モデル拡張を用いた段階的事前学習による モデル系列の効率的な構築

矢野 一樹 [1]　高瀬 翔 [1,2]　小林 颯介 [1]　清野 舜 [2]　鈴木 潤 [1]
[1] 東北大学　[2]SB Intuitions 株式会社
yano.kazuki@dc.tohoku.ac.jp　{sosuke.kobayashi.b2,jun.suzuki}@tohoku.ac.jp
{sho.takase,shun.kiyono}@sbintuitions.co.jp

概要

大規模言語モデルの実応用では，7B，13B，70Bといったパラメータ数の異なる複数のモデル（モデル系列）を提供することが一般的である．モデル系列の構築は，素朴には各サイズのモデルを個別に構築する必要があり，計算コストは加算的に増加する．本研究では，小さなモデルから段階的に学習を進め，サイズを拡張させながら，モデル系列を構築する手法を提案する．実験では，提案手法が計算コストを削減しつつ，個別にモデル系列を学習する場合と比較して同等以上の性能を達成できることを示す．

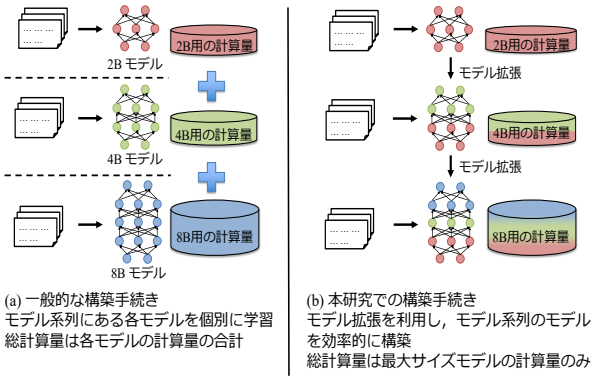

図 1　モデル系列の一般的な構築手続き (左図 (a)) と本研究での構築手続き (右図 (b)) の概要図..

1　はじめに

大規模言語モデル (LLM) が実応用で幅広く活用されるなかで，異なるパラメータ数の複数のモデル (以下，**モデル系列**) を，構築・提供することが一般的となっている．モデル系列の提供は，多様な計算資源の制約や用途に対応するための重要なアプローチとなっている．例えば，Llama 2 では 7B，13B，70B のモデル [1] が，Qwen2 では 0.5B，1.5B，7B，72B のモデル [2] が公開されている．小規模なモデルは，日常的なタスクにおける効率的な処理と高速な応答を備えつつ，スマートフォンやエッジデバイスといった計算資源の制約が厳しい環境での展開を可能にする [3]．一方，大規模なモデルは，高度な推論能力や複雑なタスクの処理が求められる場面で使用され，通常は大規模なサーバ上に配備される．このように異なる特性を持つモデルを系列として提供することで，ユーザからの幅広い要求に応えることが可能である．

こうしたモデル系列の構築では各サイズのモデルを個別にスクラッチから学習する手続きが一般的であり，必要な計算資源は単純に加算的に増加していく．特に大規模モデルの学習には数千 GPU 日という計算資源を要する [4] ため，モデル系列を開発する際の総計算コストは研究機関や企業にとって大きな負担となっている．

大規模モデルの学習に要する計算コストを削減する手法として，学習済みの小規模モデルを拡張して大規模モデルの初期値とする手法が注目されている [5, 6]．本研究では，この手法を繰り返し適用することで，小規模モデルから大規模モデルへの段階的な構築を行い，モデル系列構築の総訓練コストを削減する手法を提案する (図 1)．実験により，提案手法では，個々のモデルを個別に構築する場合よりも少ない計算コストでモデル系列を得られることを示す．また，モデルサイズに応じた学習率の段階的な調整により，提案手法により構築したモデル系列は，個別に構築したモデルと同等以上の性能を達成可能であることを示す．加えて，構築したモデル系列に対して SFT と DPO による事後学習を実施し，提案手法の有効性が事後学習後も維持されることも示す．

2 方法

本研究では，複数のサイズの言語モデルを段階的に構築する手法を提案する．既存のアプローチでは，各サイズのモデルを個別にスクラッチから学習している．提案手法では，小さなモデルから学習を開始し，そのモデルを拡張しながら段階的に大きなモデルを構築する．この過程で得られる中間段階のモデルも，それぞれ独立したモデルとして利用可能であることを目指す．提案手法により，モデル系列を個別に構築する方法よりも低いコストで系列全体を構築可能である．

2.1 モデル系列の構築

モデルサイズの単調増加列 $[X_1, X_2, \ldots, X_n]$ をなすモデル系列に対して，対応するモデルパラメータの系列 $[\theta_1, \theta_2, \ldots, \theta_n]$ を構築する．ここで $\theta_i \in \mathbb{R}^{X_i}$ は i 番目のモデルのパラメータであり，モデルサイズは単調増加する（$X_{i+1} > X_i$）．

各モデルサイズ X_i に対して，その学習に用いるトークン数を T_i とする．このとき，サイズ X_i のモデルをスクラッチから学習する際に必要な計算量を $\mathrm{FLOPs}(X_i, T_i) = 6X_iT_i$ と定義する [7]．

2.2 段階的な学習

まず初期モデル θ_1 をスクラッチから学習する．これには $\mathrm{FLOPs}(X_1, T_1)$ の計算量を要する．続く各段階では，モデル拡張法 f を用いて次のモデルを初期化する：

$$\theta_{i+1} = f(\theta_i; X_{i+1}) \quad (i \geq 1) \tag{1}$$

ここで $f(\cdot; X_{i+1}) : \mathbb{R}^{X_i} \to \mathbb{R}^{X_{i+1}}$ はモデル拡張法であり，拡張後のパラメータ θ_{i+1} が効率的な学習の初期値として機能するように設計される．本研究ではモデル拡張法として，bert2BERT [5] を用いた．[1] bert2BERT では，Transformer モデル [8] の幅や深さを拡張させる．具体的には，幅方向の拡張では線形層の重みを複製して，モデルの隠れ層の次元を拡大する．また，深さ方向の拡張では学習済みの層を上層に複製して積み重ねる．

bert2BERT による初期化後，各段階 $i+1$ $(i \geq 1)$ のモデルを $\mathrm{FLOPs}(X_{i+1}, T_{i+1}) - \sum_{j=1}^{i} \mathrm{FLOPs}(X_j, T_j)$ の計算量で追加学習する．これは，サイズ X_{i+1} のモデルをスクラッチから学習する際の計算量から，それまでの段階で使用した計算量を差し引いたものである．この手続きにより，サイズ X_{i+1} のモデルの学習に必要な計算量 $\mathrm{FLOPs}(X_{i+1}, T_{i+1})$ で，サイズ X_1 から X_{i+1} までのモデル系列全体を得ることができる．

3 実験 1: モデル系列の事前学習

提案手法である，段階的な学習によるモデル系列の構築の有効性を検証するため，事前学習を通じた実験を行なった．具体的には，8B を最大サイズとするモデル系列に対して，計算効率と最終的な性能の両面から提案手法の有効性を検証する．

3.1 データセット

事前学習の訓練・開発データとして FineWeb-Edu [9] を使用した．FineWeb-Edu は教育的コンテンツを中心としたウェブコーパスである．入力テキストのトークン化には GPT-2 [10] のトークナイザーを採用した．モデルの評価には FineWeb-Edu の検証用データセット (Valid) と複数の標準的なベンチマークデータセットを使用した．[2]

3.2 モデル設定

段階的学習法は，任意のパラメータ数の増加に対応可能であるが，本研究では各段階でパラメータ数を 2 倍にする設定を採用した．具体的には，$[X_1 = 1\mathrm{B}, X_2 = 2\mathrm{B}, X_3 = 4\mathrm{B}, X_4 = 8\mathrm{B}]$ からなるモデル系列を用意する．[3]

モデルには Llama [4] と同様のアーキテクチャを採用した．また，入力の最大系列長は 1024 とする．さらに，モデル系列内の各モデルをスクラッチから学習させる際の学習率は 3.0×10^{-4} とする．

3.3 訓練データ量

段階的訓練法では，モデルサイズ X_i からサイズ X_{i+1} への拡大時に，X_{i+1} モデルをスクラッチから訓練する場合の計算量から今まで X_i にかけた計算量の差分を計算し，その分のデータ量を X_{i+1} モデルに訓練させる．

近年の大規模言語モデルの事前学習においては，Chinchilla 則 [11] が示す最適値を大きく超えるデータ量での訓練が一般的となっている．これは，計算効率は低下するものの，最終的なモデルの性能向上

1) 原論文 [5] では，新規パラメータの初期化方法として，AKI と FPI という二つの手法が提案されているが，本研究では AKI を用いた．

2) 具体的なタスクの詳細は付録 B に記す．

3) 各モデルの具体的な幅や深さについては付録 A に記す．

が期待されるためである．例えば，Llama 3-8B [12] の事例では，Chinchilla 則における最適値の約 100 倍となる 15 兆トークンでの訓練が報告されている．そこで，スクラッチから訓練させるモデルの訓練データ量を Chinchilla 則で決定される量の 2 倍にする設定 (2x Chinchilla rule と表現する) を用意した．

提案する段階的学習アプローチは本質的により多くのデータを消費する性質を持つ．例えば 8B モデルまでの段階的学習では，1B，2B,4B モデルの方が同一の計算量において 8B モデルよりもデータ処理効率に優れる．そのため，8B モデルをスクラッチから学習する場合と比較して，段階的学習ではより多くのデータ量が必要となる．このように段階的学習法は必然的により多くのデータを消費するため，段階的学習の性能向上が単に新規データをより多く学習したことに起因する可能性が考えられる．そこで，この効果を分離して評価するため，スクラッチからの学習と同一の訓練データを用いた実験設定を導入した．具体的には，2x Chinchilla rule 設定において，訓練データをスクラッチから 8B モデルを訓練する際に使用する 320B トークンに制限し，必要に応じて同一データの再学習を許可する設定である．この設定では，1B → 2B → 4B までの段階的学習は 2x Chinchilla rule 設定と同様に進み，8B の学習段階でのみデータの再学習が発生する．この固定のデータ量で再学習を許可する設定を 8B モデル拡張 w/ Fixed Data と表現する．8B モデル拡張 と，8B モデル拡張 w/ Fixed Data との比較により，段階的学習法による性能向上が新規データの追加に依存しないことを検証する．

3.4 計算量

段階的学習法の主要な利点の一つは，複数のサイズのモデルを効率的に得られることである． 2.1 節で定義したように，計算量は各モデルのパラメータ数 X と学習トークン数 T から FLOPs = $6XT$ で計算される．例として，モデル系列 [1B, 2B, 4B, 8B] を 2x Chinchilla rule の元で各モデルを独立に学習させると，必要な FLOPs はそれぞれ [0.24Z, 0.96Z, 3.84Z, 15.4Z] となる．つまり系列全体の構築には 20.4ZFLOPs 必要となる．一方，提案手法では，モデル系列の構築に必要な計算量は最大サイズである 8B モデルの構築に必要な計算量と同等である．したがって，提案手法によるモデル系列の構築では 15.4ZFLOPs の計算量で済む．つまり，個別にモデル系列を構築する場合と比べ，段階的学習法は計算量を約 25%削減可能である．これは，段階的学習法が大規模なモデル系列の構築において，計算効率の面で優位性を持つことを示している．

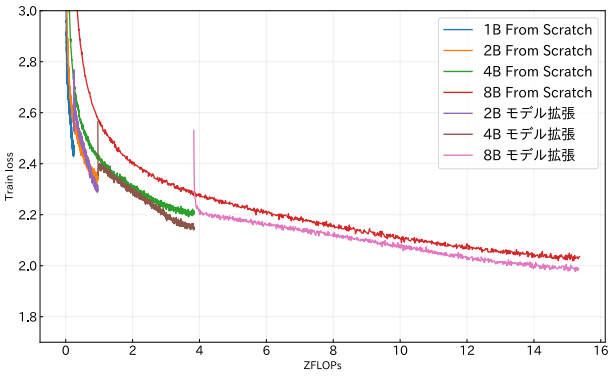

図 2 実験 1(3 節) における各モデルの計算量に対する損失関数の値．

3.5 学習率調整

大規模言語モデルの学習において，高い学習率の設定が性能向上に寄与しうるが [13]，モデルサイズが大きくなるほど学習の不安定性が増すため，より小さな学習率を使用する必要がある [4]．例えば，予備実験において，1.5×10^{-3} の学習率では 1B モデルでは安定した学習が可能であったが，8B モデルでは損失値のスパイクが発生し，学習が破綻することが確認された．

段階的学習法では小さいモデルから学習を開始するという特徴がある．そのため，小さいモデルの学習時には高い学習率で性能を高めつつ，大きいモデルでは学習率を下げて学習を安定させるといった戦略を採用可能である．

そこで本実験では，1B モデルでの 1.5×10^{-3} から 8B モデルでの 3.0×10^{-4} まで，モデルサイズの増加に応じて学習率を段階的に下げていく：

- 1B (From Scratch): 1.5×10^{-3}
- 2B (モデル拡張): 1.1×10^{-3}
- 4B (モデル拡張): 7.0×10^{-4}
- 8B (モデル拡張): 3.0×10^{-4}

3.6 結果

表 1 に，スクラッチからの学習とモデル拡張による段階的学習の各モデルサイズにおける事前学習評価結果を示す．また図 2 に各モデルの計算量に対する損失関数の値を示す．実験結果からは，提案手法

表 1　事前学習モデルの評価結果．1B から 8B パラメータの各モデルサイズにおいて，スクラッチからの学習と提案手法の性能を，Perplexity と 7 つの下流タスクの Accuracy で比較した．

	構築方法	Perplexity ↓		Accuracy ↑						
		Valid	Wikitext	LAMBADA	ARC-e	ARC-c	Winogrande	PIQA	OBQA	HellaSwag
1B	From Scratch	11.6	19.7	42.2	65.9	37.5	56.9	73.0	39.2	55.0
2B	From Scratch	10.4	17.1	48.5	66.9	38.4	56.6	**75.3**	**41.6**	58.0
	モデル拡張	**10.1**	**16.2**	**52.6**	**71.8**	**42.2**	**61.9**	75.0	40.8	**62.5**
4B	From Scratch	9.18	14.0	51.7	71.7	43.2	59.2	76.7	40.8	63.3
	モデル拡張	**8.72**	**13.1**	**55.7**	**74.4**	**47.8**	**65.2**	**77.4**	**45.8**	**68.1**
8B	From Scratch	7.85	10.2	55.2	74.5	47.4	62.4	77.0	46.4	67.9
	モデル拡張	**7.64**	10.1	**59.0**	**76.7**	48.3	**65.8**	**78.3**	46.6	71.0
	モデル拡張 w/ Fixed Data	**7.64**	**9.8**	**59.0**	76.1	**50.6**	64.7	78.2	**47.2**	**71.2**

の方がスクラッチから学習したものに比べ性能が優位であることが分かる．これらの結果は，段階的学習の有効性を示すとともに，モデルサイズの増加に応じた学習率の段階的な調整が，さらなる性能向上に結びつくことを示している．[4] また，8B モデル拡張 w/ Fixed Data の設定では，320B トークンの固定コーパスを用いた場合でも，8B モデル拡張 とほぼ同等の性能を示した．以上の結果から，段階的学習と適切な学習率の調整によって，スクラッチからの学習と同等以上の性能を達成できることが確認された．また，固定のデータ量を用いた実験から，この手法の効果が新規データ量の増加によるものではないことも示された．

4　実験 2: モデル系列の事後学習

本実験では，段階的に構築したモデルに SFT および DPO [14] による事後学習を実施する．本実験により，節 3 の結果と同様に，スクラッチから学習したモデルより，モデル拡張によるモデルが優位な性能を示すかどうかを検証する

4.1　実験設定

実験 1 で構築したモデル系列に対して事後学習を実施することで，段階的学習の有効性を検証する．事後学習には Meng ら [15] の設定を採用した．具体的には，教師付きファインチューニング（SFT）を UltraChat-200k [16] データセットを用いて行い，続いて Ultrachat Feedback [17] データセットを用いた DPO [14] を行った．SFT の学習率は 3.0×10^{-5}，DPO の学習率は 5.0×10^{-7} とし，すべてのモデルで統一した．各モデルの応答品質評価には MT-Bench [18] を採用した．比較対象として，個別にスクラッチから学習させたモデル（2B, 4B, 8B）を用意し，同様の事後学習を実施した．

4)　学習率を調整せず固定した場合の結果を付録 C にて示す．

表 2　事後学習後のモデル評価結果．SFT と DPO による事後学習を実施し，MT-Bench スコアで評価した．

	構築方法	MT-Bench ↑
2B	From Scratch	2.04
	モデル拡張	**3.22**
4B	From Scratch	3.02
	モデル拡張	**3.63**
8B	From Scratch	3.45
	モデル拡張	3.96
	モデル拡張 w/ Fixed Data	**3.98**

4.2　結果

表 2 に各モデル系列の事後学習後の MT-Bench スコアを示す．段階的に構築したモデルのスコアは，スクラッチから学習したモデルのスコアを上回っていた．この結果は，事前学習時と同様に，段階的学習と適切な学習率調整によるモデルが事後学習においても有効であることを示している．

5　おわりに

本研究では，モデル拡張を用いたモデル系列の効率的な構築方法を提案した．提案手法によって，小さなモデルから段階的に訓練を行うことによって副次的にモデル系列を構築可能である．実験により，8B サイズを最大サイズとするモデル系列において，提案手法がモデル系列を個別に構築する方法と比較して，約 25%の計算コストを削減できることを示した．さらに，モデルサイズに応じた学習率の段階的な調整により，事前学習および事後学習の両方においてスクラッチから訓練したモデルと比較して性能が向上することを示した．

謝辞

本研究の一部は，JST ムーンショット型研究開発事業 JPMJMS2011-35 (fundamental research) の支援を受けたものです.

参考文献

[1] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[2] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024.

[3] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. **arXiv preprint arXiv:2404.14219**, 2024.

[4] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. **arXiv preprint arXiv:2302.13971**, 2023.

[5] Cheng Chen, Yichun Yin, Lifeng Shang, Xin Jiang, Yujia Qin, Fengyu Wang, Zhi Wang, Xiao Chen, Zhiyuan Liu, and Qun Liu. bert2BERT: Towards reusable pretrained language models. 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)**, pp. 2134–2148, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[6] Wenyu Du, Tongxu Luo, Zihan Qiu, Zeyu Huang, Yikang Shen, Reynold Cheng, Yike Guo, and Jie Fu. Stacking your transformers: A closer look at model growth for efficient LLM pre-training. In **The Thirty-eighth Annual Conference on Neural Information Processing Systems**, 2024.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. **Advances in neural information processing systems**, Vol. 33, pp. 1877–1901, 2020.

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

[9] Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In **The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track**, 2024.

[10] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019.

[11] 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 William Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, **Advances in Neural Information Processing Systems**, 2022.

[12] AI@Meta. The llama 3 herd of models, 2024.

[13] Sho Takase, Shun Kiyono, Sosuke Kobayashi, and Jun Suzuki. Spike no more: Stabilizing the pre-training of large language models. **arXiv preprint arXiv:2312.16903**, 2023.

[14] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In **Thirty-seventh Conference on Neural Information Processing Systems**, 2023.

[15] Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. In **The Thirty-eighth Annual Conference on Neural Information Processing Systems**, 2024.

[16] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, pp. 3029–3051, Singapore, December 2023. Association for Computational Linguistics.

[17] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. ULTRAFEEDBACK: Boosting language models with scaled AI feedback. In **Forty-first International Conference on Machine Learning**, 2024.

[18] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In **Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track**, 2023.

[19] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In **International Conference on Learning Representations**, 2017.

[20] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In Katrin Erk and Noah A. Smith, editors, **Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 1525–1534, Berlin, Germany, August 2016. Association for Computational Linguistics.

[21] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. **Communications of the ACM**, Vol. 64, No. 9, pp. 99–106, 2021.

[22] Yonatan Bisk, Rowan Zellers, Ronan bras, Jianfeng Gao, and Choi Yejin. Piqa: Reasoning about physical commonsense in natural language. **Proceedings of the AAAI Conference on Artificial Intelligence**, Vol. 34, pp. 7432–7439, 04 2020.

[23] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez, editors, **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics.

[24] 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, 2018.

[25] 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 Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing**, pp. 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

表 3　各モデルサイズにおけるアーキテクチャの設定

	隠れ層の次元数	FFN の次元数	層数	ヘッド数
1B	2048	7168	18	16
2B	2560	8960	22	20
4B	3200	11200	27	25
8B	4096	14336	33	32

表 4　事前学習モデルの固定学習率を用いた評価結果．各モデルの学習率は 3.0×10^{-4} である．

	構築方法	Perplexity ↓		Accuracy ↑						
		Valid	Wikitext	LAMBADA	ARC-e	ARC-c	Winogrande	PIQA	OBQA	HellaSwag
1B	From Scratch	12.0	20.6	42.0	62.3	34.6	55.5	70.5	35.6	51.6
2B	モデル拡張	10.5	17.2	47.9	67.7	38.4	58.1	72.1	40.0	58.1
4B	モデル拡張	9.17	14.0	50.8	72.6	42.1	59.6	76.2	44.0	63.8
8B	モデル拡張	8.04	10.6	54.1	75.7	46.7	61.3	76.8	46.2	68.2
	モデル拡張 w/ Fixed Data	8.04	10.4	54.1	75.2	46.8	63.5	77.2	45.8	68.1

A　モデルサイズ

実験で用いたモデル系列では，Llama アーキテクチャを基に，1B，2B，4B，8B パラメータのモデルを構築した．表 3 に，各モデルサイズにおける具体的なアーキテクチャの設定を示す．モデル拡張に伴い，隠れ層の次元数，FFN の次元数，層数，ヘッド数を段階的に大きくしている．

B　評価タスク

評価には，FineWeb-Edu の検証用データ（Valid）と WikiText [19] における Perplexity を用いた．また，事前学習モデルの性能を総合的に評価するため，複数の下流タスクでのゼロショット性能を測定した．具体的には，言語モデリング（LAMBADA [20]），常識推論（WinoGrande [21]，PIQA [22]，HellaSwag [23]），質問応答（ARC-e，ARC-c [24]，OBQA [25]）の各タスクを採用した．

C　固定学習率における実験

実験 1(3) においては，段階的学習において，各モデルで調整された学習率を用いた．本節では，すべてのモデルで学習率を 3.0×10^{-4} に固定した実験を行なった．表 4 に，1B から 8B までの各モデルサイズにおける評価結果を示す．