Sanity Checks for Lottery Tickets: Does Your Winning Ticket Really Win the Jackpot?

Anonymous Author(s)
Affiliation
Address
email

Abstract

There have been long-standing controversies and inconsistencies over the experiment setup and criteria for identifying the "winning ticket" in literature. To reconcile such, we revisit the definition of lottery ticket hypothesis, with comprehensive and more rigorous conditions. Under our new definition, we show concrete evidence to clarify whether the winning ticket exists across the major DNN architectures and/or applications. Through extensive experiments, we perform quantitative analysis on the correlations between winning tickets and various experimental factors, and empirically study the patterns of our observations. We find that the key training hyperparameters, such as learning rate and training epochs, as well as the architecture characteristics such as capacities and residual connections, are all highly correlated with whether and when the winning tickets can be identified. Based on our analysis, we summarize a guideline for parameter settings in regards of specific architecture characteristics, which we hope to catalyze the research progress on the topic of lottery ticket hypothesis. Codes will be fully released.

1 Introduction

In recent years, the Lottery Ticket Hypothesis (LTH) [1] has drawn great attention and thorough research efforts. As an important study to investigate the initialization state and network topology of the deep neural networks (DNNs), LTH claims the existence of a winning ticket (i.e., a properly pruned subnetwork together with original weight initialization) that can achieve competitive performance to the original dense network, which highlights great potential for efficient training and network design.

Unfortunately, among the various researches on the lottery ticket hypothesis [2, 3, 4, 5, 6], there are many inconsistencies regarding the settings of training recipe, and they further lead to the controversies over the conditions for identifying winning tickets. We revisit and analyze the definition of the original lottery ticket hypothesis and find that the quality of training recipe is a critical factor for the network performance, which in fact, is largely missing in previous discussions.

In the standard LTH setup [1], key training hyperparameters such as learning rate and training epochs were not much scrutinized nor exhaustively tuned. The winning ticket can be identified in the case of small learning rate, but can fail to emerge at higher initial learning rates especially in deeper networks. For instance, in [1], the winning tickets can be identified only in the case of small learning rate 0.01 on networks such as ResNet-20 and VGG-19 on CIFAR-10. At larger learning rates, however, [7] reveals that the "winning ticket" has no accuracy advantage over the random reinitialization , which contradicts with the original definition. On the other hand, the settings in [1] train 78 epochs for ResNet-20 on CIFAR-10. Such insufficient training causes a relatively low pretraining accuracy. When pruned iteratively, the subnetwork accuracy can easily match that pretraining accuracy of the original network. Under such experimental conditions, the existence of the winning ticket is questionable.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

In addition to all the problems caused by the experimental conditions, the huge computational consumption to find a winning ticket becomes another research barrier and the practical main drawback, limiting the observations made on LTH. For instance, to reach around 90% overall sparsity ratio, iterative magnitude-based pruning (`IMP`) in [1] requires totally 11 iterations (20% of the weights are pruned in each iteration). It adds up to 1,760 total training epochs if each iteration consumes 160 epochs. On the other hand, as an efficient pruning method, one-shot magnitude-based pruning (`OMP`) prunes a pretrained DNN model to arbitrary target sparsity ratio in one shot, which greatly saves training efforts. However, `OMP` is rarely considered in the related literature, and is often deemed as "weak" without full justification. Based on the above reasons, we feel we cannot confidently draw arguments, before we are able to evaluate LTH comprehensively in regards of key factors such as different network structures, network dimensions, and training dataset sizes.

In this paper, we dive deeper into the underlying condition of the lottery ticket hypothesis. We raise the following questions: (1) What makes the comprehensive condition to define the lottery ticket hypothesis? (2) Do winning tickets exist across the major DNN architectures and/or applications under such definition? and (3) What are the intrinsic reasons for their existence or non-existence?

To answer the above questions, we present our rigorous definition of the lottery ticket hypothesis, which specifies settings of the training recipe, the principles for identifying winning tickets, and the rationality on examining the winning ticket existence. Under this rigorous definition, we perform extensive experiments with many representative DNN models and datasets. The relationships between winning tickets and various factors are quantitatively analyzed. We empirically study the patterns through our analysis, and develop a guideline to ease the process of obtaining the winning ticket. Our findings open up many new questions for future work. We summarize our contributions as follows:

I. We point out that the usage of inappropriately small learning rates, insufficient training epochs, and other inconsistent and implicit conditions for identifying winning ticket in the literature, are the main reasons that cause controversies in the lottery ticket studies.

II. We propose a more rigorous definition of the winning ticket, and evaluate the proposed definition on different training recipe, DNN architecture, network dimension, and the training data size. Somehow surprisingly, we find that under the new rigorous definition, *no* "rigorous" winning tickets are found by current methods, while there do exist winning tickets under a slightly looser definition.

III. We find that when residual connections exist in the network, using a relatively small learning rate is more likely to find (close to) winning tickets. When no residual connection exists, the `IMP` method may not be necessary because `OMP` can achieve equivalent performance.

IV. We also find that when a smaller learning rate is not favorable, initialization is likely to make no difference in finding the winning ticket (e.g., lottery initialization is not necessary). We quantitatively analyze the patterns, and present a guideline to help identify winning tickets.

2 Re-defining Lottery Ticket Hypothesis

2.1 Notations and Preliminary

In this paper, we follow the notations from [1, 5]. Detailed notations and functions are listed in Table 1. Based on Table 1, we provide several key LTH-related settings along with descriptions.

Consider a network function $f(\cdot)$ that is initialized as $f(x;\theta_0)$ where x denotes input training samples. We define the following settings:

- *Pretraining:* We train the network $f(x;\theta_0)$ for T epochs, arriving at weights θ_T and network function $f(x;\theta_T)$.
- *Pruning*: Based on the trained weights θ_T, we adopt $\texttt{OMP}(\theta_T, s)$ or $\texttt{IMP}(\theta_T, s)$ to generate a pruning mask $m_O, m_I \in \{0,1\}^{|\theta|}$. Note that for `IMP`, the same θ_0 is used in each iteration to ensure fairness to `OMP`.
- *Lottery ticket with `OMP` (LT-`OMP`):* We directly apply mask m_O to initial weights θ_0, resulting in weights $\theta_0 \odot m_O$ and network function $f(x;\theta_0 \odot m_O)$.
- *Lottery ticket with `IMP` (LT-`IMP`):* We apply m_I to initial weights θ_0, and get $f(x;\theta_0 \odot m_I)$.
- *Random reinitialization with `OMP` (RR-`OMP`):* We apply mask m_O to the random reinitialized weights θ_0', and get network function $f(x;\theta_0' \odot m_O)$.

Table 1: Summary of notations and functions.

Notation	Description
T	T is the total number of training epochs.
$\theta_0, \theta_t, \theta'_0$	$\theta_0 \sim \mathcal{D}_\theta$ denotes initial weights used for training. θ_t is the weights that is trained from θ_0 for t epochs where $t \leq T$. $\theta'_0 \sim D_\theta$ denotes a random reinitialization that is different from θ_0.
m	A sparse mask $m \in \{0,1\}^{\lvert\theta\rvert}$ is obtained from certain pruning algorithm.
s	s is the sparsity ratio, which is defined as the percentage of pruned weights in the DNN model.
θ^{SD}	θ^{SD} denotes the weight in a small-dense model that has the same number of non-zero parameters as a pruned model, i.e. $\theta^{SD} \sim \mathcal{D}_{\lvert\lvert m \rvert\rvert}$.
$\texttt{OMP}(\theta, s)$	One-shot Magnitude-based Pruning [8] that prunes θ_T and returns m, i.e. $m_O \leftarrow \texttt{OMP}(\theta_T, s)$. It prunes $s \times 100\%$ of weights in a one-time operation manner.
$\texttt{IMP}(\theta, s)$	Iterative Magnitude-based Pruning [9] that prunes θ_T and returns m, i.e. $m_I \leftarrow \texttt{IMP}(\theta_T, s)$. $\texttt{IMP}(\cdot)$ prunes 20% of remaining weights per iteration until arriving at target sparsity s [5].

- *Random reinitialization with* `IMP` *(RR-*`IMP`*):* We apply m_I to random reinitialized weights θ'_0, and get $f(x; \theta'_0 \odot m_I)$.
- *Small-dense training (SDT):* We construct a small-dense network that has the same depth and reduced width compared to the original network, and initialized by θ^{SD}, i.e. $f(x; \theta^{SD})$.

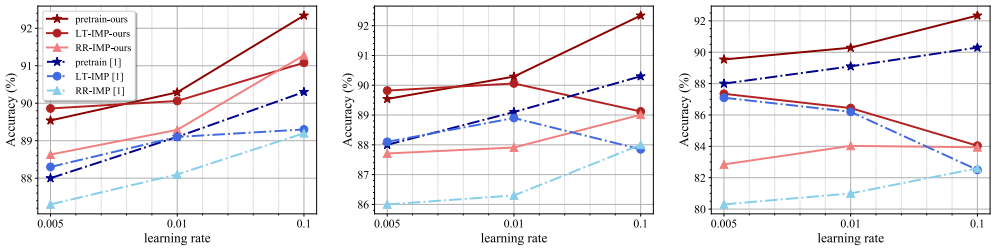
Original definition of the winning ticket: The original lottery ticket hypothesis [1] claims that there exists subnetwork $f(x; \theta_0 \odot m)$ in a randomly initialized dense network $f(x; \theta_0)$, that once trained for T epochs (or fewer) will result in similar accuracy as $f(x; \theta_T)$, under a non-trivial sparsity ratio. Additionally, the accuracy of $f(x; (\theta_0 \odot m)_T)$ should be noticeably higher than $f(x; (\theta'_0 \odot m)_T)$. Note that $(\theta_0 \odot m)_T$ and $(\theta'_0 \odot m)_T$ are the initial and the randomly reinitialized weights of the sparse subnetwork trained for T epochs, respectively. When the above conditions are met, $(\theta_0 \odot m)$ can be considered the *Winning Ticket*.

We define a network is *well-trained*, if it is trained using a sufficient training recipe (i.e., an appropriate learning rate and sufficient training epochs). However, in many prior works such as [1], the pretraining of the lottery ticket experiments used an insufficient training recipe (i.e., inappropriately small learning rate and fewer training epochs), which leads to non-optimal pretraining accuracy at relatively low levels. Apparently, a higher pretraining accuracy is more difficult for a subnetwork to match or "win the ticket", even by using a sufficient training recipe.

We further revisit the LT-`IMP` and RR-`IMP` experiments using ResNet-20 on CIFAR-10 dataset, at three different learning rates over a range of different sparsity ratios ([1] uses the small learning rate 0.01). We train the subnetworks with the same training recipe in pretraining, and we also adopt the settings in [1] to reproduce the results. Our preliminary results are shown in Figure 1.

Through Figure 1(a), 1(b), our first observation is that, under either training recipe, the "winning ticket" exists in smaller learning rates (e.g., 0.005 and 0.01), but does not exist at a relatively larger learning rate (e.g., 0.1). For instance, in the cases of the initial learning rate of 0.005 and 0.01, we find a noticeable accuracy gap between LT-`IMP` and RR-`IMP` using both training recipes, and the LT-`IMP` accuracy is close to the pretraining accuracy with a reasonable sparsity ratio (e.g., 50% or above). This is similar to the observations found in [1] on the same network and dataset. On the other hand, in the case of the initial learning rate of 0.1, the LT-`IMP` has a similar accuracy performance as the RR-`IMP`, and cannot achieve the accuracy close to the pretrained DNN with a reasonable sparsity ratio, thus no winning ticket condition is satisfied.

Through Figure 1(c), our second observation is that, at the same learning rate, the winning ticket defined in [1] can be identified by using an insufficient training recipe, but fails to satisfy the winning ticket condition when the network is well-trained. For instance, in the case of initial learning rate of 0.005, [1] uses approximately 78 epochs for training the network, which achieves 88.0% pretraining accuracy, 87.1% on LT-`IMP` and 80.3% on RR-`IMP`, respectively. The LT-`IMP` accuracy is close to the pretraining accuracy, and outperforms RR-`IMP`, thus it is claimed in [1] that the winning ticket is found. However, when we train the network with a sufficient number of epochs (160 in our settings),



(a) sparsity ratio $s = 0.59$ (b) sparsity ratio $s = 0.832$ (c) sparsity ratio $s = 0.914$

Figure 1: Preliminary results of ResNet-20 on CIFAR-10 dataset with different learning rates and sparsity ratios. We train the network using 160 epochs, while [1] uses 78 epochs.

the accuracy of pretraining, LT-`IMP`, and RR-`IMP` is 89.6%, 87.4%, and 82.9%, respectively. In this case, the accuracy gap between pretraining and LT-`IMP` is not small enough to claim that they are "similar", thus in fact no winning ticket is found.

Takeaway: The above two observations indicate that the winning tickets are more likely to exist at a small learning rate or at an insufficient training epochs, but may not exist at a relatively large learning rate or sufficient training epochs (also observed in [7]). However, we would like to point out that using a relatively large learning rate (e.g., 0.1) and sufficient training epochs (e.g., 160, which is the standard settings on CIFAR-10) result in a *notably higher accuracy* for the pretrained DNN (92.3% vs. 88.0%). This point is largely missing in the previous discussions, and questions whether the previously identified "winning tickets" are meaningful enough.

2.2 A Rigorous Definition of the Lottery Ticket Hypothesis

The above discussion reveals the inconsistency of identifying the winning ticket under different conditions. We provide a more rigorous definition of lottery ticket hypothesis to reconcile the long-standing winning ticket identification discrepancy between experiment settings. Our goal is to investigate the precise conditions on when winning ticket exists and how to identify them.

The lottery ticket hypothesis – a rigorous definition. *Under a non-trivial sparsity ratio, there exists an identically initialized subnetwork that – when trained in isolation with a decent learning rate – can reach similar accuracy with the well-trained original network using the same or fewer iterations, while showing clear advantage in accuracy compared to a randomly reinitialized subnetwork as well as an equivalently parameterized small-dense network.*

The principles for the identification of the winning tickets. From our preliminary results in Figure 1, we recognize that the pretraining of the randomly initialized dense network $f(x; \theta_0)$ with different initial learning rates achieves varying accuracy. Based on this observation and the rigorous definition of lottery ticket hypothesis, we list the conditions for identifying winning ticket as follows:

① A non-trivial sparsity ratio s and a sufficient training epochs T are adopted for the subnetwork.

② SDT of $f(x; \theta_T^{SD})$ shows clear accuracy drop compared to the well-trained subnetwork.

③ There exists a learning rate such that the subnetwork $f(x; (\theta_0 \odot m)_T)$ achieves notably higher accuracy (with a clear gap) than $f(x; (\theta_0' \odot m)_T)$ trained with any learning rates.

④ There exists a learning rate such that the subnetwork $f(x; (\theta_0 \odot m)_T)$ achieves accuracy similar to or higher than the pretrained network $f(x; \theta_T)$ at the same learning rate.

⑤ There exists a learning rate such that the subnetwork $f(x; (\theta_0 \odot m)_T)$ achieves accuracy similar to or higher than the *well-trained* original network $f(x; \theta_T)$ (i.e., trained with an appropriate learning rate and sufficient number of training epochs).

Our listed conditions *complete* the long missing but necessary aspects for identifying the winning ticket. ① formally recognizes the practical significance of the winning tickets, that a found network topology of the winning ticket should benefit the training/inference speed. It is commonly acknowledged that the overall sparsity ratio of the non-structured sparsity should exceed approximately 60% to deliver on-device acceleration. ② avoids a situation where the accuracy of the winning ticket is comparable to that of a small-dense network due to the over-parameterization of a network, which

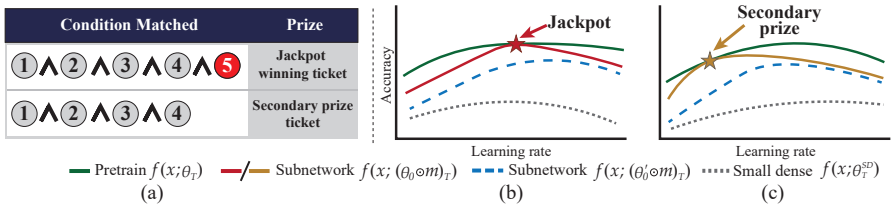


Figure 2: An illustration of the principles for identification of the winning tickets.

ensures the necessity of the winning ticket existence. ③ takes into account of the influences by different learning rates, which is missing in previous discussions. ④ is the original condition for identifying winning ticket in previous works, but it does not consider the best pretraining accuracy at a desirable learning rate. ⑤ takes the desirable training recipe into consideration, which is different from existing works and becomes the most crucial condition in our definition. We define "similar accuracy" as within 0.5% accuracy drop for CIFAR-10, 1% for CIFAR-100 and Tiny-ImageNet, and 1.5% for ImageNet-1K, and a "clear gap" between $f(x;(\theta_0 \odot m)_T)$ and $f(x;(\theta_0' \odot m)_T)$ (condition ③) should be an accuracy difference over 0.5%.[1]

We summarize the principles for identifying the winning tickets in Figure 2 (a).

- In the case that a subnetwork $f(x;(\theta_0 \odot m)_T)$ satisfies the condition ① - ⑤ as Figure 2 (b) shows, we call $(\theta_0 \odot m)$ as *Jackpot winning ticket*, for it has the potential to completely match the best performance of the original dense network.
- On the other hand, the original "winning ticket" discussed in [1] achieves the pretraining accuracy that is clearly lower than the best pretraining accuracy as Figure 2 (c). In this case, condition ① - ④ are satisfied while the condition ⑤ is not, and we consider it as a *secondary prize ticket*.

We distinguish our definition of the lottery ticket hypothesis from the weight rewinding technique [5]. Lottery ticket hypothesis, on one hand, is a study of initialization state and network topology for a neural network, while weight rewinding, on the other hand, studies the trade-off between accuracy and subnetwork searching cost. Despite the difference, we can generalize the weight rewinding technique into the winning ticket identification principle, which is shown in Appendix A. Detailed experimental evaluations of weight rewinding can also be found in Appendix B.

3 Sanity Checks for Lottery Tickets: Evaluation, Analysis and Guideline

Based on the rigorous definition of the lottery ticket hypothesis, we evaluate the lottery tickets with different types of network architectures, datasets with different sizes, and different learning rates. Detailed analysis are demonstrated for a deeper understanding of the lottery ticket hypothesis.

3.1 A Comprehensive Study Under the Rigorous Definition

Networks and datasets: In this section, we evaluate the lottery ticket hypothesis with various combinations of networks and datasets. We choose different network architectures among ResNet series [10], VGG [11], and MobileNet-v1 [12]. Specifically, the ResNet-32 is a wide version [13] with a width multiplier of 2. CIFAR-10/100 [14], Tiny-ImageNet [15] and ImageNet-1K [16] are all evaluated. Table 2 lists the details of the networks and datasets in the experiments we perform.

Table 2: Dataset and network we evaluate using the re-definition of the lottery ticket hypothesis.

Dataset	CIFAR-10			CIFAR-100		Tiny-ImageNet		ImageNet-1K	
#Images	50K/10K			50K/10K		100K/10K		1.28M/50K	
#Classes	10			100		200		1000	
Img Size	32×32			32×32		64×64		224×224	
Network	RN-20	RN-32	MBNet-v1	RN-18	VGG-16	RN-18	RN-50	RN-18	RN-50
#Params.	0.27M	1.86M	3.21M	11.22M	14.72M	11.68M	25.56M	11.69M	25.56M

Experimental setups: In this paper, we conduct our experiments using different learning rates. We empirically set the (initial) learning rate from extremely small to normal, then to very large based on the network and dataset. At each learning rate, we conduct a series of experiments described

[1] Our quantitative criteria for accuracy gaps are no different from many previous efforts.

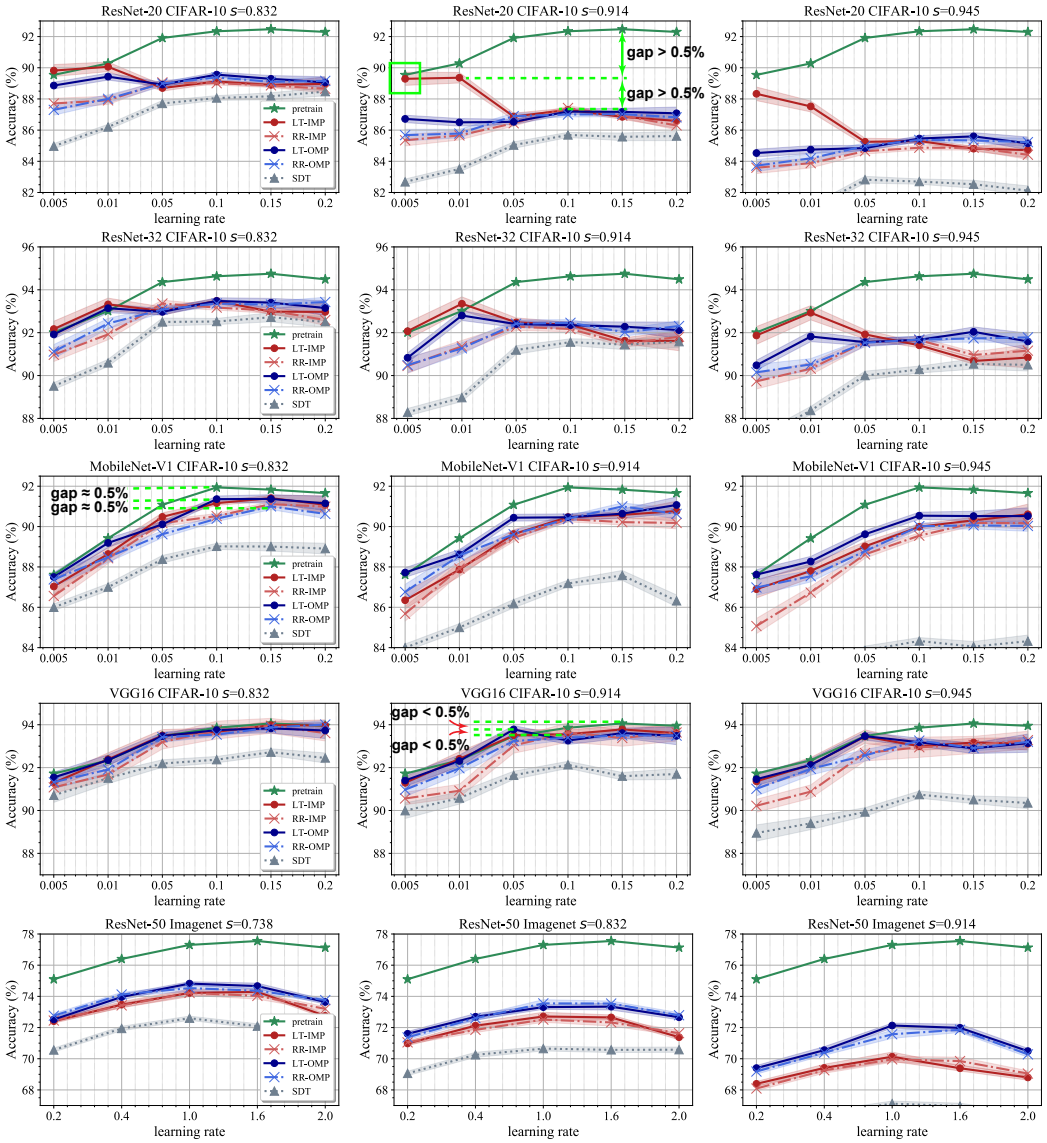


Figure 3: Lottery ticket experiments with different networks, datasets and (initial) learning rates. CIFAR-10 results are ordered by network size. ResNet-50 results on ImageNet-1K are also included.

in Section 2.1, and each experiment is run *three* times. For $\texttt{IMP}(\cdot)$, we follow the settings in [1, 5] that 20% of the weights are pruned in each iteration. For $\texttt{OMP}(\cdot)$, we directly prune the network to the same sparsity ratio as $\texttt{IMP}(\cdot)$. On CIFAR-10/100, We train the network for 160 epochs and the learning rates decrease by a factor of 10 after 80 and 120 epochs. On ImageNet-1K, We train the network for 90 epochs and cosine annealing learning rate schedule is used. We conduct our experiments on NVIDIA A100 with 8 GPUs. Detailed experiment settings are listed in Appendix C.

We plot the accuracy vs. learning rate curves for all experiments we run, and demonstrate them in Figure 3. Due to the space limits, we put the full results for all other networks, datasets and sparsity ratios in Appendix D.1. Based on the results, we summarize the observations in Table 3 and answer the following questions with detailed analysis. For the following discussion, if not otherwise specified, we use LT to denote the setting of the subnetwork training with LT-`IMP` or LT-`OMP`, and RR for RR-`IMP` or RR-`OMP`.

Do Jackpot winning tickets exist in our evaluation?

We carefully examine all the results. Unfortunately, under the rigorous definition of the lottery ticket hypothesis and current ticket searching methods ($\texttt{IMP}(\cdot)$ and $\texttt{OMP}(\cdot)$), no clear Jackpot winning tickets

Table 3: Summary of the observations of all experiments.

Dataset	RN20	RN32	MBNet-v1	RN18	VGG-16	Dataset	RN18	RN50
CIFAR-10	✗ ✓ ✓ ✓	✗ ✓ ✓ ✓	△ ✓ ✗ ✗	✗ ✓ ✓ ✗	✗ ✗ ✗ ✗	Tiny-ImageNet	✗ ✗ ✗ ✗	✗ ✗ ✗ ✗
CIFAR-100	✗ ✓ ✓ ✓	✗ ✓ ✓ ✓	✗ ✗ ✗ ✗	△ ✓ ✓ ✓	✗ ✗ ✗ ✗	ImageNet-1K	✗ ✗ ✗ ✗	✗ ✗ ✗ ✗

Jackpot Secondary Prefer small lr Prefer IMP ✓ Yes ✗ No △ At boundary

are found, and even tickets that merely reach the boundary of conditions rarely exist. According to the experiments and the preliminary results in Section 2.1, we do notice **an accuracy improvement** for both pretraining and subnetwork training with a sufficient training recipe. However, the accuracy gap between pretrained network and subnetwork is still **non-negligible**. For instance, consider the case using ResNet-20 on CIFAR-10 at $s = 0.914$ in Figure 3, the Jackpot winning ticket is not identified, because the highest accuracy of the subnetwork by LT-`IMP` has a noticeable gap ($> 0.5\%$) compared to the highest pretraining accuracy. Take VGG-16 on CIFAR-10 at $s = 0.914$ as another example, although the subnetwork achieves similar accuracy with pretraining, there is no accuracy gap ($< 0.5\%$) between LT and RR, thus no tickets are found.

Recall the principles for identifying the winning ticket, all the cases are verified at the best suited learning rate, and please note that if there exists any non-trivial sparsity ratio (please check Appendix D.1 for results at all sparsity ratios) that makes the subnetwork meet the conditions, we call the Jackpot winning ticket exist for this network. Under the rigorous definition, the odds for getting a Jackpot winning ticket is low, but we believe the Jackpot winning ticket is likely to be existing in a network with an appropriate size and trained using a desirable learning rate (please check Appendix D.2 for more details). For instance, in Figure 3, the case of MobileNet-v1 on CIFAR-10 at $s = 0.832$ reaches the boundary of Jackpot winning ticket conditions, as the accuracy gaps between LT and pretraining, and between RR and LT are both around 0.5%.

Do secondary prize tickets exist in our evaluation?

Yes. secondary prize tickets exist in most of the networks on small datasets, and please note that the "winning tickets" found in previous works are (at most) similar to the secondary prize tickets based on our definition. Again, we use ResNet-20 at $s = 0.914$ as an example. In Figure 3, secondary prize ticket exists in the green box, because the LT accuracy is similar with the pretraining accuracy at the same learning rate (0.005), while an accuracy gap ($> 0.5\%$) between LT and RR exists. However, the capacity of the network (in our cases, the number of weights in a network) determines the maximum sparsity at which a secondary prize ticket can be found. For instance, a relatively small network ResNet-20 can identify the secondary prize ticket at a maximum sparsity ratio of 0.914 on CIFAR-10, while larger networks such as ResNet-32, ResNet-18 and VGG-16 can identify secondary prize tickets on sparsity ratio of 0.945 or higher (refer to Appendix D.1). But on a medium and large-scale dataset as Tiny-ImageNet and ImageNet-1K, no clear secondary prize tickets are identified using ResNet-18 or ResNet-50. We believe a network with higher capacity may be able to identify one on ImageNet-1K.

When does θ_0 benefit subnetwork training?

We find that the secondary prize tickets are more likely to be found at a relatively small learning rate. To analyze the reason, we use a *correlation indicator* $R_p(\theta, \theta')$ to quantify the number of overlapped indices of the top-$p \cdot 100\%$ large-magnitude weights between two different sets of weights. We say the correlation between θ and θ' is weak if $R_p(\theta, \theta') \approx p$, and when $R_p(\theta, \theta') > p$, the correlation is positive. The detailed definition and explanation of the correlation indicator is shown in Appendix E. We evaluate the correlations between $(\theta_0 \odot m)$ and $(\theta_T \odot m)$, and between $(\theta_0' \odot m)$ and $(\theta_T \odot m)$ regarding different learning rates on ResNet-20 and VGG-16 as Figure 4 shows. When using a relatively small learning rate, we find that the accuracy of $f(x; (\theta_0 \odot m)_T)$ is closer to pretraining accuracy than $f(x; (\theta_0' \odot m)_T)$ does. In this case, the correlation between $(\theta_0 \odot m)$ and $(\theta_T \odot m)$ is positive while $(\theta_0' \odot m)$ and $(\theta_T \odot m)$ is weak. When the correlation between $(\theta_0 \odot m)$ and $(\theta_T \odot m)$ is positive, the weights that are large in magnitude in pretraining network are likely to also be large in a trained

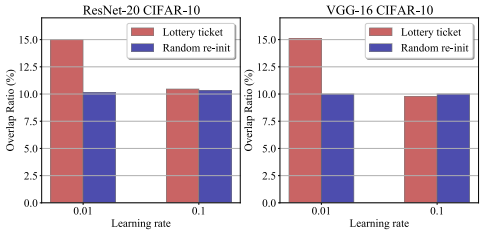
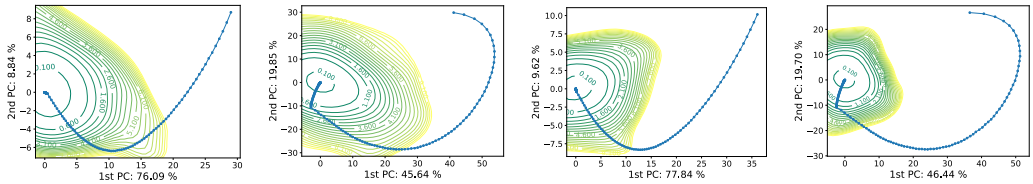
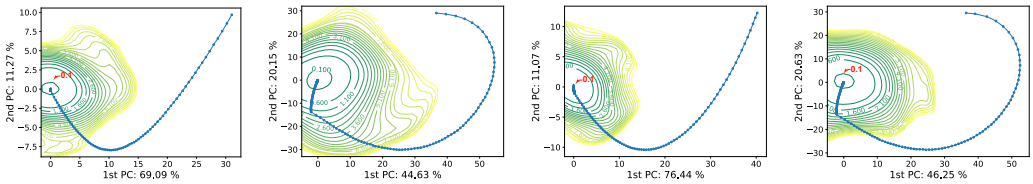


Figure 4: Correlation between weights in subnetwork and pretrained network with different learning rates. The subnetwork we use has $s = 0.832$ and we set $p = 0.1$.



(a) `IMP`, lr=0.01, Acc=92.9% (b) `IMP`, lr=0.1, Acc=91.4% (c) `OMP`, lr=0.01, Acc=91.8% (d) `OMP`, lr=0.1, Acc=91.6%

Figure 5: Training trajectories along the loss surface contours of ResNet-32 on CIFAR-10 at sparsity ratio of 0.945.



(a) `IMP`, lr=0.01, Acc=89.4% (b) `IMP`, lr=0.1, Acc=89.6% (c) `OMP`, lr=0.01, Acc=89.7% (d) `OMP`, lr=0.1, Acc=90.7%

Figure 6: Training trajectories along the loss contours of ResNet-32-like network without residual connections on CIFAR-10 at sparsity ratio of 0.945.

subnetwork, thus a relatively close accuracy is observed. When the correlation does not exist, using θ_0 or θ_0' in the subnetwork makes no difference to the final accuracy.

Which pruning method is better, `IMP`, `OMP` , or it does not matter?

Comparing the results regarding network structures, we find that when residual connections exist in the network, `IMP` is more preferable than `OMP`, and when there are no residual connections the `IMP` has no advantages over `OMP`. To further investigate it with "apple-to-apple" comparison, we construct a "ResNet-32-like" network, by removing all residual connections from ResNet-32 while leaving all else intact. We then evaluate the accuracy of `IMP` and `OMP` on ResNet-32, versus the newly constructed ResNet-32-like network. We also visualize both optimization trajectories along the contours of the loss surface, using the classical method in [17, 18].

According to Figure 5 that the residual connections exist in ResNet-32, a subnetwork using `IMP` explores a much smoother route than using `OMP` as its contour is smoother and close-to-convex (a larger landscape area with mild variance, and a larger basin in the middle of it [18]), which indicates that the optimization route may be smooth towards local minima.

When there are no residual connection as Figure 6 shows, however, we do not see much difference between `IMP` and `OMP`. Compare to the `IMP` method in Figure 5, the advantages of the `IMP` to `OMP` is diminished. Note that the landscape will become much more rugged if residual connections are removed from a network [18]. We conjecture that in our constructed no-residual ResNet-32, the optimization becomes too difficult and neither `IMP` nor `OMP` is effective enough to explore a smooth route towards local minima: hence no much difference observed between them.

What learning rate is more likely to help identifying the winning tickets?

We notice that when residual connections exist, the subnetwork achieves higher accuracy at a relatively small learning rate, while a larger learning rate is more preferable in training of a subnetwork without residual connections. In Figure 5, as the residual connection makes the landscape become much smoother [18], we can see a subnetwork trained with a small initial learning rate 0.01 achieves a larger contour and a larger basin in the middle, while the contour and basin area with large learning rate 0.1 are relatively small. We conjecture that the optimization is much easier for a smaller initial learning rate on a smooth loss surface, leading to a better network performance. Without residual connections (as Figure 6), the above observations are exactly the opposite. Note that the no-residual ResNet-32 creates a more rugged landscape, thus a small initial learning rate 0.01 is more likely to stuck in a sub-optimal local minima, while a large initial learning rate is unlikely to, therefore the SGD process is more likely to find a desired path to high quality solutions.

Does the size of the dataset have different patterns compared to small dataset?

We find the patterns for the identified winning tickets are different on a relatively large-scale dataset, such as Tiny-ImageNet and ImageNet-1K. For all the ResNet architectures we evaluate, `OMP` outper-

forms IMP, and small learning rates are not preferable in training a subnetwork. We provide more discussion in Appendix D.3.

Does weight rewinding improve the accuracy?

We find the weight rewinding technique [5] consistently improves the subnetwork accuracy. We generalize the weight rewinding technique into the winning ticket identification principles, and perform a series of experiments. Due to space limits, the results are discussed in Appendix B.

3.2 How to Quickly Win a Prize in a Lottery Game – A Guideline

In this section, we summarize the patterns we find through the extensive experimental results, and present in the form of a guideline to help quickly identify the *Jackpot winning ticket* and *secondary prize ticket* (both referred as *ticket* below for simplicity). Our guideline is presented as follows:

1. On a small dataset using networks with residual connections, IMP is better than OMP. When the network has no residual connections, IMP has no advantages over OMP.
2. On a small dataset using networks with residual connections, the subnetwork prefers a relatively small learning rate to find the tickets. When the network has no residual connections, small learning rate is not preferable.
3. When the network is redundant (e.g., a large network on a small-scale dataset), the maximum sparsity that a ticket can be found is relatively high, and vice versa.
4. When the (sub)network prefers large learning rates, using different initialization yields the similar accuracy in subnetwork training.

4 Related Works

Lottery Ticket Hypothesis. The lottery ticket hypothesis and the definition of the "winning ticket " are firstly proposed in [1]. Concurrent work [7] finds that the identical initialized weights will not provide any advantage over training with randomly initialized weights at relatively large learning rates. [7, 19, 20] also confirm that the matching subnetworks at nontrivial sparsity are hard to find in more challenging tasks. The following work [5] extends the subnetwork training from initial weights to the weights at early stage of pretraining (rewinding), and improve the accuracy in more challenging tasks at nontrivial sparsity.

Besides computer vision tasks, the lottery ticket hypothesis is also investigated in many other tasks. [6, 21] further extend the lottery ticket hypothesis to a pre-trained BERT model. On object detection task, [22] proposes a guidance to find task-specific winning tickets for object detection, instance segmentation, and keypoint estimation. [23, 24] have studied the lottery ticket hypothesis in unsupervised learning to reveal how well the tickets are transformed between different datasets.

Find Winning Ticket at Early Stage of Training. The potential of training a sparse network from initialization suggested by the lottery ticket hypothesis has motivated the study of deriving the "winning tickets" at an early stage of training, thereby accelerating training process. There is a number of work in this direction. [25] conducts a retraining process after searching sub-network topology for a few epochs. [26] examines the network state during early iterations of training, and analyzes the weight distribution and its reliance on the dataset. SNIP [27] finds the sparse mask based on the saliency score of each weight that is obtained after training the dense model for only a few iterations. GraSP [28] prunes weights based on preserving the gradient flow in the network.

5 Conclusion and Discussion of Broader Impact

In this paper, we investigate the underlying condition and rationale behind the lottery ticket hypothesis. By revisiting the original definition, we find out that the current controversies over this topic is largely related to the quality of the training recipe. We propose a rigorous definition of the lottery ticket hypothesis, as well as the principles for identifying the true "Jackpot winning ticket" or "secondary prize ticket". We perform sanity checks for the lottery tickets through extensive experiments over multiple deep models on different datasets, and empirically study the patterns we observe by quantitative analysis. Meanwhile, we develop a guideline based on our summarized patterns, which potentially facilitates the research process on the topic of the lottery ticket hypothesis. The research is scientific in nature and we do not envision it to generate any negative societal impact.

References

[1] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *ICLR*, 2018.

[2] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 67–82, 2018.

[3] Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing lottery tickets: Zeros, signs, and the supermask. 2020.

[4] Haoran You, Chaojian Li, Pengfei Xu, Yonggan Fu, Yue Wang, Xiaohan Chen, Richard G Baraniuk, Zhangyang Wang, and Yingyan Lin. Drawing early-bird tickets: Towards more efficient training of deep networks. *arXiv preprint arXiv:1909.11957*, 2019.

[5] Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing rewinding and fine-tuning in neural network pruning. *ICLR*, 2020.

[6] Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. The lottery ticket hypothesis for pre-trained bert networks. 2020.

[7] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*, 2018.

[8] Yann LeCun, John S Denker, and Sara A Solla. Optimal brain damage. In *Advances in neural information processing systems*, pages 598–605, 1990.

[9] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations (ICLR)*, 2016.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[12] Andrew Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv:1704.04861*, 2017.

[13] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[14] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[15] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7:7, 2015.

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[17] Eliana Lorch. Visualizing deep network training trajectories with pca. 2016.

[18] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. 2018.

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

[20] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pages 3259–3269. PMLR, 2020.

[21] Sai Prasanna, Anna Rogers, and Anna Rumshisky. When bert plays the lottery, all tickets are winning. *EMNLP*, 2020.

[22] Sharath Girish, Shishira R Maiya, Kamal Gupta, Hao Chen, Larry Davis, and Abhinav Shrivastava. The lottery ticket hypothesis for object recognition. *CVPR*, 2021.

[23] Ari S Morcos, Haonan Yu, Michela Paganini, and Yuandong Tian. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers. pages 4932–4942, 2019.

[24] Rahul Mehta. Sparse transfer learning via winning lottery tickets. 2020.

[25] Haoran You, Chaojian Li, Pengfei Xu, Yonggan Fu, Yue Wang, Xiaohan Chen, Richard G Baraniuk, Zhangyang Wang, and Yingyan Lin. Drawing early-bird tickets: Towards more efficient training of deep networks. *ICLR*, 2020.

[26] Jonathan Frankle, David J Schwab, and Ari S Morcos. The early phase of neural network training. *ICLR*, 2020.

[27] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. Snip: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations (ICLR)*, 2019.

[28] Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations (ICLR)*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please see Section 3.1 and Appendix C in Supplemental material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report the details of the error bar settings in Section 3.1. The main results in Figure 3 also includes variation data.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Please check Section 3.1
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Source code will be provided, and we also properly cite the creators. Please refer to Appendix C
   (b) Did you mention the license of the assets? [N/A] We cite the orginial dataset and models.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We will provide all of our source code to reproduce the results.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] The datasets we use in this paper is public, and we cite the data providers.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We use public dataset.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]