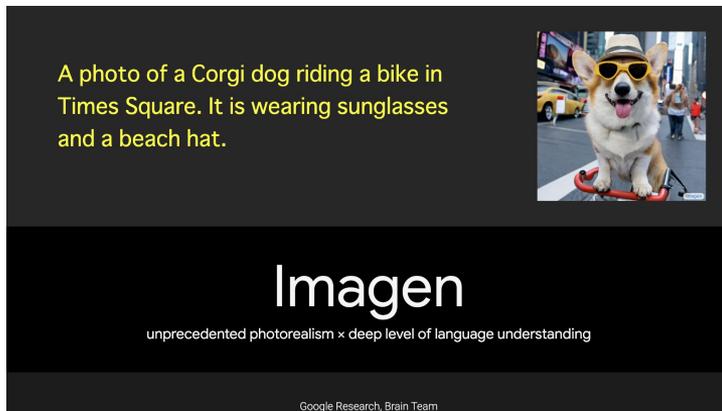
[\[Paper\]](#)

その他主要な大規模言語モデル例

Model Name	開発	発表(年)	Encoder-Decoder	Model size	
GPT2	OpenAI	2019	Decoder only	1.5B	WebText と書籍コーパスで学習させ、モデルから生成される言語の流暢さに画期的な進歩をもたらした
GPT3	OpenAI	2020	Decoder only	175B	Wikipedia やCommon Crawl(Webサイトから収集されたデータ)などから集めた45TBの膨大なテキストデータに前処理を行った570GBのデータセットを学習。1750億個のパラメータを持つ自己回帰(ある単語の次に出てくる単語を予測するモデル)型言語モデル
T0*	HF	2021	Enc-Dec	11B	16分の1のサイズのベースモデルからGPT3と同等の品質のモデルが得られることを示し、様々なタスクでファインチューニングされている
Gopher	DeepMind	2021	Decoder-only	280B	ウェブページ、書籍、ニュース記事、コードで学習 読解力、事実確認、毒舌の識別は得意だが、論理的推論や常識的なタスクは苦手な傾向に
Megatron-Turing	DeepSpeed & Microsoft	2021	Decoder-only	530B	クリーンなCommon Crawlデータで学習 PaLMと並ぶ最大規模の公開モデル

その他特殊な LLM

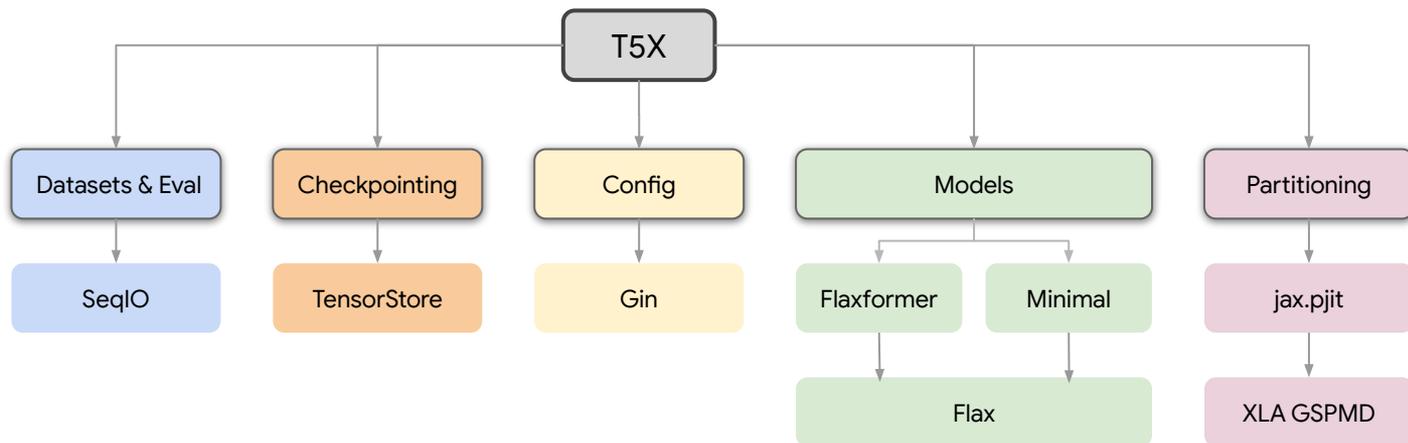
- [Imagen](#) [Google] - text-to-image diffusion model
- [Codex](#) [OpenAI] - translates natural language to code, [GitHub Copilot](#).
- [DALL-E-2](#) [OpenAI] - Text 2 Image
- [Stable Diffusion](#) - Text 2 Image
- [その他](#)



Google Cloud における T5X の活用



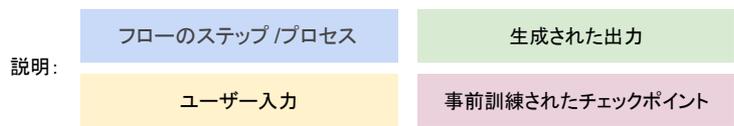
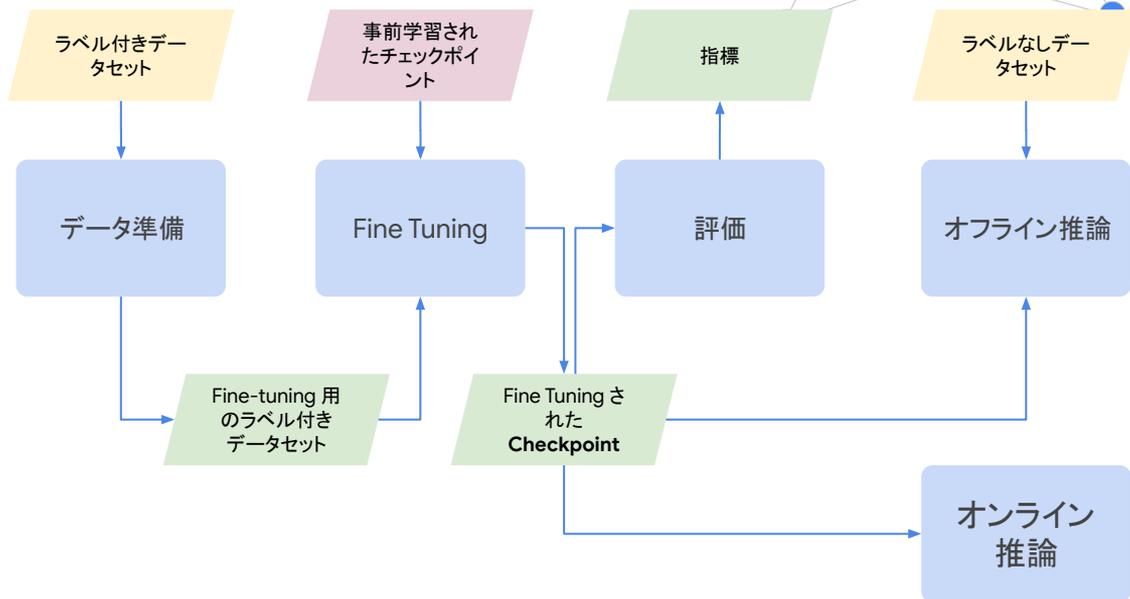
T5X とは



- さまざまな規模で JAX モデルをトレーニング、評価、および推論するためのライブラリ
- Transformer ベースの言語モデル
- TPU 用に最適化
- GPU および CPU アクセラレーションもサポート

Source: [Scaling Up Models and Data with t5x and seqio](#)

T5X によるファインチューニングの流れ



T5X で利用可能な事前学習モデル

チェックポイント:

`gs://t5-data/pretrained_models/t5x/`

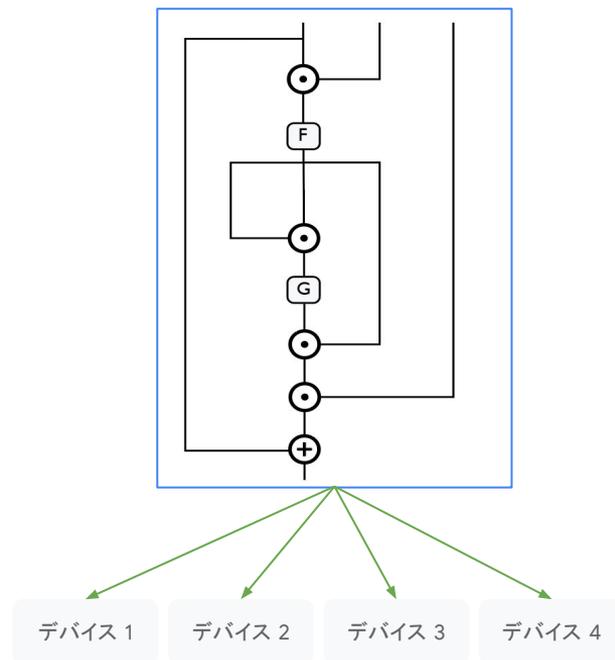
Gin ファイル:

`t5x/configs/models/`

Model	説明
T5	オリジナルのT5
T5.1.1	T5 を改良
T5.1.1 LM-Adapted	プロンプトチューニングの論文に基づきLMオブジェクトに100kステップの追加学習を実施
mT5	多言語対応T5
ByT5	入出力にUTF-8を使用した「トークン・フリー」モデル
LongT5	長いシーケンスのための効率的なT5
MoE	Moe の実験に便利
Flan-T5	Few-shot learning および fine-tuning 用の汎用 T5 チェックポイント
UL2	データセットやセットアップを問わず普遍的に有効なモデルを事前学習するための統一されたフレームワーク

T5X ジョブの並列化による学習のスケーリング: SPMD(Single Program Multi Data) 並列処理

- すべてのチップが同じプログラムを実行
- すべてのデバイスで実行する単一のプログラムを維持しながら、TPU デバイス全体の計算で個々の演算子を分割
- すべてのパーティションの 1 回限りのコンパイル、数千パーティションへの高速コンパイル



T5X における 2 種類の並列処理

SPMD 並列処理

データ 並列性

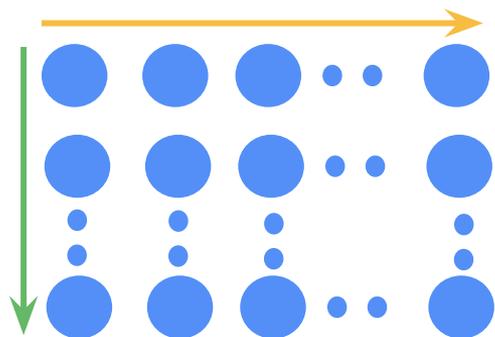
- 大規模データセットでモデルを学習
- 並列化されたモデルの複数のレプリカを作成
- 各レプリカに異なるサンプルセットを与えて計算

モデル 並列性

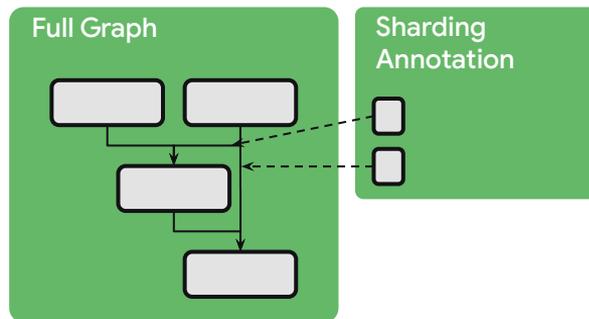
- 単一のメモリに収まらない大規模モデルを学習
 - パラメータの分割 (Tensor 並列処理)
 - レイヤーの分割 (パイプライン並列処理)

T5X はデータとモデルの並列処理の両方をサポート

- 論理メッシュ全体でのモデルパラメーター、アクティベーション、データの分割
- T5X パーティショニング: XLA SPMD パーティショナー上に構築されたjax.pjit API を使用

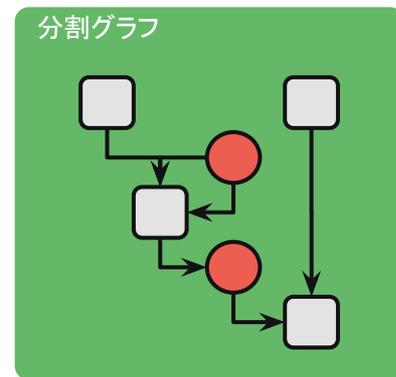


デバイス メッシュ



1つの大きなデバイス
に存在しているかのよ
うにモデルコードを記述

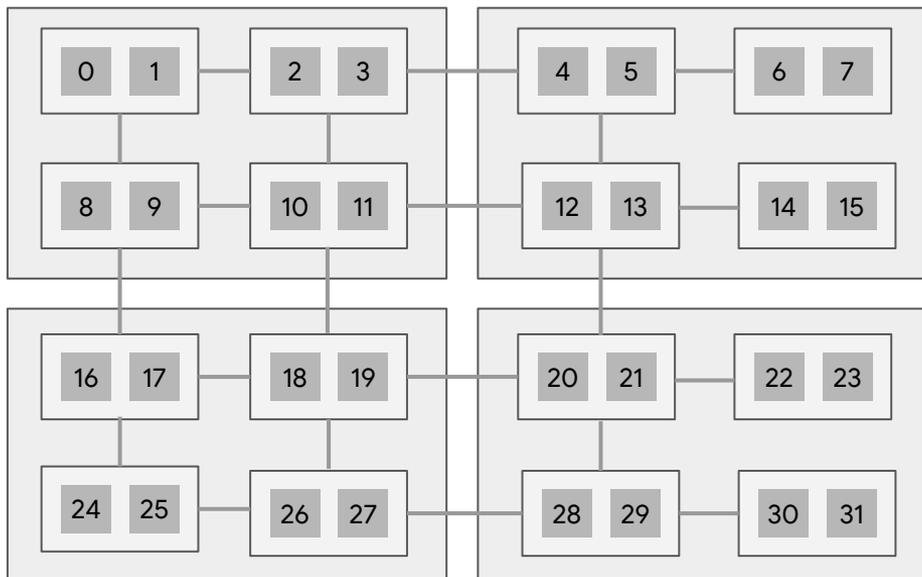
シャーディングのため
にいくつかの重要なテ
ンソルに注釈を付ける



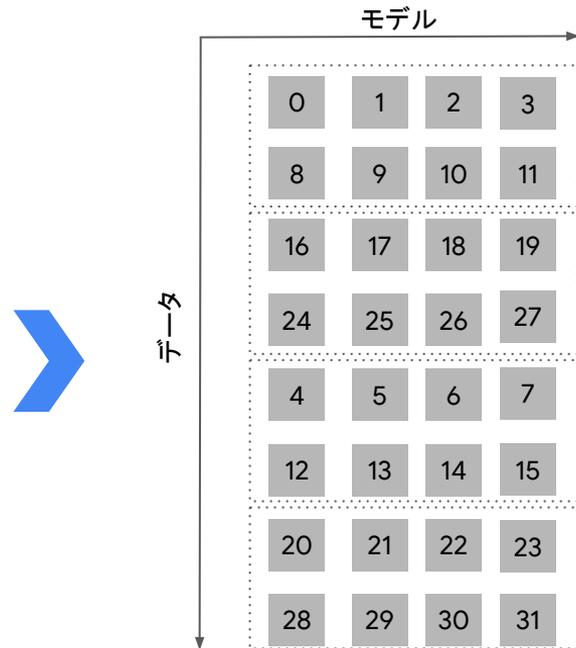
集散的なクロスコア通信操
作を使用した、デバイスごと
に分割されたグラフ

T5X パーティショニングの基礎: 物理 → 論理メッシュ マッピング

Cloud TPU v3-32 スライス物理トポロジ



次元 8 x 4 (データ x モデル) の論理メッシュ



T5X パーティショニングの設定

- T5X でのパーティショニング構成手順
 1. パラメーターとアクティベーション配列の次元の論理軸名を指定
 2. 論理名をアクセラレータ メッシュの物理軸にマッピング
- パーティショニングは、Gin 構成を使用して PjitPartitioner クラスを介して `jax.pjit` バックエンドを通じて提供

```
# gin 構成ファイル
```

```
partitioning.PjitPartitioner:
```

```
num_partitions = 4
```

```
model_parallel_submesh = (4, 1, 1, 1)
```

```
logical_axis_rules = [
```

```
    ('head', 'model'),
```

```
    ('embed', 'model'),
```

```
    ('embed', 'data'),
```

```
    ('vocab', 'model'),
```

```
]
```

モデルを分散させるコア数

物理トポロジーと論理メッシュのマッピング(サブメッシュ)を指定

(x,y,z,c) の4タプル
x,y,zはメッシュのサイズ
cはチップ上のコア数

論理軸名とハードウェア軸
(モデル、データ)の対応づけ

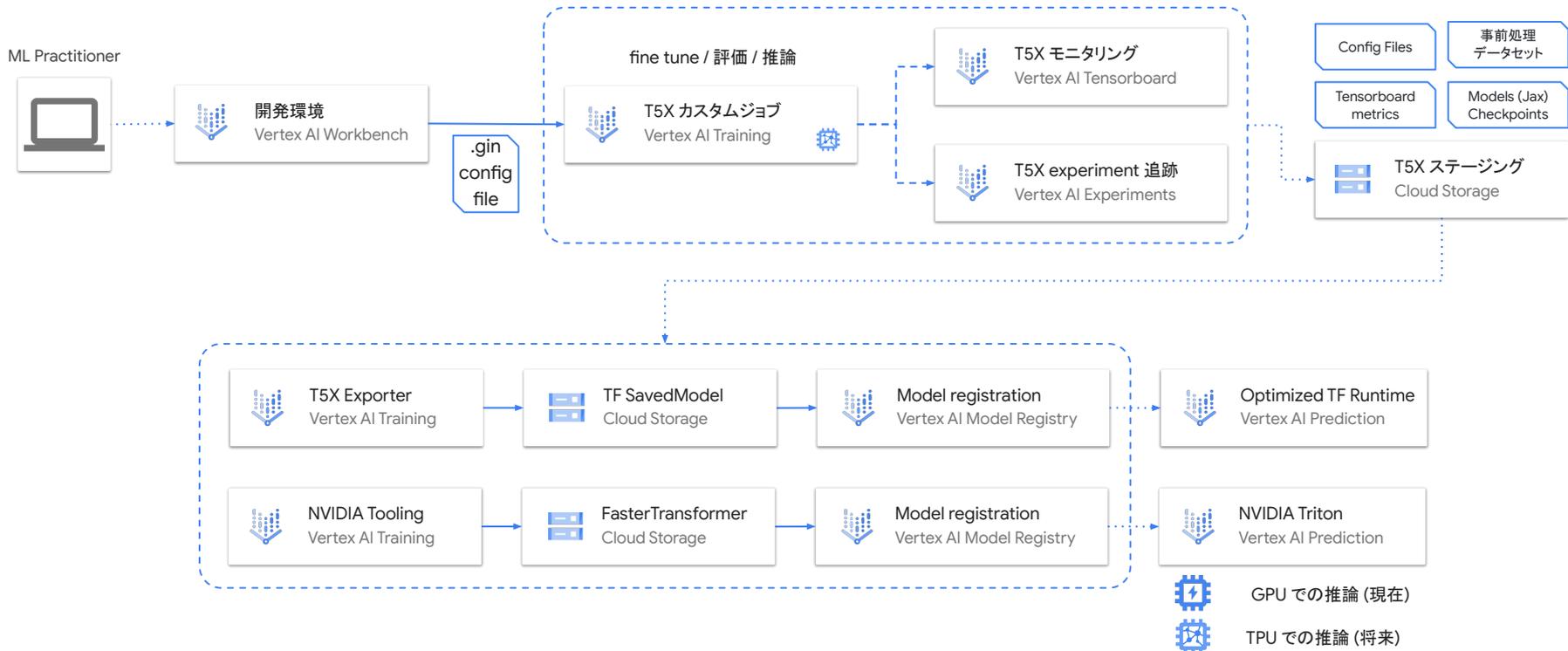
ソリューション概要

Large Language Model



ソリューション概要

Large Language Model

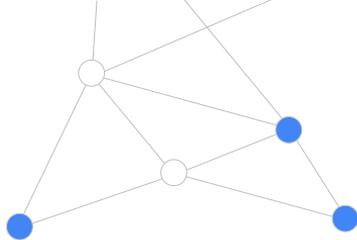


Google Cloud で LLM を利用するオプション

サービス提供モード	アクセラレーター	GCE	GKE	Vertex AI Prediction
オンラインと バッチ	GPU	A100 80GB ^{NEW} A100 40GB T4 V100	A100 80GB ^{NEW} A100 40GB T4 V100	A100 40GB ^{NEW} T4 V100
バッチ	TPU	TPU v2 TPU v3 TPU v4	TPU v2 TPU v3 TPU v4 実験的	該当なし (2023年提供予定)

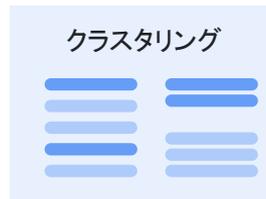
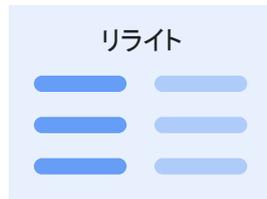
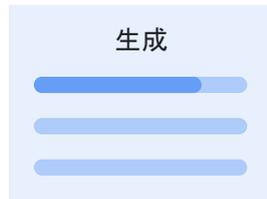
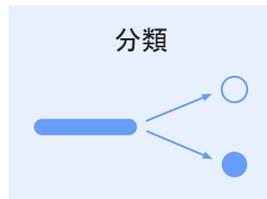
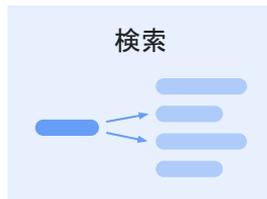
- TPU でのバッチ推論は、T5X を使用した Vertex AI トレーニングでカスタム ジョブとして実行されます
- GCE VM 上の GPU でのオンライン推論と、代替手段としての Vertex AI Prediction の調査。
- 大規模モデルを提供するためのモデル蒸留は、将来予定されています

まとめ

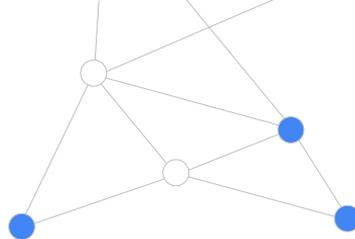


新しいユースケースの可能性を 広げるLM

大規模なオープンソースの事前学習済み言語モデルと転移学習技術の利用により、LLM のいくつかのユースケースがより身近に



Google Cloud で LLM を学習するメリット



01

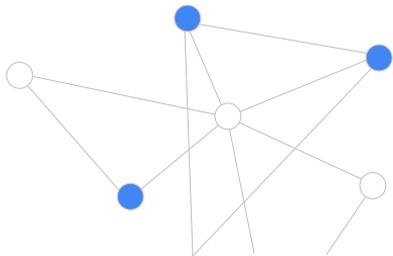
LLM の特定のハードウェアやソフトウェア構成のユースケースを示すコードサンプル

02

費用対効果の高いハードウェア運用
- TPU または GPU

03

Google の先駆的な AI 研究とクラス最高のアルゴリズムに基づいて構築



github.com/GoogleCloudPlatform/t5x-on-vertex-ai

Thank you

