CAN NETWORK PRUNING BENEFIT DEEP LEARNING UNDER LABEL NOISE?

Anonymous authors
Paper under double-blind review

ABSTRACT

Network pruning is a widely-used technique to reduce the computational cost of over-parameterized neural networks. Conventional wisdom also regards pruning as a way to improve generalization: by zeroing out parameters, pruning reduces model capacity and prevents overfitting. However, this wisdom is facing challenges in a line of recent studies, which show that over-parameterization actually helps generalization. In this work, we demonstrate the existence of a novel *double descent* phenomenon in sparse regimes, namely, in the presence of label noise, medium sparsity induced by pruning hurts model performance, while high sparsity benefits. Through extensive experiments on noisy versions of MNIST, CIFAR-10 and CIFAR-100, We show that proper pruning could consistently promise non-trivial robustness against label noise, which provides a new lens for studying network pruning. Further, we reassess some common beliefs concerning the generalization of sparse networks, and hypothesize it is the distance from initialization that is key to robustness rather than sharpness/flatness. Experimental results correlate with this hypothesis. Together, our study provides valuable insight on whether, when and why network pruning benefits deep learning under label noise.

1 INTRODUCTION

Deep neural networks (DNNs) have more learnable parameters than training examples, and can easily memorize entire random-labeled dataset (Zhang et al., 2017). With excessive learning capability, these networks are susceptible to mislabeled data and tend to overfit quickly during training. Moreover, standard regularization techniques, like weight decay and dropout, are not sufficient to eliminate overfitting by themselves (Song et al., 2020; Zhang et al., 2017). As noisy labels exist pervasively in real world datasets (Shankar et al., 2020; Northcutt et al., 2021a;b), study on deep network memorization and generalization behavior is crucial to enhance model robustness.

Prior studies have demonstrated that networks learn simpler patterns first and are less prone to memorize noisy labels with limited capacity (Arpit et al., 2017; Li et al., 2020b), which accounts for the success of early stopping and robust regularization methods in label-noise-learning scenarios (Azadi et al., 2016; Tanno et al., 2019; Hu et al., 2020; Xia et al., 2020). Following Occam's razor, network pruning that aims to reduce parameter counts could also be regarded as some kind of regularization on model capacity (LeCun et al., 1990; Hassibi & Stork, 1992). By restricting a subset of model parameters to a value of zero, pruning imposes sparsity constraints on neural networks and penalizes its redundant expressive power. Furthermore, there are other conjectures on how pruning can benefit generalization, e.g., pruning creates sparsified versions of data representation, which introduce noise and encourage flatness into neural networks (Han et al., 2017; Bartoldson et al., 2020), as flatness of minima is usually correlated with good generalization (Keskar et al., 2017; Zhu et al., 2019).

Although pruning has been widely investigated at the target of storage and computational savings, it still remains unclear whether pruning will provide an added edge on label-noise-learning robustness (Hoefler et al., 2021). Given the discussion above, it is intuitive to suppose that pruning can enhance model performance and prevent overfitting. However, we find that in the presence of label noise, the generalization behavior of sparse neural networks diverge markedly from such intuition.

In this paper, we demonstrate the existence of double descent phenomenon in sparse regimes in the presence of label noise (Figures 1 and 8). We show that at low sparsities model performance might degrade as pruning, while at non-trivial sparsities, the behaviors of sparse networks resemble that of

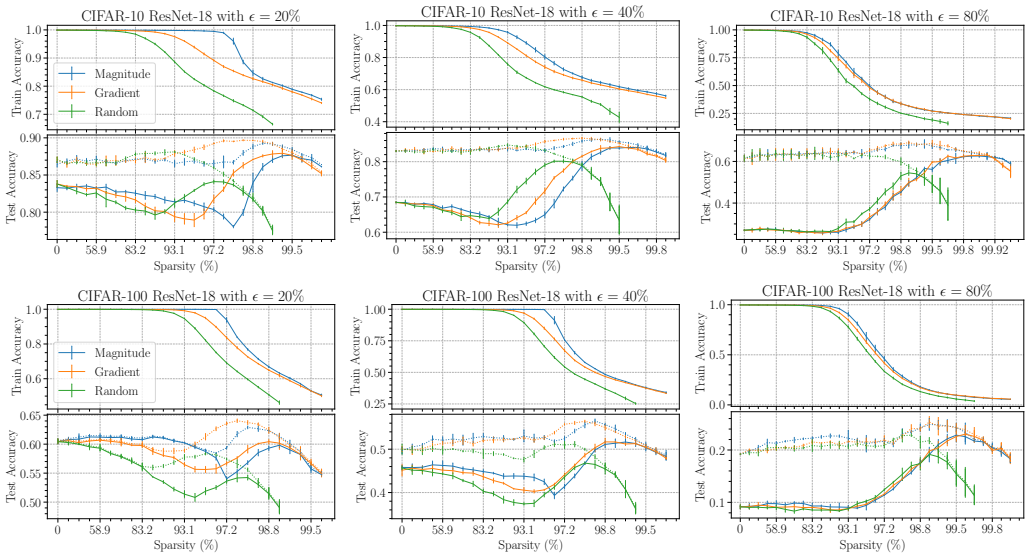


Figure 1: Double descent phenomenon in sparse regimes for ResNet-18 with three pruning strategies and varying permuted fraction ϵ. **Top**: CIFAR-10. **Bottom**: CIFAR-100. We plot **train accuracy** (solid lines in the upper sub-figures) and the **last test accuracy** of the final epoch (solid lines in the lower sub-figures), as well as the **best test accuracy** across all epochs (dotted lines).

under-parameterized dense models, exhibiting a U-like curve of bias-variance tradeoff. Moreover, at the "sweet-spot" sparsity, even with a majority of parameters removed by pruning, sparse neural networks still classify clean labels correctly and neglect noisy labels, resulting in significant robust performance at early stopping epoch and after that (Figure 2).

To verify the ubiquity of such phenomenon, we conduct thorough experiments with several commonly implemented pruning heuristics across different datasets and network architectures. Since finetuning a pruned network might confine it to sub-optimal minima, we utilize the technique proposed in *lottery ticket hypothesis* (Frankle & Carbin, 2019) to train a network from near initialization. We implement three common pruning heuristics and prune networks to a wide range of sparsities, in order to assess the pruning efficacy and label-noise robustness. The superior performance of highly sparse networks under label noise suggests a new lens for studying network pruning, and also opens new avenues to leverage existing pruning heuristics to facilitate robust training.

Furthermore, we investigate the possible causes for the benefits brought by pruning. We hypothesize that the robustness of highly sparse networks could be mainly ascribed to their incapability to move far from initialization, rather than the flatness in final solutions. To test this hypothesis, we adopt a *re-dense* training methods: after training a sparse network, we recover its pruned weights for a further retraining. The phenomenon that networks escape from highly sparse solutions during re-dense training, even with a small learning rate, provides evidence against the conjecture regarding flatness in minima. In contrast, experimental results establish a correlation between the distance from initialization and test performance of sparse and re-dense neural networks, supporting our hypothesis to a certain extent.

Our **main contributions** are summarized as follows:

- We demonstrate the double descent phenomenon in sparse regimes, where inappropriate sparsity leads to severe overfitting, while high sparsity promises significant label-noise robustness.
- We show that magnitude-based pruning enables models to withstand the least remained parameters without losing training performance; whereas gradient-based pruning attains top early-stopping test accuracy, which manifests potentials of preserving first-order information for robust pruning.
- We hypothesize that high sparsity traps optimizer into minima near initialization, and underline the critical role of the distance from initialization in the robustness of highly sparse networks. We present experimental evidence for this hypothesis.

2 RELATED WORK

Modern deep networks have the ability to memorize noise (Zhang et al., 2017), yet in practice, they do not learn via pure memorization and often achieve higher generalization performance than their compact counterparts (Arpit et al., 2017; Neyshabur et al., 2015). Such property of over-parameterized neural networks results in a double descent in test risk as we increase the model size (Belkin et al., 2019; Nakkiran et al., 2020). Furthermore, the superiority of over-parameterization casts doubt on the widely held viewpoint that pruning reduces model capacity thus helps generalization (LeCun et al., 1990; Hassibi & Stork, 1992; Molchanov et al., 2017a; Hoefler et al., 2021).

To explain the success of numerous pruning methods in pratice (Han et al., 2015; 2016; Liu et al., 2017; Molchanov et al., 2017a; Louizos et al., 2018; Frankle & Carbin, 2019), Bartoldson et al. (2020) propose that benefits of pruning for generalization attribute to the regularization effect of noise injection, which does not depend on parameter removal. Nevertheless, they mainly focus on low sparsities where pruned networks still reach full training performance. Whether reducing model capacity to a non-trivial extent can prevent overfitting and improve generalization is still unanswered.

The recent work by Chang et al. (2021) also affirms that pruned models exhibit a double descent phenomenon, which is seemingly identical to our claims. However, there are essential differences between their studies and ours on the basis of research focuses and methodologies. They grow the width of original model but fix the parameter count of pruned model, in order to manifest the advantages of over-parameterization in network pruning. While we conduct pruning on the same original network and compress it to increasing sparsities, to investigate the impact of reduced capacity on sparse networks under label noise.

So far, the relationship between sparsity, learning dynamics and generalization remains as open question and has received growing attention from researchers. Emerging studies from the perspective of loss landscape provide enlightening insight into understanding the behaviors of sparse regimes. Evci et al. (2020) reveal the existence of bad solutions in *sparse subspace* (namely, the sparsity pattern found by pruning), and illustrate the difficulty of escaping from bad solutions to good ones. And Lin et al. (2021) provide theoretical justification that sparsity can deteriorate the loss landscape by creating spurious local minima or spurious valleys. Our work is motivated by these findings, and what's more, moves a step further by empirically demonstrating that the reshaping effect on loss landscape by network pruning is actually beneficial in the presence of label noise.

While our focus has been on the characteristics of highly sparse neural networks under noisy labels, there are other research hot-spots concerning label-noise learning, e.g., designing state-of-the-art robust training algorithms (Han et al., 2018; Jiang et al., 2018; Li et al., 2019; 2020a). Among these methods, we find CDR proposed by Xia et al. (2020) particularly related regarding the way to hinder memorization. Using a similar criterion to gradient-based pruning, they identify non-critical parameters and penalize them during optimization. By deactivating redundant parameters, memorization of noisy labels is hindered, and test performance before early stopping is enhanced. While our results reveal that, with a large proportion of parameters being removed permanently, performance after early stopping could also be boosted greatly.

Finally, our findings are consistent with existing literature that discovers sparsification impairs memorization (Molchanov et al., 2017a; Lee et al., 2019; Hooker et al., 2019; Goel & Chen, 2021). Yet, to the best of our knowledge, we are the first to thoroughly investigate the label-noise robustness of pruned networks across a wide range of sparsities, and conduct further investigation into the reasons behind it. In this work, we do not chase state-of-the-art accuracy nor the computing resource efficiency; thus we simply apply unstructured iterative pruning techniques with static sparsity patterns, for they are easily adjusted to different tasks, architectures and pruning fractions.

3 PRELIMINARIES AND METHODOLOGY

3.1 NETWORKS AND DATASETS

We conduct experiments on image classification task. To verify the ubiquity of double descent phenomenon, we test manually corrupted version of three commonly used datasets, i.e., MNIST (LeCun et al., 1998), CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009). We train Lenet-300-

100 on MNIST, ResNet-18 on CIFAR-10 and CIFAR-100. We use the SGD optimizer and adopt commonly used hyperparameters for training and pruning. The model architectures and training details can be found in Appendix A.1. We repeat experiments five (MNIST) or three (CIFAR-10 and CIFAR-100) times with different seeds and plot the mean and standard deviation.

We consider symmetric label noise in this paper, which is generated by randomly permuted the labels for a fraction ϵ of the training data. The permuted fractions ϵ are set to 20%, 40% and 80%. Following work of Stephenson et al. (2021) we divide the dataset into four subsets as follows:

- *Permuted samples*: the subset of training data with labels randomly replaced by all possible labels with uniform probability. Though a small part of samples may be assigned to labels identical to their original ones after permutation, by flipping labels in each class uniformly, this subset contains no information of correct labels. Hence, high classification accuracy of permuted samples indicates a high degree of rote memorization in networks.
- *Unpermuted samples*: the subset of training data with labels that are correct and never permuted. Samples having labels assigned to the correct class after random permutation are not counted. This subset could reflect the ability of networks to learn generalized features.
- *Restored samples*: the subset of training data with examples identical to permuted samples, while keeping their original, correct labels. The more restored samples are correctly classified, the more robust the model is to label noise.
- *Test samples*: the test data that are held out for evaluation. All test samples have correct labels.

3.2 Pruning and Retraining Techniques

Network pruning is an effective technique to enhance the efficiency of deep networks with limited computational budget, by removing dispensable weights, filters or other structures from neural networks. A common approach to recover network performance after pruning is retraining, which means training the pruned networks for some extra epochs. The typical retraining based pruning procedure consists of three stages (Liu et al., 2019): 1) train a large, dense neural network to completion, 2) prune structures of the trained network according to certain heuristic, 3) retrain the network for t epochs to mitigate accuracy loss. Pruning and retraining can be repeated *iteratively*, or conducted only in *one shot* (Han et al., 2015).

Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, we define a neural network classifier function as $f(\mathbf{w}; \mathcal{D})$, where $\mathbf{w} \in \mathbb{R}^d$ is the set of weights, and d is the total number of weights. As pruning removes structures from a network, we introduce binary masks $\mathbf{m} \in \{0, 1\}^d$ as auxiliary to represent the remained weights $\mathbf{w} \odot \mathbf{m}$ and the function under sparsity constraints $f(\mathbf{w} \odot \mathbf{m}; \mathcal{D})$, where $\odot$ is the element-wise product. The sparsity of a pruned network is calculated as: $1 - \sum_{i=1}^d \mathbf{m}_i / d$.

Pruning strategies. We use three existing pruning heuristics summed up by Blalock et al. (2020). Magnitude-based pruning is one of the most commonly used baselines, and has been shown to achieve comparable performance to many complex techniques (Han et al., 2015; 2016; Gale et al., 2019). Gradient-based pruning preserves training dynamics and provides possibility to prune a network early in training (Lee et al., 2019; 2020). And random pruning is often regraded as a naive method, setting the performance benchmark that any elaborately designed method should surpass (Frankle et al., 2021). We prune weights in a network globally by comparing them across layers with the mentioned heuristics, and the details are listed below. We mainly present and discuss results of magnitude-based pruning, unless otherwise specified.

- *Magnitude-based pruning*: prunes the weights with the lowest absolute magnitudes $|\mathbf{w}|$.
- *Gradient-based pruning*: prunes the weights with the lowest absolute values of magnitude multiplies gradient $|\frac{\partial L}{\partial \mathbf{w}} \odot \mathbf{w}|$, with L be the loss function evaluated on a random batch of inputs.
- *Random pruning*: issues each weight with a random score sampled independently from the uniform distribution $\mathcal{U}(0, 1)$, and prunes the weights with the lowest scores.

Retraining methods. Along side the sparse structures induced by different pruning strategies, retraining methods also affect network performance by determining which point on the optimization landscape to start training from, i.e., near initialization or close to the final weights; or which learning rate schedule to utilize. In this work, we investigate the performance of *lottery ticket rewinding*

(LTR), which rewinds the unpruned weights by setting their value back to early iteration k in the original training phase, and retrains the sparse network from there using the same learning rate schedule as iteration k (Frankle & Carbin, 2019; Frankle et al., 2020). Networks are pruned and retrained iteratively, and during each iteration, 20% of weights will be pruned.

Sparsities. We divide sparsities into four ranges. Different from previous works (Frankle et al., 2020; 2021), we delimit boundaries with respect to particular pruning technique, depending on both training and test accuracy after retraining. *Trivial sparsities* are low sparsities where the network is so overparameterized that pruned network can still reach full training accuracy. *Critical sparsities* lie in a interval around the *interpolation threshold* (Nakkiran et al., 2020) where training accuracy starts to drop and test accuracy might decrease or increase when increasing sparsity. *High sparsities* are robust sparsities where test accuracy can significantly be boosted. *Extreme sparsities* are those beyond. To display of our results in detail, for Figures 2, 3 and 4, we sample five sparsities, i.e., the zero sparsity (dense model), the sparsity where the last test accuracy degrades the most (overfitting model), the sparsities where the best test accuracy or the last test accuracy reaches a peak (sweet-spot models), and the sparsity where both the best and the last test accuracy suffer (underfitting model).

3.3 RE-DENSE TRAINING AND LOSS FUNCTION VISUALIZATION APPROACH

Pruning induces sparsity constraints into the objective function optimization problem, which move the optimization to a lower-dimension space. To empirically investigate the impact of sparsity constraints, we present studies which allow pruned weights to return to the model, and utilize the loss surface visualization for analysis.

Dense, sparse, and re-dense training flow. Conventional wisdom believes that sparsity regularizes the neural networks and moves the optimization to a better local minima where the loss surface is flatter. To test this hypothesis in label-noise settings, we adopt the *re-dense training* step in the work by Han et al. (2017): after training a pruned network for t epochs, we recover pruned weights in the network, initialized them to zero, and retrain entire network for another t epochs with the fixed learning rate. We set the learning rate in re-dense training equal to the last learning rate of sparse training. Other learning hyperparameters (batch size, momentum, weight decay, etc.) are kept the same as original training process.

1-D loss function visualization. Visualizing the loss landscape can provide an empirical characterization of the geometry of neural network minimizers (e.g., their sharpness/flatness, or the structures of surrounding parameter space). We present linear interpolation plots of the training loss function along a line segment θ between sparse solutions θ_s and re-dense solutions θ_r using the strategy proposed by Goodfellow & Vinyals (2015). We define $\theta(\alpha) = (1-\alpha)\theta_s + \alpha\theta_r$ for $\alpha \in [0, 1]$ with increment of 0.01. If there exists a monotonically decreasing objective from sparse solutions to re-dense solutions, we may conjecture that sparsity obstructs the optimization process with less trainable parameters. We further plot 1-D loss function over a center minimizer θ using filter-wise normalized directions as introduced by Li et al. (2018), to visualize the loss curvature of θ and make comparisons between different minimizers.

4 DOUBLE DESCENT PHENOMENON IN SPARSE REGIMES

Here, we demonstrate a novel double descent phenomenon with respect to model sparsities under label noise settings. Contradict to common beliefs that pruning reduces overfitting and helps generalization, our experiments reveal the similarities between pruned sparse networks and small dense networks, e.g., they both present a peak in test error near the interpolation threshold under label noise (Figure 1). Nevertheless, sparse neural networks reach the interpolation threshold with less parameter count, and possess non-trivial robust performance at high sparsities compared with dense networks (Nakkiran et al., 2020).

Observation 1: *Medium sparsities hurt generalization, while high sparsities enhance label-noise robustness.*

Figures 1 and 8 summarize the double descent behavior of sparse neural networks across different datasets, permuted label fractions and pruning strategies. In most cases except for those with extreme label noise, increasing sparsity of networks results in a first decrease and then increase in the last test

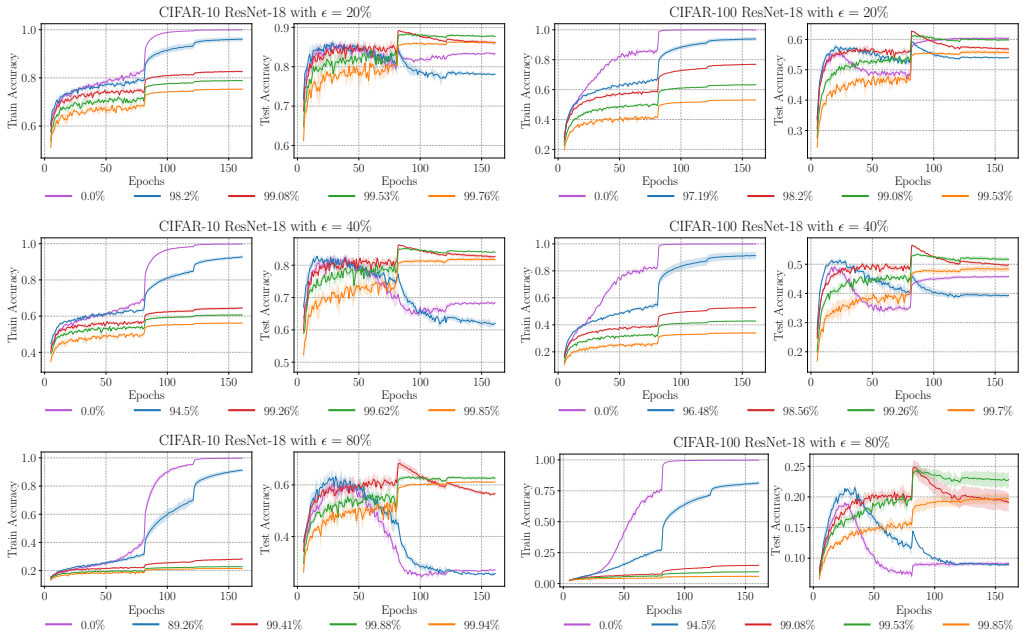


Figure 2: Training dynamics w.r.t. epochs at five sparsities across different permuted fractions ϵ. Models are ResNet-18 pruned with magnitude-based strategy. **Left**: CIFAR-10. **Right**: CIFAR-100. We plot results of dense, overfitting, two swet-spot and underfitting models.

accuracy. Note that this increase is not caused by incomplete training of models, for models across all sparsities are able to converge to steady states (Figure 2).

The enhanced performance could be explained by the reduced capability of fitting random labels. Taking magnitude-based pruning results on CIFAR-100 with $\epsilon = 40\%$ as an example (the lower center plot in Figure 1): the rapid rise period of test accuracy (or the drastic decrease period of training accuracy) w.r.t. sparsity lies in the interval between about 96.48% and 98.56%. During this period, training accuracy on permuted samples is greatly decreased, while the reduction of accuracy on unpermuted samples is relatively milder (Figure 3), showing that sparsity primarily impacts on memorization of noisy labels before on the ability of learning generalized features.

Observation 2: *Larger permuted fraction of training data requires lower sparsities for interpolation, and higher sparsities for robustness.*

Here we illustrate how the memorization effect of sparse neural networks is influenced by dataset itself (Figure 1). As it is in the double descent phenomenon (Nakkiran et al., 2020), increasing the fraction of permuted samples shift the interpolation threshold towards models with larger capacity, which is to say, lower sparsities. On the other hand, in order to combat the side effects brought by the existence of heavier labels noise, more parameters in the network need to be pruned. Moreover, by comparing the ceiling value of test accuracy across a range of permuted fractions, we show that the depressed test performance of neural networks under high label-permuted fraction settings could not be simply recovered by sparsification. Though restricting model capacity hinders memorization, the existence of noisy labels damages test performance anyway.

Observation 3: *Higher pruning efficacy doesn't necessarily guarantee better label-noise-learning robustness. Magnitude-based approach enables models to withstand the most parameters to be removed without losing training performance; whereas gradient-based method attains top early-stopping test accuracy among almost all tasks.*

Here, we first demonstrate the impact made by different pruning strategies on model capacity. For illustration, we introduce the term of *pruning efficacy*: the largest sparsity a pruning strategy can reach without hurting training performance of neural networks, and regard pruning efficacy as a quantitative measure of the pruning impact on model capacity. As is shown in Figures 1 and 8, for experiments on all datasets, magnitude-based pruning possesses the most striking efficacy, which

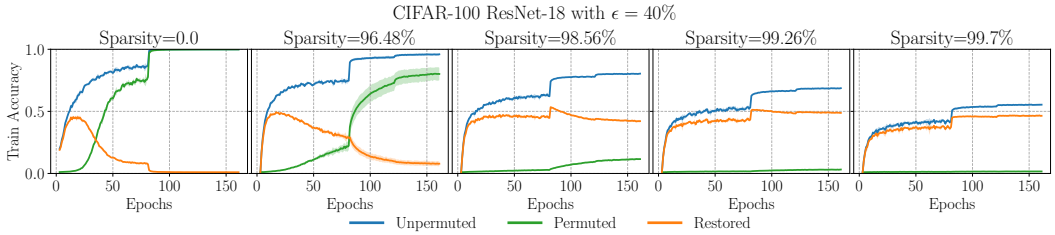


Figure 3: Memorization measured by train accuracy on unpermuted, permuted and restored samples. Models are ResNet-18 on CIFAR-100 with $\epsilon = 40\%$. We plot dense, overfitting, two swet-spot and underfitting results from left to the right.

preserves model capacity despite a large fraction of parameters is removed. And random pruning has the least efficacy, suggesting that the inherent structures might be partly disrupted when randomly disabling connections in a network.

However, the efficacy of pruning does not necessarily correlates with robustness of pruned networks. At high sparsities, the performance of gradient-based pruning consistently surpass magnitude-based pruning and random pruning, i.e., higher test accuracy at the best and the last epoch, or wider range of sparsities to hold robustness. Neither the ability to maintain training accuracy (high pruning efficacy) nor the resistance against losing parameters (low pruning efficacy) is adequate for explaining the superiority of gradient-based methods. Thus, expect for the ability to reduce parameters while minimize accuracy loss, we need to discuss properties of pruning strategies more comprehensively.

Here we'd like to propose an intuitive explanation for the different behaviors between magnitude-based pruning and gradient-based pruning. Motivated by previous works (Molchanov et al., 2017b; Lee et al., 2019), we use the change in loss ΔL_j to measure the impact of removing weights i on loss at pruning iteration j. We utilize the binary mask as an indicator of pruned weights with $\mathbf{m}_{i,j} = 1$ and $\mathbf{m}_{i,j+1} = 0$. And for simplicity, we denote the derivative of $L(\mathbf{w} \odot \mathbf{m}_j; \mathcal{D})$ with respect to its parameters $\mathbf{w} \odot \mathbf{m}_j$ as $g_j(\mathbf{w}; \mathcal{D})$. We phrase pruning as an optimization process and make a locally linear approximation near $\mathbf{m}_j$. Hence, based on Taylor expansion, the change in loss can be written as:

$$\Delta L_j = L\left(\mathbf{w} \odot \mathbf{m}_{j+1,}; \mathcal{D}\right) - L\left(\mathbf{w} \odot \mathbf{m}_j; \mathcal{D}\right) \approx g_j^T(\mathbf{w}; \mathcal{D})\left(\mathbf{w} \odot (\mathbf{m}_{j+1} - \mathbf{m}_j)\right) \tag{1}$$

Recall the pruning heuristics in Section 3.2, gradient-based method removes weights with the smallest $|g \odot \mathbf{w}|$, hence minimizing the loss change during pruning iterations. Given the favorable learning behavior where training loss starts to suffer as we discussed above, we may conjecture that smoothing the curve of training loss (or accuracy) is beneficial for robustness. During the prolonged stage of downgrade in training performance, networks gradually forget hard-to-generalize patterns, and the early-stopping performance gains. Therefore, the smoother the training curve is w.r.t sparsity, the wider range of sparsities that contributes to robust early-stopping performance can be. And a flatter accuracy-sparsity curve also makes it possible to search for the robust sparsities with a larger pruning ratio per iteration.

On the other hand, magnitude-based pruning removes parameters with small absolute magnitude, which have the minimal affect on objective, and keeps objective near zero at relatively high sparsities, therefore achieves remarkable pruning efficacy. Though such intuitive explanations are quite simple and straightforward, and do not cover complex cases like retraining with LTR, however, we can still see a coincidence between our analysis and experimental results, despite the diminished distinction between two methods with extreme permuted fraction of data (Figure 1).

5 Sparse Loss Landscape Analysis

We have demonstrated that pruning could hinder the memorization of label noise and lead to robust solutions at high sparsity. In this section, we empirically explore the possible causes for this phenomenon.

Previous works hypothesize that pruning encourages the optimizer to move towards flatter minima that benefit generalization, and this benefits do not depend on permanent parameter removal (Han

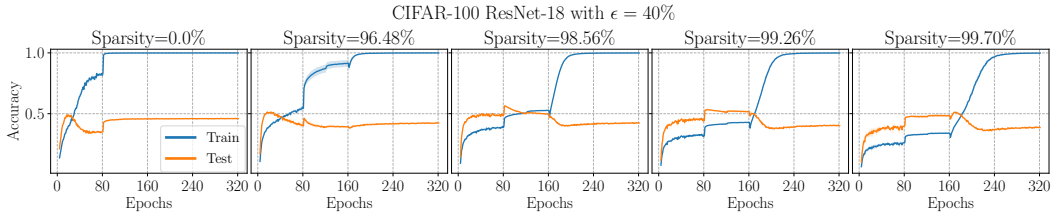
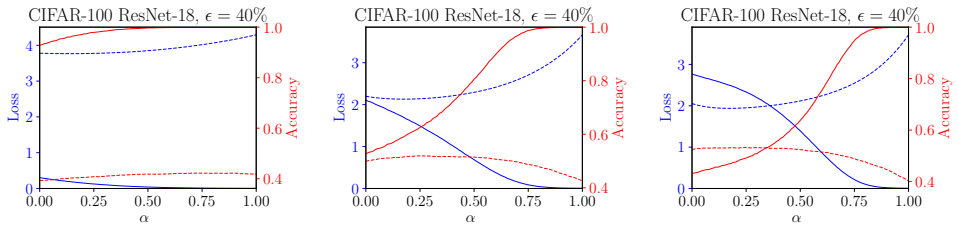


Figure 4: Accuracy as a function of epochs during sparse and re-dense training process. Models are ResNet-18 on CIFAR-100, with permuted fraction at 40%. Pruned weights are recovered at epoch 160, and trained for another 160 epochs with a fixed learning rate of 0.001. We present models at dense, overfitting, two swet-spot and underfitting sparsities from left to the right.



(a) Sparsity=96.48% (b) Sparsity=98.56% (c) Sparsity=99.26%

Figure 5: Linear interpolation plots. Models are ResNet-18 on CIFAR-100 with $\epsilon = 40\%$. $\alpha = 0$ corresponds to sparse solutions and $\alpha = 1$ corresponds to the re-dense solutions. The blue lines are loss curves and the red lines are accuracy curves; solid lines indicate training data set and dashed lines indicate testing data set. For re-dense models, sparsity is measured before recovering weights.

et al., 2017; Bartoldson et al., 2020). Here, we'd like to investigate whether flatness can explain the robustness to label noise at high sparsities, and whether the robustness could still be maintained if we bring back the pruned connections to networks.

Re-dense training escapes highly sparse solutions. We apply the re-dense training approach as introduced in Section 3.3. If the optimizer reaches a flat basin of local minima during sparse training, we may suspect that a small learning rate in the re-dense training stage will continually attract optimizer around this basin, and the final re-dense solutions will have comparable generalization performance to the sparse ones. However, as is shown in Figure 4, solutions at *high sparsities* are not stable in dense subspace. Once the sparsity constraints are removed, the objective will escape from pruned solutions and overfit to label noise severely.

High Sparsity reshapes loss landscape and creates local minima. Moreover, with linear interpolation of loss function, we find a monotonically decreasing path from the high-loss point to low-loss point (Figures 5 and 14). The existence of such path demonstrates that these highly sparse solutions are no longer minimizers in high dimensions, thus allowing for the escape phenomenon during re-dense training process. Moreover, the final solutions of re-dense training does not possess good generalization behavior, and have higher sharpness than the original dense models trained from scratch (see the 1-D visualization of solutions at various sparsities in Figure 6). Such phenomenon provides evidence *against* that highly sparse solutions stick around flat basins of minimizers.

Hypothesis for label-noise robustness of pruned networks. Given the above findings, we propose the following hypothesis: *High sparsities obstruct the objective decreasing path, and discourage optimizers to move away from initialization, which lessens model vulnerability to label noise.* As is known that neural networks need to stray far from initialization to memorize noisy labels (Li et al., 2020b; Stephenson et al., 2021), we may suspect that the distance from initialization could explain the robustness behavior of highly sparse networks to a certain extent.

Optimizers are attracted by minimizers near initialization in sparse subspace. We measure the ℓ_2 distance from initialization with all trainable parameters from convolutional and linear layers. In order to illustrate the relationship between the ℓ_2 distance and model performance, we plot both

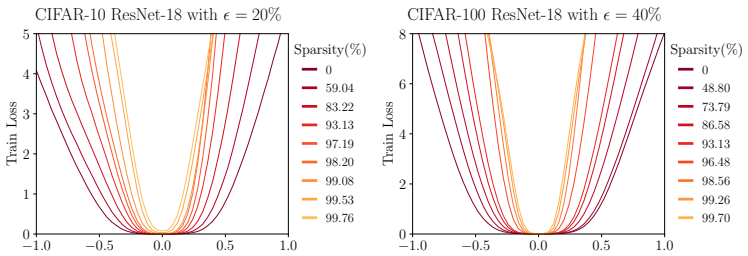


Figure 6: The 1-D loss visualization of minima found by re-dense training using filter normalization. **Left**: ResNet-18 on CIFAR-10 with $\epsilon = 20\%$. **Right**: ResNet-18 for CIFAR100 with $\epsilon = 40\%$. The sparsity of particular model is measured before the re-dense step. The higher sparsity of the original pruned network has, the sharper minima it will converge to after re-dense training.

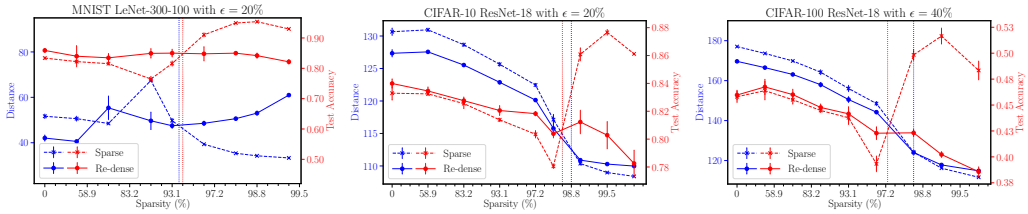


Figure 7: ℓ_2 distance from initialization and test accuracy as functions of sparsities. **Left**: LeNet-300-100 on MNIST with $\epsilon = 20\%$. **Middle**: ResNet-18 on CIFAR-10 with $\epsilon = 20\%$. **Right**: ResNet-18 on CIFAR-100 with $\epsilon = 40\%$. The blue lines are ℓ_2 distance curves and the red lines are accuracy curves; solid lines are results for re-dense solutions and dashed lines are for sparse solutions. The vertical lines indicate where the curves of sparse and re-dense results come to cross, and signs of their relative difference shift.

the distance and test accuracy against sparsity (Figure 7). Note that when increasing sparsity, both the "sparse distance" (ℓ_2 distance from initialization of sparse networks) and "re-dense distance" (ℓ_2 distance of re-dense networks) decline continuously. This phenomenon reveals that sparsity restricts the movement of optimizers, and traps them around sharp minimizers near initialization, which would be normally skipped when training dense networks.

Correlations between ℓ_2 distance and robustness. Surprisingly, the "sparse distance" with repect to sparsities of LeNet-300-100 models also exhibits a *double-descent-like* trend (seen in Figure 7, the left); furthermore, the curves of distance and accuracy are almost mirror images of each other for both sparse and re-dense networks, which suggests a strong correlation between parameter distance and model robustness. With regard to more complex architectures (ResNet-18 with convolution and batch-normalization layers) and more difficult datasets (CIFAR-10 and CIFAR-100), such correlation is not obvious in their absolute value. However, if we focus on the relative difference between sparse and re-dense results w.r.t. the same sparsity, we can confirm an similar conclusion that staying closer to initialization guarantees better robustness: at low sparsities, sparse solutions are located farther from initialization than re-dense ones, and presents an inferior performance, while at high sparsities, sparse minimizers stay closer to initial points and manifest robustness. This observation supports our proposed hypothesis, and is consistent with prior theoretical studies (Li et al., 2020b).

6 Conclusion

In this paper, we reassess some common beliefs concerning the generalization properties of sparse networks and illustrate the inapplicability of these viewpoints under label noise at high sparsities. Instead, our proposed hypothesis that highly sparse solutions are stuck near initialization thus stay invulnerable to noisy labels, correlates with empirical findings, and accounts for the robustness at high sparsities to a certain extent. We provide some insight into the optimization dynamics and memorization capability of sparse regimes, which we hope will guide progress towards more robust training and pruning algorithms for deep learning under label noise.

REFERENCES

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pp. 233–242. PMLR, 2017.

Samaneh Azadi, Jiashi Feng, Stefanie Jegelka, and Trevor Darrell. Auxiliary image regularization for deep cnns with noisy labels. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

Brian Bartoldson, Ari S. Morcos, Adrian Barbu, and Gordon Erlebacher. The generalization-stability tradeoff in neural network pruning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019. ISSN 0027-8424. doi: 10.1073/pnas.1903070116. URL https://www.pnas.org/content/116/32/15849.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? *arXiv preprint arXiv:2003.03033*, 2020.

Xiangyu Chang, Yingcong Li, Samet Oymak, and Christos Thrampoulidis. Provable benefits of overparameterization in model compression: From double descent to pruning neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6974–6983, 2021.

Utku Evci, Fabian Pedregosa, Aidan Gomez, and Erich Elsen. The difficulty of training sparse neural networks, 2020.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

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

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Pruning neural networks at initialization: Why are we missing the mark? In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Trevor Gale, Erich Elsen, and Sara Hooker. The state of sparsity in deep neural networks. *CoRR*, abs/1902.09574, 2019.

Purvi Goel and Li Chen. On the robustness of monte carlo dropout trained with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2219–2228, 2021.

Ian J. Goodfellow and Oriol Vinyals. Qualitatively characterizing neural network optimization problems. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6544.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in Neural Information Processing Systems*, 28, 2015.

Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

Song Han, Jeff Pool, Sharan Narang, Huizi Mao, Enhao Gong, Shijian Tang, Erich Elsen, Peter Vajda, Manohar Paluri, John Tran, Bryan Catanzaro, and William J. Dally. 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.

Babak Hassibi and David G Stork. Second order derivatives for network pruning: optimal brain surgeon. In *Proceedings of the 5th International Conference on Neural Information Processing Systems*, pp. 164–171, 1992.

Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *arXiv preprint arXiv:2102.00554*, 2021.

Sara Hooker, Aaron Courville, Gregory Clark, Yann Dauphin, and Andrea Frome. What do compressed deep neural networks forget? *arXiv preprint arXiv:1911.05248*, 2019.

Wei Hu, Zhiyuan Li, and Dingli Yu. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pp. 2304–2313. PMLR, 2018.

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. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=H1oyRlYgg`.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

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

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

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Snip: single-shot network pruning based on connection sensitivity. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Namhoon Lee, Thalaiyasingam Ajanthan, Stephen Gould, and Philip H. S. Torr. A signal propagation perspective for pruning neural networks at initialization. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 6391–6401, 2018.

Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Learning to learn from noisy labeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5051–5059, 2019.

Junnan Li, Richard Socher, and Steven C. H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020a. URL https://openreview.net/forum?id=HJgExaVtwr.

Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International conference on artificial intelligence and statistics*, pp. 4313–4324. PMLR, 2020b.

Dachao Lin, Ruoyu Sun, and Zhihua Zhang. On the landscape of one-hidden-layer sparse networks and beyond, 2021.

Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pp. 2736–2744, 2017.

Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Christos Louizos, Max Welling, and Diederik P. Kingma. Learning sparse neural networks through l_0 regularization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=H1Y8hhg0b.

Dmitry Molchanov, Arsenii Ashukha, and Dmitry P. Vetrov. Variational dropout sparsifies deep neural networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2498–2507. PMLR, 2017a. URL http://proceedings.mlr.press/v70/molchanov17a.html.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017b. URL https://openreview.net/forum?id=SJGCiw5gl.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

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.

Curtis Northcutt, Lu Jiang, and Isaac Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021a.

Curtis G. Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. *CoRR*, abs/2103.14749, 2021b.

Vaishaal Shankar, Rebecca Roelofs, Horia Mania, Alex Fang, Benjamin Recht, and Ludwig Schmidt. Evaluating machine accuracy on ImageNet. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8634–8644. PMLR, 13–18 Jul 2020.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *arXiv preprint arXiv:2007.08199*, 2020.

Cory Stephenson, Suchismita Padhy, Abhinav Ganesh, Yue Hui, Hanlin Tang, and SueYeon Chung. On the geometry of generalization and memorization in deep neural networks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Ryutaro Tanno, Ardavan Saeedi, Swami Sankaranarayanan, Daniel C Alexander, and Nathan Silberman. Learning from noisy labels by regularized estimation of annotator confusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11244–11253, 2019.

Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. Robust early-learning: Hindering the memorization of noisy labels. In *International Conference on Learning Representations*, 2020.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Sy8gdB9xx.

Zhanxing Zhu, Jingfeng Wu, Bing Yu, Lei Wu, and Jinwen Ma. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7654–7663. PMLR, 2019. URL http://proceedings.mlr.press/v97/zhu19e.html.

A APPENDIX

A.1 EXPERIMENTAL DETAILS

We adopt standard implementations of LeNet-300-100 from OpenLTH[1]. LeNet-300-100 is a fully-connected network with 300 units in the first layer and 100 units in the second hidden layer, and ReLU activations.

Network of ResNet-18 is a modified version of PyTorch model. To adapt ResNet-18 for CIFAR-10 and CIFAR-100, the first convolutional layer is equipped with filter of size 3x3 and the max-pooling layer that follows has been eliminated. CIFAR-10 and CIFAR-100 are augmented with per-channel normalization, randomly horizontal flipping , and randomly shifting by up to four pixels in any direction.

In pruning experiments, for LeNet-300-100, we consider all weights from linear layers except for the last layer as prunable parameters; for ResNet-18, all weights from convolutional and linear layers are set as prunable. We do not prune biases nor the batch normalization parameters. For convolutional and linear layers, the weights are initialized with Kaiming normal strategy and biases are initialized to be zero.

We run all our experiments on single 2080Ti GPU with CUDA 10.1, and provide the training hyper-parameters used in our experiments as follows. Our code is available in the supplementary material.

Network	Dataset	Epochs	Batch	Opt.	Mom.	LR	LR Drop	Drop Factor	LR(re-dense)	Weight Decay	Rewind Iter
LeNet-300-100	MNIST	200	128	SGD	—	0.1	—	—	0.1	—	0
ResNet-18	CIFAR-10	160	128	SGD	0.9	0.1	80, 120	0.1	0.001	1e-4	1000
ResNet-18	CIFAR-100	160	128	SGD	0.9	0.1	80, 120	0.1	0.001	1e-4	1000

A.2 ADDITIONAL EXPERIMENT RESULTS AND DISCUSSION

Here, we will present additional results that are not included in the main body for page limit.

A.2.1 MNIST RESULTS

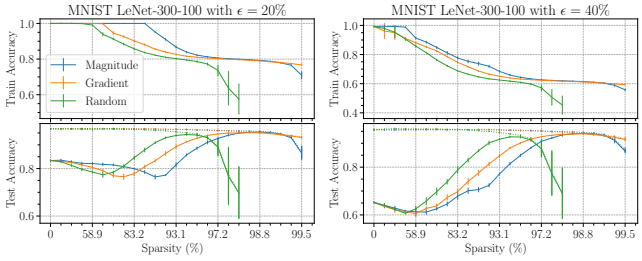


Figure 8: Double descend phenomenon in sparse regimes for LeNet-300-100 on MNIST with three pruning strategies and varying permuted fraction ϵ.

[1] https://github.com/facebookresearch/open_lth

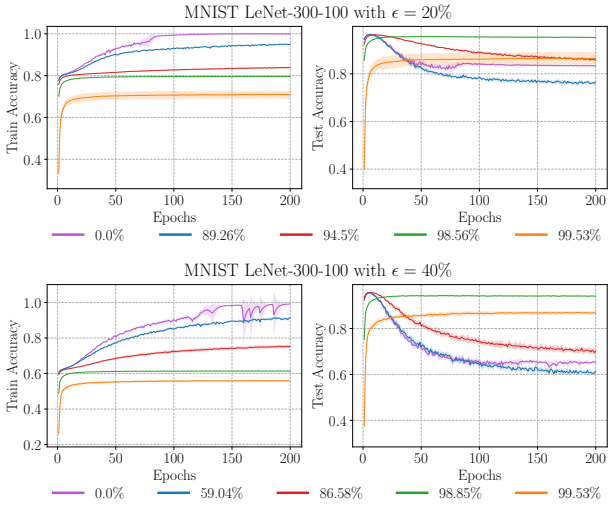


Figure 9: Training dynamics w.r.t. epochs at five sparsities across different permuted fractions ϵ. Models are LeNet-300-100 for MNIST pruned with magnitude-based strategy.

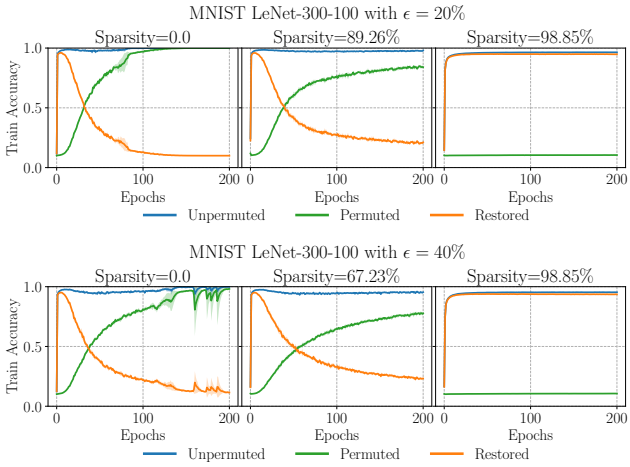


Figure 10: Memorization measured by train accuracy on unpermuted, permuted and restored samples. Models are LeNet-300-100 on MNIST with different permuted fraction. We plot dense, overfitting, swet-spot results.

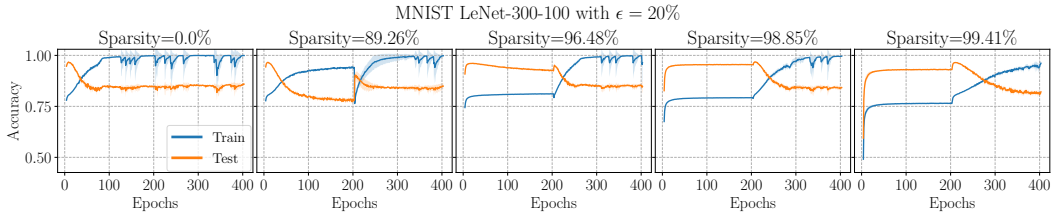


Figure 11: Accuracy curve of the sparse and re-dense training process. We recover pruned weights at epoch 200, and training them from value of zero for another 200 epochs using the last learning rate of sparse training, which is 0.1 for LeNet-300-100.

A.2.2 CIFAR-10 RESULTS

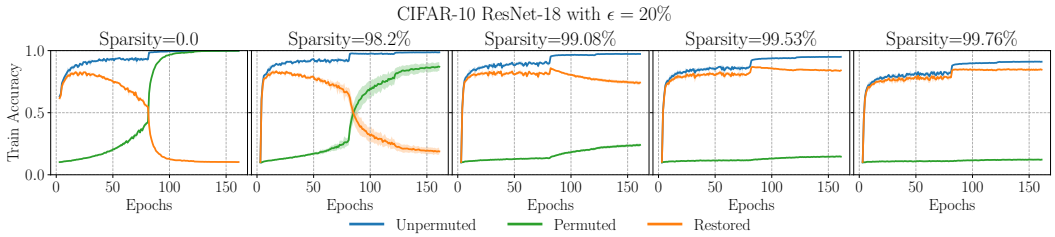


Figure 12: Memorization measured by train accuracy on unpermuted, permuted and restored samples. Models are ResNet-18 on CIFAR-10 with $\epsilon = 20\%$. We plot dense, overfitting, two swet-spot and underfitting results from left to the right. Memorization capability of neural networks is damaged as pruning.

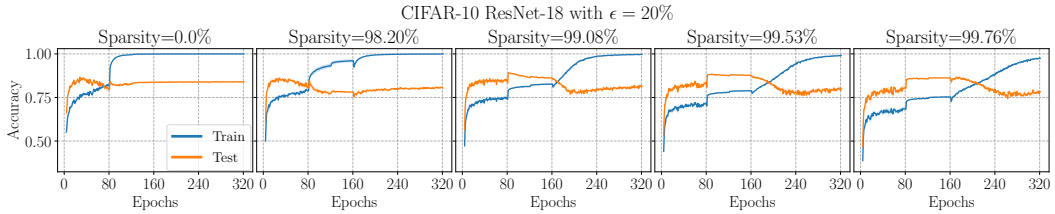


Figure 13: Accuracy curve of the sparse and re-dense training process. We recover pruned weights at epoch 160, and training them from value of zero for another 160 epochs using the last learning rate of sparse training, which is 0.001 for ResNet-18.

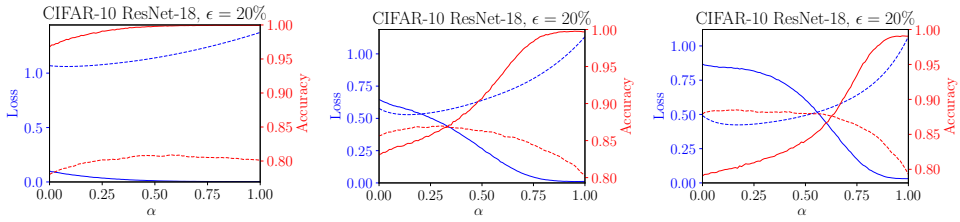


Figure 14: Linear interpolation plots. Models are ResNet-18 on CIFAR-10 with $\epsilon = 20\%$. $\alpha = 0$ corresponds to sparse solutions and $\alpha = 1$ corresponds to the re-dense solutions. The blue lines areloss curves and the red lines are accuracy curves; solid lines indicate training data set and dashed-lines indicate testing data set.

A.2.3 ADDITIONAL DISCUSSION

We'd like to further discuss why the double descent phenomenon is rarely exhibited in existing pruning literature. Several possible explanations might account for its imperception: (1) not enough points are reported in the accuracy-sparsity tradeoff curve; (2) performance loss is offset by noise injection regularization effect brought by pruning; (3) retraining techniques like finetuning keep the network trapped near initial pruned solutions. Nevertheless, under label noise settings, we can amplify the impact of reduced capacity on model performance brought about by sparsity, and reconcile the conventional understanding and the modern practice of network pruning.