事前学習における学習率スケジューラが事後学習後の性能に与える影響

矢野 一樹 [1]　清野 舜 [2]　小林 颯介 [1]　高瀬 翔 [1]　鈴木 潤 [1,3,4]
[1] 東北大学　[2]SB Intuitions 株式会社　[3] 理化学研究所　[4] 国立情報学研究所 LLMC
yano.kazuki@dc.tohoku.ac.jp　is-failab-research@grp.tohoku.ac.jp

概要

本稿では、大規模言語モデルの事前学習における学習率スケジューラについて、事後学習、特に教師あり微調整（SFT）後の下流タスク性能への影響に焦点を当てて検証する。本実験では、ウォームアップ期間終了後に学習率を一定に保つ Warmup-Stable-Only（WSO）スケジューラを導入する。1B および 8B パラメータモデルを用いた実験により、複数の学習率減衰型スケジューラと比較して、WSO スケジューラを用いたモデルが SFT 後において一貫して良好な性能を達成することを示す。これらの知見は、事前学習の評価指標を改善するために学習率の減衰を適用することが、下流タスクにおける適応能力を損なう可能性があることを示唆している。

1　はじめに

学習率スケジューラは、大規模言語モデル（LLM）の事前学習において、重要な要素の一つである。Cosine スケジューラはこれまで数多くのモデル [1, 2, 3] で採用されてきたが、継続事前学習などの学習パラダイムにおいては、減衰後の値からヒューリスティックに学習率を調整する必要があるため、柔軟性に欠ける [4, 5]。この柔軟性の欠如に対処するため、近年の研究では Warmup-Stable-Decay（WSD）という手法が提案されている。これは事前学習の大部分で学習率を一定に保ち、終盤のみ短時間にわたって減衰させる手法である [6, 7, 8]。

これらの学習率スケジューラは、いずれも事前学習モデルの性能最適化を目的として学習率を減衰させている。しかしながら、実際の応用において、より重要なのは教師あり微調整（Supervised Fine-Tuning: 以降本稿では SFT と記す）などの事後学習後の性能である。近年の研究結果 [9, 10] によれば、強力な事前学習モデルが必ずしも SFT 後の優れた性能を保証しないことが示されている。このことは、事前学習時の性能向上を目的とした学習率減衰が、必ずしも SFT 後のモデルにとって良好な選択ではない可能性を示唆している。

本研究では、事前学習段階における適切な学習率スケジューラの選択について、SFT 後の性能という観点から実証的に検討する。特に、学習率減衰を伴わない Warmup-Stable-Only（WSO）スケジューラを導入する。この方法では WSD（Warmup-Stable-Decay）スケジューラの減衰期間を省略し、学習率を最後まで一定値に維持する。実験結果から、WSO スケジューラは学習率減衰型スケジューラと比較して、1B および 8B モデルの両方において SFT 後の性能が一貫して優れていることが確認された（図 1）。

これらの知見は、事前学習時の評価指標最適化を目的とした学習率減衰が、必ずしも SFT 後の下流タスク性能向上につながらないことを示している。さらに、WSO スケジューラは減衰期間を必要としないため、実装が簡便でありながら、学習パイプライン全体を通じた最終性能の向上をもたらす。これらの結果は、高い適応性を持つ LLM を構築する上で、WSO が従来の学習率減衰型スケジューラに代わる有望な選択肢となり得ることを示唆している。

2　準備

LLM は通常、段階的な学習を用いて構築される。典型的には、LLM の学習パイプラインは、事前学習と事後学習という 2 つの段階から構成される。本節では、これらの学習段階について概説するとともに、事前学習段階で一般的に用いられる学習率スケジューラについて議論する。

事前学習　事前学習は LLM 構築の基盤となる段階であり、モデルは大規模なテキストコーパスから次トークン予測損失を最小化することで、汎用的な

This work is published without peer review and is licensed by the author(s) under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

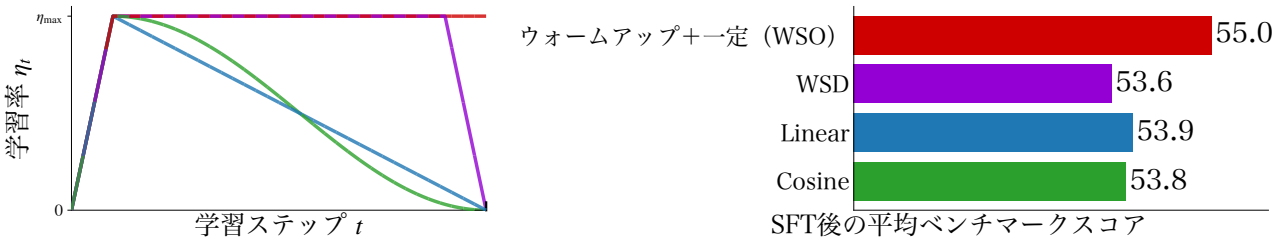

図 1　事前学習における学習率スケジューラと、それらが SFT（教師あり微調整）後の性能に与える影響。減衰期間を省略した Warmup-Stable-Only (WSO) が、SFT 後において最高性能を達成している。

言語理解能力を獲得する。近年では、標準的な事前学習に加えて中間学習 [11] といった複数段階の事前学習手法が採用されるケースが増えている。

事後学習　事後学習は、事前学習済みモデルを特定のタスクに適応させるプロセスであり、これによりモデルは人間の指示に従い、有害な出力を生成しないよう制御可能となる。事後学習の手法としては、SFT や選好チューニング（例：DPO [12]）などが挙げられる。事後学習は多段階プロセスとして設計可能であり、現在も様々な設計の選択肢が研究対象となっているが、SFT は比較的標準化された手法であり、中核的な段階として位置付けられている。本論文では、標準的な事後学習段階として SFT に焦点を当て、SFT 実施後のモデル性能を評価する[1)]。

2.1　タスク定義

実際の LLM 開発では、ある段階においてモデルを評価し、最も性能の高いモデルが次の段階の出発点として選択されている。ここで、$\mathtt{Task}_s(M)$ を、与えられた LLM M に対して、対象とする段階 s の評価に用いられる事前定義済みタスク群における性能を返す関数と定義する。ここで $s \in \{\mathrm{pre}, \mathrm{post}\}$ は学習段階を示し、pre は事前学習段階、post は事後学習段階を意味する。$M_2[M_1]$ という表記は、何らかの設定と初期化手法を用いて M_1 をベースとして学習されたモデル M_2 を表す。ここで M_{rand} は、重みがランダムに初期化されたモデルを指す。さらに、事前学習と事後学習によってそれぞれ得られたモデルの集合を $\mathcal{M}_{\mathrm{pre}}$ および $\mathcal{M}_{\mathrm{post}}$ で表す。これらの集合には、さまざまなハイパーパラメータ設定によって学習されたモデルが含まれる。したがって、LLM を構築する際の典型的な学習パイプラインは以下のように表現できる：

$$\begin{aligned} \widehat{M}_{\mathrm{pre}} &= \operatorname*{arg\,max}_{M_{\mathrm{pre}} \in \mathcal{M}_{\mathrm{pre}}} \left\{ \mathtt{Task}_{\mathrm{pre}}(M_{\mathrm{pre}}[M_{\mathrm{rand}}]) \right\}, \\ \widehat{M}_{\mathrm{post}} &= \operatorname*{arg\,max}_{M_{\mathrm{post}} \in \mathcal{M}_{\mathrm{post}}} \left\{ \mathtt{Task}_{\mathrm{post}}(M_{\mathrm{post}}[\widehat{M}_{\mathrm{pre}}[M_{\mathrm{rand}}]]) \right\}. \end{aligned} \tag{1}$$

この定式化では、最終モデル $\widehat{M}_{\mathrm{post}}$ の性能という観点から見ると、最適解を得られない可能性がある。なぜなら、中間段階で性能が最も高いモデルを選択することが、最終的に最良の性能を達成することを保証しないからである。したがって、概念的には、本学習パイプラインにおいてより優れた最終モデルを得るために、以下の探索問題を検討したい：

$$\widehat{M}_{\mathrm{post}} = \operatorname*{arg\,max}_{\substack{(M_{\mathrm{pre}}, M_{\mathrm{post}}) \\ \in (\mathcal{M}_{\mathrm{pre}}, \mathcal{M}_{\mathrm{post}})}} \left\{ \mathtt{Task}_{\mathrm{post}}(M_{\mathrm{post}}[M_{\mathrm{pre}}[M_{\mathrm{rand}}]]) \right\}. \tag{2}$$

本研究の主な目的は、事後学習段階に先行する大規模事前学習段階において、複数の学習率スケジューラを評価することで、この探索問題を実証的に検証することである。

2.2　学習率スケジューラの実践状況

近年の LLM の事前学習段階では、Cosine、Linear、あるいは WSD といった学習率減衰型スケジューラが用いられ、学習率を最大値の 0〜10%まで減衰させる設定が一般的である [3, 6, 13]。これらのスケジューラは、学習率減衰型スケジューラが損失値を、より最小化させるという実験的知見に基づき採用されており、それぞれ独立して $\mathtt{Task}_{\mathrm{pre}}(M_{\mathrm{pre}})$ の最適化を効果的に実現している。しかしながら、本来の目的はパイプライン全体が完了した後の性能、すなわち $\mathtt{Task}_{\mathrm{post}}(M_{post})$ を最大化することにある。この観点から、$\mathtt{Task}_{\mathrm{pre}}(M_{\mathrm{pre}})$ の最適化は必ずしも最適解とは言えない可能性がある。例えば、Springer ら [10] および Sun ら [9] の最近の研究結果によれば、

1)　事前学習にかかる計算コストは他の段階と比べて一般的にはるかに大きいため、事前学習の設定を最適化することは LLM 構築の効率性に重大な影響を及ぼす。本研究では、大規模事前学習時における学習率スケジューラの評価に主眼を置き、SFT 後の性能に基づいて非減衰型スケジューラの可能性について考察する。複数の事後学習段階にまたがる複雑な学習率スケジューリングの組み合わせについての検討は、今後の課題として残しておく。

This work is published without peer review and is licensed by the author(s) under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

事前学習後の良好な性能が、SFT 後の性能を保証するものではないことが明らかになっている。すなわち、事前学習時の性能を最適化するために採用された学習率減衰型スケジューラが、SFT 後のより良好な性能を達成するかは自明ではない。

2.3 学習率スケジューラの形式化

学習ステップ t における学習率 $\eta^{\texttt{Scheduler}}(t, \alpha_{\text{pre}})$ を Scheduler で指定される学習率スケジューラと事前学習時の最小学習率係数 α_{pre} を用いて表す。例えば WSD スケジューラは以下のように定義される：

$$\eta^{\text{WSD}}(t, \alpha_{\text{pre}}) = \begin{cases} \eta_{\max} \cdot \frac{t}{T_w} & t \le T_w \\ \eta_{\max} & T_w < t \le T_s \\ \eta_{\max}\left((1-\alpha_{\text{pre}})\frac{T-t}{T_d} + \alpha_{\text{pre}}\right) & t > T_s \end{cases} \tag{3}$$

ここで $\eta_{\max}$ は最大学習率、T は事前学習の総ステップ数を表す。また、$T_s = T_w + \rho \cdot (T - T_w)$, $T_d = T - (T_s + T_w)$ はそれぞれ、一定期間、減衰期間を表し、ρ は一定期間の比率を表す。

また、学習率減衰を伴わない学習率スケジューラの有効性を検証するため、WSD の簡易版である Warmup-Stable-Only（WSO）を検討する。WSO スケジューラでは減衰期間を省略しており、これは $\alpha_{\text{pre}} = 1.0$ を設定することに相当する。

$$\eta^{\text{WSO}}(t, \alpha_{\text{pre}}) = \begin{cases} \eta_{\max} \cdot \frac{t}{T_w} & t \le T_w \\ \eta_{\max} & T_w < t \le T \end{cases} \tag{4}$$

本実験では、4 種類の学習率スケジューラ $\texttt{Scheduler} \in \{\text{WSO}, \text{WSD}, \text{Cosine}, \text{Linear}\}$ を検証対象とする。Cosine スケジューラ $\eta^{\text{Cosine}}(t, \alpha_{\text{pre}})$ と Linear スケジューラ $\eta^{\text{Linear}}(t, \alpha_{\text{pre}})$ の詳細な定式化は付録 A に記載する。

3 実験

本実験では、複数の学習率減衰型スケジューラと、学習率を一定に保つ WSO スケジューラを比較し、事前学習時の学習率減衰が SFT 後の下流タスク性能に与える影響について検証する。

モデルアーキテクチャ Llama 3 アーキテクチャに準拠した 2 つのモデル規模（1B パラメータモデルと 8B パラメータモデル）を用いて実験を実施する。これらのモデルアーキテクチャは、それぞれ Llama-3.2-1B [14] および Llama-3.1-8B [15] と同一である。

表 1 1B モデルと 8B モデルの事前学習設定。

ハイパーパラメータ	1B	8B
学習設定		
総学習ステップ数	80,000	80,000
総トークン数	350B	500B
バッチサイズ	4,194,304	12,582,912
シーケンス長	2,048	2,048
最大学習率（$\eta_{\max}$）	3×10^{-4}	3×10^{-4}
重み減衰	0.1	0.1
Adam β_1	0.9	0.9
Adam β_2	0.95	0.95
Adam ϵ	1×10^{-8}	1×10^{-8}
勾配クリッピング	1.0	1.0
学習率スケジュール		
ウォームアップステップ数	1,000	1,000
ρ	0.75	0.75
α_{pre}	{0.0, 0.1, 1.0}	{0.0, 0.1, 1.0}

事前学習設定 事前学習の訓練・開発データとして FineWeb-Edu [16] データセットを用いた。また全スケジューラで、最大学習率 $\eta_{\max} = 3 \times 10^{-4}$ を設定する。本研究では、第 2.3 節で定式化した 3 種類の学習率スケジューラを検討する。具体的には、WSO（式 2.3）、WSD（式 2.3）、Cosine、および Linear スケジューラを実験対象とする。各スケジューラについて、最小学習率係数 $\alpha_{\text{pre}} \in \{0.0, 0.1, 1.0\}$ を変化させ、本稿で用いる表記法 $\eta^{\texttt{Scheduler}}(t, \alpha_{\text{pre}})$ に従って設定する。$\alpha_{\text{pre}} = 0.0$ の場合、学習率はゼロまで減衰する。この設定は事前学習性能を向上させることが Bergsma ら [13] の研究で示されている。$\alpha_{\text{pre}} = 0.1$ の場合、学習率は最大値の 10%まで減衰する。これは Chinchilla [17]、Llama 3 [18]、OLMo 2 [11] など、近年の LLM 構築において一般的に採用されている設定である。最後に、$\alpha_{\text{pre}} = 1.0$ の場合は WSO 設定に対応する。

表 1 に、事前学習に使用した詳細なハイパーパラメータを示す。

SFT の学習設定 本研究では Tulu-3 SFT 混合データセット[2)]を用いて SFT を実施する。各事前学習モデルに対して SFT 時の最適な学習率を特定するため、5×10^{-7} から 1×10^{-3} の範囲で包括的な学習率スイープを行った[3)]。

評価方法 モデルの評価は、事前学習後と SFT 後の 2 段階で実施する。事前学習後のモデルについては、質問応答タスク（ARC-Easy、ARC-Challenge [19]、

2) https://huggingface.co/datasets/allenai/tulu-3-sft-olmo-2-mixture/tree/main

3) SFT に関する詳細な手法は付録 B に記載する。

This work is published without peer review and is licensed by the author(s) under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

表 2 事前学習（PT）および SFT における相対性能。各モデルサイズおよび評価指標について、値は当該指標で最も性能が高かった学習率減衰型スケジューラとの差分（Δ）を表す。太字は最高性能を示す。

モデル	学習率スケジューラ	α_{pre}	PT Valid Loss ↓ Δ	PT Task Avg Δ	SFT Task Avg Δ
1B	WSO	1.0	*+0.071*	*-1.7*	***+0.3***
	WSD	0.1	+0.004	-1.5	+0.0
		0.0	**+0.000**	-1.2	-1.0
	Linear	0.1	+0.021	-2.0	-0.7
		0.0	+0.016	**+0.0**	-0.9
	Cosine	0.1	+0.019	-0.1	-0.7
		0.0	+0.016	-2.5	-0.7
8B	WSO	1.0	*+0.127*	*-0.8*	***+1.1***
	WSD	0.1	+0.019	-0.2	-0.8
		0.0	+0.014	**+0.0**	-0.3
	Linear	0.1	+0.013	-1.9	-0.6
		0.0	**+0.000**	-1.8	+0.0
	Cosine	0.1	+0.009	-2.2	-0.3
		0.0	+0.008	-2.3	-0.1

OpenBookQA [20]、BoolQ [21]）や常識推論タスク（HellaSwag [22]、PIQA [23]、WinoGrande [24]）におけるゼロショット性能、および検証損失（PT Valid Loss) を評価する。

SFT 後のモデルについては、OLMo [25] の設定に従い、以下の 3 つの主要な評価軸で評価を行う：指示追従能力（AlpacaEval [26]）、マルチタスク言語理解能力（MMLU [27]）、および真実性（TruthfulQA [28]）である。学習率減衰の有無が事前学習後と SFT 後の性能に与える影響の違いを明らかにするため、各段階において最適な性能を示した学習率減衰型スケジュールを基準とした相対性能指標を示す。事前学習に関しては、検証損失とゼロショットベンチマーク全タスクにおける平均精度（PT Task Avg）の両方を報告する。SFT に関しては、AlpacaEval、TruthfulQA、MMLU の各タスクにおける平均性能を報告する（SFT Task Avg）。

結果 表 2 は、学習段階ごとのモデル性能における逆転現象を示している。事前学習の評価指標に関しては、学習率減衰型スケジューラが $\alpha_{\text{pre}} = 0$ の場合に最も優れた性能を示した。具体的には、1B モデルにおいて Linear および WSD スケジューラは、$\alpha_{\text{pre}} = 0$ の条件下で PT Task Avg スコアの最高値を達成している。この傾向は、既存研究 [13] の報告と整合している。一方、SFT 後の性能に目を向けると、事前学習時点では学習率減衰型スケジューラに劣っていた WSO スケジューラが、両モデルサイズにおいて最高性能を達成した。以上の結果から、学習率減衰型スケジューラは事前学習段階の評価指標を改善する上では有効であるものの、SFT を含めた学習パイプライン全体を通じた最終的な性能においては、WSO スケジューラの方がより効果的であることが明らかとなった。

4 関連研究

LLM の事前学習における学習率スケジューリングについては、Cosine スケジューラが長らく標準的な手法として採用されてきた [1, 3]。近年では、継続事前学習への柔軟性を高めるために WSD スケジューラが提案されている [6, 8]。また、Bergsma ら [13] は学習率をゼロまで減衰させることで事前学習性能が向上することを示した。しかし、これらの研究はいずれも事前学習時の評価指標に基づいており、SFT 後の性能への影響は十分に検討されていない。本研究は、学習率スケジューラを SFT 後の下流タスク性能の観点から評価し、減衰を伴わない WSO スケジューラの有効性を明らかにした。

5 おわりに

本研究では、事前学習において有効性が広く報告されている学習率スケジューラについて、事後学習（SFT）後の性能に焦点を当て、その実用的な有効性を検証した。特に、既存の WSD スケジューラから減衰期間を排除し、学習率を一定に保つ WSO スケジューラについて検討を行った。実験の結果、WSO スケジューラによって学習されたモデルは、従来の学習率減衰型スケジューラによるモデルと比較して、SFT 後の下流タスクにおいて一貫して優れた性能を発揮することが明らかとなった。

WSO は減衰期間を必要としないため適用が容易であり、かつ事後学習後の性能向上をもたらす。したがって、高い適応性を持つモデルを構築することを目的とした大規模事前学習において、WSO は従来の学習率減衰型スケジューラに代わる有望な選択肢であると結論付ける。最後に、今後 LLM をスクラッチで構築・公開する際には、学習率減衰期間を経由されずに学習されたチェックポイントを提供することを推奨したい。これにより、LLM のチューニングに取り組む多くの研究者や実務家が、その高い適応性の恩恵を享受できるようになることを期待する。

This work is published without peer review and is licensed by the author(s) under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

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)，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」の助成を受けたものです.

参考文献

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, **Advances in Neural Information Processing Systems**, Vol. 33, pp. 1877–1901. Curran Associates, Inc., 2020.

[2] Teven Le, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. Bloom: A 176b-parameter open-access multilingual language model. **arXiv preprint arXiv:2211.05100**, 2022.

[3] 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.

[4] Alex Hägele, Elie Bakouch, Atli Kosson, Leandro Von Werra, Martin Jaggi, et al. Scaling laws and compute-optimal training beyond fixed training durations. **Advances in Neural Information Processing Systems**, Vol. 37, pp. 76232–76264, 2024.

[5] Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L Richter, Quentin Anthony, Gaël Varoquaux, and Sashank J Reddi. Simple and scalable strategies to continually pre-train large language models. **arXiv preprint arXiv:2403.08763**, 2024.

[6] Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. **arXiv preprint arXiv:2404.06395**, 2024.

[7] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. **arXiv preprint arXiv:2412.19437**, 2024.

[8] Kaiyue Wen, Zhiyuan Li, Jason S. Wang, David Leo Wright Hall, Percy Liang, and Tengyu Ma. Understanding warmup-stable-decay learning rates: A river valley loss landscape view. In **The Thirteenth International Conference on Learning Representations**, 2025.

[9] Kaiser Sun and Mark Dredze. Amuro & char: Analyzing the relationship between pre-training and fine-tuning of large language models. In Vaibhav Adlakha, Alexandra Chronopoulou, Xiang Lorraine Li, Bodhisattwa Prasad Majumder, Freda Shi, and Giorgos Vernikos, editors, **Proceedings of the 10th Workshop on Representation Learning for NLP (RepL4NLP-2025)**, pp. 131–151, Albuquerque, NM, May 2025. Association for Computational Linguistics.

[10] Jacob Mitchell Springer, Sachin Goyal, Kaiyue Wen, Tanishq Kumar, Xiang Yue, Sadhika Malladi, Graham Neubig, and Aditi Raghunathan. Overtrained language models are harder to fine-tune. **Proceedings of the International Conference on Machine Learning**, 2025.

[11] Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. **arXiv preprint arXiv:2501.00656**, 2024.

[12] 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.

[13] Shane Bergsma, Nolan Simran Dey, Gurpreet Gosal, Gavia Gray, Daria Soboleva, and Joel Hestness. Straight to zero: Why linearly decaying the learning rate to zero works best for LLMs. In **The Thirteenth International Conference on Learning Representations**, 2025.

[14] Meta. Llama-3.2-1b model card. https://huggingface.co/meta-llama/Llama-3.2-1B, 2024. Accessed: 2025-09-21.

[15] Meta. Meta-llama-3.1-8b model card. https://huggingface.co/meta-llama/Meta-Llama-3.1-8B, 2024. Accessed: 2025-09-21.

[16] 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.

[17] 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.

[18] AIat Meta. The llama 3 herd of models, 2024.

[19] 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.

[20] 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.

[21] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**, pp. 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[22] 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.

[23] 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.

[24] 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.

[25] Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, William Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah Smith, and Hannaneh Hajishirzi. OLMo: Accelerating the science of language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 15789–15809, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[26] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 5 2023.

[27] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In **International Conference on Learning Representations**, 2021.

[28] Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. 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. 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics.

This work is published without peer review and is licensed by the author(s) under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

表 3　実験で使用した SFT ハイパーパラメータ。指定された学習率に対してスイープを実施し、AlpacaEval 性能に基づいて最適な値を選択した。

ハイパーパラメータ	値
学習率	5.0×10^{-7}, 1.0×10^{-6}, 5.0×10^{-6}, 1.0×10^{-5}, 5.0×10^{-5}, 1.0×10^{-4}, 5.0×10^{-4}, 1.0×10^{-3}
グローバルバッチサイズ	128
学習率スケジューラ	ウォームアップ付き Cosine
最小学習率	0
最適化手法	AdamW
重み減衰	0.0
勾配クリッピング	1.0
ウォームアップステップ数	100
エポック数	1
訓練精度	bfloat16

A　学習率スケジューラの定式化

本節では、実験で用いた WSD、Cosine、および Linear スケジューラの定式化を示す。

WSD スケジューラ：ウォームアップ後、学習率は T_{stable} まで一定に保たれ、その後ステップ T において $\alpha_{\text{pre}}\cdot\eta_{\max}$ まで線形に減衰する：

$$\eta^{\text{WSD}}(t,\alpha_{\text{pre}})=\begin{cases}\eta_{\max}\cdot\frac{t}{T_{\text{warmup}}} & t\le T_{\text{warmup}}\\ \eta_{\max} & T_{\text{warmup}}<t\le T_{\text{stable}}\\ \eta_{\max}\cdot\left((1-\alpha_{\text{pre}})\cdot\frac{T-t}{T-T_{\text{stable}}}+\alpha_{\text{pre}}\right) & T_{\text{stable}}<t\le T\end{cases}\tag{5}$$

WSO スケジューラ：WSD において $\alpha_{\text{pre}}=1$ と設定することで得られる。ウォームアップ後、学習率は一定に保たれる：

$$\eta^{\text{WSO}}(t,\alpha_{\text{pre}})=\begin{cases}\eta_{\max}\cdot\frac{t}{T_{\text{warmup}}} & t\le T_{\text{warmup}}\\ \eta_{\max} & T_{\text{warmup}}<t\le T_{\text{stable}}\end{cases}\tag{6}$$

Cosine スケジューラ：ウォームアップ後、学習率は Cosine 関数に従って $\alpha_{\text{pre}}\cdot\eta_{\max}$ まで減衰する：

$$\eta^{\text{Cosine}}(t,\alpha_{\text{pre}})=\begin{cases}\eta_{\max}\cdot\frac{t}{T_{\text{warmup}}} & t\le T_{\text{warmup}}\\ \eta_{\max}\cdot\left(\alpha_{\text{pre}}+\frac{1-\alpha_{\text{pre}}}{2}\left(1+\cos\left(\frac{t-T_{\text{warmup}}}{T-T_{\text{warmup}}}\cdot\pi\right)\right)\right) & t>T_{\text{warmup}}\end{cases}\tag{7}$$

Linear スケジューラ：ウォームアップ後、学習率は線形に $\alpha_{\text{pre}}\cdot\eta_{\max}$ まで減衰する：

$$\eta^{\text{Linear}}(t,\alpha_{\text{pre}})=\begin{cases}\eta_{\max}\cdot\frac{t}{T_{\text{warmup}}} & t\le T_{\text{warmup}}\\ \eta_{\max}\cdot\left((1-\alpha_{\text{pre}})\cdot\frac{T-t}{T-T_{\text{warmup}}}+\alpha_{\text{pre}}\right) & t>T_{\text{warmup}}\end{cases}\tag{8}$$

すべてのスケジューラは第 2.3 節で説明したものと同じウォームアップ期間を使用し、それらの減衰は最小学習率係数 $\alpha_{\text{pre}}\in[0.0,1.0]$ によって制御される。

B　SFT 設定

すべてのモデルに対して、Tulu-3 SFT 混合データセットを用いて教師あり微調整を実施した。公式データセットには事前定義された訓練-検証分割が提供されていないため、訓練と検証をそれぞれ 9:1 の比率で独自に作成した。すべてのモデルに対して全パラメータ訓練を実施する。表 3 に、実験で使用したハイパーパラメータを示す。

This work is published without peer review and is licensed by the author(s) under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).