EXAMINING THE ROLE OF NORMALIZATION IN THE LOTTERY TICKET HYPOTHESIS

Arlene Siswanto
MIT CSAIL
siswanto@mit.edu

Jonathan Frankle
MIT CSAIL
jfrankle@mit.edu

Michael Carbin
MIT CSAIL
mcarbin@mit.edu

ABSTRACT

The original *lottery ticket hypothesis* posits that neural networks used in practice contain sparse subnetworks that are capable of training from initialization to the same accuracy as the full network. We evaluate the possibility that normalization may be responsible for the failure to find these subnetworks in larger-scale settings. To do so, we study residual networks implemented with Fixup initialization, which obviates the need for batch normalization. When the lottery ticket procedure is applied, the resulting subnetworks display behavior similar to those from standard residual networks, suggesting that normalization alone cannot explain these failures. In doing so, we the extend existing lottery ticket hypothesis findings to an entirely new setting.

1 INTRODUCTION

The lottery ticket hypothesis (Frankle & Carbin, 2019) contends that neural networks used in practice contain sparse subnetworks capable of training to the same accuracy as the full network. These *matching subnetworks* comprise less than 10-20% of weights from original fully-connected and convolutional feed-forward architectures, improving storage and computational efficiency.

Small-scale image classification networks, such as MNIST and CIFAR-10, empirically support the original claim that matching subnetworks can be found when trained from initialization. However, in larger scale settings—for example, ResNets for image classification and Transformers for machine translation—the same behavior is not found (Liu et al., 2019; Gale et al., 2019; Yu et al., 2020). Instead, existing pruning strategies find sparse, trainable networks only when using parameters from *early* in training rather than those from initialization (Frankle et al., 2019).

Many factors may explain this change in behavior in larger-scale settings. In this paper, we consider the role of normalization with respect to the lottery ticket hypothesis. Techniques such as batch normalization (BatchNorm; Ioffe & Szegedy, 2015) and layer normalization (Ba et al., 2016) have become ubiquitous in contemporary neural network architectures due to improved accuracy in a reduced training time. Settings in which winning tickets can be found lack normalization; settings in which they cannot be found (without modifications to hyperparameters) utilize normalization.

There are compelling reasons to believe that normalization could be responsible for this inability to find matching subnetworks in larger networks when trained from initialization. The precise role of BatchNorm remains a subject of debate in the literature: the technique was originally proposed as a solution to *internal covariate shift*, but reasons behind its effectiveness have been contested. (Ioffe & Szegedy, 2015; Balduzzi et al., 2017; Santurkar et al., 2018; Bjorck et al., 2018; Morcos et al., 2018; Kohler et al., 2019; Yang et al., 2019; Luo et al., 2019). One alternative explanation is that BatchNorm introduces length-direction decoupling in parameter optimization. Another is that it alters the nature of the optimization landscape, potentially interacting with *instability* behavior. Frankle et al. (2019) connect the accuracy achieved by sparse subnetworks to the structure of the optimization landscape: lottery ticket subnetworks found by pruning only train to full accuracy when the result of optimization is *stable* to stochastic gradient descent noise.

Fixup initialization. Simply removing batch normalization from a residual network would dramatically reduce its allowable range of learning rates, decreasing accuracy (Bjorck et al., 2018). To maintain target accuracy, we train residual networks for CIFAR-10 with Fixup initialization (Zhang

et al., 2019), an initialization scheme that makes it possible to train ResNets *without* batch normalization to standard performance with standard hyperparameters. By doing so, we can evaluate the properties of sparse subnetworks found in networks trained with and without batch normalization.

Potential outcomes. Applying the procedure outlined in the original lottery ticket paper (Frankle & Carbin, 2019) to residual networks, we expect one of three behaviors from the subnetworks that use Fixup initialization in place of batch normalization:

- *Matching subnetworks can be found at increased sparsities.* Such a behavior would offer evidence that batch normalization hinders the behavior of the lottery ticket hypothesis on larger networks.
- *Matching subnetworks can be found only at decreased sparsities.* This would offer evidence that the current behavior is reliant on some component of batch normalization.
- *Pruned Fixup and BatchNorm subnetworks exhibit similar properties*. This would imply that batch normalization is not responsible for the behavioral change of the lottery ticket hypothesis on larger networks. In addition, it would demonstrate that the lottery ticket observations are robust to changing the normalization scheme of the network.

Results. We find that the third outcome holds. Specifically, residual networks trained using Fixup initialization exhibit lottery ticket behavior. Neither standard ResNets (with BatchNorm) nor Fixup ResNets find winning tickets at initialization; accuracies of the subnetworks early in training for Fixup networks are comparable to those found in standard ResNets. We conclude that normalization alone cannot explain the inability to find winning tickets in larger-scale settings. However, in doing so, we reveal that subnetworks originating from Fixup's BatchNorm-free regime also display known lottery ticket behaviors, namely the presence of sparse, trainable subnetworks early in training.

2 METHODOLOGY

The original lottery ticket hypothesis describes the following procedure to find matching subnetworks from initialization (Frankle & Carbin, 2019). This procedure successfully discovers matching subnetworks in the small-scale image classification networks described earlier in the text:

1. Randomly initialize a dense neural network and store these weights
2. Train the network to completion
3. Globally prune $p\%$ of weights with the smallest magnitude
4. Reset the remaining parameters to their stored weights
5. Repeat steps 2-4 until reaching the desired level of sparsity—the resulting subnetwork should contain significantly reduced parameters yet perform to similar accuracy

Modification for larger networks. To analyze the behavior of the hypothesis on larger networks, follow-up work introduced the concept of *rewinding* (Frankle et al., 2019). To rewind to a specified point early in training, store the weights obtained at this point at each iteration (step 2). Then reset the parameters (step 4) to these stored weights rather than weights from initialization. Prior lottery ticket work on larger networks reveals that, though matching subnetworks cannot be found when rewinded to initialization, they can indeed be found when rewinded to points early in training.

Fixup implementation. To analyze performance in larger networks, we apply the lottery ticket procedure to ResNet-20 with standard batch normalization as well as ResNet-20 with *fixed-update initialization* (Fixup initialization) at various rewinding points. Fixup initialization (Zhang et al., 2019) is a scheme that aims to enable residual networks to achieve comparable performance on image classification and machine translation tasks, without the use of normalization.

The initialization was designed such that the gradient updates to the network function are independent of network depth, informed by the derivation of a lower bound of the gradient norm at initialization. In addition to several other modifications (e.g. adding scalar multiples and biases), the procedure involves scaling weight layers inside residual branches by $L^{-\frac{1}{2m-2}}$, where L denotes the number of residual branches in the network and m denotes the number of layers per branch.

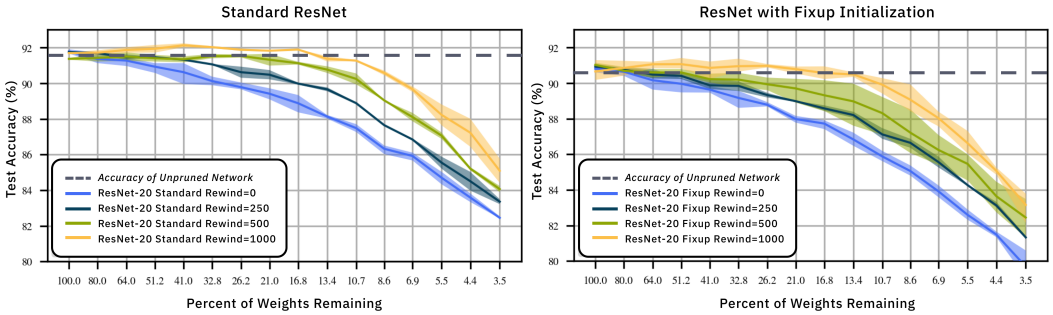

Figure 1: **Subnetwork accuracy by pruning iteration at various rewinding points** *Left:* Standard ResNet with batch normalization. *Right:* ResNet with Fixup initialization. Both exhibit similar patterns in accuracy; the overall accuracy of Fixup is slightly below that of batch normalization.

3 Results

Baseline behavior. Applying the methodology to the original and Fixup ResNet variations allows us to compare the two behaviors. The left plot in Figure 1 displays the behavior of lottery ticket hypothesis procedure on the original ResNet with BatchNorm at various rewinding points. Rewinding to initialization, the subnetwork accuracy decreases as the number of pruning iterations increases. As a result, we are unable to find matching subnetworks at extreme sparsities. Rewinding to points incrementally later in training, subnetwork accuracy gradually and steadily improves such that we begin to find matching subnetworks at later pruning iterations, uncovering behavior that mimics lottery ticket behavior on small-scale networks more closely.

Fixup behavior. In comparison to the original baseline, the Fixup implementation displays an overall accuracy lower by roughly 1-2%. However, we find that both networks follow a similar progression of subnetwork accuracy over pruning iteration. When rewinding to initialization, the subnetwork accuracy steadily drops across pruning iteration. When rewinded to points incrementally later in training, the subnetwork accuracy shows gradual improvement in a manner similar to that of the original baseline. By observation, however, this accuracy fluctuates somewhat erratically when compared to the original BatchNorm variant.

4 Discussion and Conclusion

Our foray into batch normalization alternatives began with the intention of investigating why matching subnetworks cannot be found at initialization in larger networks. Exploring the role of normalization in these networks was a logical direction, as networks in which the lottery ticket hypothesis did not perform as expected coincided with networks that adopted batch normalization, a technique that has become a standard component of larger networks. Replacing batch normalization with Fixup initialization, we find that both networks have surprisingly similar behavior: neither finds matching subnetworks at initialization, yet both gradually uncover matching subnetworks when rewound to later points. As a result, we argue that lottery ticket behavior on larger networks is *not* due to normalization and suggest that another factor is responsible for this behavior.

Future work. Alternate schemes exist as replacements to batch normalization. Group normalization (Wu & He, 2018) does not take batch size into consideration while normalizing. Weight normalization (Salimans & Kingma, 2016) decouples length and direction during reparameterization and, with initialization based on mean field approximation (Arpit et al., 2019), achieves comparable performance to batch normalization. A similar exploration of such schemes could provide further evidence that BatchNorm is not responsible for the lottery ticket behavior of larger networks.

References

Devansh Arpit, Victor Campos, and Yoshua Bengio. How to initialize your network? robust initialization for weightnorm resnets. In *Advances in Neural Information Processing Systems*, 2019.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

David Balduzzi, Marcus Frean, Lennox Leary, JP Lewis, Kurt Wan-Duo Ma, and Brian McWilliams. The shattered gradients problem: If resnets are the answer, then what is the question? In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 342–350. JMLR. org, 2017.

Nils Bjorck, Carla P Gomes, Bart Selman, and Kilian Q Weinberger. Understanding batch normalization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 31*, pp. 7694–7705. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/7996-understanding-batch-normalization.pdf.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJl-b3RcF7.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. *arXiv preprint arXiv:1912.05671*, 2019.

Trevor Gale, Erich Elsen, and Sara Hooker. The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574*, 2019.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456, Lille, France, 07–09 Jul 2015. PMLR.

Jonas Kohler, Hadi Daneshmand, Aurelien Lucchi, Thomas Hofmann, Ming Zhou, and Klaus Neymeyr. Exponential convergence rates for batch normalization: The power of length-direction decoupling in non-convex optimization. In Kamalika Chaudhuri and Masashi Sugiyama (eds.), *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pp. 806–815. PMLR, 16–18 Apr 2019. URL http://proceedings.mlr.press/v89/kohler19a.html.

Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJlnB3C5Ym.

Ping Luo, Xinjiang Wang, Wenqi Shao, and Zhanglin Peng. Towards understanding regularization in batch normalization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJlLKjR9FQ.

Ari Morcos, David GT Barrett, Neil C Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization. In *Proceeding of the International Conference on Learning Representations*, 2018.

Tim Salimans and Durk P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, 2016.

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 31*, pp. 2483–2493. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/7515-how-does-batch-normalization-help-optimization.pdf.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018.

Greg Yang, Jeffrey Pennington, Vinay Rao, Jascha Sohl-Dickstein, and Samuel S. Schoenholz. A mean field theory of batch normalization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SyMDXnCcF7.

Haonan Yu, Sergey Edunov, Yuandong Tian, and Ari S. Morcos. Playing the lottery with rewards and multiple languages: lottery tickets in rl and nlp. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1xnXRVFwH.

Hongyi Zhang, Yann N. Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization via better initialization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1gsz30cKX.