Linear Mode Connectivity in Sparse Neural Networks

Luke McDermott
Modern Intelligence
luke@modernintelligence.ai

Daniel Cummings
Modern Intelligence
daniel@modernintelligence.ai

Editors: Marco Fumero, Emanuele Rodolà, Clementine Domine, Francesco Locatello, Gintare Karolina Dziugaite, Mathilde Caron

Abstract

With the rise in interest of sparse neural networks, we study how neural network pruning with synthetic data leads to sparse networks with unique training properties. We find that distilled data, a synthetic summarization of the real data, paired with Iterative Magnitude Pruning (IMP) unveils a new class of sparse networks that are more stable to SGD noise on the real data, than either the dense model, or subnetworks found with real data in IMP. That is, synthetically chosen subnetworks often train to the same minima, or exhibit linear mode connectivity. We study this through linear interpolation, loss landscape visualizations, and measuring the diagonal of the hessian. While dataset distillation as a field is still young, we find that these properties lead to synthetic subnetworks matching the performance of traditional IMP with up to 150x less training points in settings where distilled data applies.

1 Introduction & Background

Sparse neural networks are increasingly important in deep learning to enhance hardware performance (e.g., memory footprint, inference time) and reduce environmental impacts (e.g., energy consumption), especially as state-of-the-art foundational models continue to grow significantly in parameter count. The most common form of sparsity can be found in neural network pruning literature [7]. In this field, researchers exploit sparsity for computational savings, usually at inference, by removing parameters after training. In order to reduce the cost of training as well, other works explore how to prune at initialization [10, 15], the end goal for almost any pruning research. Despite these great ambitions, pruning at initialization does not perform as we hope [5]. To further understand why this is the case, Frankle et al. [3] propose the Lottery Ticket Hypothesis: *for a sufficiently over-parameterized dense network, there exists a non-trivial sparse subnetwork that can train in isolation to the full performance of the dense model*. This is empirically validated for small settings with Iterative Magnitude Pruning with weight rewinding back to initialization. In parallel, researchers have been exploring how synthetic data representations such as those generated by dataset distillation methods can be leveraged to efficiently accelerate deep learning model training. With this in mind, we explore the training dynamics and stability of sparse neural networks in the context of synthetic data to better understand how we should be efficiently creating sparsity masks at initialization.

Research on the training dynamics of dense models have led researchers to find that dense models are connected in the loss landscape through nonlinear paths [6, 2, 8]. Linear paths or Linear Mode Connectivity (LMC) is an uncommon phenomena that only occurs in rare cases, such as MLPs on subsets of MNIST in [12]. For large networks, Frankle et al. [4] found that pretrained dense models, when fine-tuned across different shufflings of data, are linear mode connected. While these models are "stable" to noise generated through stochastic gradient descent (SGD), only the smallest

Proceedings of the I edition of the Workshop on Unifying Representations in Neural Models (UniReps 2023).

dense models are stable at initialization. As for its relationship with sparse neural networks, it was empirically found that the Lottery Ticket Hypothesis only holds for stable dense models, those that are linear mode connected across data shuffling [4]. They found that these large dense models only become stable early in training, leading to the conclusion that Iterative Magnitude Pruning (IMP) with weight rewinding, the method to find such lottery tickets, should instead rewind a model to an early point in training rather than at initialization, revising the hypothesis to fit in larger settings. Our work aims to study the properties of sparse neural networks at initialization, different than new age lottery ticket literature [13], which utilizes some pretraining to find a "good" initialization.

We find that another class of sparse subnetworks exist that are more stable at initialization: *synthetic subnetworks*. We define synthetic subnetworks as those produced during "distilled pruning" [11]. These are found by replacing the traditional data in IMP with distilled data, essentially a summarized version of the training data consisting of only 1-50 synthetic images per class (see [14] for a survey). In general, dataset distillation optimizes a synthetic dataset to match the performance of a model trained on real data. This bi-level optimization problem can be defined as minimizing the difference of average loss over all validation points:

$$\mathcal{D}_{\text{syn}} |L(\Phi(\mathcal{D}_{\text{real}}); \mathcal{D}_{\text{val}}) - L(\Phi(\mathcal{D}_{\text{syn}}); \mathcal{D}_{\text{val}})| \tag{1}$$

In distilled pruning, we perform the same training, pruning, and rewinding to initialization in order to produce the sparsity mask. This mask, as with those produced by IMP, can be applied to the dense model at initialization to create a high performing sparse neural network after training on real data. The significance is that synthetic images can be used to pick an appropriate sparsity mask for a downstream task. Recent work shows that despite synthetic subnetworks having a lower performance as a trade-off for pruning efficiency, these subnetworks have a lower need for rewinding to an early point in training due to their inherent stability [11]. We find that, with better dataset distillation methods such as Information-intensive Dataset Condensation (IDC) [9] rather than Matching Training Trajectories (MTT) [1] which was used previously, we match performance with IMP when rewinding to initialization. We achieve this with 5x less data than previous distilled pruning work which is approximately *150x less* training points than traditional IMP to find a sparsity mask. While we do use a current state-of-the-art distillation method, such methods are still limited to models up to ResNet-18 and small datasets like CIFAR-10, CIFAR-100, and subsets of ImageNet.

2 Dataset distillation for neural network pruning.

To find a suitable sparsity mask of a randomly initialized model, we first train the network to convergence on distilled data[1], prune the lowest magnitude weights, then rewind the non-pruned weights back to their initialized values, and loop until desired sparsity. The final model should have its randomly initialize weights with a sparsity mask. We can train the sparse synthetic subnetwork on real data to achieve sufficient performance at high sparsities. Using distilled data only to choose our sparsity mask allows us to better understand the architectural relationship of this data. We refer to subnetworks found with synthetic or distilled data as *synthetic subnetworks* and those with real data as *IMP subnetworks*. The only differences of IMP and distilled pruning lie in the sparsity mask they choose. Since each method uses different datasets for training, their final converged weights will be different. What is deemed "important" for real data might not be important for distilled data; therefore, distilled pruning may attempt to remove these. The performance of these sparsity masks by distilled pruning directly relates to how relevant the distilled data is to the real data. We find that distilled pruning can match the performance of IMP, when rewinding to initialization, on settings where dataset distillation applies[2].

3 Stability of subnetworks.

To understand the training dynamics of sparsity masks chosen via distilled pruning vs IMP, we conduct an instability analysis. We take a randomly initialized model, generate a sparsity mask

[1]We use dataset distillation methods that match training trajectories to ensure that training on synthetic data yields similar converged results to training on real data [9]

[2]Figure 4 showcases this performance, comparing traditional IMP to distilled pruning on CIFAR-10 with ResNet-18.

through pruning, and train it across two different orderings of the real training data. We save these two models and interpolate all the weights between them, measuring the training loss at each point in the interpolation as shown in Figures 1 and 2. We assess the linear mode connectivity of these subnetworks to determine if the model is stable to SGD noise. If the loss increases as you interpolate between two trained versions, then there is a barrier in the loss landscape, implying the trained models found different minima. In these cases, the ordering of the training data directly impacts what minima its choosing. If the loss does not increase during interpolation, then this implies they exist in the same minima or at least the same flat basin.

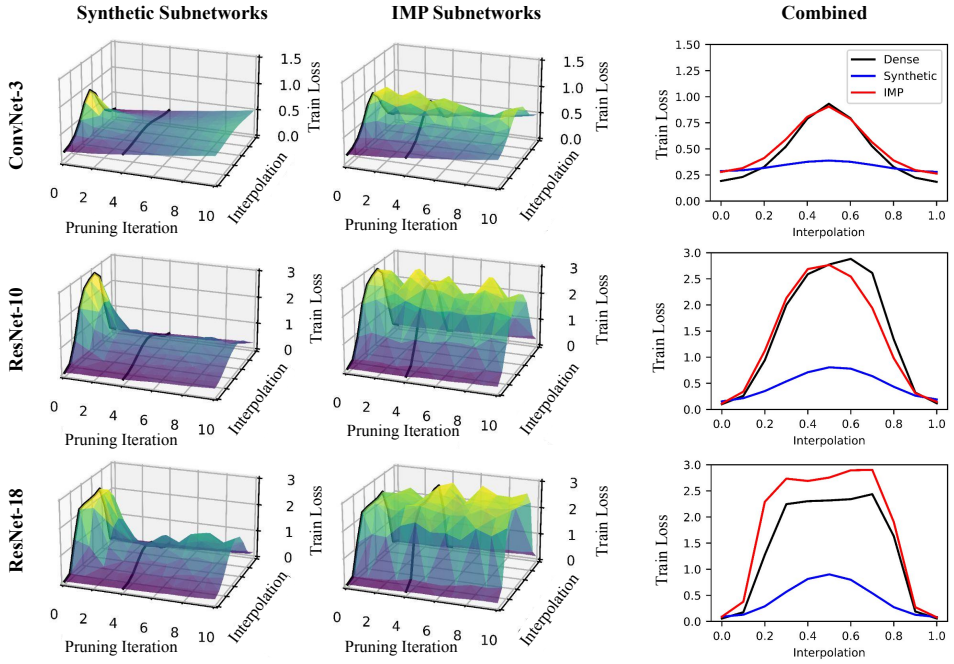

Figure 1: Comparison of the stability of synthetic vs. IMP subnetworks at initialization on CIFAR-10. We show how the loss increases as you interpolate the weights between two trained models. We measure this for subnetworks of different sparsities. The left column is reserved for subnetworks found via distilled data, and the middle column is for subnetworks found with real data. The dark lines in the 3D plots represents the pruning iteration we used for the combined plot; the dense model is iteration 0.

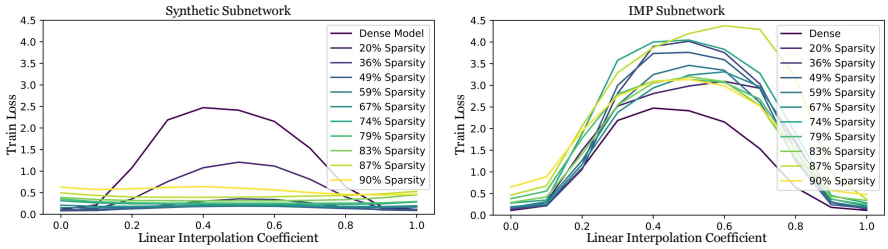

Figure 2: Comparison of the stability of synthetic vs. IMP subnetworks at initialization on ImageNet-10 and ResNet-10. An increased loss across interpolation implies instability / trained networks landing in different minima.

We see that in simpler scenarios with ConvNet-3 on CIFAR-10 & ResNet-10 on ImageNet-10, we exhibit full linear mode connectivity. We even see slightly better performance during interpolation in Figure 2. As stated before, it was shown that lottery tickets can be found with IMP only when the dense model is stable [4]. However, we find that in some cases of unstable dense models there exists a sparse subnetwork that is stable at initialization. More importantly, traditional IMP is not able to

produce stable subnetworks in these settings. Sparsity is not necessarily the answer for smoother landscapes, *where* you induce sparsity is the main factor. As pruning continues, the results exhibit more stability despite lower trainability, as seen with higher training losses. We postulate that the parameters pruned on distilled data, yet still exist in the IMP subnetwork, capture the intricacies of the real data which contribute to a sharper, but more trainable, landscape. Since IMP subnetworks are not stable, the intricacies it is learning is order dependent.

4 Loss Landscape Visualization

While linear mode connectivity is useful to study the loss landscape, this lightweight method can only show us a one dimensional slice of the bigger picture. We further examine the landscapes across two dimensions of parameters as shown in Figure 3.

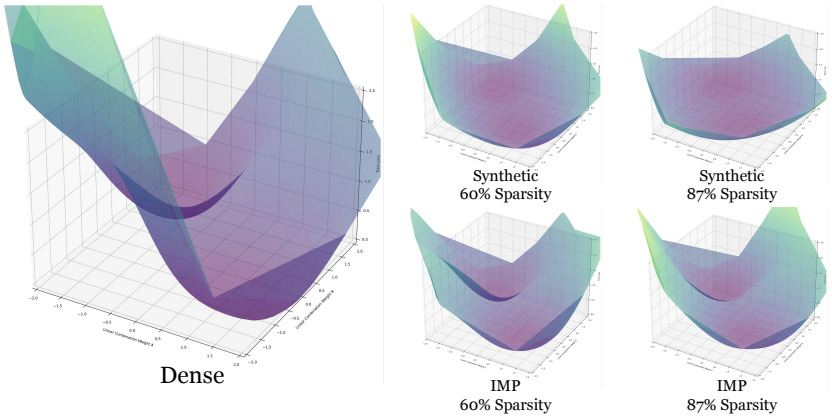

Figure 3: Loss Landscape visualization around the neigbhorhood defined by trained models on different seeds for ConvNet-3 and CIFAR-10.

We created two orthogonal vectors from trained reference models in order to map the hyperdimensional parameter space down to two dimensions. For each of the 10,000 points, we take the linear combination of the two vectors, and measure loss on real training data. Since this visualization is created after reference models are trained, reference models that are closer together will result in "zooming in" on their minima. The spatial distance is not preserved using this method. This is useful in determining the local area in which these models are training to. With post-hoc analysis, we find that spatial distance in our plot is mainly maintained, with slightly lower distances as you prune. From these visualizations, IMP chooses subnetworks that exhibit a similar landscape to the dense model. We see the trained models fall into two separate minima in both the IMP and Dense cases, explaining the loss barrier in the Figure 1. Subnetworks chosen with distilled data are falling into the same, flat basin.

Across almost all experiments, we see a general trend: subnetworks chosen via distilled pruning result in a smooth & generalizing loss landscape. As compression ratio increases, we see more stability than IMP; however, the performance trend largely depends on the distilled accuracy, in this case by using the IDC method [9]. Most notably, we achieve full linear mode connectivity for ConvNet-3 on CIFAR-10 and ResNet-10 on Imagenet-10. While there are numerous factors at play, IDC [9] optimized the synthetic data specifically for these models on each dataset, hinting that stability is a result of high performing synthetic data.

5 Performance of Distilled Pruning

We see in Figure 4 that lower test accuracy after training on distilled data does not completely translate in neural network pruning. Despite only achieving 72.8% after training on distilled CIFAR-10 with 10 ipc from IDC [9], we can use this low performance to choose weights to prune matching the performance on IMP on smaller datasets. To be clear, distilled data is only used to find a sparsity mask, the sparse model is ultimately trained on real data for validation. As previously

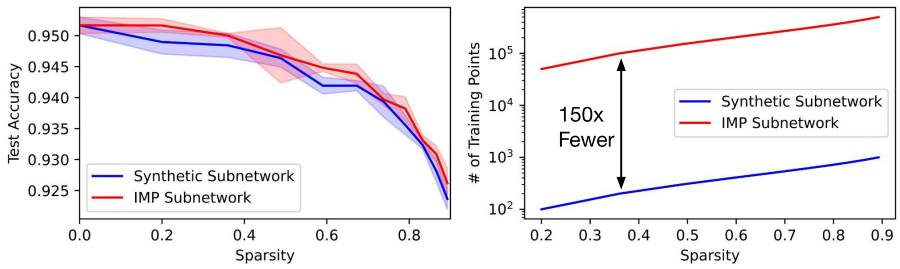

Figure 4: Performance of Distilled Pruning vs Traditional IMP on ResNet-18 & CIFAR-10. The distilled dataset consisted of 10 images per class. Error bars are plotted as we average across 4 seeds. The plot on the right measures the amount of data points used in training to find a sparsity mask at x sparsity.

mentioned, this performance does not hold as dataset complexity increases, we find that CIFAR-100 can perform decently at lower sparsities, but does not handle extreme sparsities ($> 90\%$) with larger models like ResNet-18. In those cases, the distilled data on CIFAR-100 is not maintaining task-relevant information for the model. Since distilled data contains less outliers, and is a largely more "generalizable" dataset, we find that the sparsity mask pruned weights that control the fine grained details of the real data.

6 Conclusion

This work is an initial step into exploring the impact of using synthetic data, specifically distilled data, on pruning. We thoroughly assess the linear mode connectivity of these subnetworks to determine if the model is stable to SGD noise, even finding stable subnetworks from unstable dense models. We believe the inherent compression of dataset distillation is a driving factor in synthetic subnetworks' stability. Lastly, we believe this hints at the possibility of finding lottery tickets at initialization by first searching for stable subnetworks. In turn, we invite researchers to find new ways to search for stable subnetworks, especially on the real data.

References

[1] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories, 2022.

[2] Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred A. Hamprecht. Essentially no barriers in neural network energy landscape, 2019.

[3] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks, 2019.

[4] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis, 2020.

[5] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Pruning neural networks at initialization: Why are we missing the mark?, 2021.

[6] C. Daniel Freeman and Joan Bruna. Topology and geometry of half-rectified network optimization, 2017.

[7] Trevor Gale, Erich Elsen, and Sara Hooker. The state of sparsity in deep neural networks, 2019.

[8] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns, 2018.

[9] Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. Dataset condensation via efficient synthetic-data parameterization, 2022.

[10] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Snip: Single-shot network pruning based on connection sensitivity, 2019.

[11] Luke McDermott and Daniel Cummings. Distilled pruning: Using synthetic data to win the lottery, 2023.

[12] Vaishnavh Nagarajan and J. Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning, 2021.

[13] Mansheej Paul, Brett W. Larsen, Surya Ganguli, Jonathan Frankle, and Gintare Karolina Dziugaite. Lottery tickets on a data diet: Finding initializations with sparse trainable networks, 2022.

[14] Noveen Sachdeva and Julian McAuley. Data distillation: A survey, 2023.

[15] Hidenori Tanaka, Daniel Kunin, Daniel L. K. Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow, 2020.