Continual Pre-Training of Large Language Models: How to (re)warm your model?

Kshitij Gupta [*1 2] **Benjamin Thérien** [*1 2] **Adam Ibrahim** [*1 2] **Mats Leon Richter** [1 2] **Quentin Anthony** [1 2] **Eugene Belilovsky** [3 1 2] **Irina Rish** [1 2] **Timothée Lesort** [1 2]

Abstract

Large language models (LLMs) are routinely pre-trained on billions of tokens, only to restart the process over again once new data becomes available. A much cheaper and more efficient solution would be to enable the continual pre-training of these models, i.e. updating pre-trained models with new data instead of re-training them from scratch. However, the distribution shift induced by novel data typically results in degraded performance on past data. Taking a step towards efficient continual pre-training, in this work, we examine the effect of different warm-up strategies. Our hypothesis is that the learning rate must be re-increased to improve compute efficiency when training on a new dataset. We study the warmup phase of models pre-trained on the Pile (upstream data, 300B tokens) as we continue to pre-train on SlimPajama (downstream data, 297B tokens), following a linear warmup and cosine decay schedule. We conduct all experiments on the Pythia 410M language model architecture and evaluate performance through validation perplexity. We experiment with different pre-training checkpoints, various maximum learning rates, and various warmup lengths within the first 50B tokens. Our results show that not warming up at all and keeping a constant learning rate gives the best performance for both downstream and upstream validation data.

[*]Equal contribution; authorship order determined by a coinflip [1]Department of Computer Science and Operation Research, Université de Montréal, Montréal, Canada [2]Mila, Montréal, Canada [3]Department of Computer Science and Software Engineering, Concordia University, Montréal, Canada. Correspondence to: Benjamin Thérien <benjamin.therien@umontreal.ca>.

Work presented at the ES-FoMo Workshop at the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

1. Introduction

Large pre-trained models have enabled massive performance improvements for many downstream tasks in vision (Kirillov et al., 2023; Oquab et al., 2023) and language (Brown et al., 2020; Zhao et al., 2023). However, training these foundation models is prohibitively expensive. Existing works aim to reduce the cost of large-scale model development by enabling low-cost hyperparameter optimization (Yang et al., 2022) or providing guidelines for maximizing performance under a given compute budget (Hoffmann et al., 2022). However, these works assume that models will be *trained from scratch*. As the amount of data available for pre-training is ever-growing, new and improved datasets (e.g. SlimPajama and SlimPajama (Together.xyz, 2023; Soboleva et al., 2023; Touvron et al., 2023)) will continue to become available. Should practitioners always combine existing datasets (e.g. Pile (Gao et al., 2020)) and *train from scratch* to obtain the best performance? Doing so would quickly become prohibitively expensive and fails to leverage existing pre-trained models.

Our approach circumvents the need for complete re-training by continuing to pre-train existing models on new data. We refer to this as "continual pre-training" and the goal is to minimize the loss on new data while maintaining low loss on previous data. Continual pre-training is a critical challenge since it can lead to catastrophic forgetting (French, 1999). Moreover, the potential long sequence of training stages may make common continual learning techniques such as replay (Rebuffi et al., 2017; Ostapenko et al., 2022) or regularisation (Kirkpatrick et al., 2017; Farajtabar et al., 2020) not enough compute efficient (Lesort et al., 2023). A simple and – from a compute cost perspective – scalable solution to limit forgetting in such situations is to (only) progressively decrease the learning rate every time new data becomes available (Mirzadeh et al., 2020; Winata et al., 2023). However, this solution is limited because repeatedly decreasing the learning rate would cause it to eventually become too small if the number of training stages becomes high.

In this work, we take a step towards efficient continual pre-training by studying how to re-increase a small learning rate to keep training a pre-trained language model on new data. We refer to this as *re-warming* the model. Re-warming the model should improve learning efficiency by avoiding a vanishing learning rate. We study warm-up strategies on Pythia 410M model with various amounts of data, maximum learning rates and different pre-trained checkpoints. This would allow a model trained initially on a large dataset to benefit from resuming training on a newer large dataset without having to retrain from scratch. In order to simulate this setting, we fix our initial pretraining dataset to be Pile and the newer dataset to be SlimPajama. We hope that this may guide the adaptation of existing LLMs to future new datasets.

Our results in our experimental setup show that:

1. Progressively increasing the learning rate to warm-up is not necessary but starting directly from the maximum learning rate creates an initial large spike in the loss (chaotic phase a.k.a stability gap) with no consequences later.
2. Increasing the maximum learning rate causes decreased performance when continuing to pre-training on RedPjama(downstream) or Pile (upstream/pre-training dataset).
3. Continual pre-training with the latest pretrained checkpoint improves performance.

2. Setup

In our setup, the upstream (or pre-training) dataset is the Pile (Gao et al., 2020). The downstream (or fine-tuning) dataset is SlimPajama (Soboleva et al., 2023). SlimPajama is an extensively deduplicated version of RedPajama (Together.xyz, 2023) which is built based on the LLama dataset (Touvron et al., 2023). In this work, we use "fine-tuning" and downstream continual pre-training interchangeably. However, in our continual pre-training setting, we note that the downstream dataset is on the scale of the previous pre-training dataset (i.e. very large, unlike many fine-tuning datasets).

The SlimPajama dataset is built from similar sources as the Pile but with a higher quantity of data. Therefore, some upstream data may be repeated during downstream pre-training. Our experimental setup is comparable to the setup of (Ash & Adams, 2020), where they train a classifier on half of the samples of a dataset first, and fine-tune it later on all samples. They show that warm starting for image classification is challenging. Using a model pre-trained on the Pile and continuing the pre-training on SlimPajama, we follow an analogous setup for causal language modeling.

Datasets – We use the Pile with the same weights as Black et al. (2022) for validation. We shuffle and randomly sample the SlimPajama dataset (Soboleva et al., 2023) to form the $\sim$297B token training dataset and $\sim$316M validation token dataset. We do not use replay. We use the same tokenizer as (Black et al., 2022) that is trained specifically on the Pile.

Table 1. Token counts and train data weights for our subsampled version of SlimPajama.

Dataset	Sampling %	Train	Val
StackExchange	2.0	9.95B	13.08M
Arxiv	2.5	13.77B	22.73M
Wikipedia	4.5	11.78B	15.79M
Book	4.5	14.22B	22.04M
Github	4.5	15.41B	22.42M
C4	15.0	78.49B	72.49M
Commoncrawl	67.0	153.25B	147.28M
Totals	100	296.86B	315.83M

Model – We use the 410M Pythia pre-trained on the Pile (Biderman et al., 2023), i.e. GPT-NeoX (Black et al., 2022) models. We do not use flash attention (Dao et al., 2022).

Hyperparameters – We use the AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.95$, $\epsilon = 10^{-8}$. The initial learning rate is $3 \cdot 10^{-4}$. We use cosine learning rate decay to a minimum of $0.1 \cdot \textit{MaxLr}$, and a weight decay of 0.1. While the learning rate schedule is based on the full downstream dataset size (297B tokens), we limit our experiments to the first 50B to minimize experimental compute cost. Consequently, in our experiments, the minimum learning rate is never reached on the downstream data. We set gradient clipping to 1.0. Training is conducted at half-precision (FP16), without dropout.

3. Related Work

Large Language Models: LLMs are usually trained with Adam (e.g., GPT3 (Brown et al., 2020), BLOOM (Scao et al., 2022), Gopher (Rae et al., 2021), Pythia (Biderman et al., 2023)) or AdamW (e.g., Chinchilla (Hoffmann et al., 2022), LLaMA (Touvron et al., 2023)). In all the aforementioned models, the learning rate schedule consists of a warm-up followed by a cosine decay to 10% of the maximum learning rate.

Unsupervised Continual Learning: In this paper, we investigate various warm-up strategies for the continual pre-training of LLMs. Continual pre-training uses a similar type of training objectives as continual self-supervised training. Self-supervised pre-training was also studied in vision datasets for image generation (Seff et al., 2017; Lesort et al., 2019; Zhai et al., 2019; Nguyen et al., 2018; Davari et al., 2022) or representation learning (Fini et al., 2022; Madaan et al., 2021; Rao et al., 2019). In language, continual pre-training was studied under the name of domain adaptation pre-training (Ke et al., 2023a; Scialom et al., 2022; Gururangan et al., 2021; Qin et al., 2022) where the new dataset comes from a new domain. Another setting is where different datasets are generated at different points in time (Han et al., 2021; Jin et al., 2022; Jang et al., 2021; 2022; Loureiro et al., 2022). In our setup, the scenario is closer to domain adaptation pre-training, because we do not take into account the temporality of data.

Monitoring Learning Rate for Continual Training of Language Models: In continual learning (CL), models are trained on sequences of datasets. Therefore, the data is not *independent and identically distributed* which can lead the model to lose plasticity or forget. In such situations, particular monitoring of the learning rate schedule can be beneficial. In CL of language models (Caccia et al., 2021; Ke et al., 2023a; Loureiro et al., 2022; Han et al., 2021; Loshchilov & Hutter, 2018; Scialom et al., 2022; Winata et al., 2023) different approaches have been evaluated: constant learning rate (Ke et al., 2023a; Scialom et al., 2022), progressive decrease (Winata et al., 2023) or warm-up then decrease (Caccia et al., 2021).

However, to the best of our knowledge, no existing work studies specifically the influence of the warm-up phase in the context of continual pre-training for large language models.

4. Continual Warm-up

4.1. How long to warm up?

In the literature, warm-up is usually conducted on at most 1% of the data (Zhao et al., 2023). In this experiment, we investigate if the results are sensitive to this hyper-parameter.

Setup: We experiment with different warm-up lengths for a schedule of 297B tokens: 0%, 0.5%, 1%, and 2% of the data and measure the performance after the first 50B tokens. From a different perspective, we could see this experiment as running a 1% warm-up on different amounts of data. We hypothesize that warming up for a larger number of iterations could lead to a smoother transition with subsequent performance improvements.

Results: The results of this experiment are provided in Fig. 1. They show that the amount of data used for warming up the learning rate does not significantly influence the perplexity on the downstream task (learning) or the upstream task (forgetting). These results invalidate our hypothesis that using more tokens for warm-up can smooth the transition and show that linear warmup is useless in this setting. Nevertheless, the model trained without any progressive warm up experiences an initial "choatic phase" causing a spike in the loss in its first few iterations of training, this phenomenon is also referred to as stability gap (Lange et al., 2023; Caccia et al., 2022).

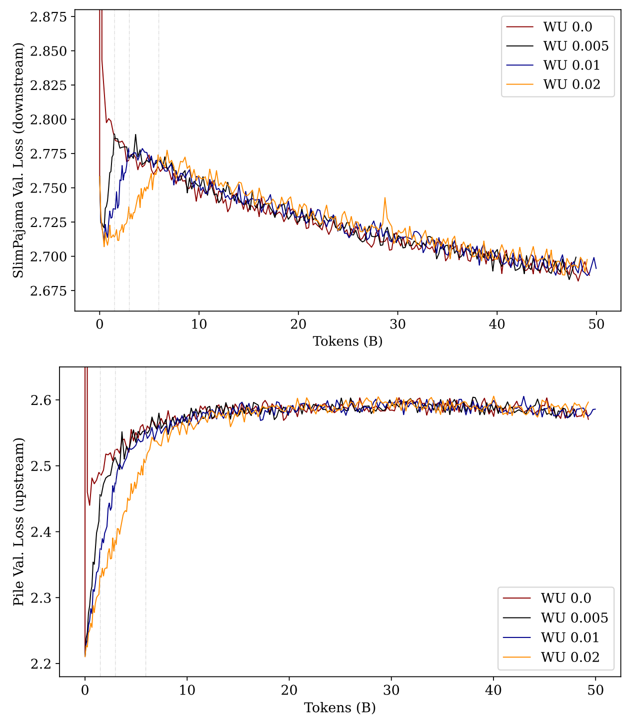

Figure 1. (*top*) Evolution of perplexity on SlimPajama while fine-tuning with various amounts of tokens for warm-up. (*bottom*) perplexity on the same experiments on the Pile validation set (upstream). $MaxLr = 3 \cdot 10^{-4}$, $MinLr = 0.1 \cdot MaxLr$. This figure shows that at that scale, the length of the warm-up phase does not significantly influence results.

4.2. How high to warm up?

One objective of re-warming the learning rate is to enable compute-efficient continual pre-training. A too-small learning rate may lead to inefficient learning on the downstream dataset. One important aspect of re-warming the learning rate is to decide how high to increase it. Therefore, in this experiment, we vary the maximum learning rate to assess its effect on performance.

Setup: We fix the length of the warm-up phase to the default amount of 1% of the training data and vary the maximum learning rate. We experiment with the default value of $3 \cdot 10^{-4}$ used for pre-training Pythia 410M (Biderman et al., 2023), $1.5 \cdot 10^{-4}$, and $6 \cdot 10^{-4}$. For the post-warmup cosine decay phase, we set the final learning rate to 10% of the maximum learning rate.

Results: The results of this experiment are provided in Fig. 2. We see that increasing the maximum learning rate leads to a corresponding increase in the perplexity on SlimPajama. The increase in perplexity on the downstream task with a higher maximum learning rate is also reflected in upstream perplexity as shown in Fig. 3. In fact, we observe that the model trained on 10B tokens with a constant learn-

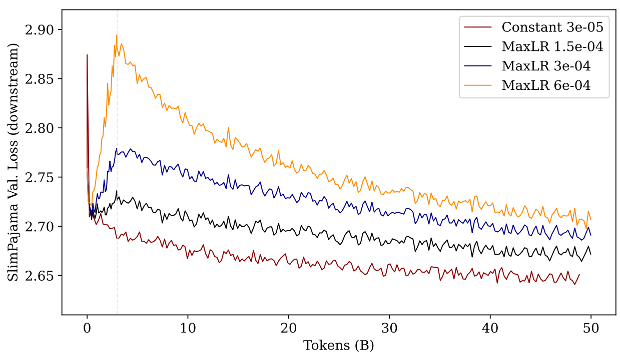

Figure 2. Evolution of loss on SlimPajama validation data with various maximum learning rates. The vertical dotted line represents the end of the warm-up phase. Growing the maximum learning rate consistently increases loss on downstream data, without quick improvements post warm-up. No warm-up seems to be the best option so far. Warm-up token=1% downstream tokens, $MinLr = 0.1 \cdot MaxLr$.

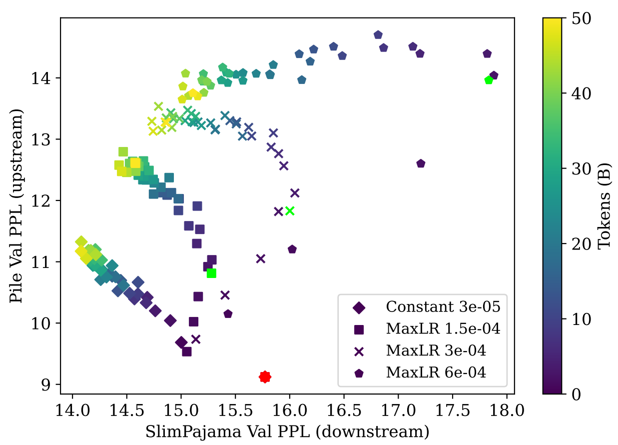

Figure 3. Perplexity downstream vs perplexity upstream, RP fine-tuning. Green points refer to the ends of the warm-up phases. The red point represents the perplexity before starting the downstream fine-tuning. Increasing the maximum learning rate is both bad for upstream and downstream. However, the performance decrease seems to stabilise after some training. Same setup as Fig. 2.

ing rate has a similar validation loss as the model trained on 50B tokens with a warm-up and maximum learning rate of $6 \cdot 10^{-4}$. While fine-tuning on SlimPajama, the model forgets what has been learned on the Pile leading to an increase in the Pile validation perplexity. This experiment suggests that increasing the maximum learning rate for warm-up is not a good strategy to improve downstream performance.

4.3. Re-warming on the same data

One hypothesis for the increase in loss is that the distribution shift between upstream and downstream data disturbs the training process. To assess this hypothesis, we apply our warm-up policy in a setting with no distribution shift. That is, we replicate our experiments from figures 2 and 3 by fine-tuning on Pile.

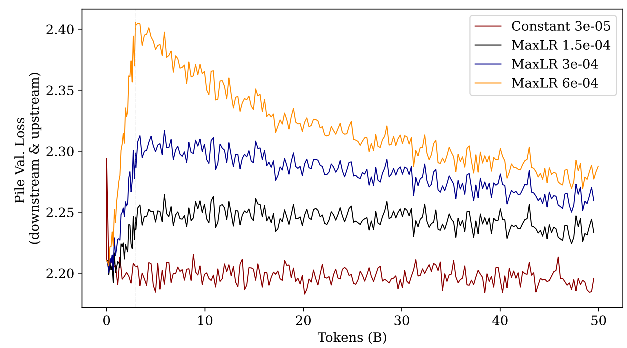

Figure 4. SlimPajama validation loss while fine-tuning again on the Pile. Warm-up phenomenon observed in Sec. 4.2 is also observed applied to fine-tuning again on the same data distribution. Warm-up token=1% downstream tokens, $MinLr = 0.1 \cdot MaxLr$.

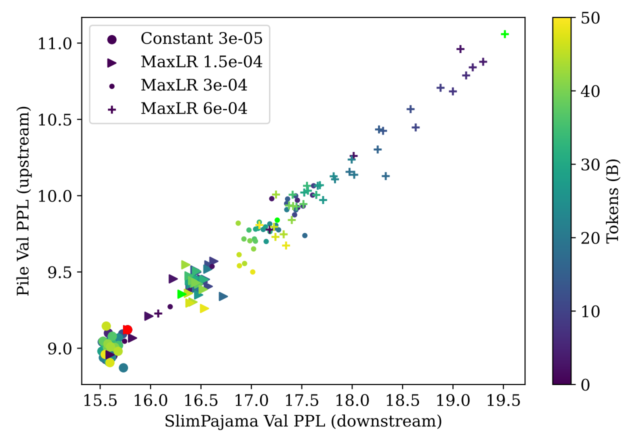

Figure 5. Perplexity on the Pile vs perplexity on SlimPajama when fine-tuning on the Pile with various maximum learning rates. Warm-up token=1% downstream tokens, $MinLr = 0.1 \cdot MaxLr$. Green points refer to the end of the warm-up phase.

Setup: In this experiment, instead of fine-tuning on SlimPajama data, we fine-tune again on the 50B tokens of the Pile data with the same parametrization of the warm-up policy as Sec. 4.2 experiments.

Results: Fig. 4, shows that re-warming the learning rate while continuing to pre-train on the Pile has a similar effect as re-warming on SlimPajama data Fig. 2 when looking at the downstream validation loss. This suggests that the distribution shift between Pile and SlimPajama is not to blame for the negative impact of re-warming the learning rate observed in sec. 4.2, leading us to believe that it may be related to complex optimization dynamics.

Fig. 5 shows that the training first increases perplexity on both the Pile and SlimPajama data but reduces after on both. Interestingly, Fig. 5 show a linear relationship between SlimPajama perplexity and the Pile perplexity when fine-tuning on the Pile, while it was not the case while fine-tuning on SlimPajama (Fig. 2). One possible explanation for this relationship is that models trained on Pile climb out of a minimum during warmup and return towards the same minimum as the learning rate is decayed, yielding the linear trend.

4.4. Evaluating Earlier Checkpoints

Setup: We select three checkpoints from model pre-training to test if warm-up strategies benefit from starting with non-converged checkpoints. Our hypothesis is that selecting checkpoints farther from convergence may benefit adaptation to the downstream task as these checkpoints may be located at more favorable points in the loss landscape

To select significantly different checkpoints, we compare the last pre-training checkpoint (i.e. Pythia 410M after $143,000$ iters), to an earlier checkpoint achieving a Pile validation loss near the maximum Pile validation loss attained by all models in Fig. 1 (bottom) (~ 2.5), and a third checkpoint in between the two other checkpoints.

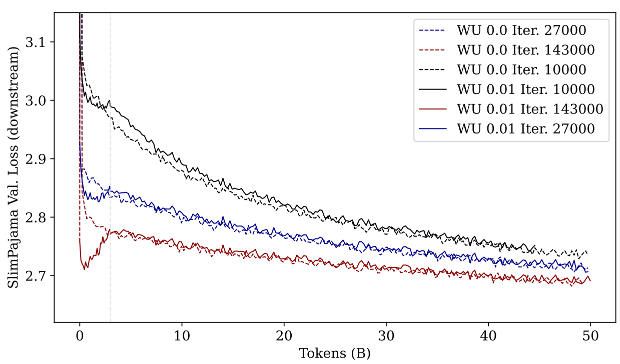

Figure 6. Pile validation loss of models trained from the fully converged checkpoint, the upstream saturation point, and $1/2$ of the upstream saturation point. Black colour designs for the earlier checkpoint, red colour the latest checkpoint and blue colour the in-between one.

Results: The evolution of the validation losses on SlimPajama are provided in Fig. 6 and the evolution of the validation losses on the Pile is provided in appendix A. We see in Fig. 6 that, in our setup, selecting earlier checkpoints for later fine-tuning does not lead to improvement in downstream performance. Therefore, selecting the latest checkpoint is the best option. We can conclude that the pre-training did not lead the model into a loss of plasticity that would make the model difficult to re-warm.

Local conclusion: The experiments conducted in this section led to the conclusion that re-warming the pre-trained model on new data is a challenging task, even when the downstream data is of similar provenance to the upstream data. Our results show that the amount of tokens used for warm-up does not significantly alter performance and that growing the maximum learning rate or selecting earlier checkpoints decreases performance on both upstream and downstream data.

5. Discussion / Limitation

Data similarity and overlapping: In our experimental setup, upstream and downstream data have a high similarity, notably because of data overlap. Since in continual learning, different types of shifts can lead to variations in performance (Lesort et al., 2021), our results may not generalize to setups with different distribution shifts, such as language domain adaptation pre-training setups (Xu et al., 2019; Gururangan et al., 2020; Ke et al., 2023a; Chakrabarty et al., 2019; Ke et al., 2023b). Nevertheless, comparing Fig. 3 and Fig. 5, we see that the results are not identical when fine-tuning on the Pile or when fine-tuning on SlimPajama. A possible explanation is that even a slight shift in data distribution can lead to a significant perturbation of the learning dynamics. For example, in the context of image classification, Igl et al. (2020) show how a sudden transition of 10 to 20 % of the labels in the dataset can have a significant impact on the downstream performance (see Fig. 5 of (Igl et al., 2020)).

Experiments Scale: As described in Sec. 2, we did not train until the end of the learning rate scheduling or until convergence. Indeed, after warm-up, the learning rate progressively decreases, following a coside decay, until the end of the 297B tokens to 10% of the maximum learning rate. Since our experiments are only on the first 50B tokens, the decrease in the learning rate is far from reaching this minimum value. While running experiments until the end of the data would have provided us with more definitive results, our experimental setup was designed to investigate early behaviour. Our hypothesis is that the difference in performance after the 50B can provide important insight into how the different schedules may perform at the end of training. However, the trends observed after 50B tokens do not provide a definitive answer. For instance, in Fig. 3, the model with a maximum learning rate of $6 \cdot 10^{-4}$ could eventually adopt a trend closer to that of curves seen in Fig. 5 and improve over the other models when trained with the full token budget on SlimPajama.

Finally, while this is a preliminary study, in future work, we plan to verify whether our conclusions hold at different model scales (e.g., 3B and 7B).Moreover, we plan to test our models throughout using benchmarks such as HELM (Liang et al., 2022) or Harness (Gao et al., 2021) instead of only loss or perplexity, as these benchmarks can provide important insight into the evolution of model capabilities. We also plan to investigate the benefits of using continual learning approaches such as replay, as done in (Scialom et al., 2022).

6. Conclusion

While warming up the learning rate should be helpful or even necessary for continual learning at scale, in our experiments, a warm-up seems to always degrade performance on both downstream and upstream data. More research is needed to find the right learning rate scheduling policy for long-term continual learning of large language models and test it to other types of models and data distribution shifts.

Software and Data

GPT-NeoX (Andonian et al., 2021), DeepSpeed (Rasley et al., 2020), nccl (NVIDIA, 2016), Apex (NVIDIA, 2019), Pytorch (Paszke et al., 2017), HuggingFace Transformers library (Wolf et al., 2020).

Acknowledgements

This work was made possible by computing resources on the Summit supercomputer within the INCITE program award "Scalable Foundation Models for Transferable Generalist AI". We would also like to acknowledge funding from the FRQNT Doctoral (B2X) scholarship [B.T.], the scholarship for Artificial Intelligence of Université de Montréal's Etudes Supérieures et Postdoctorales, This work was supported by a fellowship within the IFI program of the German Academic Exchange Service (DAAD).

References

Andonian, A., Anthony, Q., Biderman, S., Black, S., Gali, P., Gao, L., Hallahan, E., Levy-Kramer, J., Leahy, C., Nestler, L., Parker, K., Pieler, M., Purohit, S., Songz, T., Phil, W., and Weinbach, S. GPT-NeoX: Large Scale Autoregressive Language Modeling in PyTorch, 8 2021. URL https://www.github.com/eleutherai/gpt-neox.

Ash, J. and Adams, R. P. On warm-starting neural network training. *Advances in neural information processing systems*, 33:3884–3894, 2020.

Biderman, S., Schoelkopf, H., Anthony, Q., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., et al. Pythia: A suite for analyzing large language models across training and scaling. *arXiv preprint arXiv:2304.01373*, 2023.

Black, S., Biderman, S., Hallahan, E., Anthony, Q., Gao, L., Golding, L., He, H., Leahy, C., McDonell, K., Phang, J., Pieler, M., Prashanth, U. S., Purohit, S., Reynolds, L., Tow, J., Wang, B., and Weinbach, S. Gpt-neox-20b: An open-source autoregressive language model, 2022.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pp. 1877–1901, 2020. URL https://arxiv.org/abs/2005.14165.

Caccia, L., Xu, J., Ott, M., Ranzato, M., and Denoyer, L. On anytime learning at macroscale. *arXiv preprint arXiv:2106.09563*, 2021.

Caccia, L., Aljundi, R., Asadi, N., Tuytelaars, T., Pineau, J., and Belilovsky, E. New insights on reducing abrupt representation change in online continual learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=N8MaByOzUfb.

Chakrabarty, T., Hidey, C., and McKeown, K. IMHO finetuning improves claim detection. In *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. 558–563, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1054. URL https://aclanthology.org/N19-1054.

Dao, T., Fu, D., Ermon, S., Rudra, A., and Ré, C. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

Davari, M., Asadi, N., Mudur, S., Aljundi, R., and Belilovsky, E. Probing representation forgetting in supervised and unsupervised continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16712–16721, 2022.

Farajtabar, M., Azizan, N., Mott, A., and Li, A. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 3762–3773. PMLR, 2020. URL https://arxiv.org/abs/1910.07104.

Fini, E., da Costa, V. G. T., Alameda-Pineda, X., Ricci, E., Alahari, K., and Mairal, J. Self-supervised models are continual learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9621–9630, 2022.

French, R. M. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999. ISSN 13646613. doi: 10.1016/S1364-6613(99)01294-2. URL https://www.sciencedirect.com/science/article/abs/pii/S1364661399012942.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Gao, L., Tow, J., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., McDonell, K., Muennighoff, N., Phang, J., Reynolds, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. A framework for few-shot language model evaluation, September 2021. URL https://doi.org/10.5281/zenodo.5371628.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020. URL https://arxiv.org/abs/2004.10964.

Gururangan, S., Lewis, M., Holtzman, A., Smith, N. A., and Zettlemoyer, L. Demix layers: Disentangling domains for modular language modeling. *arXiv preprint arXiv:2108.05036*, 2021. URL https://arxiv.org/abs/2108.05036.

Han, R., Ren, X., and Peng, N. ECONET: Effective continual pretraining of language models for event temporal reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5367–5380, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.436. URL https://aclanthology.org/2021.emnlp-main.436.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. URL https://arxiv.org/abs/2203.15556.

Igl, M., Farquhar, G., Luketina, J., Boehmer, W., and Whiteson, S. The impact of non-stationarity on generalisation in deep reinforcement learning. *arXiv preprint arXiv:2006.05826*, 2020. URL https://arxiv.org/abs/2006.05826.pdf.

Jang, J., Ye, S., Yang, S., Shin, J., Han, J., Kim, G., Choi, S. J., and Seo, M. Towards continual knowledge learning of language models. *arXiv preprint arXiv:2110.03215*, 2021. URL https://arxiv.org/abs/2110.03215.

Jang, J., Ye, S., Lee, C., Yang, S., Shin, J., Han, J., Kim, G., and Seo, M. Temporalwiki: A lifelong benchmark for training and evaluating ever-evolving language models. 2022.

Jin, X., Zhang, D., Zhu, H., Xiao, W., Li, S.-W., Wei, X., Arnold, A., and Ren, X. Lifelong pretraining: Continually adapting language models to emerging corpora. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pp. 1–16, May 2022. doi: 10.18653/v1/2022.bigscience-1.1. URL https://aclanthology.org/2022.bigscience-1.1.

Ke, Z., Shao, Y., Lin, H., Konishi, T., Kim, G., and Liu, B. Continual pre-training of language models. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=m_GDIItaI3o.

Ke, Z., Shao, Y., Lin, H., Xu, H., Shu, L., and Liu, B. Adapting a language model while preserving its general knowledge. *arXiv preprint arXiv:2301.08986*, 2023b.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., and Girshick, R. Segment anything. *arXiv:2304.02643*, 2023.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proc. of the national academy of sciences*, 2017. URL https://www.pnas.org/content/pnas/114/13/3521.full.pdf.

Lange, M. D., van de Ven, G. M., and Tuytelaars, T. Continual evaluation for lifelong learning: Identifying the stability gap. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Zy350cRstc6.

Lesort, T., Caselles-Dupré, H., Garcia-Ortiz, M., Goudou, J.-F., and Filliat, D. Generative models from the perspective of continual learning. In *IJCNN - International Joint Conference on Neural Networks*, Budapest, Hungary, Jul 2019. URL https://hal.archives-ouvertes.fr/hal-01951954.

Lesort, T., Caccia, M., and Rish, I. Understanding continual learning settings with data distribution drift analysis. *arXiv preprint arXiv:2104.01678*, 2021.

Lesort, T., Ostapenko, O., Rodriguez, P., Arefin, M. R., Misra, D., Charlin, L., and Rish, I. Challenging common assumptions about catastrophic forgetting. 2023.

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1711.05101.

Loureiro, D., Barbieri, F., Neves, L., Espinosa Anke, L., and Camacho-collados, J. TimeLMs: Diachronic language models from Twitter. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 251–260, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-demo.25. URL https://aclanthology.org/2022.acl-demo.25.

Madaan, D., Yoon, J., Li, Y., Liu, Y., and Hwang, S. J. Representational continuity for unsupervised continual learning. In *International Conference on Learning Representations*, 2021.

Mirzadeh, S. I., Farajtabar, M., Pascanu, R., and Ghasemzadeh, H. Understanding the role of training regimes in continual learning. *Advances in Neural Information Processing Systems*, 33:7308–7320, 2020.

Nguyen, C. V., Li, Y., Bui, T. D., and Turner, R. E. Variational continual learning. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1710.10628.

NVIDIA. NVIDIA Collective Communication Library (NCCL). https://docs.nvidia.com/deeplearning/sdk/nccl-developer-guide/docs/index.html, 2016. Accessed: July 7, 2023.

NVIDIA. Pytorch extension with NVIDIA-maintained utilities to streamline mixed precision and distributed training. https://nvidia.github.io/apex/, 2019. Accessed: July 7, 2023.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.-Y., Xu, H., Sharma, V., Li, S.-W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. Dinov2: Learning robust visual features without supervision, 2023.

Ostapenko, O., Lesort, T., Rodríguez, P., Arefin, M. R., Douillard, A., Rish, I., and Charlin, L. Continual learning with foundation models: An empirical study of latent replay, 2022.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic Differentiation in PyTorch. 2017.

Qin, Y., Zhang, J., Lin, Y., Liu, Z., Li, P., Sun, M., and Zhou, J. Elle: Efficient lifelong pre-training for emerging data. *arXiv preprint arXiv:2203.06311*, 2022. URL https://arxiv.org/abs/2203.06311.

Rae, J. W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, F., Aslanides, J., Henderson, S., Ring, R., Young, S., et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021. URL https://arxiv.org/abs/2112.11446.

Rao, D., Visin, F., Rusu, A. A., Teh, Y. W., Pascanu, R., and Hadsell, R. Continual unsupervised representation learning. 2019. URL https://arxiv.org/pdf/1910.14481.pdf.

Rasley, J., Rajbhandari, S., Ruwase, O., and He, Y. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3505–3506, 2020.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017. URL https://arxiv.org/abs/1611.07725.

Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., Gallé, M., et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022. URL https://arxiv.org/abs/2211.05100.

Scialom, T., Chakrabarty, T., and Muresan, S. Fine-tuned language models are continual learners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6107–6122, 2022.

Seff, A., Beatson, A., Suo, D., and Liu, H. Continual learning in generative adversarial nets. *CoRR*, abs/1705.08395, 2017. URL http://arxiv.org/abs/1705.08395.

Soboleva, D., Al-Khateeb, F., Myers, R., Steeves, J. R., Hestness, J., and Dey, N. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama. https://www.cerebras.net/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama, 2023. URL https://huggingface.co/datasets/cerebras/SlimPajama-627B.

Together.xyz. Redpajama: An open source recipe to reproduce llama training dataset, 2023. URL https://github.com/togethercomputer/RedPajama-Data.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. URL https://arxiv.org/abs/2302.13971.

Winata, G. I., Xie, L., Radhakrishnan, K., Wu, S., Jin, X., Cheng, P., Kulkarni, M., and Preotiuc-Pietro, D. Overcoming catastrophic forgetting in massively multilingual continual learning. *arXiv preprint arXiv:2305.16252*, 2023.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. Transformers: State-of-the-Art Natural Language Processing. pp. 38–45. Association for Computational Linguistics, October 2020. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Xu, H., Liu, B., Shu, L., and Yu, P. S. Bert post-training for review reading comprehension and aspect-based sentiment analysis. *arXiv preprint arXiv:1904.02232*, 2019.

Yang, G., Hu, E. J., Babuschkin, I., Sidor, S., Farhi, D., Pachocki, J., Liu, X., Chen, W., and Gao, J. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. In *NeurIPS 2021*, March 2022. URL https://www.microsoft.com/en-us/research/publication/tuning-large-neural-networks-via-zero-shot-hyperparameter-transfer/.

Zhai, M., Chen, L., Tung, F., He, J., Nawhal, M., and Mori, G. Lifelong gan: Continual learning for conditional image generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2759–2768, 2019.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023. URL https://arxiv.org/abs/2303.18223.

A. Upstream loss when fine-tuning various checkpoints.

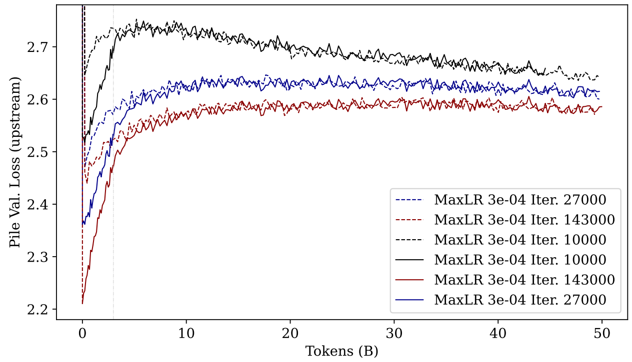

Figure 7. Pile validation loss of models trained from the fully converged checkpoint, the upstream saturation point, and $1/2$ of the upstream saturation point. The experiments for this figure are described in Sec. 4.4.

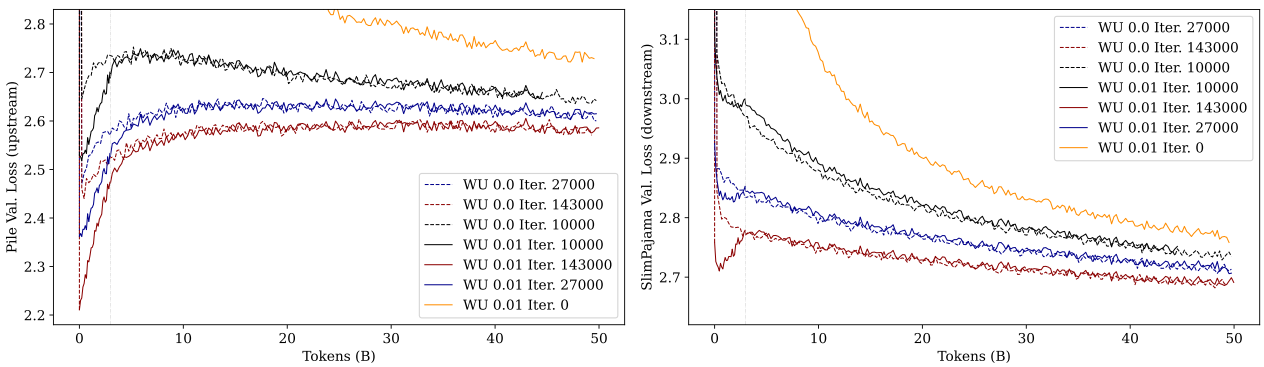

Figure 8. Training from a pre-trained checkpoint achieves lower Pile and SlimPajama validation loss faster than training from scratch.