# Have a Larger Cake and Eat it Faster Too: A Guideline to Train Larger Models Faster

Newsha Ardalani
Baidu Silicon Valley AI Lab
ardalaninewsha@baidu.com

Joel Hestness
Baidu Silicon Valley AI Lab
hestness@baidu.com

Gregory Diamos
Baidu Silicon Valley AI Lab
gregdiamos@baidu.com

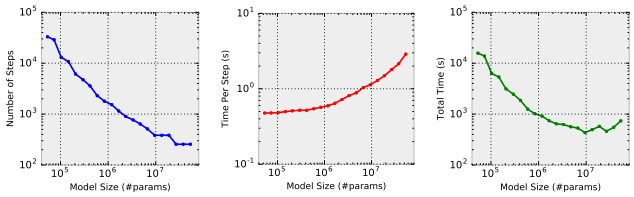
## ABSTRACT

With the increasing prevalence of deep neural networks and their growing demand for more powerful hardware, understanding the interplay of model architecture parameters, hardware architecture parameters, model and data parallelism on overall model performance (training time and accuracy) becomes ever more important in order to design next-generation deep learning (DL) hardware. To aid such understanding, this work studies the effect of scaling model size on overall performance, and debunks a long-held belief that larger models must take longer to train.

We first break the total training time into number of steps and time/step. We analytically model the training time per step and empirically study the number of steps to convergence. We observe that larger models take fewer steps to reach to minimum validation loss (halting point). Therefore, the burden is on the hardware community to improve hardware design such that the growth in training time/step would be slower than the decrease in the number of steps as model size scales. If successful, larger models will converge faster, and therefore we can have a larger cake and eat it faster too.

## 1 INTRODUCTION

The recent success in deep learning has been a driving force in the hardware industry for designing more powerful, energy-efficient GPUs [3] and ASICs [1, 2, 7–11, 15] with special support for deep learning. Recent studies suggest that deep learning accuracy scales with training data [4, 12, 14]. Therefore, there is an expectation that model size and as a result computation demands to grow rapidly with dataset size. In this paper, we make an observation that suggests the growth in model size and computation demand is expected to be even faster than the growth in dataset size. We observe that larger models take fewer steps to reach to similar levels of accuracy. Figure 1 shows this pattern for character-level language models trained on 0.01% of the Billion Word dataset [6], with batch size of 128, and using Adam optimizer with an initial learning rate of 0.001. As depicted, larger models take fewer steps to converge, while training time per step grows with model size. On this particular design, we also get better overall training time on an the existing GPU hardware (Nvidia Maxwell). This overall trend of total training time dropping by model size depends on the underlying hardware, model architecture and algorithmic properties of the implementation. Note here that the number of steps to convergence is only a function of model architecture, while the training time per step is a function of hardware architecture and the efficiency of the implementation. If we can design our hardware and/or algorithms such that the training time per step grows with a lower pace than the decrease in the number of steps, we can practically train larger models faster than smaller ones.




**(a) Number of Steps** **(b) Time per Step** **(c) Overall Time**

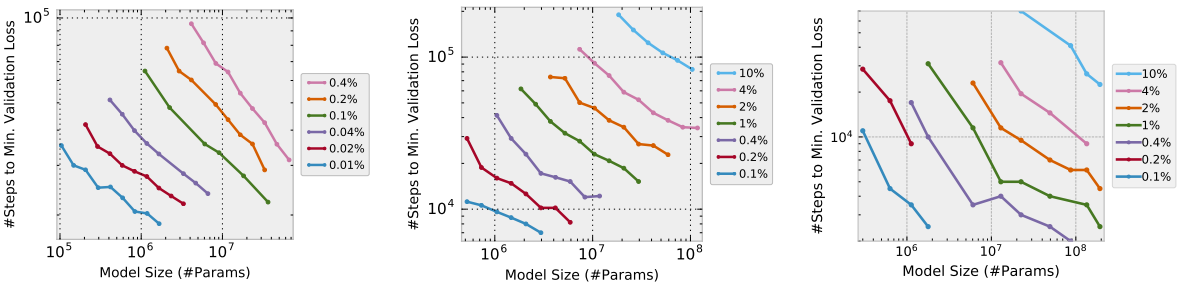
**Figure 1: Larger models take fewer steps to converge**

There are three different approaches to control training time/step as model size grows: Exploiting data parallelism, model parallelism and kernel parallelism, which require non-trivial changes in implementation and heavy support in hardware. Recently, data parallelism has been explored extensively in industry and academia [13, 17, 19, 20], however model parallelism has been used only for exotic training and deployment scenarios [18]. We make the argument that model parallelism must be considered as a first-class parallelism requirement for future deep learning scaling. Specifically, as dataset size grows, the potential to grow data and model parallelism grows comparably. However, there are hard upper-bounds on scalabilty of data parallelism [13, 17, 19, 20] that will encourage greater emphasis on ways to improve model parallelism. Further, we show that increasing model size can proceed without increasing training time. The main contributions include:

(1) To the best of our knowledge, we are the first to observe that larger models take fewer steps to converge. We predict an increasing pressure on hardware community to support larger models and specifically model parallelism.
(2) Our results suggests that number of steps to convergence has a reciprocal relationship to model size and linear relationship to dataset size ($\#Steps \approx a.\frac{Data\_Size^{k_1}}{Model\_Size^{k_2}} + b$). Meanwhile, the training time per step grows linearly in model size (without changing parallelism).
(3) Finally, we analyze the implications of these findings on future hardware/system design.

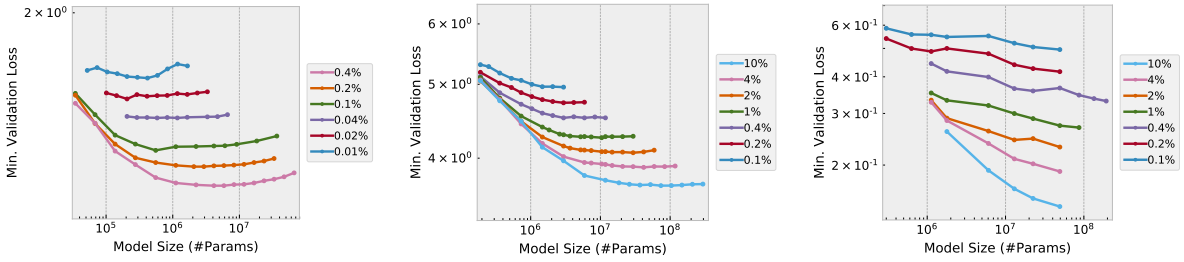
## 2 METHODOLOGY

We define time-to-convergence as the number of steps to reach within 1% of the minimum validation loss. We use early stopping (at minimum validation loss) to control overfitting. We increase the model size by increasing the number of nodes per layer, while keeping all the other architecture parameters the same (same learning rate, same batch size, etc.). We evaluate our finding on three established DL models: character-level language model (LM), word-level LM and speech recognition.




(a) Character LM (b) Word LM (c) Speech

**Figure 2: #Steps vs. Model Size Across Different Application Domains:X-axis represents model size in terms of the number of parameters in log-scale. Y-axis is the number of steps to minimum validation loss. Different lines represent different dataset sizes (percentage of full dataset). For character LM, we vary the dataset size from 0.01% to 0.4% of the 1B dataset. For word LM, we vary dataset size from 0.1% to 10% of 1B dataset. For speech, we vary the dataset size from 1% to 10% of an internal 20000 hour dataset.**




(a) Character LM (b) Word LM (c) Speech

**Figure 3: Minimum Validation Loss vs. Model Size:X-axis represents model size in terms of in log-scale. We use early stopping (at minimum validation loss) to control overfitting. Y-axis represent minimum validation loss (Lower is better).**

- **Word LM:** We implement LSTM-based word LMs as described in [16]. We restrict the vocabulary to the top 10,000 most frequent words in the Billion Word Dataset [6]. The networks are 2-layer LSTMs, with sequence length of 80, the same number of hidden nodes in each layer.
- **Character LM:** We implement char-LM using Recurrent Highway Networks (RHNs)[21]. Specifically, we train a 1-layer, depth 10 RHN, sequence length 150.
- **Speech recognition:** We train models similar to Deep Speech 2 (DS2) [5] which consist of two convolution layers followed by four bidirectional LSTM recurrent layers.

# 3 RESULTS AND ANALYSIS

In this Section, we study the trade-off between convergence time, accuracy and model size. Our preliminary results suggest that there is a potential to increase model size to improve training time without hurting accuracy.

## 3.1 Number of Steps to Minimum Validation Loss

Figure 2 shows the number of steps to convergence for the character, word and speech model. X-axis and Y-axis are in logarithmic-scale. We increase the model size on X-axis by increasing the width of each layer. Different lines represent different dataset sizes (percentage of a full dataset size). As shown, the number of steps to convergence declines with model size and grows with dataset size. In general, the number of steps to convergence can be approximated by $\frac{D^{k_1}}{M^{k_2}}$, where $D$ is the size of training set, $M$ is model size, and $k_1$ and $k_2$ are controlled by dataset characteristics and model architecture parameters, respectively. We also observe that this power law relationship eventually plateaus, i.e there exists a model size beyond which improving model size does not improve training time.

## 3.2 Accuracy

Although theoretical results suggest over-parametrizing models may lead to worse generalization error, our empirical results show that the change in accuracy is insignificant (Figure 3). As shown, loss remains almost the same after it reaches to its best performance.

## 3.3 Sensitivity Analysis

We also study the effect of change in depth and learning rate on the number of steps and the power-law relationship. We observe that the number of steps to minimum validation loss also declines as we sweep the number of layers from 1 to 128. We also observe that adaptively changing learning rate (using Adam optimizer) improves the number of steps in two ways: It shifts down the power-law curve, and also makes it steeper.

## 3.4 System/Hardware Implications

Moving forward, we expect an increasing demand for larger models, not only because they are more accurate with more data but also because they train faster. Therefore, system/hardware designers need to focus on larger models by providing support for model parallelism through improving inter-device bandwidth and computational throughput per device.

# REFERENCES


[1] 2017. Carebrase. https://www.cerebras.net. (2017). [Online; accessed 20-Nov-2017].

[2] 2017. GraphCore. https://www.graphcore.ai. (2017). [Online; accessed 20-Nov-2017].

[3] 2017. NVIDIA Tesla Volta: The New GPU Architecture Designed to Bring AI to Every Industry. https://www.nvidia.com/en-us/data-center/volta-gpu-architecture/. (2017). [Online; accessed 20-Nov-2017].

[4] Shun-ichi Amari, Naotake Fujita, and Shigeru Shinomoto. 1992. Four types of learning curves. *Neural Computation* 4, 4 (1992), 605–618.

[5] Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, JingDong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, et al. 2016. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. In *Proceedings of The International Conference on Machine Learning (ICML)*. 173–182.

[6] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling. *arXiv preprint arXiv:1312.3005* (2013).

[7] Tianshi Chen, Zidong Du, Ninghui Sun, Jia Wang, Chengyong Wu, Yunji Chen, and Olivier Temam. 2014. Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning. In *ACM Sigplan Notices*, Vol. 49. ACM, 269–284.

[8] Yunji Chen, Tao Luo, Shaoli Liu, Shijin Zhang, Liqiang He, Jia Wang, Ling Li, Tianshi Chen, Zhiwei Xu, Ninghui Sun, et al. 2014. Dadiannao: A machine-learning supercomputer. In *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture*. IEEE Computer Society, 609–622.

[9] Yu-Hsin Chen, Tushar Krishna, Joel S Emer, and Vivienne Sze. 2017. Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE Journal of Solid-State Circuits* 52, 1 (2017), 127–138.

[10] Trishul M Chilimbi, Yutaka Suzue, Johnson Apacible, and Karthik Kalyanaraman. 2014. Project Adam: Building an Efficient and Scalable Deep Learning Training System.. In *OSDI*, Vol. 14. 571–582.

[11] Zidong Du, Robert Fasthuber, Tianshi Chen, Paolo Ienne, Ling Li, Tao Luo, Xiaobing Feng, Yunji Chen, and Olivier Temam. 2015. ShiDianNao: Shifting vision processing closer to the sensor. In *ACM SIGARCH Computer Architecture News*, Vol. 43. ACM, 92–104.

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

[13] Priya Goyal, Piotr Dollàr, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. 2017. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *Facebook AI Research Publications* (2017).

[14] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Patwary, Mostofa Ali, Yang Yang, and Yanqi Zhou. 2017. Deep Learning Scaling is Predictable, Empirically. *arXiv preprint arXiv:1712.00409* (2017).

[15] Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. 2017. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*. ACM, 1–12.

[16] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the Limits of Language Modeling. *arXiv preprint arXiv:1602.02410v2* (2016).

[17] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. 2016. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. *arXiv preprint arXiv:1609.04836* (2016).

[18] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538* (2017).

[19] Cho-Jui Hsieh James Demmel Kurt Keutzer Yang You, Zhao Zhang. 2017. ImageNet Training in Minutes. *arXiv preprint arXiv:1709.05011* (2017).

[20] Yang You, Igor Gitman, and Boris Ginsburg. 2017. Scaling SGD Batch Size to 32k for ImageNet Training. *arXiv preprint arXiv:1708.03888* (2017).

[21] Julian Georg Zilly, Rupesh Kumar Srivastava, Jan Koutnik, and Jurgen Schmidhuber. 2017. Recurrent Highway Networks. In *Proceedings of The International Conference on Machine Learning (ICML)*.