The Effect of Data Dimensionality on Neural Network Prunability

Zachary Ankner*[1,2], Alex Renda[1], Gintare Karolina Dziugaite[3], Jonathan Frankle[2], and Tian Jin[1]

[1]Massachusetts Institute of Technology [2]MosaicML [3]Google Research, Brain Team

Abstract

Practitioners prune neural networks for efficiency gains and generalization improvements, but few scrutinize the factors determining the *prunability* of a neural network – the maximum fraction of weights that pruning can remove without compromising the model's test accuracy. In this work, we study the properties of input data that may contribute to the prunability of a neural network.

For high dimensional input data such as images, text, and audio, the *manifold hypothesis* suggests that these high dimensional inputs approximately lie on or near a significantly lower dimensional manifold. Prior work demonstrates that the underlying low dimensional structure of the input data may affect the sample efficiency of learning. In this paper, we investigate whether the low dimensional structure of the input data affects the prunability of a neural network.

1 Introduction

The *manifold hypothesis* states that the input data for tasks involving images, texts and sounds approximately lie in a low-dimensional manifold embedded in a high dimensional space [Fefferman et al., 2016]. Ample evidence [Cayton, 2005, Narayanan and Mitter, 2010, Ham et al., 2004] supports the manifold hypothesis for commonly used datasets in machine learning tasks. [Narayanan and Niyogi, 2009, Pope et al., 2021] argue that the manifold hypothesis is connected to the sample efficiency of deep neural network models. We investigate whether the low-dimensional manifold structure of input data may also play a role in the surprising empirical success of weight pruning – pruning techniques can remove more than 90% of weights in a neural network model without compromising its accuracy [LeCun et al., 1990, Frankle and Carbin, 2019, Han et al., 2015].

In this work, we empirically examine the impact of three different measures of input data dimensionality on the *prunability* of a neural network model, defined as the maximum fraction of weights that a pruning technique can remove without reducing model accuracy. We refer to the dimensionality of the model inputs as the *extrinsic dimensionality*, and the dimensionality of the lower dimensional manifold that the data lies on as its *intrinsic dimensionality*. We also define a new dimensionality, *task dimensionality*, which is the minimal number of intrinsic dimensions that the output of the function being learned depends on. We develop and test three hypotheses: that the extrinsic, intrinsic, and task dimensionality each contribute to the prunability of neural network models . To test these hypotheses we design three experiments: for each type of dimensionality (extrinsic, intrinsic, and task) we vary that dimensionality while holding all others fixed and observe its effect on the prunability of the neural network model.

*Correspondence to `ankner@mit.edu`

Contributions. We present the following contributions:

- We find that extrinsic dimensionality **has limited effect on** neural network prunability. As extrinsic dimension increases prunability slightly increases with weak correlation.
- We find that intrinsic dimensionality **does affect** neural network prunability. As intrinsic dimension increases prunability decreases.
- We find that task dimensionality **does not affect** neural network prunability.

2 Preliminaries

Dimensionality of data. Consider a classification task on D-dimensional inputs. We refer to the input dimensionality D as the *extrinsic dimensionality*. The *manifold hypothesis* states that natural data lies on or near a low-dimensional manifold embedded in high dimensional space [Fefferman et al., 2016]. We refer to dimensionality of the manifold that the data lies on as the *intrinsic dimensionality* d. Thus, according to the manifold hypothesis, $d \ll D$. We refer to the low dimensional manifold that the inputs lie on as the *intrinsic space*.

Pruning. We prune a neural network using *iterative magnitude pruning (IMP)* with the weight rewinding algorithm [Frankle et al., 2020], which is considered to be state-of-the-art [Rosenfeld et al., 2021, Gale et al., 2019]. IMP begins by training to a network- and data-specific iteration t early in training, and from that point on it iteratively performs the following procedure until it reaches the desired sparsity level: train a (sparse) network to convergence using standard hyperparameters, prune the smallest magnitude weights, rewind the remaining weights back to t.

A *matching subnetwork* is a sparse subnetwork with error that is equal to or lower than the error of the original dense network that the sparse subnetwork was obtained from. The highest sparsity at which a pruned subnetwork is still matching describes the *prunability* of a network with respect to a particular input data distribution and task.

Correlation testing. As we are interested in determining whether varying data dimensionality affects prunability, we quantify the correlation between the ordering of data dimensionalities and prunability using the Spearman rank correlation test. To reflect the underlying uncertainty of the observed pruning rates in the correlation coefficient, we employ a Monte Carlo method to estimate the distribution of the correlation coefficient [Curran, 2014]. To summarize the correlation of a set of experiments, we report both the mean and standard deviation of the correlation coefficient. More details on how we estimate the correlation coefficient distribution are described in Appendix A.2.

3 Extrinsic Dimensionality Has Limited Effect on Prunability

For an image-classification model, the extrinsic dimensionality of its input data is equal to the number of pixels per image. The difference between the extrinsic dimensionality and the intrinsic dimensionality of the input data is therefore a natural source of redundancy in the representation of the input data, which may contribute to the prunability of neural network models. In this section, we examine the impact of the extrinsic dimensionality of the input data on the prunability of a neural network model, while keeping other data dimensionalities fixed.

Methodology. Following the precedent of Pope et al. [2021], we vary the extrinsic dimensionality of the input data by resizing the input image using nearest-neighbor interpolation. For each target extrinsic dimensionality, we resize the spatial dimensions of the input images in a given dataset to the specified extrinsic dimensionality, creating a distinct dataset for each extrinsic dimensionality we study. Subsequently, we train and prune a specified model architecture using the aforementioned dataset. We examine the relationship between the extrinsic dimensionality of the input data and the prunability of the neural network model. We experiment on the CIFAR-10 dataset which originally has an extrinsic dimensionality of 32 (height) x 32 (width) x 3 (channel) = 3072. We vary its extrinsic dimensionality by resizing its spatial dimension to create four separate datasets of spatial dimension $16 \times 16, 32 \times 32, 64 \times 64$, and 128×128. We experiment with 3 variants of the ResNet20 model, evaluating across model widths of $16, 32$, and 64. The number of weights in a convolutional neural

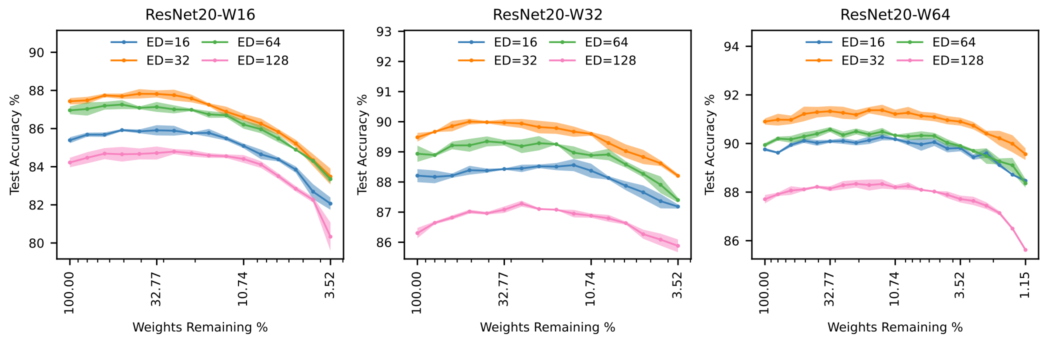

Figure 1: Test accuracy as a function of the percentage of model weights remaining for each model width and extrinsic dimensionality. Pruning across varying extrinsic dimensionalities often follows a line with similar curvature. Legend indicates extrinsic dimension.

network is independent of the spatial dimension of its input. Therefore, in our experiments, varying the extrinsic dimensionality of the input data does not alter the number of weights present in the neural network.

Results. In Figure 1 we plot the model's test accuracy as a function of the percentage of weights remaining for each of the combinations of model width and extrinsic dimensionality. We find that, while the extrinsic dimensionality has an impact on the initial accuracy of the dense model, the accuracy versus sparsity curves across each extrinsic dimensionality with the same model width have similar shapes, suggesting that extrinsic dimensionality has little effect on pruning. We also find that there is no clear relationship between the ordering of extrinsic dimensionalities and the ordering of their corresponding pruning rates. See Table 1 for numerical results. To quantify the weak correlation between the extrinsic dimensionality and prunability, we compute the Spearman rank correlation between the extrinsic dimensionality and the percentage of weights remaining in the smallest matching subnetworks. Across the different model widths, we find an average rank correlation coefficient of -0.22$\pm$0.03. This suggests that while prunability increases as the extrinsic dimensionality increases, it is only a weakly correlated relationship.

Takeaway. We find that increasing the extrinsic dimensionality of input data by upsampling their spatial dimensions is weakly correlated with increased prunability of neural network models.

4 Intrinsic Dimensionality Does Affect Prunability

In this section, we examine the effect of the intrinsic dimension on neural network prunability. The intrinsic dimensionality of the data is the dimensionality of the low dimensional manifold the data lies on. Thus, we hypothesize that as the intrinsic dimensionality of the data increases the prunability of the neural network will decrease as there are fewer redundant dimensions in the data.

Methodology. We follow the precedent of Pope et al. [2021] to generate datasets of varying intrinsic dimensionalities. Concretely, we use a class conditional Generative Adversarial Network (GAN), specifically BigGAN [Brock et al., 2018], to generate images as inputs which are labeled by the class they are conditioned on. The GAN network maps vectors in a latent space to natural images. The dimensionality of the vectors sampled within the latent space is therefore an upper bound on the intrinsic dimensionality of the generated dataset. To vary the intrinsic dimensionality, we mask out the corresponding number of latent dimensions from each vector sampled in the latent space by setting them to 0, before feeding them to the GAN network. For this experiment, we fix the extrinsic dimensionality of the images to be $32 \times 32 \times 3$ and vary the intrinsic dimensionality in $\{16, 32, 64, 128\}$. We experiment with 3 variants of the ResNet20 model, evaluating across model widths of $8, 16$, and 32.

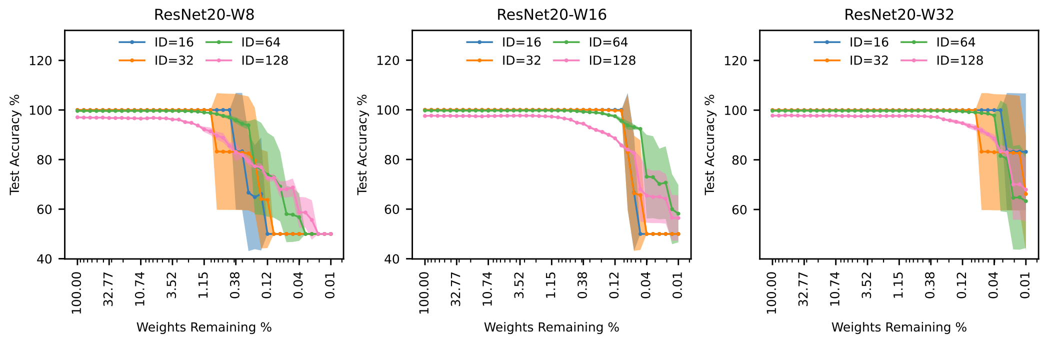

Figure 2: Test accuracy as a function of the percentage of model weights remaining for each model width and intrinsic dimension. Lower intrinsic dimension corresponds to pruned model performance matching dense accuracy for greater amounts of pruning. Legend indicates intrinsic dimension.

Results. In Figure 2 we plot the pruned model's test accuracy as a function of the percentage of weights remaining for each of the combinations of model width and intrinsic dimensionality. We find that models pruned on datasets with lower intrinsic dimensionalities can be pruned to fewer weights remaining without compromising accuracy compared with the same models pruned on datasets with higher intrinsic dimensionalities. See Table 2 for numerical results. To quantify the correlation between the intrinsic dimensionality and prunability, we compute the Spearman rank correlation between the intrinsic dimensionality and the percentage of weights remaining in the smallest matching subnetworks. Averaged across model widths, we find a high rank correlation coefficient of 0.74 ± 0.01. This correlation supports our observation that as intrinsic dimensionality increases, the percentage of weights remaining in the smallest matching subnetworks also increases.

Takeaway. For the classification datasets we generate using a GAN model, we find that the intrinsic dimensionality of the input data affects prunability. The lower the intrinsic dimensionality of input data, the larger the fraction of model weights that can be pruned without compromising test accuracy.

5 Task Dimensionality Does Not Affect Prunability

For image classification tasks, the label of an image often depends only on a subset of the features present in the image. For example, the background of an image is often irrelevant to the classification task. Does the prunability of a neural network model depend on the fraction of the input features that contribute to the task output? In this section, we formalize the intuition behind the number of input features that contribute to the task output and examine its influence on prunability.

Methodology. For this experiment, we assume that the label assigned to a datapoint is a function of its representation in the intrinsic space. Then, for a labeling function f, we define the *task dimensionality* as the number of intrinsic dimensions that the output of f depends on . To investigate the impact the task dimensionality has on prunability, we create a synthetic dataset encoding a linear classification task as follows.

To construct inputs, we first randomly sample $d-$dimensional vectors $\boldsymbol{z}$ wherein each component is sampled i.i.d. from a normal distribution $\mathcal{N}(0, 1)$. We randomly sample a D by d matrix and set the D dimensional input vectors $\boldsymbol{x} = A\boldsymbol{z}$. Each component of A is sampled i.i.d. from a uniform distribution $\mathcal{U}(-1, 1)$. By construction, such input vectors have intrinsic dimensionality of d, and extrinsic dimensionality of D.

To label inputs, we create hyperplanes defining classification boundaries in the intrinsic space. We sample and fix a random $d-$dimensional vector $\boldsymbol{h}$, representing the normal vector to the plane wherein each component is sampled i.i.d. from a uniform distribution $\mathcal{U}(-1, 1)$. Then, we randomly select t components of this hyperplane-defining normal vector to retain and zero out the remaining components. We define the label of an input vector $\boldsymbol{x}$ as $y = \operatorname{sign}(\boldsymbol{h}^\top \boldsymbol{x})$. That is, we label each input vector based on which side of the hyperplane its corresponding representation in the intrinsic

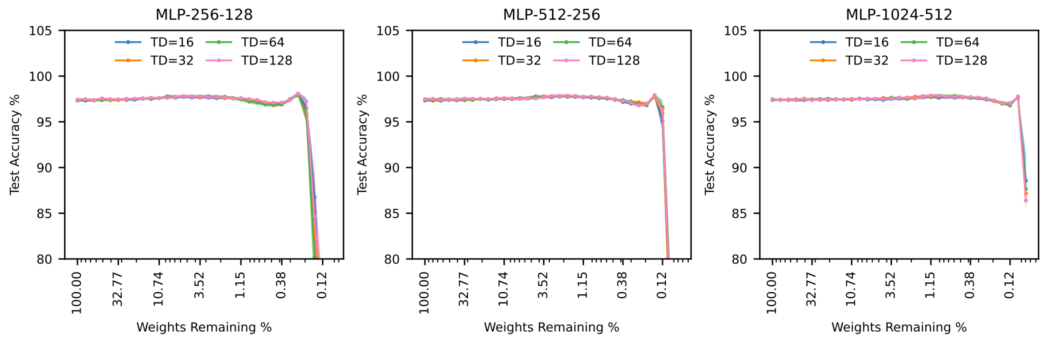

Figure 3: Test accuracy as a function of the percentage of model weights remaining for each model width and task dimension. For most runs the pruning curves are indistinguishable and overlap highly. Legend indicates task dimension.

space lies on. Zeroing out a component in $\boldsymbol{h}$ removes the dependence of classification output on the corresponding intrinsic dimension. This classification task has a task dimensionality of t.

For our experiments, we vary the task dimensionality $t \in \{16, 32, 64, 128\}$ while we fix the intrinsic dimensionality $d = 128$, and extrinsic dimensionality $D = 1024$. We train and prune a 2-layer neural network model consisting of 2 linear layers. We use MLP-P-Q to denote such 2-layer models, where P and Q correspond to the number of features in the first and second hidden layers respectively. For each task dimension, we vary (P,Q) in $(256, 128)$, $(512, 256)$, $(1024, 512)$.

Results. In Figure 3 we plot the test accuracies of pruned models as a function of the percentage of weights that remain in the pruned models. We find that across all model architectures we examine, the prunability of the model does not correlate with task dimensionality of the labeling function. With one exception (MLP-256-128 with TD=16), for each dense model, the sparsities of its smallest matching subnetworks that pruning produces across all task dimensionalities are within one standard deviation of each other. These results suggest that the task dimensionality does not affect the prunability of a neural network model. See Table 3 for numerical results. To quantify the correlation between the task dimensionality and prunability, we compute the Spearman rank correlation between the task dimensionality and the fractions of weights that remain in the smallest matching subnetworks. We obtain a correlation coefficient of 0.01 ± 0.03 suggesting that there is no correlation between task dimensionality and prunability.

Takeaway. For the synthetic linear classification tasks we construct, we find that task dimensionality does not correlate with the prunability of a neural network model.

6 Related Work

Pruning. Practitioners use pruning for many reasons: pruning may improve model generalization and explainability (e.g., [LeCun et al., 1990, Hassibi et al., 1993]); pruning also may reduce memory footprint and improve computational efficiency of model training or inference (e.g., [Han et al., 2015, Frankle and Carbin, 2019, Renda et al., 2020, Blalock et al., 2020, Lebedev and Lempitsky, 2016, Molchanov et al., 2017, Dong et al., 2017, Yu et al., 2018, Baykal et al., 2019, Lee et al., 2019, Wang et al., 2019, Serra et al., 2020, Lee et al., 2020]). While a variety of pruning techniques exist, we choose to study iterative magnitude pruning as it achieves state-of-the-art trade-off between model size and accuracy [Rosenfeld et al., 2021, Gale et al., 2019].

Cause of prunability. Despite the widespread empirical success of pruning for deep neural network models, there has been little analysis on the cause of prunability – why and when can pruning remove substantial fraction of weights in a given model without compromising its accuracy? LeCun et al. [1990], Hassibi et al. [1993], Han et al. [2015] attributed the prunability of neural network models to bias-variance trade-off: as increasing number of weights are pruned, model variance decreases while model bias increases. A smaller model size may therefore exist with the same overall test accuracy but different decomposition of error into bias and variance. Frankle and Carbin [2019] showed the existence of sparse and trainable subnetworks at initialization, thereby conjecturing that SGD seeks out and trains a subset of well-initialized weights. Outside this select subset, weights are therefore redundant. Arora et al. [2018] showed that in deep neural network models, each layer's output is robust against noise injected at preceding layers. They showed that this form of noise stability implies prunability of neural network model.

7 Discussion

In this section we further discuss our results and limitations.

Falsified Hypotheses. We begin our work with a motivating conjecture that the true low dimensional structure of input data contributes to the surprising empirical success of neural network pruning. Based on this conjecture, we develop 3 hypotheses: that extrinsic, intrinsic, and task dimensionality affect prunability. Our empirical observations falsify the hypothesis that task dimensionality affects prunability, and shows that the correlation between extrinsic dimensionality and prunability is weak. Together we present a set of positive and negative results that advances our understanding of the empirical success of neural network pruning.

Experimental Methodology. In this work, we examine the empirical success of neural network pruning from a novel perspective – we investigate how properties of the input data affect prunability. To this end, we design and validate several experimental methodologies. However, this work only presents a preliminary empirical analysis. An important next step is to derive a theoretical framework that unites the observations made across the various types of data dimensionalities studied in this work. Our empirical analysis lays the groundwork for such future studies.

8 Conclusion

To the best of our knowledge, we present the first study that systematically studies the effects of several properties of the input data distribution on prunability. We study how the extrinsic, intrinsic, and task dimensionality of the input data affects neural network prunability. Our results empirically demonstrate that while intrinsic dimensionality affects prunability, extrinsic and task dimensionality have little to no effect on prunability.

References

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 254–263. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/arora18b.html`.

Cenk Baykal, Lucas Liebenwein, Igor Gilitschenski, Dan Feldman, and Daniela 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`.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John 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`.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Lawrence Cayton. Algorithms for manifold learning. 2005.

Peter A Curran. Monte carlo error analyses of spearman's rank test. *arXiv preprint arXiv:1411.3816*, 2014.

Xin Dong, Shangyu Chen, and Sinno Pan. Learning to prune deep neural networks via layer-wise optimal brain surgeon. In I. Guyon, U. Von 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.

Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJl-b3RcF7.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

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

Jihun Ham, Daniel D. Lee, Sebastian Mika, and Bernhard Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 47, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138385. doi: 10.1145/1015330.1015417. URL https://doi.org/10.1145/1015330.1015417.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/ae0eb3eed39d2bcef4622b2499a05fe6-Paper.pdf.

B. Hassibi, D.G. Stork, and G.J. Wolff. Optimal brain surgeon and general network pruning. In *IEEE International Conference on Neural Networks*, pages 293–299 vol.1, 1993. doi: 10.1109/ICNN.1993.298572.

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

Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1990. URL https://proceedings.neurips.cc/paper/1989/file/6c9882bbac1c7093bd25041881277658-Paper.pdf.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. SNIP: SINGLE-SHOT NETWORK PRUNING BASED ON CONNECTION SENSITIVITY. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1VZqjAcYX.

Namhoon Lee, Thalaiyasingam Ajanthan, Stephen Gould, and Philip 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.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJGCiw5gl.

Hariharan Narayanan and Sanjoy Mitter. Sample complexity of testing the manifold hypothesis. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010. URL https://proceedings.neurips.cc/paper/2010/file/8a1e808b55fde9455cb3d8857ed88389-Paper.pdf.

Hariharan Narayanan and Partha Niyogi. On the sample complexity of learning smooth cuts on a manifold. In *COLT*, 2009.

Phil Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. The intrinsic dimension of images and its impact on learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=XJk19XzGq2J.

Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing rewinding and fine-tuning in neural network pruning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1gSj0NKvB.

Jonathan S Rosenfeld, Jonathan Frankle, Michael Carbin, and Nir Shavit. On the predictability of pruning across scales. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9075–9083. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/rosenfeld21a.html.

Thiago Serra, Abhinav Kumar, and Srikumar Ramalingam. Lossless compression of deep neural networks. In Emmanuel Hebrard and Nysret Musliu, editors, *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 417–430, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58942-4.

Chaoqi Wang, Roger Grosse, Sanja Fidler, and Guodong 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.

Ruichi Yu, Ang Li, Chun-Fu Chen, Jui-Hsin Lai, Vlad I Morariu, Xintong Han, Mingfei Gao, Ching-Yung Lin, and Larry S Davis. Nisp: Pruning networks using neuron importance score propagation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9194–9203, 2018.

A Appendix

A.1 Acknowledgement

This work was supported in part by a Facebook Research Award, the MIT-IBM Watson AI-LAB, Google's Tensorflow Research Cloud, and the Office of Naval Research (ONR N00014-17-1-2699).

A.2 Correlation of Input Dimensionality and Prunability

In this section we describe how we compute the Spearman rank correlation coefficient for each setting, and plot the correlation between prunability and data dimensionality.

Methodology As outlined in Section 2, we compute the Spearman rank correlation coefficient between the minimum fraction of weights remaining and the dimensionality of the data. To capture the uncertainty in our experiments in the rank correlation coefficient, we simulate experiment outcomes using the mean and standard deviation of the fraction of weights remaining. For each rollout, we generate 1000 potential experimental outcomes by randomly selecting a data dimensionality and then sampling an outcome from a normal distribution with the same mean and standard deviation as the observed trial. We then compute the rank correlation coefficient based on the 1000 sampled outcomes. To collect the distribution of correlation coefficients, we perform 5000 rollouts.

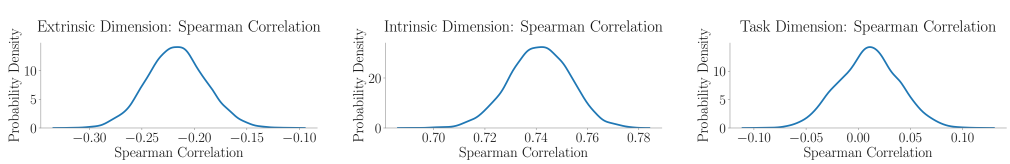

Figure 4: Distribution of Spearman rank correlation coefficients between the data dimensionality and the percentage of model weights remaining at matching sparse accuracy.

Results As can be seen in Figure 4, the intrinsic dimension experiments have a high positive correlation between input dimensionality and percentage of weights remaining, while the extrinsic and task dimension experiments have weak and no correlation respectively. This can be interpreted as higher intrinsic dimensions are correlated with a greater percentage of weights remaining and thus lower sparsity, while the extrinsic and task dimensions do not significantly influence the prunability.

A.3 Sparsity Rates

In this section we report the full set of results on pruning rates across extrinsic, intrinsic, and task dimensionality (Table 1,2,3).

Table 1: Minimum percentage of weights remaining where the sparse model matches the dense model accuracy. Results reported for each model width and extrinsic dimension.

Model	Extrinsic dimension	Weights remaining (%)
ResNet20-W16	16	15.86 ± 0.71
	32	19.15 ± 1.96
	64	18.38 ± 1.70
	128	14.53 ± 1.30
ResNet20-W32	16	13.37 ± 1.33
	32	14.03 ± 0.59
	64	13.26 ± 0.50
	128	11.76 ± 1.12
ResNet20-W64	16	10.66 ± 0.69
	32	10.17 ± 0.00
	64	10.17 ± 0.00
	128	9.39 ± 0.25

Table 2: Minimum percentage of weights remaining where the sparse model matches the dense model accuracy. Results reported for each model width and intrinsic dimension.

Model	Intrinsic dimension	Weights remaining (%)
ResNet20-W8	16	6.32 ± 0.23
	32	7.73 ± 0.00
	64	9.64 ± 0.86
	128	70.85 ± 41.22
ResNet20-W16	16	5.43 ± 0.41
	32	6.16 ± 0.11
	64	23.34 ± 22.58
	128	8.87 ± 0.23
ResNet20-W32	16	4.37 ± 0.10
	32	5.18 ± 0.29
	64	6.09 ± 0.29
	128	9.36 ± 2.49

Table 3: Minimum percentage of weights remaining where the sparse model matches the dense model accuracy. Results reported for each model width and task dimension.

Model	Task dimension	Weights remaining (%)
MLP-256-128	16	5.65 ± 0.09
	32	5.79 ± 0.00
	64	5.79 ± 0.00
	128	5.72 ± 0.09
MLP-512-256	16	5.40 ± 0.00
	32	5.40 ± 0.00
	64	5.34 ± 0.08
	128	5.40 ± 0.00
MLP-1024-512	16	5.06 ± 0.00
	32	5.06 ± 0.00
	64	5.06 ± 0.00
	128	5.06 ± 0.00