Linear Mode Connectivity in Branching SGD Trajectories: Insights into Optimization Landscape

Itai Shapira
Harvard University

Abstract

In this study, we explore the loss barrier between two equal-length SGD trajectories branching from a common initial portion of training with the same data order, focusing on linear mode connectivity during training. We aim to understand the behavior of the loss barrier as a function of the number of common gradient steps (k_1) and the number of independent steps (k_2). Our contributions include: (1) showing that even with an early split in trajectories (low k_1), linear mode connectivity is maintained after many independent steps (k_2), (2) demonstrating that as k_2 increases, linear connectivity persists if k_1 is sufficiently high, despite trajectory divergence in weight-space and prediction-space, (3) providing evidence that during early training stages, the optimization problem enters an approximately convex basin, and (4) validating these observations across various network architectures and datasets, expanding on previous work.

1 Introduction

Despite their widespread use and impressive performance, understanding the underlying mechanisms that contribute to the success of deep neural networks remains a challenging task. One of the key questions in deep learning research is understanding the optimization landscape of neural networks and how Stochastic Gradient Descent (SGD) navigates through it to find optimal solutions. In this context, the phenomenon of Linear Mode Connectivity (LMC) has recently attracted attention due to its potential implications on the optimization process and practical applications.

Linear Mode Connectivity refers to the existence of linear paths with non-increasing loss connecting different minima found by SGD in the optimization landscape. This connectivity has implications for pruning and sparse training methods, distributed optimization, and ensemble techniques, among other applications. A deeper understanding of LMC can provide insights into the factors governing the optimization process and contribute to the development of more effective training algorithms and network architectures.

Despite recent progress in understanding LMC, the phenomenon remains largely unexplained, calling for further investigation. Our research intends to examine the theoretical and empirical aspects of LMC in deep neural networks. By exploring various factors that may potentially affect LMC and evaluating their impact on the observed behavior, we aim to broaden the current understanding and offer a more comprehensive explanation of this phenomenon.

In our analysis, we examine the loss barrier between two SGD trajectories of equal length that branch off from a shared initial portion of training with a common data order. Distinct from much of the previous work, our focus lies in comprehending linear mode connectivity *during* the training process. Specifically, our primary objective is to understand the behavior of the loss barrier as a function of both the number of common gradient steps (k_1) and the number of independent steps (k_2). Our contributions are as follows:

- Demonstrated that even if the split in trajectories occurs early in training (low k_1), where it has a comparatively lower accuracy level than the final accuracy, there would still be a zero barrier or linear mode connectivity, even after many independent steps (k_2).
- Demonstrated that as k_2 increases, even though the two trajectories diverge in both weight-space linear connectivity is maintained, provided that k_1 is sufficiently high
- These two observations offer evidence to support the hypothesis that, during early stages of training, the optimization problem, restricted to the path navigated in weight-space by SGD, enters an approximately convex basin.
- Expanded upon previous work by examining different network architectures and datasets, validating the observations of linear connectivity across various scenarios.

2 Preliminaries

2.1 Formal Definitions

Let $f_{\boldsymbol{w}} : \mathbb{R}^d \longrightarrow \mathbb{R}^p$ be a neural network with weights $\boldsymbol{w}$ and $\mathcal{L}(\boldsymbol{w})$ a loss function. Let $\mathcal{E}_\alpha(\boldsymbol{w}_1, \boldsymbol{w}_2) = \mathcal{L}(\alpha \boldsymbol{w}_1 + (1-\alpha)\boldsymbol{w}_2)$ be the loss of the network created by linearly interpolating between the weights. The **loss barrier** is defined as:

$$\mathcal{B}(\boldsymbol{w}_1, \boldsymbol{w}_2) = \sup_{\alpha} \mathcal{L}\Big(\alpha \boldsymbol{w}_1 + (1-\alpha)\boldsymbol{w}_2\Big) - \Big(\alpha \mathcal{L}(\boldsymbol{w}_1) + (1-\alpha) L(\boldsymbol{w}_2)\Big) \tag{1}$$

We say that two networks $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are LMC if $B(\boldsymbol{w}_1, \boldsymbol{w}_2) \approx 0$. The definition provided above differs from the one proposed by Frankle et al. [2020], which used $\frac{1}{2}\mathcal{L}(\boldsymbol{w_1}) + \frac{1}{2}\mathcal{L}(\boldsymbol{w_2})$ instead of $\alpha\mathcal{L}(\boldsymbol{w_1}) + (1-\alpha)\mathcal{L}(\boldsymbol{w_2})$. These definitions are equivalent when $\mathcal{L}(\boldsymbol{w_1}) = \mathcal{L}(\boldsymbol{w_2})$. However, we adopt the definition suggested by Entezari et al. [2021], which assigns no barrier value to a loss that changes linearly between $\boldsymbol{w_1}$ and $\boldsymbol{w_2}$. Notice that LMC would be expected if the loss landscape were convex.

In our analysis, we focus on the loss barrier between two SGD trajectories of the same length. Each step of SGD is stochastic and dependent on different data orders. Let k denote the number of stochastic steps in a trajectory. We consider branch trajectories that split from a shared early portion of training with a common data order. We use k_1 to represent the number of common gradient steps and k_2 to represent the number of independent steps. Our goal is to understand the behavior of the loss barrier as a function of both k_1 and k_2: $\mathcal{B}(k_1, k_2)$. We refer to the model at step k_1, before the split, as the **pre-trained** model.

2.2 Related Research

The investigation of the loss landscape in deep neural networks has garnered significant attention, with a focus on the connectivity between different minima found by Stochastic Gradient Descent (SGD). Early research explored the existence of nonlinear paths of non-increasing loss in the parameter space between such minima, a phenomenon often referred to as mode connectivity Freeman and Bruna

[2016], Draxler et al. [2018]. Subsequent studies, such as those by Draxler et al. [2018] and Garipov et al. [2018], demonstrated that local minima found by SGD can be connected via piecewise linear paths. More recent research has delved into the linear connectivity of trained networks from the same initialization. Nagarajan and Kolter [2019] found that such networks are connected by linear paths of constant test error. Building upon this, both Frankle et al. [2020] and Yunis et al. [2022] revealed that, under certain conditions, the loss function is roughly convex when restricted to the convex hull of SGD solutions obtained from the same initialization.

One area of research has explored the permutation invariance of neural networks. Entezari et al. [2021] put forth a conjecture suggesting that by taking into account permutation invariance, the barriers in linear interpolation between SGD solutions could be substantially reduced. This idea implies that the loss landscape of neural networks, when considering all possible permutation symmetries of hidden units, often comprises (almost) a single basin. Building on this, Ainsworth et al. [2022] demonstrated that the conjecture is valid for wide networks (although not for ResNets, as confirmed by Benzing et al. [2022]) and devised algorithms for permuting the units of one model to align them with a reference model, thereby facilitating the fusion of the two models in weight space.

Finally, recent work has examined the impact of network width and depth on the barriers between models. Entezari et al. [2021] showed that the barrier decreases with the width of the network, while Jordan et al. [2022] initially observed that the barrier appears to increase with depth due to the ”variance collapse” phenomenon. However, after correcting for this vanishing variance, the barrier was found to actually decrease with depth.

Understanding LMC has both theoretical and practical implications. Theoretically, LMC offers insights into why SGD and its variants are effective in training large networks, despite the inherent complexity of large non-convex optimization problems. Practically, LMC is relevant to pruning and sparse training methods, distributed optimization, and ensemble methods, which rely on a deep understanding of the loss landscape and the ability to sample from solutions. For instance, the Lottery Ticket Hypothesis (LTH) (Frankle and Carbin [2018]) conjectures that neural networks contain a sparse sub-network that can be trained in isolation from initialization to achieve comparable test accuracy. Solutions that are linearly connected with no barrier have the same lottery ticket Frankle et al. [2020]. This connection between LTH and LMC suggests potential benefits for training and generalization. Furthermore, improved knowledge of mode connectivity has been shown to be essential in devising better ensemble methods (Garipov et al. [2018]), while linear mode connectivity between solutions or checkpoints facilitates the use of weight averaging techniques for distributed optimization (Scaman et al. [2019]). Fort et al. [2020] also demonstrate the connection between linear connectivity and the advantage nonlinear networks enjoy over their linearized version.

3 Empirical Analysis

3.1 Experimental Setup

Networks and datasets. We investigate image classification networks on MNIST (LeCun [1998]), CIFAR-10 (Krizhevsky et al. [2009]), and TinyImageNet (Deng et al. [2009]), as detailed in Table 1. All hyperparameters are set to standard values from established in previous research: For MNIST, we employed the LeNet-5 architecture (LeCun et al. [1998]). The network consists of a total of 7 layers, including 2 convolutional layers, followed by 2 average pooling layers, and 3 fully connected layers. For CIFAR-10, we used a reduced version of the VGG-16 architecture (Simonyan and Zisserman [2014]). The modified model consists of four layers in total, with two convolutional layers followed by two fully connected layers. Max-pooling is performed after each convolutional layer, and dropout is applied after each of the fully connected layers. This reduced VGG model has 855,000 parameters, making it less computationally intensive while maintaining the basic structure of the original VGG-16 architecture. We used the AlexNet architecture (Krizhevsky et al. [2017]) for TinyImageNet. Comprised of eight layers in total, AlexNet features five convolutional layers followed by three fully connected layers, with the final layer outputting class labels. The first two convolutional layers include batch normalization, while dropout is employed after each of the last two fully connected layers.

Using smaller batch sizes relative to the number of training samples injects more noise into the stochastic weight vector updates performed by SGD, potentially resulting in greater divergence in weight space between the two trained models. The batch sizes and training samples for each dataset are detailed in Table 1. All the architectures examined in this project exhibit overparameterization, in the sense that they have more trainable parameters than data-points.

Network	Dataset	Params	Max Steps	Max Epochs	Batch Size	Train Samples	Final Accuracy	Optimizer
LeNet5	MNIST	62K	580	10	1024	60,000	0.9818	Adam
ReducedVGG	CIFAR-10	855K	11,310	29	128	50,000	0.7136	Adam
AlexNet	TinyImageNet	60M	14,058	9	64	100,000	0.4064	Adam

Table 1: Summary of image classification networks and their performance on various datasets.

Sampling Method. We adopt an efficient training method to search multiple different branch trajectories of stochastic gradient steps, to minimize computational resources, as detailed in Algorithm 1. We first train a single model, referred to as the ”main branch” trajectory (see plot next to Algorithm 1), for the maximum number of epochs, saving checkpoints at each epoch. Subsequently, we reload the model for various k_1 values and continue the training of only one new model for an additional k_2 steps. Afterward, we couple this new model with the original ”main branch” model at the $k_1 + k_2$ position. This approach significantly reduces the computational resources required for the experiment, yet generates samples that depend on the same random initialization. To address this issue, we repeat this process twice.

When we load the checkpoint model, we also load the optimizer from the same point, which enables us to resume its learning schedule. All models were trained using the Cross Entropy loss.

For each pair of (k_1, k_2), we generate two pairs of models which we identity by their weights vectors $\boldsymbol{w}_{k_1+k_2}$ and $\boldsymbol{w}'_{k_1+k_2}$. These two models were trained using k_2 independent training steps from the same pre-trained model, previously trained for k_1 steps. We then explore the linear path in weight-space between these two models and sample the loss function along this path. To do this, we choose 25 equally-spaced points between 0 and 1, denoted by α. For each sampled α, we compute the loss function at the point $\alpha\boldsymbol{w}_{k_1+k_2} + (1-\alpha)\boldsymbol{w}'_{k_1+k_2}$.

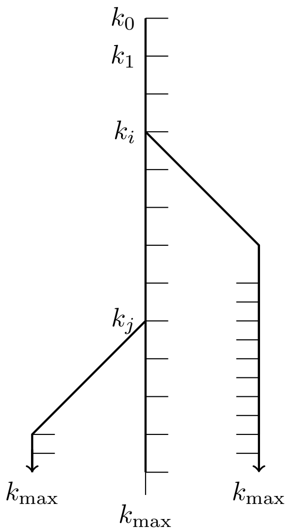
We examine the difference between the interpolation loss and the interpolated loss value, a term we refer to as the *interpolation gap*:

$$\mathcal{L}(\alpha\boldsymbol{w}_{k_1+k_2} + (1-\alpha)\boldsymbol{w}'_{k_1+k_2}) - \alpha\mathcal{L}(\boldsymbol{w}_{k_1+k_2}) + (1-\alpha)\mathcal{L}(\boldsymbol{w}'_{k_1+k_2})$$

The interpolation gap highlights the disparity between the actual loss along the linear path in weight-space and the loss value expected based on the interpolation of the two models. Note that the value is always zero at $\alpha = 0$ or $\alpha = 1$. We define the cross entropy train/loss barrier as the maximum sampled interpolation gap. In certain cases, the loss function exhibits convex behavior along the path, resulting in a decrease in loss and a negative interpolation gap. In cases where the context makes it apparent, we regard the barrier as negative in these scenarios. Additionally, we evaluate not just

Algorithm 1 Training Method

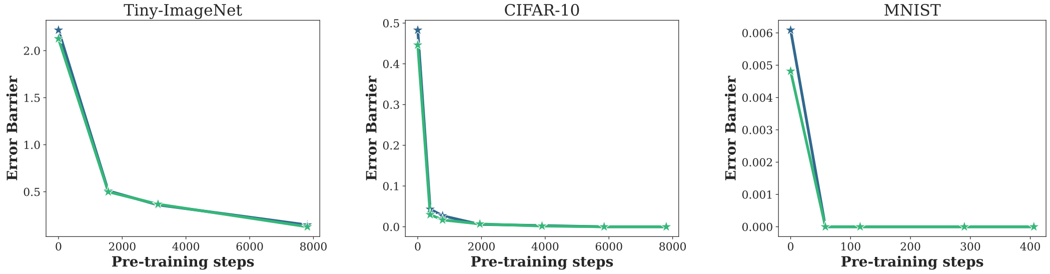
```
Require: datasets, network with random initialization, $k_{\max}$, list of $k_1$ values $K$
checkpoints $\leftarrow \{\}$
for $i \leftarrow 1$ to $k_{\max}$ do
    Train the network for one epoch with SGD        {main branch}
    checkpoints$[i] \leftarrow$ network
end for
Results $\leftarrow \{\}$
for $k_1$ in $K$ do
    model $\leftarrow$ checkpoints$[k_1]$
    for $k_2 \leftarrow k_1$ to $k_{\max}$ do
        Train model for one epoch with SGD
        Results$[k_1, k_2] = (checkpoints[k_1 + k_2], model)$
    end for
end for
return Results
```



the loss barrier, but also the error barrier, defined as one minus the accuracy. By examining both the train/loss barrier and the error barrier, our empirical analysis offers a thorough understanding of the model's performance and the characteristics of the loss function along the linear path in weight-space.

3.2 Results and Discussion

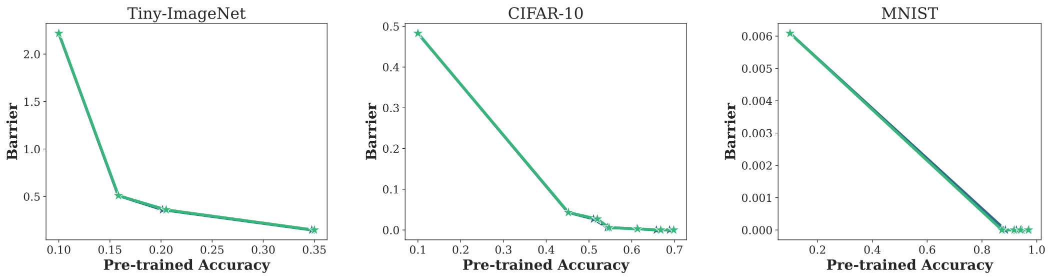
Loss Barrier and Pre-training Steps. We begin by considering the loss barrier as a function of k_1, the number of steps taken before the trajectories have split.



(a) Tiny-ImageNet (b) CIFAR-10 (c) MNIST

Figure 1: Train and Test cross entropy barriers for all three datasets when $k_2 = \text{max\_steps} - k_1$. All networks exhibits linear connectivity

Linear connectivity is observed for all datasets, even for pre-trained networks with lower accuracy levels than the final accuracy. While more challenging classification tasks display high interpolation gap at the early stages of training, i.e., smaller values of k_1, all networks eventually exhibit a form of linear connectivity. We observe almost identical train and test loss barriers. This can be attributed to the networks not overfitting the data, as train and test losses are roughly the same on all three cases. In the case of MNIST, linear connectivity is even observed at initialization (notice the scale in the Figure 1). Figure 1 shows that the train loss barrier decreases for larger values of k_1 and $k_2 = k_{\max} - k_1$.



(a) Tiny-ImageNet (b) CIFAR-10 (c) MNIST

Figure 2: Cross entropy barriers against the pre-trained accuracy. Each point in the plot is a model that was trained for $k_1 + k_2 = k_{\max}$ steps.

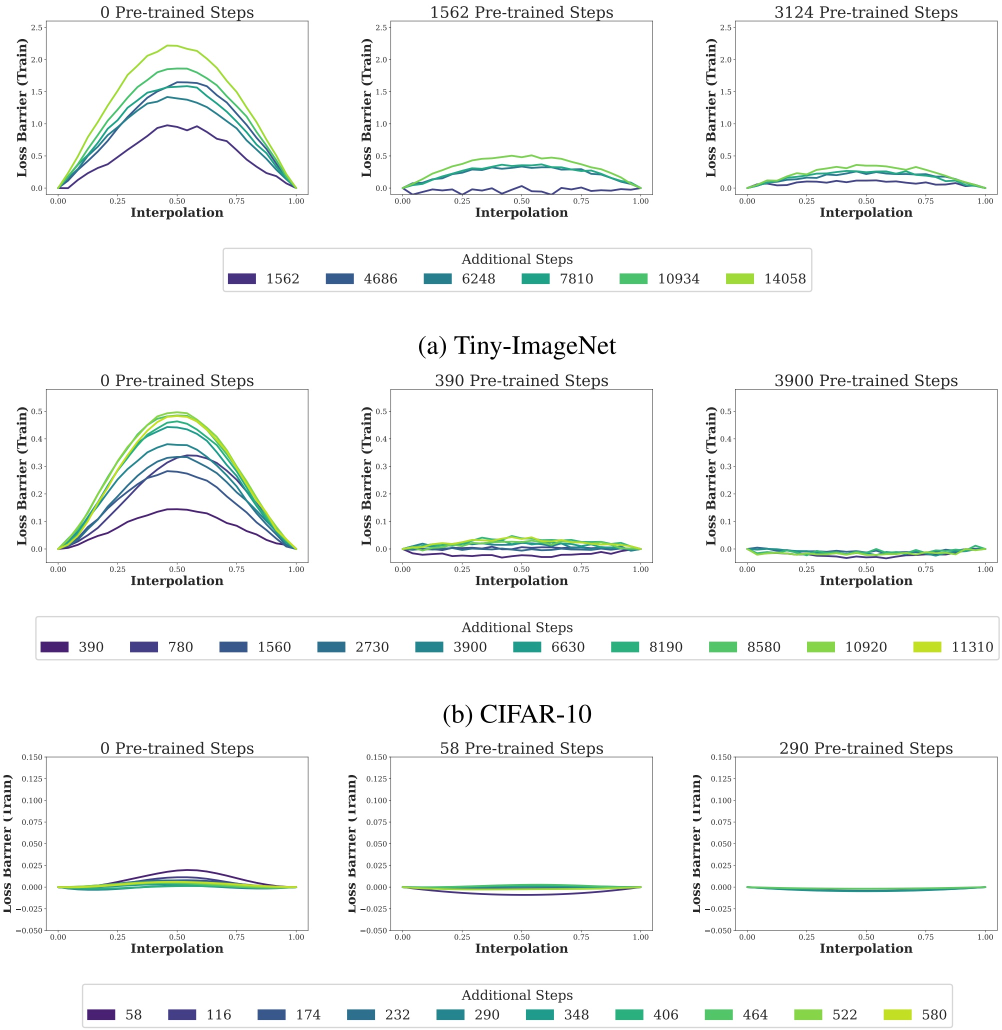
For CIFAR-10 and MNIST, the interpolation gap becomes negative at high values of k_1, indicating a convex loss landscape along the linear path between the two points (see the right column in Figure 3, see the loss landscape in Figure 5).

One might argue (as noted by Frankle et al. [2020]) that linear connectivity occurs due to the network converging during pre-training. However, we present two arguments against this notion. First, Figure 2 demonstrates that linear connectivity holds true even for pre-trained networks with comparatively lower accuracy levels than the final accuracy. For example, two networks originating from the same pre-trained model with approximately 50% accuracy on CIFAR-10 were trained to reach around 70% accuracy and exhibited linear connectivity. Second, the two models have a non-negligible difference in both weight-space (see Figure 4(a)) and prediction-space (the cross-entropy distance between the two train prediction vectors).

These observations suggest that even if a pre-trained model has a non-trivial, albeit not necessarily high, accuracy, two networks trained independently from it still exhibit linear connectivity. Despite the two trained copies learning the underlying function more effectively than the original pre-trained model and following distinct paths in the optimization landscape, they maintain their linear connection.

Our findings are consistent with those of Frankle et al. [2020] for different networks and partially different datasets. The observation that networks become linearly connected at relatively low accuracy levels supports the hypothesis that, at a certain point during training, the network learns a non-trivial representation of the data such that, with high probability over the sampling of the batches, the loss function restricted to the convex hull of possible paths SGD can take is convex.

Loss Barrier and Independent Steps. We now consider the relationship between the loss barrier and the number of independent steps taken during the training process. Notice that, assuming the gradients of the loss function $\nabla\mathcal{L}(\boldsymbol{w})$ are bounded by some constant, we might expect the barrier to behave like $O(k_2)$, where the O notation conceals constants that do not depend on k_2. Figure 3 plots the train interpolation gap for various networks with different values of k_1. We observe that, for low levels of k_1, the barrier increases with k_2 across all cases. However, when k_1 exceeds a certain threshold, the barrier remains constant at a low level, despite an increase in the number of independent updates. This suggests that, from a specific point in pre-training, the loss landscape between the two networks maintains linear connectivity, even after a considerable number of separate SGD steps.



(a) Tiny-ImageNet

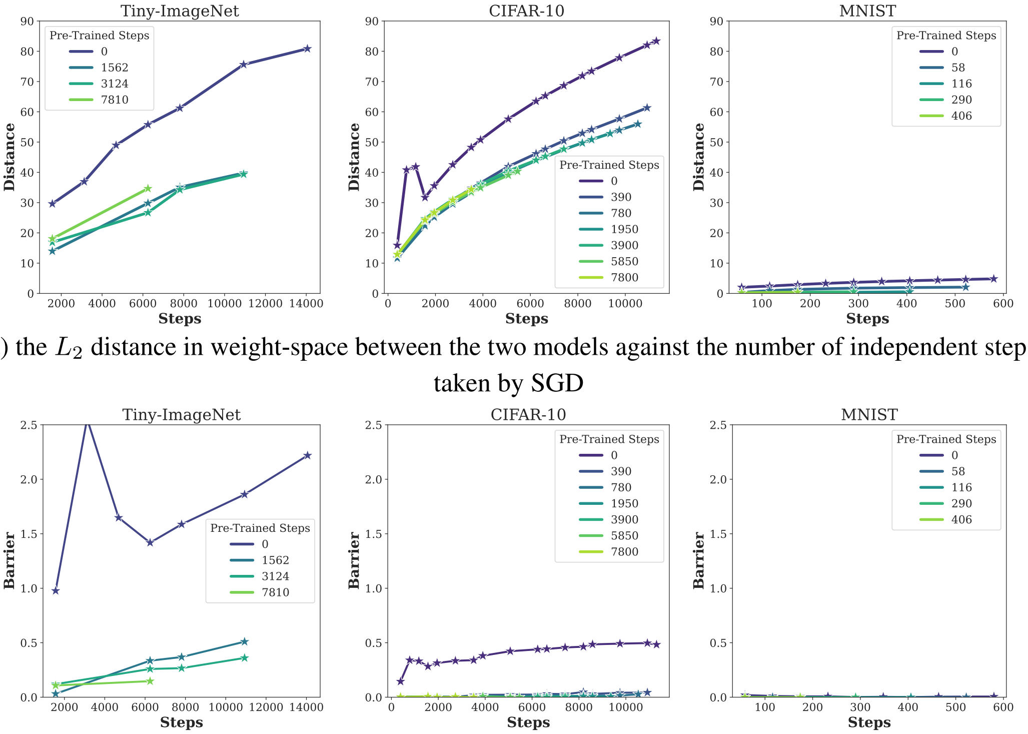
(b) CIFAR-10

(c) MNIST

Figure 3: Train interpolation gap along the linear path connecting two solutions found by SGD in weight space for different models with varying values of k_1 (the pre-trained steps). Brighter colors correspond to larger values of k_2 (the number of independent steps). The loss barrier is the maximum value of the curve, typically observed at $\alpha \approx 0.5$. All datasets exhibit a decrease in the error barrier as k_1 increases. Larger networks or more difficult classification tasks display larger loss barriers, with MNIST networks exhibiting negligible barriers even at $k_1 = 0$. Note the different y-scale in each row. For CIFAR-10 and MNIST, the interpolation gap becomes negative at high values of k_1, indicating a convex loss landscape along the linear path between the two points. The barrier increases with k_2 during the early stages of training.

Figure 4 demonstrates that as k_2 increases, the two networks diverge in weight-space and maintain significant distance from each other. Concurrently, the cross-entropy between the train prediction vectors of the two networks also diverges as k_2 grows. By evaluating the cross-entropy between each network's prediction vectors and averaging this metric across the entire dataset, we can discern that the networks become increasingly functionally dissimilar as k_2 expands.

Despite the observed divergence, linear connectivity is preserved in the cases of CIFAR-10 and MNIST. This observation suggests that LMC is not solely a characteristic of local convexity within a small neighborhood surrounding the next stochastic step. Instead, it signifies that the optimization problem, constrained by the path traversed in weight-space by SGD, enters an approximately convex basin at a specific stage during training, prior to reaching convergence.



(a) the L_2 distance in weight-space between the two models against the number of independent steps taken by SGD

(b) Plot of the Train Loss Barrier against the number of independent steps taken by SGD

Figure 4: Models originating from the same pre-trained model diverge in weight-space as the number of SGD steps increases. Despite the growing distance between the weight vectors, the loss barrier between them ceases to increase beyond a certain point during the pre-trained model's training. This observation suggests that the loss function, confined to the path traversed in weight-space by SGD, turns convex after reaching a specific stage in training, but before achieving convergence.

4 Conclusion

In this paper, we have explored the phenomenon of Linear Mode Connectivity (LMC) in deep neural networks. Our investigation focused on the behavior of the loss barrier between two SGD trajectories of the same length that branched off from a shared early portion of training with a common data order.

We have demonstrated that LMC is preserved even when trajectories split early in training and that it remains intact despite the divergence of the trajectories in both weight-space and prediction-space as we increase the number of independent steps.

This project is currently a work in progress, and our primary motivation is to delve deeper into the factors that underlie the very intriguing phenomenon of Linear Mode Connectivity. LMC is particularly interesting because it drastically reduces the complexity of the loss landscape in neural networks. In our study, we only considered SGD trajectories that branched off from a shared early portion of training. However, Entezari et al. [2021] conjectured that if permutation symmetries of hidden units are taken into account, all trajectories would lie within a single convex basin. Ainsworth et al. [2022] and Benzing et al. [2022] showed only partial support for this claim.

Additionally, I am interested in understanding the role overparameterization plays in LMC. Further research in this direction can provide valuable insights into the optimization process of deep neural networks.

References

Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint arXiv:2209.04836*, 2022.

Frederik Benzing, Simon Schug, Robert Meier, Johannes Von Oswald, Yassir Akram, Nicolas Zucchet, Laurence Aitchison, and Angelika Steger. Random initialisations performing above chance and how to find them. *arXiv preprint arXiv:2209.07509*, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In *International conference on machine learning*, pages 1309–1318. PMLR, 2018.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. *arXiv preprint arXiv:2110.06296*, 2021.

Stanislav Fort, Gintare Karolina Dziugaite, Mansheej Paul, Sepideh Kharaghani, Daniel M Roy, and Surya Ganguli. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. *Advances in Neural Information Processing Systems*, 33:5850–5861, 2020.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

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

C Daniel Freeman and Joan Bruna. Topology and geometry of half-rectified network optimization. *arXiv preprint arXiv:1611.01540*, 2016.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information processing systems*, 31, 2018.

Keller Jordan, Hanie Sedghi, Olga Saukh, Rahim Entezari, and Behnam Neyshabur. Repair: Renormalizing permuted activations for interpolation repair. *arXiv preprint arXiv:2211.08403*, 2022.

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

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

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

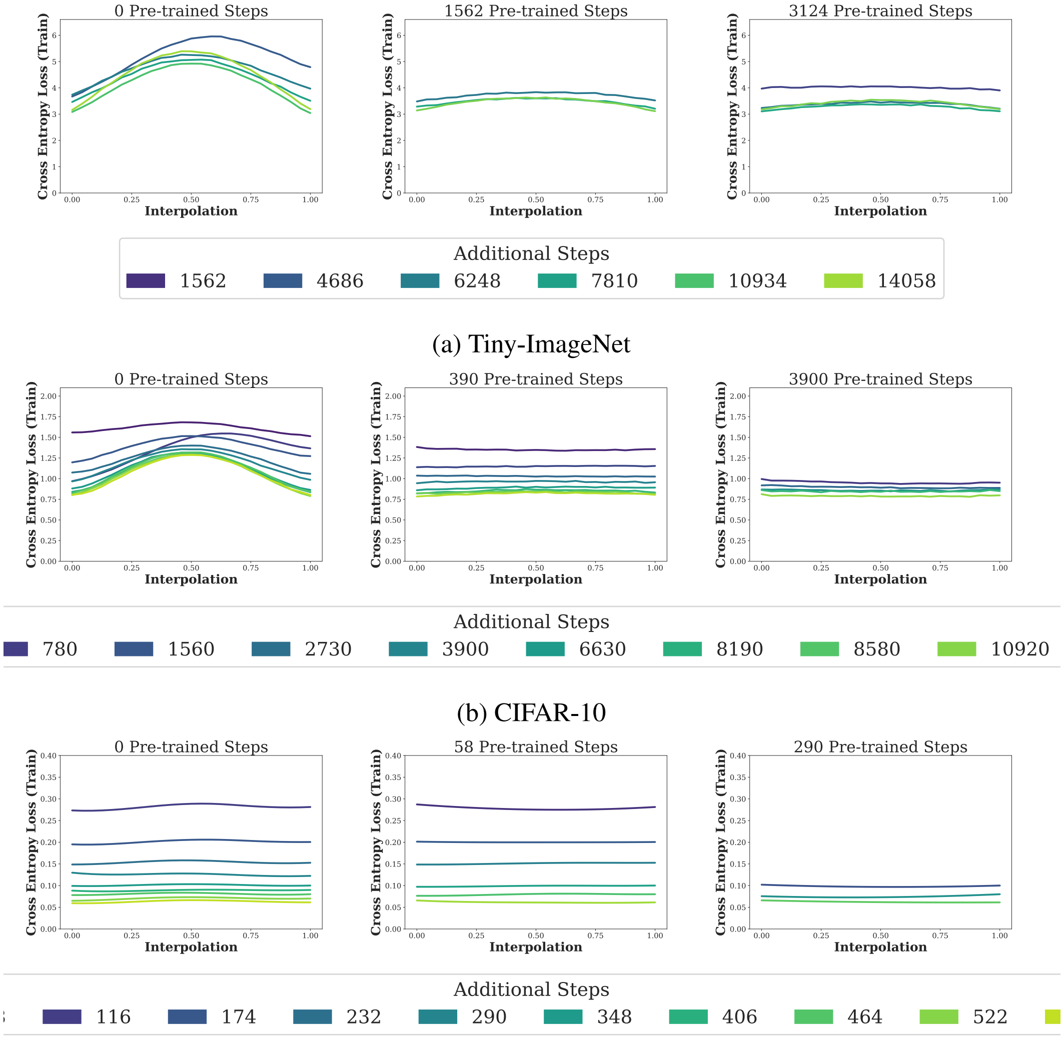
Vaishnavh Nagarajan and J Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Lee, and Laurent Massoulié. Optimal convergence rates for convex distributed optimization in networks. *Journal of Machine Learning Research*, 20:1–31, 2019.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

David Yunis, Kumar Kshitij Patel, Pedro Henrique Pamplona Savarese, Gal Vardi, Jonathan Frankle, Matthew Walter, Karen Livescu, and Michael Maire. On convexity and linear mode connectivity in neural networks. In *OPT 2022: Optimization for Machine Learning (NeurIPS 2022 Workshop)*, 2022.

A Additional Plots

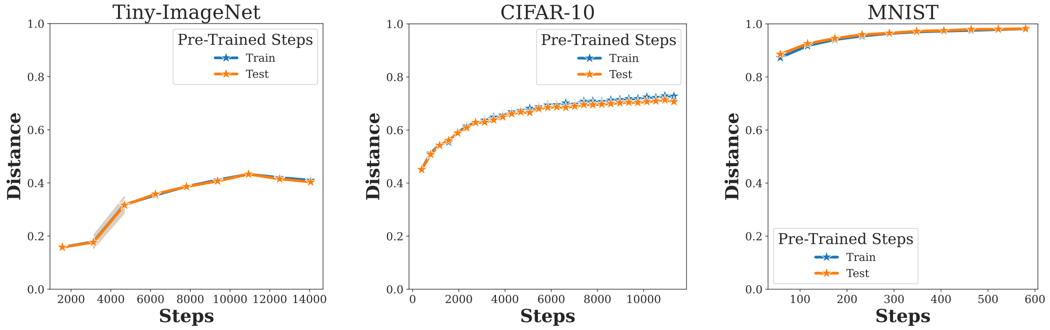


(a) Tiny-ImageNet

(b) CIFAR-10

(c) MNIST

Figure 5: Train loss landscape



(a) Tiny-ImageNet

(b) CIFAR-10

(c) MNIST

Figure 6: Train and Test accuracy on the ”main branch” of training (Algorithm 1)