

サマースクール2023 大規模言語モデル Day5

Parameter Efficient Fine-Tuning

Parameter Efficient Fine-Tuning

目次

- 01 Day5 イン트로ダクション
- 02 大規模言語モデルの Fine-Tuning
- 03 Instruction Tuning
- 04 **Parameter Efficient Fine-Tuning**
- 05 Day5 まとめ

LLM Fine-Tuning における問題意識

問題意識

- 大規模言語モデルの性能改善や様々なタスク・ドメインへの適応を実現したい
- 莫大なリソースを要する Pre-Training は多くの主体にとってハードルが高い

- 大規模言語モデルは膨大なパラメータを有するため、Fine-Tuning であっても全てのパラメータを扱えない場合がある
- Catastrophic Forgetting や過学習で、事前学習モデルの性能を毀損する恐れ

解決の方向性

- **Fine-Tuning** によって事前学習済みモデルの性能改善やタスク・ドメイン適応を実現
- 特に **Instruction Tuning** によって、対話性能や Zero/Few-Shot 性能を向上

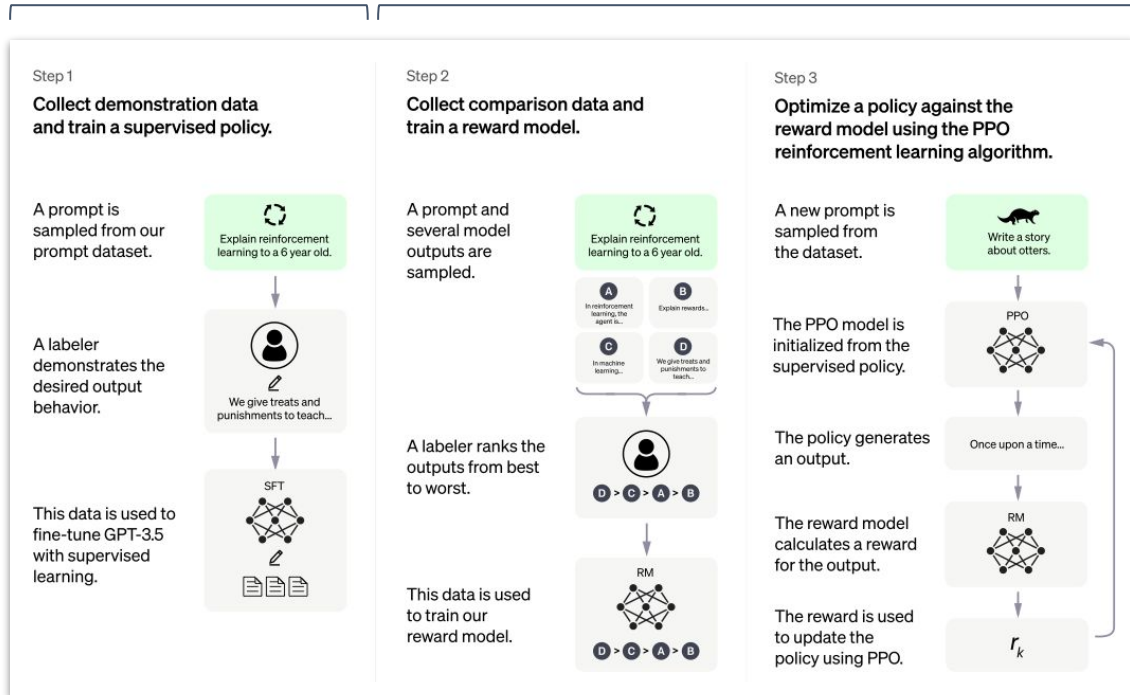
- 追加的に設定したパラメータや、一部のパラメータのみを訓練・更新の対象とすることで、効率的な Fine-Tuning を実現
- このような手法を特に **Parameter Efficient Fine-Tuning (PEFT)** と呼ぶ

1. Day5 イン트로ダクション

LLM Fine-Tuning 事例 | ChatGPT

Supervised Fine-Tuning

Reinforcement Learning from Human Feedback (RLHF)



- 事前学習済みLLMは高い性能を示すが、必ずしも人間の価値観に沿った出力をしない
- **ChatGPT**では、InstructGPT論文^{※1}で提案された手法に則って、上記の問題に対処
- 具体的に以下を組み合わせ、人間の価値観への調整を実現
 - **Supervised Fine-Tuning** (Day5 トピック)
 - **RLHF** (Day6 トピック)

OpenAI "[Introducing ChatGPT](#)"より引用し、一部改変

※1 Ouyang, Long, et al. Advances in Neural Information Processing Systems 35 (2022): 27730-27744.

LLM Fine-Tuning 事例 | GPT-3.5 Fine-Tuning

GPT-3.5 Turbo fine-tuning and API updates

Developers can now bring their own data to customize GPT-3.5 Turbo for their use cases.

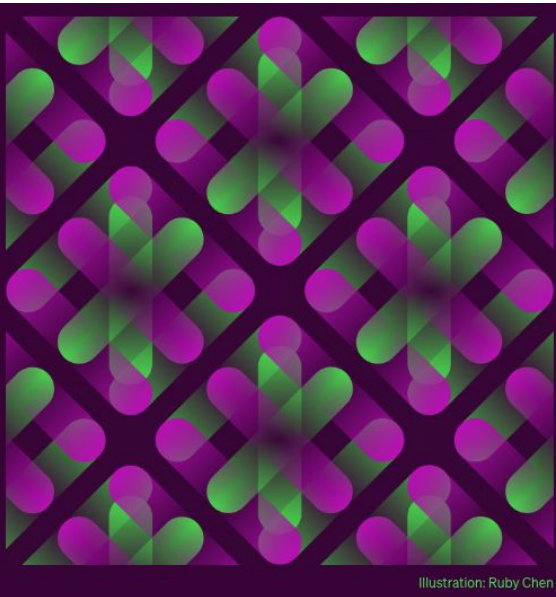


Illustration: Ruby Chen

- OpenAI API で GPT-3.5 の Fine-Tuning 機能がリリース
- 自前のデータセットを用いた Fine-Tuning が実施可能に
- 以下のような用途が例示
 - 言語指定・話し方の調整
 - 応答フォーマットの指定
- Promptingと比較して、以下のような利点が例示
 - トークン・処理時間節約
 - 応答の品質・制御性向上

1. Day5 イン트로ダクション

LLM Fine-Tuning 事例 | Med-PaLM

You are a helpful medical knowledge assistant. Provide useful, complete, and scientifically-grounded answers to common consumer search queries about health.

Question: How do you treat skin redness?

Complete Answer: It depends on the cause of the skin redness. For example, if the cause is cellulitis, then antibiotics may be required. However, this might be inappropriate for other causes of redness such as eczema. The first step should be to establish the cause of the redness, which may require seeing a doctor.

You are a helpful medical knowledge assistant. Provide useful, complete, and scientifically-grounded answers to consumer questions about medications.

Question: Why take simvastatin?

Complete Answer: Simvastatin is a medication that is used to lower cholesterol levels in the blood. Simvastatin is typically used to treat high cholesterol and to prevent heart disease, and to reduce the risk of heart attacks and strokes in those with the greatest risk of these happening (or re-occurring). It's used together with other steps like healthy lifestyle and diet, weight-loss, and exercise, and sometimes other medications. Simvastatin may also be used to decrease the amount of...

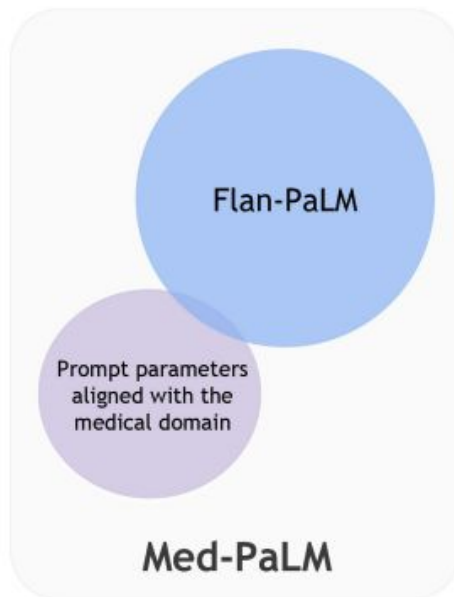
You are a helpful medical knowledge assistant. Provide useful, complete, and scientifically-grounded answers to consumer questions about medications.

Question: How can I split a 3mg or 4mg warfarin pill so I can get a 2.5mg pill?

Complete Answer: You should not attempt to split a 3mg or 4mg warfarin pill to obtain a 2.5mg dose. If you are prescribed warfarin in a 2.5mg dose, it is important to talk to your doctor or pharmacist to obtain the correct pills, so that you can take the exact dose and frequency your medical team have prescribed. Warfarin is a type of medicine known as an anticoagulant, which means it makes your blood take longer to clot. The exact amount required to achieve the desired effect can vary.

Extended Data Fig. 1 | Instruction prompt tuning for Med-PaLM. We use instructions and exemplars from a panel of qualified clinicians for each of the consumer medical question answering datasets and use them to instruction

→
Instruction
Prompt Tuning



prompt tune Flan-PaLM. Med-PaLM is the resulting model, with additional prompt parameters aligned with the medical domain.

※1より引用

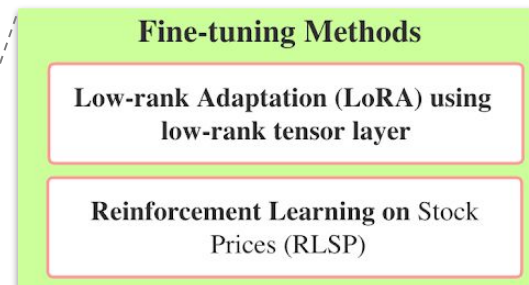
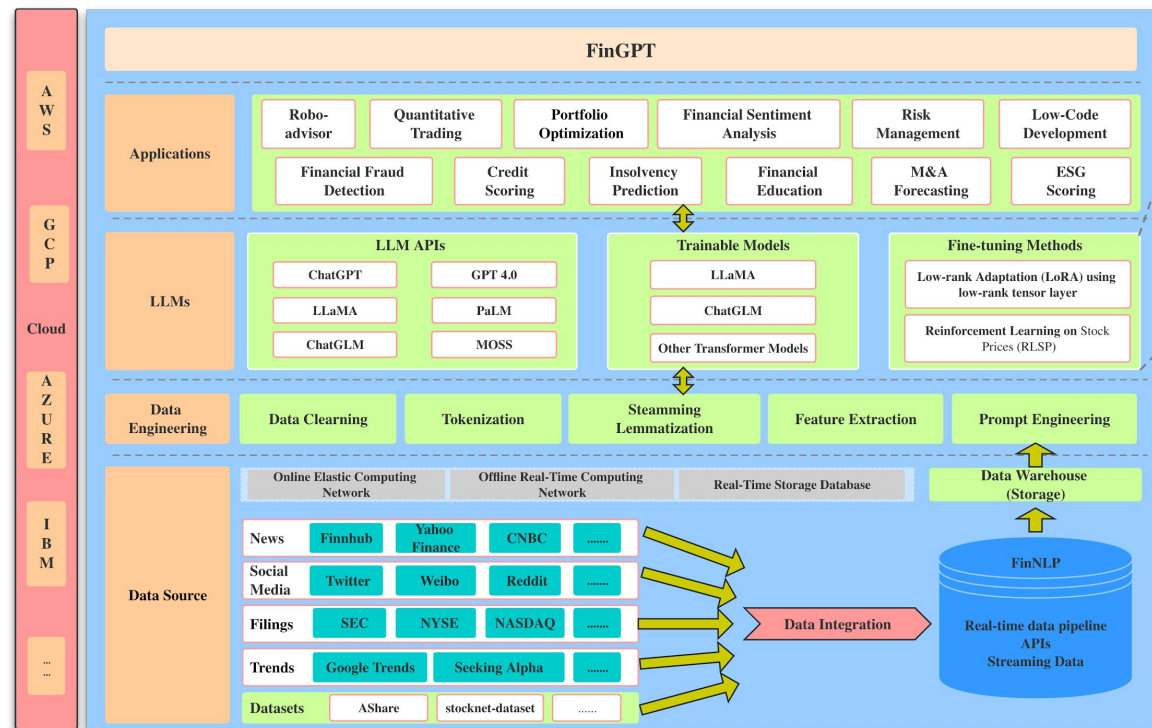
- **Med-PaLM^{※1}** :
Google が開発したLLM **PaLM^{※2}** を医療向けに Fine-Tuning したモデル
- 医療質疑応答タスクで SOTA
- 複数の Fine-Tuning 手法を組み合わせ、Instruction Prompt Tuning を適用

※1 Singhal, Karan, et al. "Large language models encode clinical knowledge." *Nature* (2023): 1-9.

※2 Chowdhery, Aakanksha, et al. "Palm: Scaling language modeling with pathways." *arXiv preprint arXiv:2204.02311* (2022).

1. Day5 イン트로ダクション

LLM Fine-Tuning 事例 | FinGPT



- **FinGPT^{※1} :**
金融分野に特化したLLMを開発するためのオープンソース・フレームワーク
- 汎用の事前学習済みモデルをFine-Tuningする手法を推進

※1 Yang, Hongyang, Xiao-Yang Liu, and Christina Dan Wang. "FinGPT: Open-Source Financial Large Language Models." arXiv preprint arXiv:2306.06031 (2023).

※1より引用し、一部改変

大規模言語モデル講座 Day5 の目標

Goal 1

大規模言語モデルの典型的な訓練フローにおいて、**Fine-Tuning** が **Pre-Training** (Day3-4) や **RLHF** (Day6) に対してどう位置付けられるか説明できる

Goal 2

大規模言語モデルの **Fine-Tuning** において、特に重要なアプローチである **Instruction Tuning** や **PEFT** が既存手法に対してどう位置付けられるか説明できる

Goal 3

Instruction Tuning および **PEFT** について、その理論や目的を十分に理解した上で、実際にそれらを実装し、大規模言語モデルの性能改善を実現できる

Parameter Efficient Fine-Tuning

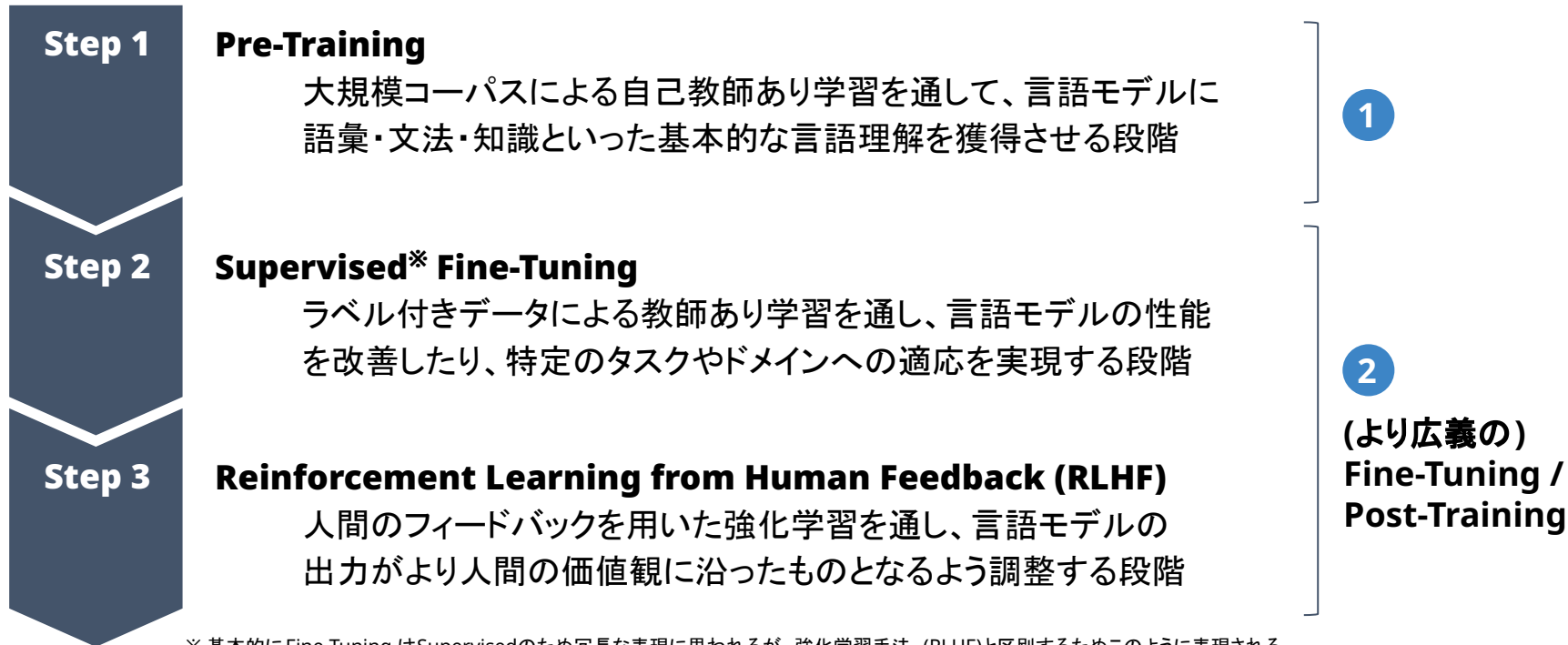
目次

- 01 Day5 イン트로ダクション
- 02 大規模言語モデルの Fine-Tuning**
- 03 Instruction Tuning
- 04 Parameter Efficient Fine-Tuning
- 05 Day5 まとめ

2. 大規模言語モデルの Fine-Tuning

LLM訓練フローにおける Fine-Tuning

x : 次ページで整理



※ 基本的にFine-Tuning はSupervisedのため冗長な表現に思われるが、強化学習手法 (RLHF)と区別するためこのように表現される。
また、あえてこのように表現する場合には、一般の教師あり Fine-Tuningではなく、後述の Instruction Tuningを指すことが多い。

2. 大規模言語モデルの Fine-Tuning

Pre-Training vs. Fine-Tuning

: Day5のトピック

1 Pre-Training

2 Fine-Tuning / Post-Training

目的

- 語彙・文法・知識・推論能力などの言語能力を、言語モデルに導入

- 事前学習済みモデルの性能改善や、様々なタスクに対する適応を実現

一般的な手法

- 自己教師あり学習
 - Next Token Prediction
 - Masked Language Model

- 教師あり学習
 - 下流タスクへの特化
 - Instruction Tuning
- 強化学習 (RLHF, Day6トピック)

データ

- 大規模データセット
 - 例 CommonCrawl (GPT-3): 410B tokens (570GB)

- 良質な小規模データセット
 - 例 LIMA: 1000サンプル (3MB)
- 人間・モデルによるフィードバック

2. 大規模言語モデルの Fine-Tuning

大規模言語モデルの Fine-Tuning

従来の Fine-Tuning

大規模言語モデル特有の Fine-Tuning

主目的

- 事前学習済みモデルをベースとし、特定の下流タスクを高い精度で解けるモデルを効率的に獲得
- 事前学習済みモデルの出力内容や形式を用途に応じて調整・制御
- 事前学習済みモデルの未知タスクに対するZero/Few-shot性能を改善

1

タスク設計

- 解きたいタスクで教師あり学習
- 例: 感情分析・自然言語推論
- 指示文を入力、それに対する理想的な出力文を正解として教師あり学習 (**Instruction Tuning**)

2

重み更新

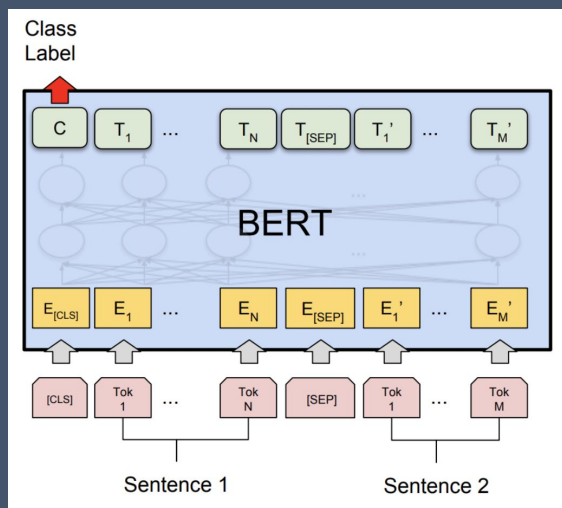
- 事前学習済みモデルが有する全てのパラメータについて更新を実施 (対比的に**Full FT**と呼ぶことがある)
- 学習済みパラメータは更新せず、別途設定した追加パラメータを更新 (**Parameter Efficient Fine-Tuning**)

2. 大規模言語モデルの Fine-Tuning

1 Fine-Tuning のタスク設計

従来の Fine-Tuning

- 特定の下流タスクで教師あり学習を実施
- 主に下流タスク用の特殊トークンを活用



PyTorch Tutorial “[Dynamic Quantization on BERT](#)”より引用

Instruction Tuning

- 指示文に対して、理想的な出力文を正解とする教師あり学習を実施
- 様々なタスクがこの入出力形式に内包

Finetune on many tasks (“instruction-tuning”)

Input (Commonsense Reasoning)

Here is a goal: Get a cool sleep on summer days.
How would you accomplish this goal?
OPTIONS:
-Keep stack of pillow cases in fridge.
-Keep stack of pillow cases in oven.

Target

keep stack of pillow cases in fridge

Input (Translation)

Translate this sentence to Spanish:
The new office building was built in less than three months.

Target

El nuevo edificio de oficinas se construyó en tres meses.

Sentiment analysis tasks

Coreference resolution tasks

...

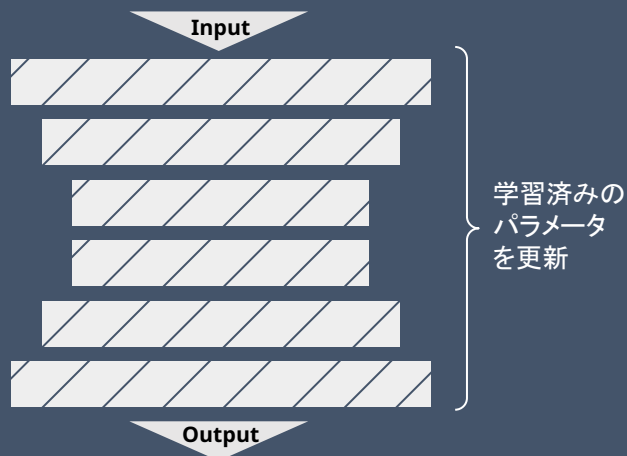
Wei, Jason, et al. "Finetuned language models are zero-shot learners." arXiv preprint arXiv:2109.01652 (2021). より引用し、一部改変

2. 大規模言語モデルの Fine-Tuning

② Fine-Tuning の重み更新

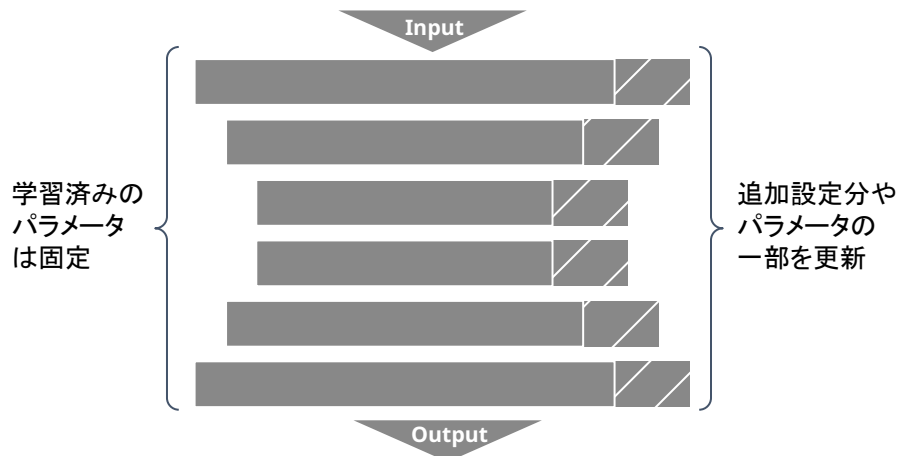
従来の Fine-Tuning (Full-FT)

- 事前学習済みモデルが持つ全てのパラメータについて、更新を実施
- より確実な性能改善が期待される一方、多くのリソースを必要とする



Parameter Efficient Fine-Tuning

- 追加的に設定したパラメータや、一部のパラメータのみを訓練・更新
- 適切に用いることができれば、少ないリソースで性能改善を達成できる



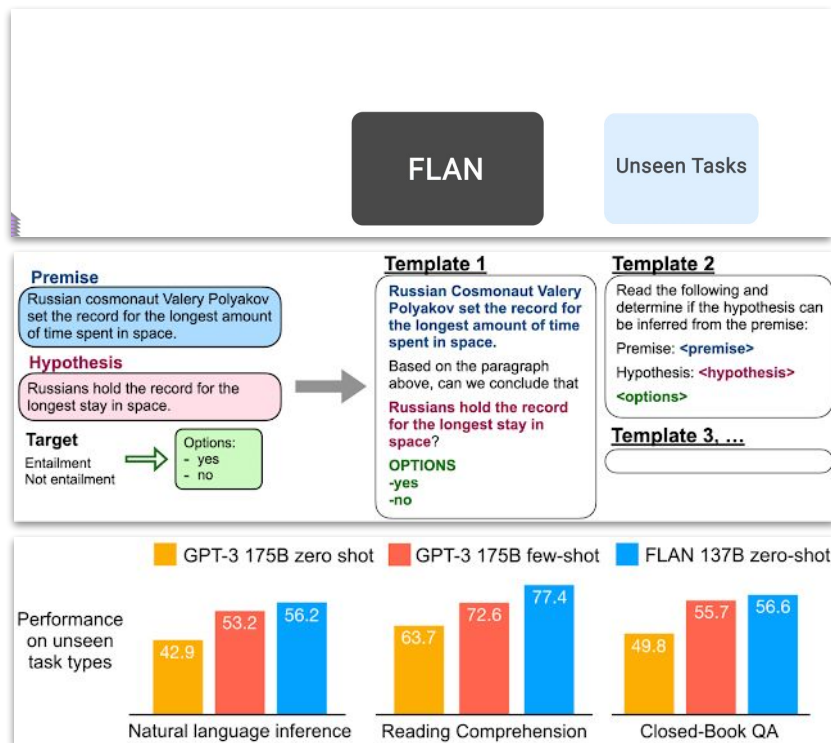
Parameter Efficient Fine-Tuning

目次

- 01 Day5 イン트로ダクション
- 02 大規模言語モデルの Fine-Tuning
- 03 Instruction Tuning**
- 04 Parameter Efficient Fine-Tuning
- 05 Day5 まとめ

3. Instruction Tuning

Instruction Tuning 概要 | FLAN論文による提案



Google Research "[Introducing FLAN: More generalizable Language Models with Instruction Fine-Tuning](#)"より引用

- Wei, Jason, et al. "**Finetuned language models are zero-shot learners.**" arXiv preprint arXiv:2109.01652 (2021).
- 様々なタスクを指示・回答という形式に統一したデータセットにより、言語モデルを Fine-Tuning する手法を提案 (**Instruction Tuning**)
- このように Fine-Tuning されたモデルは評価に用いられた25のタスクの内、
 - 21タスクで、Zero-shot性能が向上
 - 20タスクで、よりパラメータ数の多い GPT-3と比べ、より高いZero-shot性能

3. Instruction Tuning

Instruction Tuning 概要 | タスク構成と入出力例

	入力 (Instruction)	出力 (Instance)
構成	<ul style="list-style-type: none">- タスクを指定する指示文- (Optional) 付随する補足情報	<ul style="list-style-type: none">- 与えられた指示文に対する、理想的な回答例
具体例 (FLAN)	<ul style="list-style-type: none">- "Víte, rozhodl jsem se, že si pořídím psa. Translate to English"	<ul style="list-style-type: none">- "You know, I decided to get a dog."
	<ul style="list-style-type: none">- "i'm 10x cooler than all of you! What is the sentiment of this tweet?"	<ul style="list-style-type: none">- "positive"
	<ul style="list-style-type: none">- "Solve $0 = 56*x + 16*x - 720$ for x."	<ul style="list-style-type: none">- "10"

3. Instruction Tuning

Instruction Tuning の有効性と限界

：次ページ以降で詳解

Instruction Tuning の有効性

- Zero-shot性能の向上
 - **FLAN**^{※1}
Instruction Tuning した137Bモデルが、パラメータ数で勝る GPT-3を
超えるZero-shot性能を示した
- 指示応答性能の向上
 - **Alpaca**^{※2}
Instruction Tuning した7Bモデルが、
パラメータ数で大幅に勝る GPT-3.5
と同様の指示応答挙動を示した

Instruction Tuning の限界

- データセット作成上の困難
 - 高品質・無害で多様なデータセットを用意することが必要
 - それらを両立するためには、人的・技術的リソースを要する
- 知識は導入可能か
 - **Kung and Peng (2023)**^{※3}
性能改善がタスクの理解ではなく、出力形式のような表面的な事項の
学習に起因する可能性を指摘

※1 Wei, Jason, et al. arXiv preprint arXiv:2109.01652 (2021).

※2 Taori, Rohan, et al. Stanford Center for Research on Foundation Models.
<https://crfm.stanford.edu/2023/03/13/alpaca.html> 3.6 (2023): 7.

※3 Kung, Po-Nien, and Nanyun Peng. "Do Models Really Learn to Follow Instructions? An Empirical Study of Instruction Tuning." arXiv preprint arXiv:2305.11383 (2023).

3. Instruction Tuning

Instruction データセットの構築手法

ラベル付き
データセットの
統合

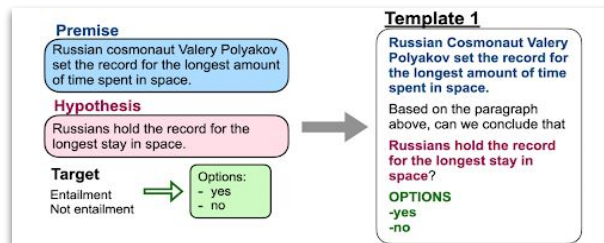
- 既存のラベル付きデータセットを、
テンプレートを用いて変換
- **FLAN**^{※1} : 62個のデータセットを統合

人間による
データ作成

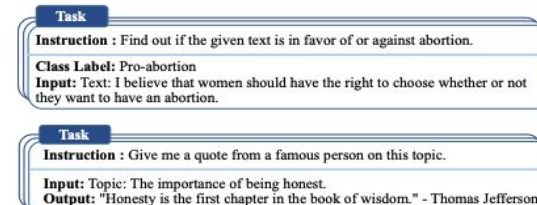
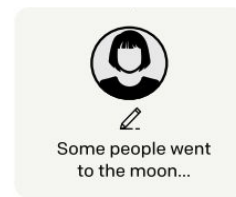
- 指示文に対する回答を人間が作成
- **InstructGPT**^{※2} : 人間が作成した
指示文に対し、人間が回答を作成

LLMによる
データ生成

- 指示文に対する回答をLLMが生成
- **Self-Instruct**^{※3} : LLMによる指示文
と回答の生成フレームワークを提案



A labeler
demonstrates the
desired output
behavior.



※1 Wei, Jason, et al. arXiv preprint arXiv:2109.01652 (2021).

※2 Ouyang, Long, et al. Advances in Neural Information Processing Systems 35 (2022): 27730-27744.

※3 Wang, Yizhong, et al. arXiv preprint arXiv:2212.10560 (2022).

3. Instruction Tuning

Instruction データセット作成上の要点

データの質

- **LIMA**^{※1}: Instruction Tuning ではデータの量より質が重要だと主張
- 1000件と少量の高品質データを用いた Instruction Tuning のみにより、RLHF で訓練されたモデルよりも高品質な回答を生成できたことを報告

データの無害性

- 事前学習済みモデルについて懸念される有害な出力を抑制するため、Instruction Tuning では有害なデータを避けて学習を実施したい
- **Llama 2**^{※2}: 無害なデータセット構築の実例を提示（次ページで詳解）

指示形式の多様性

- Sanh, Victor, et al. "**Multitask prompted training enables zero-shot task generalization.**" arXiv preprint arXiv:2110.08207 (2021).
- タスクごとの指示形式の多様化により、未知タスクに対する性能が向上

※1 Zhou, Chungting, et al. "Lima: Less is more for alignment." arXiv preprint arXiv:2305.11206 (2023).

※2 Touvron, Hugo, et al. "Llama 2: Open foundation and fine-tuned chat models." arXiv preprint arXiv:2307.09288 (2023).

3. Instruction Tuning

Instruction データセット構築事例 | Llama2※

アノテーターの選定

- アノテーターが様々なデータ作成タスクに取り組む上での資質と適性を評価するため、以下のテストを実施
1. 文法・読解・執筆テスト
 2. センシティブトピック判定・回答ランキング・回答作成テスト
 3. 品質評価テスト
 4. 回答作成テスト

※ Touvron, Hugo, et al. "Llama 2: Open foundation and fine-tuned chat models." arXiv preprint arXiv:2307.09288 (2023).



アノテーターへの指示

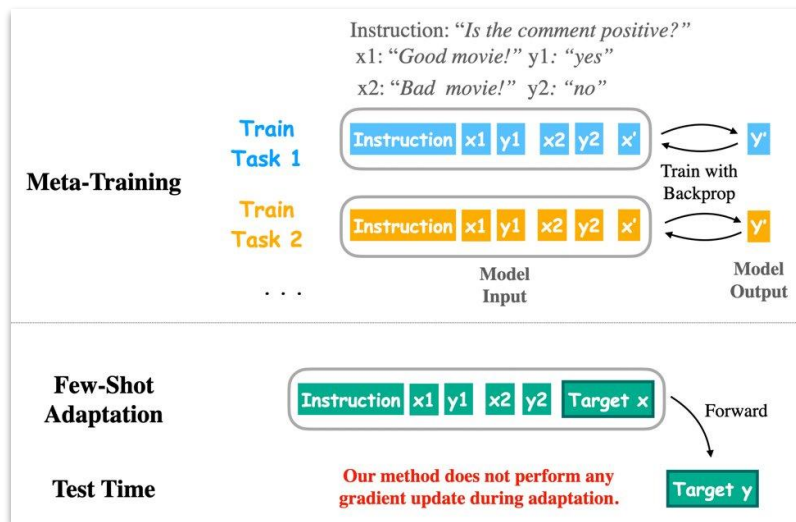
- 選定されたアノテーターに、以下を満たす指示文・回答の作成を依頼
 - Informative - Truthful
 - Relevant - Clear
 - Harmless
- 問題のある回答を誘導する指示文に対しては、無害性を優先するよう指示
- 例: 指示文作成で避けるべき項目
 - 犯罪行為の助長
 - 危険・攻撃的・性的な言動の助長

3. Instruction Tuning

Instruction Tuning の派生手法

In-Context Tuning (2022)

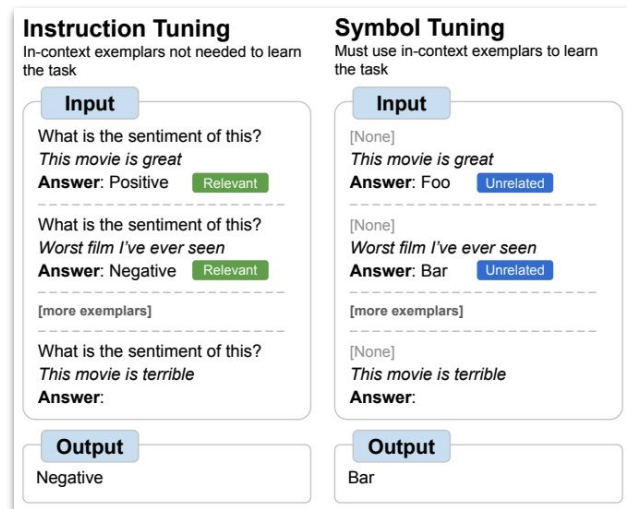
- In-Context Learning が促されるよう
事前学習済みモデルを Fine-Tuning する
ことで、Few-shot性能が向上



Chen, Yanda, et al. "Meta-learning via language model in-context tuning." arXiv preprint arXiv:2110.07814 (2021). より引用

Symbol Tuning (2023)

- 正解ラベルを無関係なシンボルに置換したデータで Fine-Tuning し、入出力関係の学習を強制することで、Few-shot性能が向上



Wei, Jerry, et al. "Symbol tuning improves in-context learning in language models." arXiv preprint arXiv:2305.08298 (2023). より引用

Parameter Efficient Fine-Tuning

目次

- 01 Day5 イン트로ダクション
- 02 LLM Fine-Tuning
- 03 Instruction Tuning
- 04 Parameter Efficient Fine-Tuning**
- 05 Day5 まとめ

4. Parameter Efficient Fine-Tuning

Full-FT と Parameter Efficient Fine-Tuning (PEFT)

：次ページで詳解

	Full-FT	Parameter Efficient Fine-Tuning (PEFT)
概要	<ul style="list-style-type: none">- 事前学習済みモデルの全パラメータについて、別タスクで更新を実施	<ul style="list-style-type: none">- 追加的に設定したパラメータや、一部のパラメータのみで更新を実施
計算リソース	<ul style="list-style-type: none">- 大規模なモデルでは、莫大な計算リソースが必要- 例 GPT-3 : 1.2TBのGPUメモリ	<ul style="list-style-type: none">- 大規模なモデルについても、限定的な計算リソースで性能改善を実現- 例 GPT-3 LoRA : 350GBのGPUメモリ※
保存領域	<ul style="list-style-type: none">- 元モデルと同サイズのパラメータを保存するため、大きな領域が必要- 例 GPT-3 : 350GBの保存領域	<ul style="list-style-type: none">- 更新部分のパラメータのみを保存すればよく、小さな保存領域で十分- 例 GPT-3 LoRA: 35MBの保存領域※

※ Hu, Edward J., et al. "Lora: Low-rank adaptation of large language models." arXiv preprint arXiv:2106.09685 (2021).

4. Parameter Efficient Fine-Tuning

PEFT によるGPUメモリ使用量の削減

- 7Bモデルの 16-bit Fine-Tuning を想定し、Full-FT と PEFT のGPUメモリ使用量を概算比較
- 以下で、全パラメータ数 $N_{\text{all}} = 7\text{B}$, 浮動小数点数のサイズ $\text{size(float)} = 2\text{MB}$ の状況に対応

VRAM by Steps	Estimation	Full-FT ($N_{\text{train}} : 7\text{B}$)	PEFT ($N_{\text{train}} : 1\text{M}$)
Model Loading	$\text{size(float)} * N_{\text{all}}$	~ 13GB	~ 13GB
Backward (Gradients)	$\text{size(float)} * N_{\text{train}}$	~ 13GB	~ 2MB
Optimizer (Adam)	$2 * \text{size(float)} * N_{\text{train}}$	~ 26GB	~ 4MB
Total	(※ 上記の他に + α として、bsに比例するForward分や、ライブラリ確保領域がある)	~ 52GB + α	~ 13GB + α

4. Parameter Efficient Fine-Tuning

PEFT 手法を評価する上での主要な観点

性能改善	<ul style="list-style-type: none">- Full-FE を実施した場合と比べて、性能改善に大きな劣化がないか- 事前学習済みモデルのサイズに依らず、性能改善が実現されるか
運用性	<ul style="list-style-type: none">- 更新する*パラメータが少なく、小さいストレージで運用が可能か- それができると複数モデルの並列運用やバージョンングが容易に
訓練効率	<ul style="list-style-type: none">- 学習する*パラメータが少なく、小さいGPUメモリでも実現可能か- GPUの効率的な活用によって高速化が可能な手法となっているか
推論効率	<ul style="list-style-type: none">- 追加するパラメータが多いことで、推論コストを増大させないか- 入力文の系列長が長くなることで、推論コストを増大させないか

※「学習するパラメータは少ないが、それに基づいて多くのパラメータが更新される」ということがあるため、「更新するパラメータ」と「学習するパラメータ」という似たような表現も、ここでは区別して使っている。

4. Parameter Efficient Fine-Tuning

様々な PEFT 手法

	運用性	訓練効率	推論効率
Method	Storage	Memory	Inference overhead
Adapters (Houlsby et al., 2019)	yes	yes	Extra FFN
AdaMix (Wang et al., 2022)	yes	yes	Extra FFN
SparseAdapter (He et al., 2022b)	yes	yes	Extra FFN
Cross-Attn tuning (Gheini et al., 2021)	yes	yes	No overhead
BitFit (Ben-Zaken et al., 2021)	yes	yes	No overhead
DiffPruning (Guo et al., 2020)	yes	no	No overhead
Fish-Mask (Sung et al., 2021)	yes	maybe ⁵	No overhead
LT-SFT (Ansell et al., 2022)	yes	maybe ⁵	No overhead
Prompt Tuning (Lester et al., 2021)	yes	yes	Extra input
Prefix-Tuning (Li and Liang, 2021)	yes	yes	Extra input
Spot (Vu et al., 2021)	yes	yes	Extra input
IPT (Qin et al., 2021)	yes	yes	Extra FFN and input
MAM Adapter (He et al., 2022a)	yes	yes	Extra FFN and input
Parallel Adapter (He et al., 2022a)	yes	yes	Extra FFN
Intrinsinc SAID (Aghajanyan et al., 2020)	no	no	No overhead

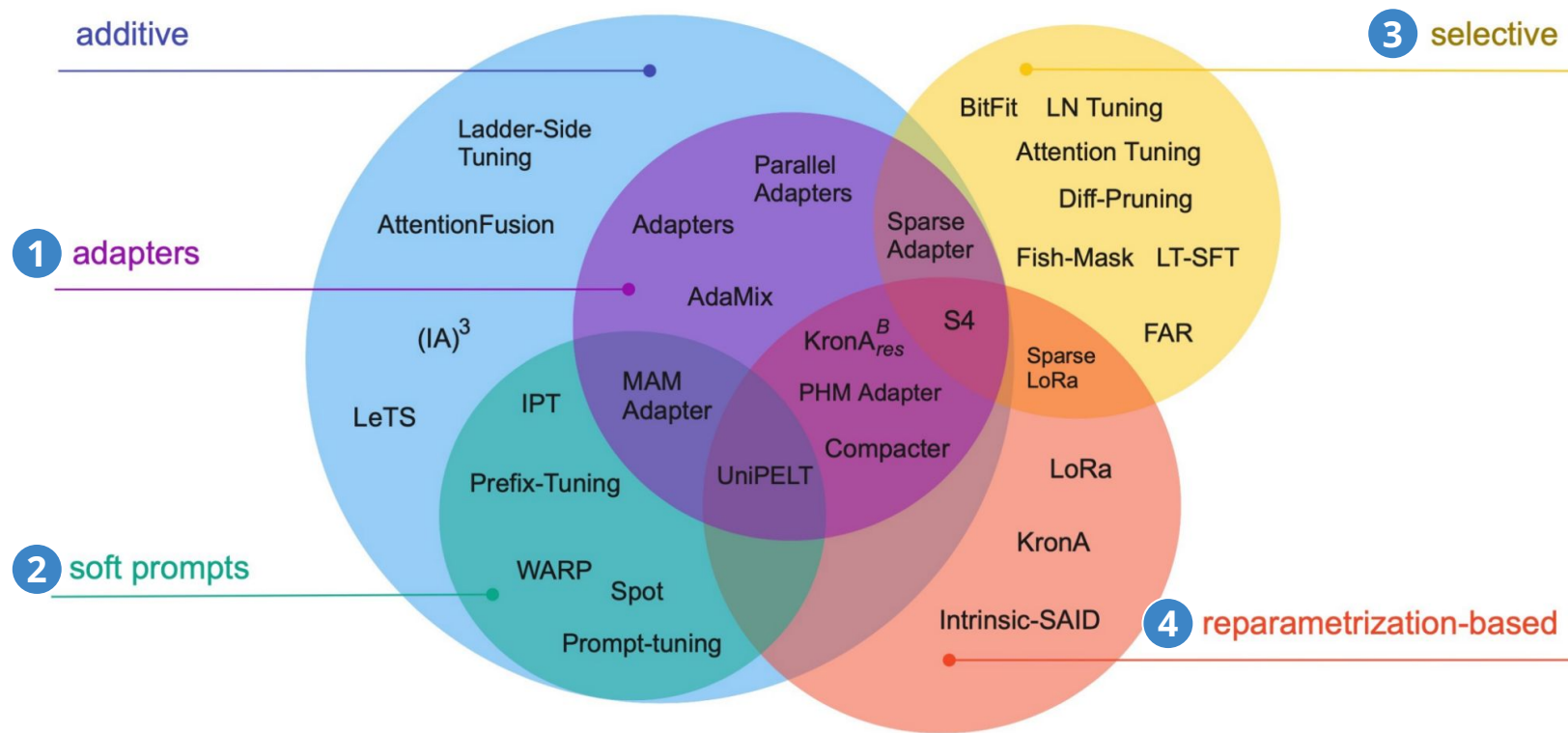
Extra FFN
FFN層の追加によって、
推論にオーバーヘッド

No overhead:
推論にオーバーヘッドを伴
わない手法

Extra input:
入力系列への追加で、
推論にオーバーヘッド

4. Parameter Efficient Fine-Tuning

PEFT 手法のカテゴリー分類



Lialin, Vladislav, Vijeta Deshpande, and Anna Rumshisky. "Scaling down to scale up: A guide to parameter-efficient fine-tuning." arXiv preprint arXiv:2303.15647 (2023). より引用

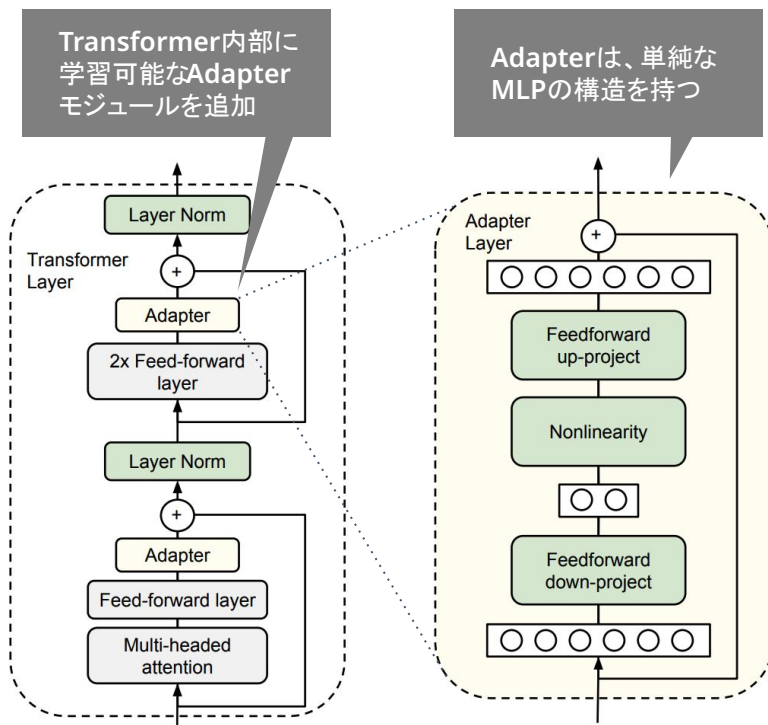
4. Parameter Efficient Fine-Tuning

PEFT 手法の代表的なカテゴリー

	概要	代表例
1 Adapter型	Transformer内部にMLP層 (Adapter) を追加し、そのみの学習を実施	Adapter (2019)
2 Soft Prompt型	入力系列にタスクごとのベクトル (Soft Prompt) を付加し、学習を実施	Prompt Tuning (2021)
3 Selective型	事前学習済みモデルが持つパラメータのうち、一部のみで学習を実施	BitFit (2021)
4 Reparametrization型	行列分解に基づき、再パラメータ化された重みについて学習を実施	LoRA (2021)

4. Parameter Efficient Fine-Tuning

1 Adapter型 | Adapter (2019)



Houlsby, Neil, et al. "Parameter-efficient transfer learning for NLP." International Conference on Machine Learning. PMLR, 2019. より引用し、一部改変

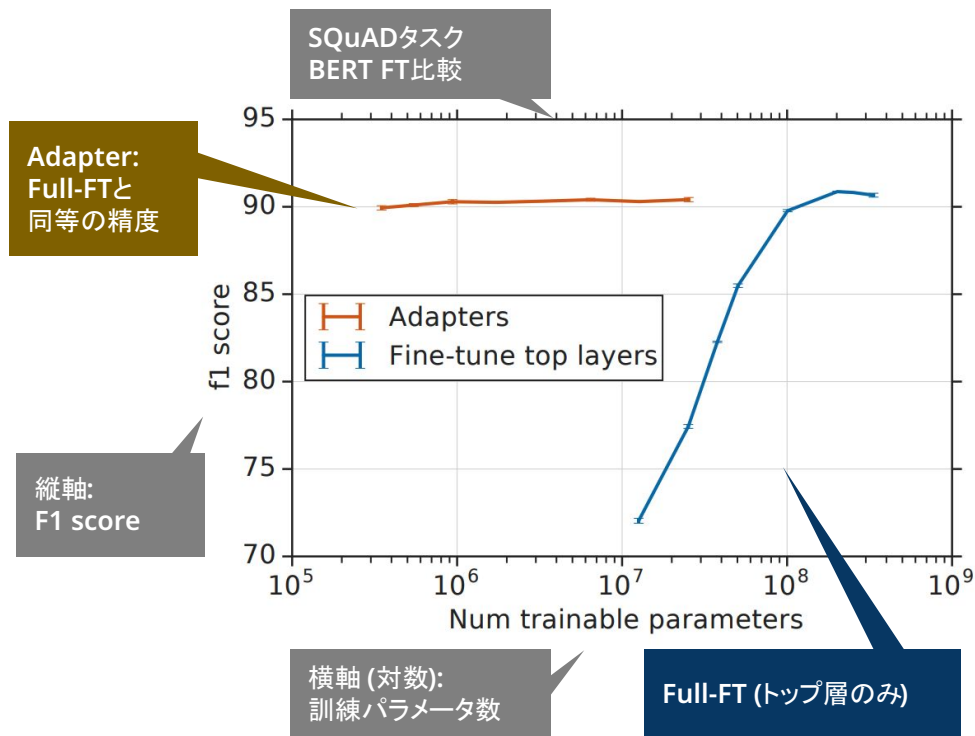
- Transformer内部に学習可能なAdapterモジュールを追加・学習
- 追加位置の異なる亜種が存在 (例: Parallel Adapter※は左図とは異なり、並列的にAdapterを追加)
- Adapterは単純なMLPの構造を持つ

```
def transformer_block_with_adapter(x):
    residual = x
    x = SelfAttention(x)
    x = FFN(x) # adapter
    x = LN(x + residual)
    residual = x
    x = FFN(x) # transformer FFN
    x = FFN(x) # adapter
    x = LN(x + residual)
    return x
```

※ He, Junxian, et al. "Towards a unified view of parameter-efficient transfer learning." arXiv preprint arXiv:2110.04366 (2021).

4. Parameter Efficient Fine-Tuning

1 Adapter型 | Adapter (2019)



Houlsby, Neil, et al. "Parameter-efficient transfer learning for NLP." International Conference on Machine Learning. PMLR, 2019. より引用し、一部改変

- Pros

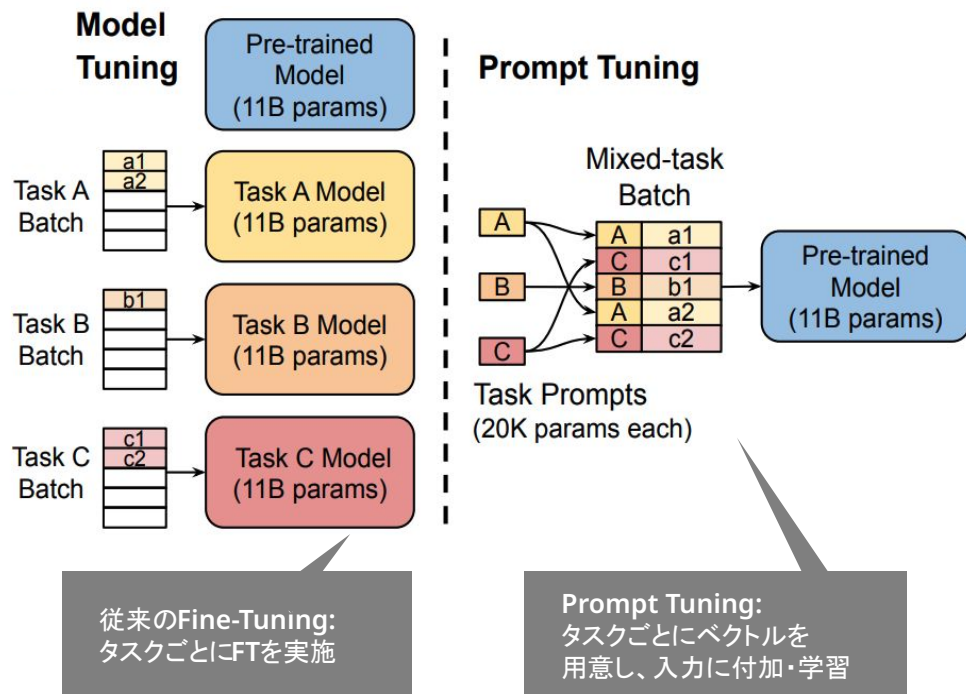
- Full-FTに対し 10^{-1} から 10^{-2} ほど小さい訓練パラメータ数で、Full-FTと同等の精度 (左図)
- Adapterのみ保存すればよく、柔軟に付け替え対応が可能

- Cons

- Adapterが追加されることで、推論にオーバーヘッドが発生

4. Parameter Efficient Fine-Tuning

② Soft-Prompt型 | Prompt Tuning



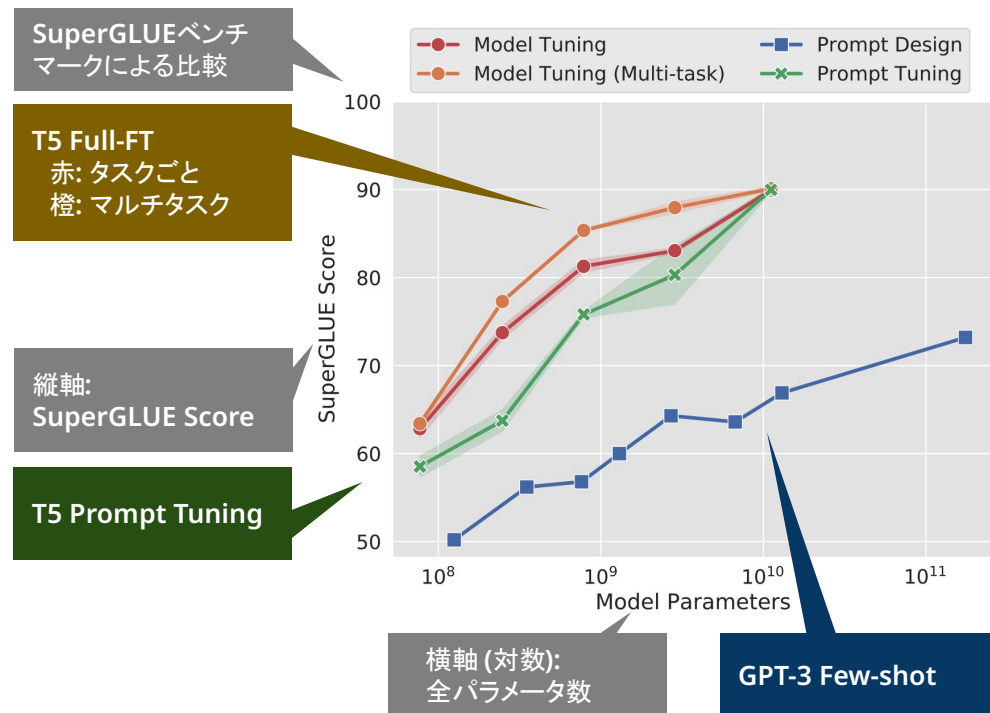
- 各タスクに対応したベクトル (**Soft Prompt**) を入力系列に付加し、そのパラメータを学習
- **Soft Prompt** は、文章の形で設計されたプロンプト (Hard Prompt) に対する呼び方・考え方
- つまり、各タスクごとに特化したプロンプトエンジニアリングを学習していると捉えることが可能

```
def soft_prompted_model(input_ids):
    x = Embed(input_ids)
    x = concat([soft_prompt, x], dim=seq)
    return model(x)
```

Lester, Brian, Rami Al-Rfou, and Noah Constant. "The power of scale for parameter-efficient prompt tuning." arXiv preprint arXiv:2104.08691 (2021). より引用し、一部改変

4. Parameter Efficient Fine-Tuning

2 Soft-Prompt型 | Prompt Tuning



Lester, Brian, Rami Al-Rfou, and Noah Constant. "The power of scale for parameter-efficient prompt tuning." arXiv preprint arXiv:2104.08691 (2021). より引用し、一部改変

- Pros

- モデルサイズが大きい場合、Prompt Tuning は Full-FT と同等の精度 (左図)

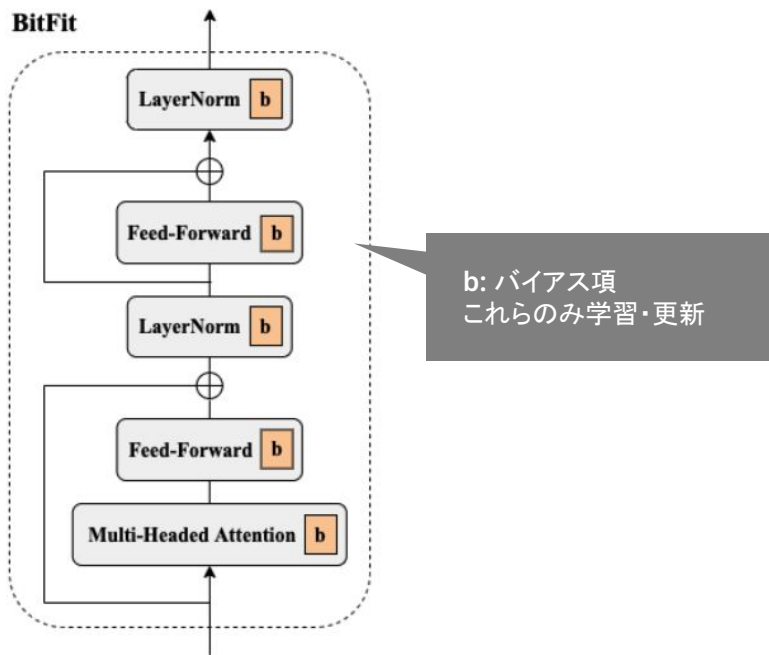
- T5-XXL (11B)で Soft Prompt の長さを100とすると、訓練するパラメータ数は $4096 * 100$
これはFull-FTの0.007%に相当

- Cons

- Soft Prompt が入力系列を圧迫
- プロンプトエンジニアリングの拡張として捉えると、解釈性に欠けた結果となっている

4. Parameter Efficient Fine-Tuning

3 Selective型 | BitFit (2021)



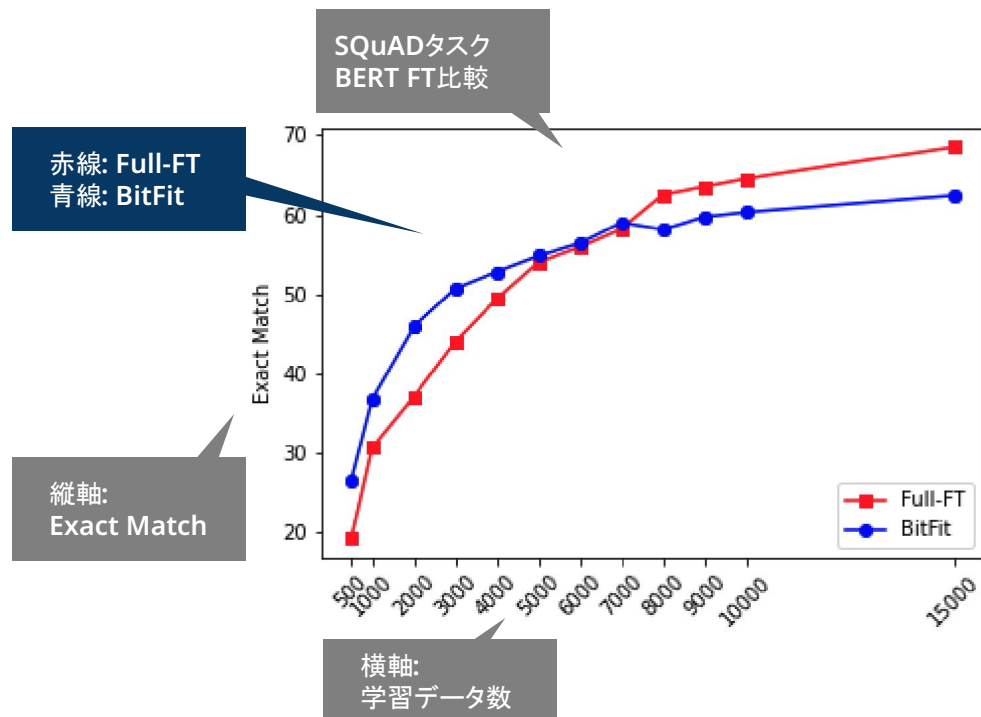
Zaken, Elad Ben, Shauli Ravfogel, and Yoav Goldberg. "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models." arXiv preprint arXiv:2106.10199 (2021). より引用し、一部改変

- Transformerの各モジュールに含まれる、バイアス項のみにについて学習・更新を実施
- 具体的に、以下に含まれているバイアス項が該当
 - Attention
 - Feed-Forward Network
 - Layer Normalization

```
params = (p for n, p
           in model.named_parameters()
           if "bias" in n)
optimizer = Optimizer(params)
```

4. Parameter Efficient Fine-Tuning

3 Selective型 | BitFit (2021)



Zaken, Elad Ben, Shauli Ravfogel, and Yoav Goldberg. arXiv preprint arXiv:2106.10199 (2021). より引用し、一部改変

- Pros

- 学習データ数が小さい領域では、BitFitがFull-FTよりも高い精度を示した (左図)
- BERT(Base)モデルで、BitFitによる訓練パラメータ数は、Full-FTに対して0.1%ほど

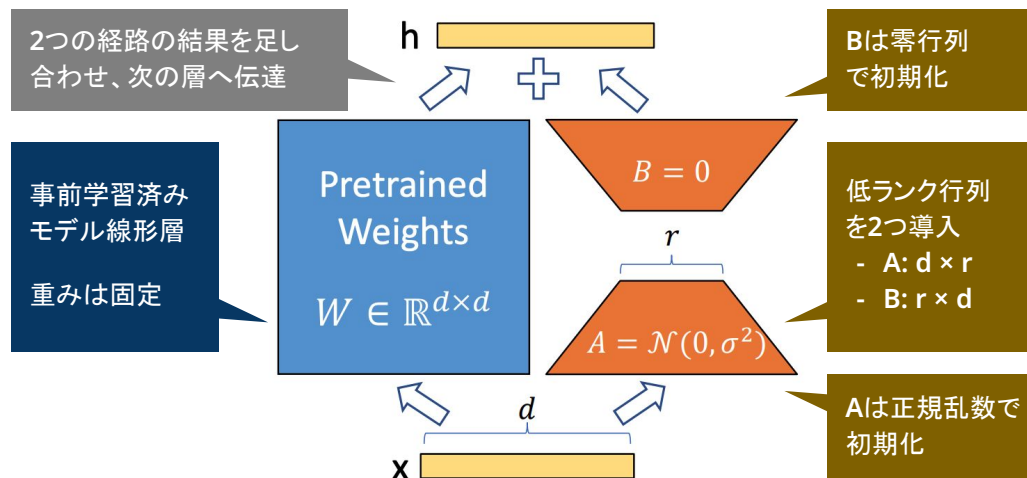
- Cons

- GPT-3 といった、より大規模なモデルでは、Full-FT や他のPEFT手法よりも精度が劣る※

※ Hu, Edward J., et al. "Lora: Low-rank adaptation of large language models." arXiv preprint arXiv:2106.09685 (2021).

4. Parameter Efficient Fine-Tuning

4 Reparametrization型 | LoRA (2021)



※ Hu, Edward J., et al. "Lora: Low-rank adaptation of large language models." arXiv preprint arXiv:2106.09685 (2021). より引用し、一部改変

- Fine-Tuning によって更新された重み W は一般に、元の重み W_0 と増分重み ΔW の和として表せる

$$W = W_0 + \Delta W$$

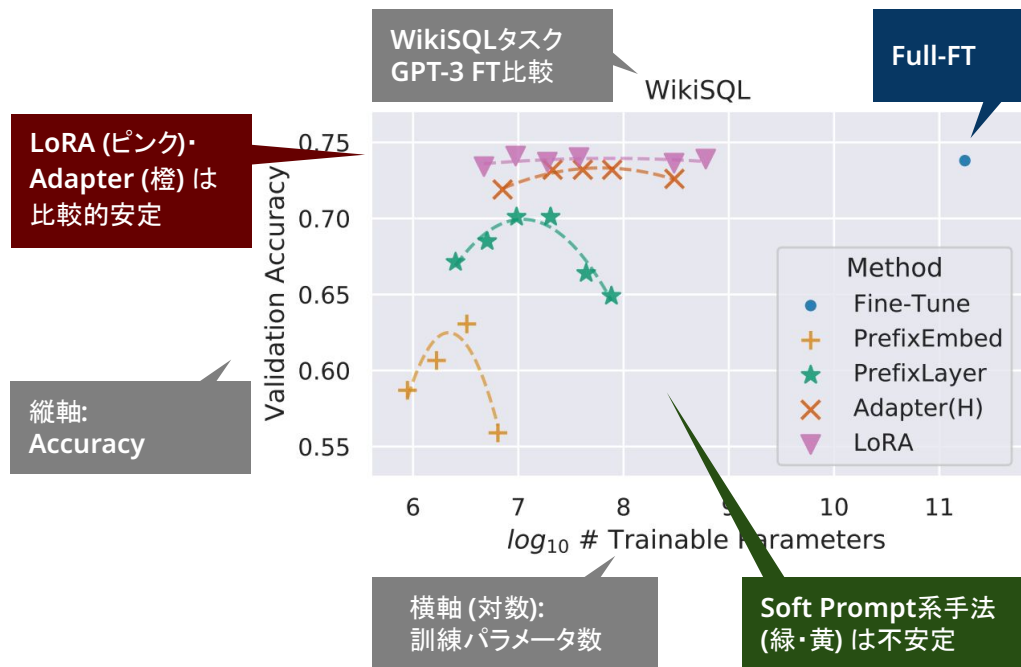
- LoRAでは、この増分重み ΔW を2つの低ランク行列 A, B の積とし、それらについて学習を実施

$$\Delta W = BA$$

```
def lora_linear(x):
    h = x @ W # regular linear
    h += x @ W_A @ W_B # low-rank update
    return scale * h
```

4. Parameter Efficient Fine-Tuning

4 Reparametrization型 | LoRA (2021)



※ Hu, Edward J., et al. "Lora: Low-rank adaptation of large language models." arXiv preprint arXiv:2106.09685 (2021). より引用し、一部改変

- Pros

- Full-FTに対し 10^{-2} から 10^{-4} ほど小さい訓練パラメータ数で、Full-FTと同等の精度 (左図)
- 推論時には、得られた重みを元の重みに予め足しておけば、オーバーヘッドが生じない

- Cons

- 特に難易度の高いタスク (例 GSM8k, 数学的推論) で、Full-FTに対しては著しい性能の劣後が生じる※

※ anyscale "[Fine-Tuning LLMs: LoRA or Full-Parameter? An in-depth Analysis with Llama 2](#)"

4. Parameter Efficient Fine-Tuning

4 Reparametrization型 | LoRA (2021)

- Q. 訓練パラメータ数を一定としたとき、LoRAを適用する層の種類をより増やすべきか、ランク r をより大きく取るべきか
- A. LoRAを適用する層の種類を増やした方が、ランク r が小さく なくても、より高い性能となることが示された
- ※ LoRA論文ではAttentionモジュール内を適用対象としたが、その後の研究では他の線形層も対象とすることで性能が改善

Weight Type

- q: Query
- k: Key
- v: Value projection
- o: Output projection

訓練パラメータ数を18Mに固定

	# of Trainable Parameters = 18M						
Weight Type Rank r	W_q 8	W_k 8	W_v 8	W_o 8	W_q, W_k 4	W_q, W_v 4	W_q, W_k, W_v, W_o 2
WikiSQL ($\pm 0.5\%$)	70.4	70.0	73.0	73.2	71.4	73.7	73.7
MultiNLI ($\pm 0.1\%$)	91.0	90.8	91.0	91.3	91.3	91.3	91.7

※ Hu, Edward J., et al. "Lora: Low-rank adaptation of large language models." arXiv preprint arXiv:2106.09685 (2021). より引用し、一部改変

4. Parameter Efficient Fine-Tuning

4 Reparametrization型 | LoRA (2021)

- Q. LoRAを適用する層の種類を固定して考える場合に、ランク r はどの程度の値を設定する必要があるか
- A. LoRAのランク r は、2から8の範囲で性能が高い結果
- ※ (タスク依存だが) ランク1で十分な性能が出る場合も

	Weight Type	$r = 1$	$r = 2$	$r = 4$	$r = 8$	$r = 64$
WikiSQL($\pm 0.5\%$)	W_q	68.8	69.6	70.5	70.4	70.0
	W_q, W_v	73.4	73.3	73.7	73.8	73.5
	W_q, W_k, W_v, W_o	74.1	73.7	74.0	74.0	73.9
MultiNLI ($\pm 0.1\%$)	W_q	90.7	90.9	91.1	90.7	90.7
	W_q, W_v	91.3	91.4	91.3	91.6	91.4
	W_q, W_k, W_v, W_o	91.2	91.7	91.7	91.5	91.4

※ Hu, Edward J., et al. "Lora: Low-rank adaptation of large language models." arXiv preprint arXiv:2106.09685 (2021). より引用し、一部改変

4. Parameter Efficient Fine-Tuning

4 Reparametrization型 | LoRAの派生アプローチ

	目的	手法
QLoRA	さらに少ない計算リソースによっても LoRA による Fine-Tuning を実現したい <small>Dettmers, Tim, et al. "Qlora: Efficient finetuning of quantized llms." arXiv preprint arXiv:2305.14314 (2023).</small>	LoRAに4ビット量子化等のテクニックを適用し、メモリ使用量を大幅に削減
AdaLoRA	LoRA において、全ての層のランクが 単一の値に制限されている問題の解決 <small>Zhang, Qingru, et al. "Adaptive budget allocation for parameter-efficient fine-tuning." arXiv preprint arXiv:2303.10512 (2023).</small>	増分重みの特異値分解に基づいて、層ごとのランクを適応的に変化させる
ReLoRA	LoRA を用いることで、少ない計算リソースで Pre-Training も実現したい <small>Lialin, Vladislav, et al. "Stack More Layers Differently: High-Rank Training Through Low-Rank Updates." arXiv preprint arXiv:2307.05695 (2023).</small>	学習率を変えながら LoRA 適用・初期化を繰り返し、高ランクの学習を実現

Parameter Efficient Fine-Tuning

目次

- 01 Day5 イン트로ダクション
- 02 LLM Fine-Tuning
- 03 Instruction Tuning
- 04 Parameter Efficient Fine-Tuning
- 05 Day5 まとめ**

5. Day5 まとめ

大規模言語モデル講座 Day5 の目標

Goal 1

大規模言語モデルの典型的な訓練フローにおいて、**Fine-Tuning** が **Pre-Training** (Day3-4) や **RLHF** (Day6) に対してどう位置付けられるか説明できる

Goal 2

大規模言語モデルの **Fine-Tuning** において、特に重要なアプローチである **Instruction Tuning** や **PEFT** が既存手法に対してどう位置付けられるか説明できる

Goal 3

Instruction Tuning および **PEFT** について、その理論や目的を十分に理解した上で、実際にそれらを実装し、大規模言語モデルの性能改善を実現できる