The Student Becomes the Teacher: A Reverse Distillation Approach for Data-Efficient Knowledge Transfer Between Language Models

Anonymous submission

Abstract

The Transformer architecture has revolutionized the Natural Language Processing (NLP) community by providing immense gains in accuracy for several NLP tasks, especially through the creation of Large Language Models (LLMs). Transformers will not remain state-of-the-art, however. As superior architectures, especially those implemented on neuromorphic accelerators, become available, we will need cross-architecture pretraining methods that efficiently transfer knowledge from outdated machine learning models to more advanced ones. This paper presents *superstilling*, an adaptation of Hinton et al.'s well-known distillation technique to transfer parametric knowledge between models with vastly different sizes, forward propagation methods, and weight update algorithms. We validate this method on one of these three possibilities - transferring knowledge from a small model to a much larger one - and show that superstilling can decrease sample complexity by up to 50% during early pretraining, and by more than 10% at the knowledge saturation point.

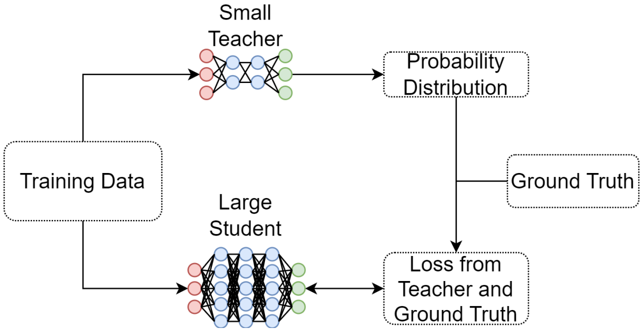

Figure 1: Superstilling: In this reversed distillation paradigm, the output probabilities of the Teacher model are used in addition to one-hot ground truth labels during model pretraining. The cross-entropy loss from both sources is then weighted and summed to produce the final loss.

Introduction

The Transformer architecture (Vaswani et al. 2017) shows enormous promise in domains such as common-sense reasoning, few-shot learning, conversational text generation, and API management, and is gaining widespread traction in commercial industry. And yet, despite this prowess, transformers will almost certainly be supplanted by newer, more powerful machine learning models. Retentive networks, introduced by Sun et al. (2023), leverage a chunkwise recurrent algorithm with parallelization benefits comparable to the transformer's famed attention mechanisms while requiring far less computation during inference. Other researchers have proposed adaptations of attention mechanisms for spiking neural networks, a combination which shows promise for future implementation on low-power neuromorphic hardware (Zhou et al. 2023; Li, Lei, and Yang 2022). As these and other innovations gain traction, a host of new Large Language Models (LLMs) will inevitably be brought online, each of them requiring Terabytes of data and millions of dollars to train. Even without architectural innovations, prior trends suggest that transformer-based LLMs will continue to increase in capabilities with more size, data, and compute (Kaplan et al. 2020).

Ironically, most of these new LLMs will be trained on the same or on similar data as their less advanced counterparts, due to limitations in the amount of high-quality text data available for training language models of such scale (Villalobos et al. 2022; Muennighoff et al. 2023). This fact raises an interesting question. If we cannot copy the model weights directly to the new LLM, can we instead leverage the learned internal representations to transfer pretraining information more effectively than training the new LLM from scratch?

This paper introduces an application of model distillation (Hinton, Vinyals, and Dean 2015) toward data-efficient training of large-scale language models. Distillation is a process whereby behaviors learned by a large machine learning model, called the Teacher, are evoked within a much smaller model, called the Student. This is achieved by providing the student model with the full output probability distribution of the teacher, essentially increasing the amount of information provided by each training example and thus simplifying the loss surface of the task to be learned. Typically, distilled models are able to replicate or nearly-replicate the performance of the teacher model by replicating its output probabilities, even when the student would not be able to learn the task directly from the original training data. Model distillation has been used to reduce inference costs by as much as 40%, while still retaining about 97% of the original capabilities (Sanh et al. 2019). In some cases, successful distillation has been achieved using a student that was only 2.6% as large as the initial model (Costa-jussà et al. 2022).

Inspired by the remarkable ability of distillation to pack more information into fewer parameters, our research team began to wonder: Rather than using this process to reduce model size, could we instead use it to improve training efficiency? Specifically, we asked whether model distillation could be used to train new LLMs in a data efficient and architecturally agnostic way.

We term this reverse distillation method *superstilling*. In contrast to the standard use of model distillation as a technique for reducing model size, our superstilling method (Fig. 1) is designed to improve sample efficiency by leveraging knowledge from a deprecated model in the training of its successor. In this paradigm, the roles of the teacher and student are reversed, and the output distribution of a smaller and/or representationally inferior teacher model is used to train a student model with superior representational capacity.

Critically, unlike fine-tuning or direct weight transfer, our superstilling method imposes no constraints on the relative architectures of the two models. Thus, the method may be feasibly employed whenever a company or group of researchers adopts a new machine learning architecture, expands their compute budget, or acquires new resources. Superstilling can be applied even when the teacher's original training data is unwanted or no longer available, for example by pairing the teacher's output probability distribution with ground truth knowledge from a different data source. It can therefore be applied in federated learning scenarios as well as scenarios where the teacher model's weights are available solely via API access.

The remainder of this paper is structured as follows: (1) We present the superstilling method and apply it to train transformer-based language models with up to 160M parameters, (2) We validate our superstilling approach by comparing sample efficiency with respect to perplexity, and show that gains of 10% to 50% can be achieved, (3) We discuss practical guidelines for the application of superstilling in commercial contexts, including methods for reduced compute cost over successive superstilling iterations, as well as possibilities for a slight speedup in training time under specific conditions.

Background and Related Work

The Transformer architecture (Vaswani et al. 2017) is currently the backbone of large-scale language model research, and has been applied with great success in domains ranging common-sense reasoning (Bosselut et al. 2019; Rytting and Wingate 2021) to few-shot learning (Brown et al. 2020; Radford et al. 2019), conversational text generation (Zhang et al. 2020; OpenAI 2023), and API management (Schick et al. 2023). Like models that have come before, however, it will likely be supplanted by novel innovations in the field. In addition to advancements mentioned in the introduction, current lines of research are exploring modified machine learning architectures that incorporate external knowledge via graph embeddings; are embodied and/or knowledge-grounded (Driess et al. 2023; Zakka et al. 2023; Nakano et al. 2021); incorporate visual input (Zhang et al. 2021; Alayrac et al. 2022; Yang et al. 2023); leverage architectures based on squared RELUs and depth-wise attentional convolutions (So et al. 2021); or combine attention mechanisms with fast recurrence for more efficient training (Lei 2021).

Training Cost of Large-Scale Language Models

The efficiency of LLM model training is an active area of machine learning research (Shen et al. 2022; Ma et al. 2019; Lei 2021). GPT-3, a powerful and popular LLM, was trained on a 500-billion token dataset (Brown et al. 2020). Training for BERT, another common language model, required a dataset of 3.3 billion words (Kenton and Toutanova 2019). Given that transformer-based LMs have been shown to continually increase in capabilities with more size, data, and compute (Kaplan et al. 2020), it seems likely that the size of the average large-scale language model will continue to increase. Of particular importance to our research is the fact that many of these LLMs are trained using similar or identical data.

Many researchers are exploring methods to reduce the compute cost of LLM training. Of particular interest is the work of (Korthikanti et al. 2023), who introduced methods for reducing the re-computation of network activations, resulting in significant memory savings. Exploring an alternate approach, So et al. (2021) performed a search over variations to the transformer architecture in order to find one that might be more efficient. Primer, the variant architecture they discovered, significantly reduced training costs. These efforts are orthogonal to our own in method, but are similar in aim.

Model Distillation

Model Distillation, introduced by Hinton, Vinyals, and Dean (2015), is a process that transfers the knowledge contained in one neural network or ensemble of networks to another (typically smaller) neural network. Distillation functions by using the entire probability distribution produced by a previously trained model, called the Teacher, to calculate the training loss for smaller Student model. The student model learns both from the Teacher's output as well as from the ground truth labels contained in the training data. Through this process, the student learns from the teacher not only the most likely output given the input, but also the relative likelihood of all other tokens or possible output categories. This allows the student to learn much more information per training sample. Distillation has been used frequently and with great success. For example, Sanh et al. (2019) used it to create a smaller version of BERT, leading to a 40% reduction in model size while retaining about 97% of the model's performanece. Costa-jussà et al. (2022) utilized Distillation for a set of machine-translation models created as part of Facebook's No Language Left Behind program. The distilled models, trained specifically on low-resource languages, were highly capable despite being only about 2.5% as large as the initial model. Gordon and Duh (2020) observed similar benefits from distillation in machine translation across multiple domains of language use.

Reverse Distillation

Reverse Distillation is a relatively small research area, and much of the existing literature resides in the domain of computer vision. For example, Yuan et al. (2020) showed that there is a link between knowledge distillation and label smoothing regularization. While they experimented with reverse distillation, the core purpose of their experiments was to show that various distillation methods – including reverse distillation – can improve accuracy on vision tasks. They do not address LLMs. Chaudhury et al. (2021) similarly show that reverse distillation leads to performance gains in the field of computer vision and also discuss a link between distillation and label smoothing. Jiang and Deng (2023) use reverse distillation to calibrate the confidence of neural networks, making them safer to use in high-risk scenarios. To combat the effects of noise inherent in natural data, Raipuria, Bonthu, and Singhal (2021) experiment with reverse distillation, determining that it is quite effective.

To our knowledge, the current work comprises the first exploration of reverse distillation with the specific aim of reducing sample complexity. In addition to focusing on the critical and timely challenge of improving efficiency for large-scale language models, we also provide a validation of prior observations within a new domain, and identify several reasons why this method may be particularly applicable within the economic landscape of LLM development.

Method

We conducted experiments at two scales to investigate the effects of superstilling on LLM sample complexity. At the smaller scale, with models consisting of 53 million parameters, we show that superstilling gains much more knowledge per training sample than traditional pretraining methods. We also show that superstilling is most useful in the earliest epochs of training, with efficiency gains leveling off as the student model nears the teacher model in accuracy.

We also conducted experiments using models with up to 161 million parameters – about the size of GPT-1 (Radford et al. 2018). Results from these experiments show that superstilling continues to increase sample efficiency on models of more significant size. We show in these experiments that superstilling can in certain cases achieve better performance in less clock time.

Training Data

Our small-scale experiments leveraged a limited dataset consisting of 5 classic books acquired from Project Gutenberg (Hart 2023): *Little Women*, *Peter Pan*, *The Railway Children*, *Black Beauty*, and *The Blue Fairy Book*. These works were selected as a small but representative sample of Western literary aesthetics from the 19th and 20th centuries. Results from these initial experiments were validated using a larger dataset containing over 1500 books (118 million tokens). These datasets as compiled contain only modern English text, a fact which simplifies training and limits the vocabulary size. (In the future, extending this work to other languages would be desirable.)

Training data was preprocesed by converting all characters to lowercase, and was then tokenized using a standard spaCy tokenizer. For the larger dataset, vocabulary size was clipped at 50,000 tokens, with the least common words pruned.

Model Specifications

We conducted both small- and large-scale experiments to determine the effectiveness of superstilling in reducing sample complexity. The smaller experiments used a teacher model comprised of a 2-layer transformer decoder-only mode trained on 5 classic children's books. The student model was exactly twice the size of the teacher (i.e. 4 transformer decoder blocks rather than 2) and was trained on the same dataset. Both teacher and student had an inner model dimension of 768, feed-forward dimension of 3072, and 12 attention heads per layer. They were trained using a set of hyperparameters in line wit GPT-1 (Radford et al. 2018): a learning rate of 3.0e-4 and an Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 1.0$e-8. Context window was 256 tokens, and the batch size was 16.

The large-scale experiments used a 4-layer model as the teacher with a 12-layer model as the student. Both models were trained on the the larger dataset (1500 books) with the same hyperparameters and model dimensions as the small-scale experiments, except that the context window was extended to 512 tokens and the batch size was increased to 32.

In both the small- and large-scale experiments, a baseline model equal in size to the student was trained for comparison, using the same hyperparameters and context size as the superstilled models.

Preliminary experiments suggested that it might be beneficial to superstill only for the first 20-30% of training, due to the declining benefit of teacher-supplied output distributions as the student's capabilities approach (or exceed) those of the teacher. The experiments reported here leverage this insight, and report efficiency results for each of two transition points. Early in the training cycle, superstilling is leveraged; after transition, the student model is trained only on the original data.

Superstilling Procedure

Following Hinton, Vinyals, and Dean (2015), we train the student model with the same data as the teacher using the following loss function:

$$L = w_1 * T^2 * l_t + w_2 * l_d$$

where T is the temperature used to soften the teacher's outputs, and l_t and l_d are the cross-entropy losses of the student model with the teacher model and the data, respectively. w_1 and w_2 weight the two different aspects of the loss. We found via a coarse parameter search that using a temperature $T = 6$ and setting $w_1 = 0.6$ and $w_2 = 0.4$ worked well.

In our small-scale experiments, we first train a 2-layer transformer for 50 epochs. This is the teacher model. We then trained two 4-layer student models (53M parameters) via the superstilling method; one student model superstilled for all 50 training epochs while the other transitioned to traditional cross-entropy loss after epoch 14. We also trained a

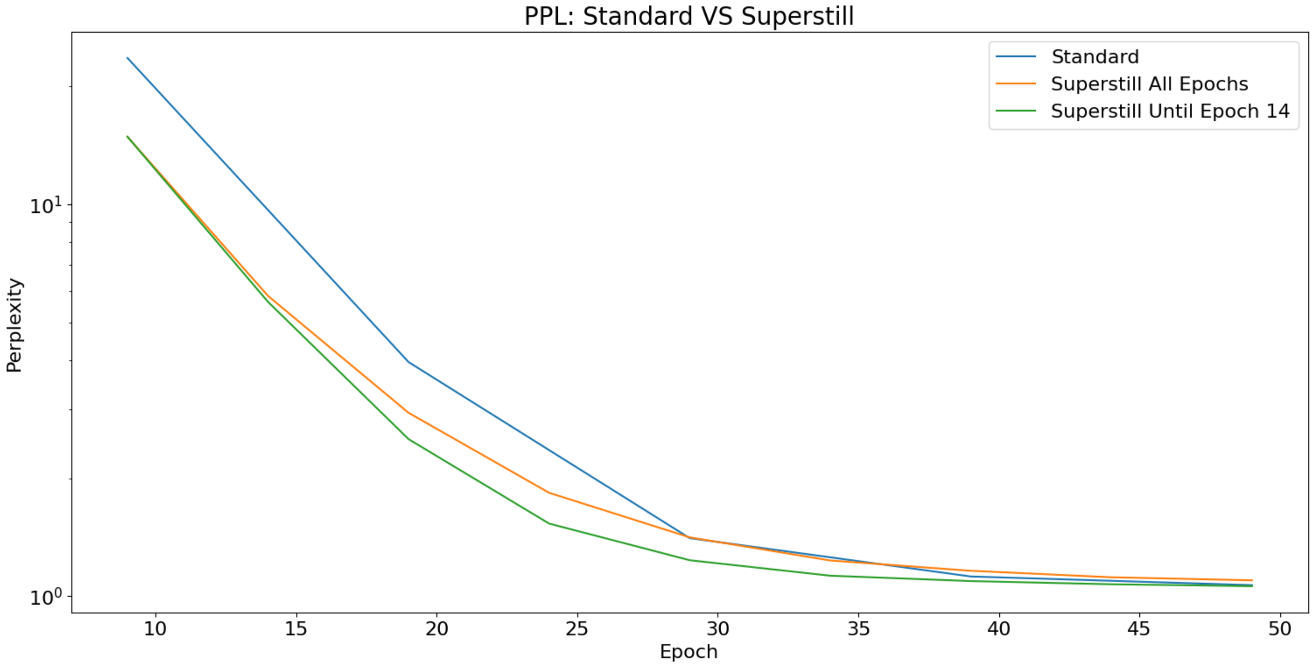

Figure 2: Model perplexity calculated over a datset containing 5 children's books. The vertical axis is plotted on a log-scale. This experiment was run using a small-scale transformer architecture with 4 layers and 53 million trainable parameters. Perplexity was measured every 10 epochs.

4-layer baseline model using the standard method (i.e. cross-entropy loss with respect to the training data).

The large-scale experiments were conducted in a similar manner, using a 4-layer teacher model and 12-layer student model (161M parameters). Due to the increased dataset size and existing resource constraints, the number of training epochs was capped at 9. To compare the results of each set of models, we compute perplexity with respect to the training data. In the case of the large models, we compute perplexity on the first 1% of the data.

Evaluation

Knowledge acquisition for each model was measured in terms of perplexity, or the ability of the model to predict its own training data. Perplexity is a well-established metric in the language model literature, and can be loosely interpreted as the extent to which a model is "surprised" by a statistical sample (Jelinek et al. 1977). In our experiments, lower perplexity scores suggest that the model is approaching knowledge saturation. In contrast, a high perplexity suggests that the model is unable to predict its own training data, and hence is not well adapted to its task.

Because perplexity is a function not only of the model's suitedness for the task, but also of the inherent uncertainty within the task itself, perplexity values will vary across evaluation datasets. In the case of language model training, larger datasets will tend to produce higher perplexity values, even when the model has mastered its task quite well. This occurs because language is inherently nondeterministic, with many possible completions for a given input sequence. Larger datasets expose the model to a high number of these completions, and hence reduce its ability to predict any specific completion with confidence.

Table 1: Final perplexity achieved by small-scale models

Model	PPL
Standard Transformer	1.066
Superstill All Epochs	1.097
Superstill Until Epoch 14	**1.06**

Results

Results are presented in Figures 2 and 3. From Figure 2, it can be seen that for the small-scale models, superstilling achieves a markedly lower perplexity during early training. However, by epoch 30, superstilling alone no longer provides any advantage, whereas a model with a superstilling "head start" followed by traditional training continues to outperform the baseline model well past epoch 40. We hypothesize that this is due to the increased representational power of the student model. Once the student model achieves a certain level of performance, the (sometimes inaccurate) information from the teacher model provides less benefit than the ground truth labels alone. As Figure 2 shows, the combined method ("Superstill Until Epoch 14") demonstrates consistent and stronger gains in performance per epoch trained.

The large-scale experiments (Figure 3) confirm the benefit

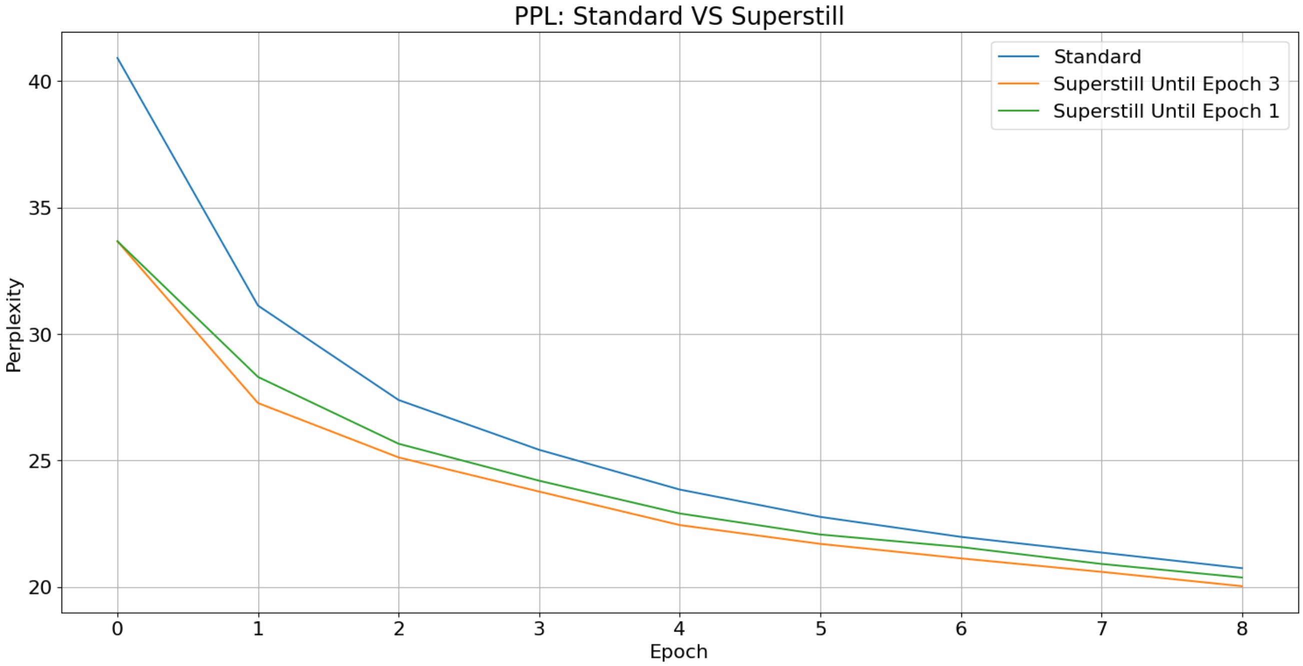

Figure 3: Model perplexity calculated over the entire large dataset of 1500 books (large-scale transformer architecture with 12 layers and 161 million trainable parameters). Perplexity was measured every epoch. Epoch 0 is the first pass over the data

Table 2: Final perplexity achieved by large-scale models

Model	PPL
Standard Transformer	20.746
Superstill Until Epoch 1	20.378
Superstill Until Epoch 3	**20.034**

of superstilling. As before, superstilling on the first 33% of the data enables significantly lower perplexity at each epoch compared to the standard method. At any given epoch n, the superstilled model demonstrates gains commensurate with the epoch n+1 of the standard model. In particular, we can see from Table 3 that the superstilled model has reduced the sample complexity by 50% at epoch 1 (because it has reached comparable performance with only 50% of the data seen by the standard model at epoch 2). These gains become less dramatic the longer the models train. By epoch 8, when knowledge saturation is nearly complete, the superstilled model achieves only a 12% reduction in sample complexity. We note with interest that, given recent analysis of the Chinchilla point for LLM training (Hoffmann et al. 2022), future state-of-the-art language models are unlikely to achieve knowledge saturation using currently available training data. Henc,e the gains achieved by our superstilling method would fall somewhere between 50% and 12% in real-world applications.

In addition to the experiments shown in Table 3, we also explored large-scale models that had been superstilled for only 1 training epoch, and found that model performance exceeded that of the standard method, while also achieving a slight reduction in wall-clock time required to reach a given level of perplexity. We discuss this result further in the next section.

Discussion

Our results suggest that superstilling may be a viable technique for data-efficient training of large-scale language models. In addition to the performance gains achieved during early training[1], this method is model agnostic, and unlike fine-tuning or direct weight transfer methods may be employed even when the model architectures are radically different. Additionally, the reduced sample complexity of the superstilled training method may enable significant gains in terms of financial cost and wall-clock time.

On the topic of wall clock time, consider the following: our current superstilled training method requires data to pass through both the teacher and the student models during training. This additional compute cost largely negates the benefits of the observed increases in sample complexity. However, one can envision a more streamlined implementation in which the teacher's distributions have been stored as a precomputed dataset which can be used over and over again with increasingly sophisticated language models. In this scenario, the extra compute cost is incurred only once, but the financial and wall-clock benefits of superstilling can be carried forward across multiple state-of-the-art LLMs.

[1] We note that, given known limitations of high-quality text data, future state-of-the-art language models may be unable to train to convergence. See Hoffmann et al. (2022).

Table 3: Reduction in Sample Complexity

Model	Epo. 0	Epo. 1	Epo. 2	Epo. 3	Epo. 4	Epo. 5	Epo. 6	Epo. 7	Epo. 8
Standard Transformer	40.919	31.128	27.395	25.427	23.855	22.774	21.984	21.363	20.746
Superstill Until Epoch 3	33.671	27.274	25.125	23.775	22.45	21.707	21.133	20.600	20.034
% Sample Complexity Reduced	NA	50%	33%	25%	20%	17%	14%	12%	

Superstilling is a powerful method for improving the sample complexity of LLMS, and shows strong potential for energy- and data-efficiency of future language model training. Additionally, it may be particularly well suited to low-resource settings involving specialized and/or proprietary datasets, where limitations in the amount of available data demand that each training instance provide as much information as possible. Further work is needed to explore this possibility.

Limitations

This work was conducted on language models with up to 161 million trainable parameters. While the results appear to be consistent across model sizes, a full-scale analysis on models up to 70 million parameters is recommended. As this would incur significant energy expenditure and corresponding environmental cost, we advise that any such efforts be combined with the development of novel LLMS for definitive useful purposes, in order to maximize benefit and avoid energy waste.

The data used in these experiments is exclusively in the English language, and comprises only narrative prose from the 19th and 20th centuries. Given the potential influence of corpus selection on language model attributes, as noted by Fulda (2020), the comparative effectiveness of this method on other languages and corpora cannot be guaranteed.

Additionally, we acknowledge that in recommending the training of new models from old ones, any biases from the old model will likely be incorporated in the new one due to the direct transfer of knowledge inherent in our method.

Conclusion

In this paper we propose superstilling as a training method to reduce sample complexity during the creation of state-of-the-art LLMs. By leveraging the parametric knowledge that is already present in pretrained models, we can reduce the amount of data needed to attain a specific perplexity score by more than 10%, and outline a path toward attaining wall clock speedups via this same method. The superstilling method is complementary to other work on language model training efficiency, and can be combined with various architectural innovations to further increase training efficiency. Our experiments demonstrate the effectiveness of superstilling on Transformer-based models of up to 161M parameters on English-language text generation tasks. We hope that this will be a useful tool for other researchers.

References

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35: 23716–23736.

Bosselut, A.; Rashkin, H.; Sap, M.; Malaviya, C.; Celikyilmaz, A.; and Choi, Y. 2019. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. In Korhonen, A.; Traum, D.; and Màrquez, L., eds., *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4762–4779. Florence, Italy: Association for Computational Linguistics.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Chaudhury, S.; Shelke, N.; Sau, K.; Prasanalakshmi, B.; and Shabaz, M. 2021. A novel approach to classifying breast cancer histopathology biopsy images using bilateral knowledge distillation and label smoothing regularization. *Computational and Mathematical Methods in Medicine*, 2021: 1–11.

Costa-jussà, M. R.; Cross, J.; Çelebi, O.; Elbayad, M.; Heafield, K.; Heffernan, K.; Kalbassi, E.; Lam, J.; Licht, D.; Maillard, J.; et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Driess, D.; Xia, F.; Sajjadi, M. S.; Lynch, C.; Chowdhery, A.; Ichter, B.; Wahid, A.; Tompson, J.; Vuong, Q.; Yu, T.; et al. 2023. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*.

Fulda, N. 2020. You Are What You Read: The Effect of Corpus and Training Task on Semantic Absorption in Recurrent Neural Architectures. In *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 201–206. IEEE.

Gordon, M.; and Duh, K. 2020. Distill, Adapt, Distill: Training Small, In-Domain Models for Neural Machine Translation. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, 110–118.

Hart, M. 2023. Project Gutenberg. In *The Project Gutenberg Literary Archive Foundation*. https://www.gutenberg.org.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hoffmann, J.; Borgeaud, S.; Mensch, A.; Buchatskaya, E.; Cai, T.; Rutherford, E.; de Las Casas, D.; Hendricks, L. A.; Welbl, J.; Clark, A.; Hennigan, T.; Noland, E.; Millican, K.; van den Driessche, G.; Damoc, B.; Guy, A.; Osindero, S.; Simonyan, K.; Elsen, E.; Vinyals, O.; Rae, J.; and Sifre, L. 2022. An empirical analysis of compute-optimal large language model training. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 30016–30030. Curran Associates, Inc.

Jelinek, F.; Mercer, R. L.; Bahl, L. R.; and Baker, J. K. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1): S63–S63.

Jiang, X.; and Deng, X. 2023. Knowledge reverse distillation based confidence calibration for deep neural networks. *Neural Processing Letters*, 55(1): 345–360.

Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T. B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; and Amodei, D. 2020. Scaling Laws for Neural Language Models. arXiv:2001.08361.

Kenton, J. D. M.-W. C.; and Toutanova, L. K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, 4171–4186.

Korthikanti, V. A.; Casper, J.; Lym, S.; McAfee, L.; Andersch, M.; Shoeybi, M.; and Catanzaro, B. 2023. Reducing activation recomputation in large transformer models. *Proceedings of Machine Learning and Systems*, 5.

Lei, T. 2021. When Attention Meets Fast Recurrence: Training Language Models with Reduced Compute. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 7633–7648. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Li, Y.; Lei, Y.; and Yang, X. 2022. Spikeformer: A Novel Architecture for Training High-Performance Low-Latency Spiking Neural Network. arXiv:2211.10686.

Ma, X.; Zhang, P.; Zhang, S.; Duan, N.; Hou, Y.; Zhou, M.; and Song, D. 2019. A tensorized transformer for language modeling. *Advances in neural information processing systems*, 32.

Muennighoff, N.; Rush, A. M.; Barak, B.; Scao, T. L.; Piktus, A.; Tazi, N.; Pyysalo, S.; Wolf, T.; and Raffel, C. 2023. Scaling Data-Constrained Language Models. *arXiv preprint arXiv:2305.16264*.

Nakano, R.; Hilton, J.; Balaji, S.; Wu, J.; Ouyang, L.; Kim, C.; Hesse, C.; Jain, S.; Kosaraju, V.; Saunders, W.; et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

OpenAI. 2023. GPT-4 Technical Report. *ArXiv*, abs/2303.08774.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving Language Understanding by Generative Pre-Training.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Raipuria, G.; Bonthu, S.; and Singhal, N. 2021. Noise robust training of segmentation model using knowledge distillation. In *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part I*, 97–104. Springer.

Rytting, C.; and Wingate, D. 2021. Leveraging the inductive bias of large language models for abstract textual reasoning. *Advances in Neural Information Processing Systems*, 34: 17111–17122.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; Lomeli, M.; Zettlemoyer, L.; Cancedda, N.; and Scialom, T. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*.

Shen, S.; Walsh, P.; Keutzer, K.; Dodge, J.; Peters, M.; and Beltagy, I. 2022. Staged training for transformer language models. In *International Conference on Machine Learning*, 19893–19908. PMLR.

So, D.; Mańke, W.; Liu, H.; Dai, Z.; Shazeer, N.; and Le, Q. V. 2021. Searching for efficient transformers for language modeling. *Advances in Neural Information Processing Systems*, 34: 6010–6022.

Sun, Y.; Dong, L.; Huang, S.; Ma, S.; Xia, Y.; Xue, J.; Wang, J.; and Wei, F. 2023. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Villalobos, P.; Sevilla, J.; Heim, L.; Besiroglu, T.; Hobbhahn, M.; and Ho, A. 2022. Will we run out of data? An analysis of the limits of scaling datasets in Machine Learning. arXiv:2211.04325.

Yang, A.; Nagrani, A.; Seo, P. H.; Miech, A.; Pont-Tuset, J.; Laptev, I.; Sivic, J.; and Schmid, C. 2023. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10714–10726.

Yuan, L.; Tay, F. E.; Li, G.; Wang, T.; and Feng, J. 2020. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3903–3911.

Zakka, C.; Chaurasia, A.; Shad, R.; and Hiesinger, W. 2023. Almanac: Knowledge-grounded language models for clinical medicine. *arXiv preprint arXiv:2303.01229*.

Zhang, P.; Li, X.; Hu, X.; Yang, J.; Zhang, L.; Wang, L.; Choi, Y.; and Gao, J. 2021. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of* *the IEEE/CVF conference on computer vision and pattern recognition*, 5579–5588.

Zhang, Y.; Sun, S.; Galley, M.; Chen, Y.-C.; Brockett, C.; Gao, X.; Gao, J.; Liu, J.; and Dolan, B. 2020. DIALOGPT : Large-Scale Generative Pre-training for Conversational Response Generation. In Celikyilmaz, A.; and Wen, T.-H., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 270–278. Online: Association for Computational Linguistics.

Zhou, Z.; Zhu, Y.; He, C.; Wang, Y.; YAN, S.; Tian, Y.; and Yuan, L. 2023. Spikformer: When Spiking Neural Network Meets Transformer. In *The Eleventh International Conference on Learning Representations*.