On Neural Network Pruning's Effect on Generalization

Anonymous Author(s)
Affiliation
Address
email

Abstract

Practitioners frequently observe that pruning improves model generalization. A long-standing hypothesis attributes such improvement to model size reduction. However, recent studies on over-parameterization characterize a new model size regime, in which larger models achieve better generalization. A contradiction arises when pruning is applied to over-parameterized models – while theory predicts that reducing size harms generalization, pruning nonetheless improves it. Motivated by such a contradiction, we re-examine pruning's effect on generalization empirically. We demonstrate that pruning's generalization-improving effect cannot be fully accounted for by weight removal. Instead, we find that pruning can lead to better optimization, improving training loss. We find that pruning can also lead to stronger regularization, mitigating the harmful effect of noisy examples. Pruning extends model training time and reduces model size, which improves optimization and strengthens regularization respectively. We empirically demonstrate that both factors are essential to explaining pruning's benefits to generalization fully.

1 Introduction

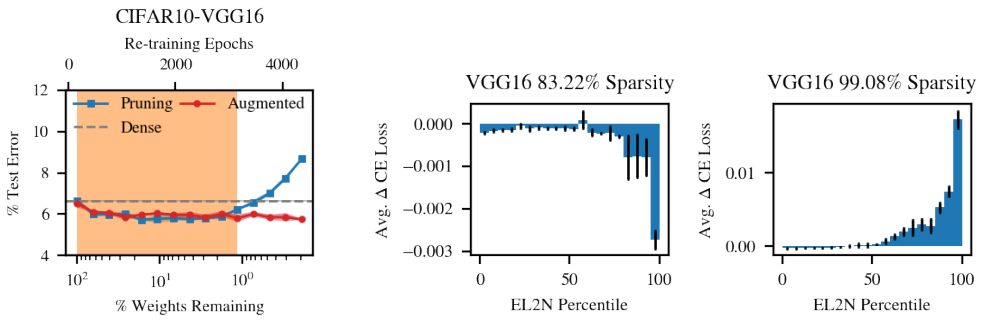
Neural network pruning techniques remove unnecessary weights to reduce the memory footprint and computational requirements of a model. In practice, practitioners can remove a large fraction (often 80-90%) of weights without impacting generalization, measured by test error [25, 16, 10]. While recent pruning research [16, 10, 37, 5, 24, 32, 9, 50, 3, 27, 47, 38, 28, 29, 41, 39, 7, 51, 43] focuses on reducing model footprint, improving generalization has been a core design objective for earlier works on pruning [25, 19]; recent pruning literatures also frequently note that it improves generalization [15, 10].

How does pruning improve generalization? An enduring hypothesis claims that pruning benefits generalization by reducing the number of weights, and thus model size.[1] We refer to this hypothesis as the *size-reduction hypothesis*. However, much has evolved since the proposal of this hypothesis.

Firstly, pruning algorithms grew increasingly complex. State-of-the-art pruning algorithm [37] prunes weights *iteratively*. Each iteration consists of *weights removal*, where the algorithm zeros out a set of unimportant weights, followed by *retraining*, where the algorithm replays the original learning rate schedule. Across multiple iterations, such a retraining scheme effectively adopts a cyclic learning rate schedule, which may positively influence generalization [42].

[1] This hypothesis can be traced back to seminal works in pruning such as Optimal Brain Surgeon [19]: "without such weight elimination, overfitting problems and thus poor generalization will result." And again in pioneering work [16] applying pruning to deep neural network models: "we believe this accuracy improvement is due to pruning finding the right capacity of the network and hence reducing overfitting." Because the latter hypothesis leaves the definition of network capacity unspecified, we examine an instantiation of it, measuring network capacity with model size.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.



(a) (b)

Figure 1: (a) Retraining the model exactly as done in the pruning algorithm, but without removing any weights produces models with similar generalization as pruning to the shaded range of weights remaining. The x-axis for pruning is shown below whereas the x-axis for the augmented pruning algorithm is shown above. (b) Left: We find that pruning can improve generalization by decreasing training loss, with a particular decrease on examples with a high influence on generalization. Right: We find that pruning can also improve generalization by increasing training loss, also with a particular increase on examples with a high influence on generalization.

Secondly, the classical bias-variance trade-off implies that when the model size decreases, generalization may improve. The size-reduction hypothesis is based on pruning's effect in this bias-variance trade-off regime, predicting that decreasing model size may improve generalization. However, recent work on the *double descent* phenomenon [4, 34] reveals that beyond this classical bias-variance trade-off regime lies an over-parameterized regime where models are large enough to achieve near-zero training error. In this regime, bigger models achieve better generalization [4, 34]. Practitioners often apply modern pruning algorithms to models in this over-parameterized regime; in this regime, while theory predicts that reducing model size should harm generalization, pruning nevertheless improves it. This contradiction casts doubt on the size-reduction hypothesis.

The evolution of pruning algorithms and the renewed theoretical understanding of the role of over-parameterization motivate us to re-examine pruning's generalization-improving effect empirically.

A New Direction. Our first result is that the size-reduction hypothesis does not fully explain pruning's effect on generalization. Figure 1a compares the generalization of a family of sparse models of different sizes generated by a state-of-the-art pruning algorithm [37] and the generalization of a family of models generated by the same algorithm, except modified to no longer remove weights. The interventions remaining for the augmented pruning algorithm to generate different models are the number of training epochs and the cyclic learning rate schedule.

The results show that for any sparse VGG16 model generated by the pruning algorithm with more than 1% of its original weights, its generalization is indistinguishable from that of the model generated by the augmented algorithm that does not remove weights but is trained for the same number of epochs.[2] These results demonstrate that the size-reduction hypothesis does not provide a complete account of pruning's effect on generalization.

Our Approach. We instead develop an alternative explanation for pruning's beneficial impact on generalization through an analysis of pruning's effect on each training example as measured by the change in its training loss before and after pruning. We further evaluate the extent to which a training example influences generalization by using an approximate measure – the L2 distance between the predicted probabilities and one hot labels early in training, called the EL2N score, developed by Paul et al. [36]. With our analysis, we uncover the following results.

Better Optimization. The left plot in Figure 1b presents the change in training loss after pruning for the training examples as grouped by percentiles of their EL2N scores for a VGG16 on CIFAR-10. The results show that pruning improves generalization by decreasing training loss, with a particular

[2] In Appendix C we show this phenomenon for 4 additional benchmarks, and other pruning algorithms.

emphasis on those examples with a high influence on generalization (i.e, high EL2N score). We refer to pruning's ability to reduce training loss as *better optimization*.

Stronger Regularization. The right plot in Figure 1b also presents the change in training loss of examples as grouped by percentiles of their EL2N scores, for the same benchmark but with the pruning algorithm configured to produce a smaller, sparse model. In this setting, the results show that pruning can also improve generalization by increasing training loss, also with a similar, particular emphasis on highly influential examples.

While it is perhaps counter-intuitive that increasing training loss, particularly on influential examples, may improve generalization, we hypothesize that it is due to an observation that Paul et al. [36] made: in image datasets, highly influential training examples include a small fraction that have ambiguous or erroneous labels, training on which impairs generalization. Therefore, it is instead better for the model to not fit them.

We examine this hypothesis by introducing random label noise into the training data. We find that for models with the best generalization, pruning increases training loss on these noisy examples, while still preserving the training loss of examples from the original training data. The result is an overall improvement in generalization for the sparse model. We use the term *stronger regularization* to describe pruning's effect to avoid fitting noisy examples.

Implications. It has been a long-standing hypothesis that pruning improves generalization by reducing the number of weights (via minimum description length principle [14] or Vapnik–Chervonenkis theory [46]). Our empirical observations suggest that the theory for pruning should focus on the two effects of modern pruning algorithms: better optimization, and stronger regularization. To understand the implication of pruning's regularization effect on generalization, our results suggest that it is important to quantify the heterogeneous effect of pruning across different dataset subgroups.

Contributions

1. We find that pruning leads to *better optimization*, improving model training loss.
2. We find that pruning also leads to *stronger regularization*, mitigating the detrimental effect of noisy examples to generalization.
3. We demonstrate that our deconstruction of pruning's effect into optimization and regularization cannot be further simplified — alternative explanations such as increased training time and reduced model size only partially explain pruning's generalization-improving effect.

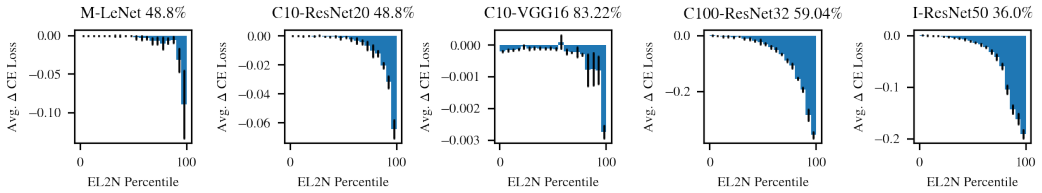
2 Preliminaries

We study *Iterative Magnitude Pruning (IMP) with learning-rate rewinding* [10, 11]. This algorithm consists of four steps: (1) Train a neural network to completion. (2) Remove a specified fraction of the smallest-magnitude weights. (3) Reset the learning rate to its value in an earlier epoch t [11] and (4) Repeat steps 1-3 iteratively until the model reaches a desired overall level of sparsity.

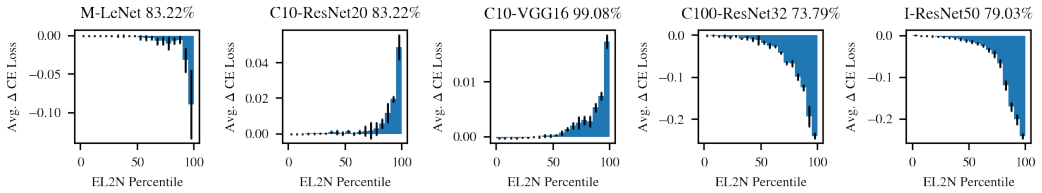
Terminology. We refer to models without any weights removed as *dense models*; in contrast, we refer to models with weights removed by pruning as *sparse models*. We use the term *optimal sparse model* to refer to the model with the sparsity that achieves the best validation error. We call such sparsity the *optimal sparsity*. We define *generalization* as classification error on the test set.

Experimental Methodology. We use standard architectures: LeNet [26], VGG-16 [40], ResNet-20, ResNet-32 and ResNet-50 [20], and train on benchmarks (MNIST, CIFAR-10, CIFAR-100, ImageNet) using standard hyperparameter settings, following Frankle and Carbin [10], Frankle et al. [11], Wang et al. [48]. Also following [10, 11], we set the t hyperparameter of IMP to $t = 0$ for LeNet-MNIST benchmark and $t = 10$ for the others. Appendix A presents our hyperparameter settings in full detail.

We report all results by running the same experiment three times with distinct random seeds. We conclude that any real-valued results of two experiments differ if and only if the mean difference between three independent runs of respective experiments is at least one standard deviation away from zero, otherwise we say that the results *match*. We run all experiments on TPUs with PyTorch [35] using the OpenLTH library [10].



(a) At sparsity level achieving the best generalization



(b) At highest sparsity level with generalization exceeding that of the dense counterpart

Figure 2: Change of average training loss on subgroups of examples with distinct EL2N score percentile range. A negative value indicates that loss is decreased by pruning.

3 Generalization Improvement from Better Optimization

It is well-appreciated that not all training examples have the same effect on training and generalization (see a vast literature on data pruning [45, 36], fairness [6, 18], and many more). Recent work by Hooker et al. [21] illuminates that pruning has varying effects on subgroups of test examples.

In this section, we identify which subgroups of training examples are more affected by pruning, and the special properties of these subgroups: it is the subgroups that are most difficult for the dense model to learn. We then connect these examples to effects on generalization via leave-subgroup-out retraining and empirically show that the subgroups of training examples that are most affected by pruning also tend to have the highest impact on generalization.

Method. Paul et al. [36] propose a metric (EL2N) that scores individual training examples based on their error early in training. Paul et al. [36] shows that the EL2N score correlates with the difficulty of individual examples, which they measure with how frequently the model misclassifies the example during training [45, 36]. For a range of models, Paul et al. [36] present empirical evidence that the EL2N score of a training example correlates with its influence on generalization. The authors further draw theoretical connections between EL2N score and influence on generalization We present details of EL2N score calculation in Appendix A.3.

To measure the change in (cross entropy) loss on training examples due to pruning, we partition the training set into M subgroups, each with a different range of EL2N scores. For each subgroup, we then compute the average change in training loss that pruning causes on examples in the subgroup. We compute the change in training loss by subtracting the training loss of the dense model from that of the sparse model. A negative value indicates that pruning improves the subgroup training loss. We measure training loss at two sparsities of interest: the sparsity that achieves the best generalization, and the highest sparsity that still attains better generalization than the dense model. In Appendix B, we present pruning's effect on subgroup training loss at a wider range of sparsities. We pick $M = 20$ as it is the largest value of M that enables us to clearly present per-subgroup training loss change.

Results. We visualize pruning's effect on subgroup training loss in Figure 2. Our results consistently show that pruning most significantly affects examples with the highest range of EL2N scores. The average magnitude of pruning's effect on examples with top 20% EL2N scores is 3.5-71x times that on examples with bottom 80% EL2N scores.

Figure 2a shows that at the sparsity that achieves the best generalization, pruning's effect on all subgroups of the training dataset is to reduce their training loss. This confirms our expectation that pruning leads to better optimization, which may improve generalization.

The largest training loss reduction happens to the examples with high EL2N scores, which have been linked to high importance on generalization [36]. In Appendix D, we empirically validate the importance of pruning-affected examples to generalization: excluding 20% of pruning-affected examples hurts generalization more than excluding a random subset of the same size.

4 Generalization Improvement from Stronger Regularization

Pruning to the highest sparsity that improves generalization (Figure 2b) displays a similar trend as pruning to the optimal sparsity (Figure 2a) – generalization improves with an overall reduction in training loss, except for CIFAR10 dataset, where pruning improves generalization while at the same time causing the sparse models to avoid fitting all subgroups of examples. Specifically, for CIFAR10-ResNet20 at 83.22% sparsity (Figure 2b, second from left) and CIFAR10-VGG16 at 99.08% sparsity (Figure 2b, third from left), the change in training loss on most subgroups is positive, indicating worse training loss. While pruning always increases training loss when it removes a large enough fraction of model weights, it is surprising to observe the accompanied generalization improvement.

The second and third plots of Figure 2b show that the loss increase is especially pronounced on a particular subgroup of examples with high EL2N scores, that are slow to learn and have an outsized effect on generalization. Paul et al. [36] observed that although examples with high EL2N scores are predominately beneficial for model generalization, there exist a small fraction of noisy examples, such as ones with ambiguous or erroneous labels, that have high EL2N scores and exert a significant influence on model generalization, albeit in a harmful way.[3]

Here we zoom in on the effect of pruning on noisy examples. By injecting random label noise, we have control over which examples are noisy.

Method. We inject $p\%$ random label noise by selecting $p\%$ examples uniformly at random and changing the label of each example to one of the other labels in the dataset sampled uniformly at random. We refer to the data that was not affected as *clean data*. We sample and fix the random labels before each independent run of an experiment.

We report the following results for models trained on datasets with random label noise: (1) pruning's impact on generalization versus sparsity (Figure 3); (2) pruning's impact on training loss for the clean data and random label data(Figure 4);[4] (3) pruning's impact on training loss for subgroups of examples, each with a different range of EL2N scores (Figure 5).

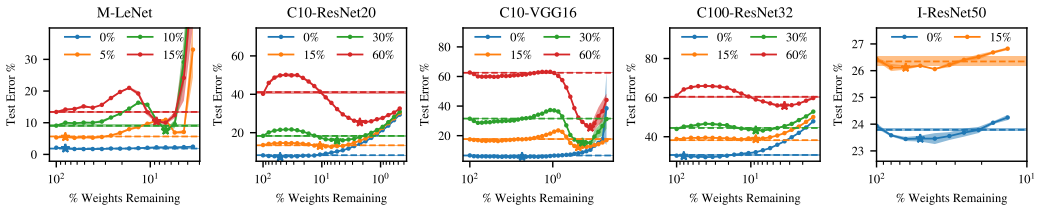


Figure 3: Pruning improves generalization in the presence of substantive random label noise. Generalization of the sparse models are shown as a function of percentage weights remaining. Generalization of the dense models are shown using horizontal dashed lines. Legend shows the label noise level.

Generalization results. As seen in Figure 3, in the presence of random label noise, the generalization of the optimal sparse models is better than that of the dense models on 11 out of 13 benchmarks and matches that of the dense models on the rest of the benchmarks. Our results provide evidence that pruning to the optimal sparsity improves generalization in the presence of random label noise.

The extent of pruning's generalization improvement grows as the portion of random label noise increases. For example, on the CIFAR10-VGG16 benchmark, pruning to the optimal sparsity improves generalization by 5.2%, 16.1% and 34.6% on 15%, 30% and 60% label noise.

[3] We present leave-subgroup-out retraining experiments for these examples in Appendix G.

[4] Numerical results available in Table 9 in Appendix M

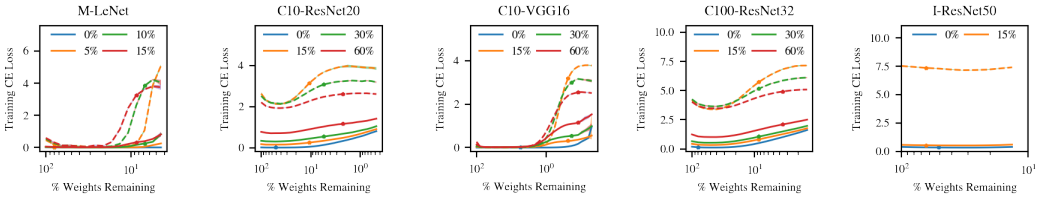


Figure 4: Pruning to the optimal sparsity increases training loss especially to noisy examples. Training loss on the clean and noisy data are shown using dashed and solid lines respectively, as functions of % weights remaining. We mark the optimal sparsity level with circle. Legend shows noise level.

Clean versus noisy data. Figure 4 shows that pruning from a dense model to a sparse one, the average loss on the noisy examples is initially *reduced* (for example, training loss on noisy data reduces when pruning the CIFAR10-ResNet20 model to *more than* 10% weighs remaining), suggesting better optimization. As the fraction of weights remaining drops further, pruning leads to stronger regularization – the loss on noisy data starts to go up: the gap between the average training loss of clean versus noisy examples widens up (for example, training loss on noisy data increases when pruning the CIFAR10-ResNet20 model to *less than* 10% weighs remaining), and eventually diminishes as the sparsity level increases further. In particular, for a range of sparsities where generalization improves over the dense model, the sparse model fits clean examples significantly better than those with randomized labels. Taking the dense VGG16 model trained on the CIFAR10 dataset with 15% - 60% label noise as an example, we see that this model incurs 0.03 - 0.13 higher training loss on the partition with injected random label noise than on the partition without. Pruning to the optimal level of sparsity expands this gap to 1.38 - 2.97.

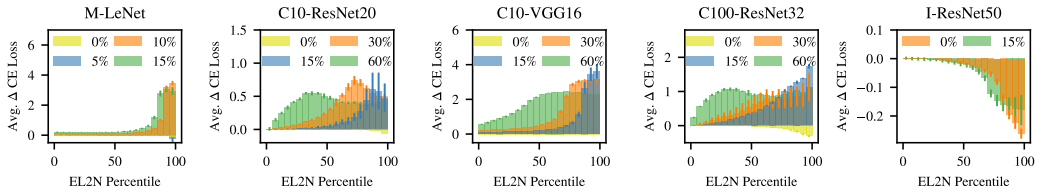


Figure 5: Pruning to the optimal sparsity level often reduces training loss to highly influential examples in the presence of substantive random label noise (except for ImageNet-ResNet50). Change in average cross entropy loss due to pruning to the optimal sparsity for subgroup of examples with distinct EL2N score percentile range is shown. Legend indicates label noise level. For MNIST-LeNet benchmark with 5% label noise, best generalization is achieved with the dense model.

Change in loss by subgroups. Similarly, as in Figure 2, we report the change in loss on subgroups of examples with different EL2N scores, but this time in the presence of label noise. As shown in Figure 5, pruning to the optimal sparsity *increases* the loss of high EL2N examples, contrary to what we observe in Figure 2 when training on clean data only. Note that random label examples have high EL2N scores [36]. Intuitively, noisy examples can significantly harm generalization, and a model that increases the training loss on noisy examples may improve generalization. In Appendix E, we empirically validate that excluding examples misclassified by the sparse models from the training dataset improves dense model generalization similar to pruning.

Conclusion. In this section, using random label noise, we enhance the two effects of pruning: better optimization and stronger regularization. At relatively low pruning levels, regularization effects are relatively small, and better optimization is observed via reduced loss of slow-to-learn (high EL2N score) examples. As the pruning level increases, regularization effects strengthen, as seen from the increase in training loss on noisy examples. It is worth emphasizing that across all sparsities we observe a *highly nonuniform effect of pruning-induced regularization* across EL2N score subgroups on both standard datasets and datasets with random label noise.

Crucially, the optimal optimization–regularization trade-off depends on the noise levels in the dataset. On datasets with few noisy examples, Figure 2 show that at the optimal sparsity, better optimization and negligible regularization is the key to good generalization, as indicated by reduced training loss. On datasets with injected random label noise, the optimal sparsity is the one at which pruning-induced regularization effects increase noisy data loss.

5 Isolating the Effects of Extending Training Time and Reducing Model Size

We have identifed two key components in pruning's effect on generalization: better optimization, and stronger regularization. These two components intersect with two components of the pruning algorithm itself: increased training time and weight reduction, respectively. In this section we study these two components in isolation, concluding that neither of them alone can explain the effect of pruning on generalization across all sparsities. In Appendix K, we also evaluate, via ablation studies, the individual contribution of design choices of pruning algorithm itself to generalization.

5.1 Extended Training Time

In sufficiently overparameterized models, Nakkiran et al. [34] observed that extending the training time results in smaller training *and* test error simultaneously. Pruning also effectively extends the training time. It is natural to ask what happens to generalization if the weight removal part is removed from pruning, and only the retraining part remains. To answer, we train the dense models exactly as done in the pruning algorithm, for the same number of gradient steps and learning rate schedule, but without removing any weights. The comparison between the generalization of dense models we train with the said dense training algorithm and the generalization of sparse models we obtain with pruning shows that longer training time alone fails to explain pruning's effect on generalization.

Method. We study the effect of training dense models for extra gradient steps on generalization, without removing any weights. When training the dense models, we use the same learning rate schedule as pruning, replaying the original learning rate schedule designed for the dense model multiple times, effectively adopting a cyclic learning rate schedule. We refer to this training algorithm as *extended dense training*. We report and compare the test errors of models that we obtain with pruning and extended dense training algorithm.

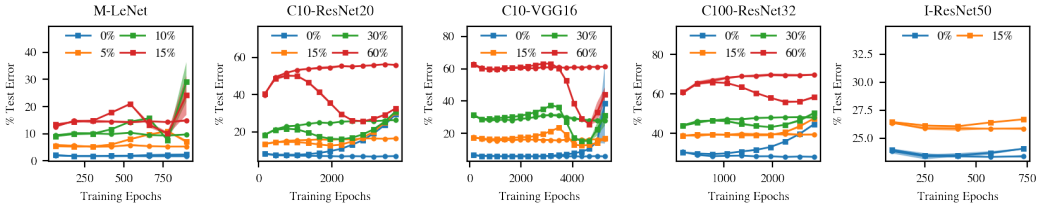


Figure 6: Extended dense training versus pruning. Test errors of the dense and sparse models are shown with circle and square markers, respectively. Legend shows random label noise level.

Results. We plot test errors of models that we obtain using extended dense training and pruning as a function of training epochs in Figure 6.[5] Extended dense training and pruning have similar generalization in the standard dataset regime, where pruning improves optimization at optimal sparsity. Indeed, in Appendix. M, we show that longer training time leads to better optimization. However, extended dense training underperforms pruning on 9 out of the 13 benchmarks that include random label noise. On LeNet, ResNet20, VGG16, ResNet32 and ResNet50 benchmarks with random label noise, extended dense training achieves test errors that are worse than pruning by 0 (matching) to 2.6%, 0.4 to 14.5%, 3.6 to 34.2%, 0 (matching) to 4.8% and -0.3%, respectively.

Conclusion. At low sparsities, generalization of dense models is indistinguishable from generalization of sparse models. However, at high sparsities, generalization of dense models no longer

[5] Numerical results available in Table 3 in Appendix M.

matches generalization of sparse models, especially in the presence of substantive random label noise. Therefore, longer training time alone does not account for pruning's generalization-improving effect in the presence of substantive random label noise.

5.2 Size-Reduction

Pruning reduces model size by reducing the number of weights in a given model. We study the effects of size-reduction on generalization alone *without* increasing training time. We do so by comparing pruning with model width down-scaling, since both techniques reduce the number of weights in a model. At optimal sparsities on datasets without random label noise, in Section 1 we show that size reduction is *not necessary* to replicate generalization improvements; in this section, we show that size reduction is also *not sufficient* to replicate generalization improvement.

Method. We reduce model size by down-scaling the width of a dense model: we train a sequence of dense models, where next model in the sequence has 80% of the width of the model preceding it. We then compare generalization of models we obtain with pruning versus width down-scaling.

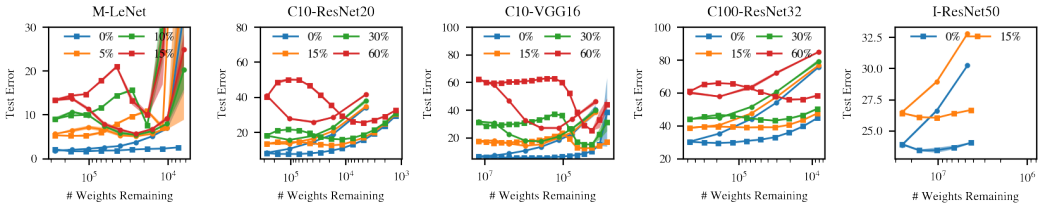


Figure 7: Weights reduction does not explain the full extent of pruning's benefits to generalization. Test errors of the dense and sparse models are shown with round and square-shaped markers respectively. Legend shows random label noise level.

Results. We plot the test errors of models we obtain with pruning and width down-scaling, using square and circle-shaped markers respectively in Figure 7.[6] Width down-scaling under-performs pruning on 4 out of 5 benchmarks on the original training dataset without random label noise. When training with random label noise, width down-scaling improves generalization similar to pruning. In Appendix M, we show that size reduction leads to stronger regularization. Despite this improvement, a gap remains between the minimum test error achieved by pruning and width down-scaling: pruning does better on LeNet, ResNet20, VGG16, ResNet32 and ResNet50 benchmarks by -4 to 0.4%, 0% (matching), 1.5 to 1.8%, 0 (matching) to 2.1%, and 0.36% in test error, respectively.

Conclusion. We conclude that size reduction alone, through down-scaling model width, cannot explain changes in generalization due to pruning. In particular, at optimal sparsity levels pruning achieves better generalization on both standard datasets, and datasets with random label noise.

6 Related Work

Pruning. Pruning has an extended history: early works found that pruning enhances model interpretability and generalization [33, 25, 19]. Recently, motivated by the advent of deep neural network models, pruning is used to reduce the storage and computational demand of such models [16, 17, 31, 5, 24, 32, 9, 50, 3, 27, 47, 38, 28, 29, 41, 39, 7, 51, 43]. For our work, pruning methods can be categorized by 1) time of onset, 2) structuredness and granularity, and 3) iterative or one-shot. We choose to study pruning methods that prune only after the completion of dense model training; such pruning methods often produce models with the best accuracy/parameter count trade-off compared with alternative ones with an earlier time of onset, such as pruning before training [12]. Similarly, we focus on pruning individual elements in weight tensors in an unstructured manner; alternatively, structured, coarse-grained pruning methods impose additional constraints on pruning, and therefore may lead to inferior pruning outcomes. Our study concerns state-of-the-art iterative

[6] To view individual label noise level result in detail, refer to Appendix H.

pruning methods [10, 37], where weights are removed gradually in multiple iterations, with retraining interspersed between iterations to repair the damaged networks; compared with its one-shot counterpart, iterative pruning methods can prune models to more extreme sparsity without destabilizing training [11].

Pruning's Effect on Generalization. Pruning literature often remarks on pruning's beneficial impact on model generalization [25, 16, 10]. This intriguing effect has sparked interest in examining pruning as a technique to gain additional generalization benefits beyond common regularization methods: Giles and Omlin [13] demonstrated that pruning can improve the generalization of recurrent neural network models better than weight decay. Thimm and Fiesler [44], Augasta and Kathirvalavakumar [1] compared the effect on the generalization of several contemporary pruning techniques. Recently, [2] discovered a positive correlation between pruning stability and model generalization, where pruning stability refers to the drop in test accuracy immediately following weights removal. The authors attributed this correlation to an increased flatness of minima due to pruning. Compared with prior literature, our work differs in scope and methodology. Our object of study involves a distinct class of state-of-the-art iterative unstructured-pruning methods, and we derive novel characterizations of the underlying mechanism by examining pruning's impact model training.

Pruning's Effect on Data. Our work contributes to a growing line of work [21, 30] investigating pruning's effect on examples beyond prediction error. Hooker et al. [21] discovered that model compression can have a disproportionately large impact on the prediction performance of underrepresented long-tail of the data distribution. Liebenwein et al. [30] studied the performance and stopping criterion for pruning using metrics such as out-of-distribution generalization and resilience to noise, and find that the maximum weight sparsity that preserves prediction performance varies significantly across different tasks. To the best of our knowledge, our work is the first to examine pruning's impact on examples in the *training set*.

7 Closing Discussion

Societal Impact. Our work examines pruning's effect on generalization. However, we measure generalization using test error and Hooker et al. [21] show that test error may be an imperfect characterization of model prediction quality due to a lack of consideration of fairness and equity. Concretely, our work does not address the potential disproportionate impact of model pruning on label categories. Our work must therefore be interpreted within the context of the community's current and future understanding of pruning's potential contribution to systemic bias, especially against minority groups that may be underrepresented in existing training datasets.

Limitations. Our work is empirical in nature. Though we validate each of our claims with extensive experiment results using 5 dataset-model architecture combinations, each with 2-4 random label noise levels, culminating in 18 independent benchmarks, we recognize that our claims may still not generalize to all circumstances due to the empirical nature of our study.

Conclusion. We show that the long-standing size-reduction hypothesis attributing pruning's generalization benefits to its reduction of the size of a model does not fully explain the relationship between pruning and generalization. Inspired by prior studies [21, 30] that demonstrate pruning's non-uniform impact on examples, we developed a new analysis to interpret pruning's impact on individual examples by assessing its impact on an example's training loss in relation to the example's influence on generalization.

With this analysis, we find that pruning leads to both better optimization, improving model training loss, as well as stronger regularization, mitigating the harmful effect of noisy examples to generalization. Both aspects work to improve generalization of a model.

Our findings along with our novel analytical technique advance the empirical understanding of pruning as a learning algorithm.

References

[1] M. Augasta and T. Kathirvalavakumar. Pruning algorithms of neural networks — a comparative study. *Central European Journal of Computer Science*, 3:105–115, 2013.

[2] B. R. Bartoldson, A. S. Morcos, A. Barbu, and G. Erlebacher. The generalization-stability tradeoff in neural network pruning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[3] C. Baykal, L. Liebenwein, I. Gilitschenski, D. Feldman, and D. Rus. Data-dependent coresets for compressing neural networks with applications to generalization bounds. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=HJfwJ2A5KX`.

[4] M. Belkin, D. Hsu, S. Ma, and S. 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. doi: 10.1073/pnas.1903070116. URL `https://www.pnas.org/doi/abs/10.1073/pnas.1903070116`.

[5] D. Blalock, J. J. Gonzalez Ortiz, J. Frankle, and J. Guttag. What is the state of neural network pruning? In I. Dhillon, D. Papailiopoulos, and V. Sze, editors, *Proceedings of Machine Learning and Systems*, volume 2, pages 129–146, 2020. URL `https://proceedings.mlsys.org/paper/2020/file/d2ddea18f00665ce8623e36bd4e3c7c5-Paper.pdf`.

[6] T. Calders, F. Kamiran, and M. Pechenizkiy. Building classifiers with independency constraints. In *2009 IEEE International Conference on Data Mining Workshops*, pages 13–18, 2009. doi: 10.1109/ICDMW.2009.83.

[7] T. Chen, B. Ji, T. Ding, B. Fang, G. Wang, Z. Zhu, L. Liang, Y. Shi, S. Yi, and X. Tu. Only train once: A one-shot neural network training and pruning framework. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 19637–19651. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper/2021/file/a376033f78e144f494bfc743c0be3330-Paper.pdf`.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[9] X. Dong, S. Chen, and S. Pan. Learning to prune deep neural networks via layer-wise optimal brain surgeon. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/c5dc3e08849bec07e33ca353de62ea04-Paper.pdf`.

[10] J. Frankle and M. Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019.

[11] J. Frankle, G. K. Dziugaite, D. M. Roy, and M. Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[12] J. Frankle, G. K. Dziugaite, D. Roy, and M. Carbin. Pruning neural networks at initialization: Why are we missing the mark? In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=Ig-VyQc-MLK`.

[13] C. Giles and C. Omlin. Pruning recurrent neural networks for improved generalization performance. *IEEE Transactions on Neural Networks*, 5(5):848–851, 1994.

[14] P. D. Grünwald. *The minimum description length principle*. MIT press, 2007.

[15] S. Han, J. Pool, J. Tran, and W. Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, 2015.

[16] S. Han, J. Pool, J. Tran, and W. J. Dally. Learning both weights and connections for efficient neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1135–1143, Cambridge, MA, USA, 2015. MIT Press.

[17] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations*, 2016.

[18] M. Hardt, E. Price, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/9d2682367c3935defcb1f9e247a97c0d-Paper.pdf.

[19] B. Hassibi, D. Stork, and G. Wolff. Optimal brain surgeon and general network pruning. In *IEEE International Conference on Neural Networks*, 1993.

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[21] S. Hooker, A. C. Courville, Y. N. Dauphin, and A. Frome. Selective brain damage: Measuring the disparate impact of model pruning. *CoRR*, abs/1911.05248, 2019.

[22] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[23] A. Krizhevsky, V. Nair, and G. Hinton. Cifar-10 (canadian institute for advanced research).

[24] V. Lebedev and V. Lempitsky. Fast convnets using group-wise brain damage. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[25] Y. LeCun, J. Denker, and S. Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems*, 1990.

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

[27] N. Lee, T. Ajanthan, and P. Torr. SNIP: SINGLE-SHOT NETWORK PRUNING BASED ON CONNECTION SENSITIVITY. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1VZqjAcYX.

[28] N. Lee, T. Ajanthan, S. Gould, and P. H. S. Torr. A signal propagation perspective for pruning neural networks at initialization. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJeTo2VFwH.

[29] L. Liebenwein, C. Baykal, H. Lang, D. Feldman, and D. Rus. Provable filter pruning for efficient neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJxkOlSYDH.

[30] L. Liebenwein, C. Baykal, B. Carter, D. Gifford, and D. Rus. Lost in pruning: The effects of pruning neural networks beyond test accuracy. In *Proceedings of Machine Learning and Systems*, 2021.

[31] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[32] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz. Pruning convolutional neural networks for resource efficient inference. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJGCiw5gl.

[33] M. C. Mozer and P. Smolensky. Skeletonization: A technique for trimming the fat from a network via relevance assessment. In *International Conference on Neural Information Processing Systems*, 1988.

[34] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=B1g5sA4twr.

[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, 2019.

[36] M. Paul, S. Ganguli, and G. K. Dziugaite. Deep learning on a data diet: Finding important examples early in training. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=Uj7pF-D-YvT`.

[37] A. Renda, J. Frankle, and M. Carbin. Comparing rewinding and fine-tuning in neural network pruning. In *International Conference on Learning Representations*, 2020.

[38] T. Serra, A. Kumar, and S. Ramalingam. Lossless compression of deep neural networks. *CoRR*, abs/2001.00218, 2020. URL `http://arxiv.org/abs/2001.00218`.

[39] T. Serra, X. Yu, A. Kumar, and S. Ramalingam. Scaling up exact neural network compression by relu stability. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 27081–27093. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper/2021/file/e35d7a5768c4b85b4780384d55dc3620-Paper.pdf`.

[40] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[41] S. P. Singh and D. Alistarh. Woodfisher: Efficient second-order approximation for neural network compression. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18098–18109. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/d1ff1ec86b62cd5f3903ff19c3a326b2-Paper.pdf`.

[42] L. N. Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472, 2017. doi: 10.1109/WACV.2017.58.

[43] H. Tanaka, D. Kunin, D. L. Yamins, and S. Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6377–6389. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/46a4378f835dc8040c8057beb6a2da52-Paper.pdf`.

[44] G. Thimm and E. Fiesler. Evaluating pruning methods. In *National Chiao-Tung University*, page 2, 1995.

[45] M. Toneva, A. Sordoni, R. T. des Combes, A. Trischler, Y. Bengio, and G. J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=BJlxm30cKm`.

[46] V. Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.

[47] C. Wang, R. Grosse, S. Fidler, and G. Zhang. EigenDamage: Structured pruning in the Kronecker-factored eigenbasis. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 6566–6575. PMLR, 2019. URL `http://proceedings.mlr.press/v97/wang19g.html`.

[48] C. Wang, G. Zhang, and R. Grosse. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=SkgsACVKPH`.

[49] J. Wei, Z. Zhu, H. Cheng, T. Liu, G. Niu, and Y. Liu. Learning with noisy labels revisited: A study using real-world human annotations. *arXiv preprint arXiv:2110.12088*, 2021.

[50] R. Yu, A. Li, C. Chen, J. Lai, V. I. Morariu, X. Han, M. Gao, C. Lin, and L. S. Davis. NISP: pruning networks using neuron importance score propagation. *CoRR*, abs/1711.05908, 2017. URL http://arxiv.org/abs/1711.05908.

[51] X. Yu, T. Serra, S. Ramalingam, and S. Zhe. The combinatorial brain surgeon: Pruning weights that cancel one another in neural networks. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 25668–25683. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/yu22f.html.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

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] Section 7
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Section 7
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Appendix A
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Section 2
   (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] Appendix A
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] Section 2
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (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]