量子化 bit 幅の異なる基盤モデルに対する Adapter の転移性を活用した Low-Rank Adaptation

神田 悠斗 [1] 波多野 賢治 [2]
[1] 同志社大学大学院 文化情報学研究科 [2] 同志社大学 文化情報学部
{kanda@mil,khatano@mail}.doshisha.ac.jp

概要

大規模言語モデルは，重みの勾配や最適化状態を保持するため，推論時よりも訓練時に要求される GPU メモリ量が多いことが知られている．したがって，例えば同一の計算資源上で基盤モデルの訓練と推論を行う際には，基盤モデルのサイズは訓練時の資源制約から決定されるため，推論時には GPU メモリの余剰が発生する．本研究ではこの推論時のメモリの余剰に着目し，これを活用して LoRA モデルの性能を向上させる新たな 量子化-LoRA フレームワークとして，Post LoRA Restoration（PLR）を提案する．評価実験の結果，訓練時の計算コストはそのままに，PLR による最大 12 倍の精度向上が確認できた．

1 はじめに

近年の Large Language Model（**LLM**）のパラメータサイズの大規模化は，モデルの性能を飛躍的に向上させた一方で，計算コストの増加という新たな課題を生んでいる [1, 2]．その結果，LLM の運用に伴うインフラコストが上昇し，十分な計算資源を持たない組織やユーザーが LLM を構築・運用することが困難となっている．

言語モデルは学習外の知識に対してはその性能を十分に発揮できないことが知られており [3]，未学習のドメインに対して LLM を適応させる Fine-Tuning（**FT**）が必要となるが，これには膨大な計算資源が求められる．こうした背景から，計算資源に制限がある状況下で LLM の FT と推論を行うために，Low-Rank Adaptation（**LoRA**）と Quantization（量子化）を組み合わせた技術が研究されてきた（[4, 5, 6, 7]）．LoRA は，基盤モデルの FT の際にモデルの重みを直接更新するのではなく，代わりに Adapter と呼ばれる低ランクな補助行列を LLM の各層にマージし，それを訓練することで計算コストを削減する手法である．一方で量子化は，通常は 16-bit のデータ型で表現される LLM の重みを 8-bit や 4-bit，あるいはそれ以下に圧縮し，モデル自体の軽量化を図る技術である．この二つの技術の併用により，LLM の訓練・推論時の GPU メモリを大幅に削減できるため，少ない計算資源で高精度なモデルを構築することが可能となる．

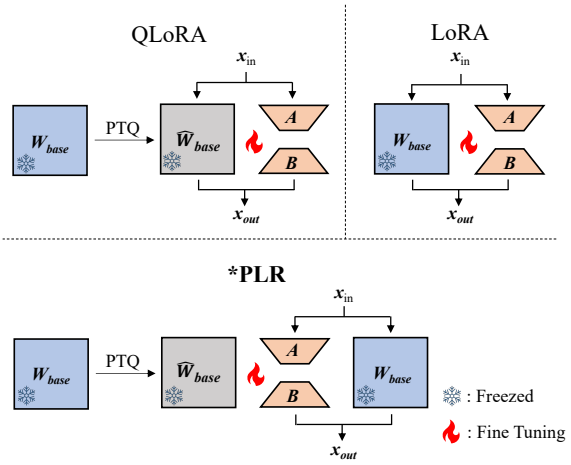


図 1: 提案手法（PLR）および従来手法（LoRA, QLoRA）の概念図．量子化後のモデルに対し Adapter の学習を行う QLoRA に対し，PLR では高 bit 幅へ復元された基盤モデルに QLoRA で学習した Adapter がマージされる．

既存の量子化-LoRA フレームワークでは，訓練時と推論時の二つのフェーズで利用可能な計算資源（特に GPU メモリ）を比較し，よりメモリ制約の厳しいフェーズに合わせて基盤モデルのパラメータサイズが選択される．しかし，このプロセスにより，もう一方のフェーズでは計算資源の余剰が生じ，計算資源を有効に活用できていない可能性がある．

このような状況が想定される具体例の一つに，同一の計算資源上で LoRA の訓練と推論を行うシナリオが考えられる．一般的に，言語モデルの訓練時に

は，訓練の対象となるパラメータの勾配とその最適化状態を保持する必要があるため，推論時よりも多くの GPU メモリが必要となる．したがって，訓練時に要求されるメモリ量に合わせて基盤モデルを採用した際には，推論時に GPU メモリの余剰が生じる．

そこで本研究では，訓練時と推論時に要求される計算コストのギャップから生じる GPU メモリの余剰を有効に活用するためのフレームワークとして，Post LoRA Restoration（**PLR**）を提案する．PLR は，訓練時の資源制約から生まれる推論時のメモリの余剰を活用し，量子化された基盤モデルの復元を行う．これにより，より重みの精度が高い基盤モデルを推論時に活用することが可能となり，言語モデルの各タスクに対する処理性能の向上が期待できる．

2 関連研究

Dettmers et al. は，量子化と LoRA を組み合わせることで FT に関わる計算コストを大幅に削減する，QLoRA を提案している [4]．QLoRA は，まず初めに基盤モデルの重みを低い bit 幅へ量子化し，その後量子化された基盤モデル上で LoRA を学習する．このフレームワークにより，量子化前の重みの精度が高い状態では計算資源の制約から LoRA が不可能であった，パラメータサイズが大きく性能の高い基盤モデルに対して LoRA の適用が可能となるため，低計算資源下でも性能の高い言語モデルの活用が可能となる．

3 提案手法

PLR では，推論時のメモリ余剰に着目して，量子化-LoRA モデルを学習したのちに 量子化-LoRA の基盤モデルの重みの精度をより高い bit 幅へ復元する．

PLR の具体的な流れは，以下の通りである．基盤モデルの重みを $\boldsymbol{W}_{\mathbf{base}}$，LoRA adapter の重みを $\boldsymbol{\Delta W}_{\mathbf{LoRA}}$ とすると，LoRA 適用後の各層の重み $\boldsymbol{W}$ は，

$$\boldsymbol{W} = \boldsymbol{W}_{\mathbf{base}} + \boldsymbol{\Delta W}_{\mathbf{LoRA}} \tag{1}$$

で表される．量子化-LoRA 手法では，基盤モデルの重みを量子化した状態で LoRA Adapter を学習する．したがって，量子化後の重み $\hat{\boldsymbol{W}}$ は，

$$\hat{\boldsymbol{W}}_{\mathbf{base}} = Quantize(\boldsymbol{W}_{\mathbf{base}}) \tag{2}$$

によって得られ，これを基盤モデルの重み $\boldsymbol{W}$ と置き換えた，

$$\hat{\boldsymbol{W}} = \hat{\boldsymbol{W}}_{\mathbf{base}} + \boldsymbol{\Delta W}_{\mathbf{LoRA}} \tag{3}$$

における，$\boldsymbol{\Delta W}_{\mathbf{LoRA}}$ を学習する．PLR では，学習の後に $\hat{\boldsymbol{W}}_{\mathbf{base}}$ を 量子化前の重み $\boldsymbol{W}_{\mathbf{base}}$ に復元することによって，最終的な各層の重み，

$$\boldsymbol{W}_{\mathbf{PLR}} = \boldsymbol{W}_{\mathbf{base}} + \boldsymbol{\Delta W}_{\mathbf{LoRA}} \tag{4}$$

を得る．

一般的に，LLM の量子化は低い bit 幅であるほど近似値に丸め込まれた際に生じる量子化誤差が大きく，性能が低下することが知られている．したがって，PLR を適用することでより bit 幅が大きく性能の高い基盤モデルが推論時に活用できるため，量子化-LoRA モデルの性能向上が期待できる．

4 評価実験

本節では，提案手法の有効性を示すために行う評価実験について紹介する．評価実験では，既存の量子化-LoRA 手法に対して提案手法である PLR を適用した際に，評価用の各タスクの正解率が向上するかを検証する．

実験に使用する基盤モデルは，Llama3 ファミリーの，Llama3.2-1B，Llama3.2-3B，Llama3-8B（[2]）の 3 種類とする．なお，モデルのパラメータサイズは，それぞれ 1B（10 億），3B（30 億），8B（80 億）である．

次に，LoRA の評価対象とするタスクには，Grade School Math 8K （**GSM8K**） [8] および SQL Create Context（**SCC**）[9] を使用する．GSM8K は小学校程度の簡単な算数の問題を解くタスクであり，SCC は自然言語で表されたデータベースに対する操作から SQL 文を生成するタスクである．GSM8K は，約 7,500 件の訓練データと 約 1,300 件の評価データが，SCC は 約 78,600 件 の訓練データが収録されている．したがって，GSM8K は訓練データの 1 割を検証データに，SCC は同じく 1 割を検証データ，もう 1 割を評価データに分割し，モデルの訓練と評価に用いる．

GSM8K には算数の設問とその解答が，SCC にはデータベースのスキーマとのデータベースに関する質問，および回答を導くための SQL 文がそれぞれ収録されている．これらを用いて，FT の教師デー

表 1: QLoRA に対して PLR を適用した際の各タスクの正解率（%）

Datasets	Models	16-bit LoRA	8-bit QLoRA		4-bit QLoRA			3-bit QLoRA				2-bit QLoRA				
			QLoRA	**PLR16**	QLoRA	**PLR8**	**PLR16**	QLoRA	**PLR4**	**PLR8**	**PLR16**	QLoRA	**PLR3**	**PLR4**	**PLR8**	**PLR16**
GSM8k	Llama3-1B	21.60	21.83	**21.91**	**17.43**	15.39	16.60	11.22	**15.31**	**15.76**	**16.52**	3.18	3.11	**3.49**	2.80	2.43
	Llama3-3B	42.91	**41.77**	40.71	**39.80**	36.61	37.90	29.56	**31.53**	**36.84**	**37.30**	4.16	**6.60**	**15.85**	**10.99**	**15.62**
	Llama3-8B	59.66	59.28	**61.25**	54.43	**56.10**	**57.08**	50.34	**52.31**	**58.15**	**59.59**	5.15	**18.27**	**30.09**	**35.86**	**34.34**
	Avg.	41.39	40.96	**41.29**	**37.22**	36.03	37.19	30.37	**33.05**	**36.92**	**37.80**	4.16	**9.32**	**16.47**	**16.55**	**17.46**
SCC	Llama3-1B	69.21	60.23	**63.29**	**35.66**	26.19	31.56	23.50	20.67	**36.16**	**37.18**	0.00	0.00	0.00	**0.08**	**1.49**
	Llama3-3B	72.30	81.25	**82.80**	**83.60**	79.38	80.41	66.52	66.12	**75.22**	**74.83**	3.20	**31.91**	**40.02**	**36.81**	**37.21**
	Llama3-8B	84.96	85.12	**85.15**	82.46	**83.26**	**84.43**	48.91	**74.93**	**78.53**	**80.33**	4.00	**38.91**	**38.21**	**35.51**	**36.51**
	Avg.	75.49	75.53	**77.08**	**67.24**	62.94	65.47	46.31	**53.91**	**63.30**	**64.11**	2.40	**23.61**	**26.07**	**24.13**	**25.07**
Avg.	Llama3-1B	45.40	41.03	**42.60**	**26.55**	20.79	24.08	17.36	**17.99**	**25.96**	**26.85**	1.59	1.55	**1.74**	1.44	**1.96**
	Llama3-3B	57.61	61.51	**61.76**	**61.70**	58.00	59.16	48.04	**48.82**	**56.03**	**56.06**	3.68	**19.25**	**27.93**	**23.90**	**26.41**
	Llama3-8B	72.31	72.20	**73.20**	68.45	**69.68**	**70.76**	49.63	**63.62**	**68.34**	**69.96**	4.58	**28.59**	**34.15**	**35.69**	**35.42**
	Avg.	58.44	58.25	**59.19**	**52.23**	49.49	51.33	38.34	**43.48**	**50.11**	**50.96**	3.28	**16.47**	**21.27**	**20.34**	**21.27**

タは以下のように作成する．

GSM8K の教師データ

Answer the following math question.
Question:{QUESTION}
Answer:{ANSWER}

SCC の教師データ

Write SQLite query to answer the following question given the database schema.
Schema: {SCHEMA}
Question: {QUESTION}
Answer: {ANSWER}

FT したモデルの評価では, 各データの {ANSWER} を取り除いたものをプロンプトとして入力した際に LLM が適切に回答できた件数から正解率を算出する．GSM8K で求められる最終的な解答は整数であるが，LLM の出力には途中式などが含まれるため，出力から全ての整数を正規表現で抽出し，最初または最後に出現した整数と正解が一致している場合のみを正答としてカウントする．また，SCC においては，出力内に正解と完全一致する文字列が出現した場合を正答としてカウントする．

次に，LoRA の設定について述べる．提案手法の比較対象となる 量子化-LoRA 手法には QLoRA[4] を使用する．LoRA および QLoRA では，Adapter の Rank r と α は，すべてのモデルで 16 に設定し，LoRA を適用する LLM の層は Attention 層の q_proj, k_proj，v_proj，o_proj の四つの層とする．学習率は 2e-4，バッチサイズは 32 とし，8 エポック学習した内，検証データに対する Loss が最も低いエポックの重みを評価に使用する，QLoRA における基盤モデルの量子化には，8bit および 4-bit への量子化には単純な丸め込み手法である Round-to-Nearest を，3-bit および 2-bit では GPTQ[10] を使用した．GPTQ は，重みの量子化後にキャリブレーションデータを用いて補正を行い，量子化誤差を最小限に抑える手法である．キャリブレーションデータには，訓練データから無作為抽出した 500 件を使用する．

これらの処理は，Python 3.11 上で実装し，GPU コンピューティングには NVIDIA A6000 および Cuda 11.8 を用いる．実験で用いる Python の主要なモジュールとしては，PyTorch 2.5.0+cu118, Transformers=4.47.0，Peft=0.14.0 を使用する．実験で使用する Numpy，Random，Pytorch モジュールの乱数の seed 値は，すべて 42 とした．

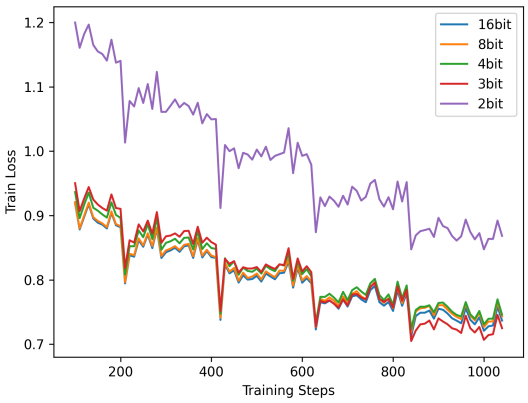
これらの実験設定で，LoRA および QLoRA を学習し，QLoRA に対して PLR を適用した際の各タスクに対する正解率を表 1 に示す．なお，表中の PLR16 や PLR8 は，QLoRA の学習後にそれぞれ 16-bit, 8-bit へ基盤モデルを復元することを指す．

5 考察

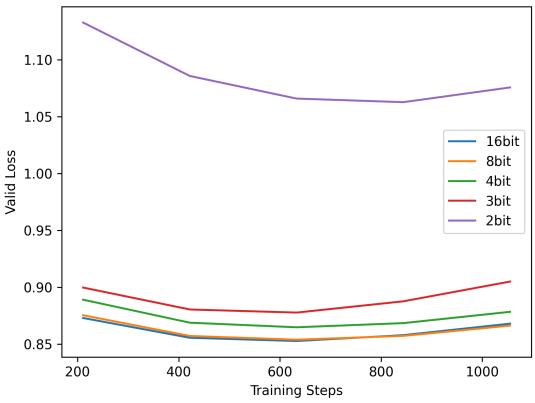
本節では，評価実験の結果に対する考察を述べる．

5.1 bit 幅の異なる基盤モデルに対する Adapter の転移性

表 1 を見ると，ほぼ全ての設定下で PLR が QLoRA に勝っており，提案手法の有効性が確認できた．モデルサイズごとに確認すると，llama3-8B は全ての bit 幅で精度向上が認められており，モデルサイズが大きく賢いモデルほど PLR の恩恵を受けられるこ



(a) 訓練データに対する Loss の推移



(b) 検証データに対する Loss の推移

図 2: GSM8K に対する Llama3-8B の LoRA（16-bit）と QLoRA（8,4,3,2-bit）におけるモデルの Loss の推移

とを示唆している．

特に，GSM8K の 3-bit QLoRA に対する PLR16 では，通常の 16-bit LoRA と比較して僅か 0.1% の性能劣化にとどめている．これは，低 bit 幅においてタスクに対する解法を LoRA Adapter が学習可能であることを示しており，かつその性能を異なる bit 幅の基盤モデルに対しても転移可能であることを表している．さらに，8-bit の PLR16 においては，16-bit LoRA を上回る精度を達成している．これは，量子化が正則化効果をもたらし，Adapter の学習が効率的に進んだことに起因すると考えられる．

一方で，llama3-1B および llama3-3B では，いくつかの設定下で QLoRA に劣る結果となっており，特に 4-bit QLoRA に対しては，二つのデータセットで共に性能を低下させている．これは，4-bit QLoRA に用いた量子化手法が GPTQ よりも 量子化誤差の大きい RTN であり，4-bit の量子化誤差に対して Adapter が過学習したことで転移性が十分に発揮されなかったことが要因であると考えられる．したがって，4-bit および 8-bit においても量子化手法に GPTQ を用いた評価実験を行う必要がある．

5.2 2-bit 幅における学習の失敗

8-bit から 3-bit までの量子化では，16-bit の基盤モデルを用いた LoRA と比較して、わずかな性能劣化で学習が可能であったのに対し，2-bit QLoRA では大幅な性能劣化が確認され，PLR の適用でも性能が回復しきれていない．これは，2-bit への量子化で発生した量子化誤差に Adapter が適応できず，学習が失敗したことが要因であると解釈できる．実際に，LoRA および QLoRA の学習過程の Loss を示した図 2 を見ると，2-bit の QLoRA の Loss は他の bit 幅と比べて 2 割程度大きく，量子化誤差の影響の大きさが窺える．この現象に対しては，QLoRA の改良手法である QA-LoRA[5] や LoftQ[6] の使用で対処できる可能性があるため，2-bit 幅における PLR の有効性を検証するためには，これらの手法を用いた追加の評価実験を行う必要がある．

5.3 本研究の課題と今後の展開

提案手法は訓練時と推論時にメモリの差があることを利用した手法であるが，実際に推論時にはどれほどの余剰が生まれるかまでは調査できていない．また，余剰がない場合でも，オフロード技術等を用いることで，GPU メモリ以上のサイズの基盤モデルを推論時に活用することが可能であるが，その際には推論速度の低下が懸念される．PLR の適用範囲とオフロードを併用した際の推論速度についての詳細な調査は，今後の課題としたい．

6 おわりに

本研究では，訓練時と推論時の計算コストのギャップから生まれる GPU メモリの余剰に着目し，bit 幅の異なる基盤モデルに対する Adapter の転移性を活用した量子化-LoRA フレームワークである PLR を提案した．評価実験の結果，特にモデルサイズが大きく賢いモデルほど PLR の有効性が確認でき，大規模化する近年の言語モデルに対して LoRA の訓練時のコストはそのままに高精度化できる本手法は，極めて有望であると言える．

謝辞

本研究は JSPS 科研費 JP23K21726 および 同志社大学大学院文化情報学研究科研究推進補助金の助成を受けたものである.

参考文献

[1] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. **arXiv preprint arXiv:2001.08361**, 2020.

[2] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. **arXiv preprint arXiv:2407.21783**, 2024.

[3] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 8342–8360, 2020.

[4] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: efficient finetuning of quantized llms. In **Proceedings of the 37th International Conference on Neural Information Processing Systems**, pp. 10088–10115, 2023.

[5] Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, XIAOPENG ZHANG, and Qi Tian. Qa-lora: Quantization-aware low-rank adaptation of large language models. In **The Twelfth International Conference on Learning Representations**, 2024.

[6] Yixiao Li, Yifan Yu, Chen Liang, Nikos Karampatziakis, Pengcheng He, Weizhu Chen, and Tuo Zhao. Loftq: Lora-fine-tuning-aware quantization for large language models. In **The Twelfth International Conference on Learning Representations**, 2024.

[7] Hyesung Jeon, Yulhwa Kim, and Jae joon Kim. L4q: Parameter efficient quantization-aware fine-tuning on large language models, 2024.

[8] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. **arXiv preprint arXiv:2110.14168**, 2021.

[9] b mc2. sql-create-context dataset, 2023. This dataset was created by modifying data from the following sources: [11, 12].

[10] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. In **The Eleventh International Conference on Learning Representations**, 2023.

[11] Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. **CoRR**, Vol. abs/1709.00103, , 2017.

[12] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. **arXiv preprint arXiv:1809.08887**, 2018.