Transformer LLM における層単位の FFN 層の重要度検証

池田航 [1] 矢野一樹 [1] 高橋良允 [1] 李宰成 [1] 柴田圭悟 [1] 鈴木潤 [1,2,3]
[1] 東北大学 [2] 理化学研究所 [3] 国立情報学研究所
ikeda.wataru.q5@dc.tohoku.ac.jp

概要

Transformer に基づく大規模言語モデル (LLM) の構成要素の一つであるフィードフォワードネットワーク (FFN) に着目し，モデル内の配置場所に依存した重要度を検証する．具体的には，モデル全体のパラメータ数を維持したまま，一部の連続する層で FFN の中間次元を拡大し，残りの層から FFN を除去したモデル構成を用いて標準的なタスク性能を比較する．複数のモデルサイズで評価を行った結果，全層数の 70 – 90%の連続した中間から後方の層に FFN を集中配置することで，複数の下流タスクでベースラインの性能を上回った．この結果から FNN は入力に近い層より中間から後方の層で特に重要であると示唆される結果が得られた．

1 はじめに

Transformer [1] に基づく大規模言語 モデル（Large Language Model: LLM) では，各モデルによって詳細設計は異なるが，自己注意機構とフィードフォーワードネットワーク (Feed-Forward Network: FFN) の二つを主な構成要素として一つの Transformer 層を構成しているモデルが多い [1, 2]．図 1(a) に Transformer 層を図示する[1)]．特に Pre-LN [3] と呼ばれる Transformer のモデル構成の場合，これら 2 つを 1 層として計算された結果を層の数だけ入力層であるトークンの埋め込みベクトルに加算し，最終的な隠れ状態ベクトルを算出する計算手順になる．

一般論として，自己注意機構は，主にトークンの埋め込みから得られた情報の混ぜ合わせを担っており，FFN は，学習データ内にある知識などを記憶することが主な役割と説明されることが多い [4, 5, 6][2)]．知識が FFN 層に埋め込まれているという仮説が正しいとした場合，FFN が知識を獲得する上で最もよい形式なのか，実際に Transformer 内の複数ある層の中でどのあたりに知識が埋め込まれるのかなど未解明な事象も多い．そこで，本研究では FFN に着目し，FFN を削除したモデルや FFN を大きくした設定など，いくつかの通常とは違う設定で LLM の事前学習を実施することで，FFN の役割や機能の一部を検証する．

1) 本稿では，自己注意機構と FFN をまとめた Transformer 層を略して「層」と表記する．また，層正規化は本研究では重要な要素ではないので，簡略化のため説明から除外する．
2) 現在は，それ以外の様々な機能や効果があることが検証により示されている [7].

2 関連研究

Transformer に基づく LLM における FFN の役割や機能を検証している論文は多く存在している．文献 [4, 5, 6] では，FFN が知識の記憶装置として機能することを示し，特定のニューロンが事実的知識の表現や想起に重要な役割を果たしていることを明らかにした．一方，文献 [7] では，分析を通じて，FFN が層正規化と合わせて入力の文脈化に寄与するという FFN の役割に関する新たな解釈を与えた．

このように，幾つもの重要な知見が得られているが，これらの知見は，事前学習済み LLM に対する分析結果から得られたものであり，標準的なモデル構成に限定される．本稿では，Transformer 各層の FFN の一部を削除したり，次元数を増やしたりするなど，モデル構成そのものを変化させた際の影響を検証するという点で，従来研究とは違う検証方法となっている．また，従来と異なる検証方法を採用することで，FFN の機能や役割に関する新たな知見を得ることを試みている．

3 モデル設計と層の構成

3.1 ベースラインモデル

本研究ではベースラインモデルとして, LLaMA [2] で提案されたモデル設計を用いて実験を行う．LLaMA では，FFN には SwiGLU 活性化関数 [8] が採用されており，FFN は入力ベクトル $x \in \mathbb{R}^d$ を受け取り，内部で中間表現の次元数 d_f まで拡張して処

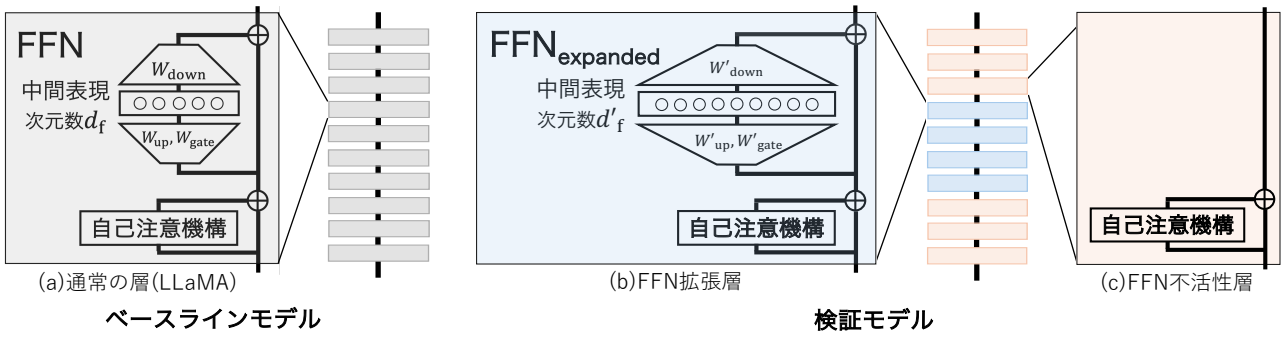

図 1　ベースラインモデルと検証モデルの異なる層の構成 通常の LLaMA の層の積み重ねであるベースラインモデルに対し (上段左)，検証モデルでは一部の層の FFN の中間表現の次元を拡張し ((b)FFN 拡張層)，残りの層から FFN を除去する ((c)FFN 不活性層) ことにより (上段右)，ベースラインモデルに対して全体のパラメータ数は維持しつつ FFN の計算能力 (=パラメータ数) を特定の層に集中させたモデルを実現する．

理を行う（$W_{\text{gate}}, W_{\text{up}} \in \mathbb{R}^{d_f \times d}$，$W_{\text{down}} \in \mathbb{R}^{d \times d_f}$）：

$$\text{FFN}(x) = W_{\text{down}}(\text{Swish}(W_{\text{gate}}x) \otimes W_{\text{up}}x) \quad (1)$$

$$\text{Swish}(x) = x\sigma(x) \quad (2)$$

3.2　検証モデル

ベースラインモデルで採用した標準的な LLaMA モデルの層を **FFN 拡張層**，または，**FFN 不活性層**で置き換えたモデルを検証モデルとする．

FFN 拡張層は，標準的な自己注意機構に加え，拡張された中間次元数を持つ $\text{FFN}_{\text{expanded}}$ を配置する (図 1 (b) 参照)．ここで，$\text{FFN}_{\text{expanded}}$ は以下のように定義する：

$$\text{FFN}_{\text{expanded}}(x) = W'_{\text{down}}(\text{Swish}(W'_{\text{gate}}x) \times W'_{\text{up}}x) \quad (3)$$

ただし，$W'_{\text{gate}}, W'_{\text{up}} \in \mathbb{R}^{d'_f \times d}$，$W'_{\text{down}} \in \mathbb{R}^{d \times d'_f}$ であり，中間次元数 d'_f はベースラインの層の中間次元数 d_f より大きい値をとる．この FFN 拡張層の中間次元数 d'_f は，ベースラインのすべての層を FFN 拡張層か後述する FFN 不活性層に置き換えてもモデル全体のパラメータ数がほぼ一致するように決定する．

FFN 不活性層は，前述の標準的な LLaMA モデルの層から FFN を除去し，自己注意層のみとした層である（図 1 (c) 参照）．

本研究では，ベースラインモデルの層を，特定の配置で，これら FFN 拡張層または FFN 不活性層にて置き換えた検証モデルを設定し，FNN の位置に依存した重要度の検証実験に用いる．

4　実験

本研究では，前節で説明したベースラインモデルと検証モデルを事前学習した後に標準的なタスク性能を評価し，どの位置にある FFN を除去しても性能が劣化しないか，或いは，逆に性能が向上するかなどの挙動を調査し，その結果から位置に依存した FFN の重要度を検証する．

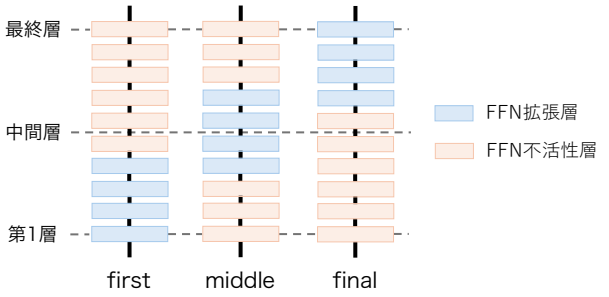

図 2　FFN 拡張層の異なる配置位置 検証モデルの FFN 拡張層を入力層に近い位置（first），中間層（middle），出力層に近い位置（final）のいずれかに配置し，各位置における効果を検証した．

4.1　ベースラインモデルの設定

本研究では，285M および 570M パラメータの 2 つの異なるサイズのベースラインモデルを用いる．285M パラメータモデルは 12 層で構成され，隠れ層の次元数 d は 1280 とする．570M パラメータモデルでは，285M と同じ隠れ層の次元数を維持しながら，層数を 24 層に拡張する．また，FFN における中間次元数 d_f は各モデルサイズのベースラインモデル共通であり，全ての層で 4480 とする．

4.2　検証モデルの設定

検証モデルにおいても，層数や隠れ層の次元数を含めて基本的なモデル設定はベースラインと共通とする．検証モデルでは，以下の 2 つの要素に従ってベースラインモデルの層を FFN 拡張層，または，FFN 不活性層に置き換える．

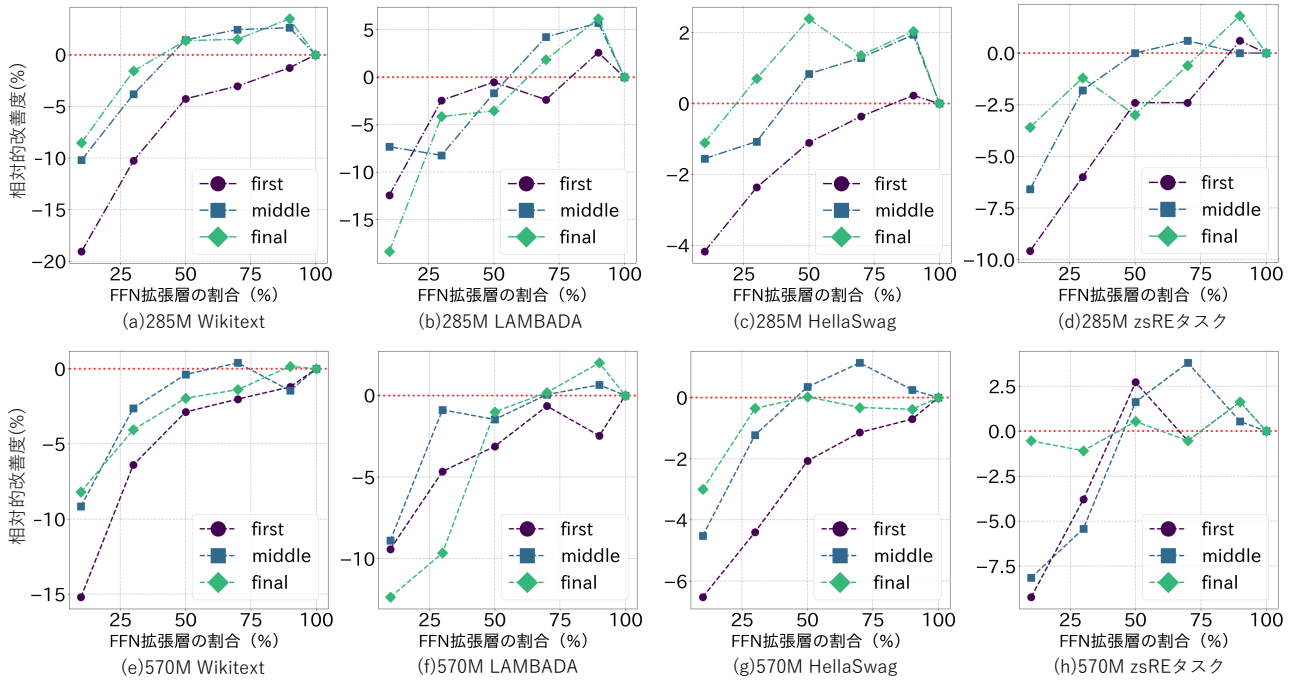

図 3　FFN 拡張層の割合に対するタスク性能の推移 上段 (a)〜(d)，下段 (e)〜(g) はそれぞれモデルサイズ 285M，570M にでの各タスクの評価結果．横軸：FFN 拡張層の割合，縦軸：ベースラインに対する相対的な改善度，異なる色のプロット：FFN 拡張層の異なる配置位置（`first`, `middle`, `final`），水平赤点線：相対的改善度が 0（ベースラインと同等の性能）．

1. **全層数に対する FFN 拡張層の割合** r%．ただし，$r \in \{10, 30, 50, 70, 90, 100\}$ を用いる．FFN 拡張層の総数は層数 L との積の小数点以下を切り捨てた値（$\lfloor rL/100 \rfloor$）となる．
2. **FFN 拡張層を配置する位置** {`fisrt`, `middle`, `final`}．図 2 に示す通り，FFN 拡張層の配置位置として `first`（第 1 層から後続する層に配置），`middle`（中間層（$L/2$ 層目）を起点に対称になるように配置），`final`（最終層から先行する層に配置）を設定した[3)]．

以上の 2 つの要素の組み合わせにより，計 15 種類（FFN 拡張層の割合 5 通り × 配置位置 3 通り）の配置パターンの検証モデルを設定する．

4.3　事前学習と評価

ベースラインモデルと検証モデルの事前学習には標準的な事前学習法を用いる[4)]．また，事前学習済みモデルの評価については下流タスク性能およびモデルの知識量の観点から評価を行う[5)]．モデルの知識量の評価に関しては，2 節で示した，FFN が知識を記憶するという先行研究の知見に基づき，Zero-Shot Relation Extraction (zsRE) データセット [9] を用いた知識量測定タスク (zsRE タスク) で評価する [10, 11][6)]．

各タスクの評価結果について，検証モデルとベースラインとの比較を簡単にするために以下で定義される，ベースラインに対する相対的な改善度（Relative Improvement:RI）を算出する:

$$\mathrm{RI}(m, T) = \frac{\mathrm{score}(m, T) - \mathrm{score}(\mathrm{baseline}, T)}{\mathrm{score}(\mathrm{baseline}, T)} \times 100[\%] \tag{4}$$

ここで，$\mathrm{score}(m, T)$ はモデル m のタスク T における評価スコア，$\mathrm{score}(\mathrm{baseline}, T)$ はベースラインモデルの評価スコアを表す．Perplexity など値が小さいほど性能が良いとされる評価指標については，符号を反転させている．相対的改善度が 0 の場合はベースラインと同等の性能，正の値はベースラインより性能が高く，負の値は性能が低いことを示す．

5　実験結果および考察

異なる配置パターン（FFN 拡張層の割合 × 配置位置[7)]）の検証モデルに対する各タスクの評価結果

3)　なお本研究では，連続する層に FFN 拡張層を配置する設定に限定して検証を行う．実際には飛び飛びの層に FFN 拡張層を配置するなど配置パターンは無数にあるが，これらの検証は今後の課題とする．
4)　事前学習の詳細設定については付録 A に記載
5)　具体的な評価タスクは付録 B に記載
6)　zsRE タスクの詳細は B.2 に記載
7)　FFN 拡張層の割合が 100%のモデルはベースラインモデルと一致する．図 3 中のこのモデルのプロットは FFN 拡張層の配置位置に関わらず全て赤点線上（ベースラインと同等の

表 1　各モデルサイズにおける上位 5 モデルの平均相対的改善度（RI） 平均 RI は 4 つの評価タスク（Wikitext, LAMBADA, HellaSwag, zsRE）の平均値を示す．

285M モデル		570M モデル	
モデル	平均 RI (%)	モデル	平均 RI (%)
final_90	+3.37	**middle_70**	+1.35
middle_90	+2.57	**final_90**	+0.85
middle_70	+2.14	middle_50	+0.03
final_70	+1.03	middle_90	−0.00
first_90	+0.54	final_70	−0.52

表 2　モデルサイズ 2B における検証モデルの相対的改善度の平均 全ての評価タスクの平均を算出．評価の詳細は C を参照．

	平均 RI（%）
middle_70	-0.71
final_90	**+1.06**

を図 3 に示す[8]．

5.1　FFN 拡張層の割合の影響

FFN 拡張層の割合が性能に与える影響に観察すると，FFN 拡張層の割合が低い（10 – 30%）モデルでは，図 3 (c) の final_30_285M[9] の条件を除くすべての下流タスクおよび知識量に関するタスクにおいて，ベースラインを大きく下回る性能を示した．つまり，FFN を一部の層に極端に集中させる設定はモデルの性能を著しく低下させる．

一方，FFN 拡張層の割合を増やすと性能は徐々に改善し，FFN 拡張層の割合が 70 – 90% の高い範囲では，すべてのタスクにおいて，少なくとも 1 つの検証モデルでベースラインを上回る性能が達成された．興味深い点として，ベースラインのように全ての層に均等に FFN を配置するよりも，一部の少数の層から FNN を除去し，残りの層（全層数の 70 – 90%）に緩やかに集中させるとタスク性能が顕著に向上する可能性があることが示唆された．

5.2　FFN 配置位置の影響

middle 設定は，図 3 (e) の middle_90_570M を除き，FFN 拡張層の割合が 70 – 90% の範囲ですべての評価タスクにおいてベースラインを上回る性能を示した．一方，first や final では，同じ FFN 拡張層の割合でもより多くのタスクでベースラインを下回った．また，特に下流タスク性能において（Wikitext, LAMBADA, HellaSwag），first のモデルが middle や final と比較して大きく性能が下回る傾向が顕著に見られた（図 3(a), (b), (c), (e), (f), (g)）．これらの観測から，中間部分から後半部分にある FFN がより効果的に機能していることが示唆され，逆に前半部分は効果が薄い可能性が示唆された．

5.3　各検証モデルの個別の評価

表 1 に各モデルサイズにおける平均の相対的改善度上位 5 モデル構成を示す．final_90（285M で 1 位（+3.37%），570M で 2 位（+0.85%））と middle_70（285M で 3 位（+2.14%），570M で 1 位（+1.35%））が全体の中で一貫して良い性能を示した．

5.4　より大きなモデルサイズでの再現性

285M および 570M の両方で安定して高い性能を示した前述の final_90 と middle_70 のモデル構成について，2B パラメータモデルの設定でも追加の検証を行った．表 2 にモデルサイズ 2B における評価結果を示す[10]．表 2 より明らかなように final_90 がすべての評価タスクを平均してベースラインモデルを上回っており，モデルサイズを拡張しても表 1 で示した結果と概ね一貫していることが観測された．これは，モデルサイズが変わっても傾向が大きくずれない可能性が高いことを示唆しており，良い性質と言える．

6　おわりに

Transformer に基づく大規模言語モデル (LLM) の構成要素の一つである FFN に対して，モデル全体での配置場所に依存した重要度を検証した．複数のモデルサイズで評価を行い，全層数の 70 – 90% の連続した中間から後方の層に FFN を集中配置することで，複数の下流タスクでベースラインの性能を上回る結果を得た．この結果から FNN は入力に近い層より中間から後方の層で，より効果を発揮する可能性が高いことが示唆された．FFN の配置を工夫することで，手軽に標準的な Transformer モデルを改善できるのであれば，モデル構成のチューニングなど新たな進展や可能性が期待できる．

性能）であることを確認されたい．

8）　一部の評価データではベースラインモデルがチャンスレートを下回ったため本稿の議論からは除外した．

9）　以降，個別の検証モデルを “[FFN 拡張層の配置位置]_[FFN 拡張層の配置割合]_[モデルサイズ]” の形式で表す．例として，FFN 拡張層の配置位置が middle，配置割合が 50%，モデルサイズが 570M の場合は “middle_50_570M” と表す．

10）　2B モデルの評価の詳細については付録 C に記載．

謝辞

本研究は，JST ムーンショット型研究開発事業 JPMJMS2011-35 (fundamental research)，および，文部科学省の補助事業「生成 AI モデルの透明性・信頼性の確保に向けた研究開発拠点形成」の支援を受けたものです.

本研究は九州大学情報基盤研究開発センター研究用計算機システムの一般利用を利用しました.

本研究成果（の一部）は，データ活用社会創成プラットフォーム mdx [12] を利用して得られた物です.

参考文献

[1] A Vaswani. Attention is all you need. **Advances in Neural Information Processing Systems**, 2017.

[2] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. **CoRR**, Vol. abs/2302.13971, , 2023.

[3] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture. In **Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event**, Vol. 119 of **Proceedings of Machine Learning Research**, pp. 10524–10533. PMLR, 2020.

[4] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021**, pp. 5484–5495. Association for Computational Linguistics, 2021.

[5] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022**, pp. 8493–8502. Association for Computational Linguistics, 2022.

[6] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, **Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022**, 2022.

[7] Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Analyzing feed-forward blocks in transformers through the lens of attention maps. In **The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024**. OpenReview.net, 2024.

[8] Noam Shazeer. GLU variants improve transformer. **CoRR**, Vol. abs/2002.05202, , 2020.

[9] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. **CoRR**, Vol. abs/1706.04115, , 2017.

[10] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale. In **The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022**. OpenReview.net, 2022.

[11] Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021**, pp. 6491–6506. Association for Computational Linguistics, 2021.

[12] Toyotaro Suzumura, Akiyoshi Sugiki, Hiroyuki Takizawa, Akira Imakura, Hiroshi Nakamura, Kenjiro Taura, Tomohiro Kudoh, Toshihiro Hanawa, Yuji Sekiya, Hiroki Kobayashi, Yohei Kuga, Ryo Nakamura, Renhe Jiang, Junya Kawase, Masatoshi Hanai, Hiroshi Miyazaki, Tsutomu Ishizaki, Daisuke Shimotoku, Daisuke Miyamoto, Kento Aida, Atsuko Takefusa, Takashi Kurimoto, Koji Sasayama, Naoya Kitagawa, Ikki Fujiwara, Yusuke Tanimura, Takayuki Aoki, Toshio Endo, Satoshi Ohshima, Keiichiro Fukazawa, Susumu Date, and Toshihiro Uchibayashi. mdx: A cloud platform for supporting data science and cross-disciplinary research collaborations. In **2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)**, pp. 1–7, 2022.

[13] Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. Fineweb-edu: the finest collection of educational content, 2024.

[14] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, **Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022**, 2022.

[15] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In **7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019**. OpenReview.net, 2019.

[16] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. **OpenAI blog**, Vol. 1, No. 8, p. 9, 2019.

[17] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 07 2024.

[18] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset, Aug 2016.

[19] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In **5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings**. OpenReview.net, 2017.

[20] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In **The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020**, pp. 8732–8740. AAAI Press, 2020.

[21] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In **Thirty-Fourth AAAI Conference on Artificial Intelligence**, 2020.

[22] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, 2019.

[23] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the AI2 reasoning challenge. **CoRR**, Vol. abs/1803.05457, , 2018.

[24] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In **EMNLP**, 2018.

表 3　モデルサイズ 2B における検証モデルのベースラインモデルに対する相対的改善度（%）

モデル	Wikitext	LAMBADA	ARC-e	ARC-c	Winogrande	PIQA	OBQA	HellaSwag	平均
middle_70	**+1.60**	−9.01	0.00	**+6.32**	−1.61	**+1.11**	−4.35	**+0.24**	−0.71
final_90	**+1.07**	−7.88	−0.56	**+5.17**	**+2.68**	**+0.55**	**+6.52**	**+0.94**	**+1.06**

A　事前学習の詳細設定

事前学習には，FineWeb-Edu [13] データセットを使用する．学習トークン数は，Chinchilla の法則 [14] に従い，285M，570M，2B パラメータモデルに対してそれぞれ，5.7B，11.4B，40B トークンとする．バッチサイズと最大シーケンス長（1024 トークン）から算出される 1 ステップあたりのトークン数をもとに，総ステップ数を 20,000 ステップと設定する．事前学習の設定として，オプティマイザには AdamW [15] を使用し，パラメータは $\beta_1 = 0.9$，$\beta_2 = 0.95$ とする．重み減衰係数は 0.1 を採用する．学習率のスケジューリングにはコサインスケジューラを採用し，最初の 1000 ステップで線形に増加させて最大学習率 3e-4 に到達させた後，コサイン波形に従って減衰させる．トークナイズには GPT-2 のボキャブラリを使用する [16]．

B　評価手法の詳細

B.1　下流タスク性能の評価

下流タスク性能については `lm-evaluation-harness` フレームワーク [17] を利用し，複数タスクから多角的な評価を実施する．評価には以下のタスクを使用する：

LAMBADA[18] は文章全体の文脈を理解し最後の単語を予測するタスク，Wikitext[19] は Wikipedia 記事を用いた言語モデリングタスクである．Winogrande[20] は常識的推論に基づく 2 択形式のタスク，PIQA[21] は日常生活における物理的な常識の理解を評価する 2 択形式のタスクである．HellaSwag[22] は文脈から最も自然な文を選択する 4 択形式のタスクである．ARC[23] は科学的知識と推論能力を評価する 4 択形式のタスクで，比較的単純な Easy Set (ARC-e) と深い推論が必要な Challenge Set (ARC-c) で構成される．OpenBookQA（OBQA）[24] は科学的知識を用いた応用的な推論能力を評価する 4 択形式のタスクである．

評価指標として，2 択形式の Winogrande と PIQA（チャンスレート 50%），4 択形式の HellaSwag，ARC，OBQA（チャンスレート 25%）では正解率（Accuracy）を用いる．LAMBADA と Wikitext では正解率に加えて，語彙サイズに依存する Perplexity も測定する．

B.2　モデルの知識量の評価

Zero-Shot Relation Extraction (zsRE) データセット [9] の各事例は，知識に基づく質問文とその答えのペアで構成される．知識量を測定するタスク (zsRE タスク)[10, 11] では，評価時に，質問文のみ，または質問文と答えの一部を入力として与え，モデルに次のトークンを生成させる．具体的には，まず質問文のみのプロンプトから 1 トークン生成させ，答えの最初のトークンとの一致を調べる．次に答えの最初のトークンを最初のプロンプトに付加したものを 2 番目プロンプトとし，1 トークン生成させ，答えの 2 番目のトークンとの一致を調べる，という手順を繰り返す．答えの全トークンに対する一致率をその事例に対する正解率とし，全事例（約 20,000 件）の正解率の平均をモデルの知識量の指標とする．

C　2B モデルの評価詳細

表 3 に 2B モデルにおける評価タスク毎のベースラインに対する相対的な改善度（Relative Improvement:RI）を示す．`middle_70` と `final_90` の評価結果から，いくつかの興味深い知見が得られた．Wikitext や PIQA，HellaSwag では両モデルともわずかな性能向上（+0.24%〜+1.60%）を示した一方で，文脈全体の理解を必要とする LAMBADA では大幅な性能低下（-9.01%，-7.88%）が観察された．特筆すべき点として，科学的知識と複雑な推論を必要とする ARC-c では両モデルとも顕著な性能向上（+6.32%，+5.17%）を示し，OBQA では `final_90` で+6.52%の大幅な改善がみられた．総合的に見ると，`final_90` は Winogrande（+2.68%）を含む多くのタスクで性能向上を示し，平均でも+1.06%の改善を達成した一方，`middle_70` は一部のタスクで性能低下が見られ平均で-0.71%となった．