Analyzing Lottery Ticket Hypothesis from PAC-Bayesian Theory Perspective

Keitaro Sakamoto
The University of Tokyo
sakakei-1999@g.ecc.u-tokyo.ac.jp

Issei Sato
The University of Tokyo
sato@g.ecc.u-tokyo.ac.jp

Abstract

The lottery ticket hypothesis (LTH) has attracted attention because it can explain why over-parameterized models often show high generalization ability. It is known that when we use iterative magnitude pruning (IMP), which is an algorithm to find sparse networks with high generalization ability that can be trained from the initial weights independently, called *winning tickets*, the initial large learning rate does not work well in deep neural networks such as ResNet. However, since the initial large learning rate generally helps the optimizer to converge to flatter minima, we hypothesize that the winning tickets have relatively sharp minima, which is considered a disadvantage in terms of generalization ability. In this paper, we confirm this hypothesis and show that the PAC-Bayesian theory can provide an explicit understanding of the relationship between LTH and generalization behavior. On the basis of our experimental findings that IMP with a small learning rate finds relatively sharp minima and that the distance from the initial weights is deeply involved in winning tickets, we offer the PAC-Bayes bound using a spike-and-slab distribution to analyze winning tickets. Finally, we revisit existing algorithms for finding winning tickets from a PAC-Bayesian perspective and provide new insights into these methods.

1 Introduction

The high generalization ability of modern neural networks can be attributed to the heavier overparameterization and effective learning algorithms [22, 30, 41]. This increase in the number of parameters leads to high computational cost and high memory usage, and network pruning is one of the effective techniques for addressing these problems [12, 14, 23]. After pruning a significant number of parameters, the pruned network can often work well with little or no accuracy loss. However, training this sparse subnetwork independently from the initial weights often does not work, and we can only obtain the sparse subnetwork through pruning after training the whole network.

Frankle and Carbin [10] presented "Lottery Ticket Hypothesis (LTH)" that states the existence of winning tickets: small but critical subnetworks which can be trained independently from scratch. They proposed an algorithm called iterative magnitude pruning (IMP) to obtain a winning ticket. They pointed out that, for deeper networks such as VGG [35] and ResNet [16], small learning rate is required to obtain a winning ticket. However, since a large learning rate helps the generalization ability of neural networks [25], the learning rate should be well controlled to find a winning ticket that has a higher test accuracy. Frankle et al. [11] found a correlation between the stability to SGD noise and the ability to find the winning ticket and empirically showed that the large learning rate moves weights too much under the low-stability learning process.

In this paper, we first empirically show that winning tickets are actually more vulnerable to label noise setting compared to the subnetwork created with the large learning rate; that is, the generalization ability of winning tickets is degraded due to the learning rate constraint. In this connection, we then

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

focus on the two concepts flatness and the distance from the initial weights of the winning tickets. We next apply the PAC-Bayesian theory to LTH on the basis of the flatness motivation and show that it can explain the relationship between LTH and generalization behavior. We use the PAC-Bayes bound for a spike-and-slab distribution to analyze winning tickets, which is based on our experimental findings that reducing the expected sharpness restricted to an unpruned parameter space and adding the regularization of distance from the initial weights can enhance the test performance of winning tickets. Finally, we revisit existing algorithms such as IMP, continuous sparsification [34] from the point of view of the PAC-Bayes bound optimization. This consideration gives an interpretation of these methods as an approximation of bound optimization.

To sum up, our contributions are as follows.

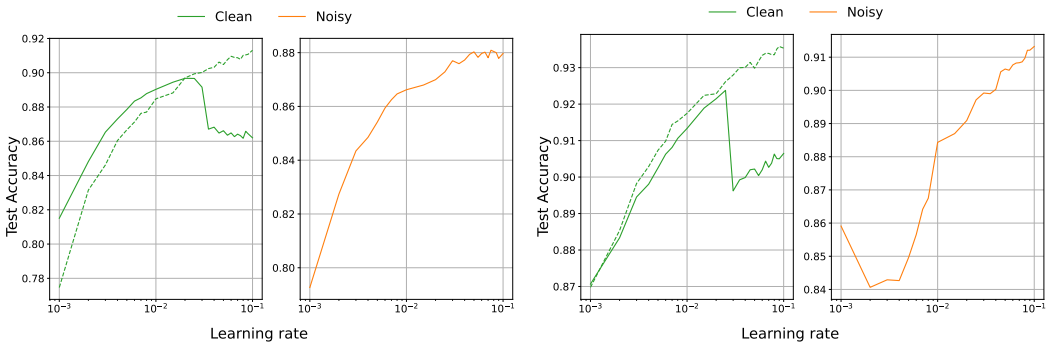
- We experimentally show that IMP with a small learning rate finds relatively sharper minima and that the distance from the initial weights is critical for IMP, i.e., balancing the distance and the training error helps to find them.
- On the basis of our findings, we reveal that the PAC-Bayesian formulation for a spike-and-slab distribution effectively captures the winning tickets behavior.
- We revisit the existing algorithms from the PAC-Bayesian perspective and explain their behavior.

2 Related Work

Learning Rate The initial large learning rate often improves generalization of deep neural networks [19, 25]. The large learning rate generally helps an optimizer to converge to flatter minima [24, 38], and these flatter minima are advantageous for generalization ability rather than sharper minima [18, 20, 21]. The relationship between flatness and generalization can be considered from a PAC-Bayesian perspective [5, 31, 32] (see Appendix A). Frankle et al. [11] provided the insights in why IMP with the large learning rate fails into find winning tickets in some problem settings. They proposed IMP with rewinding to an early epoch to avoid the very early training process because training is not stable to SGD noise in the early training regime (see Appendix A).

Empirical Results on LTH Some studies have also investigated the flatness and the distance from the initial weights on LTH. Bain [2] plotted the loss landscape of winning tickets visually and found that IMP produces more convex and sharper minimum relative to random pruning. Bartoldson et al. [3] found that pruning regularizes similarly to noise injection, and they discussed the generalization of pruned networks considering flatness. They measured flatness by using the trace of Hessian and gradient covariance matrix. He et al. [17] refers the distance from the initial weights to analyze the label noise robustness of winning tickets; they stated that the influence of the label noise is lessened if the distance is suppressed. Liu et al. [26] discussed the relationship between winning tickets and the learning rate considering the similarity between initial and trained weights in terms of pruning mask overlap instead of the distance from the initial weights. There is a recent study on finding a mask that shows good accuracy without training weights at all [33, 43].

Theoretical Results on LTH Malach et al. [29] demonstrated a subnetwork with a comparable accuracy to the original network in a sufficiently over-parameterized network, without any training. Zhang et al. [42] analyzed the winning tickets generalization based on sample complexity. Although it is limited to the case of a one-hidden-layer neural network, they provided insight into why the sparsity increases the generalization ability. For a PAC-Bayesian theory, Hayou et al. [15] also used spike-and-slab prior and posterior distributions. However, their motivation and purpose are completely different from our work. Their aim was to obtain a sophisticated pruning mask by optimizing the PAC-Bayes bound, and this is mainly in the context of network pruning rather than LTH. Our work differs in that we use it to analyze the generalization behavior of a given winning ticket on the basis of our empirical findings on the learning rate, the flatness, and the distance from the initial weights. In addition to these, we also discuss the relationship with existing algorithms.



(a) ResNet20 (b) VGG16

Figure 1: Test accuracy on CIFAR10 whose labels are randomly flipped (clean: green line, noisy: orange line) when ResNet20 (90% sparse) and VGG16 (99% sparse) are trained. These subnetworks are produced by IMP with various learning rates. The dashed green line shows the unpruned baseline with the same learning rate. In the label noise setting, the model has not yet converged at the same number of epochs as the clean setting; therefore we train them until convergence, and training epoch is increased from the setting of Frankle and Carbin [10].

3 Empirical Analysis

We first show some empirical results because our new findings about winning tickets in this empirical analysis motivate the PAC-Bayesian analysis for LTH; thus, we interpret the results on the basis of the PAC-Bayesian perspective in the next section.

We empirically investigated the properties of winning tickets mainly related to the learning rate. The small learning rate is good for finding the winning ticket, which is contrary to the intuition that a large learning rate is good in terms of generalization ability. First, we show the small learning rate is actually disadvantageous for generalization ability of subnetworks through experiments under label noise. Next, as a reason for this, we show that the small learning rate finds the relatively sharper minima, and it is possible to find the winning tickets in the flatter minima using sharpness regularization. We also focused on the distance from the initial weights as a reason for why the small learning rate is important for IMP. On the basis of these findings, we will develop a discussion from a PAC-Bayesian perspective in the next section. We will show this perspective will theoretically support the findings in this section. We followed the experimental setting of Frankle and Carbin [10] and used a modified version of OpenLTH repository [9].

3.1 Vulnerability to label noise

We examined the test accuracy when some fractions of the labels in the training set are randomly flipped to see whether the generalization behavior of winning tickets is degraded by the constraint that the learning rate must be small. Figure 1 shows the test accuracy on clean and label noise datasets of sparse subnetworks produced by IMP with different learning rates. As for the no label noise setting (green line), there is an accuracy drop at some point as the learning rate is increased. We added the original unpruned baseline (dashed green line) to discuss if the subnetwork is a winning ticket, and this baseline shows no such accuracy drop when increasing the learning rate. The subnetwork eventually performs worse than the original unpruned network, which means that IMP ultimately fails to find the winning ticket. In contrast, the test accuracy generally continues to improve without decreasing as the learning rate is increased in the high label noise setting (orange line). The test accuracy increases even when it is not a winning ticket for no label setting at the same learning rate. To sum up, the large learning rate is not suitable for finding winning tickets under a clean dataset but is advantageous in the high label noise setting.

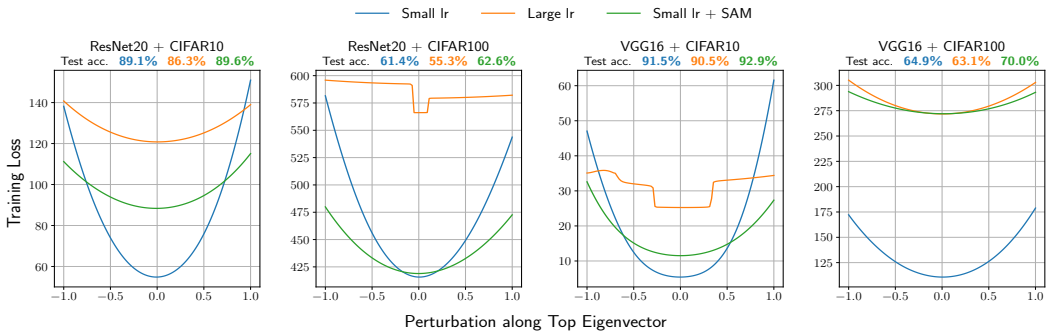


Figure 2: 1-d training loss landscape along eigenvector corresponding to largest eigenvalue of Hessian around trained parameters. The test accuracies in the caption correspond to the parameter at the center of perturbation (0.0 on the horizontal axis) and show accuracy for the small learning rate (blue), the large learning rate (orange), and the small learning rate + SAM (green) form left to right. The subnetwork of ResNet20 has 90% sparsity, and that of VGG16 has 99% sparsity. The top eigenvectors are calculated by using PyHessian [40].

3.2 Flatness

Now that we have seen that winning tickets have undesirable properties due to the small learning rate, we investigated whether this result comes from the difference in flatness around the found solution. There are many previous studies that discuss the relationship between the learning rate and flatness [24, 38]. In this experiment, perturbations are added only to unpruned weights to consider flatness on pruned networks. We will discuss this justification in detail in Section 4.

First, we visualized the loss landscape shape of subnetworks produced by IMP. Figure 2 shows the 1-d loss landscape of the subnetwork that has a high sparsity. This landscape shows the training loss around the trained weights adding perturbation restricted to unpruned parameters. We can see that the parameters of the winning ticket (small learning rate) are in the sharper minimum compared with the large learning rate. The large learning rate can find a flatter minimum; however, the training loss is higher and the test accuracy is worse than the small learning rate. This graph is a fixed sparsity loss landscape; therefore, it is not possible to discuss whether or not the subnetwork is a winning ticket from this graph. There is actually an accuracy drop in the large learning rate setting when sparsity is changed, which means this is not a winning ticket.

Given that IMP requires a small learning rate rather than a large one, these results suggest two possible interpretations; 1) sharp minimum is essential to find winning tickets, therefore the large learning rate fails in IMP, 2) sharp minimum is simply the result of small learning rate training, and flat minimum is better for winning tickets if possible. To investigate these possibilities, we used sharpness-aware minimization (SAM) [8] and neural variable risk minimization (NVRM) [37] to search for parameters that lie in neighborhoods having uniformly low loss. They differ in that SAM minimizes the maximum loss in the neighborhood, whereas NVRM minimizes the expected loss in the neighborhood. We used both of them to compare with normal SGD as a baseline. These optimizers are based on SGD with the same setting as the original LTH paper [10], and the noise considered in these methods is limited to the unpruned parameters. Figure 2 also shows the loss landscape when SAM is used; It can actually reach a flatter minimum. The loss landscape is as flat as the large learning rate setting; however, unlike this, the test accuracy is higher than that of the small learning rate.

Next, we investigated the trace of Hessian to analyze the flatness of winning tickets created by these optimizers. The trace of Hessian is used as a measure of flatness in the prior work [4, 21, 39]. Table 1 shows that SGD with the small learning rate finds relatively a sharper minimum and that IMP can find a flatter one by using SAM or NVRM. These optimizers can find relatively flatter minima, which improves the test accuracy to some extent compared with the SGD with the small learning rate (see Appendix B.1).

We also found that the large learning rate still cannot find winning tickets even though we use SAM instead of SGD (see Appendix B.2). The large learning rate has already found relatively flatter

Table 1: Trace of Hessian for ResNet20 and VGG16 trained on CIFAR10 and CIFAR100. We used three optimizers, SAM, NVRM, and SGD. The hyperparameter of SAM ρ and NVRM b are chosen from $\{0.05, 0.1, 0.2, 0.5\}$ and $\{0.014, 0.018, 0.022, 0.026\}$ respectively with the highest test accuracy. As a baseline, we show the results of SGD with a small learning rate, and the learning rate is also set to small for SAM and NVRM. The trace of Hessian is calculated by PyHessian [40]. We averaged over three different subnetworks.

		ResNet20			VGG16		
Dataset	Sparsity (%)	SGD	SAM	NVRM	SGD	SAM	NVRM
CIFAR10	90	1556	365	1238	107	67	70
	95	1786	791	1218	171	79	62
CIFAR100	90	4010	2071	3190	231	119	159
	95	6340	758	2997	411	226	190

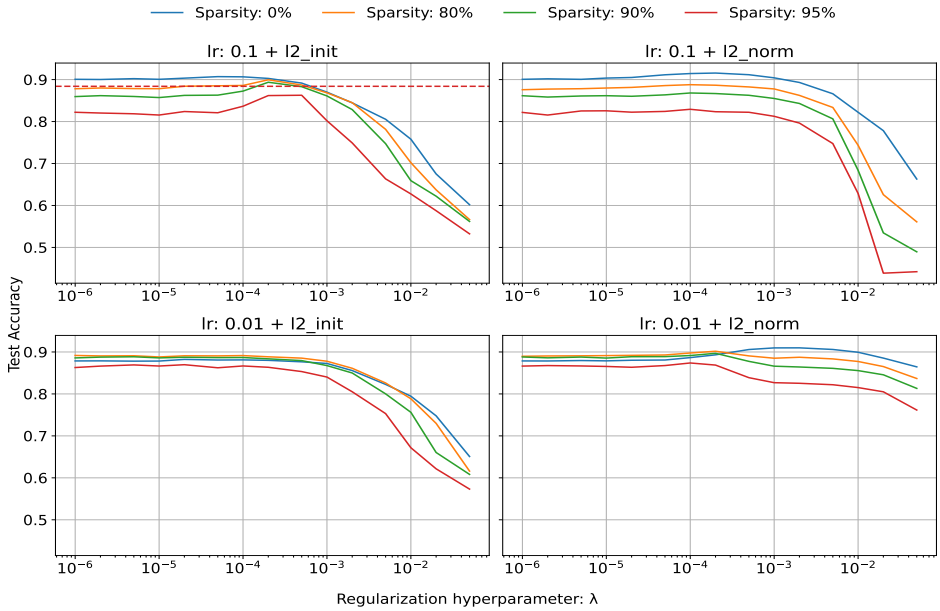


Figure 3: Test accuracy of ResNet20 trained on CIFAR10 when regularization term is added. Top row shows the results of the large learning rate (0.1) and bottom row shows those of the small learning rate (0.01). Left: $l2\_init$, Right: $l2\_norm$. The dashed red line in the top left shows the unpruned baseline with the small learning rate (0.01).

minima as shown in Figure 2; therefore, it is considered that the results do not change by using SAM. This fact implies that the flatness around the parameters found by SGD with a large learning rate and by SAM with a small learning rate are similar, however winning tickets cannot be found with the large learning rate because they find different solutions in terms of some other properties. Next, we analyze this difference by focusing on the distance from the initial weights.

3.3 Distance from initial weights

As a reason for the small learning rate constraint, we hypothesized that winning tickets can only be found by IMP within a range not far from the initial weights. In order to confirm this, we ran IMP suppressing this distance and compare it with the usual regularization by the L_2 norm regularization. In addition, we also discussed the pruning mask structure in Appendix B.3 in relation to the distance from the initial weights.

We empirically confirmed that the winning ticket can be obtained via the regularization of this distance even under the large learning rate setting. Let $\mathcal{L}_{\mathcal{S}}$ be the empirical risk on the training samples $\mathcal{S}$,

λ be a regularization hyperparameter, $\boldsymbol{\theta}$ be network weights, and $\boldsymbol{m} \in \{0,1\}^{|\boldsymbol{\theta}|}$ be a pruning mask. In this experiment, the parameters were regularized only for unpruned weights. Specifically, we designed the loss function with $l2\_init$ and $l2\_norm$, respectively, as follows.

$$\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}) + \lambda \|\boldsymbol{m} \odot (\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{init}})\|_2^2; \quad \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}) + \lambda \|\boldsymbol{m} \odot \boldsymbol{\theta}\|_2^2. \tag{1}$$

Figure 3 shows the test accuracy with different regularizations by changing the hyperparameter λ. We plot 80%, 90% and 95% sparsity subnetworks and the original unpruned network as a baseline. If the subnetwork accuracy is close to the whole network accuracy, it is considered to be successful in finding the winning ticket. As discussed previously, the large learning rate fails to find winning tickets unlike the small learning rate setting; however adding the regularization from the initial weights changes this trend. In Figure 3, λ around 2×10^{-4} shows that a winning ticket is found since the accuracy drop from the whole network is suddenly reduced, and there is no such a trend when L_2 norm regularization is added. This means that IMP can obtain winning tickets suppressing the distance from the initial weights even with the large learning rate. This finding is shown more clearly in Appendix B.4. We confirmed it by examining the test accuracy when sparsity is changed for problem settings other than ResNet20 + CIFAR10.

We can also obtain interesting results when the learning rate is small. The accuracy drop is small while λ is small; however it becomes large increasing λ. In the case of $l2\_init$, it is possible that the gap widens because sparse networks are more affected by strong regularization (it is the same for large learning rate setting), but in the case of $l2\_norm$, the gap widens significantly even though generalization ability increases because of the regularization. The strong norm regularization makes IMP fail to find a winning ticket even if a winning ticket could be found originally.

These results are related to the prior-mean selection from the PAC-Bayesian perspective. In the training of a normal network, there is a trade-off between suppressing the parameter norm, i.e., reducing the KL term, and reducing the training loss. Therefore, it is important to ensure a balance between them. As for IMP, there is a specific situation where suppressing the parameter norm makes the training loss large due to the failure to find a winning ticket. If we take the initial weights as a prior mean instead of 0, the training loss can decrease by suppressing the KL term when the trained weights are far from the initial weights. This experiment corresponds to what to take as a prior mean in terms of minimizing PAC-Bayes bound, and this result indicates that setting the initial weights to a prior mean seems to be compatible with minimizing the PAC-Bayes bound in the case of winning tickets.

4 PAC Bayesian Analysis on LTH

First, we present some possible definitions of subnetwork flatness and consider a PAC-Bayes bound of a spike-and-slab formulation based on our experiments. Next, we show that this bound captures the generalization behavior of winning tickets and revisit existing algorithms from the perspective of this bound.

4.1 Flatness on winning tickets

While we could simply consider the neighborhood of the trained weights in an unpruned neural network, it is not trivial to define the flatness of pruned networks depending on how the pruned weights are taken into consideration. The possible measure of the expected sharpness are as follows.

1. Add noise to the parameters restricted on unpruned parameter space.
2. Add noise to the parameters including the pruned weights.
3. Recover pruned weights and train the whole network to convergence (re-dense training [13]) and measure its flatness.

Measure 2 is the same as the unpruned neural network, but the sparse weights in the whole parameter space can no longer be in the local minimum; thus, it is uncertain whether flatness has any meaning in such a setting. He et al. [17] conducted re-dense training and showed that solutions at high sparsity are no longer minimizers in high dimensions. They also found that the winning ticket has higher sharpness than the original network based on Measure 3 and concluded that highly sparse solutions do not stick around the flat basins of minimizers. However, none of these metrics has any justification.

It is known that flatness can be viewed from a PAC-Bayesian perspective [5, 31, 32]. We use a PAC-Bayes bound of a spike-and-slab formulation, where expected sharpness corresponds to Measure 1. We will also compare this formulation with the normal Gaussian formulation (Measure 2) through numerical experiments and show that the spike-and-slab formulation captures the generalization behavior of winning tickets better. This supports our findings that using SAM and NVRM on pruned networks can enhance the generalization accuracy of winning tickets.

4.2 Spike-and-slab formulation

There are several problems when it comes to using the Gaussian distribution as prior and posterior for analyzing winning ticket properties. As discussed in Section 3.3, the distance from the initial weights of the winning ticket is expected to be small. We set the prior mean as the initial weights to take advantage of this property; however the original PAC Bayesian formulation based on the Gaussian distribution has the disadvantage that the pruned weights have weight 0 and the norm of the initial weights corresponding to these weights remain in the KL part. This not only results in a large bound but also behaves contrary to the purpose of getting sparse subnetworks when the bound is optimized. This is because the more sparse the subnetwork is, the larger the bound becomes. In addition, noise is inevitably added to the pruned weights when considering expected sharpness since the variance of the pruned part cannot be set to zero.

In order to limit the distance from the initial weights and noise added in expected sharpness only to the unpruned weights, we use a spike-and-slab distribution, which is the mixture of the Gaussian distribution $\mathcal{N}$ and Dirac delta distribution with a peak at zero-weight $\delta_{\{0\}}$, in the PAC-Bayes bound. Let $\boldsymbol{\sigma}_p$ and $\boldsymbol{\sigma}_q$ be vectors whose i^{th} element is a variance of Gaussian distribution, $\boldsymbol{\lambda}_p$ and $\boldsymbol{\lambda}_q$ be vectors that represents the mixture ratio of prior and posterior, respectively. $\boldsymbol{\theta}$ represents the network parameter, and $\boldsymbol{\theta}_{\text{init}}$ is the initial weights and $\bar{\boldsymbol{\theta}}$ is the trained weights.

We design the prior $\mathbb{P}$ and posterior $\mathbb{Q}$ as follows.

$$\begin{aligned}\mathbb{P}(\theta_i) &= (1-\lambda_{p,i})\delta_{\{0\}} + \lambda_{p,i}\mathcal{N}(\theta_i \mid \theta_{\text{init},i}, \sigma_{p,i}),\\ \mathbb{Q}(\theta_i) &= (1-\lambda_{q,i})\delta_{\{0\}} + \lambda_{q,i}\mathcal{N}(\theta_i \mid \bar{\theta}_i, \sigma_{q,i}).\end{aligned} \tag{2}$$

The KL divergence of the spike-and-slab distribution [36] can be calculated as

$$\begin{aligned}&\mathrm{KL}\left[\mathbb{Q}(\boldsymbol{\theta})\|\mathbb{P}(\boldsymbol{\theta})\right]\\ &= \sum_i \left(\lambda_{q,i}\left(\log\frac{\sigma_{p,i}}{\sigma_{q,i}} + \frac{\sigma_{q,i}^2 + (\bar{\theta}_i - \theta_{\text{init},i})^2}{2\sigma_{p,i}^2} - \frac{1}{2}\right) + \mathrm{kl}\left[\lambda_{q,i}\|\lambda_{p,i}\right]\right),\end{aligned} \tag{3}$$

where

$$\mathrm{kl}\left[\lambda_{q,i}\|\lambda_{p,i}\right] = \lambda_{q,i}\log\frac{\lambda_{q,i}}{\lambda_{p,i}} + (1-\lambda_{q,i})\log\left(\frac{1-\lambda_{q,i}}{1-\lambda_{p,i}}\right). \tag{4}$$

Here, $\lambda_{p,i}$ is set to the target sparsity, so if we want 90% sparsity winning tickets, then $\lambda_{p,i}$ is set to 0.1. Since the structure of the pruned network is given, we set the element of $\lambda_{q,i}$ to 0 or 1 asymptotically according to the pruning mask $\boldsymbol{m}$ and obtain the following kl divergence about λ. This operation has been conventionally done in the entropy discussion [28].

$$\mathrm{kl}\left[\lambda_{q,i}\|\lambda_{p,i}\right] = \begin{cases} -\log\lambda_{p,i} & \text{(unpruned)}\\ -\log(1-\lambda_{p,i}) & \text{(pruned)}\end{cases}. \tag{5}$$

The expected sharpness is as follows.

$$\mathcal{L}_{\mathcal{S}}(\mathbb{Q}(\boldsymbol{\theta})) - \mathcal{L}_{\mathcal{S}}(\bar{\boldsymbol{\theta}}) = \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}}\left[\mathcal{L}_{\mathcal{S}}(\bar{\boldsymbol{\theta}} + \boldsymbol{\epsilon})) - \mathcal{L}_{\mathcal{S}}(\bar{\boldsymbol{\theta}})\right],$$

where

$$\epsilon_i \sim \begin{cases}\mathcal{N}(x \mid 0, \sigma_{q,i}) & \text{(unpruned)}\\ \delta_{\{0\}} & \text{(pruned)}\end{cases}.$$

This means that flatness of pruned networks can be discussed adding noise to only unpruned weights. The advantage of this definition compared to the Gaussian distribution setting, where we cannot avoid adding noise to pruned weights, will be discussed in the next subsection.

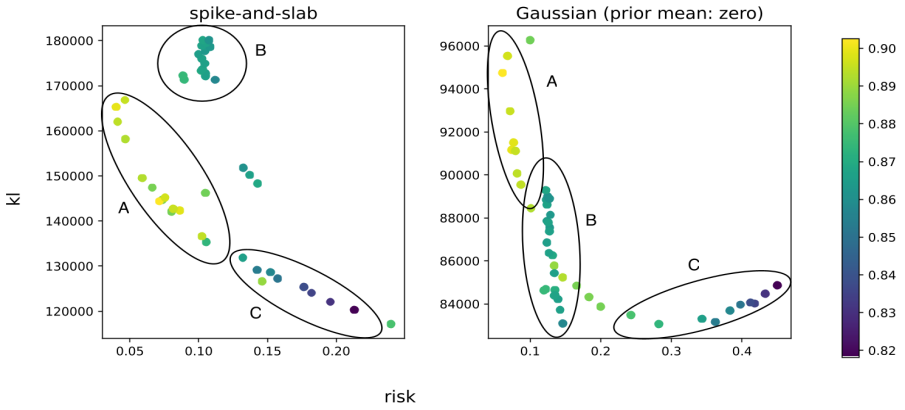


Figure 4: Correlation between training risk term and KL term when PAC-Bayes bound is optimized. We plot over subnetworks (90% sparse) generated by different learning rates of IMP (ResNet20 + CIFAR10). Left: spike-and-slab distribution, Right: Gaussian distribution with zero-mean prior.

4.3 Numerical Experiment

We conducted numerical experiments to confirm that our PAC-Bayes formulation can adequately explain the behavior of winning tickets. We optimized the posterior variance to minimize the PAC-Bayes bound, and plot KL term and the training loss on the posterior distribution to investigate that the bound can capture the test accuracy of the subnetworks produced by IMP. As a comparison, we also experiment with the Gaussian distribution setting using the zero-mean prior. The PAC-Bayesian bound used here is the variational KL bound [6] (Theorem D.2, and see also Appendix B.7), and we implemented the code following Dziugaite et al. [7].

Figure 4 shows the distribution of the training risk term and KL term when optimizing the PAC-Bayes bound, and the actual test accuracy is colored. We show that the bound with the spike-and-slab formulation successfully explains the behavior of winning tickets, dividing the point cloud into three groups: A) winning tickets (moderate learning rate), B) subnetworks that failed to find winning tickets (too high learning rate) and C) not much trained subnetworks (too small learning rate).

In the left figure, as the learning rate is increased, the distance from the initial weights increases and the training risk gradually decreases from C to A; The same trend also appears on the right figure. On the other hand, the distribution of B differs greatly between left and right. The right figure does not capture the test accuracy well because B should have higher test accuracy considering its KL term and training risk term. This means that, as the learning rate is increased and winning tickets can no longer be found, the KL term when prior mean is zero becomes much smaller than that of A. This is because many parameters become close to zero rather than near their initial weights when the winning ticket fails to be found (see Appendix B.5). In terms of not only the bound optimization but also the analysis of the existing winning ticket, it is preferable to use the spike-and-slab distribution to set the prior mean as the initial weights.

4.4 Revisiting existing algorithms

We reconsider the existing algorithms for winning ticket searching from the perspective of optimizing the PAC-Bayes bound. Although we focus only on IMP and continuous sparsification, this view could be helpful for other methods as well.

4.4.1 IMP

IMP is a heuristic method and does not have an explicit target function. Here, we explain how IMP behaves in the sense of a PAC-Bayes bound with our formulation instead of viewing IMP as a direct bound optimization problem.

The risk term and KL term in the PAC-Bayes bound are basically in a trade-off relationship; choosing a complex model to fit the training data will increase the KL term, while choosing a simple model

Table 2: Drop of training accuracy with different pruning criteria. We averaged the accuracy drop of five winning tickets (90% sparsity) produced by different learning rate (ResNet20, CIFAR10).

Criteria	After pruning (%)	After rewinding (%)	After retraining (%)
large_final	-19.8	-88.6	-0.3
small_final	-89.7	-89.5	-8.0
random	-88.8	-88.6	-5.7

may not have a high accuracy on the training data. We point out that the two steps of IMP: 1) train the subnetwork, 2) prune the subnetwork and revert its weights, optimize the overall bound by alternately reducing one term while suppressing the increase in the other term.

In the first step, IMP trains the subnetwork from initial weights under a given pruning mask. This process reduces the training risk, and the increase in KL term is not expected to be so large in our formulation because the prior mean is set to the initial weights and the trained weights of IMP are not far from the initial weights as shown in Sections 3.3 and 4.3.

In the second step, IMP prunes a certain percentage of the smallest magnitude weights and reverts the trained weights to the initial state. Reverting $\boldsymbol{\theta}$ to $\boldsymbol{\theta}_{\text{init}}$ and changing part of λ_q from 1 to 0 make the KL term small. The number of Gaussian KL summations decreases and the distance from initialization gets to 0, and the λ KL part is also reduced if the prior mixture ratio λ_p is set to the final target sparsity. This KL reduction is not dependent on the pruning criterion. The problem here is how to minimize the increase in training loss, which is related to what heuristic pruning criterion we choose and why pruning weights with a small absolute value works well.

Table 2 lists the training accuracy drop when we use three different pruning criteria; $large\_final$ leaves a large absolute value of weights and corresponds to IMP, $small\_final$ conversely leaves small weights, and $random$ prunes randomly. This notation of criteria follows that of Zhou et al. [43]. As expected, $large\_final$ has a smaller drop in training accuracy after pruning than the others. Training accuracy gets lower after reverting; If we assume that retraining can reach weights that show the same or better accuracy because weights achieving good accuracy with the same structure exist, it seems to make sense to use $large\_final$ to decrease the KL term while suppressing the increase in training loss. The results in Table 2 confirm this assumption empirically.

We can also discuss the reason why $large\_final$ pruning criterion is good for IMP by considering the following simple Taylor expansion (see Appendix B.6).

$$|f(\boldsymbol{\theta}+\Delta\boldsymbol{\theta})-f(\boldsymbol{\theta})| \leq \frac{1}{2}\|\Delta\boldsymbol{\theta}\|_2^2 \sup_{\gamma\in[0,1]} \lambda_{\max}^{\boldsymbol{\theta}+\gamma\Delta\boldsymbol{\theta}}, \tag{6}$$

where $\lambda_{\max}^{\boldsymbol{\theta}+\gamma\Delta\boldsymbol{\theta}}$ is a top eigenvalue of Hessian $H(\boldsymbol{\theta}+\gamma\Delta\boldsymbol{\theta})$.

This provides a brief insight into why IMP succeeds by pruning small magnitude weights under the assumption that the maximum eigenvalues are not very different.

4.4.2 Continuous Sparsification

Continuous sparsification [34] is a method to find winning tickets by removing the parameters continuously instead of alternating between training and pruning. This target function is as follows,

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d, \boldsymbol{m}\in\{0,1\}^d} \mathcal{L}_{\mathcal{S}}(\boldsymbol{m}\odot\boldsymbol{\theta}) + \eta\cdot\|\boldsymbol{m}\|_1, \tag{7}$$

where $\eta > 0$ is a hyperparameter. Continuous sparsification is formulated as the training loss minimization with the L_0 regularization of weights, and a sigmoid function σ is used for the continuous relaxation of the regularization term as follows.

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d, \boldsymbol{s}\in\mathbb{R}^d_{\neq 0}} \lim_{\beta\to\infty} \mathcal{L}_{\mathcal{S}}(\sigma(\beta\boldsymbol{s})\odot\boldsymbol{\theta}) + \eta\cdot\|\sigma(\beta\boldsymbol{s})\|_1. \tag{8}$$

We can regard this function as an approximation of the PAC-Bayes bound of our formulation. Let $\phi_i \geq 0$ be the Gaussian KL part in 3, the summation of training risk and KL is as follows.

$$\mathcal{L}_{\mathcal{S}}(\mathbb{Q}) + \sum_i \phi_i\lambda_{q,i} + \sum_i \mathrm{kl}\left[\lambda_{q,i}\|\lambda_{p,i}\right]. \tag{9}$$

We make three approximations: 1) replace $\mathbb{Q}$ with $\mathbb{E}[\boldsymbol{\theta}]$ over the spike-and-slab distribution by first-order Taylor expansion on the training risk, 2) simplify the second term to the L_1 norm of $\boldsymbol{\lambda_q}$ because the second term can be viewed as a weighted summation of $\lambda_{q,i}$, and 3) remove the third term, which can be regarded as a regularization to the target sparsity. This yields the following, which is similar to Eq. 8.

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d, \lambda_{q,i}\to\{0,1\}} \mathcal{L}_{\mathcal{S}}(\boldsymbol{\lambda_q} \odot \boldsymbol{\theta}) + \eta \cdot \|\boldsymbol{\lambda_q}\|_1. \tag{10}$$

Since the Gaussian KL ϕ_i is approximated, the distance from the initial weights is not taken into account in this setting. The authors adopt a problem setting where the weights trained a few epochs ahead instead of the initial weights are used for ticket search following Frankle et al. [11]; therefore their work does not have to consider suppressing the learning rate, i.e., the distance from the initial weights. Note that Hayou et al. [15] proposed PAC-Bayes pruning (PBP) by optimizing the PAC-Bayes bound. However, the limitation of our analysis is that we cannot reveal an explicit relationship between continuous sparsification and PBP.

5 Conclusion

In this work, we explored the fact that a small learning rate is required to find winning tickets, and we provided empirical analysis related to flatness and the distance from the initial weights. On the basis of these findings, we used the PAC-Bayesian framework to analyze winning tickets and experimentally showed that it captures the generalization behavior. Finally, we reconsidered IMP and continuous sparsification from a PAC-Bayesian perspective. In this study, we do not analyze the case where no solution exists near the initial weights, which needs IMP with rewinding to early epoch.

References

[1] Pierre Alquier, James Ridgway, and Nicolas Chopin. On the properties of variational approximations of gibbs posteriors. *The Journal of Machine Learning Research*, 17(1):8374–8414, 2016.

[2] Robert Bain. Visualizing the loss landscape of winning lottery tickets. *arXiv preprint arXiv:2112.08538*, 2021.

[3] Brian Bartoldson, Ari Morcos, Adrian Barbu, and Gordon Erlebacher. The generalization-stability tradeoff in neural network pruning. *Advances in Neural Information Processing Systems*, 33:20852–20864, 2020.

[4] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pages 1019–1028. PMLR, 2017.

[5] Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *In Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence*, 2017.

[6] Gintare Karolina Dziugaite, Kyle Hsu, Waseem Gharbieh, Gabriel Arpino, and Daniel Roy. On the role of data in pac-bayes. In *International Conference on Artificial Intelligence and Statistics*, pages 604–612. PMLR, 2021.

[7] Gintare Karolina Dziugaite, Kyle Hsu, Waseem Gharbieh, Gabriel Arpino, and Daniel M Roy. On the role of data in pac-bayes bounds. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

[8] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2020.

[9] Jonathan Frankle. Openlth: A framework for lottery tickets and beyond. `https://github.com/facebookresearch/open_lth`, 2020.

[10] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2018.

[11] 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.

[12] Song Han, Jeff Pool, John Tran, and William J Dally. Learning both weights and connections for efficient neural network. In *NIPS*, 2015.

[13] Song Han, Jeff Pool, Sharan Narang, Huizi Mao, Enhao Gong, Shijian Tang, Erich Elsen, Peter Vajda, Manohar Paluri, John Tran, et al. Dsd: Dense-sparse-dense training for deep neural networks. *In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net*, 2017.

[14] Babak Hassibi, David G Stork, and Gregory J Wolff. Optimal brain surgeon and general network pruning. In *IEEE international conference on neural networks*, pages 293–299. IEEE, 1993.

[15] Soufiane Hayou, Bobby He, and Gintare Karolina Dziugaite. Probabilistic fine-tuning of pruning masks and pac-bayes self-bounded learning. *arXiv preprint arXiv:2110.11804*, 2021.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] Zheng He, Quanzhi Zhu, and Zengchang Qin. Can network pruning benefit deep learning under label noise? 2022.

[18] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural computation*, 9(1):1–42, 1997.

[19] Stanislaw Jastrzebski, Maciej Szymczak, Stanislav Fort, Devansh Arpit, Jacek Tabor, Kyunghyun Cho*, and Krzysztof Geras*. The break-even point on optimization trajectories of deep neural networks. In *International Conference on Learning Representations*, 2020.

[20] Yiding Jiang*, Behnam Neyshabur*, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020.

[21] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *ICLR*, 2017.

[22] Steve Lawrence, C Lee Giles, and Ah Chung Tsoi. What size neural network gives optimal generalization? convergence properties of backpropagation. Technical report, 1998.

[23] Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

[24] Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari. The large learning rate phase of deep learning: the catapult mechanism. 2020.

[25] Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[26] Ning Liu, Geng Yuan, Zhengping Che, Xuan Shen, Xiaolong Ma, Qing Jin, Jian Ren, Jian Tang, Sijia Liu, and Yanzhi Wang. Lottery ticket preserves weight correlation: Is it desirable or not? In *International Conference on Machine Learning*, pages 7011–7020. PMLR, 2021.

[27] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. In *International Conference on Learning Representations*, 2018.

[28] David JC MacKay, David JC Mac Kay, et al. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[29] Eran Malach, Gilad Yehudai, Shai Shalev-Schwartz, and Ohad Shamir. Proving the lottery ticket hypothesis: Pruning is all you need. In *International Conference on Machine Learning*, pages 6682–6691. PMLR, 2020.

[30] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. In *ICLR (Workshop)*, 2015.

[31] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017.

[32] Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A PAC-bayesian approach to spectrally-normalized margin bounds for neural networks. In *International Conference on Learning Representations*, 2018.

[33] Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. What's hidden in a randomly weighted neural network? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[34] Pedro Savarese, Hugo Silva, and Michael Maire. Winning the lottery with continuous sparsification. *Advances in Neural Information Processing Systems*, 33:11380–11390, 2020.

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

[36] Francesco Tonolini, Bjørn Sand Jensen, and Roderick Murray-Smith. Variational sparse coding. In *Uncertainty in Artificial Intelligence*, pages 690–700. PMLR, 2020.

[37] Zeke Xie, Fengxiang He, Shaopeng Fu, Issei Sato, Dacheng Tao, and Masashi Sugiyama. Artificial neural variability for deep learning: on overfitting, noise memorization, and catastrophic forgetting. *Neural computation*, 33(8):2163–2192, 2021.

[38] Zeke Xie, Issei Sato, and Masashi Sugiyama. A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima. In *International Conference on Learning Representations*, 2021.

[39] Zhewei Yao, Amir Gholami, Qi Lei, Kurt Keutzer, and Michael W Mahoney. Hessian-based analysis of large batch training and robustness to adversaries. *Advances in Neural Information Processing Systems*, 31, 2018.

[40] Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 581–590. IEEE, 2020.

[41] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *ICLR*, 2017.

[42] Shuai Zhang, Meng Wang, Sijia Liu, Pin-Yu Chen, and Jinjun Xiong. Why lottery ticket wins? a theoretical perspective of sample complexity on sparse neural networks. *Advances in Neural Information Processing Systems*, 34, 2021.

[43] Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing lottery tickets: Zeros, signs, and the supermask. *Advances in neural information processing systems*, 32, 2019.

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? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
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]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (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]
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]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
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]

A Background

In this section, we explain the background knowledge for IMP and the PAC-Bayesian theory.

IMP Iterative pruning is a method of obtaining a subnetwork by repeating training and pruning in stages. Frankle and Carbin [10] showed that iterative pruning (Algorithm 1) could find the winning ticket by adding an operation to restore the weights to the initial weights after pruning.

Algorithm 1 Iterative Pruning for LTH

1: Randomly initialize the parameters $\boldsymbol{\theta}$ of a neural network to $\boldsymbol{\theta}_{\text{init}}$ and initialize a mask $\boldsymbol{m}$ to $1^{|\boldsymbol{\theta}_{\text{init}}|}$
2: Train the network $f(x; \boldsymbol{w} \odot \boldsymbol{m})$ for T iterations, producing network $f(x; \boldsymbol{\theta}_T \odot \boldsymbol{m})$
3: Produce a new mask based on the current mask criterion. Rank the unmasked weights by their scores, set the mask value to 0 for the bottom $p\%$, the top $(100-p)\%$ to 1.
4: If the target pruning ratio is satisfied, the resulting network is $f(x; \boldsymbol{\theta}_T \odot \boldsymbol{m})$
5: Otherwise, reset $\boldsymbol{\theta}$ to $\boldsymbol{\theta}_{\text{init}}$, and go back to step 2

In Frankle and Carbin [10], the mask criterion is simply to keep the weights with a large final magnitude, $|\theta_T|$; This is called Iterative Magnitude Pruning (IMP). This paper follows their setting: p is set to 0.2 and the models are pruned globally.

Frankle et al. [11] proposed an iterative pruning with rewinding to avoid lower-stability phase to SGD noise in the early training in each IMP step. The algorithm does not return to the exact initial weights, but instead returns the weights trained slightly in advance as the initial weights. It can find the winning ticket even with the initial large learning rate or in the harder settings such as ImageNet; however it revises the original LTH setting. Our paper focuses on analyzing the properties of winning tickets rather than improving the accuracy or robustness of winning tickets; thus, we will not consider this problem setting.

Liu et al. [27] claimed that the large learning rate performs better in the larger model settings such as ResNet56 and ResNet110, contrary to Frankle and Carbin [10], which stated that IMP requires the small learning rate to find winning tickets. From our findings in Section 3.3 and 4.3, we suppose that there is no good solution near the initial weights in such a setting; therefore the small learning rate cannot find the winning tickets, and that IMP with rewinding is effective in such a setting because it shifts the initial weights closer to the final trained weights by pretraining a little and searches for the winning ticket in a better lottery.

PAC-Bayesian Theory First, we provide the notations used in this paper. Denote the training sample $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^m \in (\mathcal{X} \times \mathcal{Y})^m$, which is randomly sampled from an underlying data distribution $\mathcal{D}$. Let $\mathcal{H}$ be a set of hypotheses, and $\ell : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to [0, 1]$ be a loss function. Given $f \in \mathcal{H}$, we formulate the empirical risk on $\mathcal{S}$ and the generalization error on $\mathcal{D}$ as

$$\mathcal{L}_{\mathcal{S}}(f) = \frac{1}{m} \sum_{i=1}^{m} \ell\left(f, x_i, y_i\right); \quad \mathcal{L}_{\mathcal{D}}(f) = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \ell(f, x, y). \tag{11}$$

The PAC-Bayesian framework gives a bound on the generalization error of a posterior distribution $\mathbb{Q}$ over the hypothesis $\mathcal{H}$; we denote it $\mathcal{L}_{\mathcal{D}}(\mathbb{Q})$. It assumes that we have a prior distribution $\mathbb{P}$ on $\mathcal{H}$ which does not depend on training data, and we update it to $\mathbb{Q}$ through the learning process. Optimizing the PAC-Bayes bound controls the balance between the empirical risk and the closeness to the prior (small complexity of the model). Although there are several types of PAC-Bayes bound, we consider the following well-used form of the PAC-Bayes bound.

Theorem 1 (Alquier et al. bound [1]) *Given a real number $\delta \in (0, 1]$, a non-negative real number η, and a prior distribution $\mathbb{P}$ on $\mathcal{H}$ defined before seeing any training sample $(X, Y) \in \mathcal{S}$, with probability at least $1 - \delta$, for all $\mathbb{Q}$ on $\mathcal{H}$*

$$\mathcal{L}_{\mathcal{D}}(\mathbb{Q}) \leq \mathcal{L}_{\mathcal{S}}(\mathbb{Q}) + \frac{1}{\eta}\left(\mathrm{KL}[\mathbb{Q}\|\mathbb{P}] + \log\frac{1}{\delta} + \Psi(\eta, m)\right), \tag{12}$$

where

$$\Psi(\eta, m) \coloneqq \log \mathop{\mathbb{E}}_{\substack{h\sim\mathbb{P}, \\ \mathcal{S}\sim\mathcal{D}^m}} \left[\exp\left(\eta\left(\mathcal{L}_{\mathcal{D}}(h) - \mathcal{L}_{\mathcal{S}}(h)\right)\right)\right]. \tag{13}$$

It is known that flatness is closely related to the generalization ability of neural networks [21] [20]. As shown in previous studies ([5], [31] [32]), it can be viewed in the expected sense from a PAC-Bayesian perspective. We decompose the PAC-Bayes bound as follows.

$$\mathcal{L}_{\mathcal{D}}(\mathbb{Q}) \leq \mathcal{L}_{\mathcal{S}}(f) + \underbrace{\mathcal{L}_{\mathcal{S}}(\mathbb{Q}) - \mathcal{L}_{\mathcal{S}}(f)}_{\text{expected sharpness}} + \frac{1}{\eta}\left(\underbrace{\mathrm{KL}[\mathbb{Q}\|\mathbb{P}]}_{\mathrm{KL}} + \log\frac{1}{\delta}\right) + \Psi(\eta, m), \tag{14}$$

where $f \in \mathcal{H}$ is a solution obtained by a training. The expected sharpness term represents the amount of change in empirical risk around the trained weights, and the solutions in flatter minima are expected to have a relatively smaller value of this term. In the PAC-Bayesian framework, the role flatness plays in generalization behavior can be understood in this way.

For example, we consider the case where the Gaussian distribution is used for the prior and posterior. Let $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ be the Gaussian distribution, where $\boldsymbol{\mu}$ is the mean and $\boldsymbol{\Sigma}$, is the covariance matrix and $\boldsymbol{\theta}$ be the parameters of a neural network. We set a prior $\mathbb{P}$ to be $\mathcal{N}(\mathbf{0}, \sigma^2\boldsymbol{I})$ and a posterior $\mathbb{Q}$ to be $\mathcal{N}(\boldsymbol{\theta}, \sigma^2\boldsymbol{I})$, where $\sigma > 0$, and the KL term is calculated as $\|\boldsymbol{\theta}\|_2^2/(2\sigma^2)$. This is consistent with the conventional understanding: solutions in flat minima obtained with norm regularization can achieve good generalization accuracy.

When considering this PAC-Bayesian framework in the pruned networks, we should note that prior $\mathbb{P}$ has to be defined without depending on the training samples $\mathcal{S}$. It can be thought that if the parameter space is restricted to an unpruned weight subspace, we do not have to consider the pruned network case differently. However this is not valid because this prior depends on the structure of the pruning mask $\boldsymbol{m}$, which is found after seeing the training samples $\mathcal{S}$. Target sparsity β does not depend on $\mathcal{S}$, so we can use it in the prior.

B Other Experiments

B.1 The test accuracy when SAM and NVRM are used

Table 3: Test accuracy on CIFAR10 and CIFAR100 when SAM and NVRM are used. The hyper-parameter of SAM ρ is selected from $\{0.05, 0.1, 0.2, 0.5\}$, and that of NVRM b is selected from $\{0.014, 0.018, 0.022, 0.026\}$. As a baseline, we show the results of SGD with a small learning rate, and the learning rate is equal for SAM and NVRM. We averaged over three times.

		ResNet20			VGG16		
Dataset	Sparsity (%)	SGD	SAM	NVRM	SGD	SAM	NVRM
CIFAR10	90	89.1	**89.7**	89.3	91.8	**92.9**	92.7
	95	**87.3**	**87.3**	87.1	91.8	**93.1**	92.7
CIFAR100	90	61.2	**62.2**	61.7	67.1	70.4	**71.2**
	95	45.2	45.2	**46.7**	66.4	70.5	**71.0**

In Section 3.2, we observe that the winning ticket is in a relatively sharper minima due to a small learning rate and that IMP can find a flatter minimum by using SAM or NVRM. Table 3 shows the test accuracy when a flatter solution is obtained by using SAM and NVRM. They can achieve a test accuracy the same as or even better than the SGD with a small learning rate, and the improvement in test accuracy can be seen especially in VGG16. It is considered that since VGG16 has a larger number of parameters than ResNet20, a flatter solution with a relatively small training loss could be found. We found no significant difference in test accuracy between SAM and NVRM.

B.2 SAM with the large learning rate

Figure 5 shows the test accuracy when we use SAM with a large learning rate; we experimented only ResNet20 + CIFAR10 due to computational resource limitations. This figure shows that SAM does not improve test accuracy from that of IMP with a large learning rate and that SAM does not help to find winning tickets because the test accuracy continues to decline as sparsity increases. Since IMP with a large learning rate already produces relatively flatter solutions, SAM will not change the

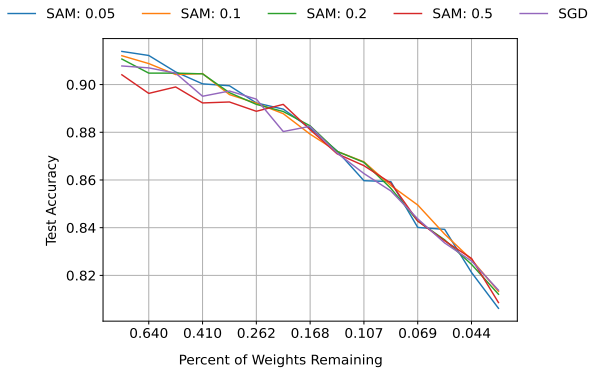


Figure 5: Test accuracy on CIFAR10 when we train ResNet20 using SAM for IMP with a large learning rate. We show the results of different SAM hyperparameters $\rho \in \{0.05, 0.1, 0.2, 0.5\}$ and SGD result as a baseline.

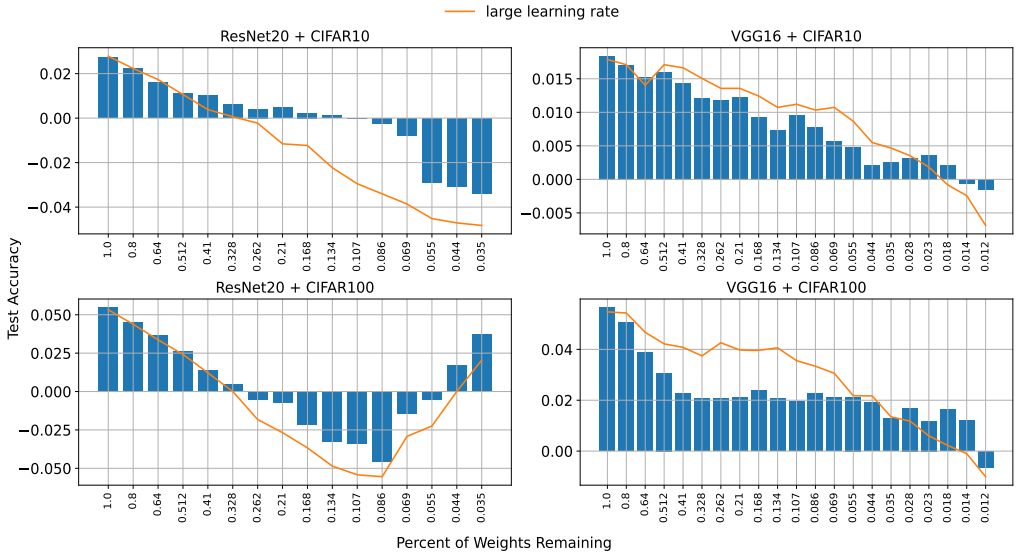


Figure 6: Test accuracy improvement on CIFAR10 and CIFAR100 when we ran IMP with a small learning rate to the target sparsity and train the subnetwork from initialization with a large learning rate (blue bar) instead of a small learning rate (0, baseline). For comparison, we plot the case of IMP with a large learning rate that corresponds to the orange line.

results as expected. This also confirms that the improved generalization accuracy when we use SAM instead of SGD for IMP with a small learning rate is because of finding flatter solution, not some other side effects of SAM.

B.3 Mask structure with different learning rates

We conducted the following experiment related to distance from initial weights. In Figure 6, we found that IMP with different learning rates produce the pruning masks with different properties; learning with a large learning rate on a sub-network obtained with a small learning rate also does not provide winning tickets. Retraining with a large learning rate for a given mask greatly improves test accuracy at a low sparsity. In contrast, as sparsity is increased, this improvement decreases or the test accuracy worsens; it is generally the same trend as IMP with the large learning rate (orange line). We found that the mask obtained by IMP with a small learning rate has a structure that performs well with weights around the initialization, and it does not work well when trained directly of the subnetwork with a large learning rate.

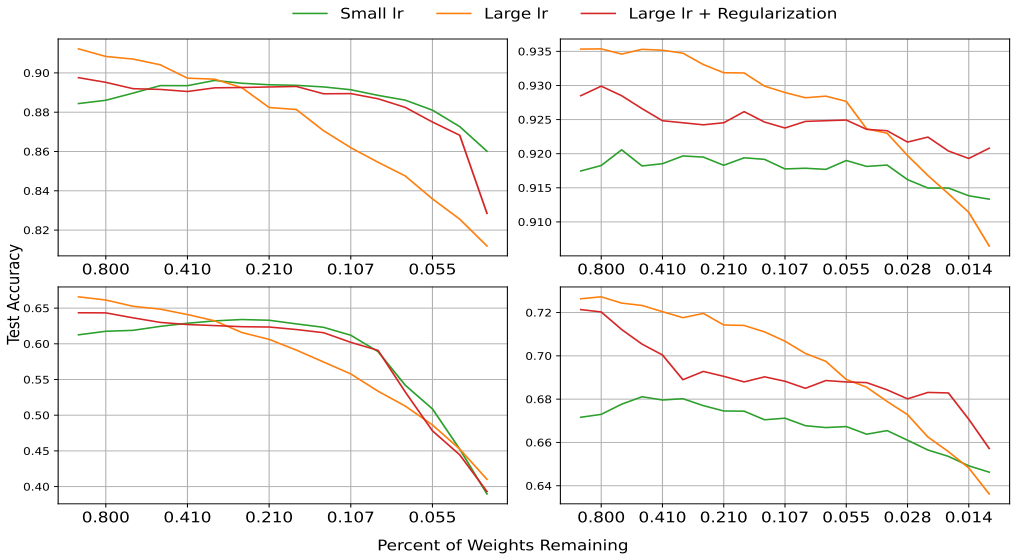


Figure 7: Test accuracy on CIFAR10 and CIFAR100 when regularization from the initial weights is added. Top Left: ResNet20 + CIFAR10, Top Right: VGG16 + CIFAR10, Bottom Left: ResNet20 + CIFAR100, Bottom Right: VGG16 + CIFAR100.

B.4 Regularization of the distance from the initial weights

Figure 7 shows the test accuracy on CIFAR10 and CIFAR100 when we trained ResNet20 and VGG16 with a regularization from the initial weights. For a large learning rate setting (orange line), increasing sparsity significantly reduces the test accuracy, which means that IMP cannot find the winning ticket. By adding regularization from the initialization, this decrease in the test accuracy can be reduced (red line), showing a similar trend for a small learning setting (green line), which is successful in finding winning tickets.

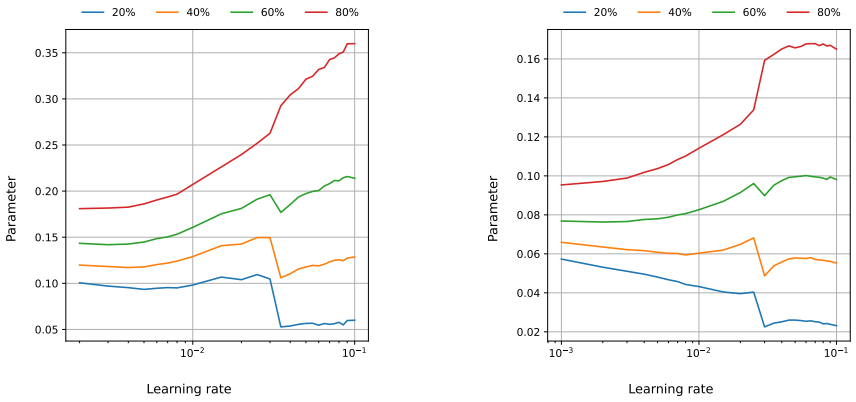
B.5 The distribution of parameter changing the learning rate

Figure 8 shows the parameter distribution of ResNet20 and VGG16 changing the learning rate. We trained them on CIFAR10 and plot the unpruned weights in order from the smallest to largest: 20%, 40%, 60%, 80%. The same sparsity where the winning tickets cannot be found in Figure 1 shows a change in the trend of the distribution. Although the difference is not apparent when we measure L2 norm, we confirmed that each parameter, which was near the initial weights originally, becomes distributed in a wider range when the learning rate is increased.

B.6 Proof of Eq. 6

We estimate the deviation of f when the $\Delta\boldsymbol{\theta}$ moves from the trained weights $\boldsymbol{\theta}$ using Taylor's theorem. Let H be a Hessian matrix, then we have

$$
\begin{aligned}
|f(\boldsymbol{\theta}+\Delta\boldsymbol{\theta}) - f(\boldsymbol{\theta})| &= |\nabla f(\boldsymbol{\theta})\cdot\Delta\boldsymbol{\theta} + \frac{1}{2}\Delta\boldsymbol{\theta}^\top H(\boldsymbol{\theta}+\gamma\Delta\boldsymbol{\theta})\Delta\boldsymbol{\theta}| \\
&= \frac{1}{2}\|\Delta\boldsymbol{\theta}^\top H(\boldsymbol{\theta}+\gamma\Delta\boldsymbol{\theta})\Delta\boldsymbol{\theta}\|_2 \\
&\leq \frac{1}{2}\|\Delta\boldsymbol{\theta}\|_2^2 \cdot \|H(\boldsymbol{\theta}+\gamma\Delta\boldsymbol{\theta}\|_2 \\
&\leq \frac{1}{2}\|\Delta\boldsymbol{\theta}\|_2^2 \sup_{\gamma\in[0,1]} \|H(\boldsymbol{\theta}+\gamma\Delta\boldsymbol{\theta})\|_2 \\
&= \frac{1}{2}\|\Delta\boldsymbol{\theta}\|_2^2 \sup_{\gamma\in[0,1]} \lambda_{\max}^{\boldsymbol{\theta}+\gamma\Delta\boldsymbol{\theta}},
\end{aligned}
\tag{15}
$$



(a) ResNet20 (b) VGG16

Figure 8: Parameter distribution of ResNet20 and VGG16 trained on CIFAR10. In each learning rate, the trained parameters are plotted in the increasing order of weights: 20%, 40%, 60%, 80%. The subnetwork of ResNet20 has 90% sparsity, and that of VGG16 has 99% sparsity.

where $\lambda_{\max}^{\boldsymbol{\theta}+\gamma\Delta\boldsymbol{\theta}}$ is a top eigenvalue of $H(\boldsymbol{\theta}+\Delta\boldsymbol{\theta})$.

First equation comes from quadratic Taylor's theorem, second equation comes from the fact that θ is a trained weights and $\nabla f(\boldsymbol{\theta}) = 0$, and third inequality holds because of sub-multiplicativity of matrix norm.

B.7 Variational KL bound

In Section 4.3, we optimized the following variational KL bound [6], [15]. Given a real number $\delta \in (0, 1]$, with probability $1-\delta$ over the training sample $\mathcal{S}$,

$$\min \left\{ \begin{array}{l} \mathcal{L}_{\mathcal{S}}(\mathbb{Q}) + B + \sqrt{B\left(B + 2\mathcal{L}_{\mathcal{S}}(\mathbb{Q})\right)} \\ \mathcal{L}_{\mathcal{S}}(\mathbb{Q}) + \sqrt{\frac{B}{2}} \end{array} \right. , \tag{16}$$

where

$$B = \frac{\mathrm{KL}(\mathbb{Q}\|\mathbb{P}) + \log \frac{2\sqrt{|\mathcal{S}|}}{\delta}}{|\mathcal{S}|}. \tag{17}$$

We used this bound in our experiment because it can avoid selecting variables that appear in the PAC-Bayes bound, such as η in Eq. 12.