The Lottery Ticket Hypothesis at Scale

Jonathan Frankle [1] **Gintare Karolina Dziugaite** [2 3] **Daniel M. Roy** [4 5] **Michael Carbin** [1]

Abstract

Recent work on the *lottery ticket hypothesis* proposes that randomly-initialized, dense neural networks contain much smaller, fortuitously initialized subnetworks (*winning tickets*) capable of training to similar accuracy as the original network at a similar speed. While strong evidence exists for the hypothesis across many settings, it has not yet been evaluated on large, state-of-the-art networks and there is even evidence against the hypothesis on deeper networks.

We modify the lottery ticket pruning procedure to make it possible to identify winning tickets on deeper networks. Rather than set the weights of a winning ticket to their original initializations, we set them to the weights obtained after a small number of training iterations (*late resetting*). Using late resetting, we identify the first winning tickets for Resnet-50 on Imagenet

To understand the efficacy of late resetting, we study the *stability* of neural network training to pruning, which we define as the consistency of the optimization trajectories followed by a winning ticket when it is trained in isolation and as part of the larger network. We find that later resetting produces stabler winning tickets and that improved stability correlates with higher winning ticket accuracy. This analysis offers new insights into the lottery ticket hypothesis and the dynamics of neural network learning.

1. Introduction

Neural network compression techniques like pruning (e.g., Han et al. (2015)) and distillation (e.g., Hinton et al. (2015)) are known to dramatically reduce the number of parameters necessary to represent the functions learned by trained networks. Until recently, however, it was believed that these compressed networks could not be trained directly (Han et al., 2015; Li et al., 2016).

Frankle & Carbin (2019) showed that the networks that result from pruning are indeed capable of learning effectively from the start, matching or exceeding the test accuracy of the original networks while learning at least as fast. In order to achieve this performance, the connections that survive the pruning process must receive the same initial values as when they were part of the original network.

Based on these results, Frankle & Carbin proposed *the lottery ticket hypothesis*: for a randomly-initialized, dense, feed-forward network $f(x; W_0)$, there exists a mask $m \in \{0,1\}^{|W_0|}$ such that $\|m\|_0 \ll |W|$ and $f(x; m \odot W_0)$ trains to accuracy at least that of $f(x; W_0)$ in a similar number of steps. Informally: dense, trainable neural networks contain equally-capable subnetworks termed *winning tickets*. The original initialization is vital for achieving this performance; when randomly reinitialized, winning ticket accuracy and learning speed only worsen under further pruned. While the authors do not explain why the initialization is important, the presumption is that the combination of initial weights and connectivity are well-suited for the task at hand.

1.1. Pruning Before Training

The lottery ticket hypothesis is one of several recent efforts to understand whether smaller networks can learn as effectively as their larger counterparts. Doing so offers the prospect of improving the performance of training.

Liu et al. (2019) recently demonstrated that—in several cases— a network can be randomly pruned and reinitialized (producing a fresh, smaller network) and trained to accuracy similar to that of the original network. These results seemingly disagree with the emphasis that the lottery ticket hypothesis places on initialization.

Figure 1 compares the accuracy of randomly reinitialized winning tickets (orange) and winning tickets with the original initialization (blue) for two standard networks for CIFAR10: VGG19 (Simonyan & Zisserman (2014); adapted by Liu et al.) and Resnet-18 (He et al., 2016). The results for VGG19 (left) support the findings of Liu et al. that pruned, randomly reinitialized networks can match the accuracy of the original network: VGG19 can do so when pruned by up

[1]MIT CSAIL [2]University of Cambridge [3]Element AI [4]University of Toronto [5]Vector Institute. Correspondence to: Jonathan Frankle <jfrankle@mit.edu>.

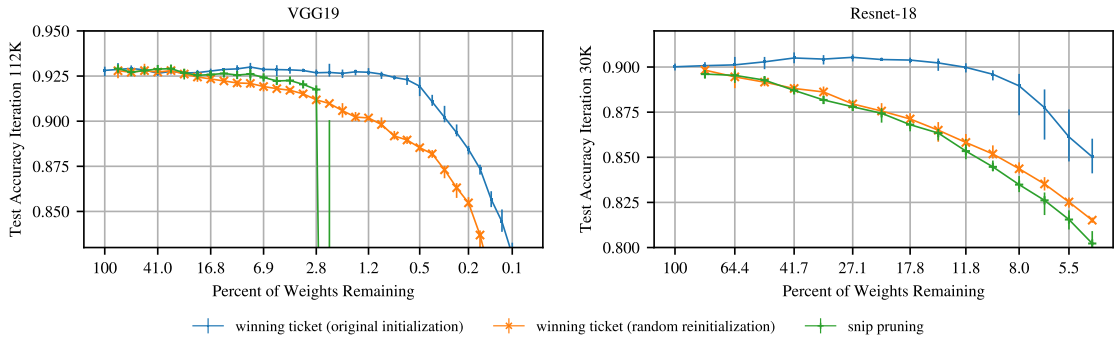


Figure 1. The accuracy achieved by VGG19 (left) and Resnet-18 (right) on CIFAR10 when pruned to the specified size using iterative pruning and SNIP. Networks are trained with warmup and the learning rate hyperparameters used by Frankle & Carbin.

to 80% (after which Liu et al. do not provide data). However, beyond this point, the accuracy of the randomly reinitialized networks declines steadily. In contrast, winning tickets with the original initialization match the accuracy of the original network when pruned by up to 99%. For Resnet-18 (right), which has 75x fewer parameters, the randomly reinitialized networks lose accuracy much sooner.

Alternatively, Lee et al. (2018) propose SNIP, a one-shot pruning technique that removes connections immediately after initialization and prior to any training. SNIP considers the *sensitivity* of the loss to each weight (based on one mini-batch of data) and removes those weights to which the loss is least sensitive. Sensitivity is measured by multiplying each weight w by a virtual parameter $c = 1$ and computing $\frac{\partial L}{\partial c}$. The green lines in Figure 1 show a replication of SNIP on VGG19 and Resnet-18. While SNIP is a promising improvement over random reinitialization on VGG19, there is still a performance gap between SNIP and the winning tickets—an opportunity to further improve the performance of pruning before training.

1.2. Existing Work is Limited in Scale

While these papers offer insights into the behavior of neural networks, none has been shown to readily extend to large-scale networks and datasets as embodied by, for example, Resnet-50 (He et al., 2016) and ImageNet (Russakovsky et al., 2015). SNIP provides results only for Tiny ImageNet, a significantly restricted version of ImageNet with only 200 classes of the standard 1000. Liu et al.'s results for Resnet-50 under sparse pruning show that accuracy declines when only 30% of parameters are pruned.

Frankle & Carbin do not present results for ImageNet; VGG19 is the largest network studied. Moreover, their pruning procedure does not find winning tickets for deeper networks (i.e., VGG19 and Resnet-18) under standard hyperparameters; the subnetworks it finds perform no better than when randomly reinitialized. Gale et al. (2019) recently found this to be the case for Resnet-50 on ImageNet as well. Frankle & Carbin identify training at the standard learning rate as the issue. For VGG19 and Resnet-18, they instead use a smaller initial learning rate that increases during training to the standard learning rate (*warmup*). Although warmup makes it possible to find winning tickets, it adds complexity to hyperparameter selection and limits the generality of the lottery ticket findings.

1.3. Contributions

We propose a small but critical change to Frankle & Carbin's procedure for finding winning tickets that makes it possible to overcome the scalability challenges with deeper networks. After training and pruning the network, do not reset each weight to its initialization at the beginning of training; instead, reset it to its value at an iteration very close to the beginning of training. We term this practice *late resetting*, since the weights that survive pruning are reset back to their values at an iteration slightly later than initialization.

Late resetting makes it possible to replicate and improve upon existing results for CIFAR10 networks and eliminate the need for warmup. We leverage late resetting to identify winning tickets for Resnet-50 on ImageNet, extending the lottery ticket results to a large, state-of-the-art network and benchmark. We find winning tickets when Resnet-50 is pruned by more than 79%; top-1 accuracy drops by only 1% when Resnet-50 is pruned by 89%.

Finally, we study a possible mechanism for the efficacy of late resetting and—more broadly—the lottery ticket hypothesis: *stability to pruning*. The stability of a network to pruning is the extent to which pruning produces a subnetwork that follows a similar optimization trajectory regardless of whether the surviving weights are trained in isolation or as part of the full network. Winning tickets initialized such that they are stabler to pruning reach higher accuracy, and late resetting leads to improvements in both of these metrics.

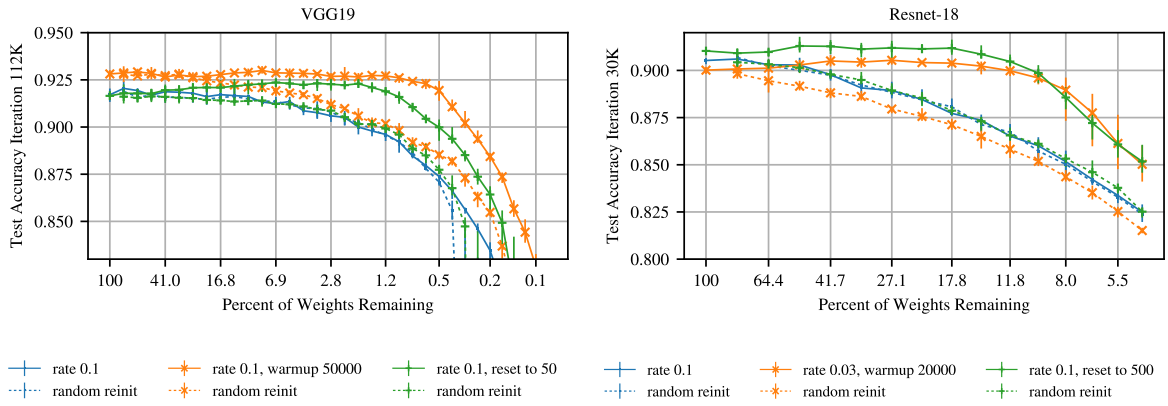


Figure 2. Accuracy of winning tickets (iterative pruning) on VGG19 (left) and Resnet-18 (right) on CIFAR10 with various learning rates.

2. Winning Tickets with Late Resetting

The *lottery ticket hypothesis* (Frankle & Carbin, 2019) attempts to reconcile two seemingly conflicting neural network behaviors: the functions learned by trained networks can be compressed dramatically (sometimes by 90% or more), but these compressed network architectures cannot be trained to equivalent accuracy from the start. Frankle & Carbin demonstrate that, for small image-classification networks for MNIST and CIFAR10, pruned architectures are indeed trainable so long as each surviving weight is initialized to the same value it received at the beginning of training. In other words, the original, overparameterized networks contained small, trainable subnetworks termed *winning tickets*. Winning tickets tend to learn faster than the original network and reach higher accuracy.

Formal statement. The lottery ticket hypothesis considers two randomly-initialized, feed-forward neural networks. The first, $N = f(x; W_0)$, is dense and has random initialization W_0. The second, $N_m = f(x, m \odot W_0)$, has the same initialization but with some weights pruned by mask $m \in \{0, 1\}^{|W_0|}$. In j training iterations, N reaches accuracy test a; in j_m training iterations, N_m reaches test accuracy a_m. The lottery ticket hypothesis states that $\exists m$ such that $j_m \leq j$ (*commensurate training time*), $a_m \geq a$ (*commensurate accuracy*), and $||m||_0 \ll |W_0|$ (*fewer parameters*).

Finding winning tickets. Frankle & Carbin identify the structure of a winning ticket by pruning: randomly initialize a network $f(x; W_0)$, train it to completion, and prune the weights with the lowest magnitudes. To initialize the winning ticket, reset each remaining weight to its value in W_0. There are two different strategies for finding a winning ticket with $p\%$ of weights pruned: *one-shot pruning* and *iterative pruning*. In one-shot pruning, the network is trained once, $p\%$ of weights are pruned, and the surviving weights are reset to create a winning ticket. In iterative pruning, the network is repeatedly trained, pruned by $p^{\frac{1}{n}}\%$, and reset; this process occurs n times so that, by the end, $p\%$ of weights have been pruned. Iterative pruning generally finds smaller winning tickets than does one-shot pruning.

Warmup. This pruning-based heuristic for finding winning tickets is brittle when applied to deeper networks. Figure 2 presents the results of performing iterative pruning on VGG19 and Resnet-18 for CIFAR10. When training the network with standard hyperparameters (most notably high initial learning rates), no winning tickets are found (blue line). However, training at a learning rate an order of magnitude lower does yield winning tickets. Frankle & Carbin blend these observations by linearly warming up the learning rate. (He et al. (2016); Goyal et al. (2017) describe similar warmup schemes for learning rates.) Doing so makes it possible to find winning tickets (orange line) that perform far better with the original initialization than when randomly reinitialized (dashed orange line).

Frankle & Carbin provide no principled basis for the failures of iterative pruning and the efficacy of warmup. While warmup is somewhat effective, it has several drawbacks. It introduces additional dimensions to hyperparameter search: the eventual high learning rate and the number of iterations to reach it. It also influences the overall accuracy of the network. On VGG19, the accuracy of the original network actually increases; on Resnet-18, warmup still cannot reach the standard learning rate, resulting in lower overall accuracy. More broadly, the need for warmup undermines the claimed generality of the lottery ticket hypothesis.

Late Resetting. Instead of warmup, we propose *late resetting*, in which the weights of a winning ticket are set to their values after a small amount of training rather than their initializations at iteration 0. Formally, late resetting produces a network $f(x; m \odot W_t)$, where W_t are the weights of the original network at iteration t and $t \ll j$ (the number of iterations for which the original network was trained). Informally, a sparse, trainable subset of the original network emerges early in training (rather than at initialization).

Our results show that late resetting identifies winning tick-

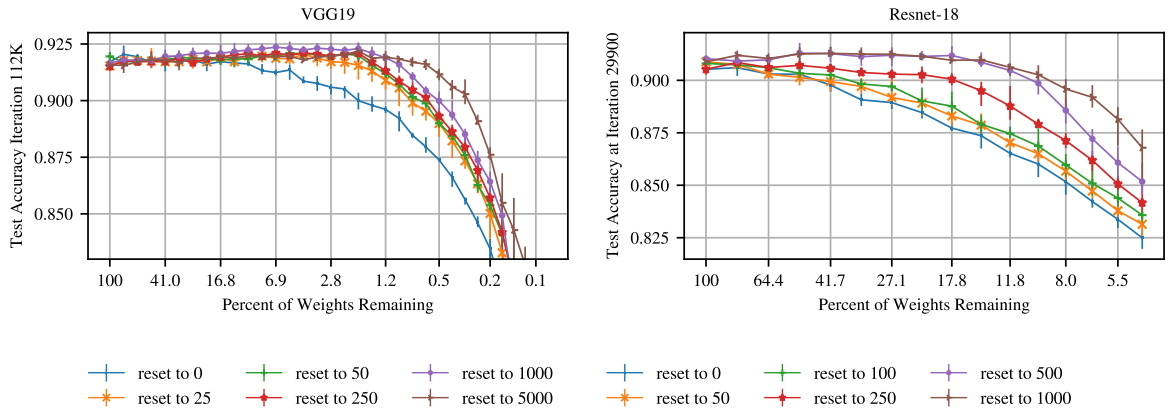


Figure 3. The accuracy achieved by winning tickets found via iterative pruning on VGG19 (left) and Resnet-18 (right) on CIFAR10 with different late resetting iterations.

ets for VGG19 and Resnet-18 without any hyperparameter modification. The green lines in Figure 2 show the result of applying late resetting to VGG19 at iteration 1,000 (out of 112,000 total) and to Resnet-18 at iteration 500 (out of 30,000 total)—1.4 epochs each—with the standard hyperparameters for each network. In both cases, iterative pruning with late resetting successfully recovers winning tickets. For Resnet-18, these winning tickets surpass the accuracy of the original network, unlike those found with warmup. When randomly reinitialized, the winning tickets match the poor performance of those networks whose weights were reset to iteration 0.

The iteration at which to perform late resetting need not be very large. Figure 3 shows the effect of the iteration of late resetting on the accuracy of the winning tickets that are produced. Up to a certain point, the later the resetting takes place, the higher the accuracy of the resulting winning tickets. This point of diminishing returns is reached quite early in the training process: iteration 100 (epoch 0.14) for VGG19 and iteration 500 (epoch 1.4) for Resnet-18.

3. Understanding Late Resetting

In light of the empirical evidence that late resetting improves pruning's ability to identify winning tickets, it is worth searching for the underlying mechanism that explains this behavior. One starting point is the observation from Figure 2 that networks found by iterative pruning without late resetting perform similarly to winning tickets that have been randomly reinitialized.

3.1. Stability

We hypothesize that, after a sufficient number of iterations, training is relatively unaffected by pruning. Concretely, for a specific network architecture, iteration, setting of weights, and hyperparameters, we describe the *stability of training to*

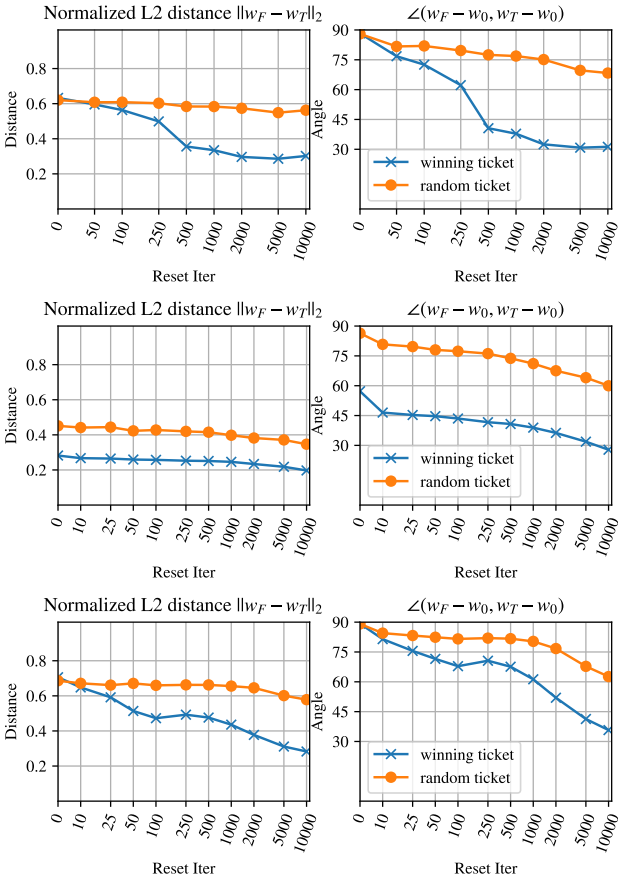


Figure 4. Stability metrics at the end of training for Resnet-18 (top), Lenet (middle), and VGG19 (bottom). W_F and W_T are unpruned weights when they are trained as part of the full network and in isolation, respectively. Random tickets are formed by randomly permuting the per-layer pruning masks of the winning ticket.

pruning (with respect to a specific pruning mask) as the extent to which the optimization trajectories of the unmasked weights produced by training without the mask (i.e., the entire network) and with the mask (i.e., the pruned network) are similar. When training is stable to pruning, the final values of the unpruned weights in both cases are *close*. In contrast, training is unstable to pruning if the two trajectories diverge and, therefore, the final values of the unpruned weights are *far* apart. Furthermore, we hypothesize that winning tickets identifiable by pruning only emerge once the network has become stable to pruning, at which point late resetting becomes effective.

Methodology. To evaluate this hypothesis, we study the stability to pruning of the Lenet (LeCun et al., 1998) fully-connected network for MNIST (where late resetting is unnecessary) and Resnet-18 and VGG19 for CIFAR10 (where late resetting is necessary without warmup). To do so, we compare two network configurations:

1. The full network, trained to final weights W_n at iteration n from initial weights W_0 and then pruned with mask m. We represent this quantity as $W_F = m \odot W_n$.
2. The winning ticket, trained to final weights $m \odot W'_n$ from initial weights $m \odot W_i$ (the full network at iteration i). We represent this quantity as $W_T = m \odot W'_n$.

The mask m is derived by iteratively pruning the full network with late resetting to iteration i. From iteration i, all training stochasticity (i.e., data ordering) is identical between W_F and W_T. We subsequently compare W_F and W_T using two metrics:

1. The normalized L_2 distance between the shared weights of W_F and W_T, i.e., those in m. Specifically, we compute the L_2 distance between W_F and W_T and normalize it by the L_2 distance between the final values of all weights in two training runs of the full network with different initializations and data orders.
2. The angle between $(W_F - W_0)$ and $(W_T - W_0)$.

If the network is stabler to pruning and the winning ticket's weights follow a similar trajectory regardless of whether it is trained in isolation or as part of the full network, then we expect the normalized L_2 distance and angle to be smaller.

3.2. Results

Resnet-18. Figure 4 (top) shows this stability analysis for Resnet-18 in blue. The left plot contains the L_2 distance metric, and the right plot contains the angle metric. Appendix A displays these metrics throughout the entirety of training; Figure 4 includes data only at the end of training.

Resnet-18 requires late resetting to find winning tickets. As depicted in Figure 3 (right), performance begins improving when late resetting to iteration 250, and it saturates when late resetting to iteration 500. The stability data closely follows accuracy. When late resetting to iteration 0, the normalized L_2 distance of the winning ticket is highest, and the weights of the winning ticket trained in isolation are orthogonal to the same weights in the full network. Although the winning ticket slowly becomes stabler with later resetting, it remains little stabler than for iteration 0 until iteration 250—the same iteration at which accuracy begins to improve. By iteration 500, stability improves substantially and then saturates, the same pattern that accuracy follows. This behavior supports our hypothesis on the connection between stability and the efficacy of late resetting.

To contextualize these results, we also consider the stability behavior of a *random ticket*—a subnetwork formed by randomly permuting the masks of the winning ticket within each layer. If this experiment supports our hypothesis, then random tickets will remain unstable regardless of when late resetting takes place. We find this behavior for Resnet-18 (orange line in Figure 4 (top)). For late resetting iteration 0, the winning tickets and random tickets reach the same normalized L_2 distance and angle. In contrast to the winning tickets, the random tickets show no reduction in normalized L_2 distance and little reduction in angle with late resetting. This result enhances our understanding of winning tickets by demonstrating that they are *particularly* stable subnetworks.

Lenet. Unlike Resnet-18, the lottery ticket pruning algorithm can identify winning tickets for Lenet on MNIST when resetting weights to their values at iteration 0. We therefore hypothesize that the winning tickets should be equally stable to pruning for any amount of late resetting. Figure 4 (middle) supports this conjecture: Lenet's stability to pruning (as measured in normalized L_2 distance or angle) changes very little no matter when late resetting takes place. Across every late resetting iteration, the random ticket is equivalently less stable than the winning ticket, matching the gap that we find for Resnet-18 winning tickets.

VGG19. To further evaluate our hypothesis on another network that requires late resetting, we study the same quantities for VGG19 in Figure 4 (bottom). These results again support our proposed relationship between late resetting and stability. Resetting to iteration 0 does not produce winning tickets; likewise, at iteration 0, the winning ticket and random ticket are equally unstable. With late resetting, stability quickly improves, reaching a plateau from iteration 100 to iteration 500. This pattern matches the accuracy improvements seen in Figure 3 (left). After this plateau, stability continues to gradually improve with later resetting, as does accuracy. In comparison, the random ticket generally remains unstable no matter when late resetting takes place, although the angle between the random ticket and the full network decreases somewhat for extremely large iterations

of late resetting. This decrease is expected, since the random ticket has fewer iterations for which to optimize for large late-resetting values.

Summary. Across all three networks, the degree of a winning ticket's stability to pruning for a particular late resetting iteration appears to correlate with the performance it achieves when it is initialized starting at that iteration. In Section 4, we find this behavior for Resnet-50 on ImageNet.

3.3. Discussion

We find that there is an iteration after which the training process becomes more stable to pruning; moreover, this iteration correlates with the iteration at which late resetting becomes more effective. If this link between stability and late resetting is indeed causal, then we can explain the late resetting behavior we observe in terms of stability as follows: Frankle & Carbin have demonstrated that winning tickets are sensitive to their initializations. If a winning ticket is initialized to an iteration before training becomes stable to pruning, then the instability will cause the winning ticket to follow a different optimization trajectory to a less desirable optimum. This instability so destabilizes a winning ticket that it is tantamount to randomly reinitializing it. In contrast, if a winning ticket is initialized to an iteration after training becomes stable to pruning, then the original optimization trajectory is preserved, leading the subnetwork to an optimum similar to that reached by the full network.

These stability results provide circumstantial evidence that stability to pruning is a property of winning tickets that are identified by pruning. When networks so identified are not stable to pruning, they are unable to match or exceed the accuracy of the original network, meaning they fail to meet the definition of a winning ticket. Late resetting makes it possible to initialize these subnetworks such that they become stable to pruning, correspondingly restoring the typical lottery ticket accuracy behavior.

4. Resnet-50 on ImageNet

The same challenges encountered in Resnet-18 and VGG19 manifest for Resnet-50 (He et al., 2016) for ImageNet (Russakovsky et al., 2015): naively applying the lottery ticket pruning strategy to a standard Resnet-50 implementation with unmodified hyperparameters does not yield winning tickets. However, our results show that late resetting produces trainable winning tickets more than 79% smaller than the original network. These are the first winning tickets to be found for Resnet-50 on ImageNet.

4.1. Methodology

We use a standard Resnet-50 implementation for TPUs (Google, 2018) with unmodified hyperparameters. The network has a total batch size of 1024 across eight TPU cores. Training begins with a learning rate of 0 that is linearly warmed up to 4 over the first five epochs, after which it decreases by an order of magnitude at epochs 30, 60, and 80 with training ending at epoch 90. Each data point is the average of three experiments, with error bars for the maximum distance of any experiment from the average. The original network reaches top-1 accuracy of 76.14±0.08%, which serves as our accuracy standard for a winning ticket.

Late Reset Epoch	Fraction of Network Remaining				
	70%	50%	30%	20%	10%
R	75.55±0.07	74.79±0.08	73.43±0.13	72.31±0.13	69.32±0.20
0	75.52±0.06	74.78±0.03	73.62±0.20	72.24±0.11	69.43±0.03
2	75.84±0.04	75.52±0.06	74.69±0.07	73.81±0.04	71.20±0.20
4	*76.25±0.21**	*76.14±0.14**	75.96±0.09	75.35±0.13	73.18±0.13
6	*76.25±0.16**	76.11±0.13	75.93±0.08	75.50±0.25	73.61±0.22
8	*76.15±0.19**	*76.22±0.10**	75.99±0.10	75.61±0.06	73.57±0.09
10	*76.25±0.14**	*76.16±0.18**	75.93±0.14	75.54±0.21	73.66±0.12

Figure 5. The effect of late resetting epoch (rows) at various levels of sparsity (columns) on test accuracy for Resnet-50 when trained to completion and pruned in one-shot. The row marked R is the result of randomly reinitializing the network. Test accuracy is averaged over three trials. * indicates a winning ticket.

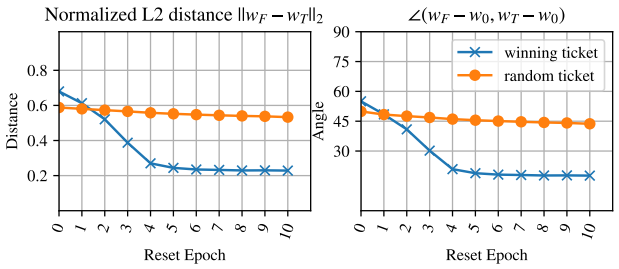


Figure 6. Stability metrics at the end of training for Resnet-50. W_F and W_T are unpruned weights when they are trained as part of the full network and in isolation, respectively. Random tickets are formed by randomly permuting the per-layer pruning masks of the winning ticket. The L_2 distance is normalized by the maximum distance between two full networks W_F and W'_F at any iteration.

4.2. One-shot Pruning

Figure 5 shows the effect of late resetting on one-shot pruning's ability to find winning tickets. Without late resetting, we are unable to find winning tickets even with 70% of the original network remaining, matching the findings of Liu et al. (2019) and Gale et al. (2019). The accuracy of the pruned networks increases as resetting becomes later, culminating in winning tickets when 50% of the weights remain when late resetting to epoch 4 and above (and near-winning tickets when 70% of the weights remain). Without late resetting, the network performs identically to when randomly reinitialized (top row). After epoch 4, the benefits of further late resetting diminish, so we reset to epoch 6 in the rest of

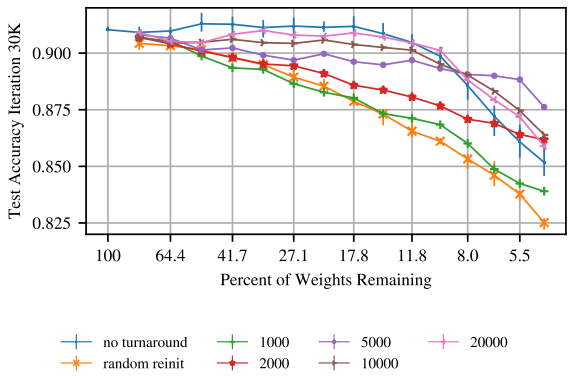


Figure 7. The accuracy achieved by winning tickets found via iterative pruning on Resnet-18 (late resetting to iteration 500) on CIFAR10 with early turnaround at the specified iterations.

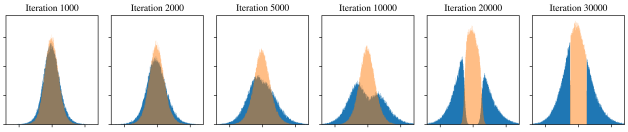


Figure 8. Snapshots of the distribution of weights in Resnet-18 at different iterations (columns). Light color is weights that are eventually pruned. Dark color is weights that remain.

our experiments.

These observations closely align with the stability metrics for Resnet-50 (Figure 6). When resetting to epoch 0, the winning ticket and random ticket are equally unstable, just as the accuracy of the winning ticket is no better than when randomly reinitialized. With later resetting, stability and accuracy gradually improve. Both of these quantities saturate with late resetting to epoch 4, after which neither stability nor accuracy further improve. In contrast, the random ticket remains equally unstable no matter when late resetting occurs. This close correspondence between the effect of late resetting on accuracy and stability under pruning matches the results in Section 3 for Resnet-18 and VGG19.

		Fraction of Network Remaining				
		70%	50%	30%	20%	10%
Turn. Epoch	15	75.76±0.03	75.27±0.10	74.65±0.08	73.71±0.15	71.70±0.12
	30	*76.15±0.17**	75.98±0.11	75.43±0.20	74.83±0.12	72.93±0.03
	45	*76.28±0.19**	76.12±0.14	75.75±0.11	75.34±0.06	73.56±0.30
	65	*76.30±0.12**	*76.19±0.07**	75.97±0.19	75.63±0.14	73.56±0.11
	90	*76.25±0.21**	*76.14±0.14**	75.96±0.09	75.35±0.13	73.18±0.13

Figure 9. The effect of early turnaround epoch (rows) at various levels of sparsity (columns) on test accuracy for Resnet-50 on ImageNet when trained to completion and pruned in one-shot by the specified amount. Late resetting is at epoch 6. Asterisks indicate winning tickets.

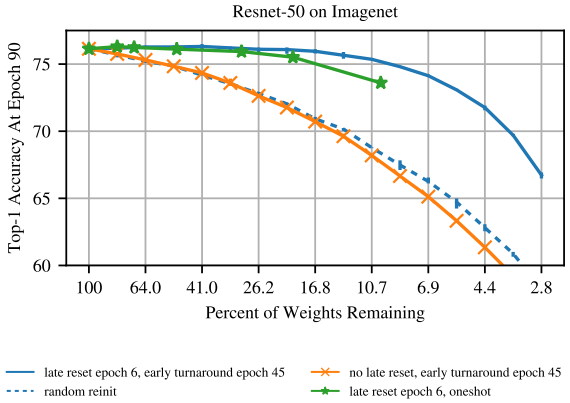


Figure 10. The accuracy achieved by winning tickets found via one-shot and iterative pruning on Resnet-50 for ImageNet. Each line is the average of three trials. Error bars are present but small.

4.3. Iterative Pruning

Frankle & Carbin achieve the smallest winning tickets when they perform iterative pruning: training a network, pruning a small fraction of weights, resetting, and repeating several times to gradually whittle a large network into a winning ticket. This procedure is particularly expensive on Resnet-50, which takes nearly a day to train. To reduce the cost of iterative pruning, we explore a technique that we term *early turnaround* in which we prune a network before it has completed the training process.

Early Turnaround. Early turnaround involves a tradeoff between computational efficiency and winning ticket quality: it can substantially reduce the cost of finding winning tickets, but it adversely affects the performance of the winning tickets that are eventually identified.

Figure 7 shows the effect of early turnaround on the accuracy of winning tickets (with iterative pruning) for Resnet-18. The later the turnaround, the better the accuracy of the winning tickets. Even performing early turnaround at iteration 20,000 still underperforms pruning at the end of training. Figure 8 shows the magnitude distributions of weights that are eventually pruned (one-shot, 67% pruned) from Resnet-18 at the end of training. Performing early turnaround before iteration 20,000 will prune many weights that would otherwise have been in the winning ticket.

Figure 9 shows the effect of the iteration of early-turnaround on the accuracy of winning tickets found via one-shot pruning, which we use as a guide for selecting our hyperparameters for iterative pruning. In general, the later pruning is performed, the higher the accuracy of the winning tickets. For our iterative pruning experiments, turnaround at epoch 45, which balances computational efficiency and accuracy.

Results. Figure 10 shows the results of the lottery ticket

experiment on Resnet-50 for ImageNet, including iterative (blue) and one-shot (green) results with late resetting (epoch 6) and early turnaround (epoch 45). It also includes lines for randomly reinitializing the network (blue dashed) and performing iterative pruning with no late resetting (orange). The smallest winning ticket we find uses iterative pruning, removing 20% of weights per iteration based on the weights at epoch 45. This configuration reaches on average 76.1% top-1 accuracy when pruned by 79%. Top-1 accuracy remains within a percentage point of the full network when pruning by up to 89%. When randomly reinitialized, winning tickets follow the typical lottery ticket pattern, with accuracy steadily diminishing under further pruning. Late resetting is crucial for achieving these results; without it, the networks found by iterative pruning barely outperform the random reinitialization experiments.

5. Limitations

Our results remain within the scope of vision datasets MNIST, CIFAR10, and ImageNet. While we extend Frankle & Carbin's work to include ImageNet, our core technique for identifying winning tickets is still unstructured, magnitude-based pruning (among the wide variety of contemporary pruning techniques, e.g., Hu et al. (2016); Srinivas & Babu (2015); Dong et al. (2017); Li et al. (2016); Luo et al. (2017); He et al. (2017)). Moreover, given that unstructured pruning does not necessarily yield networks that execute more quickly with commodity hardware or libraries, we primarily intend our results to convey insight on neural network behavior rather than suggest immediate opportunities to improve neural network performance.

6. Discussion

Winning ticket initialization. Our stability analysis takes a step toward describing a mechanism underlying the existence and efficacy of winning tickets: we only find winning tickets with pruning when the tickets are stable to pruning. The initialization of a winning ticket is important insofar as it provides a starting point from which a winning ticket can follow similar optimization trajectories whether it is trained in isolation or as part of the larger network. This picture matches the observations of Frankle & Carbin (2019), who show that individual weights within a winning ticket change in magnitude more drastically than other weights (in order to refute the hypothesis that winning tickets simply start in a well of a minimum on the optimization landscape). With this data, they argue that winning tickets are initialized such that the optimization algorithm can find a feasible optimization trajectory with high probability over the distribution of possible data orders.

Winning ticket structure. Frankle & Carbin conjecture that the structure of winning tickets (i.e., the unpruned weights within the full network) encode an inductive bias (Cohen & Shashua, 2016) tailored to the task at hand. We believe that late resetting provides additional evidence for this hypothesis. Specifically, we conjecture that the initial stage of learning up until the moment of late resetting identifies a data-dependent inductive bias—i.e., a projection of the full network—for which training is stable.

Neural network optimization. Recent results show that overparameterized networks (with restrictions) optimized with SGD converge to a global optimum (Du et al., 2019; Allen-Zhu et al., 2018). In light of the lottery ticket hypothesis, one interpretation of this area of work is that there exist small subnetworks within overparameterized networks that can train to the same accuracy. It is worth exploring whether SGD seeks out a winning ticket or another variety of low-dimension representation during optimization. Studying the emergence of low-dimensional structures that are stable to pruning provides a starting point for this direction.

7. Conclusion

The lottery ticket hypothesis hints at future techniques that identify small subnetworks capable of matching the accuracy of larger networks in a similar number of iterations. To date, this line of work and other related research have focused on compressing neural networks at initialization time. In this work, we find that other moments early in the training process may present better opportunities for this class of techniques.

We show that one of the key challenges encountered by the lottery ticket pruning regime on larger networks at higher learning rates can be overcome by resetting pruned networks to their weights just after initialization (rather than at iteration 0, as originally proposed). This technique, which we term *late resetting*, makes it possible to find small, trainable subnetworks (*winning tickets*) for Resnet-50 on ImageNet.

In order to understand the efficacy of late resetting, we study a measure of the stability of neural network training in response to pruning. We find that, for a task for which winning tickets can be reinitialized to iteration 0 (Lenet on MNIST), the winning ticket's weights follow a similar trajectory throughout training, regardless of whether they are trained in isolation or as part of the larger network. In contrast, for tasks that requires late resetting (Resnet-18 and VGG19 on CIFAR10 and Resnet-50 on ImageNet), these weights are unstable at initialization but become more stable as accuracy improves under late resetting.

In summary, we present a novel technique for scaling the lottery ticket hypothesis to state-of-the-art benchmarks while offering new insights into the dynamics underlying the lottery ticket hypothesis and neural network training.

References

Allen-Zhu, Z., Li, Y., and Song, Z. A convergence theory for deep learning via over-parameterization, 2018.

Cohen, N. and Shashua, A. Inductive bias of deep convolutional networks through pooling geometry. *arXiv preprint arXiv:1605.06743*, 2016.

Dong, X., Chen, S., and Pan, S. Learning to prune deep neural networks via layer-wise optimal brain surgeon. In *Advances in Neural Information Processing Systems*, pp. 4860–4874, 2017.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1eK3i09YQ.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *Int. Conf. Represent. Learn.*, 2019.

Gale, T., Elsen, E., and Hooker, S. The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574*, 2019.

Google. Resnet50 for imagenet on tpus, 2018. URL https://github.com/tensorflow/tpu/tree/master/models/official/resnet.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch SGD: training Imagenet in 1 hour, 2017.

Han, S., Pool, J., Tran, J., and Dally, W. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, pp. 1135–1143, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

He, Y., Zhang, X., and Sun, J. Channel pruning for accelerating very deep neural networks. In *International Conference on Computer Vision (ICCV)*, volume 2, pp. 6, 2017.

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

Hu, H., Peng, R., Tai, Y.-W., and Tang, C.-K. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. *arXiv preprint arXiv:1607.03250*, 2016.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lee, N., Ajanthan, T., and Torr, P. H. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*, 2018.

Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. P. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

Liu, Z., Sun, M., Zhou, T., Huang, G., and Darrell, T. Rethinking the value of network pruning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJlnB3C5Ym.

Luo, J.-H., Wu, J., and Lin, W. Thinet: A filter level pruning method for deep neural network compression. *arXiv preprint arXiv:1707.06342*, 2017.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3): 211–252, 2015.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Srinivas, S. and Babu, R. V. Data-free parameter pruning for deep neural networks. *arXiv preprint arXiv:1507.06149*, 2015.

A. Detailed Stability Data

A.1. Overview

In Figures 4 and 6, we present stability metrics at the end of training for our sample networks (Lenet for MNIST, Resnet-18 for CIFAR10, VGG19 for CIFAR10, and Resnet-50 for ImageNet). In this Appendix, we present these metrics across the entire training process, along with additional metrics on the relationship between weights in the full network and weights in the winning and random tickets.

This data appears in Figures 11 (for Lenet), 12 (for Resnet-18), 13 (for VGG19), and 14 (for Resnet-50). Each figure has two subparts, comparing the full network and winning ticket (part a) and the full network and random ticket (part b). Each part has one column for each late-resetting value and four rows plotting different stability metrics for each of these late-resetting values.

Row 1: weight correlation. A scatter plot of the value of a weight in W_T (y-axis) in comparison to the value of the same weight in W_F (x-axis). Colors represent the density of points as determined using kernel density estimation.

Row 2: L_2 distance between networks. The L_2 distance between the full network and the winning ticket or random ticket (i.e., $||W_F - W_T||_2$)—the same quantity portrayed in the left column of Figures 4 and 6. Where those Figures only show this distance at the end of training, row 2 plots this distance at every training iteration. In Figures 11–13, this distance is normalized by the distance between all weights in two training runs of the full network with different initializations and data orders. In Figure 14, this distance is normalized by the maximum value of $||W_F - W'_F||_2$ at any iteration.

Row 3: L_2 distance from initialization. The normalized L_2 distance between W_F (blue) or W_T (orange) and the initial weights W_0 for each training iteration. This distance is normalized by the same quantities as row 2.

Row 4: angles between networks. The angle between various networks for each iteration of training. It includes three different angle measurements:

- Blue: the angle between $(W_F - W_0)$ and $(W_T - W_0)$—the same quantity portrayed in the right column of Figures 4 and 6.
- Orange: the angle between $(W_T - W_0)$ and $(W_* - W_0)$ (where W_* are the final trained weights of W_F).
- Green: the angle between $(W_F - W_0)$ and $(W_* - W_0)$.

A.2. General Discussion

The second and fourth rows in Figures 11–14 track the distance and angle metrics as described in Section 3 under Methodology for our sample networks during training. The L2 distance between a ticket and the corresponding full network grows quickly at the start of training. After this initial growth, the L2 distance plateaus for the rest of training. This growth-and-plateau pattern appears for all tickets—random and winning—and for all sample networks.

The growth in distance coincides with the growth of the L2 norm between the network weights and the initial weights, as presented in the third row in the figures. The third row plots also demonstrates that the norm of W_F is consistently smaller than the norm of W_T. One explanation is that some of the weights in W_T have to become large to compensate for all the zeroed weights. This compensation might ensure that the network parameterized by W_T produces outputs of a similar magnitude to those produced by the full network parameterized by W. The weights in W_F are only a subset of the network that was trained, so we do not expect the L2 norms of W_T and W_F to be identical.

The top row of each figure measures the correlation between a weight's value in the full network and the ticket network. The performance of the full network drops after it is pruned. Therefore, perfect correlation between W_T and W_F might have a detrimental effect on the performance of a subnetwork. In particular, very high correlation would mean that we can do no better than a trained-and-pruned network by re-training the winning tickets. When the winning tickets perform comparably to the full network, we expect to see some but not perfect correlation of W_T and W_F at the end of training. Our empirical findings summarized in the top row of Figures 11–14 are consistent with this intuition. The correlation between W_T and W_F starts increasing and is positive at around the same late reset iteration for which a winning ticket emerges. For the random tickets, we see no to little correlation with W_F.

The final row in Figures 11–14 shows the dynamics of the angle during training between three network pairs (as described in the previous section). Fixing the origin at W_0, the angle between W_F and $W*$ (the full network weights at the end of training) changes during training, suggesting that the weights of the full network W_F are not only growing in L2 norm, but are also slowly rotating around W_0. The orange line tracks the angle between the ticket weights W_T and trained full network weights W^*. When the chosen subnetwork is not a winning ticket, W_T is nearly perpendicular to W^* throughout the whole training time. In contrast, the winning ticket weights are more aligned with W_F at every point in training.

A.3. Case Study: Lenet and Resnet-18

In this Subsection, we discuss the stability behavior of Lenet (which does not require late-resetting) and Resnet-18 (which does). The patterns for Resnet-18 generally repeat for VGG19 and Resnet-50, both of which also require late resetting.

Lenet winning ticket. As we observe in Section 3, Lenet remains similarly stable across all four metrics no matter when late resetting occurs, and all four metrics indicate stability:

- Ticket and full-network weights generally correlate well with one another (row 1).
- The normalized L_2 distance between the full network and the winning ticket is small (row 2).
- The distances between the origin and the full network or the winning ticket are nearly identical throughout

training (row 3).

- The angle between the winning ticket and the full network is well below 90 degrees (row 4)

These observations, which are consistent across late resetting iterations, are consistent with the fact that Lenet does not require late resetting to find winning tickets.

Lenet random ticket. In contrast, the Lenet random ticket appears less stable.

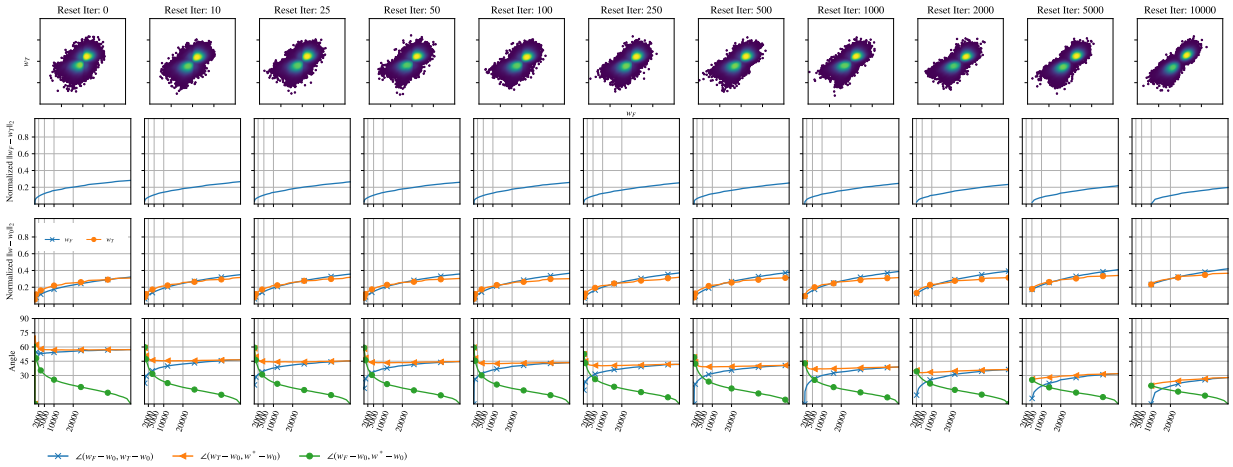
- There is far less correlation between a weight's value in the random ticket and and in the full network (row 1).
- The L_2 distance between the full ticket and random ticket is larger (row 2).
- The random ticket weights travel much further from the origin than the same weights in the full network. The random ticket weights in the full network travel much less far than do the winning ticket weights in the full network (row 3).
- The angle between the random ticket weights and full ticket weights is almost 90 degrees for iteration 0. It decreases when training starts later, likely because the network is trained for fewer iterations so the weights cannot travel as far afield (row 4).

Resnet-18 winning ticket. As we observe in Sections 3, the stability of a winning ticket for Resnet-18 improves with late resetting, reaching a point of diminishing returns at about iteration 500, and accuracy follows a similar pattern. Likewise:

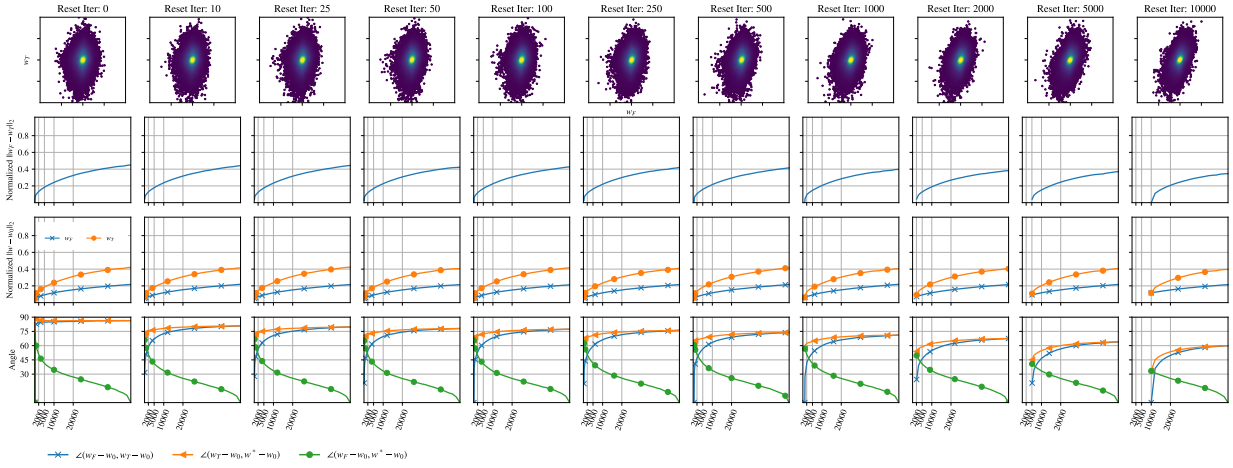
- Correlation improves until this iteration (row 1).
- The distance between the full network and winning ticket decreases until this iteration (row 2).
- The distances traveled from the origin by the full network and winning ticket become more similar until this iteration and stabilize afterwards (row 3).
- The angles between the full network and winning ticket start off at 90 degrees for resetting to iteration 0, and stabilize at about 45 degrees by late resetting iteration 500 with small improvements thereafter as seen with Lenet (row 4).

Resnet-18 random ticket. In contrast, the random ticket remains unstable no matter when late resetting takes place.

- There is little correlation between weights in the full ticket and random ticket for any late resetting iteration (row 1).
- The distance between the full network and random ticket remains high no matter when resetting takes place (row 2).
- The random ticket weights in the full network remain much closer to the origin than do the same weights in the random ticket. For the winning ticket, these quantities became closer with later resetting (row 3).
- The angle between the full network and random ticket remains close to 90 degrees regardless of when late resetting takes place, although it decreases slightly as in all of the other experiments (row 4).

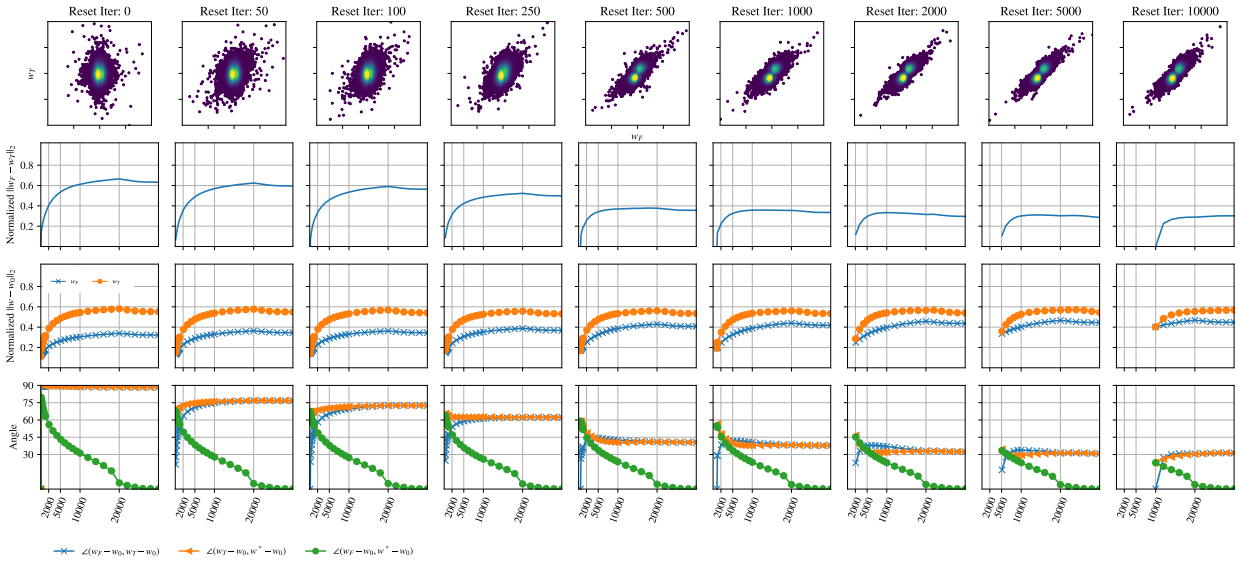


(a) Winning tickets.

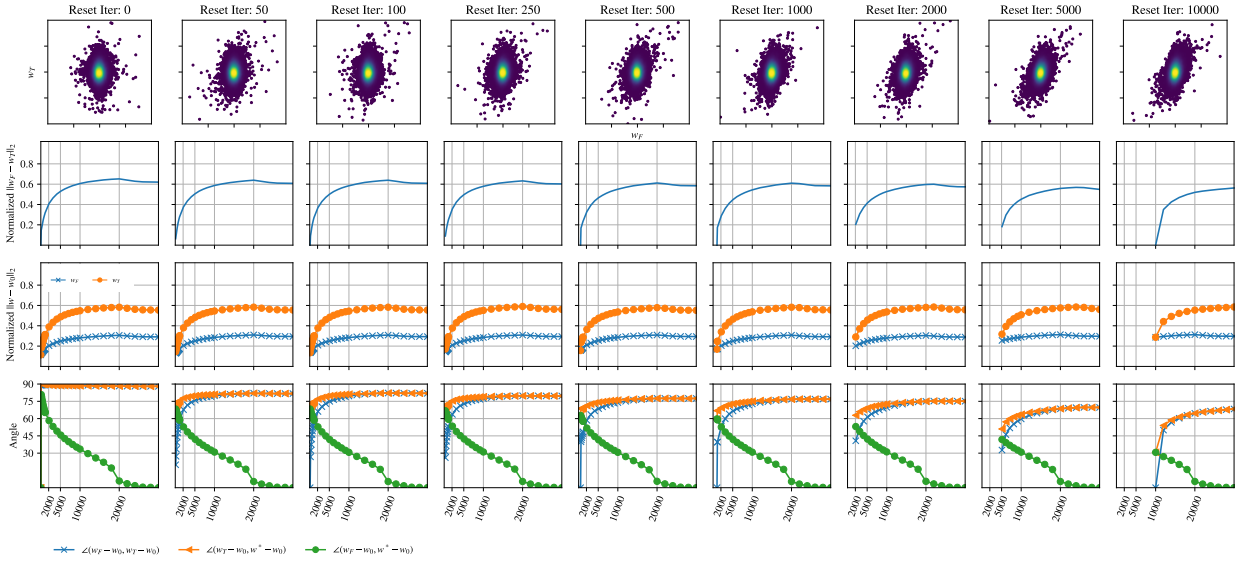


(b) Random tickets.

Figure 11. Lenet.

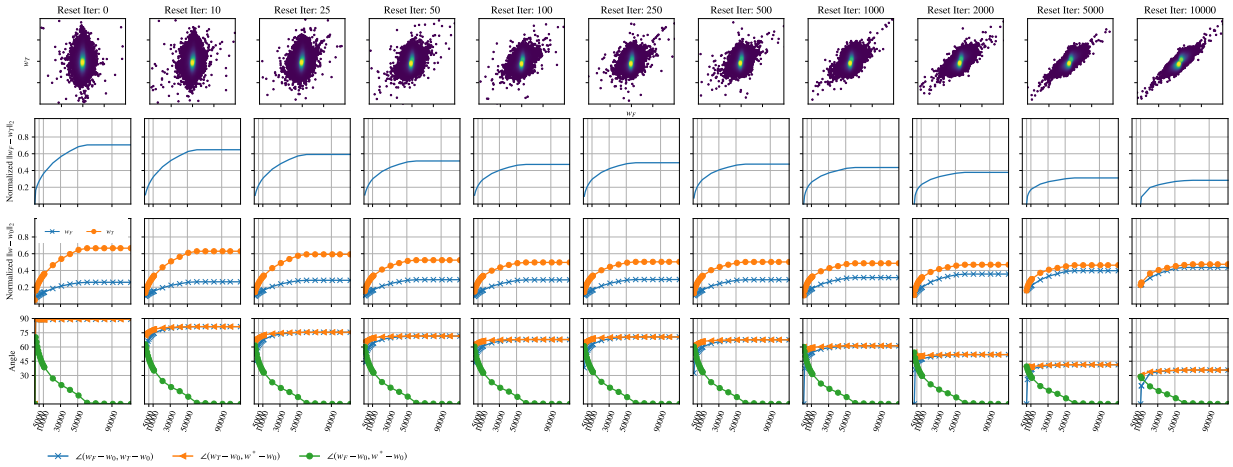


(a) Winning tickets.

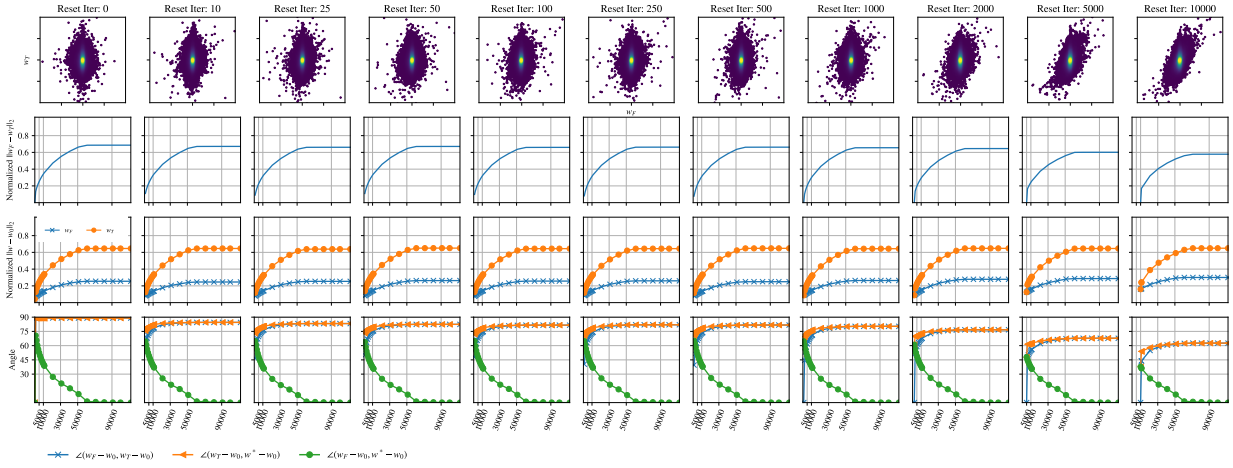


(b) Random tickets.

Figure 12. Resnet-18.

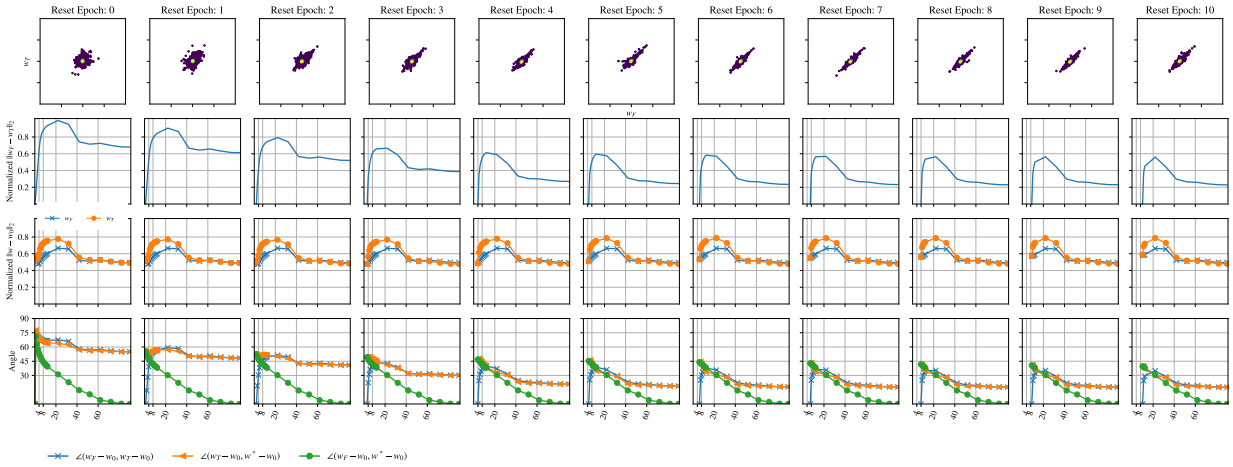


(a) Winning tickets.

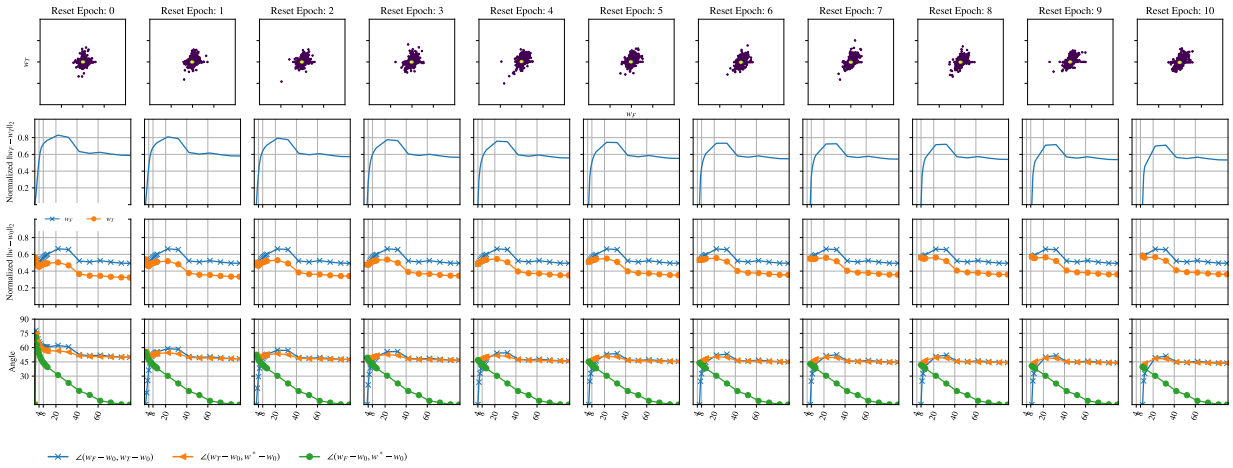


(b) Random tickets.

Figure 13. VGG19.



(a) Winning tickets.



(b) Random tickets.

Figure 14. Resnet-50.